Data Assimilation

Second Edition

Geir Evensen

Data Assimilation

The Ensemble Kalman Filter

Second Edition

Springer

Prof. Geir Evensen
Statoil Research Centre
PO box 7200
5020 Bergen
Norway

and

Mohn-Sverdrup Center for Global Ocean Studies
and Operational Oceanography
at Nansen Environmental and Remote Sensing Center
Thormølensgt 47
5600 Bergen
Norway
Geir.Evensen@gmail.com

ISBN 978-3-642-03710-8 e-ISBN 978-3-642-03711-5
DOI 10.1007/978-3-642-03711-5
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009933770

Cover design: deblik, Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Tina, Endre, and Linn

Preface to the second edition

The second edition of this book provides a more consistent presentation of the square root algorithm in Chap 13. The presentation in the first edition is less mature and there has been a significant development and enhanced understanding of the square root algorithm following the publication of the first edition.

A new chapter "Spurious correlations, localization, and inflation" is included and discusses and quantifies the impact of spurious correlations in ensemble filters caused by the use of a limited ensemble size. The chapter suggests and discusses inflation and localization methods for reducing the impact of spurious correlations and among others presents a new adaptive inflation algorithm.

The improved sampling algorithm in Chap. 11 is improved and takes into account the fact that sampling using too few singular vectors can lead to physically unrealistic and too smooth realizations.

The experiments in Chapters 13 and 14 are all repeated with the updated square root algorithms. In Chap. 14 a new section on the validity of the analysis equation, when using an ensemble representation of the measurement error covariance matrix, is included.

Finally the material in the Appendix is reorganized and the list of references is updated with many of the more recent publications on the EnKF.

I am greateful for the interaction and many discussions with Pavel Sakov and Laurent Bertino during the preparation of the second edition of this book.

Bergen, June 2009 *Geir Evensen*

Preface

The aim of this book is to introduce the formulation and solution of the data assimilation problem. The focus is mainly on methods where the model is allowed to contain errors and where the error statistics evolve through time. So-called strong constraint methods and simple methods where the error statistics are constant in time are only briefly explained, and then as special cases of more general weak constraint formulations.

There is a special focus on the Ensemble Kalman Filter and similar methods. These are methods which have become very popular, both due to their simple implementation and interpretation and their properties with nonlinear models.

The book has been written during several years of work on the development of data assimilation methods and the teaching of data assimilation methods to graduate students. It would not have been completed without the continuous interaction with students and colleagues, and I particularly want to acknowledge the support from Laurent Bertino, Kari Brusdal, François Counillon, Mette Eknes, Vibeke Haugen, Knut Arild Lisæter, Lars Jørgen Natvik, and Jan Arild Skjervheim, with whom I have worked closely for several years. Laurent Bertino and François Counillon also provided much of the material for the chapter on the TOPAZ ocean data assimilation system. Contributions from Laurent Bertino, Theresa Lloyd, Gordon Wilmot, Martin Miles, Jennifer Trittschuh-Vallès, Brice Vallès and Hans Wackernagel, on proof-reading parts of the final version of the book are also much appreciated.

It is hoped that the book will provide a comprehensive presentation of the data assimilation problem and that it will serve as a reference and textbook for students and researchers working with development and application of data assimilation methods.

Bergen, June 2006 *Geir Evensen*

Contents

List of symbols

a	De-correlation lengths in variogram models (11.2–11.4); Error in initial condition for scalar models (5.5), (5.22), (8.22) and (12.23)
$A(z)$	Vertical diffusion coefficient in Ekman model, Sect. 5.3
$A_0(z)$	First guess of vertical diffusion coefficient in Ekman model, Sect. 5.3
$\boldsymbol{a}$	Error in initial condition for vector models (5.70), (6.8–6.10), (7.2)
$\boldsymbol{A}_i$	Ensemble matrix at time t_i, Chap. 9
$\boldsymbol{A}$	Ensemble matrix, Chap. 9
$\boldsymbol{b}$	Vector of coefficients solved for in the analysis scheme (3.38)
$\boldsymbol{b}(\boldsymbol{x}, t)$	Error in boundary condition, Chap. 7
$\boldsymbol{b}_0$	Stochastic error in upper condition of Ekman model, Sect. 5.3
$\boldsymbol{b}_H$	Stochastic error in lower condition of Ekman model, Sect. 5.3
c	Constant in Fourier spectrum (11.10) and (11.14); constant multiplier used when simulating model errors (12.21)
c_i	Multiplier used when simulating model errors (12.55)
c_d	Wind-drag coefficient in Ekman model, Sect. 5.3
c_{d_0}	First guess value of wind-drag coefficient in Ekman model, Sect. 5.3
c_{rep}	Constant multiplier used when modelling model errors in representer method, Chap. 12
$C_{\psi\psi}$	Covariance of a scalar state variable ψ
$C_{c_d c_d}$	Covariance of error in wind-drag c_{d_0}
$C_{AA}(z_1, z_2)$	Covariance of error in vertical diffusion $A_0(z)$
$C_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$	Covariance of scalar field $\psi(\boldsymbol{x})$ (2.25)

$\boldsymbol{C}_{\psi\psi}$	Covariance of a discrete $\boldsymbol{\psi}$ (sometimes short for $\boldsymbol{C}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$)
$\boldsymbol{C}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$	Covariance of a vector of scalar state variables $\boldsymbol{\psi}(\boldsymbol{x})$
$C_{\epsilon\epsilon}$	Used for variance of ϵ in Chap. 3
C_{aa}	Scalar initial error covariance
C_{qq}	Model error covariance
$\boldsymbol{C}$	Matrix to be inverted in the ensemble analysis schemes, Chap. 9
$\boldsymbol{C}_{\epsilon\epsilon}$	Covariance of measurement errors $\boldsymbol{\epsilon}$
$\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$	Low-rank representation of measurement error covariance
$\boldsymbol{C}_{aa}$	Initial error covariance
$\boldsymbol{C}_{qq}$	Model error covariance
d	Measurement
$\boldsymbol{d}$	Vector of measurements
$\boldsymbol{D}$	Perturbed measurements, Chap. 9
$\boldsymbol{D}_j$	Perturbed measurements at data time j, Chap. 9
$\boldsymbol{E}$	Measurement perturbations, Chap. 9
$f(\boldsymbol{x})$	Arbitrary function, e.g. in (3.55)
$f(\psi)$	Probability density, e.g. $f(\psi)$ or $f(\boldsymbol{\psi})$ where $\boldsymbol{\psi}$ is a vector or a vector of fields
F	Distribution function (2.1)
$g(\boldsymbol{x})$	Arbitrary function, e.g. in (3.55) and (3.59)
G	Model operator for a scalar state; linear (4.1) or nonlinear (4.14)
$\boldsymbol{G}$	Model operator for a vector state; linear (4.11) or nonlinear (4.21), (9.1) and (7.1)
$h()$	Arbitrary function used at different occasions
H	Depth of bottom boundary in Ekman model, Sect. 5.3
$\boldsymbol{h}$	Innovation vector (3.51); spatial distance vector (11.1)
$i(j)$	Time index corresponding to measurement j, Fig. 7.1
$\boldsymbol{I}$	Identity matrix
J	Number of measurement times, Fig. 7.1
$\boldsymbol{k}$	Vertical unit vector $(0, 0, 1)$ in Ekman model Sect. 5.3; wave number $\boldsymbol{k} = (\kappa, \lambda)$, Chap. 11
$\boldsymbol{k}_{\mathrm{h}}$	Permeability, Chap. 17
$\boldsymbol{K}$	Kalman gain matrix (3.85)
m_j	Total number of measurements at measurement time j , Fig. 7.1
m	Sometimes used as abbreviation for m_j
$m(\psi)$	Nonlinear measurement functional in the Appendix

M	Total number of measurements over the assimilation interval
$\boldsymbol{M}$	Measurement matrix for a discrete state vector (3.76); measurement matrix operator (10.20)
n	Dimension of state vector $n = n_\psi + n_\alpha$, Chap. 9 and 10
n_α	Number of parameters, Chaps. 9 and 10
n_ψ	Dimension of model state, Chaps. 7–10
n_x	Gridsize in x-direction, Chap. 11
n_y	Gridsize in y-direction, Chap. 11
N	Sample or ensemble size
p	Error of first guess of a scalar or scalar field, Chap. 3; matrix rank, Chap. 14; probability (6.24)
p_A	Error of first guess vertical diffusion coefficient in Ekman model, Sect. 5.3
p_{c_d}	Error of first guess wind drag coefficient in Ekman model, Sect. 5.3
P	Reservoir pressure, Chap. 17
q	Stochastic error of scalar model used in Kalman filter formulations
$\boldsymbol{q}(i)$	Discrete model error at time t_i, (6.16)
$\boldsymbol{q}$	Stochastic error of vector model used in Kalman filter formulations
$\boldsymbol{Q}$	Ensemble of model noise, Sect. 11.4
r	De-correlation length in Fourier space (11.10)
r_1	De-correlation length in principal direction in Fourier space (11.11)
r_2	De-correlation length orthogonal to principal direction in Fourier space (11.11)
r_x	De-correlation length in principal direction in physical space (11.23)
r_y	De-correlation length orthogonal to principal direction in physical space (11.23)
$\boldsymbol{r}(\boldsymbol{x}, t)$	Vector of representer functions (3.39) and (5.48)
$\boldsymbol{r}$	Matrix of representers (3.80)
$\boldsymbol{R}$	Representer matrix (3.63)
R_s	Gas in a fluid state at reservoir conditions, Chap. 17
R_v	Oil in a gas state at reservoir conditions, Chap. 17
S_w	Water saturation, Chap. 17
S_g	Gas saturation, Chap. 17

S_o	Oil saturation, Chap. 17
$\boldsymbol{s}(\boldsymbol{x}, t)$	Vector of adjoints of representer functions (5.49)
$\boldsymbol{S}_j$	Measurement of ensemble perturbations at data time j, Chap. 9
$\boldsymbol{S}$	Measurement of ensemble perturbations, Chap. 9
t	Time variable
T	Final time of assimilation period for some examples
u	Dependent variable (5.99)
$\boldsymbol{u}(z)$	Horizontal velocity vector in Ekman model, Sect. 5.3
$\boldsymbol{u}_0(z)$	Initial condition for velocity vector in Ekman model, Sect. 5.3
$\boldsymbol{U}$	Left singular vectors from the singular value decomposition, Sect. 11.4 and (14.68)
$\boldsymbol{U}_0$	Left singular vectors from the singular value decomposition (14.19)
$\boldsymbol{U}_1$	Left singular vectors from the singular value decomposition (14.52)
$\boldsymbol{v}$	Dummy vector (5.101)
$\boldsymbol{V}$	Right singular vectors from the singular value decomposition, Sect. 11.4 and (14.68)
$\boldsymbol{V}_0$	Right singular vectors from the singular value decomposition (14.19)
$\boldsymbol{V}_1$	Right singular vectors from the singular value decomposition (14.52)
w_k	Random realization with mean equal to zero and variance equal to one (11.33)
W_{aa}	Inverse of scalar initial error covariance
$\boldsymbol{W}$	Matrix (14.63) and (14.64)
$\boldsymbol{W}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$	Functional inverse of $\boldsymbol{C}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, e.g. (3.27)
$\boldsymbol{W}_{aa}(\boldsymbol{x}_1, \boldsymbol{x}_2)$	Functional inverse of initial error covariance
$\boldsymbol{W}_{aa}$	Inverse of initial error covariance
$\boldsymbol{W}_{\eta\eta}$	Smoothing weight (6.19)
$\boldsymbol{W}_{\epsilon\epsilon}$	Matrix inverse of the covariance $\boldsymbol{C}_{\epsilon\epsilon}$
$\boldsymbol{x}$	Independent spatial variable
x_n	x-position in grid $x_n = n\Delta x$, Chap 11
$\boldsymbol{X}_0$	Matrix (14.26) and (14.51)
$\boldsymbol{X}_1$	Matrix (14.30) and (14.55)
$\boldsymbol{X}_2$	Matrix (14.34) and (14.59)
$x,\ y,\ z$	Dependent variables in Lorenz equations (6.5–6.6)

$\boldsymbol{x}$	Dependent variable $\boldsymbol{x}^{\mathrm{T}} = (x, y, z)$ in Lorenz equations
$\boldsymbol{x}_0$	Initial condition $\boldsymbol{x}_0^{\mathrm{T}} = (x_0, y_0, z_0)$ in Lorenz equations
y_m	y-position in grid $y_m = m\Delta y$, Chap 11
$\boldsymbol{Y}$	Matrix (14.65)
$\boldsymbol{Z}$	Matrix of eigenvectors from eigenvalue decomposition
$\boldsymbol{Z}_1$	Matrix of eigenvectors from eigenvalue decomposition (14.27)
$\boldsymbol{Z}_p$	Matrix of p first eigenvectors from eigenvalue decomposition (14.15)
$\mathcal{B}$	Penalty function in measurement space, e.g. $\mathcal{B}[\boldsymbol{b}]$ in (3.66)
$\mathcal{D}$	Model domain
$\partial\mathcal{D}$	Boundary of model domain
$\mathcal{H}$	Hamiltonian, used in hybrid Monte Carlo algorithm (6.25)
$\boldsymbol{\mathcal{H}}$	Hessian operator (second derivative of model operator)
$\mathcal{J}$	Penalty function, e.g. $\mathcal{J}[\boldsymbol{\psi}]$
$\mathcal{M}$	Scalar measurement functional (3.24)
$\boldsymbol{\mathcal{M}}$	Vector of measurement functionals
$\mathcal{N}$	Normal distribution
$\boldsymbol{\mathcal{P}}$	Matrix to be inverted in representer method (3.50)
α	Parameter in Sect. 11.4
α_1, α_2	Coefficients used in Chap. 3
α_{ij}	Coefficient used in (17.1)
$\boldsymbol{\alpha}(\boldsymbol{x})$	Poorly known model parameters to be estimated, Chap. 7
$\boldsymbol{\alpha}'(\boldsymbol{x})$	Errors in model parameters, Chaps. 7 and 10
β	Coefficient in Lorenz equations (6.7); constant (also β_{ini} and β_{mes}), Sect. 11.4
$\delta\psi$	Variation of ψ
ϵ	Real measurement error, Chap. 3
$\boldsymbol{\epsilon}_{\mathcal{M}}$	Representation errors in measurement operator, Sect. 5.2.3
$\boldsymbol{\epsilon}_d$	Actual measurement errors, Sect. 5.2.3
$\boldsymbol{\epsilon}$	Random or real measurement errors, Sect. 5.2.3
$\boldsymbol{\eta}$	Smoothing operators used in gradient method (6.19)
γ	Constant used in smoothing norm analysis (6.32); step length (6.22)
$\gamma(\boldsymbol{h})$	Variogram (11.1)
$\kappa_2(\boldsymbol{A})$	Condition number, Chap. 11
κ_l	Wave number in x direction, Chap. 11
λ_p	Wave number in y direction, Chap. 11

λ	Eigenvalue (13.18); scalar adjoint variable (5.37)
$\boldsymbol{\lambda}$	Vector adjoint variable
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues from eigenvalue decomposition
$\boldsymbol{\Lambda}_1$	Diagonal matrix of eigenvalues from eigenvalue decomposition (14.26)
$\boldsymbol{\Lambda}_p$	Diagonal matrix of eigenvalues from eigenvalue decomposition (14.14)
μ	Sample mean (2.20)
$\mu(\boldsymbol{x})$	Sample mean (2.23)
ω	Frequency variable (6.33)
ω_i	Unit variance noise process (8.10)
$\boldsymbol{\Omega}$	Error covariance of ω_i noise process (8.12)
ϕ	Scalar variable, Chap. 2
$\phi(\boldsymbol{x})$	Porosity in Chap. 17
$\phi_{l,p}$	Uniform random number (11.10)
Φ	Random scalar variable, Chap. 2
π	3.1415927
$\boldsymbol{\pi}$	Momentum variable used in hybrid Monte Carlo algorithm (6.25)
ψ	Scalar state variable (has covariance $C_{\psi\psi}$)
$\psi(\boldsymbol{x})$	Scalar state variable field (has error covariance $\boldsymbol{C}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$)
$\widehat{\psi}(\boldsymbol{k})$	Fourier transform of $\psi(\boldsymbol{x})$, Chap. 11
$\boldsymbol{\psi}$	Vector state variable, e.g. from a discretized $\psi(\boldsymbol{x})$ (has error covariance $\boldsymbol{C}_{\psi\psi}$)
$\boldsymbol{\psi}(\boldsymbol{x})$	Vector of scalar state variables (has error covariance $\boldsymbol{C}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$)
Ψ	Random scalar variable, Chap. 2
Ψ_0	Best guess initial condition for dynamical scalar models, may be function of $\boldsymbol{x}$
$\boldsymbol{\psi}(\boldsymbol{x})$	Vector of fields, sometimes written just $\boldsymbol{\psi}$
$\boldsymbol{\psi}_0$	Estimate of initial condition $\boldsymbol{\Psi}_0$, Chap. 7
$\boldsymbol{\psi}_b$	Estimate of boundary condition $\boldsymbol{\Psi}_b$, Chap. 7
$\boldsymbol{\Psi}$	Combined state vector, Chap. 10
$\boldsymbol{\Psi}_0$	Best guess initial condition
$\boldsymbol{\Psi}_b$	Best guess boundary condition, Chap. 7
ρ	Correlation parameter (11.33); coefficient in Lorenz equations (6.6)

$\boldsymbol{\Sigma}$	Matrix of singular values from the singular value decomposition, Sect. 11.4 and (14.68)
$\boldsymbol{\Sigma}_0$	Matrix of singular values from the singular value decomposition (14.19)
$\boldsymbol{\Sigma}_1$	Matrix of singular values from the singular value decomposition (14.52)
σ	Standard deviation defined in Chap. 2; used as coefficient in Lorenz equations (6.5); singular values, Sect. 11.4, Chap. 13 and Chap. 14
τ	De-correlation time (12.1)
θ	Pseudo temperature variable used in simulated annealing algorithm; rotation of principal direction (11.11)
$\boldsymbol{\Theta}$	Random rotation used in SQRT analysis scheme, Chap. 13
ξ	Random number used in Metropolis algorithm
$\boldsymbol{\xi}$	Coordinate running over boundary of model domain
$\mathbf{1}_N$	$N \times N$ matrix with all elements equal to 1
$\delta()$	Dirac delta function (3.24)
$\boldsymbol{\delta}_{\psi_i}$	Vector used to extract a component of the state vector, Chap. 7
$E[]$	Expectation operator
$\mathcal{O}()$	Order of magnitude function
$\Re$	Space of real numbers, e.g. $\Re^{n \times m}$ for a real $n \times m$ matrix

1

Introduction

Does the solution of a dynamical model with conditions have any statistical meaning or scientific purpose?

A model consists of a number of mathematical equations which are defined to represent the interaction between various variables through certain physical processes. In many cases the model excludes several processes or scales which are believed to have less importance for the applications at hand. Even if the model is a perfect representation of reality, its solution will not describe reality unless we have perfect knowledge about the initial and boundary conditions which are often difficult to prescribe with high accuracy.

From a single model integration we obtain a solution or realization without knowledge about its uncertainty. In fact, the model solution is just one out of infinitively many equally likely realizations. Thus, we should really consider the time evolution of the probability density function (pdf) for the model state. With knowledge of the pdf for the model state we can extract information about the most likely estimate of the model state as well as its uncertainty.

In many applications we have an approximate dynamical model with uncertain estimates of initial and boundary conditions. In addition we may have measurements of the model solution collected at different space and time locations. The computation of the pdf of the model solution conditioned on the measured observations defines the data assimilation or inverse problem considered in the following chapters.

The accurate representation of the full pdf becomes extremely expensive for high dimensional simulation models. Thus, data assimilation and inverse methods must normally represent the pdf using statistical moments or an ensemble of model states and then search for estimators such as the mean and maximum likelihood with the associated covariance representing uncertainty.

There is now a large class of different data assimilation and inverse methods which for practical and computational efficiency implement different statistical and conceptual approximations. The different methods have different properties which may depend on the dynamical system to which they are applied. Some methods will work well with linear dynamics but be completely

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_1,

useless for nonlinear dynamics. Other methods may handle nonlinearity well but computational requirements limit their use to low dimensional dynamical systems.

Parameter estimation in dynamical models is a field of research which has developed side by side with the developments in data assimilation. Traditionally one searches for a set of parameters in the model which results in a model solution that is consistent with a set of measurements. The methods used are in many cases strongly related to traditional data assimilation methods. Still there has been limited communication between the two communities. Statements like "one should not fiddle with model parameters but focus on the estimation of the state" has been followed by statements similar to "state estimation does not provide any scientific knowledge, what matters is to identify the parameters". So who should we trust?

This book aims to explain the fundamental data assimilation and inverse problem and the derivation and properties of the various methods which can be used to solve it. It may serve as a text book for students who take an introductory course in data assimilation and inverse methods, but is also intended as a reference book on the interpretation and implementation of advanced ensemble methods. The book has been organized with fairly basic discussions of traditional sequential and variational assimilation methods in the first chapters. This is followed by a more elaborate discussion of the fully nonlinear combined state and parameter estimation problem while the final part of the book is giving an extensive discussion on the practical implementation of ensemble methods.

Note also that much of the code used in the ensemble Kalman filter experiments is available from the EnKF home page:

`http://enkf.nersc.no`,

together with other information which is useful for the implementation of the EnKF.

The outline of the book is the following:

Chap. 2 summarizes basic statistical notation. This is just meant to be a quick reference and it does not give a complete introduction to the subject.

In Chap. 3 we consider the time independent inverse problem; i.e. given a first guess of a variable or model state and a set of measurements, what is the best estimate of the state given the prior estimate and the measurements. A linear unbiased variance minimizing analysis scheme is derived and shown to be the optimal solution as long as the prior error statistics are Gaussian.

In Chap. 4 we introduce the time evolution of the model state through a dynamical model and show how this problem can be solved using the Kalman Filter (KF), the Extended Kalman Filter (EKF) and the Ensemble Kalman Filter (EnKF). The methods rely on the analysis scheme derived in the previous chapter, and differ in the representation of error statistics and how this evolve in time. Simple examples are used to illustrate the properties of the

methods and we indicate issues related to the use of the methods with nonlinear dynamics.

Chap. 5 introduces the variational inverse problem. It discusses the implications of using the model as a weak or strong constraint but focus on the solution of the weak constraint problem. The Euler–Lagrange equations are derived and it is shown how they can be solved using the representer method.

In Chap. 6 the nonlinear variational inverse problem is considered. This may alternatively be solved using substitution methods like gradient descent. The different methods are used in simple examples which illustrate the properties of the nonlinear variational inverse problem.

Then in Chap. 7 we reformulate the data assimilation or inverse problem as a combined state and parameter estimation problem using Bayesian statistics. A fundamental result from this chapter is that if measurements at different times are independent, they can be processed sequentially in time. Thus, the Bayesian problem becomes a sequence of Bayesian subproblems. This result is exploited when deriving sequential data assimilation algorithms for the nonlinear assimilation problem in the following chapters.

In Chap. 8 the generalized inverse formulation is derived from the Bayesian formulation, and it is shown that the solution becomes the maximum likelihood estimator of the joint conditional pdf. Further, Euler–Lagrange equations for the generalized inverse are derived and they include the parameter estimation case which is solved in a simple illustration.

The ensemble methods are rederived in Chap. 9 starting from the Bayesian formulation. This leads to the Ensemble Smoother (ES) and the Ensemble Kalman Smoother (EnKS) as ensemble methods for solving the generalized inverse problem. The ensemble Kalman filter (EnKF) is then derived as a special case of the EnKS where information is only carried forward in time. Finally, the ensemble methods are examined in an example with the chaotic Lorenz equations.

In Chap. 10 a simple but nonlinear optimization problem is considered. It is shown that it can be solved using a statistical minimization method based on the EnKS. This example illustrates the impact of non-Gaussian statistics in the ensemble analysis update.

Chap. 11 discusses is some detail issues related to the sampling of ensemble realizations. It presents a simple methodology for generating random realizations of smooth pseudo-random fields with anisotropic covariance structure. An improved sampling scheme is presented which can be used to generate an ensemble with better rank properties, and experiments are presented which demonstrate the impact of ensemble size and the improved sampling algorithm.

Chap. 12 discusses the use of model errors and in particular the case of time correlated model errors. It includes a simple example illustrating how model errors, as well as model bias and model parameters, can be estimated using the EnKF and EnKS.

Recently developed square root schemes, which avoid the perturbation of measurements, are discussed in Chap. 13. The derivation of the square root schemes is discussed and it is shown that an additional randomization of the ensemble updates is still required. The square root schemes are evaluated and compared with results from the original EnKF scheme.

In Chap. 14 we discuss how it is possible to consistently compute the inversion in the different analyses schemes when the number of measurements is much larger than the number of ensemble members. This discussion leads to the final form of the EnKF analysis scheme where different pseudo-inversion schemes can be used in combination with either the traditional analysis update or the square root analysis. In particular the development of a sub-space inversion has lead to a very efficient algorithm which is useful even with very large data sets.

In Chap. 15 we evaluate the impact of spurious correlations caused by the use of a finite ensemble size. The actual magnutude of the spurious correlations is quantified and localization and inflation methods are demonstrated as tools that may be used to reduce the influence of spurious correlations.

An operational ocean prediction system, which is based on the EnKF, is presented in Chap. 16. The purpose is to illustrate what is really possible today using state of the art ocean circulation models together with advanced data assimilation schemes.

Another application, based on a reservoir simulation model, is given in Chap. 17, where both the model state and model parameters are estimated using the EnKF.

In Appendix A, some special issues related to the practical implementation of the EnKF, as well as ensemble methods in general, are given, including the use of nonlinear and non-synoptic measurements and the use of so-called time difference data.

Finally in Appendix B a chronological listing of previous publications related to the EnKF and other ensemble methods is included.

2

Statistical definitions

Basic statistical definitions which will be used in the following chapters are explained. The following is only meant to be a quick reference on statistical notation and definitions and more elaborate textbooks can be used for a comprehensive introduction to the subject.

2.1 Probability density function

Given a continuous random variable Ψ, we can associate a *distribution function* $F(\psi)$. This is also named the cumulative density function or probability distribution function, and it describes the probability that a realization of Ψ takes a value less than or equal to ψ. We can relate it to a continuous probability density function $f(\psi)$, through

$$F(\psi) = \int_{-\infty}^{\psi} f(\psi')d\psi', \tag{2.1}$$

thus $f(\psi)$, when it exists, is just the derivative of the distribution function

$$f(\psi) = \frac{\partial F(\psi)}{\partial \psi}. \tag{2.2}$$

The probability density function (pdf) gives the probability that a random variable Ψ will take a particular value ψ. If a probability distribution has density $f(\psi)$, then the infinitesimal interval $(\psi, \psi + d\psi)$ has probability $f(\psi)d\psi$.

The pdf must satisfy the conditions

$$f(\psi) \geq 0 \quad \text{for all } \psi, \tag{2.3}$$

which states that the probability for Ψ to take a value ψ, must be positive or zero, and

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_2,

$$\int_{-\infty}^{\infty} f(\psi)d\psi = 1, \tag{2.4}$$

that is, the probability of finding Ψ in the space of real numbers $\Re^1$, is equal to one.

Further, given $f(\psi)$, the probability that ψ takes a value in the interval $[\psi_a, \psi_b]$ is

$$\Pr\big(\Psi \in [\psi_a, \psi_b]\big) = \int_{\psi_a}^{\psi_b} f(\psi)d\psi. \tag{2.5}$$

The most common and useful distribution is the one called the *normal or Gaussian distribution*. It is defined by its *mean* and *variance* and has a bell shaped or Gaussian form. It represents a family of distributions of the same general form, characterized by their mean μ, and the variance σ^2. The *standard normal distribution* is a normal distribution with a mean of zero and a variance of one. The normal distribution has the pdf

$$f(\psi) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\psi-\mu)^2}{2\sigma^2}\right). \tag{2.6}$$

A convenient aspect of a normal population distribution is that the following empirical "rule of thumb" can be applied to the data: $\mu \pm \sigma$ spans approximately 68% of the realizations, $\mu \pm 2\sigma$ spans approximately 95% of the realizations, and $\mu \pm 3\sigma$ spans about 99% of the realizations.

The *joint pdf* describes the probability of two events together. Given two random variables Ψ and Φ we can define the joint pdf $f(\psi, \phi)$.

The *conditional pdf* describes the probability of some event Ψ, assuming the event Φ. The conditional pdf is denoted $f(\psi|\phi)$ which is read as the pdf for Ψ given Φ. It is often called the *posterior pdf*.

The *marginal pdf* is the pdf of one event, ignoring any information about the other event. It is obtained by integrating the joint pdf over the ignored event; e.g. the marginal pdf for Ψ is $f(\psi) = \int_{-\infty}^{\infty} f(\psi, \phi)d\phi$.

We also have that

$$f(\psi|\phi) = \frac{f(\psi, \phi)}{f(\phi)}, \tag{2.7}$$

or equivalently

$$f(\psi, \phi) = f(\psi|\phi)f(\phi) = f(\phi|\psi)f(\psi). \tag{2.8}$$

The variables Ψ and Φ are said to be independent if $f(\psi, \phi) = f(\psi)f(\phi)$.

From 2.8 we can write

$$f(\psi|\phi) = \frac{f(\psi)f(\phi|\psi)}{f(\phi)}. \tag{2.9}$$

This is Bayes' theorem which is a general result in probability theory giving the conditional probability distribution of a random variable Ψ given Φ in terms of the conditional probability distribution of variable Φ given Ψ, often named

the *likelihood*, and the marginal probability distribution of Ψ alone. In the context of Bayesian probability theory, the marginal probability distribution of Ψ alone is usually called the *prior* probability distribution or simply the prior. The conditional distribution of Ψ given the "data" Φ is called the *posterior* probability distribution or just the posterior. This is a general result and will be used extensively in the following chapters.

In this book we will in several occasions refer to and use Bayesian statistics to derive and explain data assimilation methods and their properties. In particular we will use a probability density function $f(\boldsymbol{\psi})$, for the event $\boldsymbol{\psi} \in \Re^n$. This is again related to the distribution function $F(\boldsymbol{\psi})$, of the random variable $\boldsymbol{\Psi} \in \Re^n$, through the equation

$$F(\psi_1, \ldots, \psi_n) = \int_{-\infty}^{\psi_1} \cdots \int_{-\infty}^{\psi_n} f(\psi'_1, \ldots, \psi'_n) d\psi'_1 \ldots d\psi'_n, \tag{2.10}$$

and the pdf is again defined as the derivative of the distribution function.

The pdf is a positive function of dimension n and it has the property that

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(\psi_1, \ldots, \psi_n) d\psi_1 \ldots d\psi_n = 1. \tag{2.11}$$

Thus, the probability that $\boldsymbol{\psi}$ is located somewhere in $\Re^n$ is one. For each value of $\boldsymbol{\psi}$, $f(\boldsymbol{\psi})$ gives the probability for this particular state. The pdf $f(\boldsymbol{\psi})$ is also named the joint pdf for $(\psi_1, \ldots, \psi_n)$.

This joint pdf can be factorized into

$$f(\psi_1, \ldots, \psi_n) = f(\psi_1) f(\psi_2|\psi_1) f(\psi_3|\psi_1, \psi_2) \cdots f(\psi_n|\psi_1, \ldots, \psi_{n-1}). \tag{2.12}$$

Here $f(\psi_2|\psi_1)$ is the likelihood of ψ_2 given ψ_1, and if $n = 2$ we get just $f(\psi_1, \psi_2) = f(\psi_1) f(\psi_2|\psi_1)$, which is interpreted as the probability of ψ_1 times the likelihood of ψ_2 given ψ_1.

If the events, $(\psi_1, \ldots, \psi_n)$ are independent we can write

$$f(\psi_1, \ldots, \psi_n) = f(\psi_1) f(\psi_2) \cdots f(\psi_n). \tag{2.13}$$

We will make frequent use of the pdf of a model state $\boldsymbol{\psi}$, and the likelihood function for a vector of measurements $\boldsymbol{d}$, of the state which is written as $f(\boldsymbol{d}|\boldsymbol{\psi})$. The joint pdf of the state and the measurements can be written

$$f(\boldsymbol{\psi}, \boldsymbol{d}) = f(\boldsymbol{\psi}) f(\boldsymbol{d}|\boldsymbol{\psi}) = f(\boldsymbol{d}) f(\boldsymbol{\psi}|\boldsymbol{d}), \tag{2.14}$$

and we must have

$$f(\boldsymbol{\psi}|\boldsymbol{d}) = \frac{f(\boldsymbol{\psi}) f(\boldsymbol{d}|\boldsymbol{\psi})}{f(\boldsymbol{d})}, \tag{2.15}$$

where the denominator is just the integral of the numerator, which normalizes the numerator such that the expression integrates to one. This is Bayes' theorem, and in this context it states that the pdf of the model state given a set of measurements is proportional to the pdf of the model state times the likelihood function for the measurements.

2.2 Statistical moments

The probability density function $f(\psi)$, contains a huge amount of information, especially for high dimensional systems, and actually much more information than is normally needed. Instead of working with the full density it is often convenient to define statistical moments of the density. These are defined from the general expression of the expected value of a function $h(\Psi)$,

$$E[h(\Psi)] = \int_{-\infty}^{\infty} h(\psi) f(\psi) d\psi. \tag{2.16}$$

2.2.1 Expected value

The expected value of a random variable Ψ with distribution $f(\psi)$, is defined as

$$\mu = E[\Psi] = \int_{-\infty}^{\infty} \psi f(\psi) d\psi. \tag{2.17}$$

The expected value (or expectation) of a random variable represents the average one "expects" if an infinite number of samples are drawn from the distribution. Note that the value itself may not be expected in the general sense, it may be unlikely or even impossible, dependent on the shape of $f(\psi)$.

2.2.2 Variance

If Ψ is a random variable, the variance is given by

$$\begin{aligned} \sigma^2 = E\Big[\big(\Psi - E[\Psi]\big)^2\Big] &= \int_{-\infty}^{\infty} \big(\psi - E[\Psi]\big)^2 f(\psi) d\psi \\ &= E\big[\Psi^2\big] - E\big[\Psi\big]^2. \end{aligned} \tag{2.18}$$

That is, it is the expected value of the square of the deviation of Ψ from its own mean. In other words, it is the average of the square of the distance of each data point from the mean. It is thus the mean squared deviation. The second line in 2.18 is often used for the practical computation of the variance. It is just the second moment minus the square of the first moment.

An inconvenience is that the variance has a unit which is the square of the data unit. For this reason it is common to use the square root of the variance which is named the *standard deviation*, denoted σ. It can also easily be shown that the variance does not depend on the mean, thus the variance of $\Psi + b$ is the same as the variance of Ψ. On the other hand the variance of $a\Psi$ is $a^2\sigma^2$.

2.2.3 Covariance

Given two random variables Ψ and Φ and their respective probability density functions $f(\psi)$ and $f(\phi)$, from which we can define the joint probability $f(\psi, \phi) = f(\psi|\phi)f(\phi) = f(\phi|\psi)f(\psi)$, their covariance is defined as

$$\begin{aligned} E\Big[\big(\Psi - E[\Psi]\big)\big(\Phi - E[\Phi]\big)\Big] \\ &= \iint_{-\infty}^{\infty} \big(\psi - E[\Psi]\big)\big(\phi - E[\Phi]\big) f(\psi, \phi) d\psi d\phi \\ &= \iint_{-\infty}^{\infty} \psi\phi f(\psi, \phi) d\psi d\phi - E[\Psi]E[\Phi]. \end{aligned} \tag{2.19}$$

Note that the same conditions (2.3) and (2.4) also apply for $f(\psi, \phi)$. In the case when the random variables Ψ and Φ are independent, $f(\psi, \phi) = f(\psi)f(\phi)$ and the covariance becomes zero.

2.3 Working with samples from a distribution

Clearly when the dimension of a probability function increases to more than about 3–4 it becomes very impractical, if not impossible, to evaluate the integrals by numerical integration on a regular grid. Suppose the dimension is 10 and we need 10 grid points in each direction to have a proper representation of the density. A grid with 10^{10} nodes would then have to be stored which would require 40 Giga bytes of storage and 10^{10} additions would be needed to calculate the integral.

Fortunately there is an alternative to the direct numerical integration which often works very well even for high dimensional systems. The approach is called the Markov Chain Monte Carlo (MCMC) methods, (see e.g. *Robert and Casella*, 2004), and assumes that we have available a large number N, of realizations from the distribution $f(\psi)$.

2.3.1 Sample mean

Having a sample of independent realizations from the distribution $f(\psi)$, i.e. ψ_i, for $i = 1, N$, then the sample mean $\overline{\psi}$, is given by

$$\mu = E[\psi] \simeq \overline{\psi} = \frac{1}{N} \sum_{i=1}^{N} \psi_i. \tag{2.20}$$

The "expected value" terminology is meant to connote that $E[\Psi]$ is, in some sense, the "best guess" as to the possible outcome of Ψ, or said in another way; the expected value is the value we expect to obtain if infinitely many data are present, and the sample mean of these is computed. This is a reason why $E[\Psi]$ is often called the mean of Ψ.

2.3.2 Sample variance

The variance can be calculated from the formula

$$\begin{aligned}\sigma^2 &= E\Big[\big(\Psi - E[\Psi]\big)^2\Big] \\ &\simeq \overline{\big(\psi - \overline{\psi}\big)^2} = \frac{1}{N-1}\sum_{i=1}^{N}\big(\psi_i - \overline{\psi}\big)^2,\end{aligned} \tag{2.21}$$

where the denominator $N-1$ is used instead of N to ensure that the formula (2.21) becomes an unbiased estimator for the variance.

2.3.3 Sample covariance

The covariance can be calculated from the formula

$$\begin{aligned}\mathrm{Cov}(\psi,\phi) &= E\Big[\big(\Psi - E[\Psi]\big)\big(\Phi - E[\Phi]\big)\Big] \\ &\simeq \overline{\big(\psi - \overline{\psi}\big)\big(\phi - \overline{\phi}\big)} = \frac{1}{N-1}\sum_{i=1}^{N}\big(\psi_i - \overline{\psi}\big)\big(\phi_i - \overline{\phi}\big).\end{aligned} \tag{2.22}$$

2.4 Statistics of random fields

Of special interest for us will be the statistics of so-called random fields $\Psi(\boldsymbol{x})$ where Ψ is now a function of $\boldsymbol{x} = (x, y, z, \ldots)$.

2.4.1 Sample mean

Having an ensemble of independent samples from the distribution $f(\psi(\boldsymbol{x}))$, i.e. $\psi_i(\boldsymbol{x})$, for $i = 1, N$, then the sample mean is given by

$$\mu(\boldsymbol{x}) \simeq \overline{\psi(\boldsymbol{x})} = \frac{1}{N}\sum_{i=1}^{N}\psi_i(\boldsymbol{x}). \tag{2.23}$$

2.4.2 Sample variance

The sample variance of an ensemble of independent samples from the distribution $f(\psi(\boldsymbol{x}))$, is given as

$$\sigma^2(\boldsymbol{x}) \simeq \overline{\big(\psi(\boldsymbol{x}) - \overline{\psi(\boldsymbol{x})}\big)^2} = \frac{1}{N-1}\sum_{i=1}^{N}\big(\psi_i(\boldsymbol{x}) - \overline{\psi(\boldsymbol{x})}\big)^2. \tag{2.24}$$

2.4.3 Sample covariance

The covariance between two different locations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for the random fields are given by

$$\begin{aligned} C_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2) &\simeq \overline{\big(\psi(\boldsymbol{x}_1) - \overline{\psi(\boldsymbol{x}_1)}\,\big)\big(\psi(\boldsymbol{x}_2) - \overline{\psi(\boldsymbol{x}_2)}\,\big)} \\ &= \frac{1}{N-1}\sum_{j=1}^{N}\big(\psi_j(\boldsymbol{x}_1) - \overline{\psi(\boldsymbol{x}_1)}\,\big)\big(\psi_j(\boldsymbol{x}_2) - \overline{\psi(\boldsymbol{x}_2)}\,\big). \end{aligned} \tag{2.25}$$

Note that if $\boldsymbol{x}_1 = \boldsymbol{x}_2$, then (2.25) reduces to the definition of variance.

The covariance of Ψ between the two locations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ defines how values of Ψ, at different locations, are "varying together" or "covarying" . For example, if the random fields Ψ are smooth we will expect that neighboring points are correlated or covarying. The covariance can therefore be a measure of smoothness.

2.4.4 Correlation

The correlation between the random variables $\Psi(\boldsymbol{x}_1)$ and $\Psi(\boldsymbol{x}_2)$ is defined by

$$\mathrm{Cor}\big(\psi(\boldsymbol{x}_1), \psi(\boldsymbol{x}_2)\big) = \frac{\mathrm{C}(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\sigma(\boldsymbol{x}_1)\sigma(\boldsymbol{x}_2)}. \tag{2.26}$$

Thus, the correlation is just a normalized covariance.

2.5 Bias

One meaning is involved in what is called a biased sample; if some elements are more likely to be chosen in the sample than others, and those have a higher/lower value of the quantity being estimated, the outcome will be higher/lower than the true value.

Another kind of bias in statistics does not involve biased samples, but rather the use of a statistics whose average value differs from the value of the quantity being estimated. Suppose we are trying to estimate the true value ψ^t of a parameter ψ using an estimator $\hat{\psi}$ (that is, some function of the observed data). Then the bias of $\hat{\psi}$ is defined to be

$$E\big[\hat{\psi}\big] - \psi^t. \tag{2.27}$$

In words, this would be "the expected value of the estimator $\hat{\psi}$ minus the true value ψ^t". This may be rewritten as

$$E\big[\hat{\psi} - \psi^t\big], \tag{2.28}$$

which would read "the expected value of the difference between the estimator and the true value".

An example of a biased estimator of variance is

$$\sigma^2_{\text{biased}} = \frac{1}{N}\sum_{i=1}^{N}\left(\psi_i - \overline{\psi}\right)^2, \tag{2.29}$$

which differs from the formula (2.21) by the division by N rather than $N-1$. The proof that this is a biased estimator of the variance is left as an exercise.

2.6 Central limit theorem

The central limit theorem can be used to say something about the convergence of the moments of a sample with increasing sample size.

Assume that we draw a number of samples of the random variable Ψ, each with sample size N. We then have the following:

- The sample mean $\mu(\psi)$ from (2.23), computed from the different samples is normally distributed, independent of the distribution for Ψ.
- The standard deviation of $\mu(\psi)$ as computed from the different samples tends towards $\sigma(\Psi)/\sqrt{N}$.

Thus, if we compute the sample mean from a given sample, we can expect that the error in the computed sample mean is normally distributed and given by $\sigma(\Psi)/\sqrt{N}$. Importantly, the error decreases proportional to $1/\sqrt{N}$.

The amazing and counter-intuitive property of the central limit theorem is that no matter what the shape of the original distribution, the sampling distribution of the mean approaches a normal distribution. Furthermore, for most distributions, a normal distribution is approached very quickly as N increases.

3

Analysis scheme

This chapter discusses the problem of how to combine a model prediction of a state variable at a given time with a set of measurements available at this particular time. It is assumed that error statistics of the model prediction as well as the measurements are known and characterized by the respective error covariances. Based on this information the so-called analysis scheme used in linear data assimilation methods is presented in some detail. First the theory is derived for the scalar case and then it is extended to the case with a spatial dimension. An extensive analysis of the properties of the analysis scheme is given and this introduces notation and concepts which are also valid for the time dependent problems treated in the following chapters.

3.1 Scalar case

We start by deriving the optimal linear and unbiased estimator for a scalar state variable combined with a single measurement.

3.1.1 State-space formulation

Given two different estimates of the true state ψ^{t} (e.g. a temperature at a particular location and time):

$$\psi^{\mathrm{f}} = \psi^{\mathrm{t}} + p^{\mathrm{f}}, \tag{3.1}$$

$$d = \psi^{\mathrm{t}} + \epsilon, \tag{3.2}$$

where ψ^{f} may be a model forecast or a first-guess estimate and d is a measurement of ψ^{t}. The term p^{f} denotes the unknown error in the forecast and ϵ is the unknown measurement error. The problem is now, to find an improved analyzed estimate ψ^{a} of ψ^{t}. Thus, additional information about the error terms must be supplied and we make the following assumptions:

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_3,

$$\begin{aligned} \overline{p^{\mathrm{f}}} &= 0, & \overline{(p^{\mathrm{f}})^2} &= C^{\mathrm{f}}_{\psi\psi}, \\ \overline{\epsilon} &= 0, & \overline{(\epsilon)^2} &= C_{\epsilon\epsilon}, \\ \overline{\epsilon p^{\mathrm{f}}} &= 0. \end{aligned} \tag{3.3}$$

Here the overbar denotes ensemble averaging or expected value.

We now seek a linear estimator

$$\psi^{\mathrm{a}} = \psi^{\mathrm{t}} + p^{\mathrm{a}} = \alpha_1 \psi^{\mathrm{f}} + \alpha_2 d, \tag{3.4}$$

where we define

$$\overline{p^{\mathrm{a}}} = 0, \qquad \overline{(p^{\mathrm{a}})^2} = C^{\mathrm{a}}_{\psi\psi}. \tag{3.5}$$

The definition (3.5) means that we assume that the error p^{a}, in the analyzed estimate is unbiased. Thus, the analyzed estimate itself becomes an unbiased estimate of the true state ψ^{t}, i.e. $\overline{\psi^{\mathrm{a}}} = \psi^{\mathrm{t}}$.

Inserting the estimates (3.1) and (3.2) in (3.4) we get

$$\psi^{\mathrm{t}} + p^{\mathrm{a}} = \alpha_1(\psi^{\mathrm{t}} + p^{\mathrm{f}}) + \alpha_2(\psi^{\mathrm{t}} + \epsilon). \tag{3.6}$$

The expectation of this equation is

$$\psi^{\mathrm{t}} = \alpha_1 \psi^{\mathrm{t}} + \alpha_2 \psi^{\mathrm{t}} = (\alpha_1 + \alpha_2)\psi^{\mathrm{t}}. \tag{3.7}$$

Thus, we must have

$$\alpha_1 + \alpha_2 = 1, \qquad \text{or} \qquad \alpha_1 = 1 - \alpha_2, \tag{3.8}$$

and a linear unbiased estimator for ψ^{t} is given as

$$\begin{aligned} \psi^{\mathrm{a}} &= (1 - \alpha_2)\psi^{\mathrm{f}} + \alpha_2 d \\ &= \psi^{\mathrm{f}} + \alpha_2(d - \psi^{\mathrm{f}}). \end{aligned} \tag{3.9}$$

Using (3.1), (3.2) and (3.4) in this equation gives an expression for the error in the analysis

$$p^{\mathrm{a}} = p^{\mathrm{f}} + \alpha_2(\epsilon - p^{\mathrm{f}}). \tag{3.10}$$

The error variance is then using (3.3)

$$\begin{aligned} \overline{(p^{\mathrm{a}})^2} = C^{\mathrm{a}}_{\psi\psi} &= \overline{(p^{\mathrm{f}} + \alpha_2(\epsilon - p^{\mathrm{f}}))^2} \\ &= \overline{(p^{\mathrm{f}})^2} + 2\alpha_2 \overline{p^{\mathrm{f}}(\epsilon - p^{\mathrm{f}})} + \alpha_2^2 \overline{\epsilon^2 - 2\epsilon p^{\mathrm{f}} + (p^{\mathrm{f}})^2} \\ &= C^{\mathrm{f}}_{\psi\psi} - 2\alpha_2 C^{\mathrm{f}}_{\psi\psi} + \alpha_2^2 (C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}), \end{aligned} \tag{3.11}$$

and the minimum variance is defined by

$$\mathrm{d}C^{\mathrm{a}}_{\psi\psi}\alpha_2 = -2C^{\mathrm{f}}_{\psi\psi} + 2\alpha_2(C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}) = 0. \tag{3.12}$$

Solving for α_2 gives

$$\alpha_2 = \frac{C^{\mathrm{f}}_{\psi\psi}}{C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}}, \tag{3.13}$$

and the analyzed estimate becomes

$$\psi^{\mathrm{a}} = \psi^{\mathrm{f}} + \frac{C^{\mathrm{f}}_{\psi\psi}}{C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}}\big(d - \psi^{\mathrm{f}}\big). \tag{3.14}$$

Further, the error variance of the analyzed estimate is now from (3.11) and (3.13)

$$\begin{aligned} C^{\mathrm{a}}_{\psi\psi} &= C^{\mathrm{f}}_{\psi\psi} - 2\frac{C^{\mathrm{f}}_{\psi\psi}}{C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}} C^{\mathrm{f}}_{\psi\psi} + \left(\frac{C^{\mathrm{f}}_{\psi\psi}}{C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}}\right)^2 \big(C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}\big) \\ &= C^{\mathrm{f}}_{\psi\psi} - \frac{(C^{\mathrm{f}}_{\psi\psi})^2}{C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}} = C^{\mathrm{f}}_{\psi\psi}\left(1 - \frac{C^{\mathrm{f}}_{\psi\psi}}{C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}}\right). \end{aligned} \tag{3.15}$$

3.1.2 Bayesian formulation

Given a probability density function $f(\psi)$ for the first-guess estimate ψ^{f}, and a likelihood function $f(d|\psi)$ for the measurement d; then, from Chap. 2 we have Bayes' theorem

$$f(\psi|d) \propto f(\psi) f(d|\psi). \tag{3.16}$$

Thus, the posterior density for ψ given the measurement d, is proportional to the product of the prior density for ψ times the likelihood function for the measurement d.

Again consider the two estimates (3.1) and (3.2) of the true state ψ^{t}. In the case with Gaussian statistics we can define the prior and likelihood as

$$f(\psi) \propto \exp\Big(-\frac{1}{2}\left(\psi - \psi^{\mathrm{f}}\right)\left(C^{\mathrm{f}}_{\psi\psi}\right)^{-1}\left(\psi - \psi^{\mathrm{f}}\right)\Big) \tag{3.17}$$

and

$$f(d|\psi) \propto \exp\Big(-\frac{1}{2}(\psi - d)C^{-1}_{\epsilon\epsilon}(\psi - d)\Big). \tag{3.18}$$

Thus, the posterior density can be written as

$$f(\psi|d) \propto \exp\Big(-\frac{1}{2}\mathcal{J}[\psi]\Big), \tag{3.19}$$

where

$$\mathcal{J}[\psi] = \left(\psi - \psi^{\mathrm{f}}\right)\left(C^{\mathrm{f}}_{\psi\psi}\right)^{-1}\left(\psi - \psi^{\mathrm{f}}\right) + (\psi - d)C^{-1}_{\epsilon\epsilon}(\psi - d). \tag{3.20}$$

The least squares solution ψ^{a}, that gives a minimum for $\mathcal{J}$, also gives a maximum of $f(\psi|d)$, i.e. it is the maximum likelihood estimate. This will always be true as long as all the error terms are normally distributed.

The minimum value of $\mathcal{J}$ is found from

$$\mathrm{d}\mathcal{J}\psi = 2\left(\psi - \psi^{\mathrm{f}}\right)\left(C^{\mathrm{f}}_{\psi\psi}\right)^{-1} + 2\left(\psi - d\right)C^{-1}_{\epsilon\epsilon} = 0. \tag{3.21}$$

Solving for ψ gives again the result ψ^{a} in (3.14), thus, the minimum variance estimate is also the maximum likelihood estimate in the case with Gaussian priors.

3.2 Extension to spatial dimensions

Now we extend the discussion to involve a variable $\psi^{\mathrm{f}}(\boldsymbol{x})$, with a spatial dimension which may be one or larger, e.g. $\boldsymbol{x} = (x, y, z)$ for a three dimensional space. In the following discussion we adopt the notation used by *Bennett* (1992) who gave a similar derivation for the time dependent problem.

3.2.1 Basic formulation

Assume now a multidimensional variable (e.g. a temperature field), and a vector of measurements $\boldsymbol{d} \in \Re^M$, which is related to the true state through the measurement functional $\boldsymbol{\mathcal{M}} \in \Re^M$, with M being the number of measurements:

$$\psi^{\mathrm{f}}(\boldsymbol{x}) = \psi^{\mathrm{t}}(\boldsymbol{x}) + p^{\mathrm{f}}(\boldsymbol{x}), \tag{3.22}$$

$$\boldsymbol{d} = \boldsymbol{\mathcal{M}}[\psi^{\mathrm{t}}(\boldsymbol{x})] + \boldsymbol{\epsilon}. \tag{3.23}$$

The term $p^{\mathrm{f}}(\boldsymbol{x})$ is the error in the first-guess field $\psi^{\mathrm{f}}(\boldsymbol{x})$, relative to the truth $\psi^{\mathrm{t}}(\boldsymbol{x})$. Further, we have defined the vector of measurement errors $\boldsymbol{\epsilon} \in \Re^M$. The measurement errors may be a composite of errors introduced when measuring the variable and additional representation errors introduced when constructing the measurement functional. This will be discussed in more detail in the following chapters.

As an example of a measurement functional, a direct measurement would be represented by a functional of the form

$$\mathcal{M}_i[\psi(\boldsymbol{x})] = \int_{\mathcal{D}} \psi(\boldsymbol{x})\delta(\boldsymbol{x} - \boldsymbol{x}_i)d\boldsymbol{x} = \psi(\boldsymbol{x}_i), \tag{3.24}$$

where $\boldsymbol{x}_i$ is the measurement location, $\delta(\boldsymbol{x}-\boldsymbol{x}_i)$ is the Dirac delta function, and the subscript i denotes the component i of the measurement functional. Note that in some of the following equations we will use a subscript on the vector form of the measurement functional, e.g. $\boldsymbol{\mathcal{M}}_{(3)}[\delta\psi(\boldsymbol{x}_3)]$ which just denote that the integration is performed on the dummy variable $\boldsymbol{x}_3$ rather than $\boldsymbol{x}$ as is used in (3.24).

The actual values of the errors $p^{\mathrm{f}}(\boldsymbol{x})$ and $\boldsymbol{\epsilon}$ are not known. Thus, to make progress, a statistical hypothesis must be used, and we make the following assumptions:

$$\begin{aligned} \overline{p^{\mathrm{f}}(\boldsymbol{x})} &= 0, & \overline{p^{\mathrm{f}}(\boldsymbol{x}_1)p^{\mathrm{f}}(\boldsymbol{x}_2)} &= C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2), \\ \overline{\boldsymbol{\epsilon}} &= \boldsymbol{0}, & \overline{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}} &= \boldsymbol{C}_{\epsilon\epsilon}, \\ \overline{p^{\mathrm{f}}(\boldsymbol{x})\boldsymbol{\epsilon}} &= \boldsymbol{0}. \end{aligned} \tag{3.25}$$

Thus, the means of the errors in the first-guess and the measurements are zero, and there are no cross correlations between these error terms. Further, we have knowledge of the forecast or first-guess error covariance between two points in space $C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)$, and the observation error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon} \in \Re^{M\times M}$. Note that the error covariance differs from the sample covariance as defined in (2.22) by referring to the true (unknown) state rather than the sample average.

We are now defining a variational functional

$$\begin{aligned} \mathcal{J}[\psi] = &\iint_{\mathcal{D}} \big(\psi^{\mathrm{f}}(\boldsymbol{x}_1) - \psi(\boldsymbol{x}_1)\big) W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\psi^{\mathrm{f}}(\boldsymbol{x}_2) - \psi(\boldsymbol{x}_2)\big) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\ &+ \big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(3)}[\psi_3]\big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi_4]\big), \end{aligned} \tag{3.26}$$

where $W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)$ is defined as a functional inverse of $C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)$ from

$$\int_{\mathcal{D}} C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2) W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_2,\boldsymbol{x}_3) d\boldsymbol{x}_2 = \delta(\boldsymbol{x}_1 - \boldsymbol{x}_3), \tag{3.27}$$

and $\boldsymbol{W}_{\epsilon\epsilon}$ is the inverse of the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$. Here we have used subscripts on the measurement operator and its argument, e.g. $\boldsymbol{\mathcal{M}}_{(3)}[\psi_3]$ indicating that the dummy variable for the integration is $\boldsymbol{x}_3$. This has no implications in this expression but it will be useful in the following derivation.

The variational functional (3.26) measures, in a weighted sense, the distance between an estimate $\psi(\boldsymbol{x})$ and the forecast or first-guess $\psi^{\mathrm{f}}(\boldsymbol{x})$, plus the distance between the estimate and the observations $\boldsymbol{d}$. The field $\psi(\boldsymbol{x})$ which minimizes (3.26) is named $\psi^{\mathrm{a}}(\boldsymbol{x})$. The use of inverses of the error covariances as weights, ensures that the variance minimizing estimate becomes equal to the maximum likelihood estimate in the case with Gaussian error statistics.

3.2.2 Euler–Lagrange equation

To minimize the variational functional, (3.26), we can calculate the variational derivative of $\mathcal{J}[\psi]$ and require that it approaches zero when the arbitrary perturbation $\delta\psi(\boldsymbol{x})$ goes to zero. Thus, we have

$$\delta\mathcal{J} = \mathcal{J}[\psi + \delta\psi] - \mathcal{J}[\psi] = \mathcal{O}(\delta\psi^2). \tag{3.28}$$

Evaluating (3.28) gives

$$\begin{aligned}\delta\mathcal{J} = &-2\iint_{\mathcal{D}} \delta\psi(\boldsymbol{x}_1) W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\psi^{\mathrm{f}}(\boldsymbol{x}_2) - \psi(\boldsymbol{x}_2)\big) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\ &- 2\boldsymbol{\mathcal{M}}_{(3)}[\delta\psi(\boldsymbol{x}_3)]^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi(\boldsymbol{x}_4)]\big) \\ &+ \mathcal{O}(\delta\psi^2) = \mathcal{O}(\delta\psi^2).\end{aligned} \tag{3.29}$$

Thus, to have an extrema of $\mathcal{J}$ we must have

$$\begin{aligned}\iint_{\mathcal{D}} \delta\psi(\boldsymbol{x}_1) W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\psi^{\mathrm{f}}(\boldsymbol{x}_2) - \psi^{\mathrm{a}}(\boldsymbol{x}_2)\big) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\ + \boldsymbol{\mathcal{M}}_{(3)}[\delta\psi(\boldsymbol{x}_3)]^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{a}}(\boldsymbol{x}_4)]\big) = 0.\end{aligned} \tag{3.30}$$

To proceed we need to get the second term in under the integral and both terms need to be proportional to $\delta\psi$. We will now show that

$$\boldsymbol{\mathcal{M}}_{(3)}[\delta\psi(\boldsymbol{x}_3)]^{\mathrm{T}} = \int_{\mathcal{D}} \delta\psi(\boldsymbol{x}_1) \boldsymbol{\mathcal{M}}^{\mathrm{T}}_{(3)}[\delta(\boldsymbol{x}_1 - \boldsymbol{x}_3)] d\boldsymbol{x}_1. \tag{3.31}$$

We start by writing out the measurement of a Dirac delta function, $\delta(\boldsymbol{x}_1 - \boldsymbol{x}_3)$, as

$$\mathcal{M}_{i(3)}[\delta(\boldsymbol{x}_1 - \boldsymbol{x}_3)] = \int_{\mathcal{D}} \delta(\boldsymbol{x}_1 - \boldsymbol{x}_3)\delta(\boldsymbol{x}_3 - \boldsymbol{x}_i) d\boldsymbol{x}_3 = \delta(\boldsymbol{x}_1 - \boldsymbol{x}_i), \tag{3.32}$$

for $i = 1, \ldots, M$ where M is the number of measurements. The subscript (3) on $\mathcal{M}_i$ defines the variable the functional is operating on, thus, the integration variable is $\boldsymbol{x}_3$. Multiplying this equation with $\delta\psi(\boldsymbol{x}_1)$ and integrating in $\boldsymbol{x}_1$ now gives

$$\begin{aligned}\int_{\mathcal{D}} \delta\psi(\boldsymbol{x}_1)\mathcal{M}_{i(3)}[\delta(\boldsymbol{x}_1 - \boldsymbol{x}_3)] d\boldsymbol{x}_1 &= \int_{\mathcal{D}} \delta\psi(\boldsymbol{x}_1)\delta(\boldsymbol{x}_1 - \boldsymbol{x}_i) d\boldsymbol{x}_1 \\ &= \mathcal{M}_{i(1)}[\delta\psi(\boldsymbol{x}_1)] \\ &= \mathcal{M}_{i(3)}[\delta\psi(\boldsymbol{x}_3)].\end{aligned} \tag{3.33}$$

where in the last line, we changed the dummy variable for the integration to $\boldsymbol{x}_3$. Thus, we have obtained (3.31).

We also have that

$$\begin{aligned}\int_{\mathcal{D}} C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)\mathcal{M}^{\mathrm{T}}_{i(3)}[\delta(\boldsymbol{x}_2 - \boldsymbol{x}_3)] d\boldsymbol{x}_2 &= C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_i) \\ &= \mathcal{M}_{i(2)}[C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)].\end{aligned} \tag{3.34}$$

Note that the second term of (3.30), i.e. the measurement term, is constant in the integration with respect to $\boldsymbol{x}_2$. Equations (3.32–3.34) are verified for $i = 1, \ldots, M$, and their results can be generalized and substituted into (3.30) which then leads to

$$\begin{aligned}\iint_{\mathcal{D}} \delta\psi(\boldsymbol{x}_1)\Big(W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\psi^{\mathrm{f}}(\boldsymbol{x}_2) - \psi^{\mathrm{a}}(\boldsymbol{x}_2)\big) \\ + \boldsymbol{\mathcal{M}}^{\mathrm{T}}_{(3)}[\delta(\boldsymbol{x}_1 - \boldsymbol{x}_3)]\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{a}}(\boldsymbol{x}_4)]\big)\Big) d\boldsymbol{x}_1 d\boldsymbol{x}_2 = 0,\end{aligned} \tag{3.35}$$

or since this must be true for all $\delta\psi$ we must have

$$\begin{aligned} W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\psi^{\mathrm{f}}(\boldsymbol{x}_2)-\psi^{\mathrm{a}}(\boldsymbol{x}_2)\big) \\ + \boldsymbol{\mathcal{M}}^{\mathrm{T}}_{(3)}[\delta(\boldsymbol{x}_1-\boldsymbol{x}_3)]\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{a}}(\boldsymbol{x}_4)]\big)=0. \end{aligned} \tag{3.36}$$

This is the Euler–Lagrange equation for the variational problem, of which the solution ψ^{a} must be a minimum of $\mathcal{J}$.

Now multiply (3.36) with $C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x},\boldsymbol{x}_1)$ and integrate with respect to $\boldsymbol{x}_1$. Using the definition (3.27) and the identity (3.34) we get the Euler–Lagrange equation of the form

$$\psi^{\mathrm{a}}(\boldsymbol{x})-\psi^{\mathrm{f}}(\boldsymbol{x})=\boldsymbol{\mathcal{M}}^{\mathrm{T}}_{(3)}[C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x},\boldsymbol{x}_3)]\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{a}}_4]\big). \tag{3.37}$$

3.2.3 Representer solution

A problem with the Euler–Lagrange equation (3.37) is that ψ^{a} is contained on both sides of the equality sign. To resolve this we first define the vector $\boldsymbol{b}\in\Re^M$ as

$$\boldsymbol{b}=\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{a}}_4]\big), \tag{3.38}$$

and then seek a solution of the form

$$\psi^{\mathrm{a}}(\boldsymbol{x})=\psi^{\mathrm{f}}(\boldsymbol{x})+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}(\boldsymbol{x}), \tag{3.39}$$

where we have introduced the vector of representers $\boldsymbol{r}(\boldsymbol{x})\in\Re^M$.

Inserting this into (3.37) gives

$$\psi^{\mathrm{f}}(\boldsymbol{x})-\psi^{\mathrm{f}}(\boldsymbol{x})+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}(\boldsymbol{x})=\boldsymbol{\mathcal{M}}^{\mathrm{T}}_{(3)}[C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x},\boldsymbol{x}_3)]\boldsymbol{b}, \tag{3.40}$$

Thus, we get the influence functions or representers $\boldsymbol{r}(\boldsymbol{x})$ defined as

$$\boldsymbol{r}(\boldsymbol{x})=\boldsymbol{\mathcal{M}}_{(3)}[C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x},\boldsymbol{x}_3)]. \tag{3.41}$$

Now using (3.39) in (3.38) gives

$$\begin{aligned} \boldsymbol{b}&=\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{f}}_4+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}_4]\big) \\ &=\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{f}}_4]\big)-\boldsymbol{W}_{\epsilon\epsilon}\boldsymbol{\mathcal{M}}_{(4)}[\boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}_4] \\ &=\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{f}}_4]\big)-\boldsymbol{W}_{\epsilon\epsilon}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\mathcal{M}}_{(4)}[\boldsymbol{r}_4], \end{aligned} \tag{3.42}$$

because of the linearity of $\boldsymbol{\mathcal{M}}$. Rearranging gives

$$\boldsymbol{b}+\boldsymbol{W}_{\epsilon\epsilon}\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\mathcal{M}}_{(4)}[\boldsymbol{r}_4])=\boldsymbol{W}_{\epsilon\epsilon}\big(\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{f}}_4]\big), \tag{3.43}$$

and, multiplying from the left with $\boldsymbol{C}_{\epsilon\epsilon}$, we obtain

$$\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{b}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\mathcal{M}}_{(4)}[\boldsymbol{r}_4]=\boldsymbol{d}-\boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{f}}_4], \tag{3.44}$$

or

$$\left(\boldsymbol{\mathcal{M}}_{(4)}^{\mathrm{T}}[\boldsymbol{r}_4] + \boldsymbol{C}_{\epsilon\epsilon}\right)\boldsymbol{b} = \boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi_4^{\mathrm{f}}], \tag{3.45}$$

which is a linear system of equations for $\boldsymbol{b}$. Rewriting by using (3.41) the equation becomes

$$\left(\boldsymbol{\mathcal{M}}_{(3)}\boldsymbol{\mathcal{M}}_{(4)}^{\mathrm{T}}[C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_3, \boldsymbol{x}_4)] + \boldsymbol{C}_{\epsilon\epsilon}\right)\boldsymbol{b} = \boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi^{\mathrm{f}}(\boldsymbol{x}_4)]. \tag{3.46}$$

A solution can now be found from the equations (3.39), (3.41) and (3.45).

3.2.4 Representer matrix

Note that with direct measurements as given in (3.24), we have

$$\mathcal{M}_{i(3)}\mathcal{M}_{j(4)}^{\mathrm{T}}[C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_3, \boldsymbol{x}_4)] = C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3.47}$$

The matrix $C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is often called the representer matrix and with direct measurements it describes the covariances of the first-guess between the two locations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

3.2.5 Error estimate

It is possible to derive an error estimate for the analysis (3.39). The simplest is to use the procedure as derived by *Bennett* (1992) for the time dependent problem. From the definition of the error covariance in (3.25) we can write

$$C_{\psi\psi}^{\mathrm{a}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \overline{(\psi^{\mathrm{t}}(\boldsymbol{x}_1) - \psi^{\mathrm{a}}(\boldsymbol{x}_1))(\psi^{\mathrm{t}}(\boldsymbol{x}_2) - \psi^{\mathrm{a}}(\boldsymbol{x}_2))}, \tag{3.48}$$

and insert the equation for the analysis to get

$$\begin{aligned} C_{\psi\psi}^{\mathrm{a}}(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}_1)(\psi_2^{\mathrm{t}} - \psi_2^{\mathrm{f}} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}_2)} \\ &= \overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})(\psi_2^{\mathrm{t}} - \psi_2^{\mathrm{f}})} - 2\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})\boldsymbol{b}^{\mathrm{T}}}\boldsymbol{r}_2 + \boldsymbol{r}_1^{\mathrm{T}}\overline{\boldsymbol{b}\boldsymbol{b}^{\mathrm{T}}}\boldsymbol{r}_2. \end{aligned} \tag{3.49}$$

We have used that $\boldsymbol{b}$ is a function of ψ and the representers $\boldsymbol{r}$, are functions of the covariance matrix and then $\overline{\psi}$. Further, we used the property $(\boldsymbol{AB})^{\mathrm{T}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}$ for matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, and that the covariance is symmetrical in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

The first term is just $C_{\psi\psi}^{\mathrm{f}}$ while the two other terms will be treated next and we now define for convenience

$$\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{M}}_{(3)}\boldsymbol{\mathcal{M}}_{(4)}^{\mathrm{T}}[C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_3, \boldsymbol{x}_4)] + \boldsymbol{C}_{\epsilon\epsilon}, \tag{3.50}$$

and the residual or innovation

$$\boldsymbol{h} = \boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi_4^{\mathrm{f}}]. \tag{3.51}$$

Using (3.41), (3.50) and (3.51) in (3.45) gives $\boldsymbol{b} = \boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h}$. Furthermore, by using (3.23), (3.25), (3.41) and (3.45), in addition to the two definitions above, the second term in (3.49) becomes

$$\begin{aligned}
&-2\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})\boldsymbol{b}^{\mathrm{T}}}\boldsymbol{r}_2 \\
&\quad = -2\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})(\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h})^{\mathrm{T}}}\boldsymbol{r}_2 \\
&\quad = -2\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})\left(\boldsymbol{\mathcal{P}}^{-1}(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}[\psi_4^{\mathrm{f}}])\right)^{\mathrm{T}}}\boldsymbol{r}_2 \\
&\quad = -2\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})\left(\boldsymbol{\mathcal{P}}^{-1}(\boldsymbol{\mathcal{M}}_{(4)}[\psi_4^{\mathrm{t}}] + \boldsymbol{\epsilon} - \boldsymbol{\mathcal{M}}_{(4)}[\psi_4^{\mathrm{f}}])\right)^{\mathrm{T}}}\boldsymbol{r}_2 \\
&\quad = -2\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})\boldsymbol{\mathcal{M}}_{(4)}^{\mathrm{T}}[\psi_4^{\mathrm{t}} - \psi_4^{\mathrm{f}}]}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 + 0 \\
&\quad = -2\boldsymbol{\mathcal{M}}_{(4)}^{\mathrm{T}}[\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})(\psi_4^{\mathrm{t}} - \psi_4^{\mathrm{f}})}]\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = -2\boldsymbol{\mathcal{M}}_{(4)}^{\mathrm{T}}[C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_1, \boldsymbol{x}_4)]\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = -2\boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2.
\end{aligned} \tag{3.52}$$

Here we have also used that $\overline{\boldsymbol{\epsilon}} = 0$ from (3.25), and that $\boldsymbol{\mathcal{P}}$ is a symmetrical function of the covariance and can be moved outside the averaging.

Further, using $(\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h})^{\mathrm{T}} = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}$, the last term becomes

$$\begin{aligned}
&\boldsymbol{r}_1^{\mathrm{T}}\overline{\boldsymbol{b}\boldsymbol{b}^{\mathrm{T}}}\boldsymbol{r}_2 \\
&\quad = \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}\overline{\boldsymbol{h}\boldsymbol{h}^{\mathrm{T}}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}\overline{(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(1)}[\psi_1^{\mathrm{f}}])(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(2)}[\psi_2^{\mathrm{f}}])^{\mathrm{T}}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}\overline{(\boldsymbol{\mathcal{M}}_{(1)}[\psi_1^{\mathrm{t}}] + \boldsymbol{\epsilon} - \boldsymbol{\mathcal{M}}_{(1)}[\psi_1^{\mathrm{f}}])(\boldsymbol{\mathcal{M}}_{(2)}[\psi_2^{\mathrm{t}}] + \boldsymbol{\epsilon} - \boldsymbol{\mathcal{M}}_{(1)}[\psi_2^{\mathrm{f}}])^{\mathrm{T}}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}\overline{(\boldsymbol{\mathcal{M}}_{(1)}[\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}}] + \boldsymbol{\epsilon})(\boldsymbol{\mathcal{M}}_{(2)}[\psi_2^{\mathrm{t}} - \psi_2^{\mathrm{f}}] + \boldsymbol{\epsilon})^{\mathrm{T}}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}(\boldsymbol{\mathcal{M}}_{(1)}\boldsymbol{\mathcal{M}}_{(2)}^{\mathrm{T}}[\overline{(\psi_1^{\mathrm{t}} - \psi_1^{\mathrm{f}})(\psi_2^{\mathrm{t}} - \psi_2^{\mathrm{f}})}] + \overline{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}})\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2 \\
&\quad = \boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{r}_2.
\end{aligned} \tag{3.53}$$

Thus, an error estimate is given as

$$\begin{aligned}
C_{\psi\psi}^{\mathrm{a}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_1, \boldsymbol{x}_2) \\
- \boldsymbol{r}^{\mathrm{T}}(\boldsymbol{x}_1)\left(\boldsymbol{\mathcal{M}}_{(3)}\boldsymbol{\mathcal{M}}_{(4)}^{\mathrm{T}}[C_{\psi\psi}^{\mathrm{f}}(\boldsymbol{x}_3, \boldsymbol{x}_4)] + \boldsymbol{C}_{\epsilon\epsilon}\right)^{-1}\boldsymbol{r}(\boldsymbol{x}_2).
\end{aligned} \tag{3.54}$$

where the definition for $\boldsymbol{\mathcal{P}}$ has been used.

3.2.6 Uniqueness of the solution

By expressing the solution as in (3.39) not all arbitrary functions can be represented. To show that the solution (3.39) is the unique variance minimizing

linear solution we proceed with the following argumentation using a geometrical formulation, identical to the formulation used for the time dependent problem by *Bennett* (1992). First define the inner product

$$< f(\boldsymbol{x}_1), g(\boldsymbol{x}_2) >= \iint_{\mathcal{D}} f(\boldsymbol{x}_1) W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2) g(\boldsymbol{x}_2) d\boldsymbol{x}_1 d\boldsymbol{x}_2. \tag{3.55}$$

Note that

$$\begin{aligned} &< C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_3, \boldsymbol{x}_1), \psi(\boldsymbol{x}_2) > \\ &\quad = \iint_{\mathcal{D}} C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_3, \boldsymbol{x}_1) W^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2) \psi(\boldsymbol{x}_2) d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\ &\quad = \psi(\boldsymbol{x}_3), \end{aligned} \tag{3.56}$$

thus, $C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_3, \boldsymbol{x}_1)$ is a "reproducing kernel" for the inner product (3.55) and the expression (3.56) is true for every field ψ in any point $\boldsymbol{x}$.

Recalling the definition of the representer (3.41) we get

$$\begin{aligned} < \boldsymbol{r}(\boldsymbol{x}_1), \psi(\boldsymbol{x}_2) > &=< \boldsymbol{\mathcal{M}}_{(1)}[C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_3, \boldsymbol{x}_1)], \psi(\boldsymbol{x}_2) > \\ &= \boldsymbol{\mathcal{M}}_{(1)}[< C^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_3, \boldsymbol{x}_1), \psi(\boldsymbol{x}_2) >] \\ &= \boldsymbol{\mathcal{M}}_{(1)}[\psi(\boldsymbol{x}_1)] \end{aligned} \tag{3.57}$$

Thus, the measurement of a field $\psi(\boldsymbol{x})$ is equivalent to projecting the field onto the representer using the inner product (3.55).

The penalty function (3.26) can now be written entirely in terms of inner products as

$$\mathcal{J}[\psi] =< \psi^{\mathrm{f}} - \psi, \psi^{\mathrm{f}} - \psi > + (\boldsymbol{d} - < \psi, \boldsymbol{r} >)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{d} - < \psi, \boldsymbol{r} >). \tag{3.58}$$

Assume now that the minimizing solution is expressed as

$$\psi^{\mathrm{a}}(\boldsymbol{x}) = \psi^{\mathrm{f}}(\boldsymbol{x}) + \boldsymbol{b}^{\mathrm{T}} \boldsymbol{r}(\boldsymbol{x}) + g(\boldsymbol{x}), \tag{3.59}$$

where $g(\boldsymbol{x})$ is an arbitrary function orthogonal to the representers, i.e.

$$< g, \boldsymbol{r} >= \boldsymbol{0}. \tag{3.60}$$

Because of this identity the field g may be regarded as unobservable. Substituting (3.59) into (3.58) gives

$$\begin{aligned} \mathcal{J}[\psi^{\mathrm{a}}] =& < \boldsymbol{r}^{\mathrm{T}} \boldsymbol{b} + g, \boldsymbol{r}^{\mathrm{T}} \boldsymbol{b} + g > \\ &+ (\boldsymbol{d} - < \psi^{\mathrm{a}}, \boldsymbol{r} >)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{d} - < \psi^{\mathrm{a}}, \boldsymbol{r} >) \\ &= \boldsymbol{b}^{\mathrm{T}} < \boldsymbol{r}, \boldsymbol{r}^{\mathrm{T}} > \boldsymbol{b} + \boldsymbol{b}^{\mathrm{T}} < \boldsymbol{r}, g > + < g, \boldsymbol{r}^{\mathrm{T}} > \boldsymbol{b} + < g, g > \\ &+ (\boldsymbol{d} - < \psi^{\mathrm{f}}, \boldsymbol{r} > - \boldsymbol{b}^{\mathrm{T}} < \boldsymbol{r}, \boldsymbol{r}^{\mathrm{T}} > - < g, \boldsymbol{r} >)^{\mathrm{T}} \\ &\quad \times \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{d} - < \psi^{\mathrm{f}}, \boldsymbol{r} > - \boldsymbol{b}^{\mathrm{T}} < \boldsymbol{r}, \boldsymbol{r}^{\mathrm{T}} > - < g, \boldsymbol{r} >). \end{aligned} \tag{3.61}$$

Defining the residual

$$\boldsymbol{h} = \boldsymbol{d} - < \psi^{\mathrm{f}}, \boldsymbol{r} >, \tag{3.62}$$

and using the definition of the representer matrix,

$$\boldsymbol{R} = < \boldsymbol{r}_3, \boldsymbol{r}_4^{\mathrm{T}} > = \boldsymbol{\mathcal{M}}_{(3)}[\boldsymbol{r}_3^{\mathrm{T}}], \tag{3.63}$$

and (3.41) and (3.47), we get the penalty function of the form

$$\mathcal{J}[\psi^{\mathrm{a}}] = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{b} + < g, g > + (\boldsymbol{h} - \boldsymbol{R}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{W}_{\epsilon\epsilon}(\boldsymbol{h} - \boldsymbol{R}\boldsymbol{b}). \tag{3.64}$$

The original penalty function (3.26) has now been reduced to a compact form where the disposable parameters are $\boldsymbol{b}$ and $g(\boldsymbol{x})$. If $\boldsymbol{\psi}$ minimizes $\mathcal{J}$ then clearly $< g, g >= 0$ and thus

$$g(\boldsymbol{x}) \equiv 0. \tag{3.65}$$

The unobservable field g must be discarded, reducing $\mathcal{J}$ from the infinite dimensional quadratic form (3.26) to the finite dimensional quadratic form

$$\mathcal{B}[\boldsymbol{b}] = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{b} + (\boldsymbol{h} - \boldsymbol{R}\boldsymbol{b})^{\mathrm{T}}\boldsymbol{W}_{\epsilon\epsilon}(\boldsymbol{h} - \boldsymbol{R}\boldsymbol{b}), \tag{3.66}$$

where $\mathcal{B}[\boldsymbol{b}] = \mathcal{J}[\psi^{\mathrm{a}}]$.

3.2.7 Minimization of the penalty function

The minimizing solution for $\boldsymbol{b}$ can again be found by setting the variational derivative of (3.66) with respect to $\boldsymbol{b}$ equal to zero,

$$\mathcal{B}[\boldsymbol{b}+\delta\boldsymbol{b}] - \mathcal{B}[\boldsymbol{b}] = 2\delta\boldsymbol{b}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{b} + 2\delta\boldsymbol{b}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{W}_{\epsilon\epsilon}(\boldsymbol{R}\boldsymbol{b}-\boldsymbol{h}) + \mathcal{O}(\delta\boldsymbol{b}^2) = \mathcal{O}(\delta\boldsymbol{b}^2), \tag{3.67}$$

which gives

$$\delta\boldsymbol{b}^{\mathrm{T}}\big(\boldsymbol{R}\boldsymbol{b} + \boldsymbol{R}\boldsymbol{W}_{\epsilon\epsilon}(\boldsymbol{R}\boldsymbol{b} - \boldsymbol{h})\big) = 0, \tag{3.68}$$

or

$$\boldsymbol{R}\boldsymbol{b} + \boldsymbol{R}\boldsymbol{W}_{\epsilon\epsilon}(\boldsymbol{R}\boldsymbol{b} - \boldsymbol{h}) = 0, \tag{3.69}$$

since $\delta\boldsymbol{b}$ is arbitrary. This equation can be written as

$$\boldsymbol{R}(\boldsymbol{b} + \boldsymbol{W}_{\epsilon\epsilon}\boldsymbol{R}\boldsymbol{b} - \boldsymbol{W}_{\epsilon\epsilon}\boldsymbol{h}) = 0, \tag{3.70}$$

which leads to the standard linear system of equations

$$(\boldsymbol{R} + \boldsymbol{C}_{\epsilon\epsilon})\boldsymbol{b} = \boldsymbol{h}, \tag{3.71}$$

or

$$\boldsymbol{b} = \boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h}, \tag{3.72}$$

as the solution for $\boldsymbol{b}$. Note that we have used that $\boldsymbol{R} = \mathcal{M}_{(i)}[\boldsymbol{r}_i]$ for all i.

3.2.8 Prior and posterior value of the penalty function

Inserting the first-guess value ψ^{f}, into the penalty function (3.58) gives

$$\mathcal{J}[\psi^{\mathrm{f}}] = (\boldsymbol{d} - <\psi^{\mathrm{f}}, \boldsymbol{r}>)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{d} - <\psi^{\mathrm{f}}, \boldsymbol{r}>) = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{h}. \tag{3.73}$$

This is known as the prior value of the penalty function.

Similarly by inserting the minimizing solution (3.72) into the penalty function (3.66) we get the following,

$$\begin{aligned}
\mathcal{J}[\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h}] &= (\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h})^{\mathrm{T}} \boldsymbol{R} (\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h}) + (\boldsymbol{h} - \boldsymbol{R}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h})^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{h} - \boldsymbol{R}\boldsymbol{\mathcal{P}}^{-1}\boldsymbol{h}) \\
&= \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{R} \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{h} + \boldsymbol{h}^{\mathrm{T}} (\boldsymbol{R}\boldsymbol{\mathcal{P}}^{-1} - \boldsymbol{I}) \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{R}\boldsymbol{\mathcal{P}}^{-1} - \boldsymbol{I}) \boldsymbol{h} \\
&= \boldsymbol{h}^{\mathrm{T}} \left\{ \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{R} \boldsymbol{\mathcal{P}}^{-1} + (\boldsymbol{R}\boldsymbol{\mathcal{P}}^{-1} - \boldsymbol{I}) \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{R}\boldsymbol{\mathcal{P}}^{-1} - \boldsymbol{I}) \right\} \boldsymbol{h} \\
&= \boldsymbol{h}^{\mathrm{T}} \left\{ \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{R} \boldsymbol{\mathcal{P}}^{-1} + \boldsymbol{\mathcal{P}}^{-1} (\boldsymbol{R} - \boldsymbol{\mathcal{P}}) \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{R} - \boldsymbol{\mathcal{P}}) \boldsymbol{\mathcal{P}}^{-1} \right\} \boldsymbol{h} \\
&= \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\mathcal{P}}^{-1} \left\{ \boldsymbol{R} + (\boldsymbol{R} - \boldsymbol{\mathcal{P}}) \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{R} - \boldsymbol{\mathcal{P}}) \right\} \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{h} \\
&= \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\mathcal{P}}^{-1} \left\{ \boldsymbol{R} + \boldsymbol{C}_{\epsilon\epsilon} \right\} \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{h} \\
&= \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{\mathcal{P}} \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{h} \\
&= \boldsymbol{h}^{\mathrm{T}} \boldsymbol{\mathcal{P}}^{-1} \boldsymbol{h} \\
&= \boldsymbol{h}^{\mathrm{T}} \boldsymbol{b},
\end{aligned} \tag{3.74}$$

as long as $\boldsymbol{b}$ is given from (3.72). This is known as the posterior value of the penalty function.

It is explained by *Bennett* (2002, section 2.3) that the reduced penalty function is a $\mathcal{X}^2_M$ variable. Thus, we have a mean to test the validity of our statistical assumptions, by checking if the value of reduced penalty function is a Gaussian variable with mean equal to M and variance equal to $2M$. This could be done rigorously by repeated minimizations of the penalty function using different data sets.

3.3 Discrete form

When discretized on a numerical grid, (3.22–3.23) are written as

$$\boldsymbol{\psi}^{\mathrm{f}} = \boldsymbol{\psi}^{\mathrm{t}} + \boldsymbol{p}^{\mathrm{f}}, \tag{3.75}$$

$$\boldsymbol{d} = \boldsymbol{M}\boldsymbol{\psi}^{\mathrm{t}} + \boldsymbol{\epsilon}, \tag{3.76}$$

where $\boldsymbol{M}$, now called the measurement matrix, is the discrete representation of $\boldsymbol{\mathcal{M}}$.

The statistical null hypothesis $\mathcal{H}_0$ is then

$$\begin{aligned}
\overline{\boldsymbol{p}^{\mathrm{f}}} &= \boldsymbol{0}, & \overline{\boldsymbol{p}^{\mathrm{f}} (\boldsymbol{p}^{\mathrm{f}})^{\mathrm{T}}} &= \boldsymbol{C}^{\mathrm{f}}_{\psi\psi}, \\
\overline{\boldsymbol{\epsilon}} &= \boldsymbol{0}, & \overline{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}} &= \boldsymbol{C}_{\epsilon\epsilon}, \\
\overline{\boldsymbol{p}^{\mathrm{f}} \boldsymbol{\epsilon}^{\mathrm{T}}} &= \boldsymbol{0}.
\end{aligned} \tag{3.77}$$

By using the same statistical procedure as in Sect. 3.1, or alternatively by minimizing the variational functional

$$\mathcal{J}[\boldsymbol{\psi}^{\mathrm{a}}] = (\boldsymbol{\psi}^{\mathrm{f}} - \boldsymbol{\psi}^{\mathrm{a}})^{\mathrm{T}} (\boldsymbol{C}^{\mathrm{f}}_{\psi\psi})^{-1} (\boldsymbol{\psi}^{\mathrm{f}} - \boldsymbol{\psi}^{\mathrm{a}}) + (\boldsymbol{d} - \boldsymbol{M}\boldsymbol{\psi}^{\mathrm{a}})^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{d} - \boldsymbol{M}\boldsymbol{\psi}^{\mathrm{a}}), \quad (3.78)$$

with respect to $\boldsymbol{\psi}^{\mathrm{a}}$, one get,

$$\boldsymbol{\psi}^{\mathrm{a}} = \boldsymbol{\psi}^{\mathrm{f}} + \boldsymbol{r}^{\mathrm{T}} \boldsymbol{b}, \quad (3.79)$$

where the influence functions (e.g. error covariance functions for direct measurements) are given as

$$\boldsymbol{r} = \boldsymbol{M}\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}, \quad (3.80)$$

i.e. "measurements" of the error covariance matrix $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$. Thus, $\boldsymbol{r}$ is a matrix where each row contains a representer for a particular measurement. The coefficients $\boldsymbol{b}$ are determined from the system of linear equations

$$(\boldsymbol{M}\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}_{\epsilon\epsilon})\boldsymbol{b} = \boldsymbol{d} - \boldsymbol{M}\boldsymbol{\psi}^{\mathrm{f}}. \quad (3.81)$$

In addition the error estimate (3.54) becomes

$$\boldsymbol{C}^{\mathrm{a}}_{\psi\psi} = \boldsymbol{C}^{\mathrm{f}}_{\psi\psi} - \boldsymbol{r}^{\mathrm{T}} \left(\boldsymbol{M}\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}_{\epsilon\epsilon} \right)^{-1} \boldsymbol{r}. \quad (3.82)$$

Thus, the inverse estimate $\boldsymbol{\psi}^{\mathrm{a}}$, is given by the first-guess $\boldsymbol{\psi}^{\mathrm{f}}$, plus a linear combination of influence functions $\boldsymbol{r}^{\mathrm{T}}\boldsymbol{b}$, one for each of the measurements. The coefficients $\boldsymbol{b}$ are clearly small if the first-guess is close to the data, and large if the residual between the data and the first-guess is large.

Note that a more common way of writing the previous equations is the following:

$$\boldsymbol{\psi}^{\mathrm{a}} = \boldsymbol{\psi}^{\mathrm{f}} + \boldsymbol{K}(\boldsymbol{d} - \boldsymbol{M}\boldsymbol{\psi}^{\mathrm{f}}), \quad (3.83)$$

$$\boldsymbol{C}^{\mathrm{a}}_{\psi\psi} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{M})\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}, \quad (3.84)$$

$$\boldsymbol{K} = \boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\boldsymbol{M}^{\mathrm{T}}(\boldsymbol{M}\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}_{\epsilon\epsilon})^{-1}, \quad (3.85)$$

where the matrix $\boldsymbol{K}$ is often called the Kalman gain. This can be derived directly from (3.79)–(3.82) by rearranging terms, and it is the standard way of writing the analysis equations for the Kalman filter to be discussed in Chap. 4. The numerical evaluation of these equations, however, is simpler and more efficient using the form (3.79)–(3.82)

4

Sequential data assimilation

In the previous chapter we considered a time independent problem and computed the best conditional estimate given a prior estimate and measurements of the state.

For time dependent problems, sequential data assimilation methods use the analysis scheme from the previous chapter to sequentially update the model state. Such methods have proven useful for many applications in meteorology and oceanography, including operational weather prediction systems where new observations are sequentially assimilated into the model when they become available.

If a model forecast $\boldsymbol{\psi}^{\mathrm{f}}$, and the forecast error covariance $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$, are known at a time t_k, where we have available measurements $\boldsymbol{d}$, with a measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$, it is possible to calculate an improved analysis $\boldsymbol{\psi}^{\mathrm{a}}$, with its analyzed error covariance $\boldsymbol{C}^{\mathrm{a}}_{\psi\psi}$. A major issue is then how to estimate or predict the error covariance $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$ for the model forecast at the time t_k.

This chapter will briefly outline the Kalman Filter (KF) originally proposed by *Kalman* (1960), which introduces an equation for the time evolution of the error covariance matrix. Further, the problems associated with the use of the KF with nonlinear dynamical models will be illustrated. Finally a basic introduction is given to the Ensemble Kalman Filter (EnKF) proposed by *Evensen* (1994a).

4.1 Linear Dynamics

For linear dynamics the optimal sequential assimilation method is the Kalman filter. In the Kalman filter an additional equation for the second-order statistical moment is integrated forward in time to predict error statistics for the model forecast. The error statistics are then used to calculate a variance minimizing estimate whenever measurements are available.

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_4,

4.1.1 Kalman filter for a scalar case

Assume now that a discrete dynamical model for the true state of a scalar ψ can be written as

$$\psi^{\mathrm{t}}(t_k) = G\psi^{\mathrm{t}}(t_{k-1}) + q(t_{k-1}), \tag{4.1}$$

$$\psi^{\mathrm{t}}(t_0) = \Psi_0 + a, \tag{4.2}$$

where G is a linear model operator, q is the model error over one time step and Ψ_0 is an initial condition with error a.

The model error is normally not known and a numerical model will therefore evolve according to

$$\psi^{\mathrm{f}}(t_k) = G\psi^{\mathrm{a}}(t_{k-1}) \tag{4.3}$$

$$\psi^{\mathrm{a}}(t_0) = \Psi_0. \tag{4.4}$$

That is, given a best estimate ψ^{a}, for ψ at time t_{k-1}, a forecast ψ^{f}, is calculated at time t_k, using the approximate equation (4.3).

Now subtract (4.3) from (4.1) to get

$$\psi^{\mathrm{t}}_k - \psi^{\mathrm{f}}_k = G(\psi^{\mathrm{t}}_{k-1} - \psi^{\mathrm{a}}_{k-1}) + q_{k-1}. \tag{4.5}$$

where we have defined $\psi_k = \psi(t_k)$ and $q_k = q(t_k)$. The error covariance matrix for the forecast at time t_k is

$$\begin{aligned} C^{\mathrm{f}}_{\psi\psi}(t_k) &= \overline{(\psi^{\mathrm{t}}_k - \psi^{\mathrm{f}}_k)^2} \\ &= G^2\overline{(\psi^{\mathrm{t}}_{k-1} - \psi^{\mathrm{a}}_{k-1})^2} + \overline{q^2_{k-1}} + 2G\overline{(\psi^{\mathrm{t}}_{k-1} - \psi^{\mathrm{a}}_{k-1})q_{k-1}} \\ &= G^2 C^{\mathrm{a}}_{\psi\psi}(t_{k-1}) + C_{qq}(t_{k-1}). \end{aligned} \tag{4.6}$$

We have defined the error covariance for the model state

$$C^{\mathrm{a}}_{\psi\psi}(t_{k-1}) = \overline{(\psi^{\mathrm{t}}_{k-1} - \psi^{\mathrm{a}}_{k-1})^2}, \tag{4.7}$$

the model error covariance

$$C_{qq}(t_{k-1}) = \overline{q^2_{k-1}}, \tag{4.8}$$

and the initial error covariance

$$C_{\psi\psi}(t_0) = C_{aa} = \overline{a^2}. \tag{4.9}$$

It is also assumed that there are no correlations between the error in the state, $\psi^{\mathrm{t}}_{k-1} - \psi^{\mathrm{a}}_{k-1}$, the model error q_{k-1}, and the initial error a.

Thus, we have a consistent set of dynamical equations for the model evolution (4.3 and 4.4), and the error (co)variance evolution (4.6), (4.8) and (4.9). At times when there are measurements available, an analyzed estimate can be calculated using the equations (3.14) and (3.15), and when there are no measurements available we just set $\psi^{\mathrm{a}} = \psi^{\mathrm{f}}$ and $C^{\mathrm{a}}_{\psi\psi} = C^{\mathrm{f}}_{\psi\psi}$, and continue the integration. These equations define the Kalman filter for a linear scalar model, and thus constitute the optimal sequential data assimilation method for this model given that the priors are all Gaussian and unbiased.

4.1.2 Kalman filter for a vector state

If the true state $\psi^{\mathrm{t}}(\boldsymbol{x})$ is discretized on a numerical grid, it can be represented by the state vector $\boldsymbol{\psi}^{\mathrm{t}}$. It is assumed that the true state evolves according to a dynamical model

$$\boldsymbol{\psi}_k^{\mathrm{t}} = \boldsymbol{G}\boldsymbol{\psi}_{k-1}^{\mathrm{t}} + \boldsymbol{q}_{k-1}, \tag{4.10}$$

where $\boldsymbol{G}$ is a linear model operator (matrix) and $\boldsymbol{q}$ is the unknown model error over one time step. In this case a numerical model will evolve according to

$$\boldsymbol{\psi}_k^{\mathrm{f}} = \boldsymbol{G}\boldsymbol{\psi}_{k-1}^{\mathrm{a}}. \tag{4.11}$$

That is, given the best possible estimate for $\boldsymbol{\psi}$ at time t_{k-1}, a forecast is calculated at time t_k, using the approximate equation (4.11).

The error covariance equation is derived using a similar procedure as was used for (4.6), and becomes

$$\boldsymbol{C}_{\psi\psi}^{\mathrm{f}}(t_k) = \boldsymbol{G}\boldsymbol{C}_{\psi\psi}^{\mathrm{a}}(t_{k-1})\boldsymbol{G}^{\mathrm{T}} + \boldsymbol{C}_{qq}(t_{k-1}). \tag{4.12}$$

Thus, the standard Kalman filter consists of the dynamical equations (4.11) and (4.12) together with the analysis equations (3.83–3.85) or alternatively (3.79–3.82).

4.1.3 Kalman filter with a linear advection equation

Here we illustrate the properties of the KF when used with a one-dimensional linear advection model on a periodic domain of length 1000 m. The model has a constant advection speed, $u = 1$ m/s, the grid spacing is $\Delta x = 1$ m and the time step is $\Delta t = 1$ s.

Given an initial condition, the solution of this model is exactly known, and this allows us to run realistic experiments with zero model errors to examine the impact of the dynamical evolution of the error covariance.

The true initial state is sampled from a distribution $\mathcal{N}$, with mean equal to zero, variance equal to one, and a spatial de-correlation length of 20 m.

The first guess solution is generated by drawing another sample from $\mathcal{N}$ and adding this to the true state. The initial ensemble is then generated by adding samples drawn from $\mathcal{N}$ to the first guess solution. Thus, the initial state is assumed to have an error variance equal to one.

Four measurements of the true solution, distributed regularly in the model domain, are assimilated every 5^{th} time step. The measurements are contaminated by errors of variance equal to 0.01, and we have assumed uncorrelated measurement errors.

The length of the integration is 300 s, which is 50 s longer than the time needed for the solution to advect from one measurement to the next (i.e. 250 s).

In Fig. 4.1 an example is shown where the model errors have been set to zero. The plots illustrate the convergence of the estimated solution at various

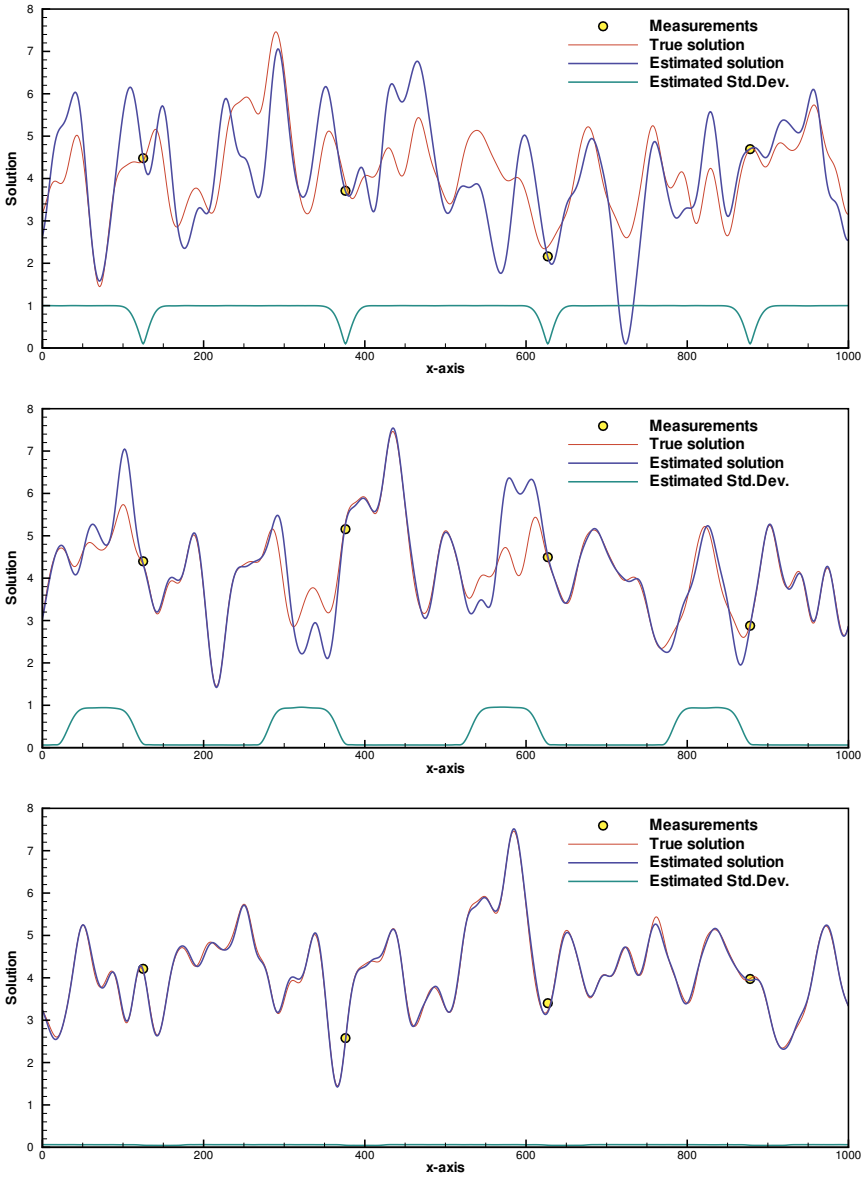

Fig. 4.1. Kalman filter experiment: reference solution, measurements, estimate and standard deviation at three different times $t = 5$ *(top)*, $t = 150$ *(middle)*, and $t = 300$ *(bottom)*

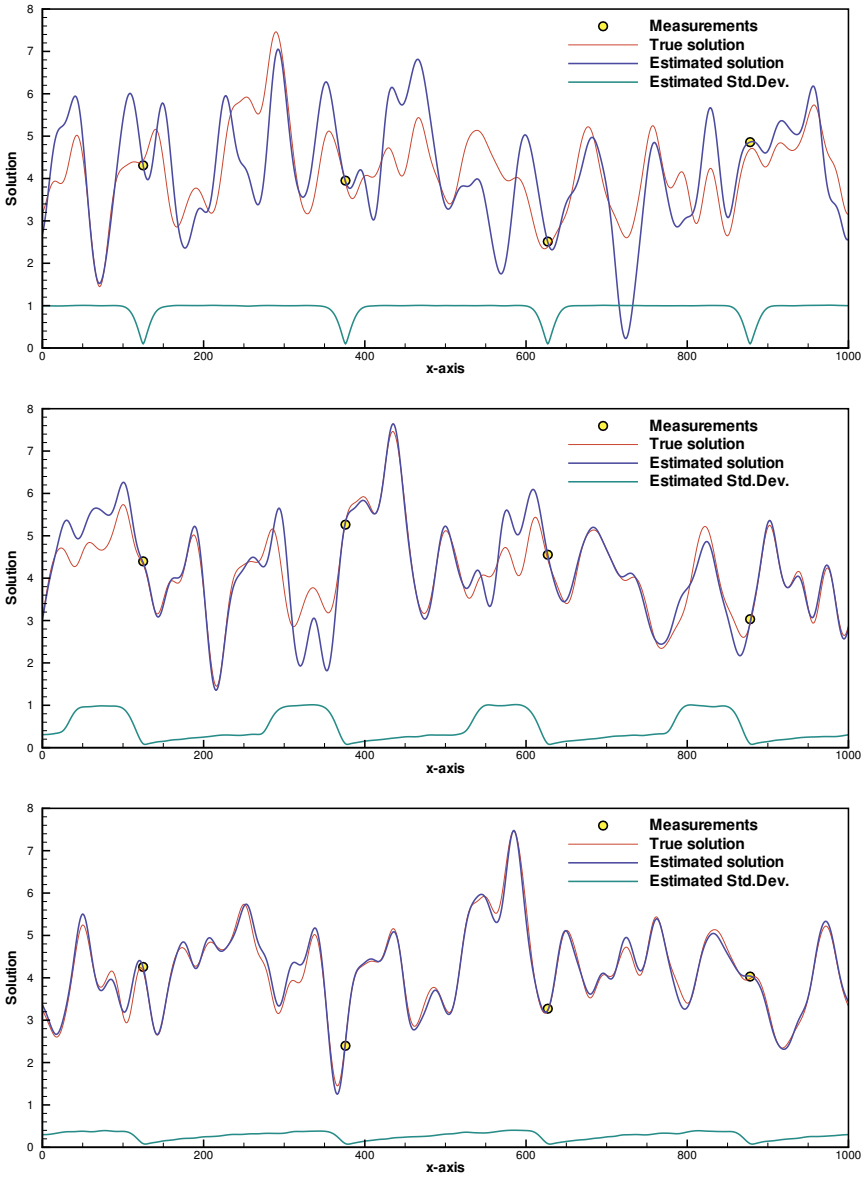

Fig. 4.2. Kalman filter experiment when system noise is included: reference solution, measurements, estimate and standard deviation at three different times $t = 5$ *(top)*, $t = 150$ *(middle)*, and $t = 300$ *(bottom)*

times during the experiment, and show how information from measurements is propagated with the advection speed and how the error variance is reduced every time measurements are assimilated.

The first plot shows the result of the first update with the four measurements. Near the measurement locations, the estimated solution is clearly consistent with the true solution and the measurements, and the error variance is reduced accordingly. The second plot is taken at $t = 150$ s, i.e. after 30 updates with measurements. Now the information from the measurements has propagated to the right with the advection speed. This is seen both from direct comparison of the estimate with the true solution, and from the estimated variance. The final plot is taken at $t = 300$ s and the estimate is now in good agreement with the true solution throughout the model domain. Note also a further reduction of the error variance to the right of the measurements. This is caused by the further introduction of information from the measurements to the already accurate estimate. In this case the estimated errors will converge towards zero since we experience a further accumulation of information and error reduction every 250 s of integration.

The impact of model errors is illustrated in Fig. 4.2. Here we note a linear increase in error variance to the right of the measurements. This is caused by the addition of model errors every time step. It is also clear that the estimated solution deteriorates far from the measurement in the advection direction. The converged error variance is larger than in the previous case. It turns out that for linear models with regular measurements at fixed locations and stationary error statistics, the error variance converges to an estimate where the increase of error variance from model errors balances the reduction from the updates with measurements.

In fact these examples were actually run using the Ensemble Kalman Filter discussed below, but for a linear model the EnKF will converge exactly to the KF with increasing ensemble size.

4.2 Nonlinear dynamics

For nonlinear dynamics the extended Kalman filter (EKF) may be applied, in which an approximate linearized equation is used for the prediction of error statistics.

4.2.1 Extended Kalman filter for the scalar case

Assume now that we have a nonlinear scalar model

$$\psi_k^{\mathrm{t}} = G(\psi_{k-1}^{\mathrm{t}}) + q_{k-1}, \tag{4.13}$$

where $G(\psi)$ is a nonlinear model operator and q is again the unknown model error over one time step. A numerical model will evolve according to the approximate equation

$$\psi_k^{\mathrm{f}} = G(\psi_{k-1}^{\mathrm{a}}). \tag{4.14}$$

Subtracting (4.14) from (4.13) gives

$$\psi_k^{\mathrm{t}} - \psi_k^{\mathrm{f}} = G(\psi_{k-1}^{\mathrm{t}}) - G(\psi_{k-1}^{\mathrm{a}}) + q_{k-1}. \tag{4.15}$$

Now use the Taylor expansion

$$\begin{aligned} G(\psi_{k-1}^{\mathrm{t}}) = G(\psi_{k-1}^{\mathrm{a}}) &+ G'(\psi_{k-1}^{\mathrm{a}})(\psi_{k-1}^{\mathrm{t}} - \psi_{k-1}^{\mathrm{a}}) \\ &+ \frac{1}{2}G''(\psi_{k-1}^{\mathrm{a}})(\psi_{k-1}^{\mathrm{t}} - \psi_{k-1}^{\mathrm{a}})^2 + \cdots, \end{aligned} \tag{4.16}$$

in (4.15) to get

$$\begin{aligned} \psi_k^{\mathrm{t}} - \psi_k^{\mathrm{f}} = &G'(\psi_{k-1}^{\mathrm{a}})(\psi_{k-1}^{\mathrm{t}} - \psi_{k-1}^{\mathrm{a}}) \\ &+ \frac{1}{2}G''(\psi_{k-1}^{\mathrm{a}})(\psi_{k-1}^{\mathrm{t}} - \psi_{k-1}^{\mathrm{a}})^2 + \cdots + q_{k-1}. \end{aligned} \tag{4.17}$$

By squaring and taking the expected value an equation for the evolution of the error variance $C_{\psi\psi}^{\mathrm{f}}(t_k)$ becomes

$$\begin{aligned} C_{\psi\psi}^{\mathrm{f}}(t_k) &= \overline{(\psi_k^{\mathrm{t}} - \psi_k^{\mathrm{f}})^2} \\ &= \overline{(\psi_{k-1}^{\mathrm{t}} - \psi_{k-1}^{\mathrm{a}})^2}(G'(\psi_{k-1}^{\mathrm{a}}))^2 \\ &+ \overline{(\psi_{k-1}^{\mathrm{t}} - \psi_{k-1}^{\mathrm{a}})^3}G'(\psi_{k-1}^{\mathrm{a}})G''(\psi_{k-1}^{\mathrm{a}}) \\ &+ \frac{1}{4}\overline{(\psi_{k-1}^{\mathrm{t}} - \psi_{k-1}^{\mathrm{a}})^4}(G''(\psi_{k-1}^{\mathrm{a}}))^2 + \cdots + C_{qq}(t_{k-1}). \end{aligned} \tag{4.18}$$

This equation can be closed by discarding moments of third and higher order, which results in an approximate equation for the error variance,

$$C_{\psi\psi}^{\mathrm{f}}(t_k) \simeq C_{\psi\psi}^{\mathrm{a}}(t_{k-1})(G'(\psi_{k-1}^{a}))^2 + C_{qq}(t_{k-1}). \tag{4.19}$$

Together with the equations for the analyzed estimate and error variance, (3.14) and (3.15), the dynamical equations (4.14) and (4.19) constitute the extended Kalman filter (EKF) in the case with a scalar state variable.

It is clear that we now have an approximate equation for the error covariance evolution, due to the linearization and closure assumption used. Thus, the usefulness of the EKF will depend on the properties of the model dynamics.

The EKF can also be formulated for measurements which are related to the state variables by a nonlinear operator (see *Gelb*, 1974).

4.2.2 Extended Kalman filter in matrix form

The derivation of the EKF in matrix form is based on the same principles as for the scalar case and can be found in a number of books on control theory

(see e.g. *Jazwinski*, 1970, *Gelb*, 1974). Again we assume a nonlinear model, but now the true state vector at time t_k is calculated from

$$\boldsymbol{\psi}_k^{\mathrm{t}} = \boldsymbol{G}(\boldsymbol{\psi}_{k-1}^{\mathrm{t}}) + \boldsymbol{q}_{k-1}, \tag{4.20}$$

and a forecast is calculated from the approximate equation

$$\boldsymbol{\psi}_k^{\mathrm{f}} = \boldsymbol{G}(\boldsymbol{\psi}_{k-1}^{\mathrm{a}}), \tag{4.21}$$

where the model is now dependent of both time and space. The error statistics are then described by the error covariance matrix $\boldsymbol{C}_{\psi\psi}^{\mathrm{f}}(t_k)$ which evolves according to the equation

$$\begin{aligned}
\boldsymbol{C}_{\psi\psi}^{\mathrm{f}}(t_k) = {} & \boldsymbol{G}'_{k-1}\boldsymbol{C}_{\psi\psi}^{\mathrm{a}}(t_{k-1})\boldsymbol{G}'^{\mathrm{T}}_{k-1} + \boldsymbol{C}_{qq}(t_{k-1}) \\
& + \boldsymbol{G}'_{k-1}\boldsymbol{\Theta}_{\psi\psi\psi}(t_{k-1})\boldsymbol{\mathcal{H}}^{\mathrm{T}}_{k-1} + \frac{1}{4}\boldsymbol{\mathcal{H}}_{k-1}\boldsymbol{\Gamma}_{\psi\psi\psi\psi}(t_{k-1})\boldsymbol{\mathcal{H}}^{\mathrm{T}}_{k-1} \\
& + \frac{1}{3}\boldsymbol{G}'_{k-1}\boldsymbol{\Gamma}_{\psi\psi\psi\psi}(t_{k-1})\boldsymbol{\mathcal{T}}^{\mathrm{T}}_{k-1} \\
& + \frac{1}{4}\boldsymbol{\mathcal{H}}_{k-1}\boldsymbol{C}_{\psi\psi}^{\mathrm{a}}(t_{k-1})\boldsymbol{C}_{\psi\psi}^{\mathrm{aT}}(t_{k-1})\boldsymbol{\mathcal{H}}^{\mathrm{T}}_{k-1} \\
& + \frac{1}{6}\boldsymbol{\mathcal{H}}_{k-1}\boldsymbol{C}_{\psi\psi}^{\mathrm{a}}(t_{k-1})\boldsymbol{\Theta}_{\psi\psi\psi}^{\mathrm{T}}(t_{k-1})\boldsymbol{\mathcal{T}}^{\mathrm{T}}_{k-1} \\
& + \frac{1}{36}\boldsymbol{\mathcal{T}}_{k-1}\boldsymbol{\Theta}_{\psi\psi\psi}(t_{k-1})\boldsymbol{\Theta}_{\psi\psi\psi}^{\mathrm{T}}(t_{k-1})\boldsymbol{\mathcal{T}}^{\mathrm{T}}_{k-1} + \cdots ,
\end{aligned} \tag{4.22}$$

where $\boldsymbol{C}_{qq}(t_{k-1})$ is the model error covariance matrix, $\boldsymbol{G}'_{k-1}$ is the Jacobi matrix or tangent linear operator,

$$\boldsymbol{G}'_{k-1} = \left.\frac{\partial \boldsymbol{G}(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}}\right|_{\psi_{k-1}}, \tag{4.23}$$

$\boldsymbol{\Theta}_{\psi\psi\psi}$ is the third order statistical moment, $\boldsymbol{\Gamma}_{\psi\psi\psi\psi}$ is the fourth order statistical moment, $\boldsymbol{\mathcal{H}}$ is the Hessian, consisting of second order derivatives of the nonlinear model operator, and $\boldsymbol{\mathcal{T}}$ is an operator containing the third order derivatives of the model operator.

The EKF is based on the assumption that the contribution from all the higher order terms in (4.22), are negligible. By discarding these terms we are left with the approximate error covariance equation

$$\boldsymbol{C}_{\psi\psi}^{\mathrm{f}}(t_k) \simeq \boldsymbol{G}'_{k-1}\boldsymbol{C}_{\psi\psi}^{\mathrm{a}}(t_{k-1})\boldsymbol{G}'^{\mathrm{T}}_{k-1} + \boldsymbol{C}_{qq}(t_{k-1}). \tag{4.24}$$

The analogy between the vector and scalar cases is obvious.

A discussion of the case where higher order approximations for the error variance evolution is used, is given by *Miller* (1994).

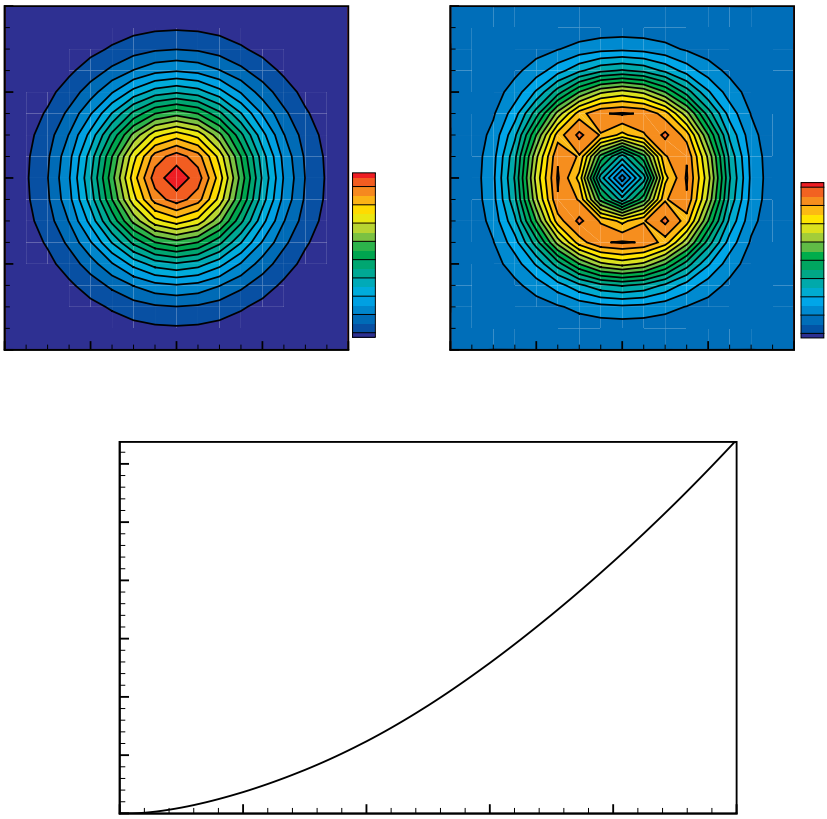

Fig. 4.3. Example of an EKF experiment from *Evensen* (1992). The upper left plot shows the stream function defining the velocity field of a stationary eddy. The upper right plot shows the resulting error variance in the model domain after integration from $t = 0$ till $t = 25$, note the large errors at locations where velocities are high. The lower plot illustrates the exponential time evolution of the estimated variance averaged over the model domain

4.2.3 Example using the extended Kalman filter

Evensen (1992) provided the first application of the EKF with a nonlinear ocean circulation model. The model was a multi-layer quasi-geostrophic model which represents well the mesoscale ocean variability. It solves a conservation equation for potential vorticity.

In *Evensen* (1992) properties of the EKF with this particular model were examined. It was found that the linear evolution equation for the error covariance matrix leads to an unbounded linear instability. This was demonstrated

in an experiment using a steady background flow defined by an eddy standing on a flat bottom and with no beta effect (see left plot in Fig. 4.3). Thus, vorticity is just advected along the stream lines with a velocity defined by the stream function.

The particular stream function results in a velocity shear and thus supports standard sheared flow instability. Thus, if we add a perturbation and advect it using the linearized equations the perturbation will grow exponentially. This is exactly what is observed in the upper right and lower plots of Fig. 4.3. We started out with an initial variance equal to one in all of the model domain and observe a strong error variance growth at locations of large velocity and velocity shear in the eddy. The estimated mean square errors, which equals the trace of $\boldsymbol{C}_{\psi\psi}$ divided by the number of grid points, indicate the exponential error variance growth.

This linear instability is not realistic. In the real world we would expect the instability to saturate at a certain climatological amplitude. As an example, in the atmosphere it is always possible to define a maximum and minimum pressure which is never exceeded, and the same applies for the eddy field in the ocean. A variance estimate which indicates unphysical amplitudes of the variability cannot be accepted, and this is in fact what the EKF may provide.

The main result from this work was the finding of an apparent closure problem in the error covariance evolution equation. The EKF applies a closure scheme where third- and higher-order moments in the error covariance evolution equation are discarded. This results in an unbounded error variance growth or linear instability in the error covariance equation in some dynamical models. If an exact error covariance evolution equation could be used all linear instabilities will saturate due to nonlinear effects. This saturation is missing in the EKF, as was later confirmed by *Miller et al.* (1994), *Gauthier et al.* (1993) and *Bouttier* (1994).

In particular *Miller et al.* (1994) gave a comprehensive discussion on the application of the EKF with the chaotic Lorenz model. The too simplified closure resulted in an estimated solution which was only acceptable in a fairly short time interval, and thereafter unreliable. This was explained by a poor prediction of error covariances $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$, resulting in insufficient gain $\boldsymbol{K}$, because of a decaying mode which reflects the stability of the attractor.

A generalization of the EKF, where third and fourth order moments and evolution equations for these were included, was also examined by *Miller et al.* (1994) and it was shown that this more sophisticated closure scheme provided a better evolution of error statistics which also resulted in sufficient gain to keep the estimate on track.

4.2.4 Extended Kalman filter for the mean

The previous derivation is the most commonly used for the EKF. A weakness of the formulation is that the so-called central forecast is used as the estimate. The central forecast is a single model realization initialized with the best

estimate of the initial state. For nonlinear dynamics the central forecast is not equal to the expected value, and it is not clear how it shall be interpreted.

A different approach is to derive a model for the evolution of the first moment or mean. This is done by expanding $G(\psi)$ around $\overline{\psi}$ to get

$$G(\psi) = G\big(\overline{\psi}\big) + G'\big(\overline{\psi}\big)\big(\psi - \overline{\psi}\big) + \frac{1}{2}G''\big(\overline{\psi}\big)\big(\psi - \overline{\psi}\big)^2 + \cdots . \tag{4.25}$$

Inserting this in (4.13) and taking the expectation or ensemble average results in the equation

$$\overline{\psi_k} = G\big(\overline{\psi_{k-1}}\big) + \frac{1}{2}G''\big(\overline{\psi_{k-1}}\big)C_{\psi\psi}(t_{k-1}) + \cdots . \tag{4.26}$$

In the vector case this equation becomes

$$\overline{\boldsymbol{\psi}_k} = \boldsymbol{G}\big(\overline{\boldsymbol{\psi}_{k-1}}\big) + \frac{1}{2}\mathcal{H}_{k-1}\boldsymbol{C}_{\psi\psi}(t_{k-1}) + \cdots . \tag{4.27}$$

One may argue that for a statistical estimator it makes more sense to work with the mean than a central forecast, after all, the central forecast does not have any statistical interpretation. This can be illustrated by running an atmospheric model without assimilation updates. The central forecast then becomes just one realization out of infinitively many possible realizations and it is not clear how one may relate this to the climatological error covariance estimate. On the other hand the equation for the mean will provide an estimate which converges towards the climatological mean and the covariance estimate thus describes the error variance of the climatological mean. Until now, all applications of the EKF for data assimilation in ocean and atmospheric models have used an equation for the central forecast. However, the interpretation using the equation for the mean will later on support the development of the Ensemble Kalman Filter.

4.2.5 Discussion

There are two major drawbacks of the Kalman filter for data assimilation in high dimensional and nonlinear dynamical models.

The first is related to storage and computational issues. If the dynamical model has n unknowns in the state vector, then the error covariance matrix $\boldsymbol{C}_{\psi\psi}$ has n^2 unknowns. Furthermore, the evolution of the error covariance matrix in time requires the cost of $2n$ model integrations. Thus, clearly, the KF and EKF in the present form, can only be practically used with fairly low-dimensional dynamical models.

The second issue is related to the use of the EKF with nonlinear dynamical models, which requires a linearization when deriving the error covariance evolution equation. This linearization leads to a poor error covariance evolution and for some models unstable error covariance growth. This may be resolved

using higher order closure schemes. Unfortunately, such an approach is not practical for a high dimensional model, since the fourth order moment requires storage of n^4 elements. In general one may conclude that a more consistent closure is needed in the error covariance equation.

4.3 Ensemble Kalman filter

Another sequential data assimilation method which has received a lot of attention is named the Ensemble Kalman Filter (EnKF). The method was originally proposed as a stochastic or Monte Carlo alternative to the deterministic EKF by *Evensen* (1994a). The EnKF was designed to resolve the two major problems related to the use of the EKF with nonlinear dynamics in large state spaces, i.e. the use of an approximate closure scheme and the huge computational requirements associated with the storage and forward integration of the error covariance matrix.

The EnKF has gained popularity because of its simple conceptual formulation and relative ease of implementation, e.g. it requires no derivation of a tangent linear operator or adjoint equations and no integrations backward in time. Furthermore, the computational requirements are affordable and comparable to other popular sophisticated assimilation methods such as the representer method by *Bennett* (1992), *Bennett et al.* (1993), *Bennett and Chua* (1994), *Bennett et al.* (1996) and the 4DVAR method which has been much studied by the meteorological community (see e.g. *Talagrand and Courtier*, 1987, *Courtier and Talagrand*, 1987, *Courtier et al.*, 1994, *Courtier*, 1997).

We will adapt a three stage presentation starting with the representation of error statistics using an ensemble of model states, then an alternative to the traditional error covariance equation is proposed for the prediction of error statistics, and finally a consistent analysis scheme is presented.

4.3.1 Representation of error statistics

The error covariance matrices for the predicted and the analyzed estimate, $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$ and $\boldsymbol{C}^{\mathrm{a}}_{\psi\psi}$, are in the Kalman filter defined in terms of the true state as

$$\boldsymbol{C}^{\mathrm{f}}_{\psi\psi} = \overline{\left(\boldsymbol{\psi}^{\mathrm{f}} - \boldsymbol{\psi}^{\mathrm{t}}\right)\left(\boldsymbol{\psi}^{\mathrm{f}} - \boldsymbol{\psi}^{\mathrm{t}}\right)^{\mathrm{T}}}, \tag{4.28}$$

$$\boldsymbol{C}^{\mathrm{a}}_{\psi\psi} = \overline{\left(\boldsymbol{\psi}^{\mathrm{a}} - \boldsymbol{\psi}^{\mathrm{t}}\right)\left(\boldsymbol{\psi}^{\mathrm{a}} - \boldsymbol{\psi}^{\mathrm{t}}\right)^{\mathrm{T}}}, \tag{4.29}$$

where the ensemble averaging defined by the overline converges to the expectation value in the case of an infinite ensemble size. However, the true state is not known, and we therefore define the ensemble covariance matrices around the ensemble mean $\overline{\boldsymbol{\psi}}$,

$$\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} = \overline{\left(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right)\left(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right)^{\mathrm{T}}}, \tag{4.30}$$

$$\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{a}} = \overline{\left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{a}}}\right)\left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{a}}}\right)^{\mathrm{T}}}, \tag{4.31}$$

where now the overline denote an average over the ensemble. Thus, we can use an interpretation where the ensemble mean is the best estimate and the spreading of the ensemble around the mean is a natural definition of the error in the ensemble mean.

Since the error covariances as defined in (4.30) and (4.31) are defined as ensemble averages, there will clearly exist an infinite number of ensembles with an error covariance equal to $\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}$. Thus, instead of storing a full covariance matrix, we can represent the same error statistics using an appropriate ensemble of model states. Given an error covariance matrix, an ensemble of finite size will provide an approximation to the error covariance matrix. However, when the size of the ensemble N increases, the errors in the Monte Carlo sampling will decrease proportional to $1/\sqrt{N}$.

Suppose now that we have N model states in the ensemble, each of dimension n. Each of these model states can be represented as a single point in an n-dimensional state space. All the ensemble members together will constitute a cloud of points in the state space. Such a cloud of points can, in the limit when N goes to infinity, be described using a probability density function

$$f(\boldsymbol{\psi}) = \frac{dN}{N}, \tag{4.32}$$

where dN is the number of points in a small unit volume and N is the total number of points. With knowledge about either $f(\boldsymbol{\psi})$ or the ensemble representing $f(\boldsymbol{\psi})$ we can calculate whichever statistical moments (such as mean, covariances etc.) we want whenever they are needed.

The conclusion so far is that the information contained by a full probability density function can be exactly represented by an infinite ensemble of model states.

4.3.2 Prediction of error statistics

In *Evensen* (1994a) it was shown that a Monte Carlo method can be used to solve an equation for the time evolution of the probability density of the model state, as an alternative to using the approximate error covariance equation in the EKF.

For a nonlinear model where we appreciate that the model is not perfect and contains model errors, we can write it as a stochastic differential equation as

$$d\boldsymbol{\psi} = \boldsymbol{G}(\boldsymbol{\psi})dt + \boldsymbol{h}(\boldsymbol{\psi})\,d\boldsymbol{q}. \tag{4.33}$$

This equation states that an increment in time will yield an increment in $\boldsymbol{\psi}$, which in addition, is influenced by a random contribution from the stochastic

forcing term $\boldsymbol{h}(\boldsymbol{\psi})d\boldsymbol{q}$, representing the model errors. The $d\boldsymbol{q}$ term describes a vector Brownian motion process with covariance $\boldsymbol{C}_{qq}dt$. The nonlinear model operator $\boldsymbol{G}$ is not an explicit function of the random variable $d\boldsymbol{q}$ so the Ito interpretation of the stochastic differential equation is used instead of the Stratonovich interpretation (see *Jazwinski*, 1970).

When additive Gaussian model errors forming a Markov process are used one can derive the Fokker-Planck equation (also named Kolmogorov's equation) which describes the time evolution of the probability density $f(\boldsymbol{\psi})$ of the model state,

$$\frac{\partial f}{\partial t} + \sum_i \frac{\partial (g_i f)}{\partial \psi_i} = \frac{1}{2} \sum_{i,j} \frac{\partial^2 f(\boldsymbol{h}\boldsymbol{C}_{qq}\boldsymbol{h}^T)_{ij}}{\partial \psi_i \partial \psi_j}, \tag{4.34}$$

where g_i is the component number i of the model operator $\boldsymbol{G}$ and $\boldsymbol{h}\boldsymbol{C}_{qq}\boldsymbol{h}^{\mathrm{T}}$ is the covariance matrix for the model errors.

This equation does not apply any important approximations and can be considered as the fundamental equation for the time evolution of error statistics. A detailed derivation is given in *Jazwinski* (1970). The equation describes the change of the probability density in a local "volume" which is dependent on the divergence term describing a probability flux into the local "volume" (impact of the dynamical equation) and the diffusion term which tends to flatten the probability density due to the effect of stochastic model errors. If (4.34) could be solved for the probability density function, it would be possible to calculate statistical moments like the mean and the error covariance for the model forecast to be used in the analysis scheme.

A linear model for a Gauss-Markov process in which the initial condition is assumed to be taken from a normal distribution will have a probability density which is completely characterized by its mean and covariance for all times. One can then derive exact equations for the evolution of the mean and the covariance as a simpler alternative than solving the full Kolmogorov's equation. Such moments of Kolmogorov's equation, including the error covariance (4.12), are easy to derive, and several methods are illustrated by *Jazwinski* (1970, examples 4.19–4.21). This is actually what is done in the KF.

For a nonlinear model, the mean and covariance matrix will not in general characterize the time evolution of $f(\boldsymbol{\psi})$. They do, however, determine the mean path and the dispersion about that path, and it is possible to solve approximate equations for the moments, which is the procedure characterizing the EKF.

The EnKF applies a so-called Markov Chain Monte Carlo (MCMC) method to solve (4.34). The probability density is then represented by a large ensemble of model states as discussed in the previous section. By integrating these model states forward in time according to the model dynamics, as described by the stochastic differential (4.33), this ensemble prediction is equivalent to solving the Fokker Planck equation using a MCMC method.

Different dynamical models can have stochastic terms embedded within the nonlinear model operator and the derivation of the associated Fokker Planck equation may become very complex. Fortunately, the Fokker Planck equation is not needed, since it is sufficient to know that it exists and the MCMC method solves it.

4.3.3 Analysis scheme

The KF analysis scheme uses the definitions of $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$ and $\boldsymbol{C}^{\mathrm{a}}_{\psi\psi}$ as given by (4.28) and (4.29). We will now give a derivation of the analysis scheme using the ensemble covariances as defined by (4.30) and (4.31).

As was shown by *Burgers et al.* (1998) it is essential that the observations are treated as random variables having a distribution with mean equal to the first guess observations and covariance equal to $\boldsymbol{C}_{\epsilon\epsilon}$. Thus, we start by defining an ensemble of observations

$$\boldsymbol{d}_j = \boldsymbol{d} + \boldsymbol{\epsilon}_j, \tag{4.35}$$

where j counts from 1 to the number of ensemble members N. It is ensured that the simulated random measurement errors have mean equal to zero. Next we define the ensemble covariance matrix of the measurement errors as

$$\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon} = \overline{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}}, \tag{4.36}$$

and, of course, in the limit of an infinite ensemble size this matrix will converge towards the prescribed error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$ used in the standard Kalman filter.

The following discussion is valid both using an exactly prescribed $\boldsymbol{C}_{\epsilon\epsilon}$ and an ensemble representation $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ of $\boldsymbol{C}_{\epsilon\epsilon}$. The use of $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ introduces an additional approximation which sometimes is convenient when implementing the analysis scheme. This approximation can be justified since normally the true observation error covariance matrix is poorly known and the errors introduced by the ensemble representation can be made less than the uncertainty in the true $\boldsymbol{C}_{\epsilon\epsilon}$ by choosing a large enough ensemble size. Further, the use of an ensemble representation for $\boldsymbol{C}_{\epsilon\epsilon}$, has less impact than the use of an ensemble representation for $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$. Further, $\boldsymbol{C}_{\epsilon\epsilon}$ only appears in the computation of the coefficients for the influence functions $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\boldsymbol{M}^{\mathrm{T}}$ while $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}$ appears both in the computation of the coefficients and it determines the influence functions. Note, however that there are specific issues related to the rank of $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ when the number of measurements becomes large as is discussed in Chap. 14.

The analysis step in the EnKF consists of the following updates performed on each of the model state ensemble members

$$\boldsymbol{\psi}^{\mathrm{a}}_j = \boldsymbol{\psi}^{\mathrm{f}}_j + \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} \left(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right)^{-1} \left(\boldsymbol{d}_j - \boldsymbol{M}\boldsymbol{\psi}^{\mathrm{f}}_j\right). \tag{4.37}$$

With a finite ensemble size, this equation will be an approximation. Further, if the number of measurements is larger than the number of ensemble members,

the matrices $\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}}$ and $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ will be singular, and a pseudo inversion must be used (see Chap. 14).

Equation (4.37) implies that

$$\overline{\boldsymbol{\psi}^{\mathrm{a}}} = \overline{\boldsymbol{\psi}^{\mathrm{f}}} + \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}}\left(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right)^{-1}\left(\overline{\boldsymbol{d}} - \boldsymbol{M}\overline{\boldsymbol{\psi}^{\mathrm{f}}}\right), \tag{4.38}$$

where $\overline{\boldsymbol{d}} = \boldsymbol{d}$ is the first guess vector of measurements. Thus, the relation between the analyzed and predicted ensemble mean is identical to the relation between the analyzed and predicted state in the standard Kalman filter, apart from the use of $\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f,a}}$ and $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ instead of $\boldsymbol{C}^{\mathrm{f,a}}_{\psi\psi}$ and $\boldsymbol{C}_{\epsilon\epsilon}$. Note that the introduction of an ensemble of observations does not make any difference for the update of the ensemble mean since this does not affect (4.38).

If the mean $\overline{\boldsymbol{\psi}^{\mathrm{a}}}$ is considered to be the best estimate, then it is an arbitrary choice whether one updates the mean using the first guess observations $\boldsymbol{d}$, or if one updates each of the ensemble members using the perturbed observations (4.35). However, it will now be shown that by updating each of the ensemble members using the perturbed observations one also creates a new ensemble with the correct error statistics for the analysis. The updated ensemble can then be integrated forward in time till the next observation time.

We now derive the analyzed error covariance estimate resulting from the analysis scheme given above, but using the standard Kalman filter form for the analysis equations. First, (4.37) and (4.38) are used to obtain

$$\boldsymbol{\psi}^{\mathrm{a}}_j - \overline{\boldsymbol{\psi}^{\mathrm{a}}} = \left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)\left(\boldsymbol{\psi}^{\mathrm{f}}_j - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right) + \boldsymbol{K}_{\mathrm{e}}\left(\boldsymbol{d}_j - \overline{\boldsymbol{d}}\right), \tag{4.39}$$

where we have used the definition of the Kalman gain,

$$\boldsymbol{K}_{\mathrm{e}} = \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}}\left(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right)^{-1}. \tag{4.40}$$

The derivation is then as follows,

$$\begin{aligned}
\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{a}} &= \overline{\left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{a}}}\right)\left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{a}}}\right)^{\mathrm{T}}} \\
&= \overline{\left(\left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)\left(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right) + \boldsymbol{K}_{\mathrm{e}}\left(\boldsymbol{d} - \overline{\boldsymbol{d}}\right)\right)\left(\cdots\right)^{\mathrm{T}}} \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)\overline{\left(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right)\left(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right)^{\mathrm{T}}}\left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)^{\mathrm{T}} \\
&\qquad + \boldsymbol{K}_{\mathrm{e}}\overline{\left(\boldsymbol{d} - \overline{\boldsymbol{d}}\right)\left(\boldsymbol{d} - \overline{\boldsymbol{d}}\right)^{\mathrm{T}}}\boldsymbol{K}^{\mathrm{T}}_{\mathrm{e}} \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\left(\boldsymbol{I} - \boldsymbol{M}^{\mathrm{T}}\boldsymbol{K}^{\mathrm{T}}_{\mathrm{e}}\right) + \boldsymbol{K}_{\mathrm{e}}\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\boldsymbol{K}^{\mathrm{T}}_{\mathrm{e}} \\
&= \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} - \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{K}^{\mathrm{T}}_{\mathrm{e}} \\
&\qquad + \boldsymbol{K}_{\mathrm{e}}(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon})\boldsymbol{K}^{\mathrm{T}}_{\mathrm{e}} \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}.
\end{aligned} \tag{4.41}$$

The last expression in this equation is the traditional result for the minimum error covariance found in the KF analysis scheme. This implies that the EnKF

in the limit of an infinite ensemble size will give exactly the same result in the computation of the analysis as the KF and EKF. Note that this derivation clearly states that the observations $\boldsymbol{d}$ must be treated as random variables to get the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}}$ into the expression. It has been assumed that the distributions used to generate the model state ensemble and the observation ensemble are independent. In Chap. 13 we will see that it is also possible to derive deterministic analysis schemes where the perturbation of measurements is avoided. This reduces sampling errors but may introduce other problems.

Finally, it should be noted that the EnKF analysis scheme is approximate in the sense that it does not properly take into account non-Gaussian contributions in the prior for $\boldsymbol{\psi}$. In other words, it does not solve the Bayesian update equation for non-Gaussian pdfs. On the other hand, it is not a pure resampling of a Gaussian posterior distribution. Only the updates are linear and these are added to the prior non-Gaussian ensemble. Thus, the updated ensemble will inherit many of the non-Gaussian properties from the forecast ensemble. In summary, we have a very computational efficient analysis scheme where we avoid traditional resampling of the posterior, and the solution becomes something between a linear Gaussian update and a full Bayesian computation. This will be elaborated on in more detail in the following chapters.

4.3.4 Discussion

We now have a complete system of equations which constitutes the ensemble Kalman filter (EnKF), and the similarity with the standard Kalman filter is maintained both for the prediction of error covariances and in the analysis scheme. For linear dynamics the EnKF solution will converge exactly to the KF solution with increasing ensemble size.

We will now examine the forecast step a little further. In the EnKF each ensemble member evolves in time according to the stochastic model dynamics. The ensemble covariance matrix of the errors in the model equations, given by

$$\boldsymbol{C}_{qq}^{\mathrm{e}} = \overline{d\boldsymbol{q}_k d\boldsymbol{q}_k}^{\mathrm{T}}, \tag{4.42}$$

converges to $\boldsymbol{C}_{qq}$ in the limit of an infinite ensemble size.

The ensemble mean then evolves according to the equation

$$\begin{aligned} \overline{\boldsymbol{\psi}_{k+1}} &= \overline{\boldsymbol{G}(\boldsymbol{\psi}_k)} \\ &= \boldsymbol{G}(\overline{\boldsymbol{\psi}_k}) + \text{n.l.}, \end{aligned} \tag{4.43}$$

where n.l. represents the terms which may arise if $\boldsymbol{G}$ is nonlinear. Compare this equation with the approximate equation for the mean (4.27) used with the EKF, where only the first correction term is included. One of the advantages of the EnKF is that it models the exact equation for the mean and there is no closure assumption used since each ensemble member is integrated by the full nonlinear model. The only approximation is the limited size of the ensemble.

The error covariance of the ensemble evolves according to

$$\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{k+1} = \boldsymbol{G}' \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{k} \boldsymbol{G}'^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{qq} + \text{n.l.}, \tag{4.44}$$

where $\boldsymbol{G}'$ is the tangent linear operator evaluated at $\boldsymbol{\psi}$ in the current time step. This is again an equation of the same form as is used in the standard Kalman filter, except for the extra n.l.-terms that may appear if $\boldsymbol{G}$ is nonlinear as seen in (4.22). Implicitly, the EnKF retains all these terms also for the error covariance evolution and there is no closure approximation used.

For a linear dynamical model the sampled $\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}$ converges to $\boldsymbol{C}_{\psi\psi}$ for an infinite ensemble size, and independently of the model, $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ converges to $\boldsymbol{C}_{\epsilon\epsilon}$ and $\boldsymbol{C}^{\mathrm{e}}_{qq}$ converges to $\boldsymbol{C}_{qq}$. Thus, in this limit the KF and the EnKF are equivalent.

For nonlinear dynamics the EnKF includes the full effect of these terms and there are no linearizations or closure assumptions used. In addition, there is no need for a tangent linear operator or its adjoint, and this makes the method very easy to implement for practical applications.

This leads to an interpretation of the EnKF as a purely statistical Monte Carlo method where the ensemble of model states evolves in state space with the mean as the best estimate and the spreading of the ensemble as the error variance. At measurement times each observation is represented by another ensemble, where the mean is the actual measurement and the variance of the ensemble represents the measurement errors. Thus, we combine a stochastic prediction step with a stochastic analysis step.

4.3.5 Example with a QG model

Evensen and van Leeuwen (1996) proved the EnKF's capabilities with nonlinear dynamics in an application where Geosat radar altimeter data were assimilated into a quasi geostrophic (QG) model to study the ring-shedding process in the Agulhas current flowing along the southeast coast of South Africa. This was the first real application of an advanced sequential assimilation method for estimating the ocean circulation. It proved that the EnKF with its fully nonlinear evolution of error statistics could be used with nonlinear and unstable dynamical models. In addition it showed that the low computational cost of the EnKF allows for reasonably sized model grids to be used.

A series of plots of the analyzed estimates for the upper layer stream function is given in Fig. 4.4 for different time steps. The results were in good agreement with the assimilated data and the assimilation run was well constrained by the data.

A conclusion from this work was that the assimilation of data helped compensate for neglected physics in the model. The QG model has a too slow final wave steepening and ring shedding, caused by the lack of ageostrophic effects in the model. This was accounted for by the assimilation of the data.

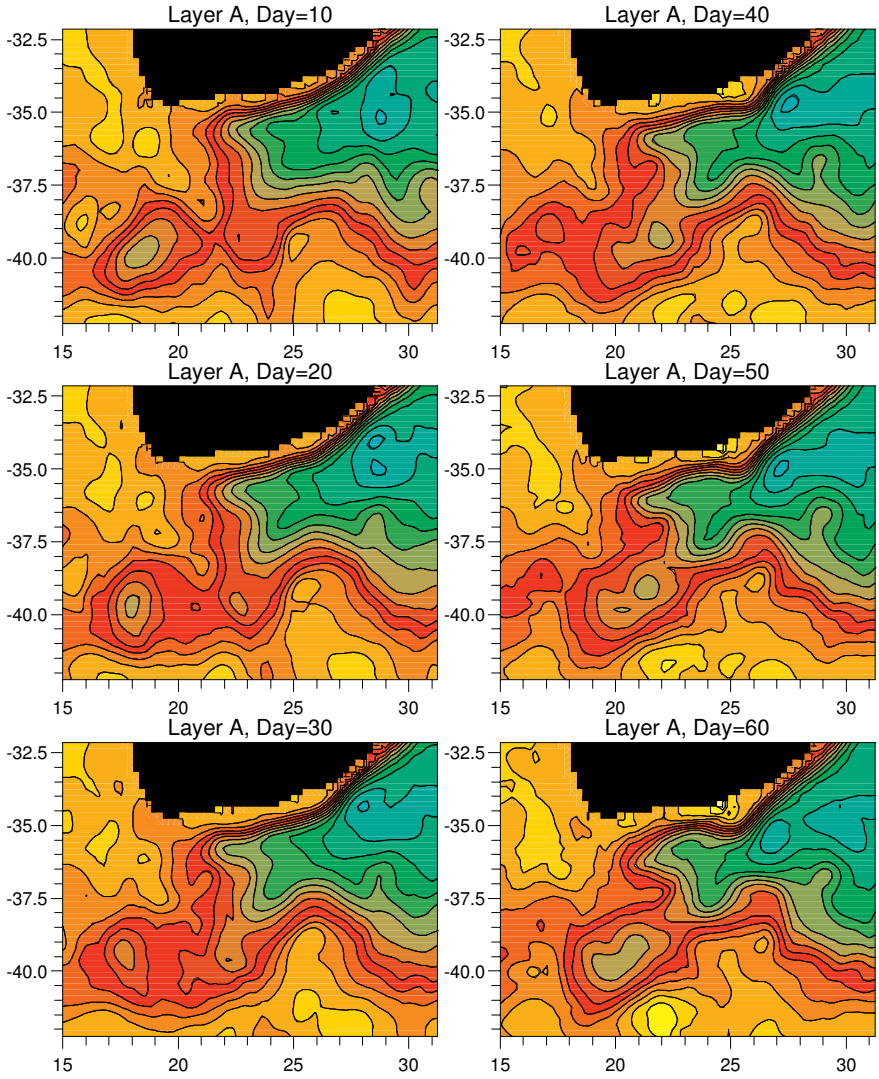

Fig. 4.4. Example of an EnKF experiment for the Agulhas current system from *Evensen and van Leeuwen* (1996)

In the experiment an ensemble size of 500 was used. The numerical grid consisted of two layers of 51×65 grid points, and the total number of unknowns was 6630, which is 13 times the number of ensemble members. The 500 ensemble members were sufficient to give a good representation of the gridded Geosat data and the space of possible model solutions.

5

Variational inverse problems

The purpose of this chapter is to introduce the basic formalism needed for properly formulating and solving linear variational inverse problems. Contrary to the sequential methods which update the model solution every time observations are available, variational methods seek an estimate in space and time where the estimate at a particular time is dependent on both past and future measurements.

We start by discussing a very simple example to illustrate the inverse problem and in particular the effect of including model errors. Thereafter a simple scalar model is used in a more typical illustration where the general formulation of the inverse problem is discussed and the Euler–Lagrange equations which determine the minimizing solution are derived.

Different methods are available for solving the Euler–Lagrange equations and we briefly discuss the popular representer method (see *Bennett*, 1992, 2002) which has proven extremely useful for solving linear and weakly nonlinear variational inverse problems.

5.1 Simple illustration

We will start with a very simple example to illustrate the mathematical properties of a variational problem and the difference between a weak and a strong constraint formulation. We define the simple model

$$\mathrm{d}\psi t = 1, \tag{5.1}$$

$$\psi(0) = 0, \tag{5.2}$$

$$\psi(1) = 2, \tag{5.3}$$

having one initial condition and one final condition. Clearly this is an overdetermined problem and it has no solution. However, if we relax the conditions by adding unknown errors to each of them the system becomes

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_5,

$$\mathrm{d}\psi t = 1 + q, \tag{5.4}$$
$$\psi(0) = 0 + a, \tag{5.5}$$
$$\psi(1) = 2 + b. \tag{5.6}$$

The system is now under-determined since we can get whatever solution we want by choosing the different error terms. A statistical hypothesis $\mathcal{H}_0$, is now needed for the error terms,

$$\begin{aligned} \overline{q(t)} &= 0, & \overline{q(t_1)q(t_2)} &= C_0\delta(t_1 - t_2), & \overline{q(t)a} &= 0, \\ \overline{a} &= 0, & \overline{a^2} &= C_0, & \overline{ab} &= 0, \\ \overline{b} &= 0, & \overline{b^2} &= C_0, & \overline{q(t)b} &= 0. \end{aligned} \tag{5.7}$$

That is, we assume that we know the statistical behaviour of the error terms through their first and second order moments. In this example the variances are all set equal to C_0 for simplicity.

It is now possible to seek the solution, which is as close as possible to the initial and final conditions while at the same time it almost satisfies the model equations, by minimizing the error terms in the form of a weak constraint penalty function

$$\mathcal{J}[\psi] = W_0 \int_0^1 (\mathrm{d}\psi t - 1)^2 \, dt + W_0\big(\psi(0) - 0\big)^2 + W_0\big(\psi(1) - 2\big)^2, \tag{5.8}$$

where W_0 is the inverse of the error variance C_0. Then ψ is an extremum of the penalty function if

$$\delta\mathcal{J}[\psi] = \mathcal{J}[\psi + \delta\psi] - \mathcal{J}[\psi] = \mathcal{O}\big(\delta\psi^2\big), \tag{5.9}$$

when $\delta\psi \to 0$. Now, using

$$\begin{aligned} \mathcal{J}[\psi + \delta\psi] = W_0 \int_0^1 (\mathrm{d}\psi t - 1 + \mathrm{d}\delta\psi t)^2 \, dt \\ + W_0\big(\psi(0) - 0 + \delta\psi(0)\big)^2 + W_0\big(\psi(1) - 2 + \delta\psi(1)\big)^2 \end{aligned} \tag{5.10}$$

in (5.9) and dropping the common nonzero factor $2W_0$, and all terms proportional to $\mathcal{O}\big(\delta\psi^2\big)$, we must have

$$\int_0^1 \mathrm{d}\delta\psi t \, (\mathrm{d}\psi t - 1) \, dt + \delta\psi(0)\big(\psi(0) - 0\big) + \delta\psi(1)\big(\psi(1) - 2\big) = 0, \tag{5.11}$$

or from integration by part,

$$\begin{aligned} \delta\psi \, (\mathrm{d}\psi t - 1)\big|_0^1 - \int_0^1 \delta\psi \frac{d^2\psi}{dt^2} \, dt \\ + \delta\psi(0)\big(\psi(0) - 0\big) + \delta\psi(1)\big(\psi(1) - 2\big) = 0. \end{aligned} \tag{5.12}$$

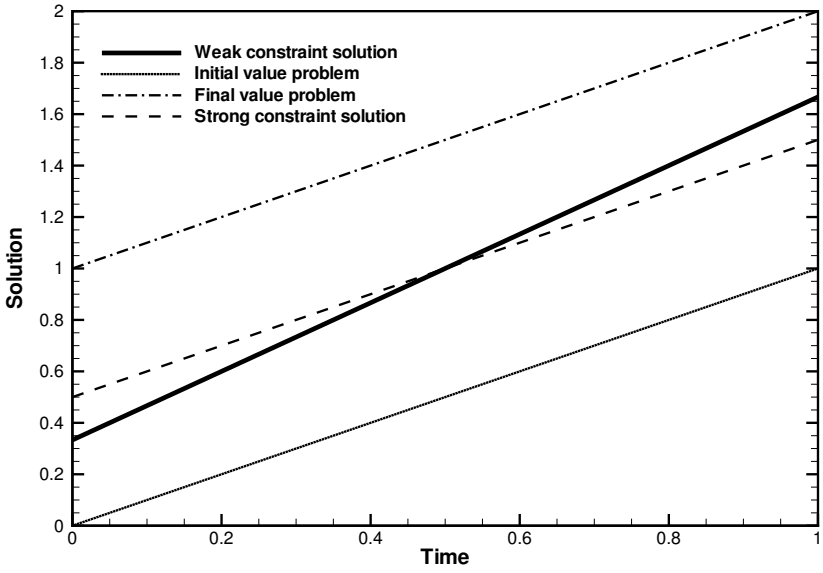

Fig. 5.1. Inverse solution from the simple example

This gives the following system of equations

$$\delta\psi(0)\ (-\mathrm{d}\psi t + 1 + \psi)|_{t=0} = 0, \tag{5.13}$$

$$\delta\psi(1)\ (\mathrm{d}\psi t - 1 + \psi - 2)|_{t=1} = 0, \tag{5.14}$$

$$\delta\psi \left(\frac{d^2\psi}{dt^2}\right) = 0, \tag{5.15}$$

or since $\delta\psi$ is arbitrary

$$\mathrm{d}\psi t - \psi = 1 \quad \text{for } t = 0, \tag{5.16}$$

$$\mathrm{d}\psi t + \psi = 3 \quad \text{for } t = 1, \tag{5.17}$$

$$\frac{d^2\psi}{dt^2} = 0. \tag{5.18}$$

This is an elliptic boundary value problem in time with mixed Dirichlet and Neumann boundary conditions. The general solution is

$$\psi = c_1 t + c_2, \tag{5.19}$$

and the constants in this case become $c_1 = 4/3$ and $c_2 = 1/3$.

In the case when we let the errors in the dynamical model go to zero, we approach the strong constraint limit where the dynamical model is assumed to be perfect. The strong constraint model solution is $\psi = t + c_2$ from (5.4), i.e. the slope is the one defined by the original model and no deviation of this

is allowed. The free constant c_2 will take a value between 0 and 1, depending on the relative magnitude between the weights on the two conditions. In this case with equal weight we will have $c_2 = 0.5$.

By allowing for model errors to account for an imperfect model, we will through a weak constraint variational formulation also allow for a deviation from the exact model trajectory. This is important for the mathematical conditioning of the variational problem, and we will later see that the weak constraint problem can be solved as easily as the strong constraint problem. The results from this example are shown in Fig. 5.1. The upper and lower curves are the respective solutions of the final and initial value problems. The weak constraint inverse estimate is seen to have a steeper slope than the exact model would allow, in order to obtain an estimate in better agreement with the two conditions. The strong constraint estimate is shown for comparison.

Finally, it is interesting to examine what the KF solution becomes in this example. The KF starts by solving the initial value problem until $t = 1$, thus for $t \in [0, 1)$ the solution is just $\psi(t) = t$. The initial error variance is set to C_0 and the increase of error variance when integrating the model over one time unit is also C_0. Thus for the prediction at $t = 1$, the error variance equals $2C_0$. The update equation (3.14) then becomes

$$\begin{aligned} \psi^{\mathrm{a}} &= \psi^{\mathrm{f}} + \frac{C^{\mathrm{f}}_{\psi\psi}}{C_{\epsilon\epsilon} + C^{\mathrm{f}}_{\psi\psi}} \left(d - \psi^{\mathrm{f}}\right) \\ &= 1 + \frac{2C_0}{C_0 + 2C_0}(2 - 1) \\ &= 5/3. \end{aligned} \tag{5.20}$$

This is in fact identical to the weak constraint variational solution at $t = 1$. Thus, could there be some connection between the problem solved by a variational method and the KF? In fact it will be shown later that for linear inverse problems, the KF and the weak constraint variational method, when both formulated consistently and using the same prior error statistics, give identical solutions at the final time. Thus for forecasting purposes, it does not matter which method is used.

5.2 Linear inverse problem

In this section we will define the inverse problem for a simple linear model and derive the Euler–Lagrange equations for a weak constraint variational formulation.

5.2.1 Model and observations

Assume now that we have given a simple scalar model with an initial condition and a set of measurements, all subject to errors,

$$\mathrm{d}\psi t = \psi + q, \tag{5.21}$$

$$\psi(0) = \Psi_0 + a, \tag{5.22}$$

$$\boldsymbol{\mathcal{M}}\big[\psi\big] = \boldsymbol{d} + \boldsymbol{\epsilon}. \tag{5.23}$$

The inverse problem can then be defined as finding an estimate which is close to the initial condition and the set of measurements, while at the same time it is almost satisfying the model equation.

5.2.2 Measurement functional

The linear measurement operator $\boldsymbol{\mathcal{M}}\big[\psi\big]$, of dimension M equal to the number of measurements, relates the observations $\boldsymbol{d}$ to the model state variable $\psi(t)$.

As an example, a direct measurement of $\psi(t)$ will have a measurement functional of the form

$$\mathcal{M}_i\big[\psi(t)\big] = \int_0^T \psi(t)\delta(t - t_i)\, dt = \psi(t_i), \tag{5.24}$$

where t_i is the measurement location in time and the subscript i denotes the component of the measurement functional.

Note for later use that the observation of the Dirac delta function becomes

$$\mathcal{M}_{i(2)}\big[\delta(t_1 - t_2)\big] = \int_0^T \delta(t_1 - t_2)\delta(t_2 - t_i)\, dt_2 = \delta(t_1 - t_i). \tag{5.25}$$

The subscript (2) on $\mathcal{M}_i$ defines the variable that the functional is operating on. Multiplying this with $\delta\psi(t_1)$ and integrating with respect to t_1 gives

$$\int_0^T \delta\psi(t_1)\mathcal{M}_{i(2)}\big[\delta(t_1 - t_2)\big]\, dt_1 = \delta\psi(t_i) = \mathcal{M}_{i(1)}\big[\delta\psi(t_1)\big]. \tag{5.26}$$

5.2.3 Comment on the measurement equation

In (3.2) and (3.23) we defined a measurement equation where we related the measurements to the true state, and $\boldsymbol{\epsilon}$ became the real measurement errors. Let us now write

$$\boldsymbol{d} = \boldsymbol{d}^{\mathrm{t}} + \boldsymbol{\epsilon}_d, \tag{5.27}$$

which defines $\boldsymbol{\epsilon}_d$ as the real measurement errors. In some cases we will also have that

$$\boldsymbol{\mathcal{M}}\big[\psi^{\mathrm{t}}\big] = \boldsymbol{d}^{\mathrm{t}} + \boldsymbol{\epsilon}_{\mathcal{M}}, \tag{5.28}$$

which states that there is an additional error associated with the measurement operator $\boldsymbol{\mathcal{M}}$. An example of such an error could be related to the interpolation on a numerical grid when a measurement is located in the center of a grid cell. We can then write

$$\begin{aligned} \boldsymbol{d} &= \boldsymbol{\mathcal{M}}\big[\psi^{\mathrm{t}}\big] + \boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}_{\mathcal{M}} \\ &= \boldsymbol{\mathcal{M}}\big[\psi^{\mathrm{t}}\big] + \boldsymbol{\epsilon}. \end{aligned} \tag{5.29}$$

Thus, we can say that the measurement is related to the true state through (5.29), where $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}_{\mathcal{M}}$ accounts for both measurement errors and errors in the measurement operator.

In the measurement equation (5.23) there is no reference to the true value ψ^{t}. In fact (5.23) is an equation which relates an estimate ψ to the measurements $\boldsymbol{d}$, allowing for a random error $\boldsymbol{\epsilon}$. Thus, we use this equation to impose an additional constraint to the model defined by (5.21) and (5.22). The random error $\boldsymbol{\epsilon}$ that represents both the errors in the measurements and the measurement operator now defines the accuracy of the measurement equation (5.23), just as the random errors a and q define the accuracy of the model equation and the initial condition.

5.2.4 Statistical hypothesis

Again a statistical hypothesis $\mathcal{H}_0$ is needed for describing the unknown error terms and we adapt the following:

$$\begin{aligned} \overline{q} &= 0, & \overline{q(t_1)q(t_2)} &= C_{qq}(t_1,t_2), & \overline{q(t)a} &= 0, \\ \overline{a} &= 0, & \overline{a^2} &= C_{aa}, & \overline{a\boldsymbol{\epsilon}} &= \mathbf{0}, \\ \overline{\boldsymbol{\epsilon}} &= \mathbf{0}, & \overline{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}} &= \boldsymbol{C}_{\epsilon\epsilon}, & \overline{q(t)\boldsymbol{\epsilon}} &= \mathbf{0}. \end{aligned} \tag{5.30}$$

In addition we will now define the functional inverse W_{qq} of the model error covariance C_{qq}, from the integral

$$\int_0^T C_{qq}(t_1,t_2)W_{qq}(t_2,t_3)\,dt_2 = \delta(t_1 - t_3), \tag{5.31}$$

and W_{aa} as the inverse of C_{aa}.

5.2.5 Weak constraint variational formulation

A weak constraint cost function can now be defined as

$$\begin{aligned} \mathcal{J}[\psi] = &\iint_0^T \left(\mathrm{d}\psi(t_1)t_1 - \psi(t_1)\right) W_{qq}(t_1,t_2) \left(\mathrm{d}\psi(t_2)t_2 - \psi(t_2)\right) dt_1\,dt_2 \\ &+ W_{aa}\big(\psi(0) - \Psi_0\big)^2 \\ &+ \Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big]\Big). \end{aligned} \tag{5.32}$$

Note that all first guesses, including the initial conditions, are penalized in (5.32). This is required in order to have a well-posed problem with a unique solution, as was shown by *Bennett and Miller* (1990).

The time-correlation in the model weight has a regularizing effect. Model errors are normally correlated in time, and the result of neglecting the time correlation is that the estimate will have discontinuous time derivatives at measurement locations.

5.2.6 Extremum of the penalty function

From standard variational calculus we know that ψ is an extremum if

$$\delta\mathcal{J} = \mathcal{J}[\psi + \delta\psi] - \mathcal{J}[\psi] = \mathcal{O}\big(\delta\psi^2\big), \tag{5.33}$$

when $\delta\psi \to 0$. Evaluating $\mathcal{J}[\psi + \delta\psi]$ we get

$$\begin{aligned}\mathcal{J}[\psi + \delta\psi] = &\iint_0^T \left(\mathrm{d}\psi t - \psi + \mathrm{d}\delta\psi t - \delta\psi\right)_1 W_{qq}(t_1, t_2) \\ &\qquad \times \left(\mathrm{d}\psi t - \psi + \mathrm{d}\delta\psi t - \delta\psi\right)_2 dt_1\, dt_2 \\ &+ W_{aa}\big(\psi(0) + \delta\psi(0) - \Psi_0\big)^2 \\ &+ \Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big] - \boldsymbol{\mathcal{M}}\big[\delta\psi\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big] - \boldsymbol{\mathcal{M}}\big[\delta\psi\big]\Big),\end{aligned} \tag{5.34}$$

where the subscripts 1 and 2 denote functions of t_1 and t_2. This can be rewritten as

$$\begin{aligned}\mathcal{J}[\psi + \delta\psi] = &\mathcal{J}[\psi] \\ &+ 2\iint_0^T \left(\mathrm{d}\delta\psi t - \delta\psi\right)_1 W_{qq}(t_1, t_2) \left(\mathrm{d}\psi t - \psi\right)_2 dt_1\, dt_2 \\ &+ 2W_{aa}\delta\psi(0)\big(\psi(0) - \Psi_0\big) \\ &- 2\boldsymbol{\mathcal{M}}^{\mathrm{T}}\big[\delta\psi\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big]\Big) + \mathcal{O}\big(\delta\psi^2\big).\end{aligned} \tag{5.35}$$

Now, evaluating the variational derivative (5.33) and requiring that the remaining terms are proportional to $\delta\psi^2$, we must have

$$\begin{aligned}&\iint_0^T \left(\mathrm{d}\delta\psi t - \delta\psi\right)_1 W_{qq}(t_1, t_2) \left(\mathrm{d}\psi t - \psi\right)_2 dt_1\, dt_2 \\ &+ W_{aa}\delta\psi(0)\big(\psi(0) - \Psi_0\big) \\ &- \boldsymbol{\mathcal{M}}^{\mathrm{T}}\big[\delta\psi\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big]\Big) = 0.\end{aligned} \tag{5.36}$$

This equation defines an extremum of the penalty function.

5.2.7 Euler–Lagrange equations

Start from (5.36) and define the "adjoint" variable λ as

$$\lambda(t_1) = \int_0^T W_{qq}(t_1, t_2)\,(\mathrm{d}\psi t - \psi)_2\, dt_2. \tag{5.37}$$

We now insert this in (5.36) and use integration by part to eliminate the derivative of the variation, i.e.

$$\int_0^T \mathrm{d}\delta\psi t \lambda dt = \delta\psi\lambda\Big|_{t=0}^{t=T} - \int_0^T \delta\psi \mathrm{d}\lambda t\, dt. \tag{5.38}$$

Then we use (5.26) to get the measurement term under the integral and proportional to $\delta\psi$.

Equation (5.36) then becomes

$$\begin{aligned} &- \int_0^T \delta\psi \Big(\mathrm{d}\lambda t + \lambda + \boldsymbol{\mathcal{M}}_{(2)}^{\mathrm{T}}[\delta(t_1 - t_2)] \boldsymbol{W}_{\epsilon\epsilon} \Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big] \Big) \Big)_1 dt_1 \\ &+ \delta\psi(0) \Big(W_{aa}\big(\psi(0) - \Psi_0\big) - \lambda(0) \Big) \\ &+ \delta\psi(T)\lambda(T) = 0. \end{aligned} \tag{5.39}$$

To obtain the final form of the Euler–Lagrange equations, we first multiply (5.37) with $W_{qq}(t, t_1)$ from the left, integrate in t_1 and use (5.31). This results in (5.40), given below. Further, assuming that the variation $\delta\psi$ in (5.39) is arbitrary, we get an equation for λ and conditions at time $t = 0$ and $t = T$. Thus, we have the following Euler–Lagrange equations:

$$\mathrm{d}\psi t - \psi = \int_0^T C_{qq}(t, t_1)\lambda(t_1)\, dt_1, \tag{5.40}$$

$$\psi(0) = \Psi_0 + C_{aa}\lambda(0), \tag{5.41}$$

$$\mathrm{d}\lambda t + \lambda = -\boldsymbol{\mathcal{M}}_{(2)}^{\mathrm{T}}[\delta(t - t_2)] \boldsymbol{W}_{\epsilon\epsilon} \Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big] \Big), \tag{5.42}$$

$$\lambda(T) = 0. \tag{5.43}$$

This system of Euler–Lagrange equations defines the extrema ψ of $\mathcal{J}$. The system consists of the original forward model forced by a term that is proportional to the adjoint variable λ in (5.40). The magnitude of this term is defined by the model error covariance C_{qq}, thus large model errors give a large contribution through the forcing term. The forward model is integrated from an initial condition which also contains a similar correction term proportional to the adjoint variable λ. The equation for λ can be integrated backward in time from a "final" condition, while forced by delta functions scaled by the residual between the measurement and forward model estimate ψ at each measurement location. Thus, the forward model requires knowledge of the adjoint variable to be integrated and the backward model uses the forward variable at measurement locations. We therefore have a coupled boundary value problem in time where the forward and backward models must be solved

simultaneously. The system comprises a well-posed problem and as long as the model is linear, it will have one unique solution, ψ.

The simplest approach for solving the Euler–Lagrange equations, may be to define an iteration. An iteration for the system (5.40)–(5.43) can be defined by using the previous iterate of λ when integrating the forward model. However, this iteration will generally not converge as pointed out by *Bennett* (1992).

5.2.8 Strong constraint approximation

A much-used approach relies on the assumption that the model is perfect, i.e. $C_{qq} = 0$ in (5.40). This leads to the so-called adjoint method originally proposed by *Talagrand and Courtier* (1987), *Courtier and Talagrand* (1987) and later discussed in a number of publications, e.g. *Courtier et al.* (1994), *Courtier* (1997). This removes the coupling to λ in the forward model. However, the system is still coupled through the λ appearing in the initial condition. One is then seeking the initial condition resulting in the model trajectory which is closest to the measurements. The so-called adjoint method uses this approach and defines a solution method where the system may be iterated as follows:

$$\mathrm{d}\psi^l t - \psi^l = 0, \tag{5.44}$$

$$\psi^l(0) = \psi^{l-1}(0) - \gamma\left(\psi^{l-1}(0) - \Psi_0 - C_{aa}\lambda^{l-1}(0)\right), \tag{5.45}$$

$$\mathrm{d}\lambda^l t + \lambda^l = -\boldsymbol{\mathcal{M}}_{(2)}^{\mathrm{T}}\left[\delta(t_1 - t_2)\right]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}_{(4)}\left[\psi_4^l\right]\Big), \tag{5.46}$$

$$\lambda^l(T) = 0. \tag{5.47}$$

The iteration defined for the initial condition uses that (5.41), or the expression in parantheses from (5.45), is the gradient of the penalty function with respect to the initial conditions. Thus, the iteration (5.45) is a standard gradient descent method where γ is the step length in the direction of the gradient. It should be noted that when $\psi = \psi(\boldsymbol{x})$, the dimension of the problem becomes infinite, and when $\psi(\boldsymbol{x})$ is discretized on a numerical grid, it becomes finite and equal to the number of grid nodes.

Note also that while the weak constraint formulation with proper knowledge of the error statistics defines a well-posed estimation problem where the estimate will be located within the statistical uncertainties of the first guesses, the strong constraint assumption violates this property of the inverse problem since one assumes that the model is better than it actually is.

5.2.9 Solution by representer expansions

For linear dynamics, it is possible to solve the Euler–Lagrange equations (5.40–5.43) exactly without using iterations. This can be done by assuming a solution of the form

$$\psi(t) = \psi_F(t) + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}(t), \tag{5.48}$$

$$\lambda(t) = \lambda_F(t) + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{s}(t), \tag{5.49}$$

as was previously also used for the time independent problem in (3.39). The dimensions of the vectors $\boldsymbol{b}$, $\boldsymbol{r}$ and $\boldsymbol{s}$ are all equal to the number of measurements, M. Assuming this form for the solution is equivalent to assuming that the minimizing solution is a first guess model solution ψ_F plus a linear combination of time dependent influence functions or representers $\boldsymbol{r}(t)$, one for each measurement. For a comprehensive discussion of this method see *Bennett* (1992, 2002). The practical implementation is discussed in great detail by *Chua and Bennett* (2001).

Inserting (5.48) and (5.49) into the Euler–Lagrange equations (5.40–5.43) and choosing first guesses ψ_F and λ_F that satisfy unforced exact equations

$$\mathrm{d}\psi_F t - \psi_F = 0, \tag{5.50}$$

$$\psi_F(0) = \Psi_0, \tag{5.51}$$

$$\mathrm{d}\lambda_F t + \lambda_F = 0, \tag{5.52}$$

$$\lambda_F(T) = 0, \tag{5.53}$$

gives us the following system of equations for the vector of representers $\boldsymbol{r}(t)$ and corresponding adjoints $\boldsymbol{s}(t)$:

$$\boldsymbol{b}^{\mathrm{T}}\left(\mathrm{d}\boldsymbol{r}t - \boldsymbol{r} - C_{qq}\boldsymbol{s}\right) = 0, \tag{5.54}$$

$$\boldsymbol{b}^{\mathrm{T}}\big(\boldsymbol{r}(0) - C_{aa}\boldsymbol{s}\big) = 0, \tag{5.55}$$

$$\boldsymbol{b}^{\mathrm{T}}\left(\mathrm{d}\boldsymbol{s}t + \boldsymbol{s}\right) = -\boldsymbol{\mathcal{M}}_{(2)}^{\mathrm{T}}\big[\delta(t-t_2)\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi_F + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}\big]\Big), \tag{5.56}$$

$$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{s}(T) = 0. \tag{5.57}$$

If we define $\boldsymbol{b}$ as

$$\boldsymbol{b} = \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi_F + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}\big]\Big), \tag{5.58}$$

then (5.56) becomes

$$\boldsymbol{b}^{\mathrm{T}}\left(\mathrm{d}\boldsymbol{s}t + \boldsymbol{s} + \boldsymbol{\mathcal{M}}_{(2)}\big[\delta(t-t_2)\big]\right) = 0, \tag{5.59}$$

and the coupling to the solution on the right side of (5.56) is removed.

Equation (5.58) is exactly the same as (3.38) and the derivation in (3.42–3.45) leads to the same linear system for the coefficients $\boldsymbol{b}$,

$$\Big(\boldsymbol{\mathcal{M}}^{\mathrm{T}}\big[\boldsymbol{r}\big] + \boldsymbol{C}_{\epsilon\epsilon}\Big)\,\boldsymbol{b} = \boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi_F\big]. \tag{5.60}$$

Given that $\boldsymbol{b}$ in general is nonzero, we now have the following set of equations in addition to (5.50–5.53):

$$\mathrm{d}\boldsymbol{r}t - \boldsymbol{r} = C_{qq}\boldsymbol{s}, \tag{5.61}$$

$$\boldsymbol{r}(0) = C_{aa}\boldsymbol{s}, \tag{5.62}$$

from (5.54) and (5.55) for the representers, and

$$\mathrm{d}\boldsymbol{s}t + \boldsymbol{s} = -\boldsymbol{\mathcal{M}}_{(2)}\big[\delta(t_1 - t_2)\big], \tag{5.63}$$

$$\boldsymbol{s}(T) = 0, \tag{5.64}$$

from (5.59) and (5.57) for the adjoints of the representers.

The equations for $\boldsymbol{s}$ can now be solved as a sequence of final value problems since they are decoupled from the forward equations for the representers. As soon as $\boldsymbol{s}$ is found the representers can be solved for. Together with the first guess solution ψ_F, found from solving (5.50)–(5.53), this provides the information needed for solving the system (5.60) for $\boldsymbol{b}$. The final estimate is then found by solving the Euler–Lagrange equation of the form

$$\mathrm{d}\psi t - \psi = \int_0^T C_{qq}(t, t_1)\lambda(t_1)\,dt_1, \tag{5.65}$$

$$\psi(0) = \Psi_0 + C_{aa}\lambda(0), \tag{5.66}$$

$$\mathrm{d}\lambda t + \lambda = -\boldsymbol{\mathcal{M}}_{(1)}^{\mathrm{T}}\big[\delta(t - t_1)\big]\boldsymbol{b}, \tag{5.67}$$

$$\lambda(T) = 0. \tag{5.68}$$

The numerical load is then $2M+3$ model integrations, but note that only two model states need to be stored in space and time. If the solution is constructed directly from (5.48) all the representers need to be stored.

Thus, the representer expansion decouples the Euler–Lagrange equations for the weak constraint problem, which can now be solved exactly without any iterations. Further, the dimension of the problem is the number of measurements, which is normally much less than the number of unknowns in a discrete state vector.

5.3 Representer method with an Ekman model

In *Eknes and Evensen* (1997), the representer method was implemented with an Ekman flow model and used to solve an inverse problem with a long time series of real velocity measurements. In addition a parameter estimation problem was treated but this will be discussed later. The model used is very simple and allows for a simple interpretation and demonstration of the method.

5.3.1 Inverse problem

The Ekman model was written in a nondimensional form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{k} \times \boldsymbol{u} = \frac{\partial}{\partial z}\left(A \frac{\partial \boldsymbol{u}}{\partial z}\right) + \boldsymbol{q}, \tag{5.69}$$

where $\boldsymbol{u}(z,t)$ is the horizontal velocity vector, $A = A(z)$ is the diffusion coefficient and $\boldsymbol{q}(z,t)$ is the stochastic model error. The initial conditions are given as

$$\boldsymbol{u}(z,0) = \boldsymbol{u}_0 + \boldsymbol{a}, \tag{5.70}$$

where $\boldsymbol{a}$ contains the stochastic errors in the first-guess initial condition $\boldsymbol{u}_0$. The boundary conditions for the model are

$$A \frac{\partial \boldsymbol{u}}{\partial z}\bigg|_{z=0} = \left(c_d \sqrt{u_a^2 + v_a^2}\right) \boldsymbol{u}_a + \boldsymbol{b}_0, \tag{5.71}$$

$$A \frac{\partial \boldsymbol{u}}{\partial z}\bigg|_{z=-H} = \boldsymbol{0} + \boldsymbol{b}_H, \tag{5.72}$$

where the position $z = 0$ is at the ocean surface and the lower boundary is at $z = -H$, c_d is the wind drag coefficient, $\boldsymbol{u}_a$ is the atmospheric wind speed, and $\boldsymbol{b}_0$ and $\boldsymbol{b}_H$ are the stochastic errors in the boundary conditions.

Now a set of measurements $\boldsymbol{d}$ of the true solution is assumed given and linearly related to the model variables by the measurement equation

$$\boldsymbol{\mathcal{M}}[\boldsymbol{u}] = \boldsymbol{d} + \boldsymbol{\epsilon}. \tag{5.73}$$

5.3.2 Variational formulation

A convenient variational formulation is

$$\begin{aligned}
\mathcal{J}[\boldsymbol{u}] = & \int_0^T dt_1 \int_0^T dt_2 \int_{-H}^0 dz_1 \int_{-H}^0 dz_2 \; \boldsymbol{q}^{\mathrm{T}}(z_1,t_1)\, \boldsymbol{W}_{qq}(z_1,t_1,z_2,t_2)\, \boldsymbol{q}(z_2,t_2) \\
& + \int_{-H}^0 dz_1 \int_{-H}^0 dz_2 \; \boldsymbol{a}^{\mathrm{T}}(z_1)\, \boldsymbol{W}_{aa}(z_1,z_2)\, \boldsymbol{a}(z_2) \\
& + \int_0^T dt_1 \int_0^T dt_2 \; \boldsymbol{b}_0^{\mathrm{T}}(t_1)\, \boldsymbol{W}_{b_0 b_0}(t_1,t_2)\, \boldsymbol{b}_0(t_2) \\
& + \int_0^T dt_1 \int_0^T dt_2 \; \boldsymbol{b}_H^{\mathrm{T}}(t_1)\, \boldsymbol{W}_{b_H b_H}(t_1,t_2)\, \boldsymbol{b}_H(t_2) \\
& + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{\epsilon}.
\end{aligned} \tag{5.74}$$

A simpler way of writing this may be

$$\begin{aligned}
\mathcal{J}[\boldsymbol{u}] = & \boldsymbol{q}^{\mathrm{T}} \bullet \boldsymbol{W}_{qq} \bullet \boldsymbol{q} \\
& + \boldsymbol{a}^{\mathrm{T}} \circ \boldsymbol{W}_{aa} \circ \boldsymbol{a} \\
& + \boldsymbol{b}_0^{\mathrm{T}} * \boldsymbol{W}_{b_0 b_0} * \boldsymbol{b}_0 \\
& + \boldsymbol{b}_H^{\mathrm{T}} * \boldsymbol{W}_{b_H b_H} * \boldsymbol{b}_H \\
& + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{\epsilon},
\end{aligned} \tag{5.75}$$

where the bullets mean integration both in space and time, the open circles mean integration in space, the asterisks mean integration in time. Here $\boldsymbol{W}_{\epsilon\epsilon}$ is the inverse of the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$, while the other weights are functional inverses of the respective covariances. For the model weight, this can be expressed as $\boldsymbol{C}_{qq} \bullet \boldsymbol{W}_{qq} = \delta(z_1 - z_3)\delta(t_1 - t_3)\boldsymbol{I}$, or written out,

$$\begin{aligned} \int_0^T dt_2 \int_{-H}^0 dz_2 \, \boldsymbol{C}_{qq}(z_1, t_1, z_2, t_2)\boldsymbol{W}_{qq}(z_2, t_2, z_3, t_3) \\ = \delta(z_1 - z_3)\delta(t_1 - t_3)\boldsymbol{I}. \end{aligned} \tag{5.76}$$

These weights determine the spatial and temporal scales for the physical problem and ensure smooth influences from the measurements.

5.3.3 Euler–Lagrange equations

Following the procedure outlined in the previous sections, we can derive the Euler–Lagrange equations. This leads to the forward model

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{k} \times \boldsymbol{u} = \frac{\partial}{\partial z}\left(A\frac{\partial \boldsymbol{u}}{\partial z}\right) + \boldsymbol{C}_{qq} \bullet \boldsymbol{\lambda}, \tag{5.77}$$

with initial conditions

$$\boldsymbol{u}|_{t=0} = \boldsymbol{u}_0 + \boldsymbol{C}_{aa} \circ \boldsymbol{\lambda}, \tag{5.78}$$

and boundary conditions

$$A\frac{\partial \boldsymbol{u}}{\partial z}\bigg|_{z=0} = c_d\sqrt{u_a^2 + v_a^2}\,\boldsymbol{u}_a + \boldsymbol{C}_{b_0 b_0} * \boldsymbol{\lambda}, \tag{5.79}$$

$$A\frac{\partial \boldsymbol{u}}{\partial z}\bigg|_{z=-H} = -\boldsymbol{C}_{b_H b_H} * \boldsymbol{\lambda}. \tag{5.80}$$

In addition we obtain the adjoint model

$$-\frac{\partial \boldsymbol{\lambda}}{\partial t} - \boldsymbol{k} \times \boldsymbol{\lambda} = \frac{\partial}{\partial z}\left(A\frac{\partial \boldsymbol{\lambda}}{\partial z}\right) + \boldsymbol{\mathcal{M}}^{\mathrm{T}}\left[\delta(z - z_2)\delta(t - t_2)\right]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\left[\boldsymbol{u}\right]\Big), \tag{5.81}$$

subject to the "final" condition

$$\boldsymbol{\lambda}|_{t=T} = \boldsymbol{0}, \tag{5.82}$$

and the boundary conditions

$$\frac{\partial \boldsymbol{\lambda}}{\partial z}\bigg|_{z=0, z=-H} = \boldsymbol{0}. \tag{5.83}$$

The system (5.77) to (5.83) is the Euler–Lagrange equations which here comprise a two-point boundary value problem in space and time, and since they are coupled they must be solved simultaneously.

5.3.4 Representer solution

Assuming a solution in the standard form

$$\boldsymbol{u}(z,t) = \boldsymbol{u}_F(z,t) + \sum_{i=1}^{M} b_i \boldsymbol{r}_i(z,t), \tag{5.84}$$

$$\boldsymbol{\lambda}(z,t) = \boldsymbol{\lambda}_F(z,t) + \sum_{i=1}^{M} b_i \boldsymbol{s}_i(z,t), \tag{5.85}$$

we find the equations for the representers and their adjoints. The M representers are found by solving the initial value problems

$$\frac{\partial \boldsymbol{r}_i}{\partial t} + \boldsymbol{k} \times \boldsymbol{r}_i = \frac{\partial}{\partial z}\left(A \frac{\partial \boldsymbol{r}_i}{\partial z}\right) + \boldsymbol{C}_{qq} \bullet \boldsymbol{s}_i, \tag{5.86}$$

with initial condition

$$\boldsymbol{r}_i|_{t=0} = \boldsymbol{C}_{aa} \circ \boldsymbol{s}_i, \tag{5.87}$$

and boundary conditions

$$A \frac{\partial \boldsymbol{r}_i}{\partial z}\bigg|_{z=0} = \boldsymbol{C}_{b_0 b_0} * \boldsymbol{s}_i, \tag{5.88}$$

$$A \frac{\partial \boldsymbol{r}_i}{\partial z}\bigg|_{z=-H} = -\boldsymbol{C}_{b_H b_H} * \boldsymbol{s}_i. \tag{5.89}$$

These equations are coupled to the adjoints of the representers $\boldsymbol{s}_i$, which satisfy the "final" value problems

$$-\frac{\partial \boldsymbol{s}_i}{\partial t} - \boldsymbol{k} \times \boldsymbol{s}_i = \frac{\partial}{\partial z}\left(A \frac{\partial \boldsymbol{s}_i}{\partial z}\right) + \mathcal{M}_i[\delta(z - z_2)\delta(t - t_2)], \tag{5.90}$$

with "final" conditions

$$\boldsymbol{s}_i|_{t=T} = \boldsymbol{0}, \tag{5.91}$$

and boundary conditions

$$\frac{\partial \boldsymbol{s}_i}{\partial z}\bigg|_{z=0,\, z=-H} = \boldsymbol{0}. \tag{5.92}$$

The coefficients $\boldsymbol{b}$ are again found by solving the system (5.60).

5.3.5 Example experiment

Here a simple example will be used to illustrate the method. A constant wind with $\boldsymbol{u}_a = (10 \text{ m s}^{-1}, 10 \text{ m s}^{-1})$ has been used to spin up the vertical velocity structure in the first-guess solution, starting with an initial condition $\boldsymbol{u}(z,0) =$

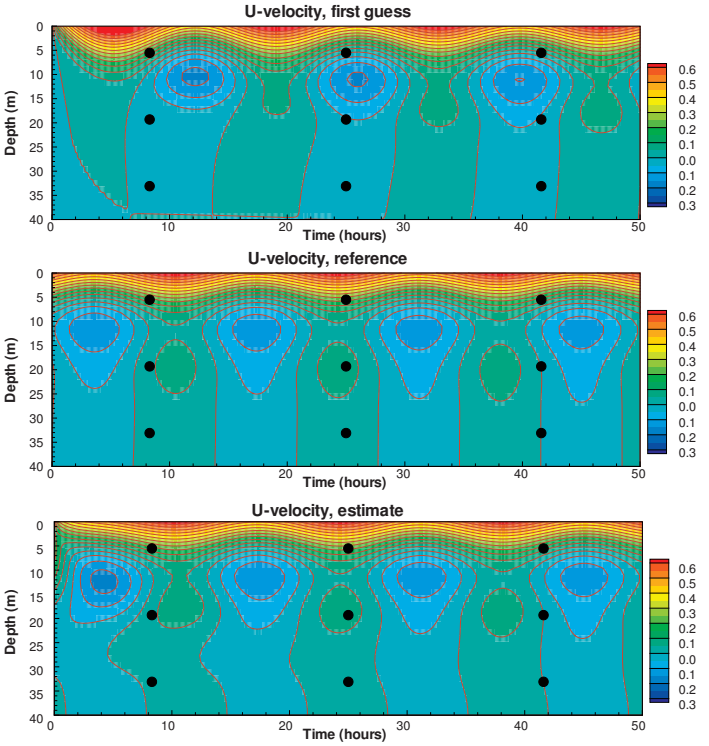

Fig. 5.2. The u components of *(from top to bottom)* the first-guess estimate $\boldsymbol{u}_F$, the reference case $\boldsymbol{u}$ and the inverse estimate of $\boldsymbol{u}$. The contour intervals are 0.05 m s^{-1} for all the velocity plots. The measurement locations are marked with a bullet. The v components are similar in structure and not shown. Reproduced from *Eknes and Evensen* (1997)

0 and then performing 50 hours of integration. The reference case, from which velocity data are extracted, is generated by continuing the integration for another 50 hours.

By measuring the reference case and adding Gaussian noise, nine simulated measurements of $\boldsymbol{u}$ were generated; that is, a total of 18 measurements of u and v components were used. The locations of the measurements are shown in Fig. 5.2.

All error terms are assumed to be unbiased and uncorrelated, and the error covariances were specified as follows:

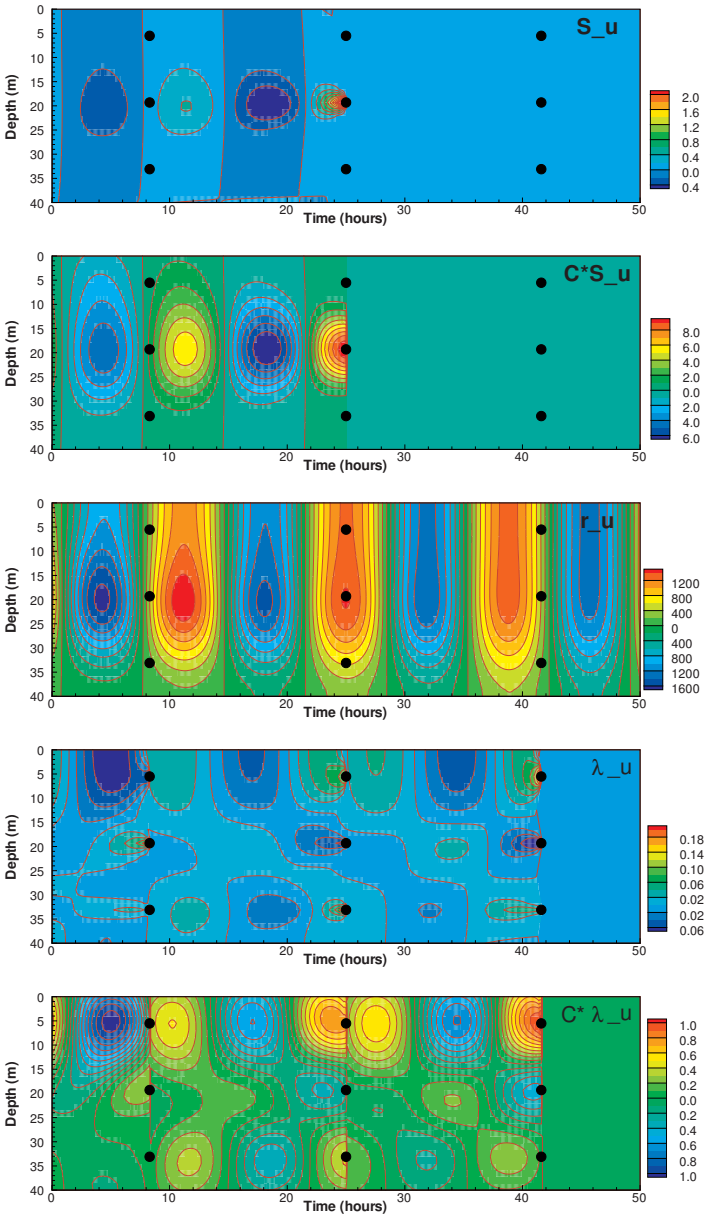

Fig. 5.3. The u component of *(top to bottom)* $\boldsymbol{s}_5$, $\boldsymbol{C}_{qq} \bullet \boldsymbol{s}_5$, $\boldsymbol{r}_5$, the adjoint $\boldsymbol{\lambda}$, and $\boldsymbol{C}_{qq} \bullet \boldsymbol{\lambda}$. The measurement locations are marked with a bullet. The v components are similar in structure and not shown. Reproduced from *Eknes and Evensen* (1997)

$$\boldsymbol{C}_{aa}(z_1, z_2) = \sigma_a^2 \exp\left(-\left(\frac{z_1 - z_2}{l_a}\right)^2\right)\boldsymbol{I}, \tag{5.93}$$

$$\boldsymbol{C}_{b_0 b_0}(t_1, t_2) = \sigma_{b_0}^2 \delta(t_1 - t_2)\boldsymbol{I}, \tag{5.94}$$

$$\boldsymbol{C}_{b_H b_H}(t_1, t_2) = \sigma_{b_H}^2 \delta(t_1 - t_2)\boldsymbol{I}, \tag{5.95}$$

$$\boldsymbol{C}_{qq}(z_1, t_1, z_2, t_2) = \sigma_q^2 \exp\left(-\left(\frac{z_1 - z_2}{l_q}\right)^2\right)\delta(t_1 - t_2)\boldsymbol{I}, \tag{5.96}$$

$$\boldsymbol{C}_{\epsilon\epsilon} = \sigma_o^2 \boldsymbol{I}. \tag{5.97}$$

Here it has been assumed that the model and the boundary errors are uncorrelated in time. This is convenient for computational reasons, but for more realistic applications, such a correlation should probably be included. The error variances all correspond to a 5–10% standard deviation of the variables or terms they represent errors in. This means that all first guesses and the model dynamics are assumed to be reasonably accurate and they all have similar impact on the inverse solution. Small perturbations in the weights give only small perturbations in the inverse estimate. However, large perturbations may cause problems; for example, with zero weights on some of the first guesses, the inverse problem may become ill-posed. The de-correlation lengths are similar to the characteristic length scales of the dynamical system. This ensures that the representers also become smooth with similar length scales as the dynamical solution.

The first-guess, the reference solution, and the inverse estimate are given in Fig. 5.2. The reference solution is regenerated quite well, even though the first-guess solution is out of phase with the reference case and the measurements do not resolve the time period of the oscillation. In fact, a single measurement may suffice for reconstructing the correct phase since the corresponding representer will carry the information both forward and backward in time, although the errors will be larger with less measurements. Note that the quality of the inverse estimate is poorest near the initial time. This is probably caused by a poor choice of weights for the initial conditions relative to the initial condition that was actually used.

To illustrate the solution procedure using the representer method in more detail, the u-components of the variables $\boldsymbol{s}_5$, $\boldsymbol{r}_5$, $\boldsymbol{\lambda}$, and the right-hand sides $\boldsymbol{C}_{qq} \bullet \boldsymbol{s}_5$ and $\boldsymbol{C}_{qq} \bullet \boldsymbol{\lambda}$, are given in Fig. 5.3. These plots demonstrate how the information from the measurements is taken into account and influences the solution. Measurement number five corresponds to the u component at the location $(z, t) = (-20.0, 25.0)$.

The upper plot shows the u-component of $\boldsymbol{s}_5$ and it is clear from (5.90) that it is forced by the δ-function at the measurement location. This information is then propagated backward in time while the u and v components interact during the integration.

Thereafter, $\boldsymbol{s}_i$ is used on the right-hand side of the forward equation for the representer and is also used to generate the initial and boundary conditions.

The convolution $\boldsymbol{C}_{qq} \bullet \boldsymbol{s}_5$, is a smoothing of $\boldsymbol{s}_5$ according to the covariance functions contained in $\boldsymbol{C}_{qq}$, as can be observed from the second plot in Fig. 5.3.

The representer $\boldsymbol{r}_5$ is smooth and is oscillating in time with a period reflecting the inertial oscillations described by the dynamical model. Note that the representers will have a discontinuous time derivative at the measurement location since the right-hand side $\boldsymbol{C}_{qq} \bullet \boldsymbol{s}_5$ is discontinuous there. However, if a correlation in time was allowed in $\boldsymbol{C}_{qq}$, then $\boldsymbol{C}_{qq} \bullet \boldsymbol{s}_5$ would be continuous and the representer $\boldsymbol{r}_5$ would be smooth.

After the representers have been calculated and measured to generate the representer matrix, the coefficient vector $\boldsymbol{b}$ is solved for and used in (5.81) to decouple the Euler–Lagrange equations. The u-component of $\boldsymbol{\lambda}$ (Fig. 5.3) illustrates how the various measurements have a different impact determined by values of the coefficients in $\boldsymbol{b}$, which again are determined by the quality of the first-guess solution versus the quality of the measurements and the residual between the measurements and the first-guess solution. After $\boldsymbol{\lambda}$ is found, the right-hand side in the forward model equation can be constructed through the convolution $\boldsymbol{C}_{qq} \bullet \boldsymbol{\lambda}$, and this field is given at the bottom of Fig. 5.3. Clearly, the role of this term is to force the solution to smooth the measurements.

5.3.6 Assimilation of real measurements

The representer implementation will now be examined using the LOTUS–3 data set (*Bowers et al.*, 1986) in a similar setup to the one used by *Yu and O'Brien* (1991, 1992). The LOTUS–3 measurements were collected in the northwestern Sargasso Sea (34° N, 70° W) during the summer of 1982. Current meters were fixed at depths of 5, 10, 15, 20, 25, 35, 50, 65, 75 and 100 m and measured the in situ currents. A wind recorder mounted on top of the LOTUS–3 tower measured the wind speeds. The sampling interval was 15 min, and the data used by *Yu and O'Brien* (1991, 1992) were collected in the period from June 30 to July 9, 1982. Here, data from the same time period are used. However, while *Yu and O'Brien* (1991, 1992) used all data collected during the 10 days, we have used a sub-sampled data set consisting of measurements collected at a 5-hour time interval at the depths 5, 25, 35, 50 and 75 m. The reason for not using all the measurements is to reduce the size of the representer matrix $\boldsymbol{\mathcal{M}}^{\mathrm{T}}[\boldsymbol{r}]$, and thus the computational cost. The inertial period and the vertical length scale are still resolved, and it is expected that mainly small-scale noise is rejected by subsampling the measurements.

The model was initialized by the first measurements collected on June 30, 1982. The standard deviation of the small-scale variability of the velocity observations was estimated to be close to 0.025 m s^{-1}, and this value was used to determine the error variances for the observations and the initial conditions. A similar approach was also used for the surface boundary conditions by looking at small-scale variability of the wind data. The model error variance was specified after a few test runs to give a relatively smooth inverse estimate

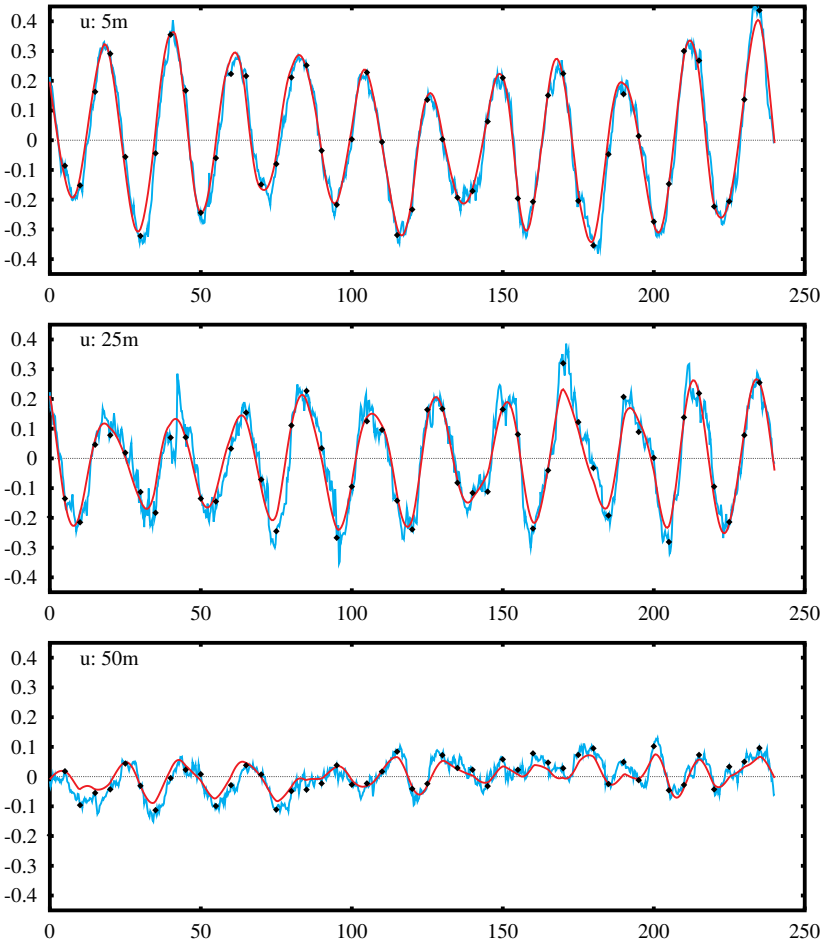

Fig. 5.4. Weak constraint results from the LOTUS–3 assimilation experiment from *Eknes and Evensen* (1997). Inverse estimate for the u component of velocity *(red lines)*, the time series of measurements *(blue lines)*, and the subsampled measurements *(bullets)*, at 5, 25 and 50 m

which seemed to be nearly consistent with the model dynamics and at the same time was close to the observations without over-fitting them.

The Ekman model describes wind driven currents and inertial oscillations only, while the measurements may also contain contributions from, e.g. pressure-driven currents. Therefore some drift in the measurements has been removed from the deepest moorings as was also done by *Yu and O'Brien* (1991, 1992).

The results from the inverse calculation are shown in Fig. 5.4 as time series of the u component of the velocity at various depths. The inverse esti-

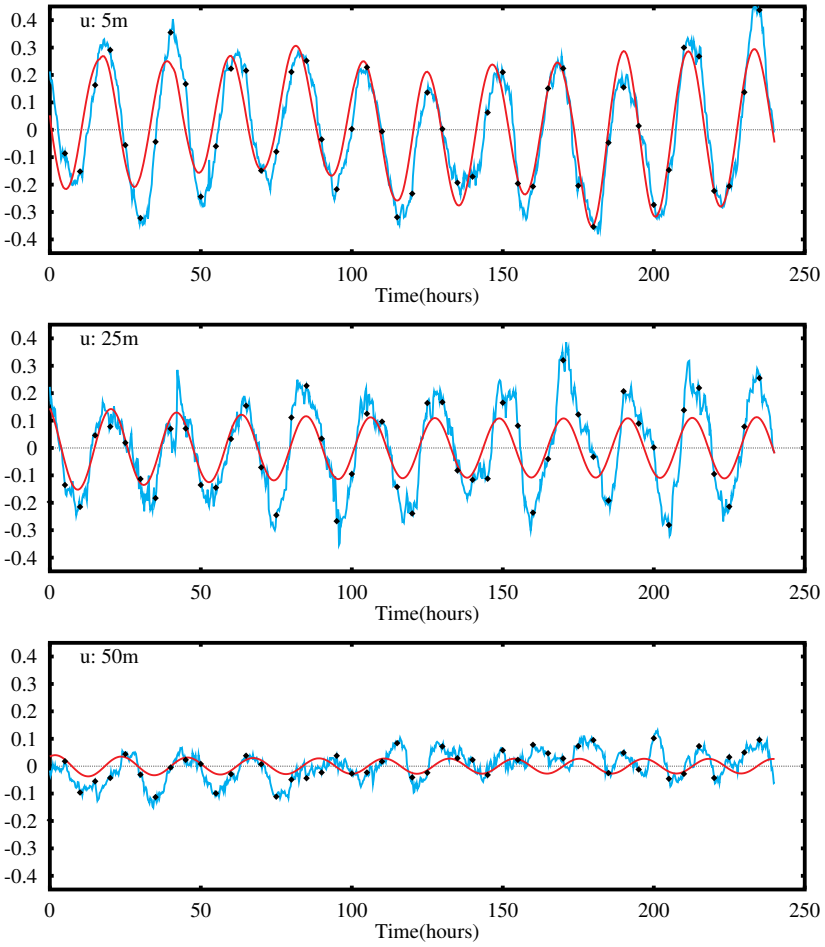

Fig. 5.5. Strong constraint results from the LOTUS–3 assimilation experiment from *Eknes and Evensen* (1997). Inverse estimate for the u component of velocity *(red lines)*, the time series of measurements *(blue lines)*, and the subsampled measurements *(bullets)*, at 5, 25 and 50 m

mate is plotted together with the full time series of the measurements. The measurements used in the inversion are shown as bullets.

It is first of all evident that both the amplitude and phase of the inverse estimate are in good agreement with the measurements at all times and depths. Note also that the inverse estimate is smooth and does not exactly interpolate the measurements. By a closer examination of the inverse estimate, it is possible to see that the time derivative of the inverse estimate is discontinuous at measurement locations. This is caused by neglecting the time correlation in the model error covariances.

For comparison, a strong constraint inversion was performed and the results are shown in Fig. 5.5. Note that the strong constraint inverse for a linear model is easily solved for without any iterations simply by calculating the representer solution with the model error covariance set to zero.

It is clear from comparisons that the strong constraint solution in the upper part of the ocean is in reasonable phase with the measurements, as determined by the initial conditions, while the amplitudes are not as good as in the weak constraint inverse. The only way the amplitudes can change when the model is assumed to be perfect is by vertical transfer of momentum from the surface. This is seen to work reasonably well near the surface, while in the deeper ocean, there is hardly any effect from the wind stress and the strong constraint inverse solution is also far from the measurements. The solution is actually rather close to a sine curve representing the pure inertial oscillations. The strong constraint results from *Yu and O'Brien* (1992) are similar to ours and also have the same problems with amplitude and phase. These results indicate that model deficiencies, such as neglected physics, should be accounted for through a weak constraint variational formulation to ensure an inverse solution in agreement with the measurements.

5.4 Comments on the representer method

Some important comments should be made regarding the representer method. For details we refer to the monographs by *Bennett* (1992, 2002).

1. As in (3.55) an inner product can be defined for the current time dependent problem, and a reproducing kernel for this inner product becomes the error covariance in time for the first guess state estimate. Thus, the same theory as was used in Chap. 3 can be used again to prove properties of the problem.
2. The representer solution provides the optimal minimizing solution of the linear inverse problem. It was shown by *Bennett* (1992) that by assuming a solution
$$\psi(t) = \psi_{\mathrm{F}}(t) + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}(t) + g(t), \tag{5.98}$$
where $g(t)$ is an arbitrary function orthogonal to the space spanned by the representers, it can be shown that we must have $g(t) \equiv 0$, using a procedure similar to the one presented for the time independent problem in Sect. 3.2.6. This also shows that the solution is searched for in the M-dimensional space spanned by the representers. Thus, we have reduced the infinite dimensional problem defined by the penalty function to an M-dimensional problem.
3. The representer method can only be used to solve linear inverse problems. However, for nonlinear dynamical models, it can still be applied if one can define a convergent sequence of linear iterates of the nonlinear model,

where each linear iterate is solved for using the representer method. As an example consider the equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \cdots . \tag{5.99}$$

If the solution of this equation can be found from the iteration

$$\frac{\partial u^i}{\partial t} + u^{i-1}\frac{\partial u^i}{\partial x} = \cdots , \tag{5.100}$$

then one can also define a convergent sequence of linear inverse problems which can be solved exactly using representer expansions. This approach has been used with many realistic ocean and atmospheric circulation models by Bennett and coworkers and has proved to work well when the nonlinearities are not too strong. It was in fact used for an inversion of a global atmospheric primitive equation model by *Bennett et al.* (1996).

4. From the algorithm as described above, it may seem like one has to solve for a representer corresponding to each individual measurement, at the cost of two model integrations for each. However, it turns out that it is possible to solve the system (5.60) without first constructing the matrix $\mathcal{M}^{\mathrm{T}}[\boldsymbol{r}]$. This is possible since only the product of $\mathcal{M}^{\mathrm{T}}[\boldsymbol{r}]$ with an arbitrary vector $\boldsymbol{v}$ is required if an iterative solver such as the conjugate gradient method is used. This product can be evaluated by two model integrations by using a clever algorithm which is described by *Egbert et al.* (1994) and *Bennett* (2002). This is easily seen if we multiply the transposes of (5.61), (5.62), (5.63) and (5.64) with $\boldsymbol{v}$ to get

$$\frac{\partial \boldsymbol{r}^{\mathrm{T}}\boldsymbol{v}}{\partial t} + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{v} = C_{qq} \bullet \big(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{v}\big), \tag{5.101}$$

$$\big(\boldsymbol{r}^{\mathrm{T}}\boldsymbol{v}\big)(0) = C_{aa}\big(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{v}\big)(0), \tag{5.102}$$

$$\frac{\partial\big(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{v}\big)}{\partial t} + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{v} = -\mathcal{M}^{\mathrm{T}}\big[\delta\big]\boldsymbol{v}, \tag{5.103}$$

$$\big(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{v}\big)(t_k) = 0. \tag{5.104}$$

Here we note that $\boldsymbol{s}^{\mathrm{T}}\boldsymbol{v} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{s}$ is in this case a scalar function of time, just like the original model state. One backward integration of the final value problem defined by (5.103) and (5.104) results in the solution $(\boldsymbol{s}^{\mathrm{T}}\boldsymbol{v})$, which is then used to solve the initial value problem, (5.101) and (5.102), for the function $(\boldsymbol{r}^{\mathrm{T}}\boldsymbol{v})$. Since the measurement operator is linear, we then get

$$\mathcal{M}\big[\big(\boldsymbol{r}^{\mathrm{T}}\boldsymbol{v}\big)\big] = \mathcal{M}^{\mathrm{T}}\big[\boldsymbol{r}\big]\boldsymbol{v}, \tag{5.105}$$

which is needed in the iterative solver.

Thus, for each linear iterate, the representer solution can be found by a number of model integrations equal to two times the number of conjugate gradient iterations to find $\boldsymbol{b}$, plus two integrations to find the final

solution. The conjugate gradient iterations converge quickly if a good preconditioner is used and often a few selected representers are computed and measured first to construct the preconditioner (*Bennett*, 2002).

5. Finally, the convolutions appearing in the Euler–Lagrange equations can also be computed very efficiently if specific covariance functions are used. In particular it is explained in *Bennett* (2002) how one can compute the convolutions by solving simple differential equations using an approach developed by *Derber and Rosati* (1989) and *Egbert et al.* (1994).
6. Note that the equation for $\boldsymbol{b}$, (5.60), is similar to the one solved in the analysis scheme in the standard Kalman filter. Furthermore, in the Kalman filter the representers or influence functions are defined as the measurements of the error covariance matrix at a particular time, while in the representer method the representers are functions of space and time. It can be shown that the representers correspond to the measurements of the space-time error covariance of the first guess solution. Thus, there are similarities between the analysis step in the Kalman filter and the representer method.

To summarize, the representer method is an extremely efficient methodology for solving linear inverse problems and it is also applicable to many nonlinear dynamical models. Note that the method requires knowledge of the dynamical equations and numerical code to derive the adjoint equations. Further, the actual derivation of the adjoint model and its implementation may be cumbersome for some models. This is contrary to the ensemble methods that will be discussed later. They only require the dynamical model as a black box for integrating model states forward in time.

6

Nonlinear variational inverse problems

This chapter considers highly nonlinear variational inverse problems and their properties. More general inverse formulations for nonlinear dynamical models will be treated extensively in the following chapters, but an introduction is in place here. The focus will be on some highly nonlinear problems which cannot easily be solved using the representer method. Examples are given were instead, so-called direct minimization methods are used.

6.1 Extension to nonlinear dynamics

It was pointed out in the previous chapter that, rather than solving one nonlinear inverse problem, one may define a convergent sequence of linear iterates for the nonlinear model equation, and then solve a linear inverse problem for each iterate using the representer method.

On the other hand, it is also possible to define a variational inverse problem for a nonlinear model. As an example, when starting from the system (5.21–5.23) but with the right-hand-side of (5.21) replaced by a nonlinear function, $G(\psi)$, we obtain Euler–Lagrange equations on the form

$$\mathrm{d}\psi t - G(\psi) = \int_0^T C_{qq}(t,t_1)\lambda(t_1)\,dt_1, \tag{6.1}$$

$$\psi(0) = \Psi_0 + C_{aa}\lambda(0), \tag{6.2}$$

$$\mathrm{d}\lambda t + G^*(\psi)\lambda = -\boldsymbol{\mathcal{M}}_{(2)}^{\mathrm{T}}\big[\delta(t-t_2)\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \boldsymbol{\mathcal{M}}\big[\psi\big]\Big), \tag{6.3}$$

$$\lambda(T) = 0, \tag{6.4}$$

where $G^*(\psi)$ is the transpose of the tangent linear operator of $G(\psi)$ evaluated at ψ. Thus, like in the EKF we need to use linearized model operators, but this time for the backward or adjoint equation. We can expect that this may lead to similar problems as was found using the EKF.

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_6,

Note that, for nonlinear dynamics the adjoint operator (or adjoint equation) does not exist, since the penalty function no longer defines an inner product for a Hilbert space. This is resolved by instead using the adjoint of the tangent linear operator.

In the following we will consider a variational inverse problem for the highly nonlinear and chaotic Lorenz equations and use this to illustrate typical problems that may show up when working with nonlinear dynamics.

6.1.1 Generalized inverse for the Lorenz equations

Several publications have examined assimilation methods with chaotic and unstable dynamics. In particular, the Lorenz model (*Lorenz*, 1963) has been examined with many different assimilation methods. Results have been used to suggest properties and possibilities of the methods for applications with oceanic and atmospheric models which may also be strongly nonlinear and chaotic.

The Lorenz model is a system of three first order coupled and nonlinear differential equations for the variables x, y and z,

$$\mathrm{d}xt = \sigma(y - x) + q_x, \tag{6.5}$$
$$\mathrm{d}yt = \rho x - y - xz + q_y, \tag{6.6}$$
$$\mathrm{d}zt = xy - \beta z + q_z, \tag{6.7}$$

with initial conditions

$$x(0) = x_0 + a_x, \tag{6.8}$$
$$y(0) = y_0 + a_y, \tag{6.9}$$
$$z(0) = z_0 + a_z. \tag{6.10}$$

Here $x(t)$, $y(t)$ and $z(t)$ are the dependent variables and we have chosen the following commonly used values for the parameters in the equations: $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. We have also defined the error terms $\boldsymbol{q}(t)^{\mathrm{T}} = (q_x(t), q_y(t), q_z(t))$ and $\boldsymbol{a}^{\mathrm{T}} = (a_x, a_y, a_z)$ which have error statistics described by the 3×3 error covariance matrices $\boldsymbol{C}_{qq}(t_1, t_2)$ and $\boldsymbol{C}_{aa}$. The system leads to chaotic solutions where small perturbations of initial conditions lead to a completely different solution after a certain time integration.

Measurements of the solution are represented through the measurement equation

$$\mathcal{M}[\boldsymbol{x}] = \boldsymbol{d} + \boldsymbol{\epsilon}. \tag{6.11}$$

Further, by allowing the dynamical model equations (6.5–6.7) to contain errors, we obtain the standard weak constraint variational formulation,

$$\begin{aligned} \mathcal{J}[x, y, z] = &\iint_0^T \boldsymbol{q}(t_1)^{\mathrm{T}}\, \boldsymbol{W}_{qq}(t_1, t_2)\, \boldsymbol{q}(t_2)\, dt_1\, dt_2 \\ &+ \boldsymbol{a}^{\mathrm{T}}\, \boldsymbol{W}_{aa}\, \boldsymbol{a} + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{\epsilon}. \end{aligned} \tag{6.12}$$

The weight matrix, $\boldsymbol{W}_{qq}(t_1, t_2) \in \Re^{3\times3}$, is defined as the inverse of the model error covariance matrix, $\boldsymbol{C}_{qq}(t_2, t_3) \in \Re^{3\times3}$, from

$$\int_0^T \boldsymbol{W}_{qq}(t_1, t_2)\boldsymbol{C}_{qq}(t_2, t_3)dt_2 = \delta(t_1 - t_3)\boldsymbol{I}, \tag{6.13}$$

and we have the weight matrices, $\boldsymbol{W}_{aa} = \boldsymbol{C}_{aa}^{-1} \in \Re^{3\times3}$ and $\boldsymbol{W}_{\epsilon\epsilon} = \boldsymbol{C}_{\epsilon\epsilon}^{-1} \in \Re^{M\times M}$.

6.1.2 Strong constraint assumption

The strong constraint assumption leads to the adjoint method which has proven to be efficient for linear dynamics, given that the strong constraint assumption is valid.

The strong constraint assumption, solved by the adjoint method, has been extensively used in the atmosphere and ocean communities. Particular effort has been invested in developing the adjoint method for use in weather forecasting systems, where it is named 4DVAR (4–dimensional variational method). 4DVAR implementations are today in operational or preoperational use at atmospheric weather forecasting centers, but common for these is that they still only works well for rather short assimilation time intervals of one day or less. The causes for this may be connected to the tangent linear approximation but also to the chaotic nature of the dynamical model.

The strong constraint inverse problem for the Lorenz equations is defined by assuming that the model is perfect, $\boldsymbol{q}(t) \equiv 0$, and only the initial conditions contain errors. A number of papers have examined the adjoint method with the Lorenz model, see e.g. *Gauthier* (1992), *Stensrud and Bao* (1992), *Miller et al.* (1994), *Pires et al.* (1996). In these works it was found that there is a strong sensitivity of the penalty function with respect to the initial conditions. In particular there is a problem when the assimilation time interval exceeds a few times the predictability time of the model.

Miller et al. (1994) found that the penalty function changed from a nearly quadratic shape around the global minimum, for short assimilation time intervals, to a shape similar to a white noise process when the assimilation time interval was extended.

This is illustrated in Fig. 6.1 which plots values of the cost function with respect to variation in $x(0)$ while $y(0) = y_0$ and $z(0) = z_0$ are kept constant at their prior estimates. It is further assumed that all components of the solution $\boldsymbol{x}(t)$ are observed at regular time intervals $t_j = j\,\Delta t_{\text{obs}}$, for $j = 1, \ldots, m$, with $\Delta t_{\text{obs}} = 1$. We can then define the measurement equation for each measurement time t_j, as

$$\mathcal{M}_j[\boldsymbol{x}] = \boldsymbol{d}_j + \boldsymbol{\epsilon}_j, \tag{6.14}$$

where $\boldsymbol{\epsilon}_j$ represents the random errors in the measurements.

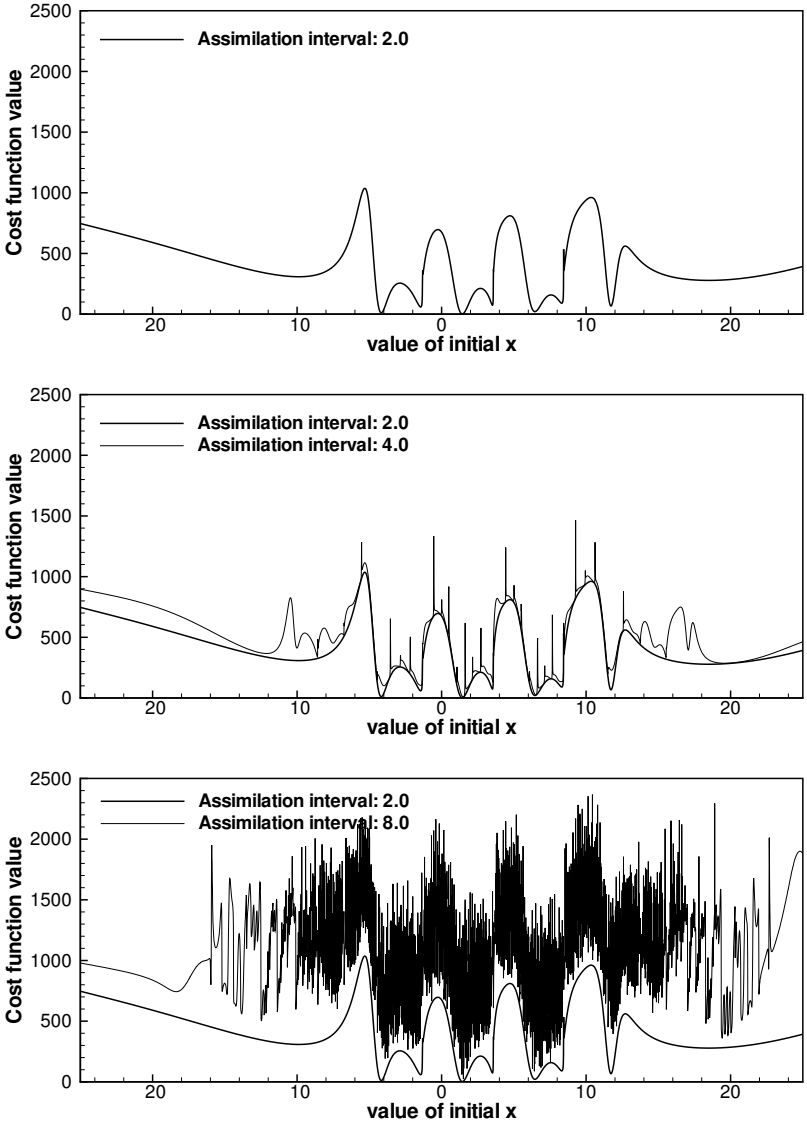

Fig. 6.1. Strong constraint penalty function for the Lorenz model as a function of the initial x-value, keeping y and z constant, when using data in the intervals $t \in [0, 2]$ *(upper plot)*, $t \in [0, 4]$ *(middle plot)*, and $t \in [0, 8]$ *(lower plot)*

The value of the penalty function can be evaluated from

$$\begin{aligned}\mathcal{J}_J[\boldsymbol{x}(0)] &= \big(\boldsymbol{x}(0)-\boldsymbol{x}_0\big)^{\mathrm{T}}\boldsymbol{W}_{aa}\big(\boldsymbol{x}(0)-\boldsymbol{x}_0\big)\\ &\quad+\sum_{j=1}^{J}\Big(\boldsymbol{d}_j-\mathcal{M}_j\big[\boldsymbol{x}\big]\Big)^{\mathrm{T}}\boldsymbol{W}_{\epsilon\epsilon}(j)\Big(\boldsymbol{d}_j-\mathcal{M}_j\big[\boldsymbol{x}\big]\Big),\end{aligned} \tag{6.15}$$

where the subscript J, defines the length of the assimilation time interval and indicates that measurements up to the J'th measurement time are included. The weights $\boldsymbol{W}_{aa}$ and $\boldsymbol{W}_{\epsilon\epsilon}(j)$ are three by three matrices and have the same interpretation as in the previous sections.

The upper plot of Fig. 6.1 is for a very short assimilation time interval of $t \in [0, 2]$, i.e. only twice the characteristic time scale of the model dynamics. It is clear that even for this short time-interval there are local minima in the cost function and a very good prior estimate of the initial state is needed for a gradient based method to converge to the global minimum near $x(0) = 1.5$. In the middle plot the assimilation interval is extended to $t \in [0, 4]$ and we see that even though the basic shape is the same there now appear some additional spikes and local minima in the cost function. When the assimilation time interval is extended to $t \in [0, 8]$ in the lower plot, the shape of the cost function appears nearly as a white noise process. It is obvious that these cost functions cannot be minimized using traditional gradient based methods, and obviously, the strong constraint problem for the Lorenz equations becomes practically impossible to solve for long assimilation time intervals, independent of the method used.

It should at this time be noted that this is mainly a result of the formulation of the problem, i.e. the assumption that the model is an exact representation of unstable and chaotic dynamics. It is not unlikely that similar problems can occur in models of the ocean and atmosphere which resolves the chaotic mesoscale circulation, and this may be one of the reasons why 4DVAR appears to be limited to short assimilation time intervals in these applications.

The approach for resolving this problem in the atmospheric community has been to solve a sequence of strong constraint inverse problems, of the form (6.15), defined for separate subintervals in time. To illustrate this, assume that we have divided the assimilation time interval into one-day sub-intervals, and we define a strong constraint inverse problem for each one-day time interval on the form (6.15). Thus:

1. We start by solving the first sub-problem for day one which results in an estimate for the initial conditions at day one.
2. Integration of the model from this initial condition provides the strong constraint inverse solution for day one.
3. We then use the inverse solution from the end of day one to specify the prior estimate of the initial conditions for day two.
4. The problem now is that, for day two, one cannot easily compute an estimate of a new updated prior error statistics $\boldsymbol{W}_{aa}$, for the initial con-

ditions, that accounts for the new information introduced in the previous inverse calculation. Thus, the original prior $\boldsymbol{W}_{aa}$ is used repeatedly for each sub-interval.

Using this procedure, there is no proper time evolution of the error covariances, thus a different problem than the originally posed strong constraint problem is solved. Estimation of the proper error covariance matrix would require the computation of the inverse of the Hessian of the penalty function, which equals the error covariance matrix for the estimated initial conditions, followed by the evolution of this error covariance matrix through the assimilation interval using an approximate error covariance equation like in the EKF.

6.1.3 Solution of the weak constraint problem

We already saw that if the dynamical model is not too nonlinear, a convergent sequence of linear iterates may be defined, and each iterate can be optimally solved using the representer method. For dynamical models with stronger nonlinearities the sequence of linear iterates may not converge and alternative methods need to be used.

Another class of methods for minimizing (6.12) is named substitution methods. These are methods that guess candidates for the minimizing solution and then evaluate the value of the penalty function. Dependent of the algorithm used the new candidate may be accepted with a specified probability if it results in a lower value for the penalty function.

A discrete version of the penalty function is now needed and we represent the model variables $x(t)$, $y(t)$, and $z(t)$ on a numerical grid in time. The variables are stored in the state vectors $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, all belonging to $\Re^n$, i.e. we have the vector $\boldsymbol{x}^{\mathrm{T}} = (x_1, x_2, \ldots, x_n)$, and similarly for $\boldsymbol{y}$ and $\boldsymbol{z}$, where n is the number of grid points in time. The discrete analog to (6.12) then becomes

$$\mathcal{J}[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}] = \Delta t^2 \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{q}(i)^{\mathrm{T}} \boldsymbol{W}_{qq}(i,j)\, \boldsymbol{q}(j) + \boldsymbol{a}^{\mathrm{T}} \boldsymbol{W}_{aa}\, \boldsymbol{a} + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{\epsilon}, \quad (6.16)$$

where $\boldsymbol{q}(i)^{\mathrm{T}} = (q_x(t_i), q_y(t_i), q_z(t_i))$. Furthermore, there will be no integration of the model equations required using the substitution methods and simple numerical discretizations based on second order centered differences for the time derivatives can be used, i.e.

$$\begin{aligned} \frac{x_{i+1} - x_{i-1}}{2\Delta t} &= \sigma(y_i - x_i) + q_x(t_i), \\ \frac{y_{i+1} - y_{i-1}}{2\Delta t} &= \rho x_i - y_i - x_i z_i + q_y(t_i), \\ \frac{z_{i+1} - z_{i-1}}{2\Delta t} &= x_i y_i - \beta z_i + q_z(t_i), \end{aligned} \quad (6.17)$$

where $i = 2, \ldots, n-1$ is the time-step index, with n the total number of time steps.

Note that the evaluation of the double sum in (6.16) is costly. Here, an alternative method like the one used for the convolutions in the representer method could be used.

An even more efficient approach was used by *Evensen and Fario* (1997). It is assumed that the model weight can be written as

$$\boldsymbol{W}_{qq}(t_1, t_2) = \boldsymbol{W}_{qq}\delta(t_1 - t_2), \tag{6.18}$$

where $\boldsymbol{W}_{qq}$ is a constant 3×3 matrix. This eliminates one of the summations in the model term in (6.16) and allows for more efficient computational algorithms. However, the correlation in time of the model errors has a time regularizing effect on the inverse estimate which has now been lost.

To ensure a smooth solution in time the regularization is instead accounted for by a smoothing term

$$\mathcal{J}_S[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}] = \Delta t \sum_{i=1}^{n} \boldsymbol{\eta}_i^{\mathrm{T}} \boldsymbol{W}_{\eta\eta} \boldsymbol{\eta}_i, \tag{6.19}$$

where $\boldsymbol{\eta}_i^{\mathrm{T}} = \big(\eta_x(t_i), \eta_y(t_i), \eta_z(t_i)\big)$, with

$$\eta_x(t_i) = \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2}, \tag{6.20}$$

and $\boldsymbol{W}_{\eta\eta}$ is a weight matrix determining the relative impact of the smoothing term.

It would have been more consistent to actually smooth the model errors instead of the inverse estimate, since it can be shown that such a smoothing constraint, used together with the penalty term for the model errors, would define a norm. Moreover, there is a unique correspondence between such a smoothing norm and a covariance matrix, as shown by *McIntosh* (1990). On the other hand, the smoothing term as included here, will improve the conditioning of the method since only smooth functions are searched for.

The penalty function now becomes

$$\begin{aligned} \mathcal{J}[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}] = \Delta t \sum_{i=1}^{n} & \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{W}_{qq} \boldsymbol{q}_i + \boldsymbol{a}^{\mathrm{T}} \boldsymbol{W}_{aa} \boldsymbol{a} + \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{w} \boldsymbol{\epsilon} \\ & + \Delta t \sum_{i=1}^{n} \boldsymbol{\eta}_i^{\mathrm{T}} \boldsymbol{W}_{\eta\eta} \boldsymbol{\eta}_i. \end{aligned} \tag{6.21}$$

For $\boldsymbol{q}_1$, $\boldsymbol{q}_n$, $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_n$ we use second order one-sided difference formulas.

6.1.4 Minimization by the gradient descent method

A very simple approach for minimizing the penalty function (6.21) is to use a gradient descent algorithm as was done by *Evensen* (1997), *Evensen and Fario*

(1997). The gradient $\nabla_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]$, with respect to the full state vector in time $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, is easily derived. When the gradient is known it can be used in a descent algorithm to search for the minimizing solution. Thus, for the Lorenz model we solve the iteration

$$\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix}^{i+1} = \begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix}^{i} - \gamma \begin{pmatrix}\nabla_{\boldsymbol{x}}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]\\ \nabla_{\boldsymbol{y}}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]\\ \nabla_{\boldsymbol{z}}\mathcal{J}[\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}]\end{pmatrix}^{i}. \tag{6.22}$$

with γ being a step length. Given a first guess estimate, the gradient of the cost function is evaluated and a new state estimate can be searched for in the direction of the gradient.

The required storage for the gradient descent method is of order the size of the state vector in space and time, which is the same as for the adjoint and representer methods.

Note that, using a gradient descent method there is no need for any model integrations. This is contrary to the representer and adjoint methods which integrate both the forward model and the adjoint model, and to the Kalman filter where the forward model is needed.

As long as the penalty function does not contain any local minima, the gradient method will eventually converge to the minimizing solution. However, the obvious drawback is that the dimension of the problem becomes huge for high dimensional problems, i.e. the number of dependent variables times the grid points in time and space. For the Lorenz model this becomes $3n$. This is normally much larger than the number of measurements which defines the dimension of the problem as solved by the representer method. Thus, a proper conditioning may be needed to ensure that high dimensional problems converge in an acceptable number of iterations.

6.1.5 Minimization by genetic algorithms

With nonlinear dynamics the penalty function is clearly not convex in general due to the first term in (6.21) containing the model residuals. However, both the measurement penalty term and the smoothing norm will give a quadratic contribution to the penalty function and if the weights, $\boldsymbol{W}_{\epsilon\epsilon}$ and $\boldsymbol{W}_{\eta\eta}$, are large enough compared to the dynamical weight $\boldsymbol{W}_{qq}$, one can expect a nearly quadratic penalty function. On the contrary, if the model residuals are the dominating terms in the penalty function, clearly a pure descent algorithm may get trapped in local minima and the solution found may depend on the first guess in the iteration.

A special class of substitution methods contains the so-called genetic algorithms. These are typically statistical methods which guess new candidates for the minimizing solution at random or using some wise candidate selection algorithm. Then an acceptance algorithm is used to decide whether the new candidate is accepted or not. The acceptance algorithm is dependent on the

value of the penalty function but also has a random component which allows it to escape local minima.

Statistical versions of the genetic methods exploit the fact that the minimizing solution can be interpreted as the maximum likelihood estimate of a probability density function,

$$f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \propto \exp\Big(-\mathcal{J}\big[\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\big]\Big). \tag{6.23}$$

Moments of $f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ could be estimated using standard numerical integration based on Monte Carlo methods using points selected at random from some distribution. However, this would be extremely inefficient due to the huge state space associated with many high dimensional models, such as models of the ocean and atmosphere.

Metropolis algorithm

Instead a method by *Metropolis et al.* (1953) is useful, and we now illustrate it for the variable $\boldsymbol{\psi}^{\mathrm{T}} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. The algorithm samples a pdf by performing a random walk through the space of interest. At each sample position $\boldsymbol{\psi}$, a perturbation is added to generate a new candidate $\boldsymbol{\psi}_1$, and this candidate is accepted according to a probability

$$p = \min\left(1, \frac{f(\boldsymbol{\psi}_1)}{f(\boldsymbol{\psi})}\right). \tag{6.24}$$

The mechanism for accepting the candidate with probability p, is implemented by drawing a random number ξ, from the uniform distribution on the interval $[0, 1]$ and then accepting $\boldsymbol{\psi}_1$ if $\xi \leq p$. The conditional uphill climb, based on the value of p and ξ, is due to *Metropolis et al.* (1953) and is named the Metropolis algorithm. They also gave a proof that the method was ergodic, i.e. any state can be reached from any other, and that the trials would sample the probability distribution $f(\boldsymbol{\psi})$. Clearly, in a high dimensional space with strongly nonlinear dynamics, the random trials may be too random and most of the time lead to candidates $\boldsymbol{\psi}_1$, with very low probabilities, which are only occasionally accepted. Thus, the algorithm becomes very inefficient.

Hybrid Monte Carlo algorithm

In *Bennett and Chua* (1994) an alternative to a random walk, which provided a significantly faster convergence, was used when solving for the inverse of a nonlinear open ocean shallow water model. The algorithm which is due to *Duane et al.* (1987) ensures that candidates with acceptable probabilities are constructed. It is based on constructing the Hamiltonian

$$\mathcal{H}[\boldsymbol{\psi}, \boldsymbol{\pi}] = \mathcal{J}[\boldsymbol{\psi}] + \frac{1}{2}\boldsymbol{\pi}^{\mathrm{T}}\boldsymbol{\pi}, \tag{6.25}$$

and then deriving the canonical equations of motion in $(\boldsymbol{\psi}, \boldsymbol{\pi})$ phase space, with respect to a pseudo time variable τ,

$$\frac{\partial \psi_i}{\partial \tau} = \frac{\partial \mathcal{H}}{\partial \pi_i} = \pi_i, \tag{6.26}$$

$$\frac{\partial \pi_i}{\partial \tau} = -\frac{\partial \mathcal{H}}{\partial \psi_i} = -\frac{\partial \mathcal{J}}{\partial \psi_i}. \tag{6.27}$$

This system is integrated for a pseudo time interval, $\tau \in [0, \tau_1]$, using the previously accepted value of $\boldsymbol{\psi}$ and a random guess for $\boldsymbol{\pi}(0)$ as initial conditions. The Metropolis algorithm can then be used for the new guess $\boldsymbol{\psi}(\tau_1)$. In *Duane et al.* (1987), it was proved that this algorithm also preserved detailed balance, i.e.

$$f(\boldsymbol{\psi}_1, \boldsymbol{\psi}_2) = f(\boldsymbol{\psi}_2|\boldsymbol{\psi}_1) f(\boldsymbol{\psi}_1) = f(\boldsymbol{\psi}_1|\boldsymbol{\psi}_2) f(\boldsymbol{\psi}_2), \tag{6.28}$$

which is needed for showing that a long sequence of random trials will converge towards the distribution (6.23).

The interpretation of the method is clear. In the Hamiltonian (6.25), the penalty function defines a potential energy while a kinetic energy is represented by the last term. The canonical equations describe motion along lines of constant total energy. Thus, with a finite and random initial momentum, the integration of the canonical equation over a pseudo time interval will result in a new candidate with a different distribution of potential and kinetic energy. Unless the initial momentum is very large this will always result in a candidate which has a reasonable probability. If the initial momentum is zero, it will result in a candidate with less potential energy and higher probability. If the initial candidate is a local minimum, the random initial momentum may provide enough energy to escape the local minimum.

Note that, after a minimum of the variational problem has been found, the posterior error statistics can be estimated by collecting samples of nearby states. Thus, by using the hybrid Monte Carlo method to generate a Markov chain that samples the probability function, a statistical variance estimate can be generated. This method may be used to generate error estimates independently of the minimization technique used to solve the weak constraint problem. Hence, it could also be used in combination with the representer method which does not easily provide error estimates.

Simulated annealing

When working with a penalty function which has many local minima, the so-called simulated annealing technique may be used to improve the convergence to the stationary distribution, based on the method's capability of escaping local minima.

The simulated annealing method (see *Kirkpatrick et al.*, 1983, *Azencott*, 1992) is extremely simple in its basic formulation and can be illustrated using an example where a penalty function $\mathcal{J}[\boldsymbol{\psi}]$, which may be nonlinear and discontinuous, is to be minimized with respect to the variable $\boldsymbol{\psi}$:

$\boldsymbol{\psi}$ first guess
for $i = 1 : \ldots$
  $\boldsymbol{\psi}_1 = \boldsymbol{\psi} + \Delta\boldsymbol{\psi}$
  if $(\mathcal{J}[\boldsymbol{\psi}_1] < \mathcal{J}[\boldsymbol{\psi}])$ **then**
    $\boldsymbol{\psi} = \boldsymbol{\psi}_1$
  else
    $\xi \in [0, 1]$ random number
    $p = \exp\Big(\big(\mathcal{J}[\boldsymbol{\psi}] - \mathcal{J}[\boldsymbol{\psi}_1]\big)/\theta\Big) \in [0, 1]$
    if $p > \xi$ **then** $\boldsymbol{\psi} = \boldsymbol{\psi}_1$
  end
  $\theta = f(\theta, i, \mathcal{J}_{\min})$
end

Here $\Delta\boldsymbol{\psi}$ might be a normal distributed random vector, but it is more efficient to simulate it using the hybrid Monte Carlo technique just described.

The temperature scheme $\theta = \theta(\theta, i, \mathcal{J}_{\min})$ is used to cool or relax the system and is normally a decreasing function of iteration counter i.

The trials will then converge towards a distribution

$$f(\boldsymbol{\psi}) \propto \exp\big(-\mathcal{J}[\boldsymbol{\psi}]/\theta\big), \tag{6.29}$$

By slowly decreasing the value of θ the distribution will approach the delta function at the minimizing value of $\boldsymbol{\psi}$. The clue is then to choose a temperature scheme where one avoids getting trapped in local minima for too many iterations, or where too many uphill climbs are accepted. In *Bohachevsky et al.* (1986), it was suggested that the temperature should be chosen so that $p \in [0.5, 0.9]$. Here also a generalized algorithm was proposed where p was calculated according to $p = \exp\big(\beta(\mathcal{J}[\boldsymbol{\psi}] - \mathcal{J}[\boldsymbol{\psi}_1])/(\mathcal{J}[\boldsymbol{\psi}] - \mathcal{J}_{\min})\big)$, where β is approximately 3.5 and $\mathcal{J}_{\min}$ is an estimate of the normally unknown minimum value of the penalty function. Then the probability of accepting a detrimental step tend to zero as the random walk approaches the global minimum. If a value of the cost function is found which is less than $\mathcal{J}_{\min}$ this value will replace $\mathcal{J}_{\min}$.

Simulated annealing was previously used by *Barth and Wunsch* (1990) to optimize an oceanographic data collection scheme. The use of the hybrid Monte Carlo method in combination with simulated annealing has been extensively discussed by *Neal* (1992, 1993) in the context of Bayesian training of back-propagation networks. The method was also used to invert an inverse for a primitive equation model on a domain with ill-posed open boundaries by *Bennett and Chua* (1994). An application with the Lorenz equations was discussed by *Evensen and Fario* (1997) and will be illustrated below.

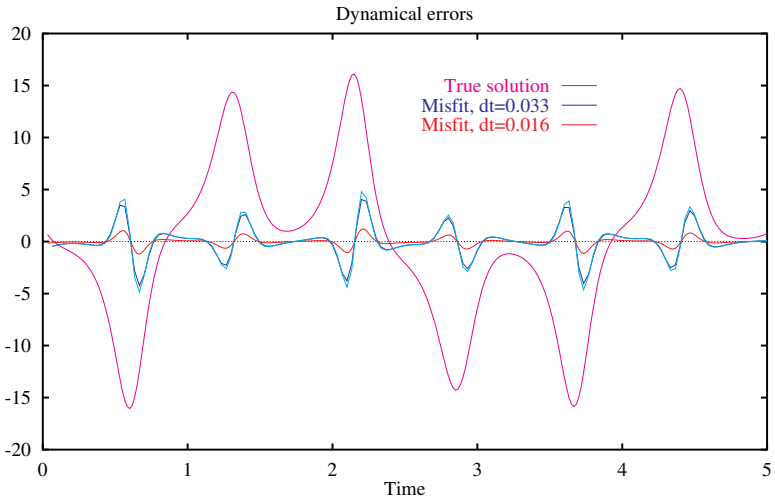

Fig. 6.2. Errors in the difference approximation used for the time derivative, plotted together with the reference solution used in the calculation of the errors. The two similar curves for $\Delta t = 0.033$ are comparing the actual calculated misfits and the lowest-order error term in the discrete time derivative. Reproduced from *Evensen and Fario* (1997)

6.2 Example with the Lorenz equations

We will now present an example where the gradient descent and the simulated annealing algorithm are used with the Lorenz equations. This example is similar to the one discussed by *Evensen and Fario* (1997).

6.2.1 Estimating the model error covariance

In an identical twin experiment it is possible to generate accurate estimates of the model error covariance. First the reference or true solution is computed using a highly accurate ordinary differential equation solver. Then the only significant contribution to the dynamical error term $\boldsymbol{q}_n$, is the error introduced in the approximate time discretization (6.17). These misfits can be evaluated and used to determine the weight matrices $\boldsymbol{W}_{qq}$ and $\boldsymbol{W}_{\eta\eta}$, which are needed in the inverse calculation.

An alternative is to evaluate the first order error term in the centered first derivative approximation used in the discrete model equations (6.17), i.e. we write for the time derivative of $x(t)$,

$$\frac{\partial x}{\partial t} = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} + \frac{1}{6}\frac{\partial^3 x}{\partial t^3}\Delta t^2 + \ldots, \tag{6.30}$$

and evaluate the error term given the true solution.

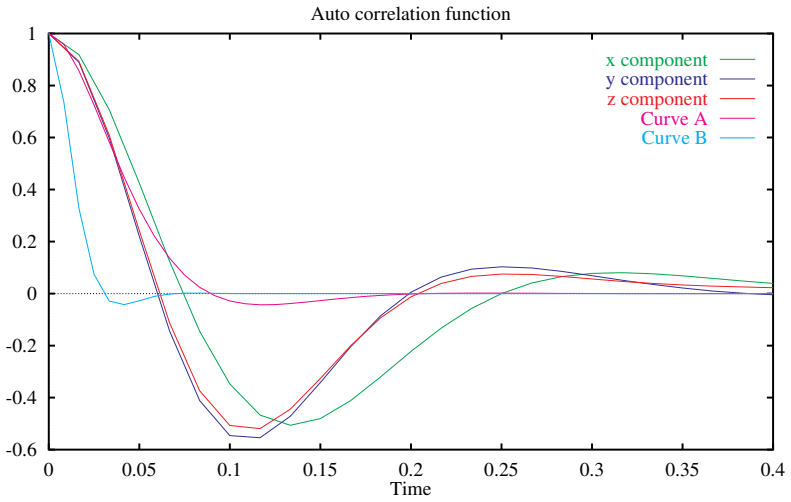

Fig. 6.3. Auto-correlation functions calculated for the computed dynamical misfits for the x, y, and z component of the solution, and two auto-correlation functions corresponding to the smoothing norm with $\gamma = 0.0008$ *(curve A)* and $\gamma = 0.00001$ *(curve B)*. Reproduced from *Evensen and Fario* (1997)

In Fig. 6.2 the dynamical misfits are plotted using two different time steps. Clearly, the errors increase with the length of the time step and the maximum errors are located at the peaks of the reference solution. The two almost identical curves for $\Delta t = 0.033$ are generated using the two different approaches just described.

The error covariance matrix $\boldsymbol{C}_{qq}$ can be estimated from a long time series of these errors, and is of course dependent on the time step used. In the experiments presented here we use a time step of $\Delta t = 0.01667$ and the corresponding error covariance matrix then becomes

$$\boldsymbol{C}_{qq} = \begin{bmatrix} 0.1491 & 0.1505 & 0.0007 \\ 0.1505 & 0.9048 & 0.0014 \\ 0.0007 & 0.0014 & 0.9180 \end{bmatrix}, \tag{6.31}$$

where the integration has been performed for a long time interval $t \in [0, 1667]$, i.e. 100 000 time steps. The inverse of this matrix is used for $\boldsymbol{W}_{qq}$ in the penalty function (6.21).

6.2.2 Time correlation of the model error covariance

The errors are also clearly correlated in time. In Fig. 6.3 the auto-correlation functions for the x, y, and z components of the dynamical errors are plotted. Since it is inconvenient to use a full space and time covariance matrix, we

introduce the smoothing term (6.19), which act as a regularization term on the minimizing solution.

It can be shown that a smoothing norm of the type

$$||\psi|| = \int_0^T \psi^2 + \gamma \psi_{tt}^2 \, dt \tag{6.32}$$

has a Fourier transform equal to

$$\hat{\psi} = \left(1 + \gamma \omega^4\right)^{-1}. \tag{6.33}$$

The limiting behaviour for increasing frequency ω is then proportional to $(\gamma\omega^4)^{-1}$; thus high frequencies are penalized most strongly in the smoothing norm. The ψ^2 term is added here, as a first guess penalty, for illustrational purposes. Without this term, the limiting behaviour for $\omega \to 0$ would be singular and the corresponding auto-correlation function would become very flat. In the actual inverse formulation, the dynamical and initial residual will provide the first guess penalty, ensuring a well-behaved limiting behaviour when $f \to 0$.

An inverse Fourier transform of the spectrum (6.33) gives an auto-correlation function which is shown in Fig. 6.3 for two values of γ, i.e. $\gamma = 0.0008$ for *curve A* and $\gamma = 0.00001$ for *curve B*. For $\gamma = 0.0008$ the auto-correlation function has a similar half width to the auto-correlation functions of the dynamical errors. However, it turned out that for this value of γ the inverse estimate became too smooth, i.e. the peaks in the solutions were to low compared to the reference solution. We decided to use $\gamma = 0.00001$ which gave an inverse estimate more in agreement with the reference solution. Based on the time series of dynamical misfits in Fig. 6.2, it is also clear that the errors are rather smooth for most of the time while they have sudden changes close to the peaks of the reference solution. The computed auto-correlation function will describe an "average" smoothness of the dynamical misfits which is too smooth near the peaks in the reference solution. This can then justify the use of the smaller smoothing weight $\gamma = 0.00001$.

The error covariance matrix $\boldsymbol{C}_{aa}$ for the errors in the initial conditions, and the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$, are both assumed to be diagonal and with the same error variance equal to 0.5. The model error covariance matrix is given by (6.31) and the smoothing weight matrix is chosen to be diagonal and given by $\boldsymbol{W}_{\eta\eta} = \gamma \boldsymbol{I}$ with $\gamma = 0.00001$.

6.2.3 Inversion experiments

For all the cases to be discussed the initial condition for the reference case is given by $(x_0, y_0, z_0) = (1.508870, -1.531271, 25.46091)$ and the observations and first guess initial conditions are simulated by adding normal distributed noise, with zero mean and variance equal to 0.5, to the reference solution.

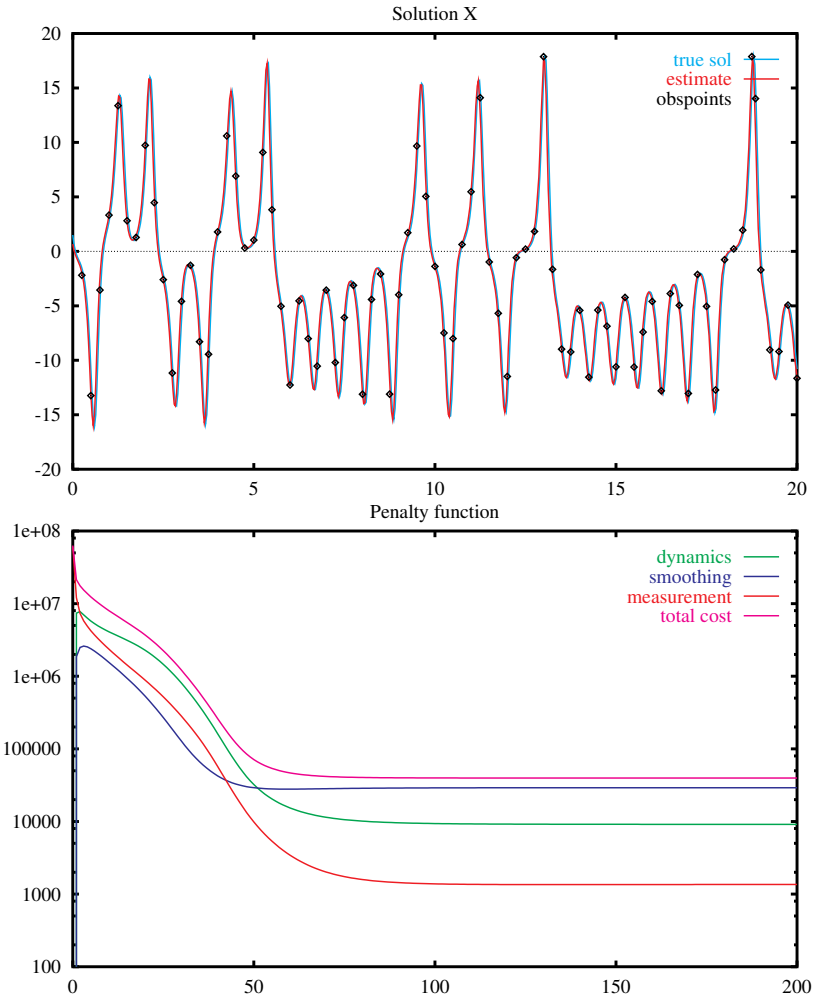

Fig. 6.4. Case A: The inverse estimate for x *(top)* and the terms in the penalty function *(bottom)*. The estimated solution is given by the solid line. The dashed line is the true reference solution, and the diamonds show the simulated observations. The same line types will be used also in the following figures. Reproduced from *Evensen and Fario* (1997)

These are lower values than the variances equal to 2.0, used in *Miller et al.* (1994) and *Evensen and Fario* (1997).

The first guess used in the gradient descent method was initially chosen as the mean of the reference solution, i.e. about $(0, 0, 23)$. However, there seems to be a possibility for a local minima close to the zero solution where both the dynamical penalty term and the smoothing penalty vanish. It is therefore

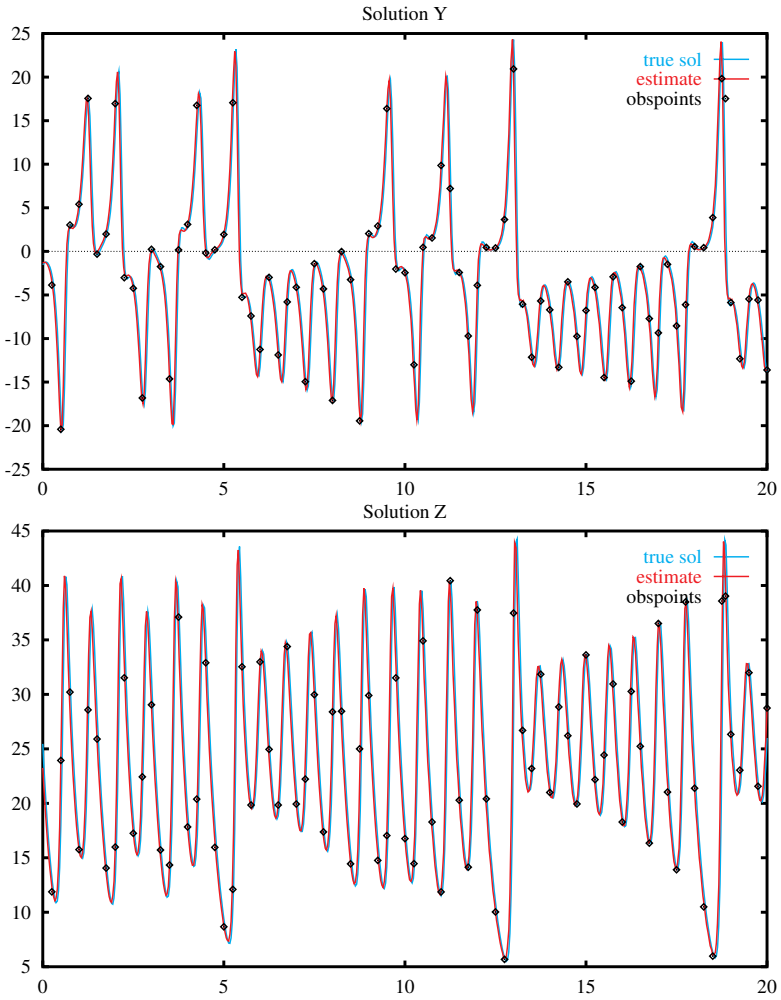

Fig. 6.5. Case A: The inverse estimate for y (top) and z (bottom). Reproduced from *Evensen and Farlo* (1997)

not wise to use an estimate close to the zero solution as the first guess in the descent algorithm. To reduce the probability of getting trapped in eventual local minima, an objective analysis estimate, consistent with the measurements, was used as a first guess in the descent algorithm. It was calculated using a smoothing spline minimization algorithm which is equivalent to objective analysis (*McIntosh*, 1990). This could easily be done by replacing the dynamical misfit term with a penalty of a first-guess estimate in the inverse formulation (6.21). Some examples will now be discussed.

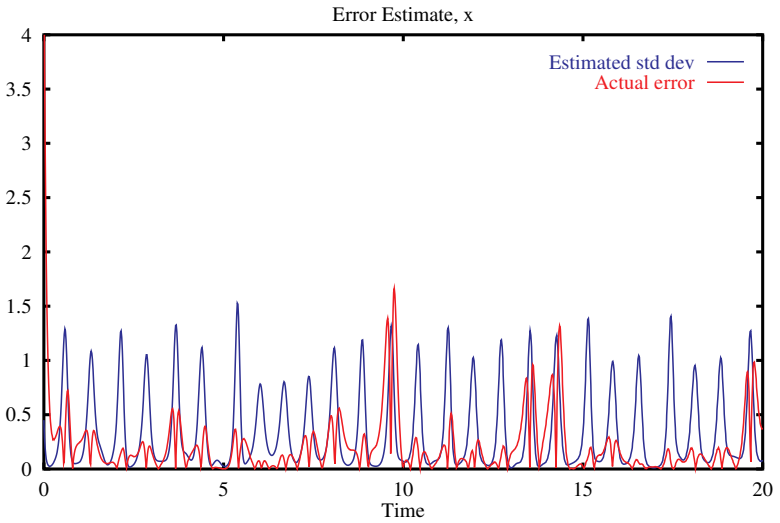

Fig. 6.6. Case A: Statistical error estimates (standard deviations) for x together with the absolute value of the actual errors. Reproduced from *Evensen and Fario* (1997)

Case A

This case can be considered as a base case and is, except for the lower measurement errors, similar to the case discussed by *Miller et al.* (1994); i.e. the time interval is $t \in [0, 20]$ and the distance between the measurements is $\Delta t_{\text{obs}} = 0.25$. The gradient descent method was in this case capable of finding the global minimum when starting from the objective analysis estimate. The minimizing solution for the three variables is given in Figs. 6.4 and 6.5 together with the terms in the penalty function as a function of iteration. We find it amazing how close the inverse estimate is to the reference solution. The quality of this inverse estimate is clearly superior to previous inverse calculations using the extended Kalman filter or a strong constraint formulation.

From the terms in the penalty function given in Fig. 6.4, it is seen that the first guess is close to the measurements and rather smooth, while the dynamical residuals are large and contribute with more than 99 % of the total value of the cost function. During the iterations, the dynamical misfit is reduced while there is an initial increase in the smoothing and measurement terms, which indicates that the final inverse solution is further from the measurements and less smooth than the first guess.

The hybrid Monte Carlo method was used to estimate the standard deviations of the errors in the minimizing solution. These are plotted together with the true differences between the estimate and the reference solution in Fig. 6.6 for the x-component. The largest errors appear around the peaks of the solution and the statistical and true errors are similar.

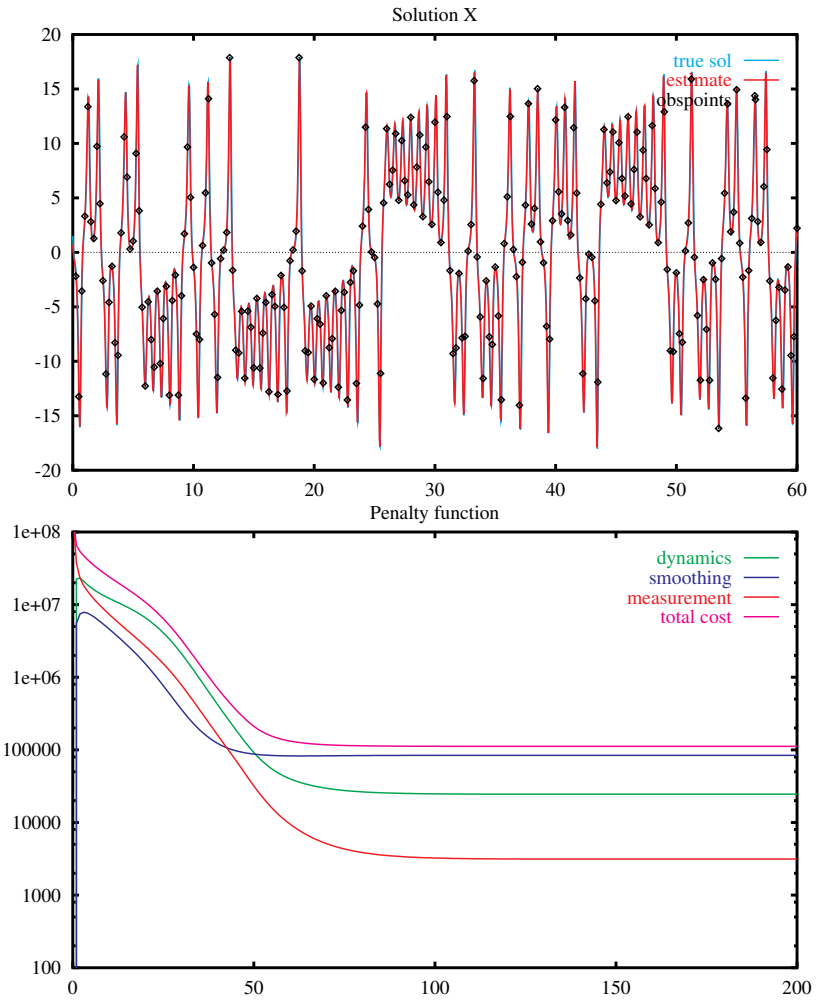

Fig. 6.7. Case B: The inverse estimate for x *(top)* and the penalty function *(bottom)*. Reproduced from *Evensen and Fario* (1997)

Case B

Here, we extended the time interval to $T = 60$, to test the sensitivity of the inverse estimate with respect to a longer time interval. The number of measurements is increased by a factor of 3 to give the same data density as in Case A. Note that the value of the cost function is also increased by about a factor of 3. This case behaves similarly to Case A, with convergence to the global minimum at a similar rate as in Case A. In Fig. 6.7 the x-component of the solution is given together with the terms in the penalty function.

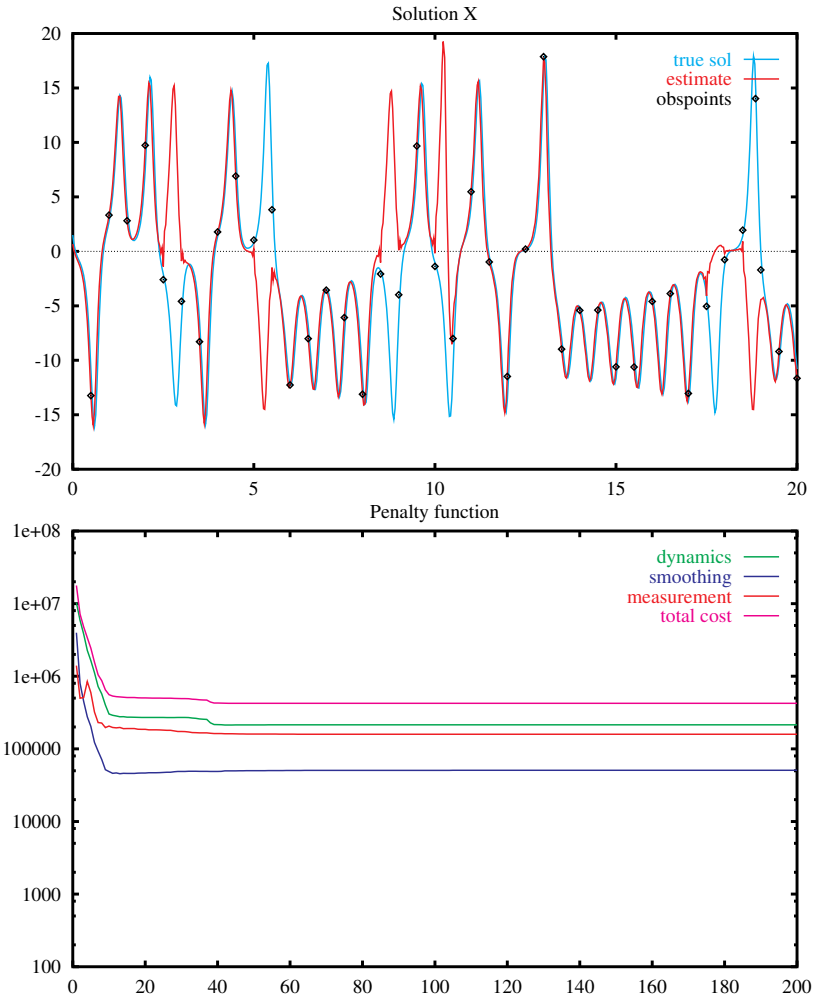

Fig. 6.8. Case C: The inverse estimate for x *(top)* and the penalty function *(bottom)*. Reproduced from *Evensen and Fario* (1997)

An important conclusion from this example is that by using a weak constraint variational formulation for the inverse, the strong sensitivity with respect to perturbations in initial conditions which is observed for strong constraint variational formulations, is completely removed. The weak constraint formulation allows the dynamical model to "forget" very past and future information. The convergence of the inverse calculation therefore has a "local" behaviour where the current estimate at two distant locations have vanishing influence on each other.

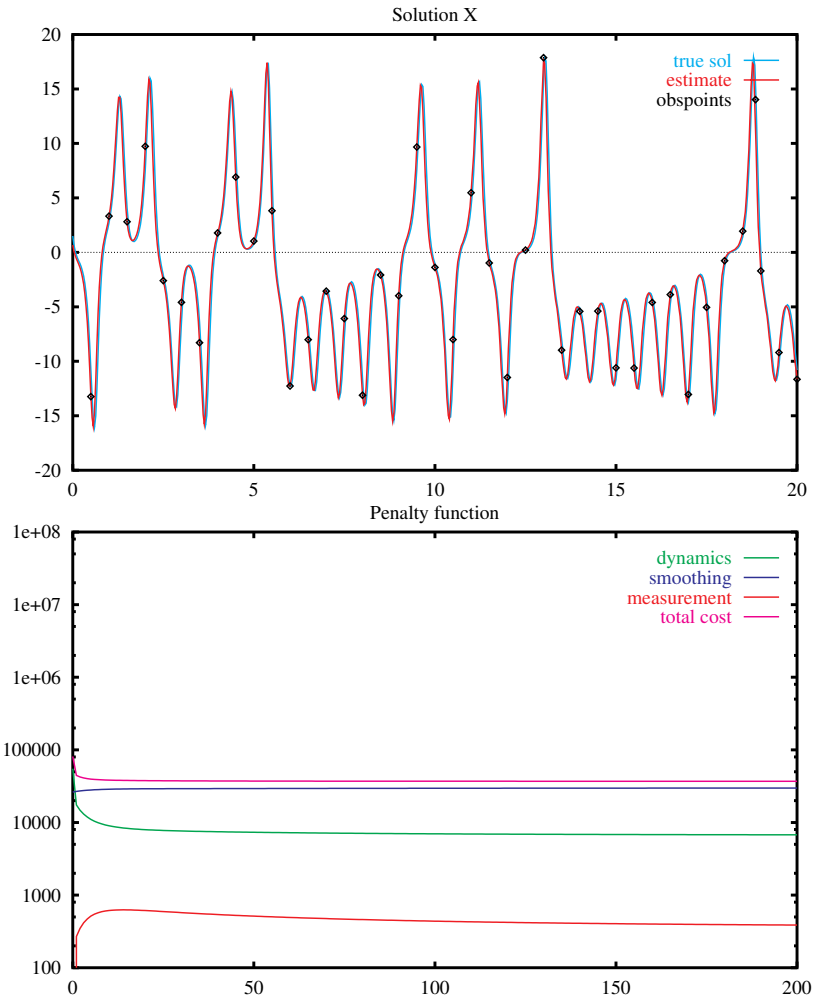

Fig. 6.9. Case C: The inverse estimate for x *(top)* and the penalty function *(bottom)* when the reference solution is used as the first guess in the gradient descent algorithm. Reproduced from *Evensen and Fario* (1997)

Case C

When the distance between the measurements is increased to $\Delta t_{\text{obs}} = 0.50$, a solution is found which misses several of the transitions, as seen in the solution for the x-component given in Fig. 6.8 together with the terms in the penalty function. This is an indication that the gradient algorithm converged to a local minimum. We can verify that this is in fact the case by running another minimization where the true reference solution is used as the first guess for the gradient method. The result is given in Fig. 6.9 where, after

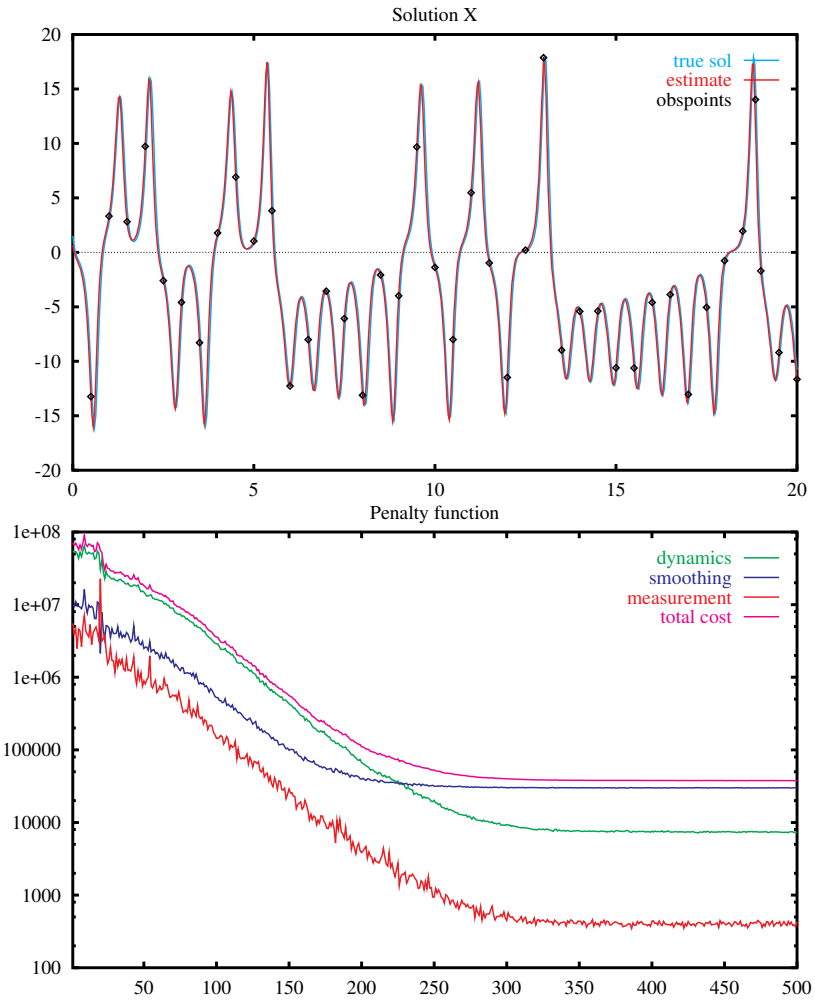

Fig. 6.10. Case C1: The inverse estimate for x *(top)* and the penalty function *(bottom)* where a genetic algorithm based on simulated annealing is used. Reproduced from *Evensen and Fario* (1997)

a minor initial adjustment, the algorithm converges to the global minimum which has a significantly lower value of the cost function and which captures all the transitions. Thus, we can conclude that when the measurement density is lowered the measurement term will give a smaller quadratic contribution to the cost function and at some stage local minima start to appear.

Case C1

This case is similar to Case C, but now the hybrid Monte Carlo method is used in combination with simulated annealing for minimizing the penalty function. The minimizing solution is in this case given in Fig. 6.10. Note that the number of iterations required for convergence is higher in this case than in the previous ones. This is due to perturbations caused by the annealing process that allows uphill moves to migrate out of local minima. The method used here is actually not proper annealing but should be denoted quenching, since the system is cooled too fast to guarantee that the global minimum will be found. In fact, in a similar case in *Evensen and Fario* (1997) a local minimum was found.

6.2.4 Discussion

A weak constraint variational formulation for the Lorenz model has been minimized using a gradient descent method.

It has been illustrated that by imposing the dynamical model as a weak constraint, by allowing the dynamics to contain errors, this leads to a better posed problem than the strong constraint formulation. The weak constraint formulation eliminates the sensitivity with respect to the initial conditions since, by allowing for model errors, the estimate can deviate from an exact model trajectory and thereby forget very past and future information. Further, there are no limitations on the length of the assimilation interval.

The inverse was calculated using the full state in "space" and time as control variables. The huge state space associated with such a formulation is the main objection against using a gradient descent method for a weak constraint inverse calculation. It could be compared to the mathematically very appealing representer method (Bennett, 1992), where the solution is searched for in a space with dimension equal to the number of measurements. On the other hand, with a gradient descent approach there is no need to integrate any dynamical equations, since a new candidate for the solution in space and time is substituted in every iteration. This gives rise to the notation substitution methods, where the important issue is the method used for proposing the solution candidates.

A gradient descent method will always provide a solution. However, it may be a local minimum if the penalty function is not convex. Statistical methods based on simulated annealing in combination with a hybrid Monte Carlo method for generating the candidates are much more expensive than a gradient descent approach but has a higher probability of finding the global minimum. The genetic methods will, for practical problems, only lead to a marginal improvement since they can only solve a slightly more difficult problem to a much larger cost. Thus, one should rather try to define a better posed problem, e.g. by introducing additional measurements.

It should be noted that with reasonable good measurement coverage the penalty function is essentially convex, but when either the number of measurements is decreased or with poorer quality of the measurements, the quadratic contribution to the penalty function from the measurement term has less influence and nonlinearities in the dynamics may give raise to local minima. Thus, the success of the substitution methods is strongly dependent on the measurement density. With sufficient number of measurements the algorithms converged to the global minimum of the weak constraint problem. When the number of measurements decreased, this resulted in a penalty function with multiple local minima and the gradient descent method was unable to converge to the global minimum.

It should also be pointed out that the gradient descent method does not directly provide error estimates for the minimizing solution. However, if the gradient descent method is first used to find the solution then the hybrid Monte Carlo method can be used to sample from the posterior distribution function and error variance estimates can be calculated.

An example of this method was used by *Natvik et al.* (2001) with a simple but nonlinear three component marine ecosystem model. In this case the dimension of the problem was equal to three variables times the number of grid nodes in time. Results similar to those found by *Evensen* (1997) were obtained, and the global minimum was found in the cases with sufficient measurement density. With a small number of measurements the gradient method converged to a local minimum.

The substitution methods solve for a state vector which consists of the model state vector in space and time. Clearly, this can be very large for realistic models and it does not appear to be a smart approach since we noted that the real dimension of the linear inverse problem equals the number of measurements. If the number of grid nodes is large, slow convergence is expected, and this was indeed a result from these studies.

In a final case, similar to case A, but only using measurements of the x-component of the solution, the global minimum was still found using the gradient descent method. In this case the estimates for y and z were entirely determined by the choice of model error covariance matrix and interactions through the dynamical equations. However, this case converged significantly slower. This is a result of poor conditioning and can be expected since the quadratic contribution from the measurement term is lower when only the x-component of the solution is measured. It also indicates that if the method is used with high dimensional problems, or with to sparse measurements, convergence problems may become crucial.

7

Probabilistic formulation

In the previous chapters we have discussed some traditional data assimilation methods and illustrated these with some simple examples. We will now present a mathematically and statistically consistent formulation of the combined parameter and state estimation problem. The starting point is Bayes' theorem which defines the posterior probability density function of the poorly known parameters and the model solution conditioned on a set of observations.

In the following chapters it will be seen that both the generalized inverse formulation and the EnKF as well as ensemble smoothers can be derived from Bayes' theorem. In addition it will be possible to properly interpret different assimilation methods and understand the assumption and approximations they relay on, and what they solve for.

The introduction of poorly known parameters does not complicate the discussion much. It is done since the parameter estimation problem is closely related to the state estimation problem, and it should in fact be treated as a combined parameter and state estimation problem. This is a fact that many works on parameter estimation have ignored, probably because the theoretical foundation for these problems and their solution methods have not previously been worked out.

7.1 Joint parameter and state estimation

The parameter estimation problem for a dynamical model can in a general form be formulated as "*how to find the joint pdf of the parameters and model state, given a set of measurements and a dynamical model with known uncertainties.*"

This is vastly different from the traditional approach which is normally formulated as either "*how to find an estimate of the parameters which is as close as possible to the first guess values of the parameters and which results in a model solution which is as close as possible to a set of measurements*" or even simpler "*how to find the parameters resulting in a model solution*

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_7,

which is as close as possible to a set of measurements". Using these definitions, the dynamical model is considered to be perfect except for the errors in the poorly known parameters. A cost function, which measures the distance between the model solution and the measurements plus the deviation between the estimated parameter and its prior with some relative weight, is normally minimized with respect to the parameters.

Alternatively, a pure state estimation problem as was considered in the previous chapters can be defined. One is then searching for *the pdf of the model solution given a number of measurements related to the model solution.*

7.2 Model equations and measurements

We define a model with associated initial and boundary conditions on the spatial domain $\mathcal{D}$ with boundary $\partial\mathcal{D}$, and a set of observations,

$$\frac{\partial \boldsymbol{\psi}(\boldsymbol{x},t)}{\partial t} = \boldsymbol{G}\big(\boldsymbol{\psi}(\boldsymbol{x},t),\boldsymbol{\alpha}(\boldsymbol{x})\big) + \boldsymbol{q}(\boldsymbol{x},t), \tag{7.1}$$

$$\boldsymbol{\psi}(\boldsymbol{x},t_0) = \boldsymbol{\Psi}_0(\boldsymbol{x}) + \boldsymbol{a}(\boldsymbol{x}), \tag{7.2}$$

$$\boldsymbol{\psi}(\boldsymbol{x},t)|_{\partial\mathcal{D}} = \boldsymbol{\Psi}_b(\boldsymbol{\xi},t) + \boldsymbol{b}(\boldsymbol{\xi},t), \tag{7.3}$$

$$\boldsymbol{\alpha}(\boldsymbol{x}) = \boldsymbol{\alpha}_0(\boldsymbol{x}) + \boldsymbol{\alpha}'(\boldsymbol{x}), \tag{7.4}$$

$$\mathcal{M}[\boldsymbol{\psi},\boldsymbol{\alpha}] = \boldsymbol{d} + \boldsymbol{\epsilon}. \tag{7.5}$$

The model state $\boldsymbol{\psi}(\boldsymbol{x},t) \in \Re^{n_\psi}$ is a vector consisting of the n_ψ model variables where each variable is a function of space and time. The nonlinear model is defined by (7.1) where $\boldsymbol{G}(\boldsymbol{\psi},\boldsymbol{\alpha}) \in \Re^{n_\psi}$ is the nonlinear model operator. More general forms can be used for the nonlinear model operator, but the present one will suffice to demonstrate the methodologies considered here.

The model state is assumed to evolve in time from the initial state $\boldsymbol{\Psi}_0(\boldsymbol{x}) \in \Re^{n_\psi}$ defined in (7.2), under the constraints of the boundary conditions $\boldsymbol{\Psi}_b(\boldsymbol{\xi},t) \in \Re^{n_\psi}$ defined in (7.3). The coordinate $\boldsymbol{\xi}$ is running over the surface $\partial\mathcal{D}$ where the boundary condition is defined.

We have defined $\boldsymbol{\alpha}(\boldsymbol{x}) \in \Re^{n_\alpha}$ as a set of n_α poorly known parameters of the model. These can be both a vector of spatial fields, in the form they are written here, or a vector of scalars, and they are assumed to be constant in time. A first guess value $\boldsymbol{\alpha}_0(\boldsymbol{x}) \in \Re^{n_\alpha}$, of the vector of parameters $\boldsymbol{\alpha}(\boldsymbol{x}) \in \Re^{n_\alpha}$, is introduced through (7.4).

Additional conditions are present in the form of the measurements $\boldsymbol{d} \in \Re^M$. These can be direct point measurements of the model solution or more complex parameters which are nonlinearly related to the model state. For the time being we will restrict ourselves to the case with linear measurements. An example of a direct measurement functional is then

$$\mathcal{M}_i[\boldsymbol{\psi}] = \iint \boldsymbol{\psi}^{\mathrm{T}}(\boldsymbol{x},t)\boldsymbol{\delta}_{\psi_i}\delta(t-t_i)\delta(\boldsymbol{x}-\boldsymbol{x}_i)dt\,dx, \tag{7.6}$$

where the integration is over the space and time domain of the model. The measurement d_i, is related to the model state variable as selected by the vector $\boldsymbol{\delta}_{\psi_i}$, and evaluated at the space and time location $(\boldsymbol{x}_i, t_i)$. If a three-variable model is used and the second variable is measured, then $\boldsymbol{\delta}_{\psi_i}$ becomes the vector $(0, 1, 0)^{\mathrm{T}}$ while $\delta(t - t_i)$ and $\delta(\boldsymbol{x} - \boldsymbol{x}_i)$ are Dirac delta functions.

In (7.1–7.5) we have also included unknown error terms which are representing the errors in the model equations, the initial and boundary conditions, the first guess for the model parameters and the measurements. Without these error terms the system as given above is over-determined and has no solution. On the other hand, when we introduce these error terms without additional conditions there are infinitively many solutions of the system. The way to proceed is to introduce a statistical hypothesis about the errors, e.g. assuming that they are normally distributed with means equal to zero and known error covariances.

7.3 Bayesian formulation

We now consider the model variables, the poorly known parameters, the initial and boundary conditions and the measurements as random variables which can be described by pdfs.

The joint pdf for the model state as a function of space and time and the parameters is $f(\boldsymbol{\psi}, \boldsymbol{\alpha})$. Further, for the measurements we can define the likelihood function $f(\boldsymbol{d}|\boldsymbol{\psi}, \boldsymbol{\alpha})$, thus we can measure both the model state and the parameters. Using Bayes' theorem the parameter estimation problem can be written as

$$f(\boldsymbol{\psi}, \boldsymbol{\alpha}|\boldsymbol{d}) \propto f(\boldsymbol{\psi}, \boldsymbol{\alpha}) f(\boldsymbol{d}|\boldsymbol{\psi}, \boldsymbol{\alpha}). \tag{7.7}$$

We have not included a denominator which normalizes the right-hand-side, thereby writing proportional to, $\propto$, rather than equal to.

Parameter estimation problems normally do not include the model state as a variable to be estimated. It is more common to first solve for the poorly known parameters alone, and then rerun the model to find the model solution. This implicitly assumes that the model, with the new estimates of the parameters, does not contain any errors. Generally, this is not a valid assumption.

In the dynamical model, we have specified initial and boundary conditions as random variables and we have included prior information about the parameters. Thus, we define the pdfs $f(\boldsymbol{\psi}_0)$, $f(\boldsymbol{\psi}_b)$ and $f(\boldsymbol{\alpha})$, for the estimates $\boldsymbol{\psi}_0$, $\boldsymbol{\psi}_b$ and $\boldsymbol{\alpha}$, of the initial and boundary conditions, and the parameters. We then write instead of $f(\boldsymbol{\psi}, \boldsymbol{\alpha})$,

$$\begin{aligned} f(\boldsymbol{\psi}, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b) &= f(\boldsymbol{\psi}, \boldsymbol{\alpha}|\boldsymbol{\psi}_0, \boldsymbol{\psi}_b) f(\boldsymbol{\psi}_0) f(\boldsymbol{\psi}_b) \\ &= f(\boldsymbol{\psi}|\boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b) f(\boldsymbol{\psi}_0) f(\boldsymbol{\psi}_b) f(\boldsymbol{\alpha}). \end{aligned} \tag{7.8}$$

Equation (7.7) should accordingly be written as

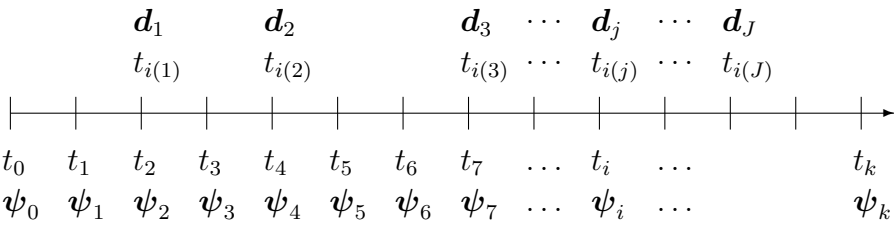

Fig. 7.1. Discretization in time. The time interval is discretized into $k+1$ nodes, at the times t_0 to t_k, where the model state vector $\boldsymbol{\psi}_i = \boldsymbol{\psi}(t_i)$ is defined. The measurement vectors $\boldsymbol{d}_j$ are available at the discrete subset of times $t_{i(j)}$, where $j = 1, \ldots, J$

$$f(\boldsymbol{\psi}, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b | \boldsymbol{d}) \propto f(\boldsymbol{\psi}|\boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b) f(\boldsymbol{\psi}_0) f(\boldsymbol{\psi}_b) f(\boldsymbol{\alpha}) f(\boldsymbol{d}|\boldsymbol{\psi}, \boldsymbol{\alpha}), \tag{7.9}$$

where it is also assumed that the boundary conditions and initial conditions are independent, although, this may not be true for their intersection at t_0. Here the pdf $f(\boldsymbol{\psi}|\boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b)$ is the prior density for the model solution given the parameters and initial and boundary conditions.

7.3.1 Discrete formulation

In the following discussion it is convenient to work with a model state which is discretized in time, i.e. $\boldsymbol{\psi}(\boldsymbol{x}, t)$ is represented at fixed time intervals as $\boldsymbol{\psi}_i(\boldsymbol{x}) = \boldsymbol{\psi}(\boldsymbol{x}, t_i)$ with $i = 0, 1, \ldots, k$. Please refer to Fig. 7.1 for further illustration.

Furthermore, we define the pdf for the model integration from time t_{i-1} to t_i as $f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1}, \boldsymbol{\alpha}, \boldsymbol{\psi}_b(t_i))$, which assumes that the model is a first order Markov process. In the general case when model errors are time correlated this could be written as $f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_k, \ldots, \boldsymbol{\psi}_{i+1}, \boldsymbol{\psi}_{i-1}, \ldots, \boldsymbol{\psi}_0, \boldsymbol{\alpha}, \boldsymbol{\psi}_b(t_i))$ which for simplicity is written as $f\big(\boldsymbol{\psi}_i | \{\boldsymbol{\psi}_{l \neq i}\}, \boldsymbol{\alpha}, \boldsymbol{\psi}_b(t_i)\big)$.

The joint pdf for the model solution and the parameters in (7.8) can now be written

$$f(\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b) \propto f(\boldsymbol{\alpha}) f(\boldsymbol{\psi}_b) f(\boldsymbol{\psi}_0) \prod_{i=1}^{k} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1}, \boldsymbol{\alpha}, \boldsymbol{\psi}_b). \tag{7.10}$$

We now assume that the measurements $\boldsymbol{d} \in \Re^M$ can be divided into subsets of measurement vectors $\boldsymbol{d}_j \in \Re^{m_j}$, collected at times $t_{i(j)}$, with $j = 1, \ldots, J$ and $0 < i(1) < i(2) < \ldots < i(J) < k$. The subset $\boldsymbol{d}_j$ will only depend on $\boldsymbol{\psi}(t_{i(j)}) = \boldsymbol{\psi}_{i(j)}$ or $\boldsymbol{\alpha}$. Further, it is assumed that the measurement errors are uncorrelated in time. We can then write

$$f(\boldsymbol{d}|\boldsymbol{\psi}, \boldsymbol{\alpha}) = \prod_{j=1}^{J} f(\boldsymbol{d}_j|\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}). \tag{7.11}$$

From Bayes' theorem, we now get

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_k,\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}) \propto f(\boldsymbol{\alpha})f(\boldsymbol{\psi}_0)f(\boldsymbol{\psi}_b)\prod_{i=1}^{k} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha})\prod_{j=1}^{J} f(\boldsymbol{d}_j|\boldsymbol{\psi}_{i(j)},\boldsymbol{\alpha}). \tag{7.12}$$

The general case, when the model is not a first order Markov process, becomes

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_k,\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}) \propto f(\boldsymbol{\alpha})f(\boldsymbol{\psi}_0)f(\boldsymbol{\psi}_b)\prod_{i=1}^{k} f\left(\boldsymbol{\psi}_i|\left\{\boldsymbol{\psi}_{l\neq i}\right\},\boldsymbol{\alpha}\right)\prod_{j=1}^{J} f(\boldsymbol{d}_j|\boldsymbol{\psi}_{i(j)},\boldsymbol{\alpha}), \tag{7.13}$$

i.e. the model state at time t_i is dependent on the model state at all other times. This is the case when time correlated model errors are used. The previous equations constitute the most general formulation of the state and parameter estimation problem.

7.3.2 Sequential processing of measurements

We will now assume that the model can be written as a first order Markov process. This is not a strong assumption or simplification. It was shown by *Reichle et al.* (2002) and *Evensen* (2003) that in the case of time correlated model errors, it is still possible to reformulate the problem as a first order Markov process by augmenting the model errors to the model state vector. A simple equation forced by white noise can be used to simulate the time evolution of the model errors.

Evensen and van Leeuwen (2000) showed that a general smoother and filter could be derived from the Bayesian formulation given in (7.12). We now rewrite (7.12) as follows:

$$\begin{aligned} f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_k,&\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}) \propto f(\boldsymbol{\alpha})f(\boldsymbol{\psi}_0)f(\boldsymbol{\psi}_b) \\ &\prod_{i=1}^{i(1)} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha})f(\boldsymbol{d}_1|\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha}) \\ &\vdots \\ &\prod_{i=i(J-1)+1}^{i(J)} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha})f(\boldsymbol{d}_J|\boldsymbol{\psi}_{i(J)},\boldsymbol{\alpha}) \\ &\prod_{i=i(J)+1}^{k} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha}). \end{aligned} \tag{7.14}$$

This expression can be evaluated sequentially in time as shown below, and the result will be identical to the one obtained by direct evaluation of (7.12),

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1) \propto f(\boldsymbol{\alpha})f(\boldsymbol{\psi}_0)f(\boldsymbol{\psi}_b) \prod_{i=1}^{i(1)} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha})f(\boldsymbol{d}_1|\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha}), \tag{7.15}$$

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(2)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1,\boldsymbol{d}_2) \propto f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1) \prod_{i=i(1)+1}^{i(2)} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha})f(\boldsymbol{d}_2|\boldsymbol{\psi}_{i(2)},\boldsymbol{\alpha}), \tag{7.16}$$

$$\vdots$$

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(J)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1,\ldots,\boldsymbol{d}_J) \propto f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(J-1)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1,\ldots,\boldsymbol{d}_{J-1}) \prod_{i=i(J-1)+1}^{i(J)} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha})f(\boldsymbol{d}_J|\boldsymbol{\psi}_{i(J)},\boldsymbol{\alpha}), \tag{7.17}$$

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_k,\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1,\ldots,\boldsymbol{d}_J) \propto f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(J)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1,\ldots,\boldsymbol{d}_J) \prod_{i=i(J)+1}^{k} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha}). \tag{7.18}$$

From these equations it is clear that, as long as the model is a first order Markov process and the measurements are available at discrete times with errors uncorrelated in time, we can process the measurements sequentially in time.

In (7.15) we compute the joint conditional pdf for the solution in the interval $[t_1, t_{i(1)}]$, the parameter $\boldsymbol{\alpha}$ and the initial and boundary condition, given the measurements $\boldsymbol{d}_1$.

This joint conditional pdf becomes the prior in (7.16) where the information from the measurements $\boldsymbol{d}_2$ are introduced and the time interval is extended to $[t_1, t_{i(2)}]$. Thus, we compute the joint conditional pdf for the solution in the interval $[t_1, t_{i(2)}]$, the parameter $\boldsymbol{\alpha}$ and the initial and boundary condition, given the measurements $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$.

We can continue this sequential updating until all measurements have been processed and we get the pdf in (7.17). Thereafter, (7.18) is the prediction of $\boldsymbol{\psi}_{i(m)+1},\ldots,\boldsymbol{\psi}_k$, starting from the joint conditional pdf from (7.17).

We note again that these equations do not introduce any important approximations and thus describe the full inverse problem. Further, we claim that for many problems this sequential procedure provides a better posed approach for solving the inverse problem than trying to process all the measurements simultaneously as is normally done in variational formulations. The sequential processing is also very convenient for typical forecasting problems where new measurements can be processed when they arrive without recomputing the full inversion.

7.4 Summary

We have formulated the combined parameter and state estimation problem using Bayesian statistics and have seen that, under a condition of measurement errors being independent in time and the dynamical model being a Markov processes, a recursive formulation can be used for Bayes' theorem where measurements are processed sequentially in time.

The assumption of the model being a Markov process can be relaxed by defining a first order auto-regressive formula for the model errors and augmenting the model errors to the model state. In this case the Bayesian formulation also solves for the model errors.

It is seen that by augmenting the poorly known parameters to the model state we obtain a formulation where the model state and the parameters are solved for simultaneously. Hence, we have a combined parameter and state estimation problem.

In the next chapter we will use the standard Bayesian formulation as given by either (7.12) or (7.13) to derive the generalized variational inverse formulation for the combined parameter and state estimation problem.

Then in Chap. 9 the Ensemble Smoother (ES) is derived from the standard Bayesian formulation while the recursive form of Bayes' theorem, given by (7.15–7.18), is used to derive the Ensemble Kalman Smoother (EnKS) and the Ensemble Kalman Filter (EnKF).

8

Generalized Inverse

The variational inverse problems discussed in Chap. 5 can be derived from the Bayesian formulation presented in the previous Chapter by assuming Gaussian statistics for the priors. This was previously demonstrated by *van Leeuwen and Evensen* (1996) using the results from *Jazwinski* (1970). We will now derive the generalized inverse formulation for the combined parameter and state estimation problem starting from Bayes' theorem. Further, the resulting Euler–Lagrange equations are derived and we discuss some solution methods which also allow for the estimation of poorly known model parameters.

8.1 Generalized inverse formulation

We start from (7.13) and define Gaussian statistics for all the priors, transition densities and likelihoods which occur on the right-hand-side.

8.1.1 Prior density for the poorly known parameters

Assume that we have available a prior estimate $\boldsymbol{\alpha}_0(\boldsymbol{x}) \in \Re^{n_\alpha}$, of $\boldsymbol{\alpha}(\boldsymbol{x}) \in \Re^{n_\alpha}$, as defined in (7.4). Furthermore, the poorly known parameters $\boldsymbol{\alpha}(\boldsymbol{x})$ are assumed to be smooth functions of the spatial coordinates with Gaussian distributed errors. These conditions have the impact of a regularization of the inverse problem since they effectively reduce the degrees of freedom of the problem.

The smoothness of the estimated parameters is controlled by the definition of an error covariance $\boldsymbol{C}_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \Re^{n_\alpha \times n_\alpha}$. Here indices on $\boldsymbol{x}$, i.e. $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$, denote dummy variables in $\mathcal{D}$. We can then define the inverse of $\boldsymbol{C}_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, as $\boldsymbol{W}_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, from

$$\int_{\mathcal{D}} \boldsymbol{C}_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_3) \boldsymbol{W}_{\alpha\alpha}(\boldsymbol{x}_3, \boldsymbol{x}_2) d\boldsymbol{x}_3 = \delta(\boldsymbol{x}_1 - \boldsymbol{x}_2) \boldsymbol{I}, \tag{8.1}$$

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_8,

where $\boldsymbol{I} \in \Re^{n_\alpha \times n_\alpha}$ is the diagonal identity matrix.

Note that a discretization of the parameter on a spatial grid leads to the use of matrices $\boldsymbol{C}_{\alpha\alpha}$ and $\boldsymbol{W}_{\alpha\alpha}$. Equation (8.1) is then replaced by a matrix-matrix multiplication, defining $\boldsymbol{C}_{\alpha\alpha}$ as the matrix inverse of $\boldsymbol{W}_{\alpha\alpha}$.

The prior pdf for $\boldsymbol{\alpha}$ then becomes

$$f(\boldsymbol{\alpha}) \propto \exp\Bigg(-\frac{1}{2}\iint_{\mathcal{D}} \big(\boldsymbol{\alpha}(\boldsymbol{x}_1) - \boldsymbol{\alpha}_0(\boldsymbol{x}_1)\big)^{\mathrm{T}} \boldsymbol{W}_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2)\big(\boldsymbol{\alpha}(\boldsymbol{x}_2) - \boldsymbol{\alpha}_0(\boldsymbol{x}_2)\big)\, d\boldsymbol{x}_1\, d\boldsymbol{x}_2\Bigg). \tag{8.2}$$

8.1.2 Prior density for the initial conditions

The errors in the initial conditions are also assumed to have a Gaussian distribution, where $\Psi_0(\boldsymbol{x}) \in \Re^{n_\psi}$ is the prior for the initial state, and $\boldsymbol{C}_{aa}(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \Re^{n_\psi \times n_\psi}$ defines the error covariance of the initial condition. As above we define the inverse of the error covariance $\boldsymbol{W}_{aa}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, from

$$\int_{\mathcal{D}} \boldsymbol{C}_{aa}(\boldsymbol{x}_1, \boldsymbol{x}_3)\boldsymbol{W}_{aa}(\boldsymbol{x}_3, \boldsymbol{x}_2)\, d\boldsymbol{x}_3 = \delta(\boldsymbol{x}_1 - \boldsymbol{x}_2)\boldsymbol{I}, \tag{8.3}$$

with $\boldsymbol{I} \in \Re^{n_\psi \times n_\psi}$.

The prior pdf for the initial state then becomes

$$f(\boldsymbol{\psi}_0) \propto \exp\Bigg(-\frac{1}{2}\iint_{\mathcal{D}} \big(\boldsymbol{\psi}_0(\boldsymbol{x}_1) - \Psi_0(\boldsymbol{x}_1)\big)^{\mathrm{T}} \boldsymbol{W}_{aa}(\boldsymbol{x}_1, \boldsymbol{x}_2)\big(\boldsymbol{\psi}_0(\boldsymbol{x}_2) - \Psi_0(\boldsymbol{x}_2)\big)\, d\boldsymbol{x}_1\, d\boldsymbol{x}_2\Bigg). \tag{8.4}$$

8.1.3 Prior density for the boundary conditions

For the boundary condition which is defined on $\partial\mathcal{D}$ for all times $t \in [t_0, t_k]$, we define the covariance $\boldsymbol{C}_{bb}(\boldsymbol{\xi}_1, t_1, \boldsymbol{\xi}_2, t_2) \in \Re^{n_\psi \times n_\psi}$ which has the inverse $\boldsymbol{W}_{bb}(\boldsymbol{\xi}_1, t_1, \boldsymbol{\xi}_2, t_2)$ defined as

$$\begin{aligned}\int_{t_0}^{t_k} \int_{\partial\mathcal{D}} \boldsymbol{C}_{bb}(\boldsymbol{\xi}_1, t_1, \boldsymbol{\xi}_3, t_3)\boldsymbol{W}_{bb}(\boldsymbol{\xi}_3, t_3, \boldsymbol{\xi}_2, t_2)\, d\boldsymbol{\xi}_3\, dt_3 \\ = \delta(\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2)\boldsymbol{\delta}(t_1 - t_2)\boldsymbol{I},\end{aligned} \tag{8.5}$$

where $\boldsymbol{x}_b$ is a coordinate over the surface $\partial\mathcal{D}$ and $\boldsymbol{I} \in \Re^{n_\psi \times n_\psi}$. The prior pdf for the boundary conditions then becomes

$$f(\boldsymbol{\psi}_b) \propto \exp\Bigg(-\frac{1}{2}\iint_{\delta\mathcal{D}}\int_{t_0}^{t_k} \big(\boldsymbol{\psi}(\boldsymbol{\xi}_1, t_1) - \boldsymbol{\psi}_b(\boldsymbol{\xi}_1, t_1)\big)^{\mathrm{T}} \boldsymbol{W}_{bb}(\boldsymbol{\xi}_1, t_1, \boldsymbol{\xi}_2, t_2)\big(\boldsymbol{\psi}(\boldsymbol{\xi}_2, t_2) - \boldsymbol{\psi}_b(\boldsymbol{\xi}_2, t_2)\big)\, dt_1\, dt_2\, d\boldsymbol{\xi}_1\, d\boldsymbol{\xi}_2\Bigg). \tag{8.6}$$

8.1.4 Prior density for the measurements

We will continue using the assumption that measurement errors are uncorrelated in time, although at least for the variational formulation this assumption is not required. With $\boldsymbol{C}_{\epsilon\epsilon}(t_{i(j)}) = \boldsymbol{W}_{\epsilon\epsilon}^{-1}(t_{i(j)}) \in \Re^{m_j \times m_j}$, with m_j being the number of measurements at time $t_{i(j)}$, we can write

$$f(\boldsymbol{d}_j|\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}) \propto \exp\Big(-\frac{1}{2}\Big(\boldsymbol{d}_j - \mathcal{M}_j\big[\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}(t_{i(j)})\Big(\boldsymbol{d}_j - \mathcal{M}_j\big[\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}\big]\Big)\Big), \tag{8.7}$$

for the prior information on the measurements. Here, we have used the vector of measurement functionals $\mathcal{M}_j \in \Re^{m_j}$, which corresponds to the vector of measurements $\boldsymbol{d}_j \in \Re^{m_j}$, and which takes the model state vector at the time $t_{i(j)}$, and possibly the parameter $\boldsymbol{\alpha}$, as arguments.

For the further discussion we write

$$\begin{aligned} f(\boldsymbol{d}|\boldsymbol{\psi}, \boldsymbol{\alpha}) &\propto \prod_{j=1}^{m} f(\boldsymbol{d}_j|\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}) \\ &= \exp\Big(-\frac{1}{2}\sum_{j=1}^{m}\Big(\boldsymbol{d}_j - \mathcal{M}_j\big[\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}(t_{i(j)})\Big(\boldsymbol{d}_j - \mathcal{M}_j\big[\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}\big]\Big)\Big) \\ &= \exp\Big(-\frac{1}{2}\Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{\psi}, \boldsymbol{\alpha}\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{\psi}, \boldsymbol{\alpha}\big]\Big)\Big), \end{aligned} \tag{8.8}$$

where $\boldsymbol{W}_{\epsilon\epsilon}$ is a matrix with the J sub-matrices, $\boldsymbol{W}_{\epsilon\epsilon}(t_{i(j)})$, on the diagonal.

8.1.5 Prior density for the model errors

Given a dynamical model we define the probability density functions for the model error using an assumption of Gaussian statistics. The model residual term is obtained from a short derivation and for simplicity we use a scalar model. The extension to a more general model like (7.1) is straight-forward.

We start by defining the discrete dynamical scalar model as

$$\psi_{i+1} = \psi_i + G(\psi_i, \alpha)\Delta t + q_i. \tag{8.9}$$

Here the function $G(\psi_i, \alpha)$ is a nonlinear model operator and q_i is an additive stochastic noise process. More general noise processes such as $G(\psi_i, q_i)$ can be treated as additive if we augment q_i to the state vector and define an additional equation which models q_i as an additive noise process.

It is useful to represent the noise as

$$q_i = \sigma\sqrt{\Delta t}\omega_i, \tag{8.10}$$

where $\overline{\omega_i\omega_j} = \Omega_{i,j}$ has unit variance and further defines correlations in time. Then σ is the standard deviation of the stochastic noise and the factor $\sqrt{\Delta t}$

ensures that the increase of variance with time will be independent of the time step used.

We can define the error covariance of the model noise as

$$C_{qq}(i,j) = \overline{q_i q_j} = \sigma^2 \Delta t \overline{\omega_i \omega_j}, \tag{8.11}$$

or

$$\boldsymbol{C}_{qq} = \sigma^2 \Delta t \boldsymbol{\Omega}, \tag{8.12}$$

i.e. for white model noise the increase in variance over a time unit is σ^2. The case with coloured noise is further treated in Chap. 12.

Now define the inverse $\boldsymbol{W}_{qq}$ of $\boldsymbol{C}_{qq}$ such that $\boldsymbol{W}_{qq}\boldsymbol{C}_{qq} = \boldsymbol{I}$, thus

$$\boldsymbol{W}_{qq} = \sigma^{-2} \Delta t^{-1} \boldsymbol{\Omega}^{-1}. \tag{8.13}$$

We can now define the squared and weighted model residual terms, $q_i W_{qq}(i,j) q_j$, and the sum over i and j defines the measure of the total model misfit. In the limit when $\Delta t \to 0$ we can write

$$\begin{aligned}
&\sum_{ij} \frac{q_i}{\Delta t} \Delta t W_{qq}(i,j) \Delta t \frac{q_j}{\Delta t} \\
&= \sum_{ij} \left(\frac{\psi_{i+1} - \psi_i}{\Delta t} - G_i \right) \Delta t W_{qq}(i,j) \Delta t \left(\frac{\psi_{i+1} - \psi_i}{\Delta t} - G_i \right) \\
&\to \iint_{t_0}^{t_k} \left(\frac{\partial \psi}{\partial t} - G(\psi, \alpha) \right)_{t_1} W_{qq}(t_1, t_2) \left(\frac{\partial \psi}{\partial t} - G(\psi, \alpha) \right)_{t_2} dt_1 dt_2,
\end{aligned} \tag{8.14}$$

where $G_i = G(\psi_i, \alpha)$. If model errors are uncorrelated in time then $W_{qq}(i,j) = \sigma^{-2} \Delta t^{-1} \delta(i-j)$, and the sum will be over $q_i W_{qq}(i) q_i$, thus we get,

$$\sum_i q_i W_{qq}(i) q_i \to \int_{t_0}^{t_k} \left(\frac{\partial \psi}{\partial t} - G(\psi) \right) W_{qq}(t) \left(\frac{\partial \psi}{\partial t} - G(\psi) \right) dt. \tag{8.15}$$

The relation to the transition densities in (7.12) and (7.13), when we assume Gaussian statistics, is

$$f(\boldsymbol{\psi}_i | \{\boldsymbol{\psi}_{l \neq i}\}, \boldsymbol{\alpha}) \propto \exp\left(-\frac{1}{2} \sum_j q_i W_{ij} q_j \right), \tag{8.16}$$

and

$$\prod_{i=1}^{k} f(\boldsymbol{\psi}_i | \{\boldsymbol{\psi}_{l \neq i}\}, \boldsymbol{\alpha}) \propto \exp\left(-\frac{1}{2} \sum_{ij} q_i W_{ij} q_j \right), \tag{8.17}$$

where we can replace the summations with the integrals from (8.14) and (8.15) in the limit when $\Delta t \to 0$.

8.1.6 Conditional joint density

Now, introducing the scalar products

$$\bullet \equiv \int_{t_0}^{t_k} \int_{\mathcal{D}} d\boldsymbol{x}\, dt, \quad \circ \equiv \int_{\mathcal{D}} d\boldsymbol{x}, \quad \star \equiv \int_{t_0}^{t_k} \int_{\partial\mathcal{D}} d\boldsymbol{\xi}\, dt, \tag{8.18}$$

we can write the conditional pdf (7.13) as

$$f(\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b | \boldsymbol{d}) \propto \exp\left(-\frac{1}{2}\mathcal{J}[\boldsymbol{\psi}, \boldsymbol{\alpha}]\right), \tag{8.19}$$

where we have defined the function

$$\begin{aligned}
\mathcal{J}[\boldsymbol{\psi}, \boldsymbol{\alpha}] = &\left(\frac{\partial \boldsymbol{\psi}}{\partial t} - \boldsymbol{G}(\boldsymbol{\psi}, \boldsymbol{\alpha})\right)^{\mathrm{T}} \bullet \boldsymbol{W}_{qq} \bullet \left(\frac{\partial \boldsymbol{\psi}}{\partial t} - \boldsymbol{G}(\boldsymbol{\psi}, \boldsymbol{\alpha})\right) \\
&+ (\boldsymbol{\psi}_0 - \boldsymbol{\Psi}_0)^{\mathrm{T}} \circ \boldsymbol{W}_{aa} \circ (\boldsymbol{\psi}_0 - \boldsymbol{\Psi}_0) \\
&+ (\boldsymbol{\psi} - \boldsymbol{\psi}_b)^{\mathrm{T}} \star \boldsymbol{W}_{bb} \star (\boldsymbol{\psi} - \boldsymbol{\psi}_b) \\
&+ (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^{\mathrm{T}} \circ \boldsymbol{W}_{\alpha\alpha} \circ (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) \\
&+ \Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{\psi}\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{\psi}\big]\Big).
\end{aligned} \tag{8.20}$$

Thus, for Gaussian priors, maximization of the conditional joint density in (7.13) is equivalent to minimization of $\mathcal{J}$ as defined in (8.20). The minimum of $\mathcal{J}$ is also the maximum likelihood solution for $\boldsymbol{\psi}$ and $\boldsymbol{\alpha}$ as defined by the conditional joint pdf in (8.19).

The penalty function as defined by $\mathcal{J}$ will have a global minimum, but it may not be unique if the model is nonlinear. It can also possess several local minima and there is a risk of converging to one of these. It is also clear that in the case with no measurements there is a unique solution. This is the prior model solution, or central forecast, from (7.1–7.4) with all error terms set to zero, which then gives a value of $\mathcal{J} \equiv 0$. It corresponds to the maximum likelihood solution of the prior joint pdf and is therefore also named the *modal trajectory* (see *Jazwinski*, 1970).

The generalized inverse problem as defined by (8.20) may appear very complex at first. The introduction of parameters to be estimated, in addition to the state variables, leads to a strongly nonlinear problem even if the dynamical model is linear. However, iterative schemes have been used for the parameters in connection with the representer method by *Eknes and Evensen* (1997) and more recently by *Muccino and Bennett* (2001). This methodology will be further discussed and illustrated with an example from *Eknes and Evensen* (1997) in the following sections. The formulation of the combined parameter and state estimation problem was also discussed by *Evensen et al.* (1998).

From these studies, it became clear that the parameter estimation problem is difficult to solve using standard minimization algorithms due to the

inherent nonlinearities. Other approaches for minimizing the penalty function (8.20) may use the direct iterative methods from Chap. 6 where candidates for a solution are generated, e.g. using the gradient of $\mathcal{J}$ with respect to the parameters and state variables, or even using genetic algorithms. Common for the direct methods is that they are extremely time consuming. The gradient methods may get trapped in local minima. The genetic algorithms should converge to a global minimum but are orders of magnitude more costly than the gradient methods. Because of this, other approaches have introduced assumptions of, e.g. zero model errors and sometimes also zero errors in the initial and/or the boundary conditions. It is clear that one then solves a different problem than the one originally posed and one will not find the correct solution unless these approximations are valid. In fact, one can find unphysical values of parameters which compensate for neglected errors in the model or conditions.

The state space associated with the variables $\boldsymbol{\psi}(\boldsymbol{x})$ and $\boldsymbol{\alpha}(\boldsymbol{x})$ can be huge. This has motivated some approaches for parameter estimation where $\boldsymbol{\alpha}(\boldsymbol{x})$ is approximated by a set of parameters with a smaller effective dimension. It should be noted that the use of a prior like (8.2) correctly reduces the effective dimension of $\boldsymbol{\alpha}(\boldsymbol{x})$ in a statistically consistent manner, and the problem with large state spaces is significantly reduced.

8.2 Solution methods for the generalized inverse problem

We will now use a simple scalar model formulation to illustrate some of the methods that may be used for minimizing (8.20). The use of a scalar model simplifies the notation and we avoid the specification of boundary conditions.

8.2.1 Generalized inverse for a scalar model

With $\psi(t)$ being a scalar model state, the system of equations now becomes

$$\frac{\partial \psi}{\partial t} = G(\psi, \alpha) + q, \tag{8.21}$$

$$\psi(t_0) = \Psi_0 + a, \tag{8.22}$$

$$\alpha = \alpha_0 + \alpha, \tag{8.23}$$

$$\mathcal{M}\big[\psi\big] = \boldsymbol{d} + \boldsymbol{\epsilon}. \tag{8.24}$$

The penalty function then simplifies to

$$\begin{aligned}\mathcal{J}[\psi, \alpha] = &\left(\frac{\partial \psi}{\partial t} - G(\psi, \alpha)\right) \bullet W_{qq} \bullet \left(\frac{\partial \psi}{\partial t} - G(\psi, \alpha)\right) \\ &+ \big(\psi(t_0) - \Psi_0\big) W_{aa} \big(\psi(t_0) - \Psi_0\big) \\ &+ \big(\alpha - \alpha_0\big) W_{\alpha\alpha} \big(\alpha - \alpha_0\big) \\ &+ \Big(\boldsymbol{d} - \mathcal{M}\big[\psi\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \Big(\boldsymbol{d} - \mathcal{M}\big[\psi\big]\Big).\end{aligned} \tag{8.25}$$

Note that, since there is no spatial dimension, we now have

$$\bullet \equiv \int_{t_0}^{t_k} dt, \tag{8.26}$$

and the product $\circ$, is replaced by scalar multiplication.

8.2.2 Euler–Lagrange equations

Note first that ψ is a function of α, since changing α will result in a different ψ. From standard variational calculus we know that $(\psi(\alpha), \alpha)$ defines an extremum if

$$\delta\mathcal{J} = \mathcal{J}\big[\psi(\alpha+\delta\alpha)+\delta\psi', \alpha+\delta\alpha\big] - \mathcal{J}\big[\psi(\alpha),\alpha\big] = \mathcal{O}\big(\delta\alpha^2, \delta\psi'^2\big), \tag{8.27}$$

when $\delta\alpha \to 0$ and $\delta\psi' \to 0$. Here, $\delta\alpha$ is a perturbation of the parameters, which also results in a perturbation ψ which becomes $\psi(\alpha+\delta\alpha) - \psi(\alpha)$. The perturbation $\delta\psi'$ is a perturbation of ψ which is independent of any perturbation of α.

Note that

$$\begin{aligned}\psi(\alpha+\delta\alpha)+\delta\psi' &= \psi(\alpha) + \psi_\alpha\delta\alpha + \delta\psi' + \mathcal{O}\big(\delta\alpha^2, \delta\psi'^2\big)\\ &= \psi(\alpha) + \delta\psi + \mathcal{O}\big(\delta\alpha^2, \delta\psi'^2\big),\end{aligned} \tag{8.28}$$

where we have defined

$$\psi_\alpha = \frac{\partial\psi}{\partial\alpha}, \tag{8.29}$$

and the total perturbation of ψ,

$$\delta\psi = \psi_\alpha\delta\alpha + \delta\psi'. \tag{8.30}$$

The nonlinear model operator can be expanded as

$$\begin{aligned}&G\big(\psi(\alpha+\delta\alpha)+\delta\psi', \alpha+\delta\alpha\big)\\ &\quad= G\big(\psi(\alpha),\alpha\big) + \frac{\partial G}{\partial\psi}(\psi_\alpha\delta\alpha+\delta\psi') + \frac{\partial G}{\partial\alpha}\delta\alpha + \mathcal{O}\big(\delta\alpha^2,\delta\psi'^2\big)\\ &\quad= G\big(\psi(\alpha),\alpha\big) + \frac{\partial G}{\partial\psi}\delta\psi + \frac{\partial G}{\partial\alpha}\delta\alpha + \mathcal{O}\big(\delta\alpha^2,\delta\psi'^2\big).\end{aligned} \tag{8.31}$$

Evaluating $\delta\mathcal{J}$ from (8.27) we get

$$\begin{aligned}\frac{\delta\mathcal{J}}{2} &= \delta\alpha W_{\alpha\alpha}(\alpha-\alpha_0)\\ &\quad+ \delta\psi(t_0)W_{aa}\big(\psi(t_0)-\Psi_0\big)\\ &\quad+ \mathcal{M}^{\mathrm{T}}\big[\delta\psi\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d}-\mathcal{M}\big[\psi\big]\Big)\\ &\quad+ \int_{t_0}^{t_k}\left(\frac{\partial\delta\psi}{\partial t} - \frac{\partial G}{\partial\psi}\delta\psi - \delta\alpha\frac{\partial G}{\partial\alpha}\right)\lambda(t)\,dt\\ &\quad+ \mathcal{O}\big(\delta\alpha^2,\delta\psi'^2\big),\end{aligned} \tag{8.32}$$

where we have defined the "adjoint" variable

$$\lambda(t_1) = \int_{t_0}^{t_k} W_{qq}(t_1, t_2) \left(\frac{\partial \psi}{\partial t} - G(\psi, \alpha) \right)_2 dt_2, \tag{8.33}$$

where the subscript 2 denotes function of t_2. Multiplying this equation with $\int_{t_0}^{t_k} dt_1 C_{qq}(t, t_1)$ from the left gives the equation

$$\frac{\partial \psi}{\partial t} - G(\psi, \alpha) = C_{qq} \bullet \lambda, \tag{8.34}$$

which is the original model with a representation of the model error involving a product between the model error covariance and the adjoint variable on the right hand side.

We now have from integration by part

$$\int_{t_0}^{t_k} \frac{\partial \delta\psi}{\partial t} \lambda \, dt = \delta\psi\lambda \Big|_{t_0}^{t_k} - \int_{t_0}^{t_k} \delta\psi \frac{\partial \lambda}{\partial t} dt. \tag{8.35}$$

Furthermore,

$$\mathcal{M}^{\mathrm{T}}\big[\delta\psi\big] = \int_{t_0}^{t_k} \delta\psi \mathcal{M}^{\mathrm{T}}\big[\delta(t - t_1)\big] dt_1, \tag{8.36}$$

which is easy to demonstrate, e.g. using a direct measurement functional,

$$\mathcal{M}_i\big[\delta(t - t_1)\big] = \int_{t_0}^{t_k} \delta(t - t_1)\delta(t_1 - t_i) dt_1 = \delta(t - t_i). \tag{8.37}$$

We can then write the variation (8.32) as

$$\begin{aligned} \frac{\delta \mathcal{J}}{2} &= \delta\alpha W_{\alpha\alpha}(\alpha - \alpha_0) \\ &+ \delta\psi(t_0) W_{aa}\big(\psi(t_0) - \Psi_0\big) \\ &+ \delta\psi(t_k)\lambda(t_k) - \delta\psi(t_0)\lambda(t_0) \\ &- \int_{t_0}^{t_k} \delta\psi \frac{\partial \lambda}{\partial t} + \delta\psi \frac{\partial G}{\partial \psi}\lambda + \delta\alpha \frac{\partial G}{\partial \alpha}\lambda + \delta\psi \mathcal{M}^{\mathrm{T}}\big[\delta\big] \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \mathcal{M}\big[\psi\big]\Big) \, dt \\ &+ \mathcal{O}\big(\delta\alpha^2, \delta\psi'^2\big). \end{aligned} \tag{8.38}$$

We then reorder the terms to be proportional to either one of the variations $\delta\alpha$, $\delta\psi$, $\delta\psi(t_0)$ and $\delta\psi(t_k)$, to get

$$\begin{aligned} \frac{\delta \mathcal{J}}{2} &= \delta\alpha \left(W_{\alpha\alpha}(\alpha - \alpha_0) - \int_{t_0}^{t_k} \frac{\partial G}{\partial \alpha}\lambda \, dt \right) \\ &+ \delta\psi(t_0)\Big(W_{aa}\big(\psi(t_0) - \Psi_0\big) - \lambda(t_0) \Big) \\ &+ \delta\psi(t_k)\lambda(t_k) \\ &- \int_{t_0}^{t_k} \delta\psi \left(\frac{\partial \lambda}{\partial t} + \frac{\partial G}{\partial \psi}\lambda + \mathcal{M}^{\mathrm{T}}\big[\delta\big] \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \mathcal{M}\big[\psi\big]\Big) \right) \, dt \\ &+ \mathcal{O}\big(\delta\alpha^2, \delta\psi'^2\big). \end{aligned} \tag{8.39}$$

If we require that $\delta\mathcal{J} = \mathcal{O}\big(\delta\alpha^2, \delta\psi'^2\big)$ we must have

$$\frac{\partial\psi}{\partial t} = G(\psi,\alpha) + C_{qq} \bullet \lambda, \tag{8.40}$$

$$\psi(t_0) = \Psi_0 + C_{aa}\lambda(t_0), \tag{8.41}$$

$$\frac{\partial\lambda}{\partial t} = -\frac{\partial G}{\partial\psi}\lambda - \mathcal{M}^{\mathrm{T}}\big[\delta\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \mathcal{M}\big[\psi\big]\Big), \tag{8.42}$$

$$\lambda(t_k) = 0, \tag{8.43}$$

$$\alpha = \alpha_0 + C_{\alpha\alpha}\int_{t_0}^{t_k}\frac{\partial G}{\partial\alpha}\lambda\,dt. \tag{8.44}$$

These equations define the Euler–Lagrange equations for the weak constraint problem. They constitute a *coupled two point boundary value problem in time* for ψ and λ. The forward model is forced by a term representing model errors while the backward model is forced by *impulses* at measurement locations. The model operator of the backward model is the adjoint of the tangent linear forward model.

8.2.3 Iteration in $\boldsymbol{\alpha}$

It is common to define an iteration in α as follows

$$\alpha_{l+1} = \alpha_l - \gamma\left(\alpha_l - \alpha_0 - C_{\alpha\alpha}\int_{t_0}^{t_k}\frac{\partial G}{\partial\alpha}\Big|_{\substack{\psi_l\\ \alpha_l}}\lambda_l\,dt\right). \tag{8.45}$$

Here, the expression in the parentheses is just the gradient of the penalty function with respect to α, and γ is a step length. Thus, the iteration (8.45) is just the gradient descent method.

8.2.4 Strong constraint problem

A majority of previous works on parameter estimation solve a simpler version of the variational problem defined by (8.20) or (8.25). The parameter is still iterated as in (8.45), but an additional common simplification is to assume that the dynamical model has zero model errors, i.e. the prior for the model error covariance C_{qq} is set to zero. This corresponds to an infinite weight on the dynamical model which then must be satisfied exactly. From the Euler–Lagrange equations (8.40–8.43), it is seen that this eliminates the coupling of the dynamical model to the adjoint variable λ, although the initial condition still depends on λ. The so-called adjoint method solves this strong constraint problem by iteration of the initial conditions, using an equation similar to

$$\psi_{l+1}(t_0) = \psi_l(t_0) - \gamma\Big(\psi_l(t_0) - \Psi_0 + C_{aa}\lambda_l(t_0)\Big), \tag{8.46}$$

where the step length γ, may differ from the one used in (8.45). One can also choose to iterate both (8.45) and (8.46) simultaneously, or use an outer iteration of (8.45) and inner iteration for (8.46).

A further simplification is to assume that the initial conditions also are perfect, i.e. $C_{aa} \equiv 0$. This is equivalent to introducing an infinite weight on the term for the initial conditions in (8.20) and it will be exactly satisfied. This additional simplification completely decouples the dynamical model from the adjoint variable. The solution is then an exact model trajectory given the estimated parameter α. This is a commonly used form for the parameter estimation problem and it corresponds to minimizing a cost function containing the data misfit term and the prior term for the parameters. It is efficiently solved using the adjoint method and iterating the parameter, i.e. solve (8.40–8.44) with C_{qq} and C_{aa} set to zero.

The Euler–Lagrange equations for the strong constraint problem is most commonly derived from a Lagrangian function where the model and initial conditions are included using Lagrangian multipliers, i.e.

$$\begin{aligned}\mathcal{L}[\alpha,\lambda,\mu] &= (\alpha-\alpha_0)W_{\alpha\alpha}(\alpha-\alpha_0)\\ &+ \big(\psi(t_0)-\Psi_0\big)\mu\\ &+ \Big(\boldsymbol{d}-\mathcal{M}\big[\psi\big]\Big)^{\mathrm{T}}\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d}-\mathcal{M}\big[\psi\big]\Big)\\ &+ \int_{t_0}^{t_k}\left(\frac{\partial\psi}{\partial t}-G(\psi,\alpha)\right)\lambda\,dt.\end{aligned} \tag{8.47}$$

Variation with respect to μ returns the initial condition while variation with respect to λ returns the model. The variation with respect to α returns the Euler–Lagrange equations for the strong constraint problem as found above, i.e. (8.40–8.44) with C_{qq} and C_{aa} equal to zero. Thus, the Euler–Lagrange equations are decoupled and a solution can be found for α if the iteration (8.45) converges. This approach is normally named the adjoint method or 4DVAR method for parameter estimation.

An alternative approach for solving the strong constraint problem can be derived as follows. Evaluating the variation of (8.47) with respect to α when realizing that ψ is a function of α gives

$$\begin{aligned}\frac{\delta\mathcal{L}}{2} &= \delta\alpha W_{\alpha\alpha}(\alpha-\alpha_0)\\ &+ \delta\alpha\psi_\alpha(t_0)\mu\\ &+ \delta\alpha\mathcal{M}^{\mathrm{T}}\big[\psi_\alpha\big]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d}-\mathcal{M}\big[\psi\big]\Big)\\ &+ \delta\alpha\int_{t_0}^{t_k}\left(\frac{\partial\psi_\alpha}{\partial t}-\frac{\partial G}{\partial\psi}\psi_\alpha-\frac{\partial G}{\partial\alpha}\right)\lambda dt\\ &+ \mathcal{O}\big(\delta\alpha^2\big),\end{aligned} \tag{8.48}$$

where we have used that $\delta\alpha$ is independent of time and that the measurement operator is linear. Since in addition λ and μ are arbitrary multipliers, we must

have

$$\frac{\partial \psi}{\partial t} = G(\psi, \alpha), \tag{8.49}$$

$$\psi(t_0) = \Psi_0, \tag{8.50}$$

$$\frac{\partial \psi_\alpha}{\partial t} = \frac{\partial G}{\partial \psi}\psi_\alpha - \frac{\partial G}{\partial \alpha}, \tag{8.51}$$

$$\psi_\alpha(t_0) = 0, \tag{8.52}$$

$$\alpha = \alpha_0 + C_{\alpha\alpha}\mathcal{M}^{\mathrm{T}}\left[\psi_\alpha\right]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \mathcal{M}\big[\psi\big]\Big). \tag{8.53}$$

Thus, we have derived a system of equations which consists of the original dynamical model with initial condition and an equation and initial condition for the sensitivity of ψ with respect to α, i.e. ψ_α. An equation for α includes the first guess value and an update term which includes the impact of measurements. It may be convenient to define an iteration in α as

$$\alpha_{l+1} = \alpha_l - \gamma\Bigg(\alpha_l - \alpha_0 - C_{\alpha\alpha}\mathcal{M}^{\mathrm{T}}\left[\psi_{\alpha l}\right]\boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d} - \mathcal{M}\big[\psi_l\big]\Big)\Bigg). \tag{8.54}$$

For each iteration in α we can solve the system (8.49–8.52) by forward integrations. There is no adjoint equation or backward integration involved. The forward models (8.49) and (8.51) should be integrated in parallel since the tangent linear operator in (8.51) is evaluated at the current estimate of the solution ψ. Note that the size of ψ_α, and the cost of solving (8.51), is proportional to the number of parameters included. In this example, we only have a single parameter and ψ_α becomes a scalar. Thus, with a low number of parameters this may be a more efficient approach than the adjoint method for solving the strong constraint parameter estimation problem. On the other hand, the adjoint method finds the gradient from one forward and one backward integration, independent of the number of parameters involved, but requires the model solution as a function of space and time to be stored and used for evaluation of the adjoint model operator.

8.3 Parameter estimation in the Ekman flow model

In Sect. 5.3 the representer method was used to solve the generalized inverse problem for an Ekman flow model. The discussion was taken from *Eknes and Evensen* (1997) which also considered the estimation of poorly known parameters in the model. In particular the first guesses of the wind drag and the vertical diffusion coefficient, c_{d_0} and $A_0(z)$, were allowed to contain errors, i.e.

$$c_d = c_{d_0} + p_{c_d}, \tag{8.55}$$

$$A(z) = A_0(z) + p_A(z), \tag{8.56}$$

where p_{c_d} and $p_A(z)$ are the unknown error terms. Thus a combined state estimation and parameter estimation problem was formulated and the penalty function for the state estimation problem given in (5.75) was extended to include two terms which penalize the deviation of estimated parameters from the first guess. Using the notation from Sect. 5.3, the generalized inverse for the combined parameter and state estimation problem was formulated as

$$\begin{aligned}\mathcal{J}[\boldsymbol{u}, c_d, A] = &\boldsymbol{q}^T \bullet \boldsymbol{W}_{qq} \bullet \boldsymbol{q} \\ &+ \boldsymbol{a}^T \circ \boldsymbol{W}_{aa} \circ \boldsymbol{a} \\ &+ \boldsymbol{b}_0^T * \boldsymbol{W}_{b_0 b_0} * \boldsymbol{b}_0 \\ &+ \boldsymbol{b}_H^T * \boldsymbol{W}_{b_H b_H} * \boldsymbol{b}_H \\ &+ p_A \circ W_{AA} \circ p_A \\ &+ p_{c_d} W_{c_d c_d} p_{c_d} \\ &+ \boldsymbol{\epsilon}^T \boldsymbol{W}_{\epsilon\epsilon} \boldsymbol{\epsilon},\end{aligned} \tag{8.57}$$

where the weight $W_{c_d c_d}$ is the inverse of the error variance $C_{c_d c_d}$ of p_{c_d}, and W_{AA} is the inverse of the error covariance C_{AA} of p_A. Since the wind drag coefficient and the vertical diffusion are allowed to contain errors, the variation of the penalty function with respect to these parameters must also be taken. This results in the additional equations

$$c_d = c_{d_0} + C_{c_d c_d} \int_0^T \boldsymbol{\lambda}^T(0,t)\, \boldsymbol{u}_a dt, \tag{8.58}$$

$$A = A_0 - C_{AA} \bullet \frac{\partial \boldsymbol{\lambda}^{\mathrm{T}}}{\partial z} \frac{\partial \boldsymbol{u}}{\partial z}, \tag{8.59}$$

for the wind drag coefficient and the diffusion parameter. The addition of the two equations (8.58) and (8.59) to the system of Euler–Lagrange equations (5.77) to (5.83) makes the overall inverse problem nonlinear.

In Sect. 5.3 it was illustrated how the representer method could be used to solve exactly the Euler–Lagrange equations for the weak constraint inverse problem when $A(z)$ and c_d are known. When the parameters are allowed to contain errors, the inverse problem becomes nonlinear and therefore an iteration was used for $A(z)$ and c_d in (8.58) and (8.59). In each iteration, the representer technique was used to solve for the corresponding inverse estimate.

The equations (8.58) and (8.59) were iterated using a gradient descent method, i.e.

$$c_d^{l+1} = c_d^l - \gamma \left(c_d^l - c_{d_0} - C_{c_d c_d} \int_{t_0}^{t_k} (\boldsymbol{\lambda}^l)^{\mathrm{T}} \sqrt{u_a^2 + v_a^2} \boldsymbol{u}_a \, dt \right), \tag{8.60}$$

$$A^{l+1}(z) = A^l(z) - \gamma \left(A^l(z) - A_0(z) + C_{AA} \bullet \left(\frac{\partial \boldsymbol{u}^l}{\partial z} \right)^{\mathrm{T}} \frac{\partial \boldsymbol{\lambda}^l}{\partial z} \right). \tag{8.61}$$

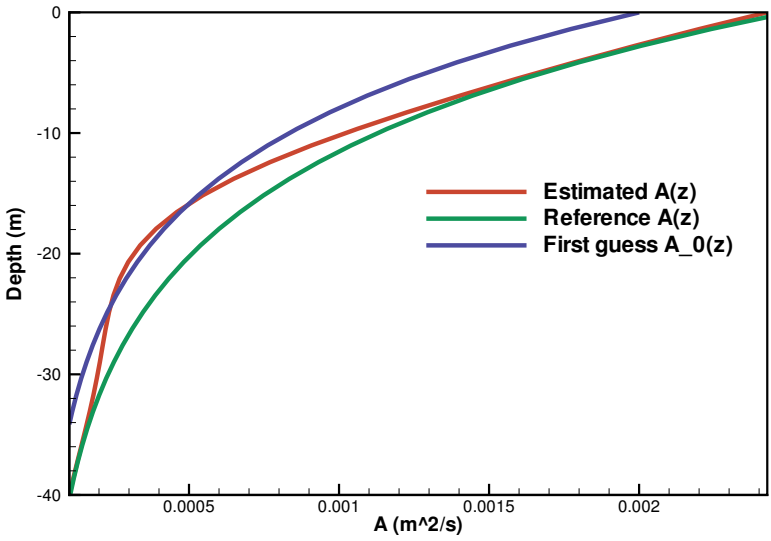

Fig. 8.1. The estimation of the eddy viscosity profile $A(z)$, from the identical twin experiment. Reproduced from *Eknes and Evensen* (1997)

Note that the expressions inside the parantheses are the actual gradients used in the gradient descent algorithm. The constant γ determines the length of the steps in the direction of the gradient in the parameter space and has an important impact on the convergence. The equations (8.60) and (8.61) are now iterated to generate new guesses c_d^{l+1} and A^{l+1}, which are used to solve for $\boldsymbol{u}^{l+1}$ and $\boldsymbol{\lambda}^{l+1}$ using the representer method.

The identical twin experiment from *Eknes and Evensen* (1997) resulted in estimates of the parameters as shown in Figs. 8.1 and 8.2. For the statistical priors used in this experiment we refer to *Eknes and Evensen* (1997). The estimation of the diffusion parameter $A(z)$ is illustrated in Fig. 8.1 where the first-guess $A_0(z)$ and the reference $A(z)$ are shown together with the estimate of $A(z)$. The weak signal below the Ekman layer makes it difficult to correct an erroneous first-guess of the diffusion parameter in the deep ocean. Note also that the estimate of $A(z)$ does not coincide with the reference diffusion parameter but is located somewhere in between the first-guess $A_0(z)$ and the exact $A(z)$ at most of the depths. At some depths the estimate is located to the left of both the first guess and the reference diffusion. This is not unexpected for this nonlinear problem where the minimum of the penalty function determines both the inverse solution and estimated parameters simultaneously, and these are mutually dependent. The estimation of the wind drag coefficient C_d is shown in Fig. 8.2. It converges to a value somewhere in between the first-guess and the reference value.

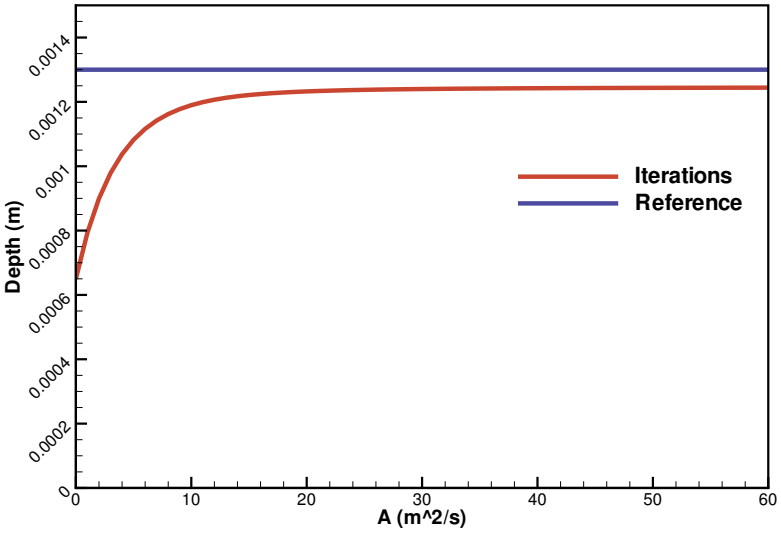

Fig. 8.2. The estimation of the wind-drag coefficient c_d, from the identical twin experiment. The number of iterations is given along the x axis. Reproduced from *Eknes and Evensen* (1997)

It was pointed out by *Bennett* (1992) and *Yu and O'Brien* (1991) that without a smoothing regularization on the diffusion coefficient $A(z)$, it is not clear if there is any difference in varying $A(0)$ or c_d in the surface condition (5.79), since $A(z)$ may then become discontinuous. However here, the non-diagonal weight will ensure a smooth $A(z)$. It is therefore expected that a vertical profile of the solution for $\boldsymbol{u}$, which is consistent with the measurements, will determine the profile for $A(z)$, while c_d will adjust to provide the correct surface forcing.

This illustration of a methodology for solving the combined state and parameter estimation problem considered a fairly simple dynamical model and it was shown that a better solution could be obtained both for the state and the parameters. The same methodology has later been examined by *Muccino and Bennett* (2001) with a nonlinear dynamical model (Korteweg-de Vries equation) containing several parameters.

They also defined an outer iteration of the parameter. Since the model dynamics is nonlinear, a sequence of linear inverse problems is next defined for each iterate of the parameter and each of these is solved using the representer method. It was found that the parameter estimation skill was limited due to the nonlinear and dispersive properties of the dynamical system. Further, they observed problems with convergence of the parameters, in particular when several parameters were estimated simultaneously. They had a fairly negative conclusion and suggested that one should rather admit errors in the

dynamical equations than fiddle with the empirical formulas in the dynamical equations.

8.4 Summary

In this chapter we have derived the generalized inverse formulation for the combined state and parameter estimation problem. The starting point was the Bayes' theorem on the form (7.13) where all the data are introduced simultaneously together with an assumption of Gaussian priors. This led to the generalized inverse formulation in the form of a penalty function which is quadratic in the errors. From the generalized inverse, we derived the Euler–Lagrange equations which, in the parameter estimation case, pose a nonlinear problem even if the dynamical model is linear. We showed how we could resolve this nonlinearity by defining an iteration for the parameters to be estimated and then use the representer method to solve for the state for each iterate of the parameters.

Note that it is also possible to define a sequence of variational problems for each of (7.15–7.18) and the solution of one variational problem would then become the prior for the next. This could be a sensible approach except that the variational methods, such as the representer and adjoint methods, do not easily provide statistical information about the errors of the estimate, which is needed when the estimate is used as a prior for the next inversion. On the other hand, the genetic algorithms result in a sample of the posterior distribution, which might be used as the prior for the next inversion.

9

Ensemble methods

The focus in this Chapter will be on three methods, the Ensemble Smoother (ES), the Ensemble Kalman Smoother (EnKS) and the Ensemble Kalman Filter (EnKF). They belong to a general class of so-called particle methods which use a Monte Carlo or ensemble representation for the pdfs, an ensemble integration using stochastic models to model the time evolution of the pdfs, and different schemes for conditioning the predicted pdf given the observations.

Specific for the ES, EnKS and EnKF is the introduction of an assumption of a Gaussian pdf for the model prediction. This makes it possible to represent the pdf for the model prediction using only the mean and covariance of the pdf and a linear update equation can then be used. The discussion below will also allow for the estimation of poorly known model parameters.

9.1 Introductory remarks

Going back to the original Bayes' problem formulated as (7.12) or (7.13), we now assume that all the prior densities are known. The joint pdf for the model prediction until t_k is given by (7.10).

In Sect. 4.3 we derived the EnKF on the assumption that errors statistics could be described by error covariances represented by an ensemble of model states. The same approach can also be used when working with general pdfs.

Given a large sample of realizations for each of the prior pdfs, the joint pdf (7.10) can be evaluated by integration of each individual realization forward in time using stochastic model equations. The prior pdfs do not need to be Gaussian distributed. The densities can be represented to a desired accuracy by using a sufficiently large number N, of realizations for each of them.

The dynamical model equation (7.1) can be rewritten as a stochastic model, similar to (4.33), as

$$d\boldsymbol{\psi} = \boldsymbol{G}(\boldsymbol{\psi}, \boldsymbol{\alpha})\, dt + \boldsymbol{h}(\boldsymbol{\psi}, \boldsymbol{\alpha})\, d\boldsymbol{q}, \tag{9.1}$$

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_9,

where we have now introduced the poorly known parameters $\boldsymbol{\alpha}$. Thus, to a small increment in time dt, is associated a random increment $d\boldsymbol{q}$, representing the model error, leading to an increment in the model state $d\boldsymbol{\psi}$. The model errors are described by the samples of $f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1}, \boldsymbol{\alpha})$.

As in Sect. 4.3 it is possible to derive Kolmogorov's equation for the evolution of the pdf in time. The use of the stochastic model (9.1) to integrate an ensemble of model states forward in time is equivalent to solving Kolmogorov's equation using a Monte Carlo method. It turns out that this is the most efficient way to solve this equation for high dimensional and nonlinear problems where analytical solutions don't exist and direct numerical integration becomes impossible due to the numerical cost. Further, using the Monte Carlo approach there are no approximations other than the use of a limited ensemble size. Thus, an ensemble representation of the prior pdfs and a stochastic ensemble integration results in a consistent ensemble representation of the joint pdf for the model evolution.

Combining the joint pdf for the model evolution (7.10) with the Bayesian update equation (7.12) we get

$$\begin{aligned} &f(\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b|\boldsymbol{d}) \\ &\qquad \propto f(\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b) \prod_{j=1}^{m} f(\boldsymbol{d}_j|\boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}). \end{aligned} \tag{9.2}$$

The computation of the Bayesian analysis (9.2) is complicated for arbitrary distributions and high dimensions. However, the use of importance sampling makes it possible to evaluate the mean and covariance of the posterior distribution in (9.2).

We adopt for simplicity a notation where $\boldsymbol{\psi}$ contains the model solution at all time instants and also includes the initial and boundary data, and the parameters. The expected value of a function of $h(\boldsymbol{\psi})$, given the posterior distribution in (9.2), then becomes

$$\begin{aligned} E[h(\boldsymbol{\psi})] &= \int h(\boldsymbol{\psi}) f(\boldsymbol{\psi}|\boldsymbol{d}) d\boldsymbol{\psi} \\ &= \frac{\int h(\boldsymbol{\psi}) f(\boldsymbol{d}|\boldsymbol{\psi}) f(\boldsymbol{\psi}) d\boldsymbol{\psi}}{f(\boldsymbol{d})} \\ &= \frac{\int h(\boldsymbol{\psi}) f(\boldsymbol{d}|\boldsymbol{\psi}) f(\boldsymbol{\psi}) d\boldsymbol{\psi}}{\int f(\boldsymbol{d}|\boldsymbol{\psi}) f(\boldsymbol{\psi}) d\boldsymbol{\psi}} \\ &\approx \frac{\sum_i h(\boldsymbol{\psi}_i) f(\boldsymbol{d}|\boldsymbol{\psi}_i)}{\sum_i f(\boldsymbol{d}|\boldsymbol{\psi}_i)}. \end{aligned} \tag{9.3}$$

The summation is over the ensemble members. Thus, we can evaluate expected values of functions of $\boldsymbol{\psi}$ using the ensemble representation for the model prediction. Using $h(\boldsymbol{\psi}) = \boldsymbol{\psi}$ results in the variance minimizing estimator which is the expected value for $\boldsymbol{\psi}$ of the posterior distribution in (9.2).

Further, defining $s(\boldsymbol{\psi}) = (\boldsymbol{\psi} - E[\boldsymbol{\psi}])(\boldsymbol{\psi} - E[\boldsymbol{\psi}])^{\mathrm{T}}$ results in the posterior error covariance.

In *van Leeuwen and Evensen* (1996) the formula (9.3) was examined for solving the inverse problem using a nonlinear ocean circulation model with 6400 unknowns. It was found that the weights for most of the ensemble members became negligible and only very few ensemble members contributed in the summation. Thus, it was concluded that a very large ensemble size would be needed to properly represent the full pdf for the posterior.

Another class of methods named particle filters solves the full Bayesian update equation using importance resampling techniques. They introduce a resampling step, which results in a new ensemble having the correct posterior distribution. Some resampling schemes are discussed in *Chen et al.* (2004) and in several of the articles in *Doucet et al.* (2001). Common for these is that they use schemes where ensemble members with low weights are rejected while multiple copies are generated of the ensemble members with large weights. This helps reducing the effect of degeneracy resulting from using an ensemble where only a few ensemble members have significant weights. There are several applications where these methods have worked well for low-dimensional systems, but common for these is the requirement of a very large number of ensemble members, a need for resampling of the posterior joint pdf, and extremely high computational cost for high-dimensional models.

Some other implementations of nonlinear filters have been based on either a kernel approximation, *Miller et al.* (1999), *Anderson and Anderson* (1999a) and *Miller and Ehret* (2002); or a particle interpretation, *Pham* (2001), *van Leeuwen* (2003) and *Chen et al.* (2004), although more research is needed before these can be claimed to be practical for realistic high dimensional systems. See also the Sequential Monte Carlo Methods Particle Filtering webpage, `www-sigproc.eng.cam.ac.uk/smc`, for more information.

9.2 Linear ensemble analysis update

For the case with a linear dynamical model and Gaussian prior pdfs, the pdf for the model prediction in (7.10) will also be Gaussian. The variance minimizing analysis in this case also equals the MLH estimate.

We can evaluate the mean of the ensemble prediction $\overline{\boldsymbol{\psi}^{\mathrm{f}}}(\boldsymbol{x}, t)$, as a function of space and time, and its associated ensemble error covariance $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1, t_1, \boldsymbol{x}_2, t_2)$. We also have the measurements $\boldsymbol{d}$, with error covariance $\boldsymbol{C}_{\epsilon\epsilon}$. The linear variance minimizing analysis or MLH estimate is then, from (9.2), using (8.8), defined by the minimum of

$$\begin{aligned} \mathcal{J}[\boldsymbol{\psi}^{\mathrm{a}}] = &\left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right)^{\mathrm{T}} \bullet \left(\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\right)^{-1} \bullet \left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\right) \\ &+ \left(\boldsymbol{d} - \mathcal{M}[\boldsymbol{\psi}^{\mathrm{a}}]\right)^{\mathrm{T}} \boldsymbol{C}^{-1}_{\epsilon\epsilon} \left(\boldsymbol{d} - \mathcal{M}[\boldsymbol{\psi}^{\mathrm{a}}]\right). \end{aligned} \tag{9.4}$$

This defines a Gauss-Markov interpolation in space and time and has the well-known minimizing solution and associated error covariance estimate given by

$$\boldsymbol{\psi}^{\mathrm{a}} = \boldsymbol{\psi}^{\mathrm{f}} + \mathcal{M}^{\mathrm{T}}\big[\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\big]\Big(\mathcal{M}^{\mathrm{T}}\big[\mathcal{M}\big[\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\big]\big] + \boldsymbol{C}_{\epsilon\epsilon}\Big)^{-1}\Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{\psi}^{\mathrm{f}}\big]\Big), \tag{9.5}$$

$$\boldsymbol{C}^{\mathrm{a}}_{\psi\psi} = \boldsymbol{C}^{\mathrm{f}}_{\psi\psi} - \mathcal{M}^{\mathrm{T}}\big[\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\big]\Big(\mathcal{M}^{\mathrm{T}}\big[\mathcal{M}\big[\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\big]\big] + \boldsymbol{C}_{\epsilon\epsilon}\Big)^{-1}\mathcal{M}\big[\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\big]. \tag{9.6}$$

These equations should be compared with the analysis equations derived in Chap. 3 for the time-independent problem, in particular (3.26) which defines the problem and (3.39), (3.46) and (3.54), for the solution and error estimate. The derivation of (9.5) and (9.6) is identical to the one given for the time independent case.

From these equations it is also seen that if we define the representer functions as the measurements of the space-time error covariance for the model prediction

$$\boldsymbol{r} = \mathcal{M}\big[\boldsymbol{C}^{\mathrm{f}}_{\psi\psi}\big], \tag{9.7}$$

then the analysis equations (9.5) and (9.6) becomes just

$$\boldsymbol{\psi}^{\mathrm{a}} = \boldsymbol{\psi}^{\mathrm{f}} + \boldsymbol{r}^{\mathrm{T}}\Big(\mathcal{M}^{\mathrm{T}}\big[\boldsymbol{r}\big] + \boldsymbol{C}_{\epsilon\epsilon}\Big)^{-1}\Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{\psi}^{\mathrm{f}}\big]\Big), \tag{9.8}$$

$$\boldsymbol{C}^{\mathrm{a}}_{\psi\psi} = \boldsymbol{C}^{\mathrm{f}}_{\psi\psi} - \boldsymbol{r}^{\mathrm{T}}\Big(\mathcal{M}^{\mathrm{T}}\big[\boldsymbol{r}\big] + \boldsymbol{C}_{\epsilon\epsilon}\Big)^{-1}\boldsymbol{r}. \tag{9.9}$$

Comparison of (9.8) with (5.60) illustrates the similarity between the representer method and Gauss-Markov interpolation in space and time. A more elaborate discussion is given by *McIntosh* (1990) and *Bennett* (1992, 2002). In *Bennett* (1992) is is actually shown that the representers equal measurements of the space time error covariance matrix. Thus, for linear dynamics and Gaussian priors, the representer method and (9.5) will provide the same result in the limit of an infinite ensemble size.

For a nonlinear dynamical model, the pdf for the model evolution will become non-Gaussian even if the prior pdfs are Gaussian. In this case (9.5) and (9.6) will provide only an approximate solution. Still these formulas may provide a useful solution if the prior pdf is nearly Gaussian. It should again be pointed out that only the update is linear and the updated ensemble will inherit some of the non-Gaussian contributions contained in the prior ensemble. Thus, the method is doing more than just resampling a Gaussian posterior pdf. The actual ensemble implementation of (9.5) is described below and results in the Ensemble Smoother method.

9.3 Ensemble representation of error statistics

The ensemble covariance is defined as

$$\boldsymbol{C}_{\psi\psi} = \overline{\left(\boldsymbol{\psi} - \overline{\boldsymbol{\psi}}\right)\left(\boldsymbol{\psi} - \overline{\boldsymbol{\psi}}\right)^{\mathrm{T}}}. \tag{9.10}$$

The ensemble mean $\overline{\boldsymbol{\psi}}$, is regarded as the best-guess estimate, while the ensemble spread defines the error variance. The covariance is determined by the smoothness of the ensemble members. A covariance matrix can always be represented by an ensemble of model states and this representation is not unique.

As in *Evensen* (2003) we have defined the matrix holding the ensemble members $\boldsymbol{\psi}(\boldsymbol{x}, t_i) \in \Re^{n_\psi}$, at time t_i, where n_ψ is the number of variables in the state vector. Further, we augment the state vector with the poorly known parameters $\boldsymbol{\alpha}(\boldsymbol{x}) \in \Re^{n_\alpha}$, where n_α is the number of parameters in $\boldsymbol{\alpha}$, and write the matrix $\boldsymbol{A}(\boldsymbol{x}, t_i) \in \Re^{n \times N}$, with $n = n_\psi + n_\alpha$, holding the N ensemble members of $\boldsymbol{\psi}$ and $\boldsymbol{\alpha}$ at time t_i, as

$$\boldsymbol{A}_i = \boldsymbol{A}(\boldsymbol{x}, t_i) = \begin{pmatrix} \boldsymbol{\psi}^1(\boldsymbol{x}, t_i) & \boldsymbol{\psi}^2(\boldsymbol{x}, t_i) & \dots & \boldsymbol{\psi}^N(\boldsymbol{x}, t_i) \\ \boldsymbol{\alpha}^1(\boldsymbol{x}, t_i) & \boldsymbol{\alpha}^2(\boldsymbol{x}, t_i) & \dots & \boldsymbol{\alpha}^N(\boldsymbol{x}, t_i) \end{pmatrix}. \tag{9.11}$$

Note that we have used a time index on $\boldsymbol{\alpha}$ even though the parameters are supposed to be constant in time. This is to be able to distinguish between the estimates of $\boldsymbol{\alpha}$ at different times, which in the EnKF and EnKS change at each update with measurements.

The ensemble mean is stored in each column of $\overline{\boldsymbol{A}}(\boldsymbol{x}, t_i)$ which can be defined as

$$\overline{\boldsymbol{A}}(\boldsymbol{x}, t_i) = \boldsymbol{A}(\boldsymbol{x}, t_i)\mathbf{1}_N, \tag{9.12}$$

where $\mathbf{1}_N \in \Re^{N \times N}$ is the matrix where each element is equal to $1/N$. We can then define the ensemble perturbation matrix as

$$\boldsymbol{A}'(\boldsymbol{x}, t_i) = \boldsymbol{A}(\boldsymbol{x}, t_i) - \overline{\boldsymbol{A}}(\boldsymbol{x}, t_i) = \boldsymbol{A}(\boldsymbol{x}, t_i)(\boldsymbol{I} - \mathbf{1}_N). \tag{9.13}$$

The ensemble covariances $\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2, t_i) \in \Re^{n \times n}$, can be defined as

$$\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2, t_i) = \frac{\boldsymbol{A}'(\boldsymbol{x}_1, t_i)\big(\boldsymbol{A}'(\boldsymbol{x}_2, t_i)\big)^{\mathrm{T}}}{N-1}. \tag{9.14}$$

Now, given the ensemble matrices for the different instants in time $\boldsymbol{A}(\boldsymbol{x}, t_{i'})$, for $i' = 1, \dots, i$, we can define the ensemble matrix for the joint state from t_0 to t_i as

$$\widetilde{\boldsymbol{A}}_i = \begin{pmatrix} \boldsymbol{A}(\boldsymbol{x}, t_0) \\ \vdots \\ \boldsymbol{A}(\boldsymbol{x}, t_i) \end{pmatrix}. \tag{9.15}$$

The space-time ensemble covariance between the model states at two arbitrary times t_1 and t_2 then becomes

$$\widetilde{\boldsymbol{C}}^{\mathrm{e}}_{\psi\psi}(\boldsymbol{x}_1, t_1, \boldsymbol{x}_2, t_2) = \frac{\widetilde{\boldsymbol{A}}'_i(\boldsymbol{x}_1, t_1)\big(\widetilde{\boldsymbol{A}}'_i(\boldsymbol{x}_2, t_2)\big)^{\mathrm{T}}}{N-1}. \tag{9.16}$$

9.4 Ensemble representation for measurements

At the data time $t_{i(j)}$, we have given a vector of measurements $\boldsymbol{d}_j \in \Re^{m_j}$, with m_j being the number of measurements at this time. We can define the N vectors of perturbed measurements as

$$\boldsymbol{d}_j^l = \boldsymbol{d}_j + \boldsymbol{\epsilon}_j^l, \qquad l = 1, \ldots, N, \tag{9.17}$$

which can be stored in the columns of a matrix

$$\boldsymbol{D}_j = \left(\boldsymbol{d}_j^1, \boldsymbol{d}_j^2, \ldots, \boldsymbol{d}_j^N\right) \in \Re^{m_j \times N}. \tag{9.18}$$

The ensemble of measurement perturbations, with mean equal to zero, can be stored in the matrix

$$\boldsymbol{E}_j = \left(\boldsymbol{\epsilon}_j^1, \boldsymbol{\epsilon}_j^2, \ldots, \boldsymbol{\epsilon}_j^N\right) \in \Re^{m_j \times N}, \tag{9.19}$$

from which we can construct the ensemble representation of the measurement error covariance matrix

$$\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}}(t_{i(j)}) = \frac{\boldsymbol{E}_j \boldsymbol{E}_j^{\mathrm{T}}}{N-1}. \tag{9.20}$$

9.5 Ensemble Smoother (ES)

The ES was proposed by *van Leeuwen and Evensen* (1996) as a linear variance minimizing smoother analysis. It computes an approximate update of (9.2) using the linear update (9.5). In fact, it can be shown that if each individual ensemble member is updated independently using (9.5), using the perturbed observations from (9.18), then the updated ensemble will have the correct mean and covariance as defined by the analysis (9.5) and (9.6). It was shown in *Burgers et al.* (1998) that the perturbation of measurements is required to obtain the correct covariance.

The linear ES analysis equation then becomes for $\widetilde{\boldsymbol{A}}_k^{\mathrm{a}}$, as defined in (9.15),

$$\widetilde{\boldsymbol{A}}_k^{\mathrm{a}} = \widetilde{\boldsymbol{A}}_k + \mathcal{M}^{\mathrm{T}}\left[\widetilde{\boldsymbol{C}}_{\psi\psi}^{\mathrm{e}}\right]\left(\mathcal{M}^{\mathrm{T}}\left[\mathcal{M}\left[\widetilde{\boldsymbol{C}}_{\psi\psi}^{\mathrm{e}}\right]\right] + \boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}}\right)^{-1}\left(\boldsymbol{D} - \mathcal{M}\left[\widetilde{\boldsymbol{A}}_k\right]\right), \tag{9.21}$$

where we have used

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_1 \\ \vdots \\ \boldsymbol{D}_m \end{pmatrix}, \qquad \mathcal{M} = \begin{pmatrix} \mathcal{M}_1 \\ \vdots \\ \mathcal{M}_m \end{pmatrix}, \tag{9.22}$$

and

$$\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}} = \begin{pmatrix} \boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}}(t_{i(1)}) & & \\ & \ddots & \\ & & \boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}}(t_{i(m)}) \end{pmatrix}. \tag{9.23}$$

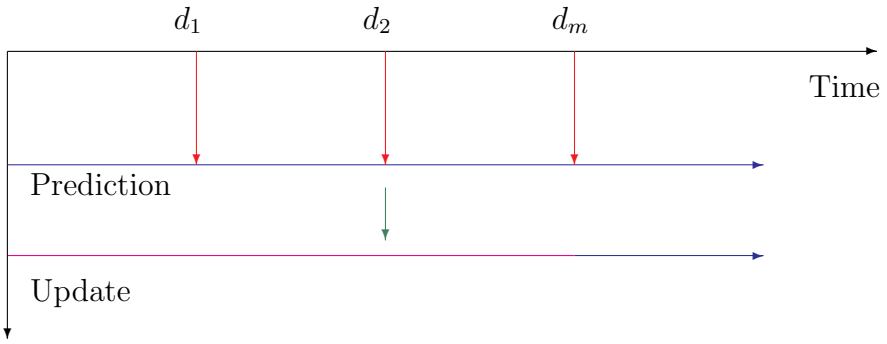

Fig. 9.1. Illustration of the update procedure used in the ES. The horizontal axis is time and the measurements are indicated at regular intervals. The vertical axis indicates the number of updates with measurements. The blue arrows represent the forward ensemble integration, while the red arrows are the introduction of measurements

The total number of measurements is $M = \sum_{j=1}^{m} m_j$. Thus, we have $\boldsymbol{D} \in \Re^{M\times N}$, $\mathcal{M} \in \Re^{M}$, and $\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}} \in \Re^{M\times M}$.

We now define the ensemble of innovation vectors as

$$\boldsymbol{D}' = \boldsymbol{D} - \mathcal{M}\big[\widetilde{\boldsymbol{A}}_k\big], \tag{9.24}$$

the measurements of the ensemble perturbations $\boldsymbol{S} \in \Re^{M\times N}$, as

$$\boldsymbol{S} = \mathcal{M}\big[\widetilde{\boldsymbol{A}}_k'\big], \tag{9.25}$$

and the matrix $\boldsymbol{C} \in \Re^{M\times M}$ as

$$\boldsymbol{C} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}}. \tag{9.26}$$

Using (9.24–9.26) together with the definitions of the ensemble error covariance matrices in (9.16) and (9.20), the analysis (9.21) can be expressed as

$$\begin{aligned}
\widetilde{\boldsymbol{A}}_k^{\mathrm{a}} &= \widetilde{\boldsymbol{A}}_k + \widetilde{\boldsymbol{A}}_k'\mathcal{M}^{\mathrm{T}}\big[\widetilde{\boldsymbol{A}}_k'\big]\left(\mathcal{M}\big[\widetilde{\boldsymbol{A}}_k'\big]\mathcal{M}^{\mathrm{T}}\big[\widetilde{\boldsymbol{A}}_k'\big] + (N-1)\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}}\right)^{-1}\boldsymbol{D}' \\
&= \widetilde{\boldsymbol{A}}_k + \widetilde{\boldsymbol{A}}_k(\boldsymbol{I}-\boldsymbol{1}_N)\boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{D}' \\
&= \widetilde{\boldsymbol{A}}_k\left(\boldsymbol{I} + (\boldsymbol{I}-\boldsymbol{1}_N)\boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{D}'\right) \\
&= \widetilde{\boldsymbol{A}}_k\left(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{D}'\right) \\
&= \widetilde{\boldsymbol{A}}_k\boldsymbol{X},
\end{aligned} \tag{9.27}$$

where we have used (9.13) and $\boldsymbol{1}_N\boldsymbol{S}^{\mathrm{T}} \equiv \boldsymbol{0}$. Thus, the updated ensemble can be considered as a combination of the forecast ensemble members.

Equation (9.27) converges towards the exact solution of the Bayesian formulation with increasing ensemble size if the assumption of Gaussian statistics is true. This requires that all priors are Gaussian and that a linear model is used. In this linear case it will also converge towards the representer solution.

The representer solution and the ES solution will differ in the case with nonlinear dynamics. Using the ES we should be concerned about the validity of the Gaussian approximation and the required ensemble size. When using the representer method we need to consider the convergence of the iteration, the validity of the tangent linear approximation, and whether the modal trajectory is a good estimator. Further, the computation of the posterior errors is not straight forward in the representer method.

In *Evensen and van Leeuwen* (2000) it was illustrated that the ES may have problems with nonlinear dynamical models. The method was examined with the nonlinear Lorenz model where it turned out that the Gaussian approximation for the pdf of the model evolution was too crude.

9.6 Ensemble Kalman Smoother (EnKS)

We will now present an alternative approach, by *Evensen and van Leeuwen* (2000), which solves the recursion (7.15–7.18) using an ensemble representation for the error statistics.

In (7.15), the joint pdf for the model prediction until $t_{i(1)}$ is

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b) \propto f(\boldsymbol{\alpha})f(\boldsymbol{\psi}_0)f(\boldsymbol{\psi}_b)\prod_{i=1}^{i(1)} f(\boldsymbol{\psi}_i|\boldsymbol{\psi}_{i-1},\boldsymbol{\alpha}). \tag{9.28}$$

Similar to the procedure used in the ES, this joint pdf can be evaluated using a large ensemble of realizations for each of the prior pdfs and integrating these forward in time using the stochastic model equations.

The stochastic integration results in an ensemble representation of the joint pdf for the model solution $\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(1)}$, the initial condition $\boldsymbol{\psi}_0$, the boundary condition $\boldsymbol{\psi}_b$, and the poorly known parameters $\boldsymbol{\alpha}$.

The major problem is now the efficient computation of the joint pdf conditional on the measurements $\boldsymbol{d}_1$, given the ensemble representation of (9.28); i.e. we need to solve (7.15) rewritten as

$$f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b|\boldsymbol{d}_1) \propto f(\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha},\boldsymbol{\psi}_0,\boldsymbol{\psi}_b)f(\boldsymbol{d}_1|\boldsymbol{\psi}_{i(1)},\boldsymbol{\alpha}), \tag{9.29}$$

which gives the update based on the first set of measurements at $t_{i(1)}$.

The EnKS is similar to the ES, except that it processes the measurements sequentially in time. Starting from the initial ensemble stored in $\boldsymbol{A}_0$, a forward

stochastic integration of the ensemble until the first available data set, gives the ensemble prediction

$$\widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(1)} = \begin{pmatrix} \boldsymbol{A}_0 \\ \boldsymbol{A}^{\mathrm{f}}_1 \\ \vdots \\ \boldsymbol{A}^{\mathrm{f}}_{i(1)} \end{pmatrix}. \tag{9.30}$$

Using the ES update (9.27) with (9.30) using the first set of measurements $\boldsymbol{d}_1$, which solves (9.29) under the assumption of a Gaussian pdf for the predicted ensemble, we get

$$\begin{aligned} \widetilde{\boldsymbol{A}}^{\mathrm{a}}_{i(1)} &= \widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(1)} + \widetilde{\boldsymbol{A}}^{\mathrm{f}\prime}_{i(1)} \mathcal{M}_1^{\mathrm{T}}\big[\widetilde{\boldsymbol{A}}^{\mathrm{f}\prime}_{i(1)}\big] \\ &\qquad \times \Big(\mathcal{M}_1\big[\widetilde{\boldsymbol{A}}^{\mathrm{f}\prime}_{i(1)}\big]\mathcal{M}_1^{\mathrm{T}}\big[\widetilde{\boldsymbol{A}}^{\mathrm{f}\prime}_{i(1)}\big] + (N-1)\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}(t_{i(1)})\Big)^{-1} \boldsymbol{D}'_1 \\ &= \widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(1)} + \widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(1)}(\boldsymbol{I} - \boldsymbol{1}_N)\boldsymbol{S}_1^{\mathrm{T}}\boldsymbol{C}_1^{-1}\boldsymbol{D}'_1 \\ &= \widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(1)}\Big(\boldsymbol{I} + (\boldsymbol{I} - \boldsymbol{1}_N)\boldsymbol{S}_1^{\mathrm{T}}\boldsymbol{C}_1^{-1}\boldsymbol{D}'_1\Big) \\ &= \widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(1)}\Big(\boldsymbol{I} + \boldsymbol{S}_1^{\mathrm{T}}\boldsymbol{C}_1^{-1}\boldsymbol{D}'_1\Big) \\ &= \widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(1)}\boldsymbol{X}_1. \end{aligned} \tag{9.31}$$

Here we have used the definitions of innovation vectors,

$$\boldsymbol{D}'_j = \boldsymbol{D}_j - \mathcal{M}_j\big[\widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(j)}\big], \tag{9.32}$$

the measurements of the ensemble perturbations $\boldsymbol{S}_j \in \Re^{m_j \times N}$,

$$\boldsymbol{S}_j = \mathcal{M}_j\big[\widetilde{\boldsymbol{A}}^{\mathrm{f}\prime}_{i(j)}\big], \tag{9.33}$$

and the matrix $\boldsymbol{C}_j \in \Re^{m_j \times m_j}$,

$$\boldsymbol{C}_j = \boldsymbol{S}_j\boldsymbol{S}_j^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}(t_{i(j)}). \tag{9.34}$$

The update (9.31) is identical to the ES update in the case where the time interval covers $t \in [t_0, t_{i(1)}]$, and the data are all contained in $\boldsymbol{d}_1$. The EnKS provides an approximate ensemble representation for the joint pdf conditional on $\boldsymbol{d}_1$, in (9.29), and this serves as a prior for a continued ensemble integration until the next time when measurements are available, and then a new update is computed.

The general update equation for the measurements at the time $t_{i(j)}$, can be written

$$\begin{aligned} f(\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b | \boldsymbol{d}_1, \ldots, \boldsymbol{d}_j) \propto \\ f(\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}, \boldsymbol{\psi}_0, \boldsymbol{\psi}_b | \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{j-1}) f(\boldsymbol{d}_j | \boldsymbol{\psi}_{i(j)}, \boldsymbol{\alpha}). \end{aligned} \tag{9.35}$$

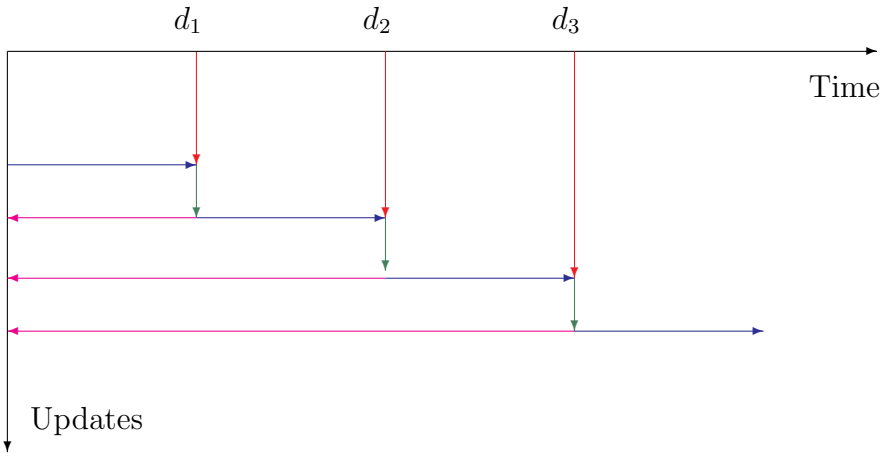

Fig. 9.2. Illustration of the update procedure used in the EnKS. The horizontal axis is time and the measurements are indicated at regular intervals. The vertical axis indicates the number of updates with measurements. The blue arrows represent the forward ensemble integration, the red arrows are the introduction of measurements, while the green arrows denote updates. Thus, the blue arrows indicate the EnKF solution as a function of time, which is updated every time measurements are available. The magenta arrows are the updates for the EnKS, which goes backward in time, and which is computed following the EnKF update every time measurements are available

Now, define the ensemble prediction matrix

$$\widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(j)} = \begin{pmatrix} \widetilde{\boldsymbol{A}}^{\mathrm{a}}_{i(j-1)} \\ \boldsymbol{A}^{\mathrm{f}}_{i(j-1)+1} \\ \vdots \\ \boldsymbol{A}^{\mathrm{f}}_{i(j)} \end{pmatrix}, \tag{9.36}$$

where the ensemble prediction $\boldsymbol{A}^{\mathrm{f}}_{i(j-1)+1}, \ldots, \boldsymbol{A}^{\mathrm{f}}_{i(j)}$ is obtained by ensemble integration starting from the final analyzed result in $\widetilde{\boldsymbol{A}}^{\mathrm{a}}_{i(j-1)}$. We can then compute the EnKS update based on (9.35), using the measurements at time $t_{i(j)}$ as,

$$\widetilde{\boldsymbol{A}}^{\mathrm{a}}_{i(j)} = \widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(j)} \boldsymbol{X}_j, \tag{9.37}$$

with $\boldsymbol{X}_j$ defined as

$$\boldsymbol{X}_j = \boldsymbol{I} + \boldsymbol{S}_j^{\mathrm{T}} \boldsymbol{C}_j^{-1} \boldsymbol{D}'_j. \tag{9.38}$$

Here the predicted ensemble $\widetilde{\boldsymbol{A}}^{\mathrm{f}}_{i(j)}$ has been updated from all previous measurements $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{j-1}$. The update from measurements at time $t_{i(j)}$ adds the incremental information included in the measurements at the time $t_{i(j)}$.

Further, the combination $\boldsymbol{X}_j$, is only dependent on the ensemble at the time $t_{i(j)}$, and then only at the measurement locations. Thus, the update can be characterized as a weakly nonlinear combination of the prior ensemble.

9.7 Ensemble Kalman Filter (EnKF)

The EnKF can be most easily characterized as a simplification of the EnKS where the analysis acts on the ensemble only at the measurement times. Thus, there is no information propagated backward in time like in the EnKS.

We now only consider the analysis step at time $t_{i(j)}$, and the analysis equation (9.37) is rewritten as

$$\boldsymbol{A}^{\mathrm{a}}_{i(j)} = \boldsymbol{A}^{\mathrm{f}}_{i(j)} \boldsymbol{X}_j, \tag{9.39}$$

where the ensembles at all prior times are discarded in the analysis.

9.7.1 EnKF with linear noise free model

Referring to the notation used in Fig. 7.1, let us examine the EnKF with a linear model with no model errors, i.e.

$$\boldsymbol{A}_{i+1} = \boldsymbol{F} \boldsymbol{A}_i. \tag{9.40}$$

It was shown in *Evensen* (2004) that, given the initial ensemble stored in $\boldsymbol{A}_0$, the ensemble forecast at time t_k, becomes

$$\boldsymbol{A}_k = \boldsymbol{F}^k \boldsymbol{A}_0. \tag{9.41}$$

If the EnKF is used to update the solution at every time t_j, where $j = 1, \ldots, J$, the ensemble solution at time t_k becomes

$$\boldsymbol{A}_k = \boldsymbol{F}^k \boldsymbol{A}_0 \prod_{j=1}^{J} \boldsymbol{X}_j, \tag{9.42}$$

where $\boldsymbol{X}_j$ is the matrix defined by (9.38) which when multiplied with the ensemble forecast matrix at time $t_{i(j)}$ produces the analysis ensemble at that time. Thus, starting with $\boldsymbol{A}_0$, the assimilation solution at time $t_{i(1)}$ is obtained by multiplication of $\boldsymbol{F}^{i(1)}$ with $\boldsymbol{A}_0$ to produce the forecast at time $t_{i(1)}$ followed by the multiplication of the forecast with $\boldsymbol{X}_1$.

Note that the expression $\boldsymbol{A}_0 \prod_{j=1}^{J} \boldsymbol{X}_j$ is the EnKS solution at time t_0. Thus, for the linear noise-free model, (9.42) can also be interpreted as a forward integration of the smoother solution from the initial time t_0, until t_k, where $\boldsymbol{A}_k$ is produced.

This means that for a linear model without model errors, the EnKF solution at all times is a combination of the initial ensemble members, and the

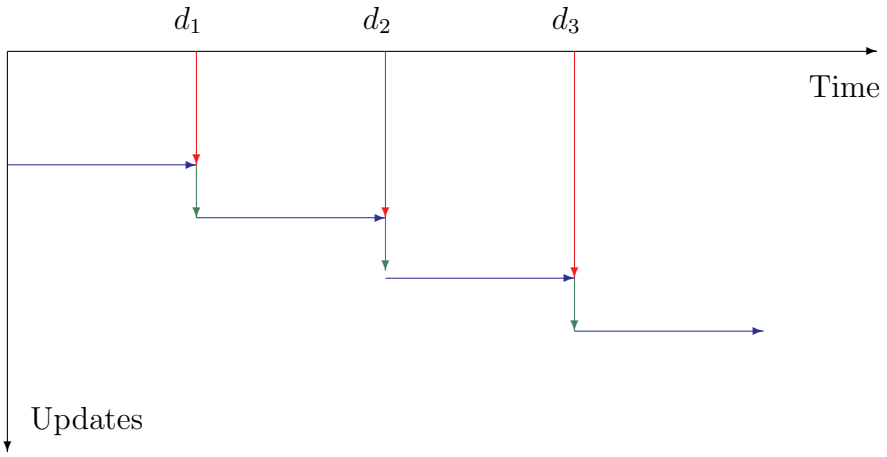

Fig. 9.3. Illustration of the update procedure used in the EnKF. The horizontal axis is time and the measurements are indicated at regular intervals. The vertical axis indicates the number of updates with measurements. The blue arrows represent the forward ensemble integration, the red arrows are the introduction of measurements, while the green arrows is the EnKF update algorithm. Thus, the blue arrows indicate the EnKF solution as a function of time, which is updated every time measurements are available

dimension of the affine space spanned by the initial ensemble does not change with time as long as the operators $\boldsymbol{F}$ and $\boldsymbol{X}_j$ are of full rank. Thus, the quality of the EnKF solution is dependent on the rank and conditioning of the initial ensemble matrix, $\boldsymbol{A}_0$.

9.7.2 EnKS using EnKF as a prior

The EnKS is a straight forward extension of the EnKF. As the EnKF uses the ensemble covariances in space to spread the information from the measurements, the EnKS uses the ensemble covariances in space and time to spread the information also backward in time.

Thus, we can write the analysis update at a time t_l from measurements available at a later time $t_{i(j)}$ as,

$$\boldsymbol{A}^{\mathrm{a}}(\boldsymbol{x}, t_l) = \boldsymbol{A}(\boldsymbol{x}, t_l) + \boldsymbol{A}'(\boldsymbol{x}, t_l)\boldsymbol{S}_j^{\mathrm{T}}\boldsymbol{C}_j^{-1}\boldsymbol{D}'_j, \tag{9.43}$$

where $\boldsymbol{D}'_j$ from (9.32), $\boldsymbol{S}_j$ from (9.33), and $\boldsymbol{C}_j$ from (9.34) are evaluated using the ensemble and measurements at the time $t_{i(j)}$.

It is then seen that the update at the time t_l, uses exactly the same combination of ensemble members as was defined by $\boldsymbol{X}_j$ in (9.38) for the EnKF analysis at the time $t_{i(j)}$. Thus, we can write the EnKS analysis at a time $t_i \in [t_{i(j-1)}, t_{i(j)})$, as

$$\boldsymbol{A}_{\mathrm{EnKS}}(\boldsymbol{x}, t_i) = \boldsymbol{A}_{\mathrm{EnKF}}(\boldsymbol{x}, t_i) \prod_{l=j}^{J} \boldsymbol{X}_l. \tag{9.44}$$

It is then a simple exercise to compute the EnKS analysis as soon as the EnKF solution has been found. This requires only the storage of the coefficient matrices $\boldsymbol{X}_j$, for $j = 1, \ldots, J$, and the EnKF ensemble matrices for the previous times where we want to compute the EnKS analysis. Note that the EnKF ensemble matrices are large, but it is possible to store only specific variables at selected locations where the EnKS solution is needed. An illustration of the sequential processing of measurements is given in Fig. 9.2.

9.8 Example with the Lorenz equations

The example from *Evensen* (1997) was in *Evensen and van Leeuwen* (2000) used to intercompare the ES, EnKS and EnKF, and the results from this intercomparison are now presented. The chaotic Lorenz model by *Lorenz* (1963) is used. It was discussed in Chap. 6, and consists of a system of three coupled and nonlinear ordinary differential equations, (6.5–6.7) with initial conditions (6.8–6.10).

9.8.1 Description of experiments

For all the cases to be discussed the initial conditions for the reference case are given by $(x_0, y_0, z_0) = (1.508870, -1.531271, 25.46091)$ and the time interval is $t \in [0, 40]$. The observations and initial conditions are simulated by adding normal distributed noise with zero mean and variance equal to 2.0 to the reference solution. All of the variables x, y and z are measured. The initial conditions used are also assumed to have the same variance as the observations. These are the same values as were used in *Miller et al.* (1994) and *Evensen* (1997).

The model error covariance is defined to be diagonal with variances equal to 2.000, 12.13, and 12.31 for the three equations (6.5–6.7), respectively. These numbers define the error variance growth expected over one time unit in the model. The reference case is generated by integrating the model equations including the stochastic forcing corresponding to the specified model error variances. The stochastic forcing is included through a term like $\sqrt{\Delta t}\sqrt{\sigma^2}d\omega$ where σ^2 is the model error variance, and $d\omega$ is drawn from the distribution $\mathcal{N}(0, 1)$.

In the calculation of the ensemble statistics an ensemble of 1000 members is used. This is a fairly large ensemble but it is chosen to prevent the possibility of drawing erroneous conclusions due to the use of a too small ensemble. The same simulation was rerun with various ensemble sizes and the differences between the results were negligible even using 50 members of the ensemble.

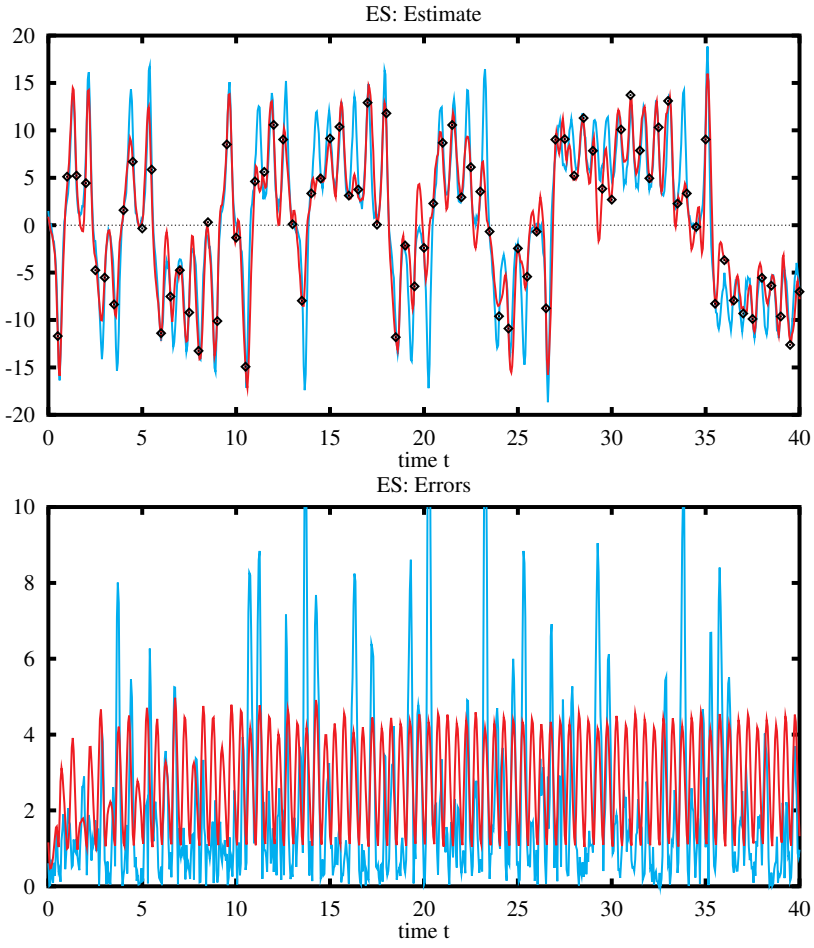

Fig. 9.4. Ensemble Smoother: The inverse estimate *(red line)* and reference solution *(blue line)* for x are shown in the upper plot. The lower plot shows the corresponding estimated standard deviations *(red line)* and the absolute value of the difference between the reference solution and the estimate, i.e. the real posterior errors *(blue line)*. Reproduced from *Evensen and van Leeuwen* (2000)

9.8.2 Assimilation Experiment

The three methods discussed above will now be examined and compared in an experiment where the distance between the measurements is $\Delta t_{\text{obs}} = 0.5$, which is similar to Experiment B in *Evensen* (1997).

In the upper plots in Figs. 9.4–9.7, the red line denotes the estimate and the blue line is the reference solution. In the lower plots the red line is the standard deviation estimated from ensemble statistics, while the blue line is the true residuals with respect to the reference solution.

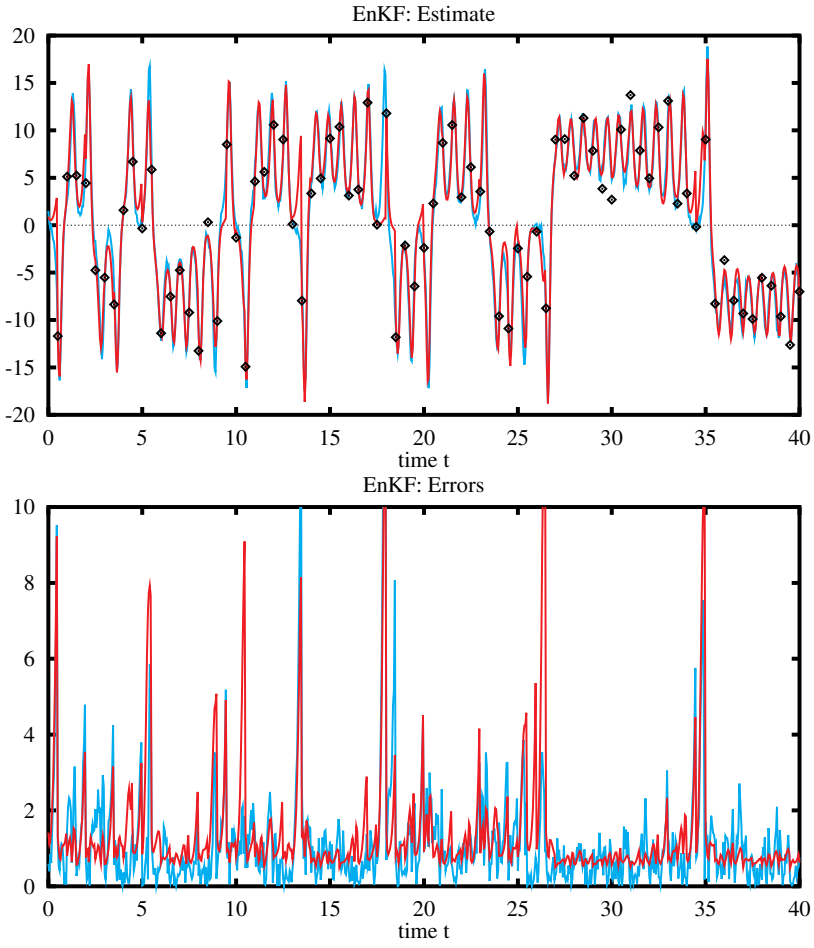

Fig. 9.5. Ensemble Kalman Filter: See explanation in Fig. 9.4. Reproduced from *Evensen and van Leeuwen* (2000)

Ensemble Smoother Solution

The ES solution for the x-component and the estimated error variance are given in Fig. 9.4. It was found that the ES performed rather poorly with the current data density. Note, however, that even if the fit to the reference trajectory is rather poor, it captures most of the transitions. The main problem is related to the estimate of the amplitudes in the reference solution. This is linked to the appearance of non-Gaussian contributions in the distribution for the model evolution, which can be expected in such a strongly nonlinear case.

Remember that the smoother solution consists of a first guess estimate, which is the mean of the freely evolving ensemble, plus a linear combination of

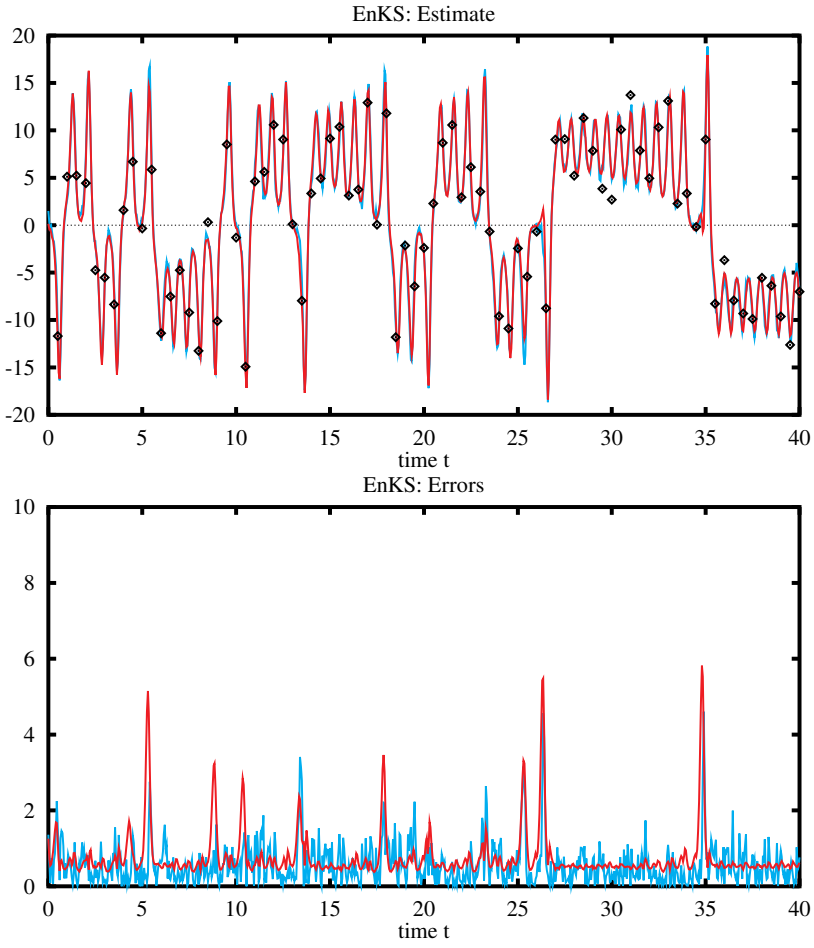

Fig. 9.6. Ensemble Kalman Smoother: See explanation in Fig. 9.4. Reproduced from *Evensen and van Leeuwen* (2000)

time-dependent influence functions or representers which are calculated from the ensemble statistics. Thus, the method becomes equivalent to a variance-minimizing objective analysis method where the time dimension is included.

In the ensemble smoother the posterior error variances can easily be calculated by performing an analysis for each of the ensemble members and then evaluating the variance of the new ensemble. Clearly, the error estimates are not large enough at the peaks where the smoother performs poorly. This is again a result of neglecting the non-Gaussian contribution from the probability distribution for the model evolution. Thus, the method assumes the distribution is Gaussian and believes it is doing well. Otherwise the error estimate looks reasonable with minima at the measurement locations and maxima in

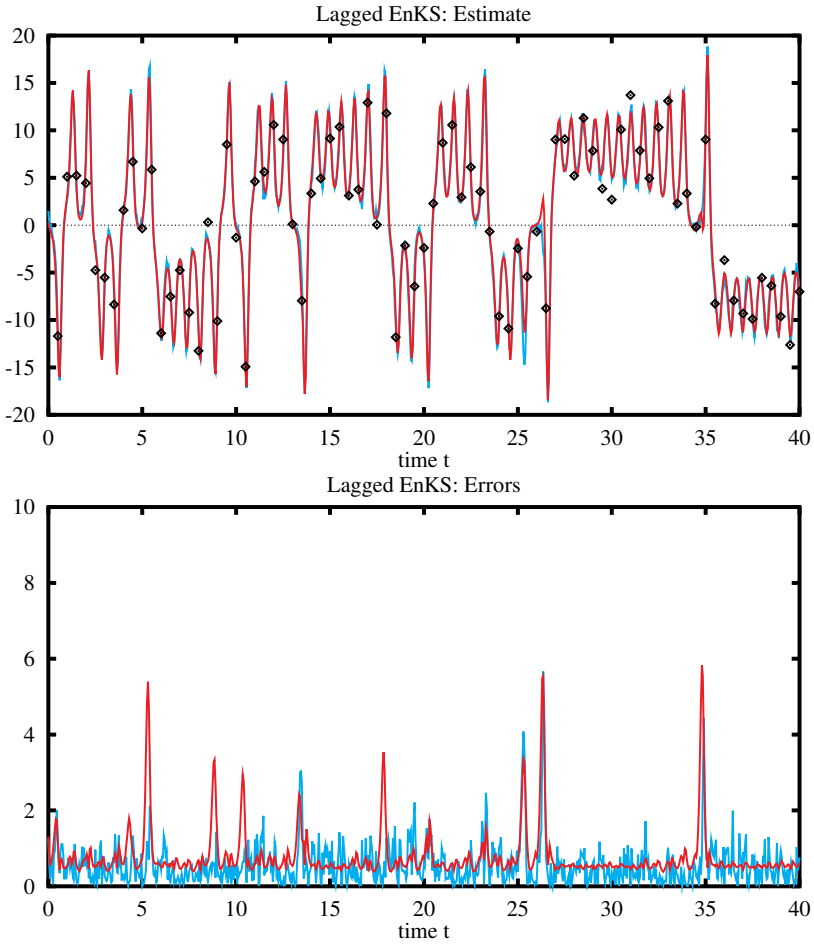

Fig. 9.7. Lagged Ensemble Kalman Smoother: See explanation in Fig. 9.4. Reproduced from *Evensen and van Leeuwen* (2000)

between the measurements. Note again that if a linear model is used the posterior density will be Gaussian and the ensemble smoother will, in the limit of an infinite ensemble size, provide the same solution as the Kalman smoother or the representer method.

Ensemble Kalman Filter Solution

The EnKF does a reasonably good job at tracking the reference solution with the lower data density, as can be seen in Fig. 9.5. One transition is missed near $t = 18$, and there are also a few other locations where the EnKF has problems, e.g. $t = 1$, 5, 9, 10, 13, 17, 19, 23, 26, and 34. The error variance

estimate is consistent, showing large peaks at the locations where the estimate obviously has problems tracking the reference solution. Note also the similarity between the absolute value of the residual between the reference solution and the estimate, and the estimated standard deviation. For all peaks in the residual there is a corresponding peak for the error variance estimate.

The error estimates show the same behaviour as was found by *Miller et al.* (1994) with very strong error growth when the model solution passes through the unstable regions of the state space, and otherwise weak error variance growth or even decay in the stable regions. Note for example the low error variance when $t \in [28, 34]$ corresponding to the oscillation of the solution around one of the attractors.

The probably surprising result is that the EnKF performs better than the ensemble smoother. This is at least surprising based on linear theory, where one has learned that the Kalman smoother solution at the end of the time interval is identical to the Kalman filter solution, and the additional information introduced by propagating the contribution of future measurements backward in time further reduces the error variance compared to the filter solution. Note again that if the model dynamics are linear, the EnKF will give the same solution as the Kalman filter, and the ensemble smoother will give the same result as the Kalman smoother, in the limit of an infinite ensemble size.

Ensemble Kalman Smoother Solution

In Fig. 9.6 the solution obtained by the EnKS is shown. Clearly, the estimate is an improvement upon the EnKF estimate. The solution is smoother in time and seems to provide a better fit to the reference trajectory. Looking in particular at the problematic locations in the EnKF solution, these are all recovered in the smoother estimate. Note, for example, the additional transitions in $t = 1$, 5, 13, and 34, in the EnKF solution which have now been eliminated in the smoother. The missed transition at $t = 17$ has also been recovered in the smoother solution.

The error estimates are reduced throughout the time interval. In particular the large peaks in the EnKF solution are now significantly reduced. As for the EnKF solution there are corresponding peaks in the error estimates for all the peaks in the residuals which proves that the EnKS error estimate is consistent with the true errors.

This is a very promising result. In fact the EnKS solution with $\Delta t_{\text{obs}} = 0.5$ seemed to do as well or better than the EnKF solution with $\Delta t_{\text{obs}} = 0.25$ (see *Evensen*, 1997).

In Fig. 9.7 the result from a lagged smoother is shown. In this case the measurement information in propagated backward in time only for a short time interval. This is motivated by the assumption that the impact of measurements is negligible outside an interval of length similar to the predictability time of the model. A time lag of 5 time units was used and the results are

almost indistinguishable from the full smoother solution. Thus, a significant saving of storage and CPU should be possible for more realistic applications when using the lagged smoother.

9.9 Discussion

The similarity or connection between the EnKS and EnKF has been clarified. The EnKS is the optimal smoother solution for the linear problems with Gaussian statistics. The EnKF is a simplification which does not project information backward in time. After the final measurement time $t_{i(m)}$, the EnKF and EnKS state and parameter estimates are identical and the EnKF is therefore ideal for forecasting purposes.

The ensemble methods introduce an approximation by using only the mean and covariance of the prior joint pdf when computing the posterior ensemble in (9.35). Thus, it is effectively assumed that the prior joint pdf is Gaussian when computing the updates. This means that the EnKS and the EnKF will not give the correct answer if the prior joint pdf has non-Gaussian contributions. On the other hand the ensemble methods have proven to work well with a large number of different nonlinear dynamical models.

The ES method is similar to simple kriging or Gauss-Markov interpolation in space and time, using an ensemble representation for the space-time error covariance matrix. For a linear problem this will give exactly the same results as solving the problem with sequential processing of measurements, or minimizing the generalized inverse formulation (8.20). However, when the model is nonlinear, the long integration of the model, unconstrained by measurements, allows for the development of strongly non-Gaussian contributions in the prior density. In *Evensen and van Leeuwen* (2000) the EnKF, EnKS, and ES were compared using the highly nonlinear Lorenz equations, and it was demonstrated that the non-Gaussian contributions in the ES lead to results which were significantly worse than those obtained using the EnKF and EnKS. Further it was suggested that the sequential introduction of measurements, with Gaussian distributed errors, actually introduced "Gaussianity" to the ensemble representing the conditional joint density.

The derivation of the ensemble methods allowed for the estimation of poorly known model parameters. Examples involving parameter estimation using the EnKF and EnKS will be presented in the following chapters.

10

Statistical optimization

Optimization problems are often solved by minimizing a cost function in search of the global minimum. The solution then corresponds to the maximum likelihood estimate. Many solution methods, e.g. gradient methods, search only for the minimum of the cost function, and do not provide information about the uncertainty of the solution. The uncertainty can be estimated using statistical sampling based on the Metropolis or hybrid Monte Carlo methods from Chap. 6, or by examining the inverse of the Hessian of the cost function around the minimum value. We will now formulate an optimization problem in a Bayesian setting and show how it can be solved using the EnKS. This results in a statistical estimate of the solution and provides error estimates. Several examples are used to illustrate the difference between the exact Bayesian solution and the approximate EnKS solution. Furthermore, the examples illustrate properties of the EnKS when used with non-Gaussian distributions and nonlinear measurement operators.

10.1 Definition of the minimization problem

The EnKS can be used to solve time independent optimization problems. A typical problem could involve a set of parameters $\boldsymbol{\alpha}(\boldsymbol{x}) \in \Re^{n_\alpha}$, which is input to a function or model which outputs a vector of fields $\boldsymbol{\psi}(\boldsymbol{x}) \in \Re^{n_\psi}$, on the spatial domain $\mathcal{D}$. In addition we have available some observations of the true field $\boldsymbol{\psi}^{\mathrm{t}}(\boldsymbol{x})$. The problem is then to find the set of input parameters $\boldsymbol{\alpha}$, which gives the best possible correspondence between the simulated fields and the observations. Such optimization problems are usually solved by first defining an appropriate cost function and then solving for the minimum. However, if the functional mapping is nonlinear, the cost function is likely to contain local minima and the global minimum may be hard to find. Furthermore, traditional methods do not allow the functional mapping to contain errors nor do they provide any information about the uncertainties of the solution.

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_10,

10.1.1 Parameters

We start by defining a set of first-guess parameters $\boldsymbol{\alpha}^{\mathrm{f}}(\boldsymbol{x}) \in \Re^{n_\alpha}$, which can be either constants or functions of the spatial coordinate, and we assume that they contain stochastic errors $\boldsymbol{\alpha}'(\boldsymbol{x}) \in \Re^{n_\alpha}$, with mean equal to zero and known covariance $\boldsymbol{C}_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \Re^{n_\alpha \times n_\alpha}$. This is represented in the following equation

$$\boldsymbol{\alpha}(\boldsymbol{x}) = \boldsymbol{\alpha}^{\mathrm{f}}(\boldsymbol{x}) + \boldsymbol{\alpha}'(\boldsymbol{x}), \tag{10.1}$$

which states that the estimated value of $\boldsymbol{\alpha}$ should be close to the prior $\boldsymbol{\alpha}^{\mathrm{f}}$, but allowed to deviate from it according to the uncertainty represented by the stochastic error term.

10.1.2 Model

We then define our function or model which connects the simulated realization $\boldsymbol{\psi}(\boldsymbol{x})$, to the parameters $\boldsymbol{\alpha}(\boldsymbol{x})$, as

$$\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{G}(\boldsymbol{\alpha}) + \boldsymbol{q}(\boldsymbol{x}), \tag{10.2}$$

where $\boldsymbol{G}(\boldsymbol{\alpha}) \in \Re^{n_\psi}$ is the nonlinear model operator and $\boldsymbol{q}(\boldsymbol{x}) \in \Re^{n_\psi}$ is an additive stochastic term representing the errors in the model. We assume that the model errors have a Gaussian distribution with mean equal to zero and known covariance $\boldsymbol{C}_{qq}(\boldsymbol{x}_1, \boldsymbol{x}_2) \in \Re^{n_\psi \times n_\psi}$. Thus, for any realization $\boldsymbol{\alpha}_j$, we can simulate a realization $\boldsymbol{\psi}_j(\boldsymbol{x})$. The case with non-additive model errors, e.g. $\boldsymbol{G}(\boldsymbol{\alpha}, \boldsymbol{q})$, can be treated using an approach which is similar to the one used for estimation of time correlated model errors in Chap. 12.

10.1.3 Measurements

The M measurements of the true mapping are stored in the data vector $\boldsymbol{d} \in \Re^M$. We assume that the measurements can be related to a simulated realization through the measurement functional

$$\mathcal{M}\big[\boldsymbol{\psi}(\boldsymbol{x})\big] = \boldsymbol{d} + \boldsymbol{\epsilon}, \tag{10.3}$$

where $\boldsymbol{\epsilon} \in \Re^M$ represents random measurement errors. Here $\mathcal{M}\big[\boldsymbol{\psi}(\boldsymbol{x})\big] \in \Re^M$ just projects the functional mapping $\boldsymbol{\psi}(\boldsymbol{x})$, onto the measurements. It will typically be similar to (7.6) but excluding the time variable in this case. Thus, given a field $\boldsymbol{\psi}(\boldsymbol{x})$, we can find the prediction of the measurement of the field by evaluating $\mathcal{M}\big[\boldsymbol{\psi}(\boldsymbol{x})\big]$. Also for the random measurement errors $\boldsymbol{\epsilon}$, we assume Gaussian statistics with zero mean and known covariance $\boldsymbol{C}_{\epsilon\epsilon} \in \Re^{M \times M}$.

10.1.4 Cost function

A cost function can be defined as

$$\begin{aligned}\mathcal{J}[\boldsymbol{\alpha},\boldsymbol{\psi}] = &\iint_{\mathcal{D}} \big(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\mathrm{f}}\big)_1^{\mathrm{T}} \boldsymbol{W}_{\alpha\alpha}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\mathrm{f}}\big)_2 d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\ &+ \iint_{\mathcal{D}} \big(\boldsymbol{\psi}-\boldsymbol{G}(\boldsymbol{\alpha})\big)_1 \boldsymbol{W}_{qq}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\boldsymbol{\psi}-\boldsymbol{G}(\boldsymbol{\alpha})\big)_2 d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\ &+ \Big(\boldsymbol{d}-\mathcal{M}\big[\boldsymbol{\psi}\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d}-\mathcal{M}\big[\boldsymbol{\psi}\big]\Big).\end{aligned} \tag{10.4}$$

This is a fairly general cost function which measures the errors in the first-guess parameters, the model and the measurements, in a weighted least squares sense. The subscripts, 1 and 2, denote functions of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively. It is natural to assume that the weights $\boldsymbol{W}_{\alpha\alpha}$ and $\boldsymbol{W}_{\epsilon\epsilon}$ are inverses of the error covariances, $\boldsymbol{C}_{\alpha\alpha}$ and $\boldsymbol{C}_{\epsilon\epsilon}$, as before, see Chap. 8. For the weight, $\boldsymbol{W}_{qq}(\boldsymbol{x}_1,\boldsymbol{x}_2)$, we define

$$\int_{\mathcal{D}} \boldsymbol{W}_{qq}(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{C}_{qq}(\boldsymbol{x}_2,\boldsymbol{x}_3)d\boldsymbol{x}_2 = \delta(\boldsymbol{x}_1-\boldsymbol{x}_3)\boldsymbol{I}, \tag{10.5}$$

with $\delta(\boldsymbol{x}_1-\boldsymbol{x}_2)$ being the Dirac delta function and $\boldsymbol{I}\in\Re^{n_\psi\times n_\psi}$ the diagonal identity matrix.

If the model is assumed to be perfect we can rewrite the cost function as

$$\begin{aligned}\mathcal{J}[\boldsymbol{\alpha}] = &\iint_{\mathcal{D}} \big(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\mathrm{f}}\big)_1^{\mathrm{T}} \boldsymbol{W}_{\alpha\alpha}(\boldsymbol{x}_1,\boldsymbol{x}_2)\big(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\mathrm{f}}\big)_2 d\boldsymbol{x}_1 d\boldsymbol{x}_2 \\ &+ \Big(\boldsymbol{d}-\mathcal{M}\big[\boldsymbol{G}(\boldsymbol{\alpha})\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d}-\mathcal{M}\big[\boldsymbol{G}(\boldsymbol{\alpha})\big]\Big).\end{aligned} \tag{10.6}$$

This is the standard cost function which is minimized in many applications.

10.2 Bayesian formalism

In a Bayesian formalism we can derive the cost function by assuming that we have given the pdf for the parameters $\boldsymbol{\alpha}$ as $f(\boldsymbol{\alpha})$, and the pdf for the model as $f(\boldsymbol{\psi}|\boldsymbol{\alpha})$. Furthermore, we have the likelihood for the measurements $\boldsymbol{d}$, given as

$$f(\boldsymbol{d}|\boldsymbol{\alpha},\boldsymbol{\psi}) = f(\boldsymbol{d}|\boldsymbol{\psi}), \tag{10.7}$$

since the measurements, in this case, are assumed to be independent of $\boldsymbol{\alpha}$.

Bayes' theorem states that

$$\begin{aligned}f(\boldsymbol{\alpha},\boldsymbol{\psi}|\boldsymbol{d}) &\propto f(\boldsymbol{d}|\boldsymbol{\alpha},\boldsymbol{\psi})f(\boldsymbol{\alpha},\boldsymbol{\psi}) \\ &= f(\boldsymbol{d}|\boldsymbol{\psi})f(\boldsymbol{\psi}|\boldsymbol{\alpha})f(\boldsymbol{\alpha}).\end{aligned} \tag{10.8}$$

If we assume Gaussian statistics for all the errors we get

$$f(\boldsymbol{\alpha}) \propto \exp\left(-\frac{1}{2}\iint_{\mathcal{D}} \left(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\mathrm{f}}\right)_1^{\mathrm{T}} \boldsymbol{W}_{\alpha\alpha}(\boldsymbol{x}_1,\boldsymbol{x}_2)\left(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{\mathrm{f}}\right)_2 d\boldsymbol{x}_1 d\boldsymbol{x}_2\right), \tag{10.9}$$

$$\begin{aligned} f(\boldsymbol{\psi}|\boldsymbol{\alpha}) \propto \exp\Bigg(-\frac{1}{2}\iint_{\mathcal{D}} &\left(\boldsymbol{\psi}-\boldsymbol{G}(\boldsymbol{\alpha})\right)_1 \\ &\times W_{qq}(\boldsymbol{x}_1,\boldsymbol{x}_2)\left(\boldsymbol{\psi}-\boldsymbol{G}(\boldsymbol{\alpha})\right)_2 d\boldsymbol{x}_1 d\boldsymbol{x}_2\Bigg), \end{aligned} \tag{10.10}$$

and

$$f(\boldsymbol{d}|\boldsymbol{\psi}) \propto \exp\left(-\frac{1}{2}\Big(\boldsymbol{d}-\mathcal{M}\big[\boldsymbol{\psi}\big]\Big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon}\Big(\boldsymbol{d}-\mathcal{M}\big[\boldsymbol{\psi}\big]\Big)\right). \tag{10.11}$$

Insertion of these into (10.8) gives

$$f(\boldsymbol{\alpha},\boldsymbol{\psi}|\boldsymbol{d}) \propto \exp\left(-\frac{1}{2}\mathcal{J}[\boldsymbol{\alpha},\boldsymbol{\psi}]\right). \tag{10.12}$$

Maximization of (10.12), which results in the maximum likelihood solution, is equivalent to minimization of the cost function as defined in (10.4).

Standard minimization of the cost function (10.4) using gradient methods may be difficult since this requires derivatives of $G(\boldsymbol{\alpha})$ and $\mathcal{M}\big[\boldsymbol{\psi}\big]$ and if the model operator in sufficiently nonlinear these methods are likely to get trapped in local minima. Furthermore, the dimension of the problem becomes high since we need to minimize with respect to both $\boldsymbol{\alpha}$ and $\boldsymbol{\psi}(\boldsymbol{x})$ simultaneously.

10.3 Solution by ensemble methods

The EnKS does not minimize the cost function directly. Rather it takes the pdfs and likelihood functions as a starting point, and represents these using large ensembles of realizations. To illustrate, we could start by sampling N realizations $\boldsymbol{\alpha}_j^{\mathrm{f}}$, from $f(\boldsymbol{\alpha})$ as defined in (10.9). We then compute the N realizations $\boldsymbol{\psi}_j^{\mathrm{f}}$, by evaluating the stochastic model (10.2) for the N parameter sets $\boldsymbol{\alpha}_j^{\mathrm{f}}$. The simulated realizations are then measured to generate an ensemble of predicted measurements. Thus, we have,

$$\boldsymbol{\alpha}_j^{\mathrm{f}} = \boldsymbol{\alpha}^{\mathrm{f}} + \boldsymbol{\alpha}_j', \tag{10.13}$$

$$\boldsymbol{\psi}_j^{\mathrm{f}}(\boldsymbol{x}) = \boldsymbol{G}(\boldsymbol{\alpha}_j^{\mathrm{f}}) + \boldsymbol{q}_j(\boldsymbol{x}), \tag{10.14}$$

$$\widehat{\boldsymbol{d}_j} = \mathcal{M}\big[\boldsymbol{\psi}_j^{\mathrm{f}}\big], \tag{10.15}$$

where $\widehat{\boldsymbol{d}_j}$ is the prediction of the measurements given $\boldsymbol{\alpha}_j$. Note that in (10.15) it would also be possible to introduce a stochastic error term to take into account representation errors in the measurement operator.

It is also possible to combine these equations and write

$$\widehat{\boldsymbol{d}_j} = \mathcal{M}\big[\boldsymbol{G}(\boldsymbol{\alpha}^{\mathrm{f}} + \boldsymbol{\alpha}'_j) + \boldsymbol{q}_j(\boldsymbol{x})\big], \tag{10.16}$$

where only $\boldsymbol{\alpha}$ is used as a state vector, but we will retain the form (10.13–10.15). The state vector which originally consisted of only $\boldsymbol{\alpha}$ can then be extended to include both the functional mapping and the predicted measurements, i.e. we define the realizations

$$\boldsymbol{\Psi}^{\mathrm{f}}_j = \begin{pmatrix} \boldsymbol{\alpha}^{\mathrm{f}}_j \\ \boldsymbol{\psi}^{\mathrm{f}}_j(\boldsymbol{x}) \\ \widehat{\boldsymbol{d}_j} \end{pmatrix}. \tag{10.17}$$

From the N realizations $\boldsymbol{\Psi}^{\mathrm{f}}_j$, it is possible to compute the symmetrical ensemble covariance

$$\boldsymbol{C}^{\mathrm{f}}_{\Psi\Psi} = \begin{pmatrix} \boldsymbol{C}^{\mathrm{f}}_{\alpha\alpha}(\boldsymbol{x}_1,\boldsymbol{x}_2) & \boldsymbol{C}^{\mathrm{f}}_{\alpha\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2) & \boldsymbol{C}^{\mathrm{f}}_{\alpha d}(\boldsymbol{x}_1) \\ \boldsymbol{C}^{\mathrm{f}}_{\psi\alpha}(\boldsymbol{x}_1,\boldsymbol{x}_2) & \boldsymbol{C}^{\mathrm{f}}_{\psi\psi}(\boldsymbol{x}_1,\boldsymbol{x}_2) & \boldsymbol{C}^{\mathrm{f}}_{\psi d}(\boldsymbol{x}_1) \\ \boldsymbol{C}^{\mathrm{f}}_{d\alpha}(\boldsymbol{x}_2) & \boldsymbol{C}^{\mathrm{f}}_{d\psi}(\boldsymbol{x}_2) & \boldsymbol{C}^{\mathrm{f}}_{dd} \end{pmatrix}. \tag{10.18}$$

Thus, we have defined the first-guess covariance matrices between the components of the state vector; $\boldsymbol{C}^{\mathrm{f}}_{\alpha\alpha} \in \Re^{n_\alpha \times n_\alpha}$, $\boldsymbol{C}^{\mathrm{f}}_{\psi\psi} \in \Re^{n_\psi \times n_\psi}$, $\boldsymbol{C}^{\mathrm{f}}_{dd} \in \Re^{M\times M}$, $\boldsymbol{C}^{\mathrm{f}}_{\alpha\psi} \in \Re^{n_\alpha \times n_\psi}$, $\boldsymbol{C}^{\mathrm{f}}_{\alpha d} \in \Re^{n_\alpha \times M}$ and $\boldsymbol{C}^{\mathrm{f}}_{d\psi} \in \Re^{M\times n_\psi}$.

We can now define the cost function

$$\begin{aligned} \mathcal{J}[\boldsymbol{\Psi}] = {} & \big(\boldsymbol{\Psi} - \boldsymbol{\Psi}^{\mathrm{f}}\big)^{\mathrm{T}} \boldsymbol{W}_{\Psi\Psi} \big(\boldsymbol{\Psi} - \boldsymbol{\Psi}^{\mathrm{f}}\big) \\ & + \big(\boldsymbol{d} - \boldsymbol{M}\boldsymbol{\Psi}^{\mathrm{f}}\big)^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} \big(\boldsymbol{d} - \boldsymbol{M}\boldsymbol{\Psi}^{\mathrm{f}}\big). \end{aligned} \tag{10.19}$$

Note that $\boldsymbol{W}_{\epsilon\epsilon}$ is the inverse of the error covariance matrix of the measurement errors $\boldsymbol{C}_{\epsilon\epsilon}$, while $\boldsymbol{C}_{dd}$ is the ensemble covariance matrix of the model predicted measurements. We have defined $\boldsymbol{M}$ as a matrix operator which extracts the predicted measurements from $\boldsymbol{\Psi}$, i.e.

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{0}^{n_\alpha \times n_\alpha} & \boldsymbol{0}^{n_\alpha \times n_\psi} & \boldsymbol{0}^{n_\alpha \times M} \\ \boldsymbol{0}^{n_\psi \times n_\alpha} & \boldsymbol{0}^{n_\psi \times n_\psi} & \boldsymbol{0}^{n_\psi \times M} \\ \boldsymbol{0}^{M \times n_\alpha} & \boldsymbol{0}^{M \times n_\psi} & \mathcal{M}^{M\times M} \end{pmatrix}. \tag{10.20}$$

The first-guess estimate is computed as the mean of the first-guess ensemble and we write, with the overline denoting ensemble average,

$$\boldsymbol{\Psi}^{\mathrm{f}} = \begin{pmatrix} \overline{\boldsymbol{\alpha}^{\mathrm{f}}} \\ \overline{\boldsymbol{\psi}^{\mathrm{f}}} \\ \widehat{\boldsymbol{d}} \end{pmatrix}, \tag{10.21}$$

where $\overline{\boldsymbol{\alpha}^{\mathrm{f}}} = \boldsymbol{\alpha}^{\mathrm{f}}$. We have defined the inverse of the covariance $\boldsymbol{C}_{\Psi\Psi}$ as $\boldsymbol{W}_{\Psi\Psi}$, using the now-familiar definitions for the inverses of covariances which are functions of the spatial coordinate.

10.3.1 Variance minimizing solution

From the theory outlined in Chaps. 3 and 9, it is easy to show that the variance minimizing solution $\boldsymbol{\Psi}^{\mathrm{a}}$, of (10.19) becomes

$$\boldsymbol{\Psi}^{\mathrm{a}} = \boldsymbol{\Psi}^{\mathrm{f}} + \boldsymbol{C}_{\Psi\Psi}\boldsymbol{M}^{\mathrm{T}}\left(\boldsymbol{M}\boldsymbol{C}_{\Psi\Psi}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}_{\epsilon\epsilon}\right)^{-1}\left(\boldsymbol{d} - \boldsymbol{M}\boldsymbol{\Psi}^{\mathrm{f}}\right). \tag{10.22}$$

This can be written in simpler form as

$$\begin{pmatrix}\boldsymbol{\alpha}^{\mathrm{a}}\\ \boldsymbol{\psi}^{\mathrm{a}}\\ \widehat{\boldsymbol{d}}^{\mathrm{a}}\end{pmatrix} = \begin{pmatrix}\boldsymbol{\alpha}^{\mathrm{f}}\\ \boldsymbol{\psi}^{\mathrm{f}}\\ \widehat{\boldsymbol{d}}\end{pmatrix} + \begin{pmatrix}\boldsymbol{C}_{\alpha d}\\ \boldsymbol{C}_{\psi d}\\ \boldsymbol{C}_{dd}\end{pmatrix}\left(\boldsymbol{C}_{dd} + \boldsymbol{C}_{\epsilon\epsilon}\right)^{-1}\Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{G}(\boldsymbol{\alpha}^{\mathrm{f}})\big]\Big), \tag{10.23}$$

or if only $\boldsymbol{\alpha}$ is solved for we write

$$\boldsymbol{\alpha}^{\mathrm{a}} = \boldsymbol{\alpha}^{\mathrm{f}} + \boldsymbol{C}_{\alpha d}\left(\boldsymbol{C}_{dd} + \boldsymbol{C}_{\epsilon\epsilon}\right)^{-1}\Big(\boldsymbol{d} - \mathcal{M}\big[\boldsymbol{G}(\boldsymbol{\alpha}^{\mathrm{f}})\big]\Big). \tag{10.24}$$

10.3.2 EnKS solution

The EnKS solves (10.23) using an ensemble representation for $\boldsymbol{\Psi}$; i.e. given an ensemble of realizations $\boldsymbol{\alpha}_j^{\mathrm{f}}$, for the parameters we compute the corresponding ensembles of realizations, $\boldsymbol{\psi}_j^{\mathrm{f}}(\boldsymbol{x})$ and $\widehat{\boldsymbol{d}_j}$, using the defined prior error statistics for the stochastic terms. The covariances in $\boldsymbol{C}_{\Psi\Psi}$ are all evaluated directly from the ensemble of realizations $\boldsymbol{\Psi}_j$.

The EnKS can be used to update the whole ensemble, $\boldsymbol{\Psi}_j$ with $j = 1, N$, not just the mean, and the result is a full ensemble of parameters $\boldsymbol{\alpha}_j^{\mathrm{a}}$, consistent with the priors and data. Further, the spread of the ensemble of parameters also determines the uncertainty of the estimated parameters.

The actual procedure is similar to the one used in Chap. 9. We store the ensemble members in the matrix $\boldsymbol{A}$, defined as

$$\boldsymbol{A} = (\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2, \ldots, \boldsymbol{\Psi}_N). \tag{10.25}$$

Then the ensemble mean is stored in each column of $\overline{\boldsymbol{A}}$ which can be defined as

$$\overline{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{1}_N, \tag{10.26}$$

where $\boldsymbol{1}_N \in \Re^{N\times N}$ is the matrix where each element is equal to $1/N$. We can then define the ensemble perturbation matrix as

$$\boldsymbol{A}' = \boldsymbol{A} - \overline{\boldsymbol{A}} = \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{1}_N). \tag{10.27}$$

The first-guess ensemble-covariance representation of $\boldsymbol{C}_{\Psi\Psi}^{\mathrm{f}}$ in (10.18), can be defined as

$$\boldsymbol{C}_{\Psi\Psi}^{\mathrm{e}} = \frac{\boldsymbol{A}'\boldsymbol{A}'^{\mathrm{T}}}{N-1}. \tag{10.28}$$

Example	$F(x)$	x_{prior}	y_{prior}	σ_x	σ_y	σ_q
1a	$y = x$	1.0	−1.0	1.0	0.3	1.0
1b	$y = x$	1.0	−1.0	1.0	0.3	0.1
2	$y = x^2$	1.0	−1.0	1.0	0.3	1.0
3	$y = x^2(x^2 - 2)$	1.0	−1.0	1.0	0.3	1.0
4a	$y = \cos(x)$	1.0	−1.0	1.0	0.3	1.0
4b	$y = \cos(x)$	1.0	−1.0	1.0	0.3	0.1
4c	$y = \cos(x)$	1.0	−1.0	4.0	0.3	1.0

Table 10.1. Parameters used in the different examples. Here x_{prior} is the first-guess of x, while y_{prior} is the "observation" of y. The standard deviations for the errors in the priors and the model are σ_x, σ_y and σ_q

We then define N vectors of perturbed measurements as

$$\boldsymbol{d}_j = \boldsymbol{d} + \boldsymbol{\epsilon}_j, \qquad j = 1, \ldots, N, \tag{10.29}$$

which can be stored in the columns of a matrix

$$\boldsymbol{D} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N) \in \Re^{M \times N}. \tag{10.30}$$

The ensemble of measurement perturbations, with mean equal to zero, can be stored in the matrix

$$\boldsymbol{E} = (\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_N) \in \Re^{M \times N}, \tag{10.31}$$

from which we can construct the ensemble representation of the measurement error covariance matrix

$$\boldsymbol{C}_{\epsilon\epsilon}^{\text{e}} = \frac{\boldsymbol{E}\boldsymbol{E}^{\text{T}}}{N-1}. \tag{10.32}$$

We can then write

$$\boldsymbol{A}^{\text{a}} = \boldsymbol{A}^{\text{f}} + \boldsymbol{A}'^{\text{f}}\big(\boldsymbol{M}\boldsymbol{A}'^{\text{f}}\big)^{\text{T}}\Big(\boldsymbol{M}\boldsymbol{A}'^{\text{f}}\big(\boldsymbol{M}\boldsymbol{A}'^{\text{f}}\big)^{\text{T}} + \boldsymbol{E}\boldsymbol{E}^{\text{T}}\Big)\Big(\boldsymbol{D} - \boldsymbol{M}\boldsymbol{A}^{\text{f}}\Big), \tag{10.33}$$

which is the equation solved in the EnKS. This equation has the nice property that the covariance of $\boldsymbol{A}^{\text{a}}$ is the correct expected covariance of the analyzed estimate.

10.4 Examples

A simple example is now used to illustrate the difference between standard minimization problems and statistical estimation. We start by defining a simple scalar model or mapping $y = F(x)$, where x now takes the role of the poorly known parameter α, and y takes the role of the observed variable ψ. The standard cost function for this problem becomes

$$J[x] = (x - x_0)^2/\sigma_x^2 + (d - F(x))^2/\sigma_y^2. \tag{10.34}$$

When using a Bayesian approach, we can evaluate the product of the Gaussian pdf for the prior and the pdf for the model evolution, assuming Gaussian model errors, i.e.

$$f(x, y) = f(y|x)f(x) \propto \exp\left(-\frac{1}{2}\frac{(x - x_0)^2}{\sigma_x^2} - \frac{1}{2}\frac{(y - F(x))^2}{\sigma_q^2}\right). \tag{10.35}$$

The joint conditional pdf becomes

$$f(x, y|d) \propto \exp\left(-\frac{1}{2}\frac{(x - x_0)^2}{\sigma_x^2} - \frac{1}{2}\frac{(y - F(x))^2}{\sigma_q^2} - \frac{1}{2}\frac{(d - y)^2}{\sigma_y^2}\right). \tag{10.36}$$

Figs. 10.1–10.7 display the resulting cost functions and pdfs for several mappings as defined in Table 10.1, and using different input parameters. The joint pdf with its marginal pdfs, modes and mean are shown in the upper left plot. The upper right plot shows the similar pdf but as estimated from a large ensemble of realizations. The lower left plot shows the joint pdf conditional on the measurement and the lower right plot is the corresponding pdf as computed from the samples conditioned on the data using the EnKS.

In Cases 1a and 1b, shown in Figs. 10.1 and 10.2, we assume the linear model $F(x) = x$. In these cases the cost function becomes quadratic, and the marginal pdfs are all Gaussian as would be expected. This case in particular illustrates the impact of model errors. In Case 1a the joint pdf for the prediction in the upper plots shows a large uncertainty while in Case 1b, it is narrow and nearly aligned along the line $y = x$. In Case 1b the most likely solution is found close to the line $y = x$ and consistent with the prior for y, i.e. the pdf for the measurement of y. It is also consistent with the minimum of the cost function. In Case 1a, a completely different solution is found which reflects that the model prediction has a great uncertainty and this leads to a situation where the measurement of y has less impact on the estimate of x. The apparent tilt of the predicted joint pdf in Case 1a is expected. The reason is that, given a value for x, the model uncertainty introduces an uncertainty in the y value (which is symmetrical in the y-direction about a point on the $y = x$ line). In Cases 1a and 1b the maximum likelihood estimate from the joint pdf is identical to the maximum likelihood estimate from the marginal pdfs as well as the estimated mean. This will be true only in the case with a linear model and Gaussian priors. It is also clear that the EnKS in this case produces a consistent result, as is expected.

In Case 2 we introduce a nonlinearity using the function $F(x) = x^2$. Still the problem has only one global minimum and no local minima. In this case we see from Fig. 10.3 that both the joint pdf and marginal pdfs become non-Gaussian. We can also differentiate between the maximum likelihood estimate from the joint and marginal pdfs as well as the mean. Thus, here we will have to choose which estimator to use. From the two lower plots it is also clear that

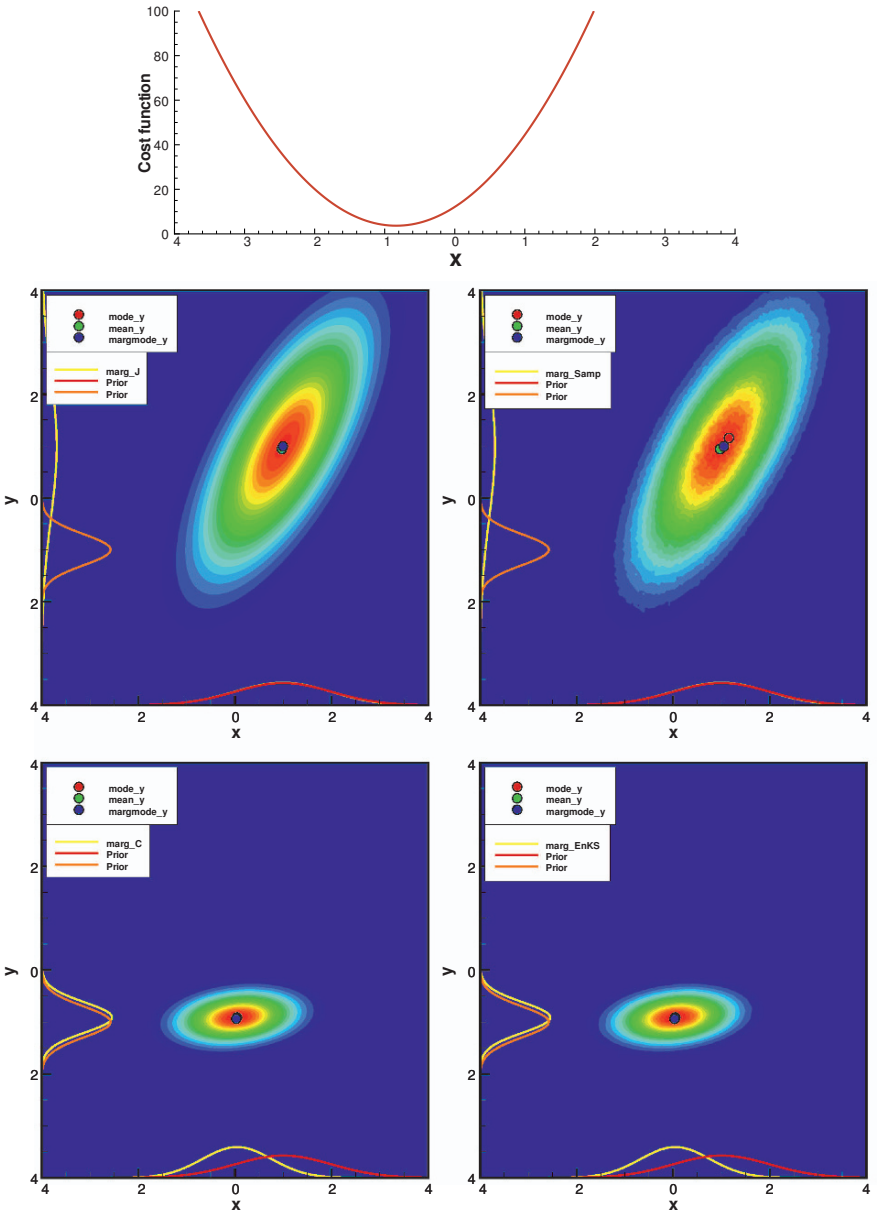

Fig. 10.1. Case 1a: Joint and conditional pdfs using the linear function $F(x) = x$

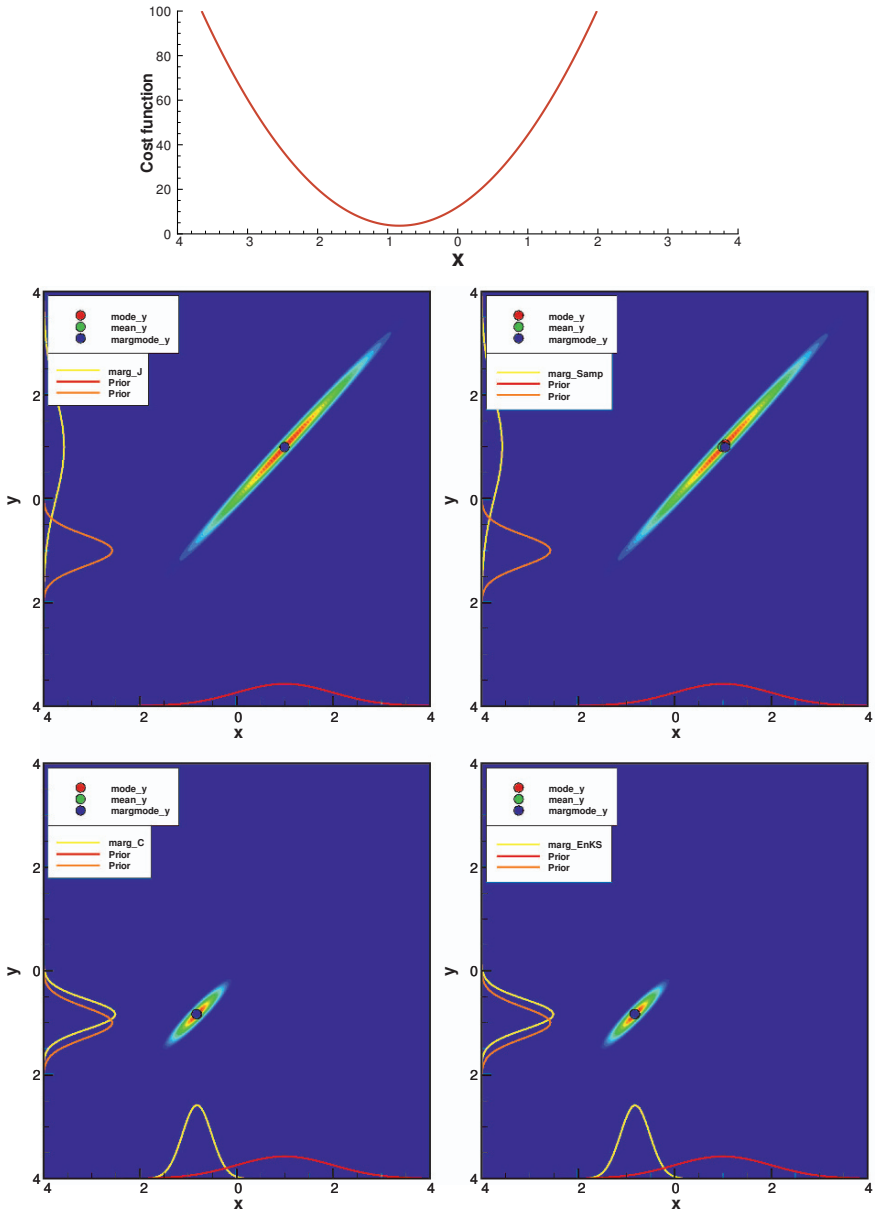

Fig. 10.2. Case 1b: Same as Fig. 10.1 but with more accurate model

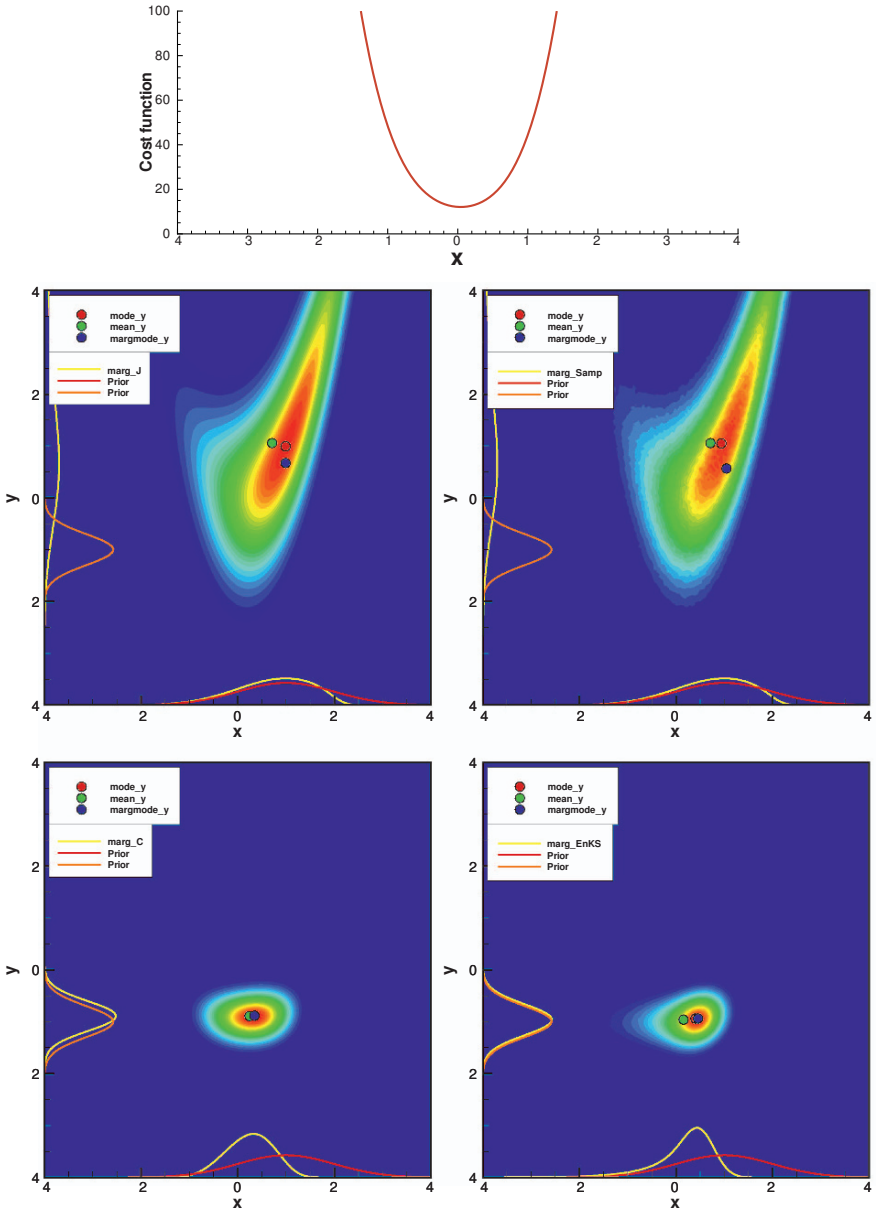

Fig. 10.3. Case 2: Joint and conditional pdfs using the quadratic function $F(x) = x^2$

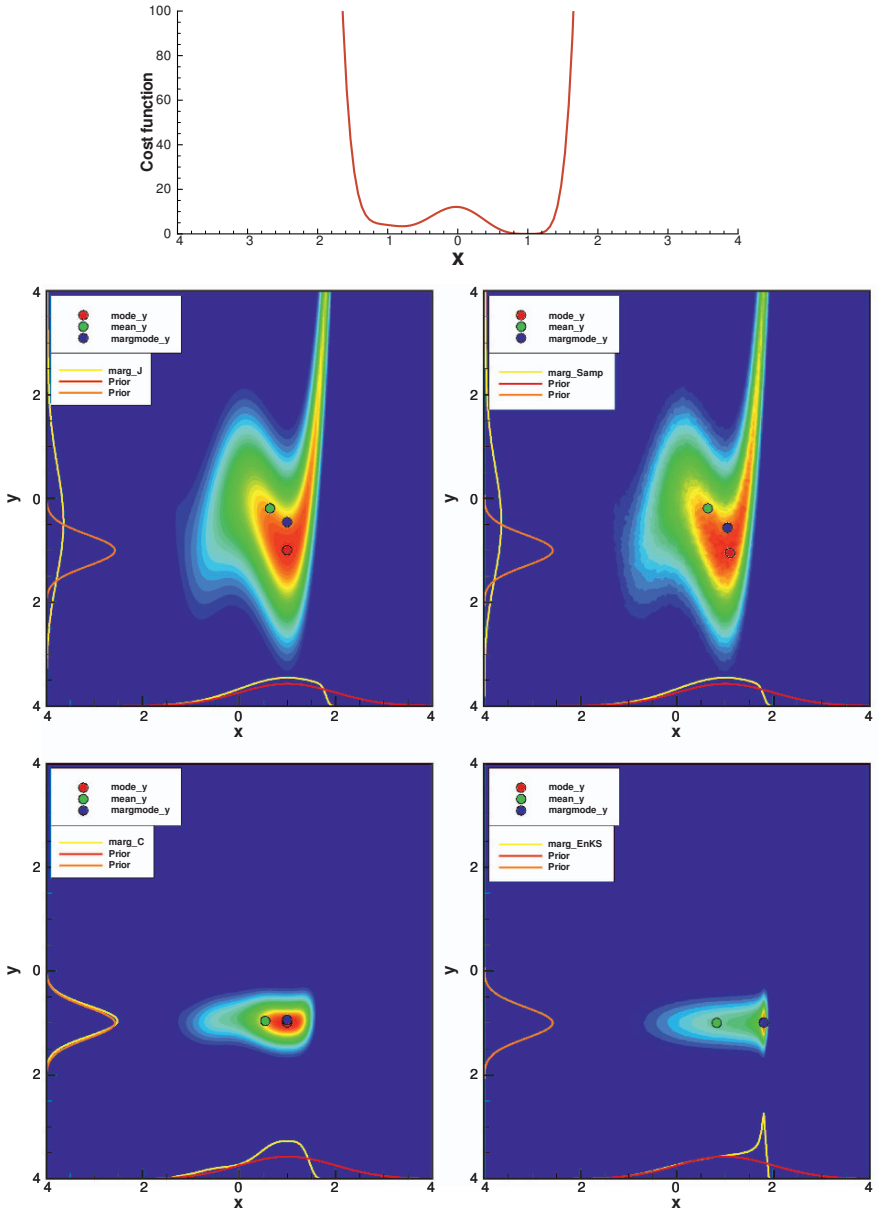

Fig. 10.4. Case 3: Joint and conditional pdfs using the nonlinear function $F(x) = x^2(x^2 - 2)$

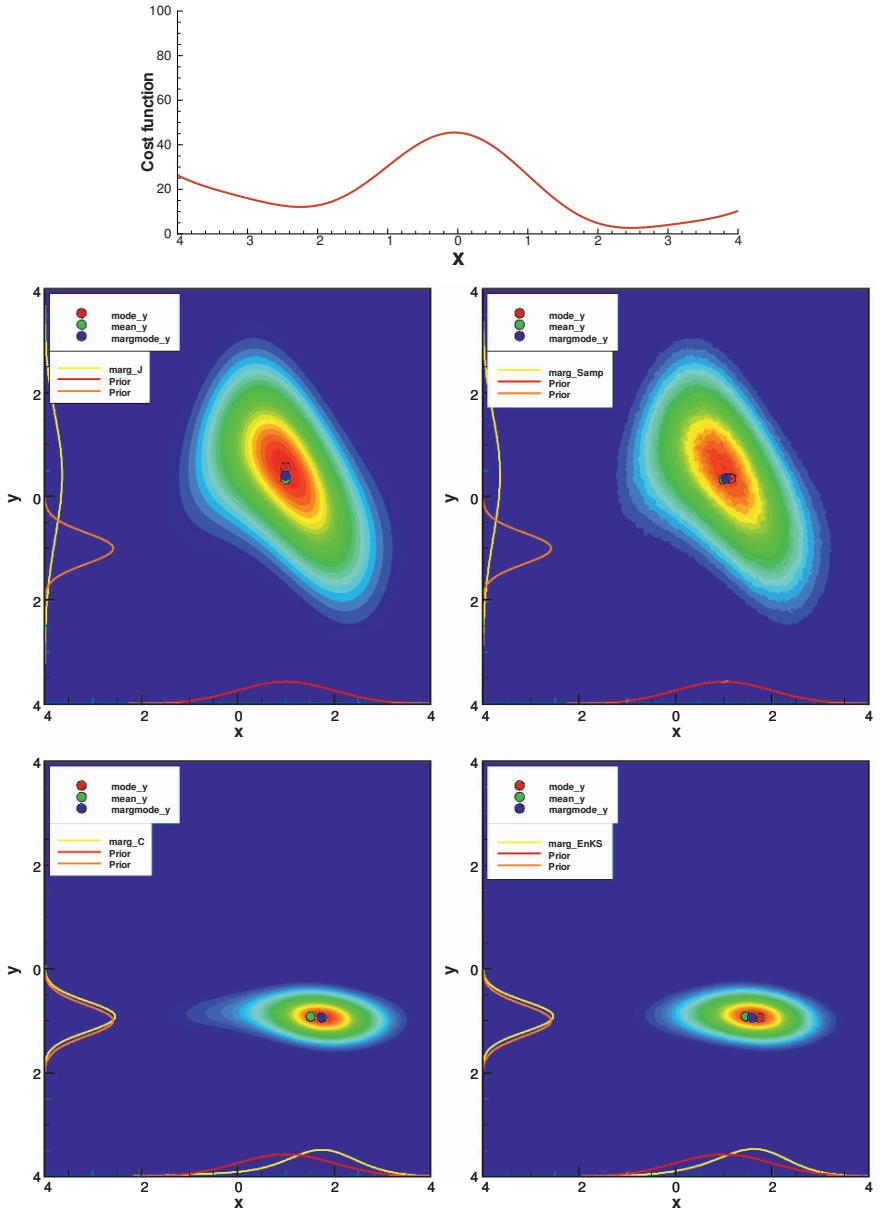

Fig. 10.5. Case 4a: Joint and conditional pdfs using the nonlinear function $F(x) = \cos(x)$

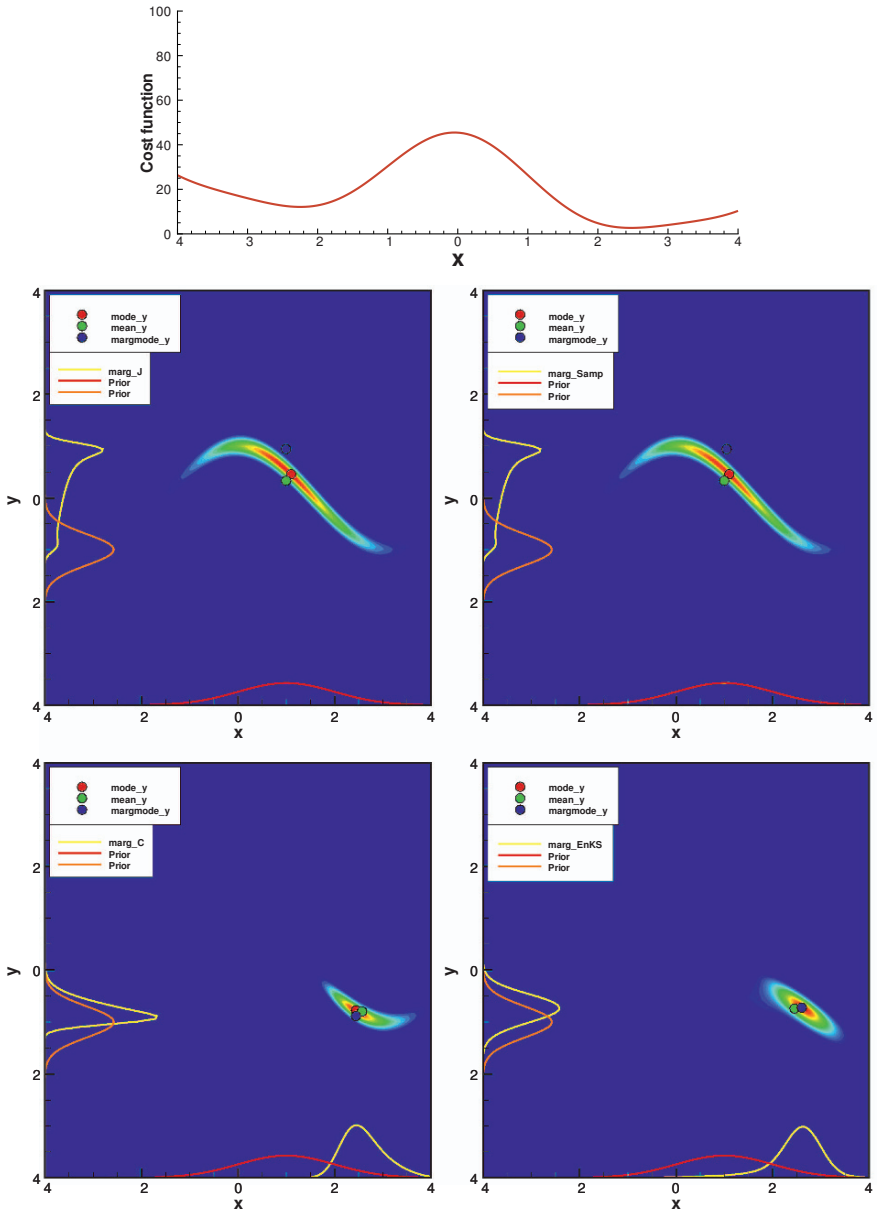

Fig. 10.6. Case 4b: Same as Fig. 10.5 but with high accuracy of the model

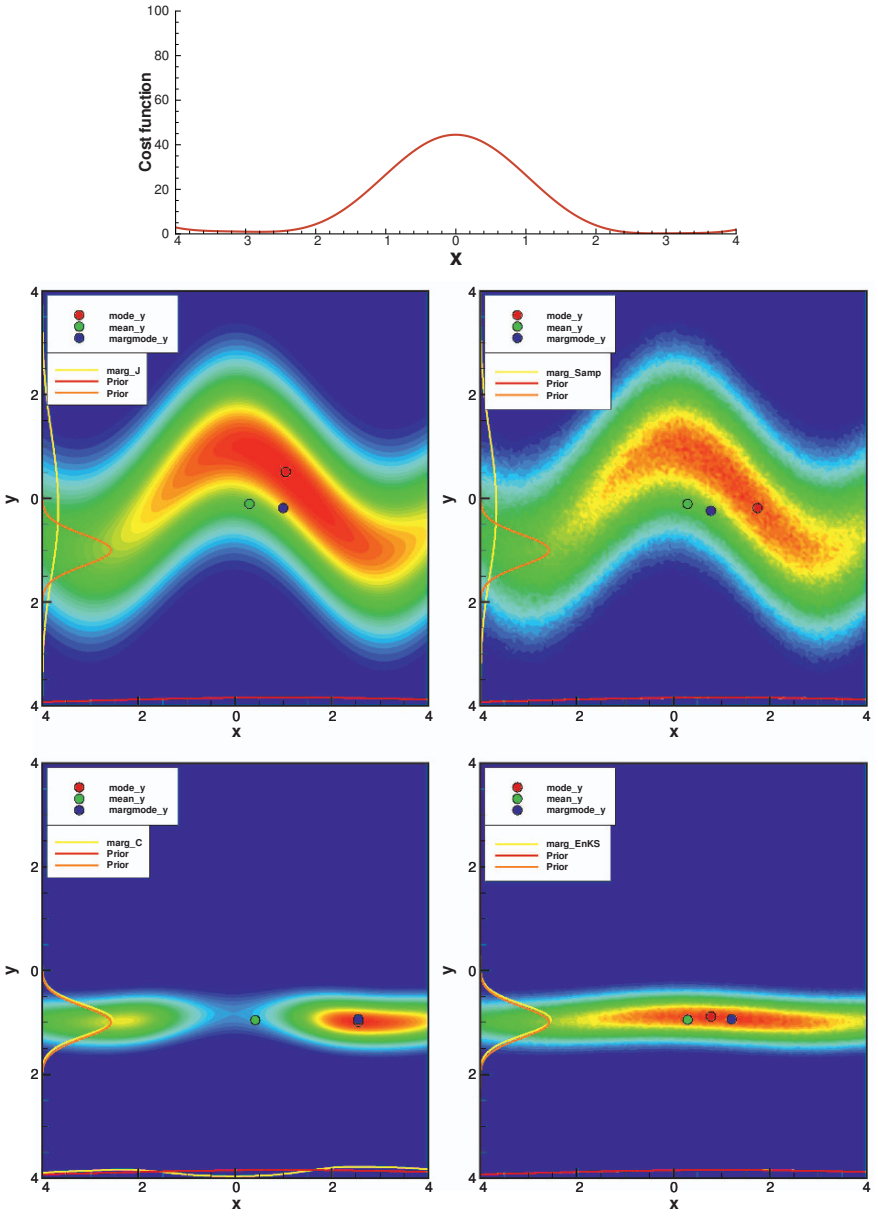

Fig. 10.7. Case 4c: Same as Fig. 10.5 but with weak penalty on first-guess which results in a bimodal pdf

the joint pdf estimated from the EnKS differs slightly from the analytical pdf. Thus, in this case the EnKS will give a slightly different estimate than an exact Bayesian solver, and this is due to the approximate linear update used.

In Case 3, shown in Fig. 10.4, we consider the function $F(x) = x^2(x^2 - 2)$, which leads to a cost function with both a local and global minimum. It is interesting to note that the introduction of model errors in this case leads to a predicted joint pdf which is unimodal. Thus, a unique solution is found and the EnKS solution contains some of the same characteristics as the exact analytical solution.

In Cases 4a–4c, shown in Figs. 10.5–10.7, we use the function $F(x) = \cos(x)$, and examine again the impact of the prior statistics for the model errors as well as errors in the initial guess. In Case 4a we set both the standard deviation for the model error and for the prior of x to one. Again the cost function contains an additional local minimum while the Bayesian approach leads to unimodal pdfs. The EnKS solution is fairly consistent. In Case 4b the model is very accurate, and again we converge towards a solution where the Bayesian estimate is close to the global minimum of the cost function. Note also that it is the rather accurate prior pdf which ensures that the joint pdf is unimodal. This is clearly illustrated in Case 4c where a low accuracy on the prior for x is used. In this case the joint conditional pdf has a bimodal structure and the mean falls between the peaks in the pdf and is not useful as an estimator. On the other hand, both the modes of the conditional joint and marginal pdfs provide realistic and similar estimates. The EnKS has a problem in this case and is not capable of reproducing the bimodal structure. It also provides a solution which has a fairly low probability.

10.5 Discussion

This chapter has considered the use of the EnKS as an optimization or parameter estimation method for nonlinear mappings. There is a clear analogy between this problem and the analysis step used in traditional data assimilation problems; e.g. if we consider the variable x, to be an initial state, and y to be a prediction by the nonlinear model, then this becomes analogous to the standard EnKS analysis step where the observation of y is assimilated. Alternatively, if we consider x to be the prediction at a certain time, and y to be a nonlinear measurement at the same time, related to x through an equation like (10.16) with $\boldsymbol{\alpha}$ replaced with x, then these examples resembles the EnKF update step using a nonlinear measurement functional.

Thus, it is clear that the EnKF and EnKS can handle certain levels of nonlinearity in both the model prediction and measurement functional. Even if the prior ensemble is non-Gaussian the ensemble methods will in many cases provide an updated ensemble having a realistic pdf. When the prior ensemble is non-Gaussian, the analyzed ensemble will inherit some of the non-Gaussian structures. On the other hand, it is also possible to make the EnKS and EnKF

fail completely; e.g. if the weight on the prior is low and a multimodal pdf develops, this may result in non-physical solutions.

From the analytical (left columns) and ensemble representation (right columns) of the joint pdfs in Figs 10.1–10.7, it is clear that the unconditioned joint and marginal pdfs are indistinguishable in all the cases. This illustrates that the stochastic ensemble integration which solves Kolmogorov's equation (4.34) gives the same result as the multiplication of the prior pdf with the transition density, as is expected. Note also that, while Kolmogorov's equation provides only the marginal densities, the ensemble integration allows for computation of the joint pdf if we track ensemble members in time; i.e we can evaluate the joint density from the pairs of points (x^l, y^l) where $l = 1, \ldots, N$.

11

Sampling strategies for the EnKF

The purpose of this Chapter is to present some algorithms for generating ensemble members, and model and measurement perturbations. There is a number of simulation methods available for generation of random realizations with different kinds of statistical properties, and we refer to the text books by *Lantuéjoul* (2002) and *Chilés* (1999) for further information. It is also shown that by selecting the initial ensemble, the model noise and the measurement perturbations wisely, it is possible to achieve a significant improvement in the EnKF results, without increasing the size of the ensemble.

11.1 Introduction

The ensemble methods use Monte Carlo sampling for generation of the initial ensemble, the model noise and the measurement perturbations. When defining an ensemble of realizations we need to specify the statistical properties of the distribution we are sampling from. In particular we need to ensure that the smoothness properties of the realizations are realistic for the physical variables they represent. The smoothness of a realization can be described by a covariance function or even better by a quantity named the variogram. For a field where the smoothness is independent of position, the variogram becomes

$$\gamma(\boldsymbol{h}) = C(0) - C(\boldsymbol{h}), \tag{11.1}$$

where $C(\boldsymbol{h})$ is the covariance of points located a distance $|\boldsymbol{h}|$ apart. It is easy to show that $\gamma(0) = 0$, $\gamma(\boldsymbol{h}) \geq 0$ and $-\gamma(\boldsymbol{h}) = \gamma(\boldsymbol{h})$. An extensive discussion of the variogram and its use in geostatistics is given in *Wackernagel* (1998).

Typical variograms are shown in Fig. 11.1 for a field with exponential, spherical and Gaussian covariance functions. The exponential covariance function is defined as

$$C_{\exp}(\boldsymbol{h}) \propto \exp\left(-\frac{|\boldsymbol{h}|}{a}\right) \tag{11.2}$$

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_11,

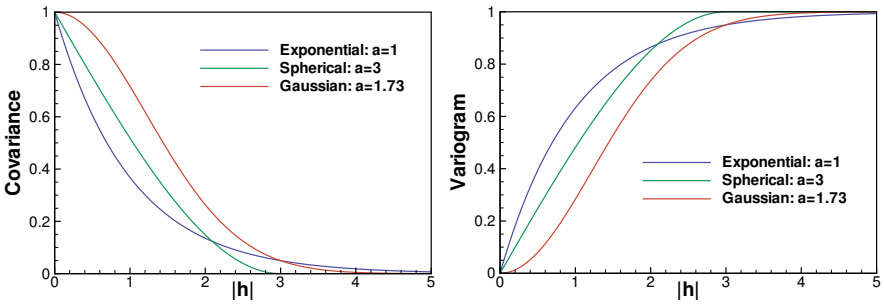

Fig. 11.1. The left plot shows exponential, spherical and Gaussian covariance functions and the right plot shows the corresponding variograms

with a being a de-correlation length. Note that the exponential correlation function is continuous but not differentiable at the origin. The spherical correlation function is given by

$$C_{\text{sphere}}(\boldsymbol{h}) = \begin{cases} 1 - 1.5|\boldsymbol{h}|/a + 0.5|\boldsymbol{h}|^3/a^3 & \text{for } 0 \leq |\boldsymbol{h}| \leq a \\ 0 & \text{for } |\boldsymbol{h}| > a, \end{cases} \tag{11.3}$$

where again a defines the de-correlation length. A Gaussian correlation function is given by

$$C_{\text{gauss}}(\boldsymbol{h}) \propto \exp\left(-\frac{|\boldsymbol{h}|^2}{a^2}\right). \tag{11.4}$$

We can define the range of the covariance functions as the distance where the covariance has a significant value. For the spherical covariance function the range is equal to a, while for the exponential and Gaussian it is common to define the ranges as $3a$ and $\sqrt{3}a$.

From the behaviour of the variograms when $|\boldsymbol{h}|$ approaches zero, it is clear that the Gaussian variogram corresponds to realizations that are rather smooth, while the exponential variogram corresponds to fields with more noisy behaviour. The spherical covariance functions corresponds to realizations with smoothness located somewhere between the exponential and Gaussian.

When simulating random fields, we need to know the statistical properties of the fields we are sampling to ensure that the realizations are physically acceptable for the process or variable they are meant to represent.

11.2 Simulation of realizations

The problem is now to simulate N realizations $\psi_i(\boldsymbol{x})$ for $i = 1 \ldots N$, which has zero mean and covariance given by $C_{\psi\psi}(\boldsymbol{x}_1, \boldsymbol{x}_2)$. The following procedure can be used to compute smooth random fields with mean equal to zero,

variance equal to one, and a specified covariance that determines the smoothness of the fields. The algorithm is an extension of the one presented in the Appendix of *Evensen* (1994b). We have used a Gaussian covariance function that makes sense in ocean simulations where smooth realizations are used. The method has some resemblance with the spectral method described by *Lantuéjoul* (2002) but uses a fast Fourier transform and exploits that the covariance matrix is diagonal in the Fourier space.

11.2.1 Inverse Fourier transform

Let $\psi = \psi(x, y)$ be a continuous field, which can be described by its Fourier transform

$$\psi(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \widehat{\psi}(\boldsymbol{k}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}} d\boldsymbol{k}. \tag{11.5}$$

We are using an $n_x \times n_y$ grid. Further, we define $\boldsymbol{k} = (\kappa_l, \lambda_p)$, where l and p are integer indices and κ_l and λ_p are wave numbers in the x and y directions, respectively. We now get a discrete version of (11.5),

$$\psi(x_n, y_m) = \sum_{l,p} \widehat{\psi}(\kappa_l, \lambda_p) e^{i(\kappa_l x_n + \lambda_p y_m)} \Delta\boldsymbol{k}, \tag{11.6}$$

where $x_n = n\Delta x$ and $y_m = m\Delta y$. For the wave numbers, we have

$$\kappa_l = \frac{2\pi l}{x_{n_x}} = \frac{2\pi l}{n_x \Delta x}, \tag{11.7}$$

$$\lambda_p = \frac{2\pi p}{y_{n_y}} = \frac{2\pi p}{n_y \Delta y}, \tag{11.8}$$

$$\Delta\boldsymbol{k} = \Delta\kappa\Delta\lambda = \frac{(2\pi)^2}{n_x n_y \Delta x \Delta y}. \tag{11.9}$$

11.2.2 Definition of Fourier spectrum

In *Evensen* (1994a) the following Gaussian form were used for the Fourier coefficients,

$$\widehat{\psi}(\kappa_l, \lambda_p) = \frac{c}{\Delta\boldsymbol{k}} e^{-(\kappa_l^2 + \lambda_p^2)/r^2} e^{2\pi i \phi_{l,p}}, \tag{11.10}$$

where $\phi_{l,p} \in [0, 1]$ is a uniformly distributed random number that introduces a random phase shift. With increasing l and p the wave numbers κ_l and λ_p will give an exponentially decreasing contribution, and large wave numbers corresponding to small scales are penalized. This choice of Fourier coefficients leads to isotropic covariances for the simulated fields, i.e. the smoothness is the same in all directions.

Here we have used the property that the Fourier transform of the Gaussian function (11.4) also becomes a Gaussian function. Clearly we can define

other Fourier coefficients, e.g. corresponding to the exponential or spherical covariances if this is what we want to simulate.

A further extension of this algorithm to account for asymmetrical and rotated covariance functions is straight-forward. Defining de-correlation lengths for the principal directions in the Fourier space as r_1 and r_2, and a rotation angle as θ, we can define

$$a_{11} = \frac{\cos^2(\theta)}{r_1^2} + \frac{\sin^2(\theta)}{r_2^2}, \tag{11.11}$$

$$a_{22} = \frac{\sin^2(\theta)}{r_1^2} + \frac{\cos^2(\theta)}{r_2^2}, \tag{11.12}$$

$$a_{12} = \left(\frac{1}{r_2^2} - \frac{1}{r_1^2}\right)\cos(\theta)\sin(\theta), \tag{11.13}$$

and the Fourier coefficients as

$$\widehat{\psi}(\kappa_l, \lambda_p) = \frac{c}{\boldsymbol{\Delta k}} e^{-\left(a_{11}\kappa_l^2 + 2a_{12}\kappa_l\lambda_p + a_{22}\lambda_p^2\right)} e^{2\pi i \phi_{l,p}}. \tag{11.14}$$

This Fourier spectrum has different scales in the two principal directions and the principal direction is rotated an angle θ. With $r_1 = r_2 = r$ this formula reduces to (11.10).

Now, (11.14) may be inserted into (11.6), and we get

$$\psi(x_n, y_m) = c\sqrt{\boldsymbol{\Delta k}} \sum_{l,p} e^{-\left(a_{11}\kappa_l^2 + 2a_{12}\kappa_l\lambda_p + a_{22}\lambda_p^2\right)} e^{2\pi i \phi_{l,p}} e^{i(\kappa_l x_n + \lambda_p y_m)}, \tag{11.15}$$

for the inverse Fourier transform that defines the random fields.

It should be noted that we want (11.15) to produce real fields only. Thus, when the summation over l, p is performed, all the imaginary contributions must add up to zero. This condition is satisfied whenever

$$\widehat{\psi}(\kappa_l, \lambda_p) = \widehat{\psi}^*(\kappa_{-l}, \lambda_{-p}), \tag{11.16}$$

where the asterisk denotes complex conjugate, and in addition

$$\text{Im } \widehat{\psi}(\kappa_0, \lambda_0) = 0. \tag{11.17}$$

11.2.3 Specification of covariance and variance

The formula (11.15) can be used to generate an ensemble of random fields with a covariance determined by the parameters c, r_1 and r_2.

An expression for the covariance is given by

$$\overline{\psi(x_1, y_1)\psi(x_2, y_2)} = (\boldsymbol{\Delta k})^2 \sum_{l,p,r,s} \overline{\widehat{\psi}(\kappa_l, \lambda_p)\widehat{\psi}(\kappa_r, \lambda_s)} e^{i(\kappa_l x_1 + \lambda_p y_1 + \kappa_r x_2 + \lambda_s y_2)} \tag{11.18}$$

By using (11.16), and by noting that the summation runs over both positive and negative r and s, we may insert the complex conjugate instead, i.e.

$$\begin{aligned}\overline{\psi(x_1,y_1)\psi(x_2,y_2)} &= (\Delta \boldsymbol{k})^2 \sum_{l,p,r,s} \overline{\widehat{\psi}(\kappa_l,\lambda_p)\widehat{\psi}^*(\kappa_r,\lambda_s)} e^{i(\kappa_l x_1-\kappa_r x_2+\lambda_p y_1-\lambda_s y_2)} \\ &= c^2 \sum_{l,p,r,s} e^{-\left(a_{11}(\kappa_l^2+\kappa_r^2)+2a_{12}(\kappa_l\lambda_p+\kappa_r\lambda_s)+a_{22}(\lambda_p^2+\lambda_s^2)\right)} \\ &\qquad \overline{e^{2\pi i(\phi_{l,p}-\phi_{r,s})}} e^{i(\kappa_l x_1-\kappa_r x_2+\lambda_p y_1-\lambda_s y_2)}.\end{aligned} \tag{11.19}$$

We assume that the fields are uncorrelated in wave space. Thus, there is only a distance dependence for the covariance, and the statistical properties of the simulated fields will be independent of the position. We can then set $l = r$ and $p = s$, and the above expression becomes

$$\begin{aligned}&\overline{\psi(x_1,y_1)\psi(x_2,y_2)} \\ &\quad = c^2 \sum_{l,p} e^{-2\left(a_{11}\kappa_l^2+2a_{12}\kappa_l\lambda_p+a_{22}\lambda_p^2\right)} e^{i(\kappa_l(x_1-x_2)+\lambda_p(y_1-y_2))}.\end{aligned} \tag{11.20}$$

The variance at the location (x, y), should be equal to 1, and from this equation we then get

$$\overline{\psi(x,y)\psi(x,y)} = 1 = c^2 \sum_{l,p} e^{-2\left(a_{11}\kappa_l^2+2a_{12}\kappa_l\lambda_p+a_{22}\lambda_p^2\right)}. \tag{11.21}$$

This equation is invariant with respect to θ and can therefore be expressed with $\theta = 0$ as

$$1 = c^2 \sum_{l,p} e^{-2\left(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2\right)}, \tag{11.22}$$

and we can solve for c.

Further, we define de-correlation lengths r_x and r_y for the spatial fields in the two principal directions, and we require the covariance along the principal directions corresponding to both distances r_x and r_y to be equal to e^{-1}. Thus, in (11.20) we set $\theta = 0$ and evaluate $\overline{\psi(x_1+r_x,y_1)\psi(x_1,y_1)}$ and $\overline{\psi(x_1,y_1+r_y)\psi(x_1,y_1)}$, which both should equal e^{-1}, to get

$$e^{-1} = c^2 \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \cos(\kappa_l r_x), \tag{11.23}$$

$$e^{-1} = c^2 \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \cos(\lambda_p r_y). \tag{11.24}$$

By inserting for c^2 from (11.22), we get

$$e^{-1} = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \cos(\kappa_l r_x) / \sum_{l,p} e^{-2\left(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2\right)}, \tag{11.25}$$

$$e^{-1} = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \cos(\lambda_p r_y) / \sum_{l,p} e^{-2\left(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2\right)}. \tag{11.26}$$

This is a system of two nonlinear equations which can be solved for r_1 and r_2. Thereafter we can compute c from (11.22). The formula (11.15) can then be used to simulate an ensemble of random fields with variance 1 and covariance determined by the de-correlation lengths r_x and r_y and the rotation angle θ.

Using that the denominator appearing in (11.25) and (11.26) is always positive and larger than zero, we can write the two conditions as

$$F_1 = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} (\cos(\kappa_l r_x) - e^{-1}) = 0, \tag{11.27}$$

$$F_2 = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} (\cos(\lambda_p r_y) - e^{-1}) = 0. \tag{11.28}$$

These are easily solved using a Newton method, where we also need the derivatives

$$\frac{\partial F_1}{\partial r_1} = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \frac{4\kappa_l^2}{r_1^3} (\cos(\kappa_l r_x) - e^{-1}), \tag{11.29}$$

$$\frac{\partial F_1}{\partial r_2} = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \frac{4\lambda_p^2}{r_2^3} (\cos(\kappa_l r_x) - e^{-1}), \tag{11.30}$$

$$\frac{\partial F_2}{\partial r_1} = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \frac{4\kappa_l^2}{r_1^3} (\cos(\lambda_p r_y) - e^{-1}), \tag{11.31}$$

$$\frac{\partial F_2}{\partial r_2} = \sum_{l,p} e^{-2(\kappa_l^2/r_1^2+\lambda_p^2/r_2^2)} \frac{4\lambda_p^2}{r_2^3} (\cos(\lambda_p r_y) - e^{-1}). \tag{11.32}$$

An efficient approach for finding the inverse transform in (11.15) is to apply a two-dimensional fast Fourier transform (FFT). The inverse FFT is calculated on a grid that is a few characteristic lengths larger than the computational domain to ensure non-periodic fields (*Evensen*, 1994b).

To summarize, we are now able to simulate two-dimensional pseudo random fields with variance equal to one and a prescribed anisotropic covariance.

11.3 Simulating correlated fields

A simple formula can be used to introduce correlations between the simulated realizations. Such correlated fields are useful in ocean and atmospheric models where there can be vertical correlations between levels or layers in the model.

As an example, a simulated temperature field at two nearby depths will be correlated if there is strong vertical mixing such as within the ocean mixed layer. Another example relates to the simulation of model errors where we expect there to be a finite time correlation.

The equation

$$\psi_k(\boldsymbol{x}) = \rho\psi_{k-1}(\boldsymbol{x}) + \sqrt{1-\rho^2}w_k(\boldsymbol{x}), \tag{11.33}$$

can be used for simulating correlated realizations. Here we assume that $w_k(\boldsymbol{x})$ is a random realization sampled from a distribution with zero mean and variance equal to one, while $\psi_{k-1}(\boldsymbol{x})$ is the previous realization, to which $\psi_k(\boldsymbol{x})$ should be correlated. The $w_k(\boldsymbol{x})$ fields are typically generated by an algorithm similar to the one described in the previous section. Thus, starting with $\psi_1(\boldsymbol{x}) = w_1(\boldsymbol{x})$ the formula (11.33) can be used to recursively simulating the correlated fields.

The coefficient $\rho \in [0, 1)$ determines the correlation of the stochastic forcing, e.g. $\rho = 0$ generates a white sequence, while $\rho = 1$ will remove the stochastic forcing and we obtain a random field identical to initial guess $\psi_0(\boldsymbol{x}) = w_0(\boldsymbol{x})$. More generally the covariance between $\psi_i(\boldsymbol{x})$ and $\psi_j(\boldsymbol{x})$ becomes

$$\overline{\psi_i(\boldsymbol{x})\psi_j(\boldsymbol{x})} = \rho^{|i-j|} = \exp\big(\ln\rho\,|i-j|\big). \tag{11.34}$$

The variance of the simulated fields will be equal to one and we obtain a sequence of random fields with an exponential variogram where $a = -1/\ln\rho$.

11.4 Improved sampling scheme

Based on the works by *Pham* (2001) and *Nerger et al.* (2005) it should be possible to introduce some improvements in the EnKF by using a more clever sampling for the initial ensemble, the model noise, and the measurement perturbations. We will now examine a sampling scheme that effectively produces results similar to those obtained in the SEIK filter by *Pham* (2001). The scheme does not add significantly to the computational cost of the EnKF and may lead to a significant improvement in the results.

The EnKF computes the update as a combination of the predicted ensemble realizations. Thus, the analysis is contained in the space spanned by the original ensemble, and clearly it will be dependent on the properties of the ensemble. In general one can say that the ensemble matrix $\boldsymbol{A}^{\mathrm{f}}$ should satisfy the following:

1. The ensemble realizations should be realistic and physically acceptable fields.
2. The rank of the ensemble should be $\mathrm{rank}(\boldsymbol{A}^{\mathrm{f}}) = \min(n, N)$
3. The condition number of the ensemble, defined as the ratio between the largest and smallest singular value, should be small.

The first condition ensures that the realizations are sampled with the correct spatial variability and smoothness, and is required for a nonlinear model to provide realistic results. The second condition just means that the ensemble spans an N-dimensional space, while the last condition says something about the linear independence between the ensemble members.

11.4.1 Theoretical foundation

In the SEIK filter by *Pham* (2001) an algorithm is used where the initial ensemble is sampled from the first dominant eigenvectors of the error covariance matrix $\boldsymbol{C}_{\psi\psi}$. The algorithm introduces a maximum rank and conditioning of the ensemble matrix, and ensures that the ensemble provides a best possible representation of the error covariance matrix for a given ensemble size.

We now start by defining an error covariance matrix $\boldsymbol{C}_{\psi\psi}$. Given $\boldsymbol{C}_{\psi\psi}$, we can compute the eigenvalue decomposition

$$\boldsymbol{C}_{\psi\psi} = \boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^{\mathrm{T}}, \tag{11.35}$$

where the matrices $\boldsymbol{Z}$ and $\boldsymbol{\Lambda}$ contain the eigenvectors and eigenvalues of $\boldsymbol{C}_{\psi\psi}$.

The full rank error covariance matrix can be approximated using its ensemble representation $\boldsymbol{C}^{\mathrm{e}}_{\psi\psi} \simeq \boldsymbol{C}_{\psi\psi}$,

$$\boldsymbol{C}^{\mathrm{e}}_{\psi\psi} = \frac{1}{N-1}\boldsymbol{A}'(\boldsymbol{A}')^{\mathrm{T}} \tag{11.36}$$

$$= \frac{1}{N-1}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^{\mathrm{T}} \tag{11.37}$$

$$= \frac{1}{N-1}\boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^{\mathrm{T}}. \tag{11.38}$$

This expression is similar to the definition (9.14) when excluding the time dimension and using a discrete representation $\boldsymbol{\psi}$, of the state. Here, $\boldsymbol{A}'$ contains the ensemble perturbations, and is defined as a discrete version of the formula (9.13), while $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{V}^{\mathrm{T}}$ result from a singular value decomposition[1], and contain the singular vectors and singular values of $\boldsymbol{A}'$. In the limit when the ensemble size goes to infinity the n singular vectors in $\boldsymbol{U}$ will converge towards the n eigenvectors in $\boldsymbol{Z}$ and the square of the singular values $\boldsymbol{\Sigma}^2$, divided by $N-1$, will converge towards the eigenvalues $\boldsymbol{\Lambda}$.

Thus, there are two strategies for defining an accurate ensemble approximation $\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}$, of $\boldsymbol{C}_{\psi\psi}$. The first approach is the standard Monte Carlo method

[1] The singular value decomposition of a rectangular matrix $\boldsymbol{A} \in \Re^{m\times n}$ is $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$ where $\boldsymbol{U} \in \Re^{m\times m}$ and $\boldsymbol{V} \in \Re^{n\times n}$ are orthogonal matrices and $\boldsymbol{\Sigma} \in \Re^{m\times n}$ contains the $p = \min(m,n)$ singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ on the diagonal. Further, $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V} = \boldsymbol{\Sigma}$. Note that numerical algorithms for computing the SVD when $m > n$ often offers to compute only the first p singular vectors in $\boldsymbol{U}$ since the remaining singular vectors (columns in $\boldsymbol{U}$) are normally not needed. However, for the expression $\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{I}$ to be true the full $\boldsymbol{U}$ must be used.

where we increase the ensemble size N, by sampling additional model states and adding these to the ensemble. As long as the addition of new ensemble members increases the space spanned by the overall ensemble, this approach will result in an ensemble covariance $\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}$ that is a more accurate representation of $\boldsymbol{C}_{\psi\psi}$.

Alternatively we can improve the rank/conditioning of the ensemble by ensuring that the first N singular vectors in $\boldsymbol{U}$ are similar to the N first eigenvectors in $\boldsymbol{Z}$. The absolute error in the representation $\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}$ of $\boldsymbol{C}_{\psi\psi}$ will be smaller for ensembles sampled in the space spanned by the first N singular vectors in $\boldsymbol{U}$ than for Monte Carlo ensembles of ensemble size N. In other words, we want to generate $\boldsymbol{A}$ such that $\mathrm{rank}(\boldsymbol{A}) = N$ and the condition number, defined as the ratio between the largest and smallest singular values, $\kappa_2(\boldsymbol{A}) = \sigma_1(\boldsymbol{A})/\sigma_N(\boldsymbol{A})$, is minimal. This second approach has a flavour of quasi-random sampling, which ensures better convergence with increasing sample size. That is, we choose ensemble members that have less linear dependence. Note that the constraint of generating physically acceptable fields implies that in some cases more than N singular vectors must be used when defining the sampling space, to avoid sampling too smooth realizations.

11.4.2 Improved sampling algorithm

For most applications the size of $\boldsymbol{C}_{\psi\psi}$ is too large to allow for the direct computation of eigenvectors. An alternative algorithm for generating an N-member ensemble with better conditioning, is to first generate a larger "start ensemble" of size αN, with α being an integer larger than one, and then to resample N members along the first βN dominant singular vectors of the large start ensemble. Here $\beta \in (1, \ldots, \alpha)$ is an integer that determines the number of singular vectors used when resampling the new improved ensemble. The algorithm from *Evensen* (2004) uses $\beta = 1$, and may in some cases sample realizations based on a too small set of singular vectors, resulting in too smooth and unphysical realizations.

Given a large ensemble of realizations $\widehat{\boldsymbol{A}}' \in \Re^{n\times\alpha N}$ we compute the singular value decomposition

$$\widehat{\boldsymbol{A}}' = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{V}}^{\mathrm{T}}, \tag{11.39}$$

with $\widehat{\boldsymbol{U}} \in \Re^{n\times n}$, $\widehat{\boldsymbol{\Sigma}} \in \Re^{n\times\alpha N}$, and $\widehat{\boldsymbol{V}} \in \Re^{\alpha N\times\alpha N}$.

The new ensemble can then be sampled from

$$\boldsymbol{A}' = \widehat{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\Theta}^{\mathrm{T}}, \tag{11.40}$$

where $\widetilde{\boldsymbol{\Sigma}} \in \Re^{n\times\beta N}$ contains the first βN singular values multiplied by $\sqrt{(\beta N)/(\alpha N)}$ to obtain the correct variance, i.e.,

$$\widetilde{\boldsymbol{\Sigma}}(:,:) = \sqrt{\frac{\beta}{\alpha}}\widehat{\boldsymbol{\Sigma}}(:, 1:\beta N), \tag{11.41}$$

defines the energy spectrum that penalizes the high wave numbers (or singular vectors). Note that we for simplicity have assumed that both αN and βN are less than the dimension of the state vector n.

The random matrix $\boldsymbol{\Theta} \in \Re^{N \times \beta N}$ has orthonormal rows and can be generated by extracting the first N rows of the random orthogonal matrix $\boldsymbol{\Theta} \in \Re^{\beta N \times \beta N}$ that is computed using the algorithm described in Sec. 11.6. Thus, each row in $\boldsymbol{\Theta}$ defines a linear combination of the scaled singular vectors that results in a random realization.

11.4.3 Properties of the improved sampling

The sampling scheme has some interesting properties.

1. When αN is large the singular vectors in $\widehat{\boldsymbol{U}}$ converge towards the eigenvectors $\boldsymbol{Z}$ of the exact covariance matrix $\boldsymbol{C}_{\psi\psi}$. Furthermore, the scaled product of singular values, $\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Sigma}}^{\mathrm{T}}/(\alpha N)$, will converge to the exact eigenvalues of $\boldsymbol{C}_{\psi\psi}$. Thus, it is important to choose α sufficiently large to ensure a good estimate of the true eigenvectors and eigenvalues.
2. By sampling from the space of the βN dominant singular vectors in $\widehat{\boldsymbol{U}}$, using the scaled truncated spectrum as stored in $\widetilde{\boldsymbol{\Sigma}}$, and using an orthogonal random matrix $\boldsymbol{\Theta}$, we generate realizations that are all contained in the βN dimensional subspace defined by the first βN modes in $\widehat{\boldsymbol{U}}$.
3. The samples are approximately orthogonal on the weighted or scaled inner product defined by

$$< \boldsymbol{a}, \boldsymbol{b} >= \frac{1}{\beta N}(\boldsymbol{a})^{\mathrm{T}} \boldsymbol{C}_{\psi\psi}^{-1} \boldsymbol{b}. \tag{11.42}$$

When we insert the eigenvalue factorization for $\boldsymbol{C}_{\psi\psi}$ and the singular value decomposition for $\boldsymbol{A}'$ we obtain

$$\begin{aligned}
< \boldsymbol{A}', \boldsymbol{A}' > &= \frac{1}{\beta N} \boldsymbol{\Theta} \widetilde{\boldsymbol{\Sigma}}^{\mathrm{T}} \widehat{\boldsymbol{U}}^{\mathrm{T}} \left(\boldsymbol{Z} \boldsymbol{\Lambda} \boldsymbol{Z}^{\mathrm{T}} \right)^{-1} \widehat{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\Theta}^{\mathrm{T}} \\
&= \frac{1}{\beta N} \boldsymbol{\Theta} \widetilde{\boldsymbol{\Sigma}}^{\mathrm{T}} \widehat{\boldsymbol{U}}^{\mathrm{T}} \left(\widehat{\boldsymbol{U}} \boldsymbol{\Lambda} \widehat{\boldsymbol{U}}^{\mathrm{T}} \right)^{-1} \widehat{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\Theta}^{\mathrm{T}} \\
&= \frac{1}{\beta N} \boldsymbol{\Theta} \widetilde{\boldsymbol{\Sigma}}^{\mathrm{T}} \boldsymbol{\Lambda}^{-1} \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\Theta}^{\mathrm{T}} \\
&\approx \frac{\alpha N}{\beta N} \boldsymbol{\Theta} \widetilde{\boldsymbol{\Sigma}}^{\mathrm{T}} \left(\widehat{\boldsymbol{\Sigma}} \widehat{\boldsymbol{\Sigma}}^{\mathrm{T}} \right)^{-1} \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\Theta}^{\mathrm{T}} \\
&\approx \boldsymbol{\Theta} \widetilde{\boldsymbol{\Sigma}}^{\mathrm{T}} \left(\widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{\Sigma}}^{\mathrm{T}} \right)^{-1} \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\Theta}^{\mathrm{T}} \\
&= \boldsymbol{\Theta} \boldsymbol{I}_{\beta N} \boldsymbol{\Theta}^{\mathrm{T}} \\
&= \boldsymbol{I} \in \Re^{N \times N},
\end{aligned} \tag{11.43}$$

where we have used (11.41).

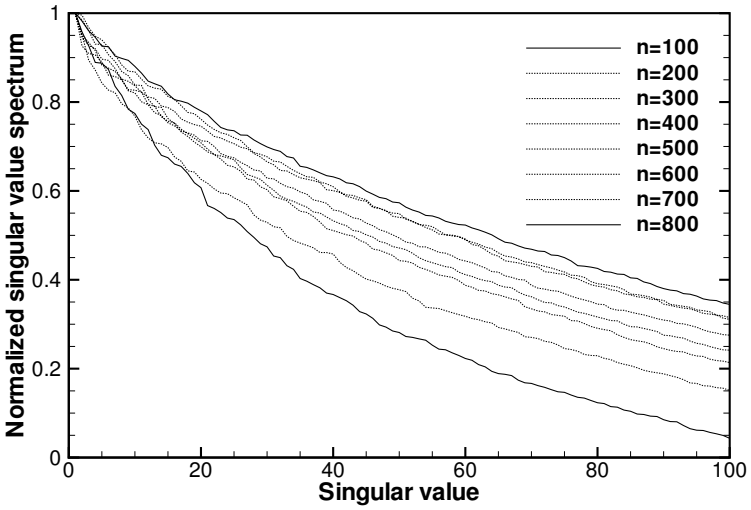

Fig. 11.2. The plot shows the normalized singular values of ensembles which are generated using start ensembles of different sizes, with the lower line corresponding to the start ensemble of 100 members. Clearly, the condition of the ensemble improves when a larger start ensemble is used

4. With $\beta = 1$ as in the original scheme, these realizations give a non-unique *low-rank best representation of the error covariance matrix*, but due to the truncation of scales, the realizations may be nonphysical and too smooth.
5. With $\beta > 1$ the realizations no longer provide the *low-rank best representation of the error covariance matrix*, but they are still orthogonal on the inner product (11.42) and with β sufficiently large it is possible to generate realizations that include the physical scales of importance.

As long as the initial ensemble is chosen large enough the algorithm just described will provide an ensemble that is similar to what is used in the SEIK filter, and the SVD algorithm has a lower computational cost than the explicit eigenvalue decomposition of $\boldsymbol{C}_{\psi\psi}$ when n is large.

Before the ensemble perturbation matrix $\boldsymbol{A}'$ is used, it is important to ensure that the mean is zero and the variance takes a value as specified. This correction can be applied by subtracting an eventual ensemble mean and then rescaling the ensemble members to obtain the correct variance. As will be seen below, ensuring that the ensemble has the correct mean and variance, makes a positive impact on the quality of the results. Note that the removal of the mean of the ensemble leaves the maximum possible rank of $\boldsymbol{A}'$ to be $N - 1$.

As an example, a 100-member ensemble has been generated using start ensembles of $100, 200, \ldots, 800$ members. The size of the one-dimensional model state is 1001 and the characteristic length scale of the solution is 4 grid cells.

The singular values (normalized to the first singular value) for the resulting ensemble is plotted in Fig. 11.2 for the different sizes of start ensemble. Clearly, there is a benefit of using this sampling strategy. The ratio between singular values 100 and 1, is 0.21 when standard sampling is used. With increasing size of the start ensemble the conditioning improves until it reaches 0.59 for 800 members in the start ensemble.

11.5 Model and measurement noise

We now assume a linear model operator defined by the full rank matrix $\boldsymbol{G}$. With zero model noise, the ensemble at a later time t_k, can be written as

$$\boldsymbol{A}_k = \boldsymbol{G}^k \boldsymbol{A}_0. \tag{11.44}$$

Thus, the rank introduced in the initial ensemble will be preserved as long as $\boldsymbol{G}$ is full rank, and $\boldsymbol{A}_k$ will span the same space as $\boldsymbol{A}_0$.

With system noise the time evolution of the ensemble becomes

$$\boldsymbol{A}_k = \boldsymbol{G}^k \boldsymbol{A}_0 + \sum_{i=1}^{k} \boldsymbol{G}^{k-i} \boldsymbol{Q}_i, \tag{11.45}$$

where $\boldsymbol{Q}_i$ denote the ensemble of model noise used at time t_i. Thus, the rank and conditioning of the ensemble will also depend on the rank and conditioning of the model noise introduced.

For a nonlinear model operator, $\boldsymbol{G}(\boldsymbol{\psi}, \boldsymbol{q})$, where $\boldsymbol{q}$ is the model noise, the evolution of the ensemble can be written as

$$\boldsymbol{A}_k = \boldsymbol{G}_k \left(\ldots \boldsymbol{G}_2 \left(\boldsymbol{G}_1 \left(\boldsymbol{A}_0, \boldsymbol{Q}_1 \right), \boldsymbol{Q}_2 \right) \ldots, \boldsymbol{Q}_k \right). \tag{11.46}$$

Using a nonlinear model there is no guarantee that the nonlinear transformations will preserve the rank of $\boldsymbol{A}$ and the introduction of wisely sampled model noise may be crucial to maintain an ensemble with good rank properties during the simulation. Thus, the same procedure as used when generating the initial ensemble can be used when simulating the system noise. This approach ensures that a maximum rank is introduced into the ensemble, and it may counteract any rank reduction introduced by the model operator.

The EnKS and EnKF analysis algorithms in (9.37) and (9.39) with $\boldsymbol{X}_j$ defined in (9.38), use perturbed measurements through $\boldsymbol{D}'_j$. The improved sampling procedure can then be used when generating the measurement perturbations. The improved sampling then leads to a better conditioning of the ensemble of perturbations and the ensemble covariance $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ becomes a better approximation of $\boldsymbol{C}_{\epsilon\epsilon}$. The impact of improved sampling of measurement perturbations is significant and will be demonstrated in the examples below.

11.6 Generation of a random orthogonal matrix

A orthogonal random matrix is best generated using the following procedure. Start with a matrix of random independent normal distributed numbers $\boldsymbol{Y} \in \Re^{N\times N}$, and compute the QR factorization

$$\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{R}, \tag{11.47}$$

where $\boldsymbol{Q} \in \Re^{N\times N}$ is random and orthogonal and $\boldsymbol{R} \in \Re^{N\times N}$ is upper triangular. The factorization is normally best computed using Householder reflections. The QR decomposition of a non-singular matrix is only unique if we require that the diagonal elements of $\boldsymbol{R}$ are all positive. Thus, we can define $\boldsymbol{\Xi}$ as a diagonal matrix with elements equal to -1 or 1, or with elements $\boldsymbol{\Xi}_{jj} = e^{i\theta_j}$ on the the unit circle in the case when $\boldsymbol{Y}$ is complex, and write

$$\boldsymbol{Q}' = \boldsymbol{Q}\boldsymbol{\Xi}, \tag{11.48}$$

$$\boldsymbol{R}' = \boldsymbol{R}\boldsymbol{\Xi}^{-1}, \tag{11.49}$$

and we have

$$\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{R} = \boldsymbol{Q}'\boldsymbol{R}'. \tag{11.50}$$

As suggested by *Mezzadri* (2007) we follow the QR decomposition by a multiplication of $\boldsymbol{Q}$ with the inverse of a diagonal matrix $\boldsymbol{\Xi} \in \Re^{N\times N}$, defined as

$$\mathrm{diag}(\boldsymbol{\Xi}) = \big(\boldsymbol{R}_{1,1}/|\boldsymbol{R}_{1,1}|, \ldots, \boldsymbol{R}_{N,N}/|\boldsymbol{R}_{N,N}|\big). \tag{11.51}$$

This procedure leads to a matrix $\boldsymbol{R}'$ where all diagonal elements are positive, and a unique random orthogonal matrix $\boldsymbol{Q}'$ that is shown by *Mezzadri* (2007) to be distributed with Haar measure.

11.7 Experiments

The impact of ensemble size and improved sampling is now discussed in some detail using the one-dimensional linear advection model from Sect. 4.1.3. The solution of this model is exactly known, which allows us to run realistic experiments with zero model errors to examine the impact of the sampling schemes used.

In most of the following experiments an ensemble size of 100 members is used. A larger start ensemble is used in many of the experiments to generate ensemble members and/or measurement perturbations that provide a better representation of the error covariance matrix. Otherwise, the experiments differ in the sampling of measurement perturbations and the analysis scheme used. In Fig. 4.1 an example is shown from one of the experiments.

Experiment	N	Sample fix	β_{ini}	β_{mes}	Residual	Std. dev.
A	100	F	1	1	0.762	0.074
B	100	T	1	1	0.759	0.053
C	100	T	2	1	0.715	0.065
D	100	T	4	1	0.683	0.062
E	100	T	6	1	0.679	0.071
H	100	T	6	30	0.627	0.053
I	100	T	1	30	0.706	0.060
B150	150	T	1	1	0.681	0.053
B200	200	T	1	1	0.651	0.061
B250	250	T	1	1	0.626	0.058

Table 11.1. Summary of experiments. The first column is the experiment name, in the second column N is the ensemble size used, then "Sample fix" is true or false and indicates if the sample mean and variance is corrected, β_{ini} is a number which defines the size of the start ensemble used for generating the initial ensemble as $\beta_{\text{ini}}N$, similarly β_{mes} denote the size of the start ensemble used for generating the measurement perturbations, followed by the analysis algorithm used. The two last columns contain the average RMS errors of the 50 simulations in each experiment and the standard deviation of these

11.7.1 Overview of experiments

Several experiments are carried out as listed in Table 11.1. For each of the experiments, 50 EnKF simulations are performed to allow for a statistical comparison. In each EnKF simulation, the only difference is the random seed used. Thus, every EnKF simulation has a different and random true state, first guess, initial ensemble, set of measurements, and measurement perturbations.

The standard version of the EnKF analysis scheme is used with a full rank matrix $\boldsymbol{C} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}$ that is factorized by computing the eigenvalue decomposition $\boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^{\mathrm{T}} = \boldsymbol{C}$, to get

$$\boldsymbol{C}^{-1} = \boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^{\mathrm{T}}, \tag{11.52}$$

where all matrices are of dimension $m \times m$. Thus, we solve the standard EnKF analysis (9.39) with the definition (9.38), where measurements are perturbed, i.e. at each assimilation time we compute

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^{\mathrm{T}}\big(\boldsymbol{D} - \mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\Big), \tag{11.53}$$

where we have dropped the update index j.

In all the experiments the residuals are computed as the Root Mean Square (RMS) errors of the difference between the estimate and the true solution taken over the complete space and time domain. For each of the experiments we have plotted the mean and standard deviation of the residuals from the 50 EnKF simulation in Fig. 11.3.

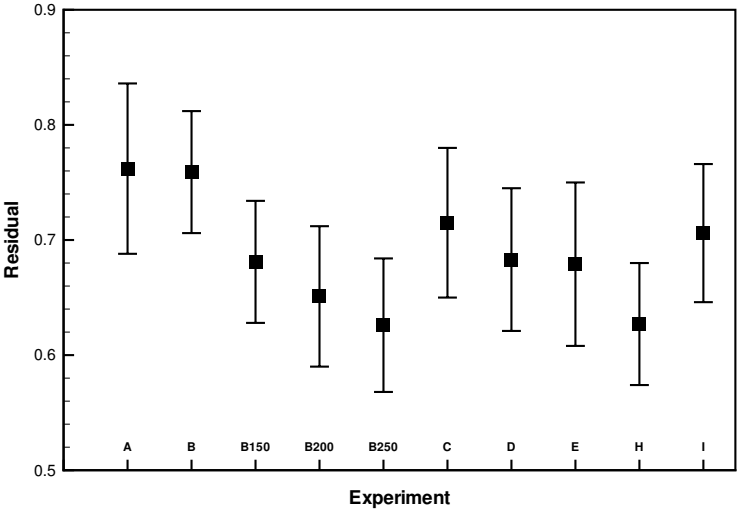

Fig. 11.3. Mean and standard deviation of the residuals from each of the experiments

It is also of interest to examine how well the predicted errors represent the actual residuals (RMS as a function of time). In the summary Figs. 11.4 and 11.5 we have plotted the average of the predicted errors from the 50 EnKF simulations as the thick full line. The thin full lines indicate the one standard deviation spread of the predicted errors from the 50 EnKF simulations. The average of the RMS errors from the 50 EnKF simulations is plotted as the thick dotted line, with the associated one standard deviation spread shown by the dotted thin lines.

Table 11.2 gives the probabilities that the average residuals from the experiments are equal, as computed from the Student's t-test. Probabilities lower than, say 0.5, indicate statistically that the distributions from two experiments are significantly different.

The further details of the different experiments are described below.

Exp. A is the pure Monte Carlo case using a start ensemble of 100 members where all random variables are sampled "randomly". Thus, the mean and variance of the initial ensemble and the measurement perturbations will fluctuate within the accuracy that can be expected using a 100 member sample size.

Exp. B is similar to *Exp. A* except that the sampled ensemble perturbations are corrected to have mean zero and the correct specified variance. The correction is applied by subtracting an eventual mean from the random sample and then dividing the members by the square root of the ensemble variance. As will be seen below, this correction leads to a small improve-

Exp	*B*	*B150*	*B200*	*B250*	*C*	*D*	*E*	*H*	*I*
A	**0.86**	0	0	0	0	0	0	0	0
B		0	0	0	0	0	0	0	0
B150			0.01	0	0.01	**0.86**	**0.86**	0	0.03
B200				0.04	0	0.01	0.04	0.04	0
B250					0	0	0	**0.91**	0
C						0.01	0.01	0	0.48
D							**0.75**	0	0.06
E								0	0.04
H									0

Table 11.2. Statistical probability that two experiments provide an equal mean for the residuals as computed using the Student's t-test. A probability close to one indicates that it is likely that the two experiments provide distributions of residuals with similar mean. The t-test numbers higher than 0.5 are printed in bold

ment in the assimilation results and is therefore used in all the following experiments. This experiment is also used as a reference case in the further discussion that illustrates the performance of the EnKF algorithm.

Exps. B150, B200 and *B250* are similar to *Exp. B* but use respectively ensemble sizes of 150, 200 and 250 members.

Exps. C, D and *E* are similar to *Exp. B* except that the start ensembles used to generate the initial 100 member ensemble contain respectively 200, 400 and 600 members. *Exp. E* is used as a reference case illustrating the impact of the improved initial sampling algorithm.

Exp. H examines the combined impact of improved sampling of both measurement perturbations and the initial ensemble. The results should be compared with those of *Exp. E* to examine the additional impact improved sampling of measurement perturbations.

Exp. I should be compared with *Exps. H* and *B*. It uses improved sampling of measurement perturbations but standard sampling for the initial conditions. Thus, comparing it with results from *Exp. B* gives the impact of improved sampling of measurement perturbations.

11.7.2 Impact from ensemble size

We now compare the experiments *Exps. B, B150, B200* and *B250* to evaluate the impact of ensemble size on the performance of the EnKF. From Fig. 11.3 it is seen that the residuals, as expected, are decreasing when the ensemble size is increased. In practical applications we are naturally limited by the number of ensemble members we can afford to run. However, from the central limit theorem, the accuracy in the EnKF estimate will improve proportionally to the square root of the ensemble size. In most published applications of the EnKF a typical ensemble size is around 100 members. This ensemble size is clearly much less than effective dimension of the solution space of many

dynamical models, but in many cases a so-called localization or local analysis computation is often used to effectively increase the dimension of the space where the solution is searched for (see Chap. 15).

When comparing the time evolution of the residuals and the estimated standard deviations for the *Exps. B, B150* and *B250* in Fig. 11.4, we observe that the residuals show a larger spread between the EnKF simulations than the estimated standard deviations. The estimated standard deviations are internally consistent between the simulations performed in each of the experiments. The residuals are also generally larger than the ensemble standard deviations, although there is a significant improvement observed due to the increase in ensemble size.

11.7.3 Impact of improved sampling for the initial ensemble

Using the procedure outlined in Sect. 11.4 several experiments are performed using start ensembles of 100–600 members to examine the impact of using an initial ensemble with better properties. The standard *Exp. B* is used as a reference while in the *Exps. C, D* and *E*, larger start ensembles of respectively 200, 400 and 600 members are used to generate the initial 100 member ensemble. In all the experiments discussed here we could sample the realizations from the first 100 singular vectors, thus $\beta = 1$ in (11.41).

In Fig. 11.3 it is seen that just doubling the size of the start ensemble to 200 members (*Exp. C*) has a significant positive effect on the results, and using a start ensemble of 400 members (*Exp. D*) leads to a further improvement. The use of an even larger start ensemble of 600 members (*Exp. E*) does not provide a statistically significant improvement over *Exp. D* in this particular application, with a rather small state space.

The time evolutions of the residuals and the estimated standard deviations for the *Exps. B* and *E* in Figs. 11.4 and 11.5, show the same trend as was found for the *Exps. B, B150* and *B250* above, where residuals show a larger spread between the simulations than the estimated standard deviations and the residuals are larger than the ensemble standard deviations. Some improvement is seen when going from *Exp. B* to *Exp. E* due to the improved sampling of the initial ensemble.

It was also found when comparing *Exps. B150* and *B200* with *Exp. E* that an ensemble size between 150 and 200 is needed in the standard EnKF to get similar improvement as was obtained with improved sampling of a 100 member initial ensemble, using a start ensemble of 600.

These experiments clearly show that the improved sampling is justified for the initial ensemble. It is computed once and the additional computational cost is marginal. Thus, the improved sampling could be utilized to apply the filter algorithm with a smaller ensemble size and less computing time than required in the normal EnKF algorithm while still obtaining a comparable result.

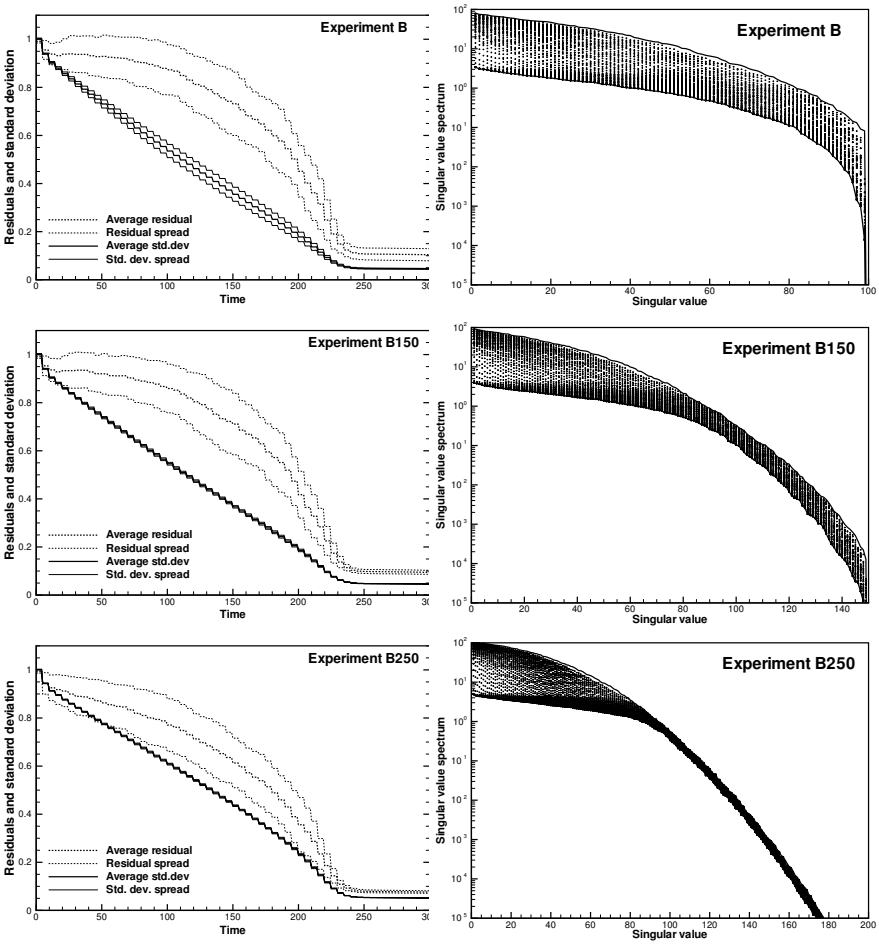

Fig. 11.4. RMS residuals and ensemble singular value spectra for some of the experiments. The left column shows the time evolution for RMS residuals *(dashed lines)* and estimated standard deviations *(full lines)*. The thick lines show the means over the 50 simulations and the thin lines show the means plus/minus one standard deviation. The right column shows the time evolution of the ensemble singular value spectra for the experiments

11.7.4 Improved sampling of measurement perturbations.

The *Exps. H* and *I* use the improved sampling of measurement perturbations with a large start ensemble of perturbations of 30 times the ensemble size. The impact of this improved sampling is illustrated by comparing the *Exp. I* with *Exps. B*, and then *Exp. H* with *Exp. E*, in Fig. 11.3. There is clearly a signif-

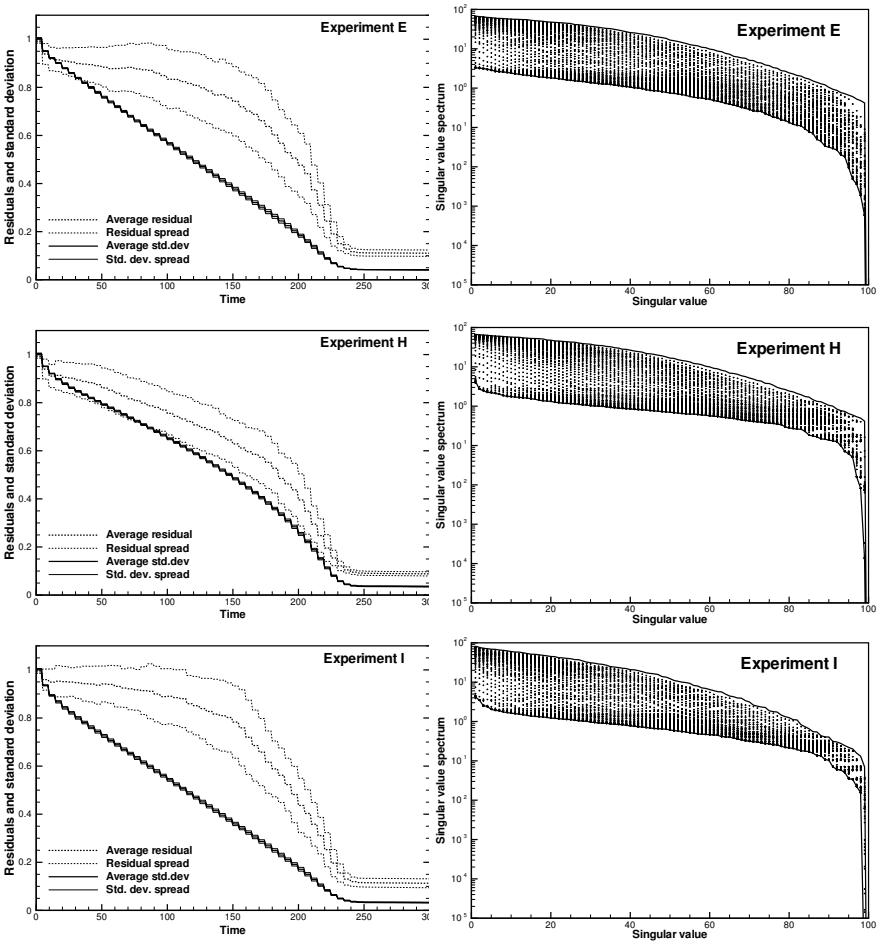

Fig. 11.5. See explanation in Fig. 11.4

icant positive impact resulting from the improved sampling of measurement perturbations.

11.7.5 Evolution of ensemble singular spectra

Finally, it is of interest to examine how the rank and conditioning of the ensemble evolves in time and is impacted by the computation of the analysis. In Figs. 11.4 and 11.5 we have plotted the singular values for the ensemble at each analysis time for some of the experiments. The initial singular spectrum of the ensemble is plotted as the upper thick line. Then the dotted lines indicate the reduction of the ensemble variance introduced at each analysis

update, until the end of the experiment where the singular spectrum is given by the lower thick line.

It is clear from *Exps. B* and *E* that the conditioning of the initial ensemble improves when the new sampling scheme is used. Furthermore, it is seen from *Exps. B, B150* and *B250* that increasing the ensemble size does not add much to the representation of variance in the error subspace. This result can be expected with the simple low-dimensional model state considered here.

11.7.6 Summary

The previous experiments have quantified the impact of using an improved sampling scheme in the EnKF. The improved sampling attempts to generate ensembles with full rank and a conditioning that is better than what can be obtained using random sampling. The improved sampling is used for the generation of the initial ensemble as well as for the sampling of measurement noise.

In the experiments discussed here it is possible to obtain a significant improvement in the results from the standard EnKF analysis scheme if improved sampling is used both for the initial ensemble and the measurement perturbations. It is expected that similar improvements can be obtained in general since the improved sampling provides a better representation of the ensemble error covariances and of the state space where the solution is searched for.

It is important to point out that these results may not be directly transferable to other more complex dynamical models. In the cases discussed here the dimension of the state vector (1001 grid cells) is small compared to typical applications with ocean and atmospheric models. Thus, although we expect that the use of improved sampling schemes in most cases can lead to an improvement in the results, it is not possible to quantify this improvement in general. Note that it is important to choose β large enough to capture the singnificant singular values, to provide realizations that are physically acceptable.

We have not examined fully the potential impact a nonlinear model will have on the ensemble evolution. The use of nonlinear models will change the basis from that of the initial ensemble, and may even reduce the rank of the ensemble. This suggests that the improved sampling should be used for the model noise as well, to help maintain the conditioning of the ensemble during the forward integration.

From these experiments we can give the recommendation that improved sampling should always be considered for both the initial ensemble and the sampling of measurement perturbations. The experiments have shown that there is a potential for either a significant reduction of the computing time or an improvement of the EnKF results, using the improved sampling schemes.

12

Model errors

We will now discuss the use of model errors in the ensemble and representer methods. A particular focus will be on the impact of time-correlated model errors. A simple scalar equation is used to illustrate the use of the ensemble and the representer methods for combined parameter and state estimation in this case.

12.1 Simulation of model errors

In the previous chapter we learned how to introduce a correlation between the random fields. We will now study in more detail how this can be used to simulate time correlated model errors, and how we can introduce the correct variance in each realization to properly represent the magnitude of the actual model error.

Again we assume that $w_k(\boldsymbol{x})$ is a sequence of white noise drawn from a distribution of smooth pseudo random fields with mean equal to zero and variance equal to one.

Equation (11.33) ensures that $q_k(\boldsymbol{x})$ is drawn from a distribution with variance equal to one as long as the variance of the distribution for $q_{k-1}(\boldsymbol{x})$ equals one. Thus, this equation will produce a sequence of time correlated pseudo random fields with mean equal to zero and variance equal to one. The covariance in time between $q_i(\boldsymbol{x})$ and $q_j(\boldsymbol{x})$ is given by the formula (11.34).

12.1.1 Determination of ρ

The factor ρ in (11.33) should be related to the time step used and a specified time de-correlation length τ. Equation (11.33), when excluding the stochastic term, resembles a difference approximation to

$$\frac{\partial q}{\partial t} = -\frac{1}{\tau} q, \tag{12.1}$$

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_12,

which states that q is damped with a ratio e^{-1}, over a time period $t = \tau$. A numerical approximation becomes

$$q_k = \left(1 - \frac{\Delta t}{\tau}\right) q_{k-1}, \tag{12.2}$$

where Δt is the time step. Thus, we define ρ as

$$\rho = 1 - \frac{\Delta t}{\tau}, \tag{12.3}$$

where $\tau \geq \Delta t$.

12.1.2 Physical model

A discrete stochastic model is now defined as

$$\psi_k(\boldsymbol{x}) = G(\psi_{k-1}(\boldsymbol{x})) + \sqrt{\Delta t}\sigma c q_k(\boldsymbol{x}), \tag{12.4}$$

where σ is the standard deviation of the model error and c is a factor to be determined. The choice of the stochastic term is explained next.

12.1.3 Variance growth due to the stochastic forcing

To explain the choice of the stochastic term in (12.4) we will use a simple random walk model for illustration, i.e.

$$\psi_k(\boldsymbol{x}) = \psi_{k-1}(\boldsymbol{x}) + \sqrt{\Delta t}\sigma c q_k(\boldsymbol{x}). \tag{12.5}$$

This equation can be rewritten as

$$\psi_k(\boldsymbol{x}) = \psi_0(\boldsymbol{x}) + \sqrt{\Delta t}\sigma c \sum_{i=0}^{k-1} q_{i+1}(\boldsymbol{x}). \tag{12.6}$$

The variance can be found by squaring (12.6) and taking the ensemble average, i.e.

$$\overline{\psi_s(\boldsymbol{x})\psi_s(\boldsymbol{x})} = \overline{\psi_0(\boldsymbol{x})\psi_0(\boldsymbol{x})} + \Delta t\sigma^2 c^2 \overline{\left(\sum_{k=0}^{s-1} q_{k+1}(\boldsymbol{x})\right)\left(\sum_{k=0}^{s-1} q_{k+1}(\boldsymbol{x})\right)} \tag{12.7}$$

$$= \overline{\psi_0(\boldsymbol{x})\psi_0(\boldsymbol{x})} + \Delta t\sigma^2 c^2 \sum_{j=0}^{s-1}\sum_{i=0}^{s-1} \overline{q_{i+1}(\boldsymbol{x})q_{j+1}(\boldsymbol{x})} \tag{12.8}$$

$$= \overline{\psi_0(\boldsymbol{x})\psi_0(\boldsymbol{x})} + \Delta t\sigma^2 c^2 \sum_{j=0}^{s-1}\sum_{i=0}^{s-1} \rho^{|i-j|} \tag{12.9}$$

$$= \overline{\psi_0(\boldsymbol{x})\psi_0(\boldsymbol{x})} + \Delta t\sigma^2 c^2 \left(-s + 2\sum_{i=0}^{s-1}(s-i)\rho^i\right) \tag{12.10}$$

$$= \overline{\psi_0(\boldsymbol{x})\psi_0(\boldsymbol{x})} + \Delta t\sigma^2 c^2 \frac{s - 2\rho - s\rho^2 + 2\rho^{s+1}}{(1-\rho)^2}, \tag{12.11}$$

where (11.34) has been used and s denote the number of time steps. The double sum in (12.9) is just summing elements in a matrix and is replaced by a single sum operating on diagonals of constant values. The summation in (12.10) has an explicit solution (*Gradshteyn and Ryzhik*, 1979, formula 0.113).

We now define the number n such that $n\Delta t = 1$, thus n is the number of time steps over one time unit. It is clear from (12.11), that if $c = 1$, then the increase in variance over s time steps is equal to

$$\frac{s\sigma^2}{n}\frac{1-\rho^2-2\rho/s+2\rho^{s+1}/s}{(1-\rho)^2}. \tag{12.12}$$

Thus, with $\rho = 0$ the increase in variance is just $s\sigma^2/n$ as would be expected. However, with coloured noise the increase in variance may become significantly higher, dependent on the value of ρ.

In cases where we know the exact statistics of the stochastic noise process, although these cases are rare, this additional variance increase is realistic. On the other hand, in many cases we may have an estimate of the expected variance increase σ^2 over a time unit, and we may anticipate that the noise process is coloured. In that case we will need to use the scaling factor c to obtain a noise process which results in a realistic variance increase per time unit.

The two equations (11.33) and (12.4) provide the standard framework for introducing stochastic model errors when using the EnKF. The formula (12.11) provides the mean for scaling the perturbations in (12.4) when changing ρ and/or the number of time steps per time unit to ensure that the ensemble variance growth over a time unit equals σ^2.

It is natural to assume that the increase in variance over s time steps should be equal to $s\sigma^2/n$, e.g. if $s = n$ this corresponds to integration over one time unit and the increase in variance becomes σ^2. We then have the formula

$$\frac{s\sigma^2}{n} = c^2\frac{s\sigma^2}{n}\frac{1-\rho^2-2\rho/s+2\rho^{s+1}/s}{(1-\rho)^2}, \tag{12.13}$$

which we can solve for c to get

$$c^2 = \frac{(1-\rho)^2}{1-\rho^2-2\rho/s+2\rho^{s+1}/s}. \tag{12.14}$$

If the sequence of model noise $q_k(\boldsymbol{x})$ is white in time, i.e. $\rho = 0$, we get $c \equiv 1$ as is expected. Thus, when (12.5) is iterated, $c = 1$ leads to the correct increase in ensemble variance given by σ^2 per time unit. The formula (12.14) is identical to the one proposed by *Evensen* (2003) but it was given for $s = n$ and integration over one time unit.

For red model noise, with $\rho \in (0, 1)$, the formula (12.14) still gives the correct answer, i.e. if the model is integrated s time steps, the variance at time step s has increased by $s\sigma^2/n$. However, a problem with this approach is that the variance increase is not linear, and if the integration is continued for

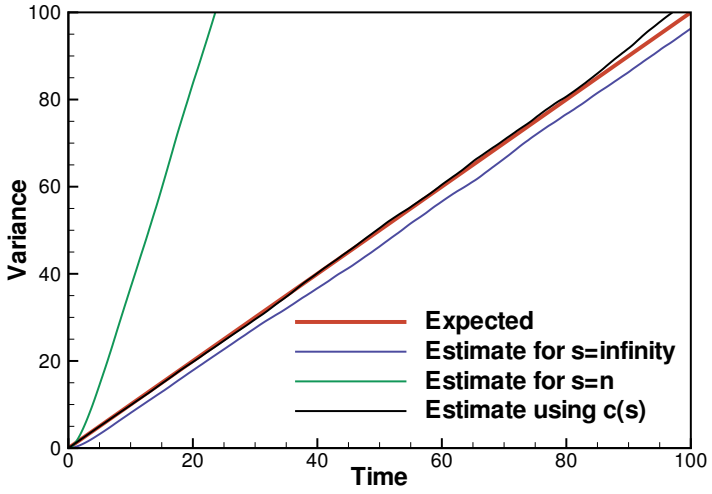

Fig. 12.1. The plot shows the variance using the different definitions for c. The expected variance is plotted as the red curve. The estimated variance from the Brownian motion (12.5) using the definitions (12.15) *(blue curve)*, and (12.14) with $s = n$ *(green curve)*. The formula (12.19) results in the black curve

more than s time steps the variance will increase too fast. This is seen from the green curve in Fig. 12.1 where c is evaluated for $s = n$, as in *Evensen* (2003), but the integration continues for a much longer time interval. The reason for the too large variance increase is that we have neglected correlations in time exceeding one time unit in the continued integration.

A better value for c, which can be used for long time integrations, is obtained by considering the limiting behaviour of the formula (12.11) when s becomes large. The solution for c when the number of time steps, s, goes to infinity in the formula (12.14) becomes

$$c^2 = \frac{1-\rho}{1+\rho}. \tag{12.15}$$

A plot of the estimated variance increase as a function of time for the Brownian motion process given by (12.5), is shown as the blue curve in Fig. 12.1. It is clear that the formula (12.15) gives a too weak variance increase initially but after an integration for a time period similar to the range of the exponential time correlation function used, the correct linear variance increase is obtained.

We can chose any value for s when evaluating the formula (12.14) for c. Thus, we can always obtain the correct variance at a certain time step, e.g. at a time when we are going to update the solution with measurements, but we would need to switch to the limiting value for c from (12.15) for the continued integration.

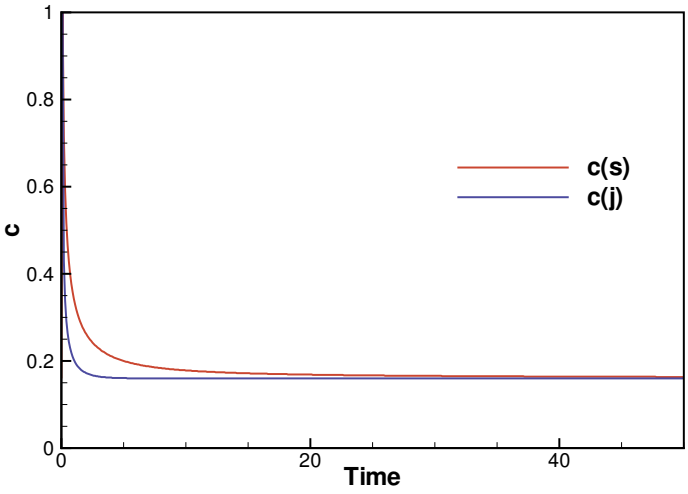

Fig. 12.2. The plot shows the value of c as a function of time, computed from (12.14) *(red curve)* and from (12.19) *(blue curve)*

It is possible to do better than this. We can use a formula with $c_i = c(i)$ being a function of the time step i, and require that the variance has the correct value at all time steps. We then need to introduce c_i inside the summation in (12.6) and we get

$$\psi_k(\boldsymbol{x}) = \psi_0(\boldsymbol{x}) + \sqrt{\Delta t}\sigma \sum_{i=1}^{k} c_i q_i(\boldsymbol{x}), \tag{12.16}$$

where we for simplicity also changed the summation index. As before we can write

$$\overline{\psi_s(\boldsymbol{x})\psi_s(\boldsymbol{x})} = \overline{\psi_0(\boldsymbol{x})\psi_0(\boldsymbol{x})} + \Delta t\sigma^2 \sum_{j=1}^{s}\sum_{i=1}^{s} c_i c_j \rho^{|i-j|}. \tag{12.17}$$

Now, assuming the increase in variance over s time steps to be equal to $s\sigma^2/n$ we get

$$\frac{s\sigma^2}{n} = \frac{\sigma^2}{n} \sum_{j=1}^{s}\sum_{i=1}^{s} c_i c_j \rho^{|i-j|}, \tag{12.18}$$

which can be rewritten as

$$s = \sum_{j=1}^{s-1}\sum_{i=1}^{s-1} c_i c_j \rho^{|i-j|} + \left(2\sum_{i=1}^{s-1} c_i \rho^{|s-i|}\right) c_s + c_s^2. \tag{12.19}$$

By using the definition for s (or rather $s-1$) in (12.18), we can write (12.19) as

$$s = s - 1 + \left(2 \sum_{i=1}^{s-1} c_i \rho^{|s-i|} \right) c_s + c_s^2, \tag{12.20}$$

where the double sum is eliminated.

Here (12.20) is a recursion of a second order scalar equations for c_s. Using that $c_1 = 1$ we can solve it recursively in each time step for c_s, $s \in (2, \infty)$, and we have resolved the issue with the too low initial variance increase. It is also clear that after a few time steps, exceeding the range of the time correlations specified, we approach the limiting value (12.15) for c. In Fig. 12.2 we have plotted c from (12.13) as a function of s as the red curve, and c from (12.19) as a function of time as the blue curve. Note that there is one sequence of positive and one of negative solutions for c_s which only differ in the sign and we can pick either.

12.1.4 Updating model noise using measurements

From the previous discussion is should be clear that when red model noise is used, correlations will develop between the red noise and the model variables. Thus, during the analysis step it is also possible to consistently update the model noise as well as the model state. This was illustrated in an example by *Reichle et al.* (2002). We now introduce a new state vector which consists of $\psi(\boldsymbol{x})$ augmented with $q(\boldsymbol{x})$. Equations (11.33) and (12.4) can then be written as

$$\begin{aligned} \begin{pmatrix} q_k(\boldsymbol{x}) \\ \psi_k(\boldsymbol{x}) \end{pmatrix} &= \begin{pmatrix} \rho q_{k-1}(\boldsymbol{x}) \\ G(\psi_{k-1}(\boldsymbol{x})) + \sqrt{\Delta t} \sigma c q_k(\boldsymbol{x}) \end{pmatrix} \\ &+ \begin{pmatrix} \sqrt{1-\rho^2} \boldsymbol{w}_{k-1} \\ 0 \end{pmatrix}. \end{aligned} \tag{12.21}$$

During the analysis we can now compute covariances between the observed model variable and the model noise vector $q(\boldsymbol{x})$, which is updated together with the state vector. This will lead to a correction of the mean of $q(\boldsymbol{x})$ as well as a reduction of the variance in the model noise ensemble. Note that this procedure estimates the actual error in the model for each ensemble member, given the prescribed model error statistics.

The form of (11.33) ensures that, over time, $q_k(\boldsymbol{x})$ will approach a distribution with mean equal to zero and variance equal to one, as long as we don't update $q_k(\boldsymbol{x})$ in the analysis scheme. In the case when $q_k(\boldsymbol{x})$ is updated it will be relaxed back towards a process with zero mean and variance equal to one.

12.2 Scalar model

We now define a simple scalar equation containing a poorly known parameter α, which has a first guess value $\alpha_0 = 0$, while the true value is $\alpha = 1$. We also

have a set of measurements of the true solution, which in this case becomes a constant $\psi(t) = 3$. Similarly to the system of equations (7.1–7.5) we now allow the model equation, the initial condition, the first guess parameter and the measurements to contain errors and write,

$$\frac{\partial \psi}{\partial t} = 1 - \alpha + q, \tag{12.22}$$

$$\psi(0) = 3 + a, \tag{12.23}$$

$$\alpha = 0 + \alpha', \tag{12.24}$$

$$\mathcal{M}[\psi] = \boldsymbol{d} + \boldsymbol{\epsilon}. \tag{12.25}$$

The model is defined on the interval $t \in [0, 50]$, thus using the notation from Chap. 7, $t_0 = 0$ and $t_k = 50$. We have used $G(\psi, \alpha) = 1 - \alpha$, so the model operator is linear and independent of ψ. There are nine measurements of ψ taken at the discrete times $t_{i(j)} = 5j$, for $j = 1, \ldots, 9$, and the measurement functional for measurement number j becomes just

$$\mathcal{M}_j[\psi] = \int_0^{50} \psi(t')\delta(t' - t_{i(j)})dt' = \psi(t_{i(j)}). \tag{12.26}$$

It should be noted that the simple form used for $G(\psi, \alpha)$, will result in a linear inverse problem even though α is included as a variable to be estimated. This will not be the case in general, since linear models containing, e.g. a product of ψ and α, will lead to nonlinear inverse problems when the parameter α, is considered as a variable to be estimated.

12.3 Variational inverse problem

The formulation of the variational inverse problem and the representer method is now derived for the simple linear combined parameter and state estimation problem, using the methodology explained in Chap. 8.

12.3.1 Prior statistics

We have to make assumptions about the statistical properties of the error terms added to (12.22–12.25). It is common to assume simple statistical forms for the priors, i.e. the error terms all have zero mean and the statistics is described by a covariance. Further, we assume that the different error terms are uncorrelated.

For the model errors q, we assume an exponential correlation in time

$$C_{qq}(t_1, t_2) = \sigma^2 \exp(-|t_2 - t_1|/\tau), \tag{12.27}$$

with σ^2 being the model error variance and τ the correlation length in time. The weight W_{qq} is defined from

$$W_{qq}(t_1, t_2) \bullet C_{qq}(t_2, t_3) = \delta(t_1 - t_3), \tag{12.28}$$

where the bullet denote integration in t_2.

The error in the initial condition a, is determined by the variance C_{aa} with inverse $W_{aa} = 1/C_{aa}$, and similarly the error in α is given by $C_{\alpha\alpha}$ with inverse $W_{\alpha\alpha} = 1/C_{\alpha\alpha}$. For the measurements the errors are described by the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$ with matrix inverse $\boldsymbol{W}_{\epsilon\epsilon}$.

12.3.2 Penalty function

The generalized inverse formulation (8.20) for the problem stated above becomes

$$\begin{aligned}\mathcal{J}[\psi, \alpha] = &\left(\frac{\partial \psi}{\partial t} - 1 + \alpha\right)_{t_1} \bullet W_{qq}(t_1, t_2) \bullet \left(\frac{\partial \psi}{\partial t} - 1 + \alpha\right)_{t_2} \\ &+ (\psi_0 - 3) W_{aa} (\psi_0 - 3) \\ &+ (\alpha - 0) W_{\alpha\alpha} (\alpha - 0) \\ &+ (\boldsymbol{d} - \mathcal{M}[\psi])^{\mathrm{T}} \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{d} - \mathcal{M}[\psi]).\end{aligned} \tag{12.29}$$

12.3.3 Euler–Lagrange equations

By setting the variational derivative of $\mathcal{J}[\psi, \alpha]$ with respect to α equal to zero, noting that ψ depends on α, we get the Euler–Lagrange equations

$$\frac{\partial \psi}{\partial t} = 1 - \alpha + C_{qq} \bullet \lambda, \tag{12.30}$$

$$\psi(0) = 3 + C_{aa} \lambda(0), \tag{12.31}$$

$$\frac{\partial \lambda}{\partial t} = -\mathcal{M}[\delta] \boldsymbol{W}_{\epsilon\epsilon} (\boldsymbol{d} - \mathcal{M}[\psi]), \tag{12.32}$$

$$\lambda(50) = 0, \tag{12.33}$$

$$\alpha = \alpha_0 - W_{\alpha\alpha} \int_0^{50} \lambda \, dt. \tag{12.34}$$

This is a coupled two point boundary problem in time, where the forward model (12.30) depends on the adjoint variable λ, and the adjoint backward model (12.32) depends on ψ. Note that the simple form of $G(\psi, \alpha)$ leads to an adjoint model (12.32) where the term $g_\psi \lambda$ in (8.42) vanishes.

If the true value of α is known, we eliminate the last equation (12.34) and are left with a linear inverse problem where the solution is defined by (12.30–12.33). This is still a coupled two point boundary value problem in time but a direct solution can be obtained using the representer method.

12.3.4 Iteration of parameter

We define an iteration of α to get a sequence of linear iterates for the Euler–Lagrange equations. Thus, we write

$$\alpha_l = \alpha_{l-1} - \gamma \left(\alpha_{l-1} - \alpha_0 + W_{\alpha\alpha} \int_0^{50} \lambda_{l-1} dt \right), \tag{12.35}$$

where the expression in the parentheses is the gradient of the penalty function with respect to the parameter α, and γ is a step length in a gradient descent method. Thus, for each iterate α_l, we need to solve

$$\frac{\partial \psi_l}{\partial t} = 1 - \alpha_l + C_{qq} \bullet \lambda_l, \tag{12.36}$$

$$\psi_l(0) = 3 + C_{aa} \lambda_l(0), \tag{12.37}$$

$$\frac{\partial \lambda_l}{\partial t} = -\mathcal{M}[\delta] \boldsymbol{W}_{\epsilon\epsilon} \left(\boldsymbol{d} - \mathcal{M}[\psi_l] \right), \tag{12.38}$$

$$\lambda_l(50) = 0. \tag{12.39}$$

12.3.5 Solution by representer expansions

Assume a solution of the form

$$\psi(t) = \psi_{\mathrm{F}}(t) + \boldsymbol{b}^{\mathrm{T}} \boldsymbol{r}(t), \tag{12.40}$$

$$\lambda(t) = \lambda_{\mathrm{F}}(t) + \boldsymbol{b}^{\mathrm{T}} \boldsymbol{s}(t), \tag{12.41}$$

i.e. the solution is a first guess solution $\psi_{\mathrm{F}}(t)$ and $\lambda_{\mathrm{F}}(t)$ plus a linear combination $\boldsymbol{b}$ of influence functions or representers $\boldsymbol{r}(t)$, and their adjoints $\boldsymbol{s}(t)$. There is one representer and associated adjoint for each measurement. We have now dropped the l-index for the iteration of the parameter α.

We insert (12.40) and (12.41) into (12.36–12.39). When assuming that $\boldsymbol{b}$ is undetermined and arbitrary we get a system of equations for the first guess solution,

$$\frac{\partial \psi_{\mathrm{F}}}{\partial t} = 1 - \alpha + C_{qq} \bullet \lambda_{\mathrm{F}}, \tag{12.42}$$

$$\psi_{\mathrm{F}}(0) = 3 + C_{aa} \lambda_{\mathrm{F}}(0), \tag{12.43}$$

$$\frac{\partial \lambda_{\mathrm{F}}}{\partial t} = 0, \tag{12.44}$$

$$\lambda_{\mathrm{F}}(50) = 0. \tag{12.45}$$

These equations have the solution $\lambda_{\mathrm{F}}(t) = 0$, and ψ_{F} is just the solution of the original dynamical model with no information from the measurements included.

By choosing the coefficients $\boldsymbol{b}$ to satisfy (5.60), we find the following set of equations for the representers and their adjoints

$$\frac{\partial \boldsymbol{r}}{\partial t} = C_{qq} \bullet \boldsymbol{s}, \tag{12.46}$$

$$\boldsymbol{r}(0) = C_{aa}\boldsymbol{s}(0), \tag{12.47}$$

$$\frac{\partial \boldsymbol{s}}{\partial t} = -\mathcal{M}[\delta], \tag{12.48}$$

$$\boldsymbol{s}(50) = 0. \tag{12.49}$$

These equations are now decoupled, i.e. (12.48) can be integrated backward in time from the final conditions (12.49) to find $\boldsymbol{s}$. Thereafter the system in (12.46) can be integrated forward in time from the initial conditions (12.47).

The symmetric positive definite representer matrix $\mathcal{M}^{\mathrm{T}}[\boldsymbol{r}]$, can be constructed by measuring the representers as soon as they have been solved for. Knowing ψ_{F}, $\boldsymbol{b}$ which is found from (5.60) and $\boldsymbol{r}$, we can construct the optimal minimizing solution of the linear iterate from (12.40), given a value for α.

12.3.6 Variance growth due to model errors

In the previous sections we found that the variance growth of a stochastic model increased when the noise process representing the model errors became coloured. This also has implications for the representer method. If we want to compare solutions using the representer method and the ensemble methods with coloured noise, we also need to introduce a correction factor in the representer method.

We start by noting that the representer corresponding to a particular direct measurement equals the space-time covariance function for the corresponding measurement location, and its value at the measurement location is equal to the prior variance at that location.

On discrete form we can write the model error covariance as the matrix

$$\boldsymbol{C}(t_i, t_j) = \sigma^2 c_{\mathrm{rep}} \exp(-|i-j|\Delta t/\tau), \tag{12.50}$$

for i and j taking values from 0 to the number of time steps and we have introduced the factor c_{rep} in the definition of the model error covariance.

Thus, we write the solution of (12.46) for each component, j of $\boldsymbol{r}$, at the corresponding measurement location $t_{i(j)}$, in discrete form as

$$r_j(t_{i(j)}) = r_j(t_{i(j)-1}) + \Delta t \sum_{i=0}^{i(j)} C(t_{i(j)}, t_i) s_j(t_i) \Delta t. \tag{12.51}$$

Note that the summation in the convolution for measurement j can be stopped at $i = i(j)$ since s_j is zero for $t_i > t_{i(j)}$. From this equation we can write

$$r_j(t_{i(j)}) = r_j(0) + \frac{\sigma^2 c_{\mathrm{rep}}}{n} \sum_{k=1}^{i(j)} \sum_{i=0}^{i(j)} \exp(-|k-j|\Delta t/\tau) s_j(t_i) \Delta t, \tag{12.52}$$

where we have used that $n = 1/\Delta t$ is the number of time steps over one time unit.

Thus, as in the previous chapter we can now determine c_{rep} so that for each representer it will have the correct variance at the measurement location $t_{i(j)}$,

$$\frac{i(j)\sigma^2}{n} = \frac{\sigma^2 c_{\text{rep}}}{n} \sum_{k=1}^{i(j)} \sum_{i=0}^{i(j)} \exp(-|k-j|\Delta t/\tau) s_j(t_i)\Delta t, \tag{12.53}$$

which we can solve for c_{rep} to get

$$c_{\text{rep}} = i(j) / \sum_{k=1}^{i(j)} \sum_{i=0}^{i(j)} \exp(-|k-j|\Delta t/\tau) s_j(t_i)\Delta t. \tag{12.54}$$

Note that we will get a slightly different value of c for measurements at different time locations, and probably a limiting value should be used, since a different value for c_{rep} at different time locations will lead to an unsymmetrical representer matrix.

12.4 Formulation as a stochastic model

For the ensemble methods we write the dynamical model (12.22) on stochastic form similarly to (12.21), i.e.

$$\begin{aligned} \begin{pmatrix} q_i \\ \psi_i \end{pmatrix} &= \begin{pmatrix} \rho q_{i-1} \\ \psi_{i-1} + (1-\alpha)\Delta t + \sqrt{\Delta t}\sigma c_i q_i \end{pmatrix} \\ &\quad + \begin{pmatrix} \sqrt{1-\rho^2}\, w_{i-1} \\ 0 \end{pmatrix}, \end{aligned} \tag{12.55}$$

where w_i is a white noise process with zero mean and unit variance, $\rho \in [0, 1)$ determines the time correlation and c_i is the factor from (12.19) which is used to tune the variance increase in time during the stochastic integration.

12.5 Examples

We will now discuss some examples where the system (12.22–12.25) is solved using the representer method, the EnKF and the EnKS. We will discuss the standard state estimation case where the parameter α is known, the state estimation case when an erroneous value is used for α and the model therefore is biased, and the case where we estimate both the model state and the parameter. We will consider both the case with white and coloured model noise. The examples are similar to, but not identical to the ones from *Evensen* (2003).

In all the cases we have used an initial and measurement variance equal to nine and the model error variance is equal to one. In the cases with time

correlated model noise the time scale of the correlation is $\tau = 2$. The number of ensemble members is 1000 and the time step is $\Delta t = 0.1$. In the cases with parameter estimation the parameter error variance is set to four.

The results from the experiments are presented in the Figs. 12.3–12.7. The measurements are plotted as bullets, the representer solution is plotted as the red line, the EnKF solution is given by the blue line and the EnKS solution is plotted as the green line. In addition we have included the EnKF and the EnKS solution plus and minus the estimated standard deviation as the blue and green dashed lines. Note that for the representer solution there is no error estimate, but if there where it would be identical to the EnKS error estimate in the limit of an infinite ensemble size.

12.5.1 Case A0

We first consider an example where the parameter $\alpha = 1$ is assumed to be known. This corresponds to a linear inverse calculation where we solve for ψ as a function of time given the observations. The results are presented in Fig. 12.3.

The representer solution is the maximum likelihood solution and can be used a reference. Note that, due to the use of white noise this curve will have discontinuous time derivatives at the measurement locations, a property of the representer and EnKS solutions when white model errors are used.

The EnKF estimate has discontinuities at the measurement locations due to the analysis updates. During the integration between the measurement locations the ensemble mean satisfies the dynamical part of the model equation, i.e. the time derivative of the solution is zero. The ensemble standard deviation is reduced at every measurement time, and increases according to the stochastic forcing term during the integration between the measurements.

The EnKS provides a continuous curve and is thus a more realistic estimate than the EnKF solution. It is clear that the EnKS solution is very similar to the representer solution, and it only differs due to the use of a finite ensemble size. Note that from the central limit theorem, we could run the EnKS experiments many times, and the resulting estimates would be normally distributed with a standard deviation given by $\sigma/\sqrt{N}$. A quick estimate is computed by setting $\sigma \approx 2$ and $N = 1000$, and we get a standard deviation of 0.06 which seems to be consistent with the difference between the EnKS and representer solution in this case and the cases to follow.

From the ensemble standard deviation for the EnKS, there is clearly an impact backward in time from the measurements and the overall error estimate is smaller than for the EnKF. The minimum errors are found at the measurement locations as expected. After the final measurement update the EnKF and EnKS solutions are identical, thus, for forecasting purposes it suffices to compute the EnKF solution.

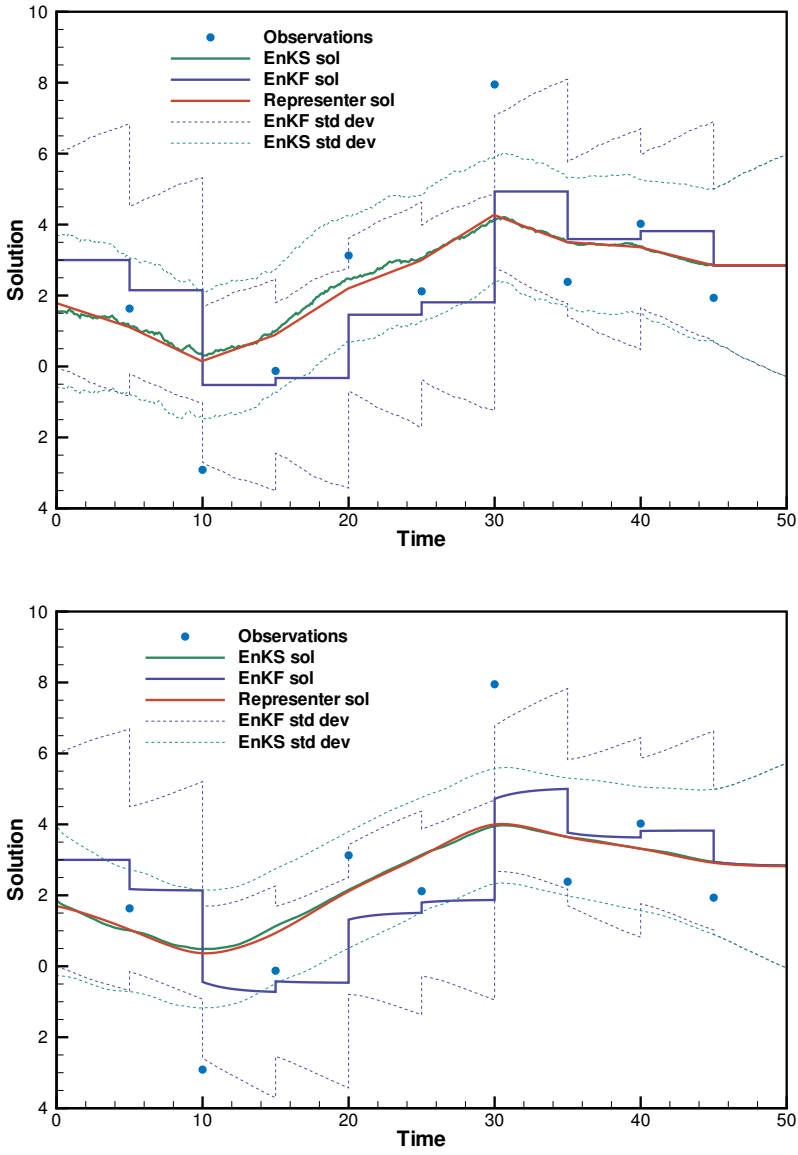

Fig. 12.3. Cases A0 and A1: Pure state estimation with unbiased model. The upper plot shows the results from Case A0 with white model noise while the lower plots shows the results from Case A1 where coloured model noise is used

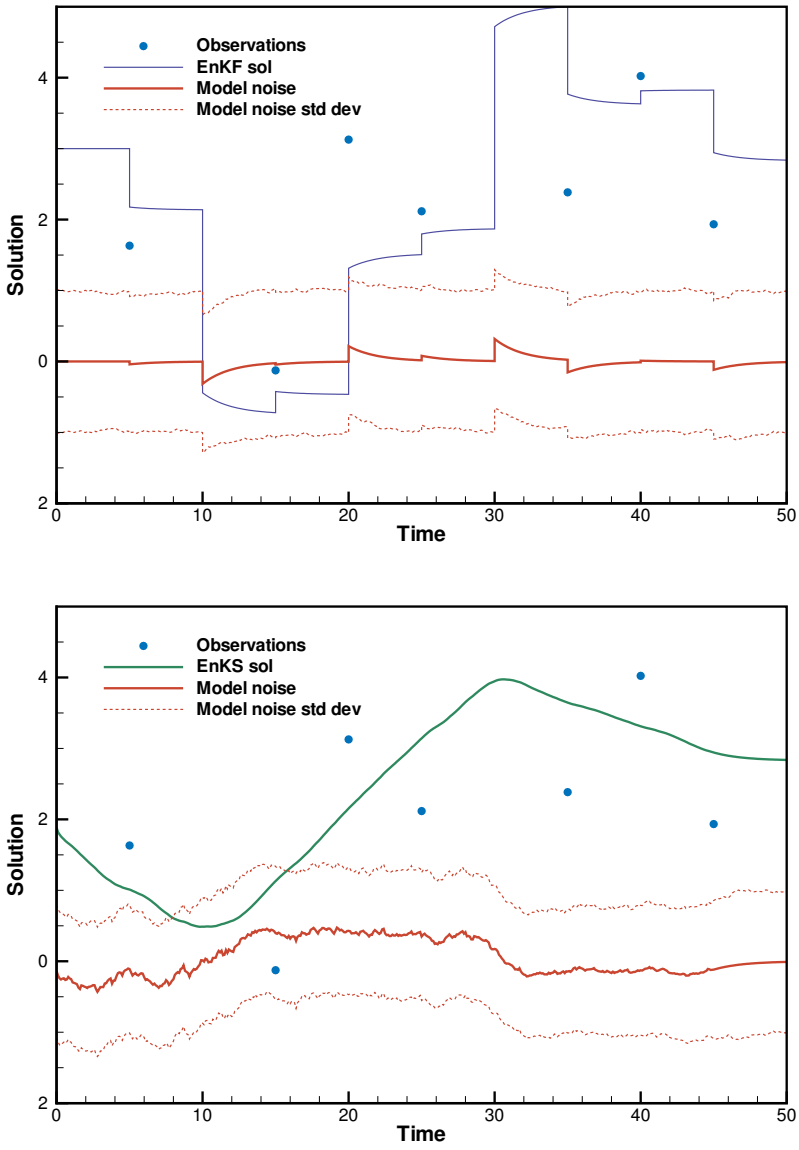

Fig. 12.4. Case A1: The system noise estimated by the EnKF (upper plot) and the EnKS (lower plot)

12.5.2 Case A1

This experiment is similar to Case A0 but we have now introduced time correlated model noise. The first impression from the lower plot in Fig. 12.3 is that all curves are smoother and less noisy in this case. The EnKS and representer solutions are now also smooth at the measurement locations as is expected when time correlated model noise is used.

An important difference between this and the previous case is that now the EnKF solution sometimes shows a positive or negative trend during the integration between the measurements. This is caused by the assimilation updates of the model noise which introduces a "bias" in the stochastic forcing. This is seen in upper plot of Fig. 12.4 which plots the EnKF solution as the blue line, the EnKF estimate for the model noise as the red line, and the standard deviation of the model noise as the red dashed lines. It is clearly seen that the model noise is being updated at the assimilation steps, e.g. note the large updates at the second and sixth measurements. These updates introduce a bias in the system noise which helps relaxing the solution in the direction of the measurements. Thus, as we will see below, the model noise can help counteract a bias in model. Note that during the integration between the measurements the bias slowly relaxes back toward zero in agreement with the equation used for the simulation of the model noise.

The estimated EnKS system noise is presented as the red solid line in the lower plot of Fig. 12.4 and also here the time derivatives are continuous at the measurement locations. In fact, this estimated model noise is the forcing needed to reproduce the EnKS solution when a single model is integrated forward in time starting from the initial condition estimated by the EnKS; i.e. the solution of

$$\begin{aligned} \psi_k &= \psi_{k-1} + \sqrt{\Delta t}\sigma c\hat{q}_k, \\ \psi_0 &= \hat{\psi}_0, \end{aligned} \tag{12.56}$$

with $\hat{q}_k$ and $\hat{\psi}_0$ being the EnKS estimated model noise and initial condition respectively, will exactly reproduce the EnKS estimate. Obviously, the estimated model noise is the same as is computed and used in the forward Euler Lagrange equation in the representer method. This points to the similarity between the EnKS and the representer method, which for linear models will give identical results when the ensemble size becomes infinite.

12.5.3 Case B

We now consider a case where we have an erroneous value, $\alpha = 0$, and the model thus contains a bias, always predicting a line with slope equal to one, while the true solution should have zero slope. In this case we are not attempting to estimate the parameter, but rather trying to solve the inverse

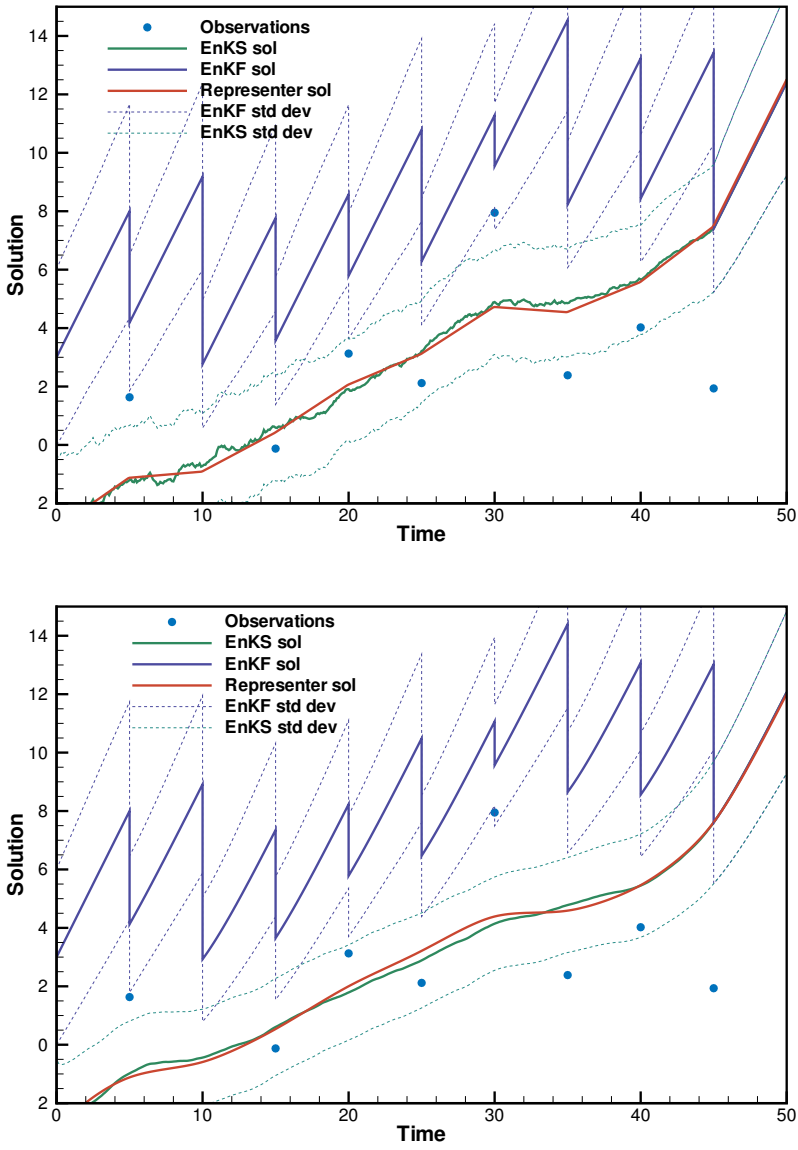

Fig. 12.5. Cases B0 and B1: Pure state estimation with biased model. The upper plot shows the results from Case B0 with white model noise while the lower plot shows the results from Case B1 where coloured model noise is used

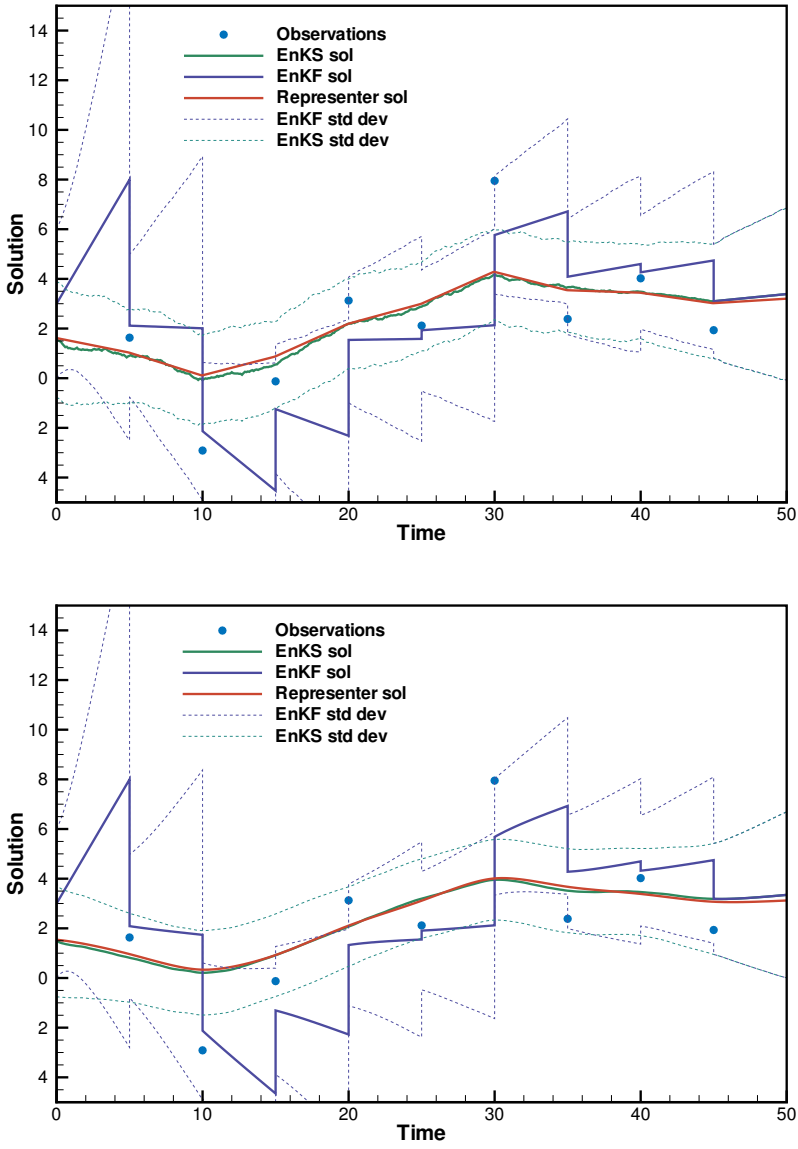

Fig. 12.6. Cases C0 and C1: Combined parameter and state estimation with biased model. The upper plot shows the results from Case C0 with white model noise while the lower plot shows the results from Case C1 where coloured model noise is used

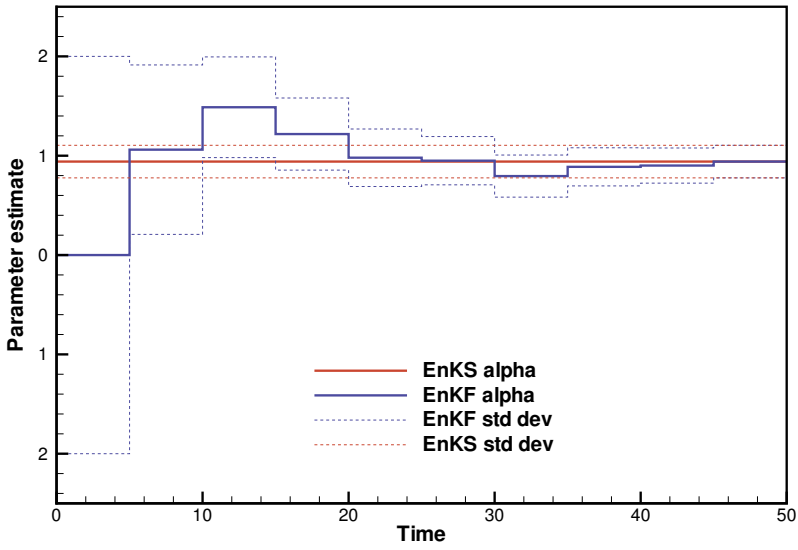

Fig. 12.7. Case C1. Convergence of parameter value over time using the EnKF and EnKS

problem in the case when the model contains a bias. The results are presented in Fig. 12.5 for the cases with $\tau = 0$ and $\tau = 2$. Again it is seen that the EnKS and representer solutions are nearly identical and they provide a good estimate of the true solution over most of the time interval. There is an exception for the estimate near the beginning and end of the time interval where the bias in the model cannot be corrected for. It is clear that the EnKS provides a significantly better result than the EnKF for this particular case. This is partly related to the measurement frequency and the fact that the information from only past and present measurements is insufficient to properly constrain the evolution of the filter.

The reason that the EnKS and the representer methods provide good results for most of the time interval is that they are both finding good estimates of the model error, i.e. q_i from (12.55) for the EnKS, and $\lambda(t)$ for the representer method, which corrects for the bias. However, this correction is not maintained after the final measurement due to the limited time correlation specified.

12.5.4 Case C

In this case we also start out with an erroneous value, $\alpha = 0$, but assume that the parameter contains an error of variance equal to 4. The inverse solution is given in Fig. 12.6. It is clear that both the representer method and the EnKS

provide realistic and very similar results. Further, the bias observed from the previous case is entirely eliminated since we have now also computed an estimate of α which in the representer method converged to 0.96 and in the EnKF and EnKS we obtained the value 0.94. Thus, as expected we obtained values in between the first guess of $\alpha = 0$ and the true value of $\alpha = 1$. We cannot expect to converge exactly to the true value since we have included a prior error statistics for the parameter. This prior is needed to ensure the existence of a solution independent of the number of measurements assimilated. We also observe that the EnKF solution initially shows a strong bias, but this is quickly reduced after a few updates with measurements.

In Fig. 12.7 we have plotted the estimated value for α as a function of time for the EnKF and EnKS. We have also included the one standard deviation of the errors in the plot. We started out with a value of α equal to zero and set the prior variance for the parameter equal to four. It is seen that the EnKF provides an update of the parameter at each measurement time, and at the same time the estimated error variance is reduced. In this example the parameter estimate converges quickly, and the standard deviation of the error is reduced at each update with measurements. The final estimate for the error standard deviation of the parameter is 0.16 corresponding to an error variance of 0.026, so a significant improvement is obtained in the parameter estimate. Note also that the EnKS propagates information backward in time and thus provides a time independent estimate of α which is identical to the final estimate from the EnKF.

Using the representer method, the iterations on α in (12.35) converged quickly in around 10 iterations when an iteration step, $\gamma = 0.01$, was used, and we did not attempt to optimize or tune this value further.

12.5.5 Discussion

The conclusion from these experiments is that the EnKF, EnKS and representer method all provide the same solution of the inverse state and parameter estimation problem as long as the model is linear and the assumption of Gaussian priors is valid. It should be emphasized that this example has used a very simple linear model and we do expect that the associated inverse formulation is fairly well posed and easy to solve. Hence, for the representer method, the penalty function for each linear iterate is quadratic without local minima, and a unique solution is always obtained.

For the EnKF, we do not have to consider effects of non-Gaussian error statistics since the model is linear. Thus, we have considered a very simple problem where we would expect both the representer method and the EnKF/EnKS methods to work well.

It is also interesting to see that the case with no measurements are accounted for using both the representer and the ensemble methods. In the representer method the solution then becomes the first guess solution ψ_F.

This corresponds to the mode, or modal trajectory, of the joint pdf defined by, e.g. (7.10), and the value of α becomes the prior value α_0.

The Ensemble methods provide a pure ensemble integration when no measurements are available. Clearly, we can store the ensemble at all times and compute the modal trajectory as well. However, we believe that the mode of the marginal pdf would be a better estimator. An argument for this is that a single model realization from a nonlinear model does not make any statistical sense. It is just one out of infinitively many possible realizations.

In the ensemble methods the mean is used as the best estimator. This is mostly a practical choice since the estimation of the mode will require the use of a much larger ensemble. Thus, the estimate from the EnKF and EnKS when no measurements are assimilated is just the evolution of ensemble mean. This corresponds to the mean of the marginal pdf which also happens to be equal to the mean of the joint pdf. Thus, in the ensemble methods the ensemble mean is the best estimate and it comes with an associated error covariance estimate.

13

Square Root Analysis schemes

The perturbation of measurements used in the EnKF standard analysis is an additional source of sampling error. The works by *Anderson* (2001), *Whitaker and Hamill* (2002), *Bishop et al.* (2001), the review by *Tippett et al.* (2003), and the paper by *Evensen* (2004), have developed "square root" implementations of the analysis scheme where the perturbation of measurements is avoided. The square root methods are intuitively very appealing but there are also some pitfalls as pointed out by *Lawson and Hansen* (2004) and *Leeuwenburgh et al.* (2005). See also the papers by *Sakov and Oke* (2008) and *Livings et al.* (2008) for a revised interpretation and mathematical analysis of the square root schemes. The version of the square root scheme from *Evensen* (2004), modified according to the findings of *Sakov and Oke* (2008) and *Livings et al.* (2008), is presented below. A simple linear advection model is used to demonstrate the impact of the different analysis schemes as well as the impact of using the improved sampling technique from the previous chapter when generating the initial ensemble and measurement perturbations.

13.1 Square root algorithm for the EnKF analysis

The square root schemes presented by *Anderson* (2001), *Whitaker and Hamill* (2002), and *Bishop et al.* (2001), all introduced some kind of approximation to make them efficient, e.g. the use of a diagonal measurement error covariance matrix or knowledge of the inverse of the measurement error covariance matrix. Here the simpler and more direct variant of the square root analysis scheme, by *Evensen* (2004), is derived, which solves for the analysis without imposing any additional approximations.

The square root algorithm is used to update the ensemble perturbations and is derived starting from the traditional analysis equation for the covariance update in the Kalman Filter (9.6). The time index is in the remainder of this chapter dropped for convenience. When using the ensemble representation for the error covariance matrix, $\boldsymbol{C}_{\psi\psi}$, as defined in (9.14), (9.6) can be written

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_13,

$$\boldsymbol{A}^{\mathrm{a}\prime}\boldsymbol{A}^{\mathrm{a}\prime\mathrm{T}} = \boldsymbol{A}'\left(\boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{S}\right)\boldsymbol{A}'^{\mathrm{T}}, \tag{13.1}$$

where we have used the definitions of $\boldsymbol{S}$ and $\boldsymbol{C}$ from (9.33) and (9.34), i.e. $\boldsymbol{S} = \mathcal{M}[\boldsymbol{A}']$ is the measurement of the ensemble perturbations and $\boldsymbol{C} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}$, with $\boldsymbol{C}_{\epsilon\epsilon}$ being the measurement error covariance matrix. We have for simplicity dropped the 'f' superscript on $\boldsymbol{A}^{\mathrm{f}}$ and $\boldsymbol{A}^{\mathrm{f}\prime}$.

13.1.1 Updating the ensemble mean

In the square root scheme, the analyzed ensemble mean is computed from the standard Kalman filter analysis equation, which can be obtained by multiplying the first line in (9.39) from the right with $\mathbf{1}_N$, so that each column in the resulting equation for the mean becomes

$$\overline{\boldsymbol{\psi}}^{\mathrm{a}} = \overline{\boldsymbol{\psi}}^{\mathrm{f}} + \boldsymbol{A}'\boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\left(\boldsymbol{d} - \boldsymbol{M}\overline{\boldsymbol{\psi}}^{\mathrm{f}}\right). \tag{13.2}$$

13.1.2 Updating the ensemble perturbations

The following derives an equation for the ensemble analysis by defining a factorization of (13.1) where there are no references to the measurements or measurement perturbations.

We start by forming $\boldsymbol{C}$ as defined in (9.34). For now we assume that $\boldsymbol{C}^{-1}$ exists, which requires that the rank of the ensemble be greater than the number of measurements. The low-rank case involves pseudo inversion and is discussed in Chapter 14. Note also that the use of a full rank $\boldsymbol{C}_{\epsilon\epsilon}$ can result in a full rank $\boldsymbol{C}$ even when $m \geq N$.

By computing the eigenvalue decomposition $\boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^{\mathrm{T}} = \boldsymbol{C}$, we obtain the inverse of $\boldsymbol{C}$ as

$$\boldsymbol{C}^{-1} = \boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^{\mathrm{T}}, \tag{13.3}$$

where $\boldsymbol{Z} \in \Re^{m\times m}$ is an orthogonal matrix and $\boldsymbol{\Lambda} \in \Re^{m\times m}$ is diagonal. The eigenvalue decomposition may be the most demanding computation required for the analysis when m is large. An efficient alternative inversion algorithm is presented in Chapter 14.

We now write (13.1) as

$$\begin{aligned}
\boldsymbol{A}^{\mathrm{a}\prime}\boldsymbol{A}^{\mathrm{a}\prime\mathrm{T}} &= \boldsymbol{A}'\left(\boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}}\boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{S}\right)\boldsymbol{A}'^{\mathrm{T}} \\
&= \boldsymbol{A}'\left(\boldsymbol{I} - (\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{S})^{\mathrm{T}}(\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{S})\right)\boldsymbol{A}'^{\mathrm{T}} \\
&= \boldsymbol{A}'\left(\boldsymbol{I} - \boldsymbol{X}_2^{\mathrm{T}}\boldsymbol{X}_2\right)\boldsymbol{A}'^{\mathrm{T}},
\end{aligned} \tag{13.4}$$

where $\boldsymbol{X}_2 \in \Re^{m\times N}$ is defined as

$$\boldsymbol{X}_2 = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{S}, \tag{13.5}$$

and where $\text{rank}(\boldsymbol{X}_2) = \min(m, N-1)$. Thus, $\boldsymbol{X}_2$ is a projection of $\boldsymbol{S}$ onto the eigenvectors of $\boldsymbol{C}$ scaled by the square root of the eigenvalues of $\boldsymbol{C}$.

Next we compute the singular value decomposition of $\boldsymbol{X}_2$ given by

$$\boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^{\mathrm{T}} = \boldsymbol{X}_2, \tag{13.6}$$

with $\boldsymbol{U}_2 \in \Re^{m \times m}$, $\boldsymbol{\Sigma}_2 \in \Re^{m \times N}$ and $\boldsymbol{V}_2 \in \Re^{N \times N}$. Since $\boldsymbol{U}_2$ and $\boldsymbol{V}_2$ are orthogonal matrices, (13.4) can be written

$$\begin{aligned}
\boldsymbol{A}^{\mathrm{a}\prime} \boldsymbol{A}^{\mathrm{a}\prime\mathrm{T}} &= \boldsymbol{A}' \Big(\boldsymbol{I} - \big(\boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^{\mathrm{T}} \big)^{\mathrm{T}} \big(\boldsymbol{U}_2 \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^{\mathrm{T}} \big) \Big) \boldsymbol{A}'^{\mathrm{T}} \\
&= \boldsymbol{A}' \Big(\boldsymbol{I} - \boldsymbol{V}_2 \boldsymbol{\Sigma}_2^{\mathrm{T}} \boldsymbol{\Sigma}_2 \boldsymbol{V}_2^{\mathrm{T}} \Big) \boldsymbol{A}'^{\mathrm{T}} \\
&= \boldsymbol{A}' \boldsymbol{V}_2 \Big(\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}} \boldsymbol{\Sigma}_2 \Big) \boldsymbol{V}_2^{\mathrm{T}} \boldsymbol{A}'^{\mathrm{T}} \\
&= \left(\boldsymbol{A}' \boldsymbol{V}_2 \sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}} \boldsymbol{\Sigma}_2} \right) \left(\boldsymbol{A}' \boldsymbol{V}_2 \sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}} \boldsymbol{\Sigma}_2} \right)^{\mathrm{T}}.
\end{aligned} \tag{13.7}$$

Thus, a solution for the analysis ensemble perturbations is

$$\boldsymbol{A}^{\mathrm{a}\prime} = \boldsymbol{A}' \boldsymbol{V}_2 \sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}} \boldsymbol{\Sigma}_2}. \tag{13.8}$$

As noted in *Wang et al.* (2004) the update equation (13.8) does not conserve the mean of the ensemble perturbations, and in fact leads to the production of outliers that contain most of the ensemble variance as explained in *Leeuwenburgh et al.* (2005), and which is further illustrated in the example below.

We now write the square root update in the more general form

$$\boldsymbol{A}^{\mathrm{a}\prime} = \boldsymbol{A}' \boldsymbol{T}, \tag{13.9}$$

where $\boldsymbol{T}$ is a square root transformation matrix.

It is shown in *Sakov and Oke* (2008) and *Livings et al.* (2008) that in order for the square root analysis scheme to be unbiased and preserve the zero mean in the updated perturbations, the vector $(1/N)\mathbf{1}$, where $\mathbf{1} \in \Re^N$ has all components equal to 1, must be an eigenvector of the square root transformation matrix $\boldsymbol{T}$. As noted in *Sakov and Oke* (2008) and *Livings et al.* (2008), this condition is not satisfied for the update in (13.8).

Multiplying (13.9) from the right with the vector $\mathbf{1}$ and assuming that $(1/N)\mathbf{1}$ is an eigenvector of $\boldsymbol{T}$, we can write

$$\mathbf{0} = \boldsymbol{A}^{\mathrm{a}\prime} \mathbf{1} = \boldsymbol{A}' \boldsymbol{T} \mathbf{1} = \lambda \boldsymbol{A}' \mathbf{1} = \mathbf{0}. \tag{13.10}$$

Equation (13.10) shows that a sufficient condition for the mean to be unbiased is that $(1/N)\mathbf{1}$ be an eigenvector of $\boldsymbol{T}$. If the transform matrix is of full rank, then this condition is also necessary (*Livings et al.*, 2008).

The symmetric square root solution for the analysis ensemble perturbations is defined as

$$\boldsymbol{A}^{\mathrm{a}\prime} = \boldsymbol{A}'\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}. \tag{13.11}$$

It is easy to show that (13.11) is also a factorization of (13.7) since $\boldsymbol{V}_2$ is an orthogonal matrix. As shown in *Sakov and Oke* (2008) and *Livings et al.* (2008), the symmetric square root has an eigenvector equal to $(1/N)\mathbf{1}$ and is unbiased. In addition, the symmetric square root resolves the issue with outliers in the factorization used in (13.8). The analysis update of the perturbations becomes a symmetric contraction of the forecast ensemble perturbations. Thus, if the predicted ensemble members have a non-Gaussian distribution, then the updated distribution retains the shape but the variance is reduced.

A randomization of the analysis update can be used to generate updated perturbations that better resemble a Gaussian distribution (*Evensen*, 2004). Thus, we write the symmetric square root solution (13.11) as

$$\boldsymbol{A}^{\mathrm{a}\prime} = \boldsymbol{A}'\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}, \tag{13.12}$$

where $\boldsymbol{\Theta}$ is a mean-preserving random orthogonal matrix, which can be computed using the algorithm from *Sakov and Oke* (2008).

13.1.3 Properties of the square root scheme

The properties of the square root schemes are illustrated in Figure 13.1, which shows the resulting ensemble updates using several variants of the EnKF analysis scheme. The Lorenz equations (6.5)–(6.7) are used since the strong nonlinearities lead to the development of a non-Gaussian distribution for the forecast ensemble. Three observations are used in the update step. Each ensemble member is plotted as a circle in the x, y plane. In both Figure 13.1a and Figure 13.1b the forecast ensemble members are plotted as the blue circles, which have a non-Gaussian distribution in the x, y plane.

In Figure 13.1a the updated analysis from the "one-sided" square root scheme in (13.8) is shown as the yellow circles. It can be seen that $N - 3$ of the updated ensemble perturbations collapse onto $(0, 0)$, while the three nonzero "outliers", one for each measurement, determine the ensemble variance. However, one of the outliers is too close to zero to be distinguished from the other points at zero. The variance of the updated ensemble is correct, but the analysis introduces a bias through a shift in the ensemble mean. The shift in the mean should come as no surprise since we do not impose a condition for the conservation of the mean when the update equation is derived. It is in fact shown in Section 13.1.6 that for a three variable model, and with three measurements and a diagonal measurement error covariance matrix, we obtain an ensemble with three outliers while the remainder of the perturbations collapse onto zero.

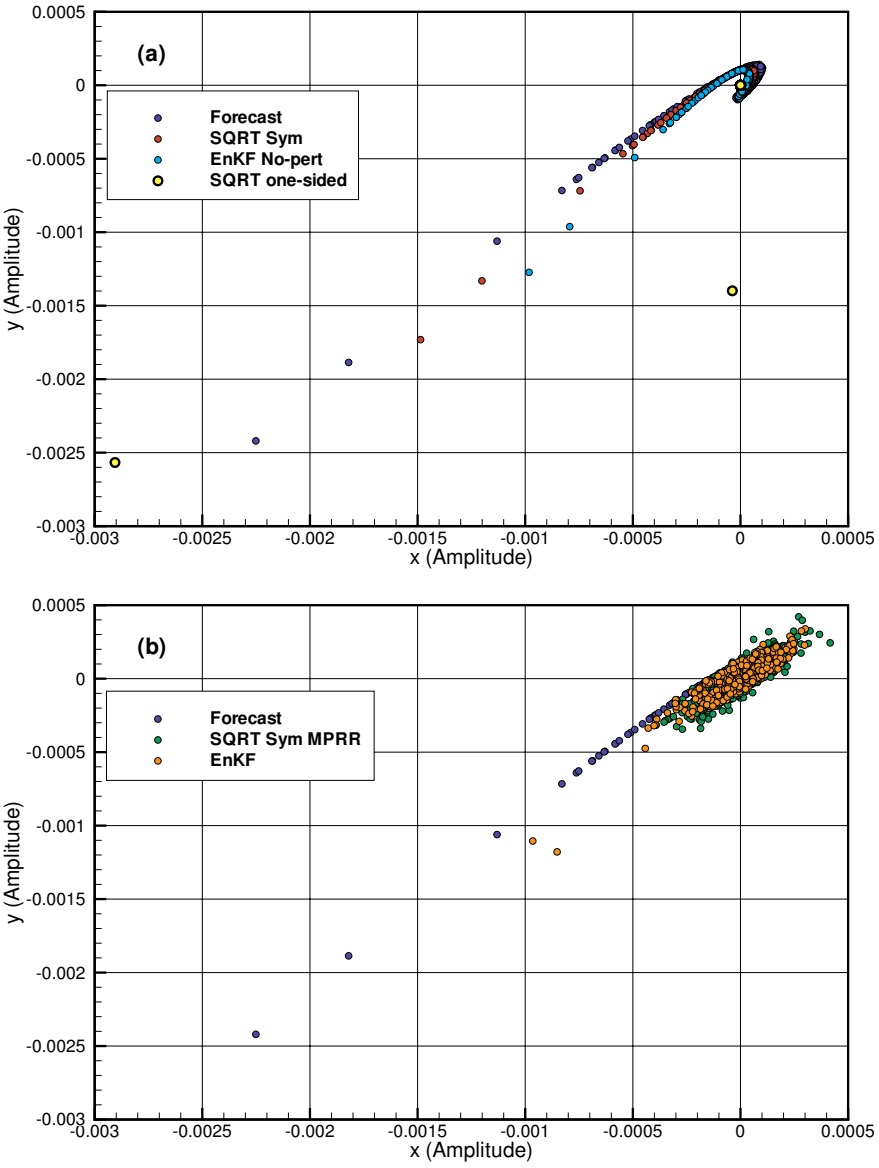

Fig. 13.1. Forecast and analysis ensembles for the Lorenz equations illustrating properties of the analysis schemes discussed in the text. The data used in these plots were contributed by Dr. Pavel Sakov.

In Figure 13.1a the updated analysis from the symmetric square root scheme in (13.11) is shown as the red circles. This scheme has the property that it rescales the ensemble of perturbations without changing the original shape of the perturbations. Thus, the scheme allows for preserving possible non-Gaussian structures in the ensemble during the update. We also note that the symmetric square root scheme from (13.11) is unbiased and thus preserves the mean (*Sakov and Oke*, 2008).

In Figure 13.1b the updated analysis from the symmetric square root scheme from (13.12) that includes an additional mean-preserving random rotation, is plotted using the green circles. It is clear that the ensemble of updated perturbations now has a Gaussian shape, and the non-Gaussian shape of the forecast ensemble perturbations is lost. The random rotation completely destroys any prior structure in the ensemble by randomly redistributing the variability among all of the ensemble members. Thus, the random rotation acts as a complete resampling from a Gaussian distribution, but represented by the ensemble space, while preserving the ensemble mean and variance.

Figure 13.1b also shows the updated analysis from the standard EnKF scheme from (9.39), where the measurements are randomly perturbed to represent their uncertainty. The standard EnKF analysis becomes similar to the symmetric square root analysis with random rotation. As with the symmetric square root analysis, most of the non-Gaussian shape of the forecast ensemble is lost. However, only the increment in the standard EnKF analysis is Gaussian, and some of the non-Gaussian properties of the forecast ensemble is retained, as indicated by the two outliers that represent the tail of the distribution seen in the forecast ensemble.

It is also interesting to consider the standard EnKF scheme when used without perturbation of measurements. It is then clear from (4.41) that the variance is reduced twice by the additional multiplication with $\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}$ resulting from $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ in (4.41) being identical to zero when the measurements are not treated as stochastic variables. Figure 13.1a shows that the EnKF scheme without perturbation of measurements preserves the shape of the forecast distribution in the same way as the symmetric square root scheme, although the variance is too low. Thus, the perturbation of measurements in EnKF both increases the ensemble variance to the "correct" value, and introduces additional randomization. The randomization is different from the one observed in (13.12) since only the increments are randomized in the EnKF scheme with perturbation of measurements.

It is currently not clear which of the analysis schemes, that is, the standard EnKF (9.39), the symmetric square root (13.11), or the symmetric square root with random rotation (13.12), is best in practice. Probably, the choice of analysis scheme depends on the dynamical model, and possibly also on the measurement density and ensemble size used. For a linear dynamical model, the forecast distribution is Gaussian, and the random rotation is not needed. Thus, we then expect the symmetric square root (13.11) to be the best choice. On the other hand, for a strongly nonlinear dynamical model where non-

Gaussian effects are dominant in the predicted ensemble, the symmetric square root with a random rotation (13.12) or EnKF with perturbed measurements (9.39) may work better. Both of these schemes introduce Gaussianity into the analysis update, and a Gaussian forecast ensemble may lead to more consistent analysis updates.

The random rotation might be considered as a re-sampling from a Gaussian distribution at each analysis update. Note again that the random rotation in the square root filter, contrary to the measurement perturbation used in EnKF, completely eliminates all non-Gaussian contributions that may be contained in the forecast ensemble.

13.1.4 Final update equation

In Chap. 9 it was shown that the EnKF analysis update can be written as

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}\boldsymbol{X}, \tag{13.13}$$

where $\boldsymbol{X}$ is an $N \times N$ matrix of coefficients. The square root schemes can also be written in the same simple form. We start by writing the analysis as the updated ensemble mean plus the updated ensemble perturbations,

$$\boldsymbol{A}^{\mathrm{a}} = \overline{\boldsymbol{A}}^{\mathrm{a}} + \boldsymbol{A}^{\mathrm{a}\prime}. \tag{13.14}$$

The updated mean can, using (13.2), be written as

$$\begin{aligned}\overline{\boldsymbol{A}^{\mathrm{a}}} &= \overline{\boldsymbol{A}} + \boldsymbol{A}'\boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\left(\overline{\boldsymbol{D}} - \mathcal{M}\left[\overline{\boldsymbol{A}}\right]\right) \\ &= \boldsymbol{A}\boldsymbol{1}_N + \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{1}_N)\boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\left(\boldsymbol{D} - \mathcal{M}\left[\boldsymbol{A}\right]\right)\boldsymbol{1}_N \\ &= \boldsymbol{A}\boldsymbol{1}_N + \boldsymbol{A}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\left(\boldsymbol{D} - \mathcal{M}\left[\boldsymbol{A}\right]\right)\boldsymbol{1}_N,\end{aligned} \tag{13.15}$$

and from (13.12) the updated perturbations become

$$\begin{aligned}\boldsymbol{A}^{\mathrm{a}\prime} &= \boldsymbol{A}'\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}} \\ &= \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{1}_N)\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}.\end{aligned} \tag{13.16}$$

Combining the previous equations we get (13.13) with $\boldsymbol{X}$ defined as

$$\boldsymbol{X} = \boldsymbol{1}_N + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\left(\boldsymbol{D} - \mathcal{M}[\boldsymbol{A}]\right)\boldsymbol{1}_N + (\boldsymbol{I} - \boldsymbol{1}_N)\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}. \tag{13.17}$$

Thus, we still search for the solution as a combination of ensemble members, and it also turns out that forming $\boldsymbol{X}$ and then computing the matrix multiplication in (13.13) is the most efficient algorithm for computing the analysis when many measurements are used. Note that we already have $\boldsymbol{C}^{-1}$ from (13.3). The mean-preserving random rotation $\boldsymbol{\Theta}$ is included in the equation but can be removed by setting $\boldsymbol{\Theta} = \boldsymbol{I}$ and the scheme then reverts to the symmetrical square root scheme.

13.1.5 Analysis update using a single measurement

We will now look at the special case where a single measurement ($m = 1$) is used. The matrix inversion in (13.3) then becomes a scalar inverse and using the notation from the eigenvalue decomposition we have $\boldsymbol{Z} = 1$ and $\boldsymbol{\Lambda}$ is the scalar $\lambda = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}$. Thus, from (13.5) we get $\boldsymbol{X}_2 = \lambda^{-\frac{1}{2}}\boldsymbol{S}$.

The singular value decomposition (13.6) of $\boldsymbol{X}_2$ then equals $\lambda^{-\frac{1}{2}}$ times the singular value decomposition of $\boldsymbol{S}$,

$$\lambda^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \lambda^{-\frac{1}{2}}\boldsymbol{S} = \boldsymbol{X}_2. \tag{13.18}$$

Here we must have $\boldsymbol{U} = \boldsymbol{U}_2 = 1$, and $\boldsymbol{\Sigma} \in \Re^{1\times N}$ has the value $\sigma = \sqrt{\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}}$ in the first location and zero in the remainder locations. Further, $\boldsymbol{V} \in \Re^{N\times N}$ has the vector $\boldsymbol{S}/\sqrt{\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}}$ in the first column and vectors orthogonal to $\boldsymbol{S}$ in the other columns. Thus, we can write the singular value decomposition (13.8) of $\boldsymbol{X}_2$ as

$$\boldsymbol{X}_2 = \boldsymbol{\Sigma}_2\boldsymbol{V}_2^{\mathrm{T}}, \tag{13.19}$$

where we have

$$\boldsymbol{\Sigma}_2 = (\lambda^{-\frac{1}{2}}\sigma, 0, \ldots, 0), \tag{13.20}$$

and $\boldsymbol{V}_2 = \boldsymbol{V}$.

The one-sided analysis equation (13.8) then gives the following at the measurement location

$$\begin{aligned}\boldsymbol{S}^{\mathrm{a}} &= \boldsymbol{S}\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2} \\ &= \lambda^{\frac{1}{2}}\boldsymbol{\Sigma}_2\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2} \\ &= \left(\sigma\sqrt{1-\sigma^2/\lambda}, 0, \ldots, 0\right).\end{aligned} \tag{13.21}$$

The matrix $\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}$ is diagonal with ones on the diagonal except for the first element which is $\sqrt{1-\sigma^2/\lambda}$. Further, the first element contains all of the variance of the analysis at the measurement location, which also implies that the mean of the updated ensemble perturbations is non-zero.

We note that $\lambda = \sigma^2 + (N-1)\boldsymbol{C}_{\epsilon\epsilon}$, thus the variance at the measurement location becomes

$$\frac{\boldsymbol{S}^{\mathrm{a}}\boldsymbol{S}^{\mathrm{a}\mathrm{T}}}{N-1} = \frac{\sigma^2}{N-1}\left(1 - \frac{\sigma^2/(N-1)}{\sigma^2/(N-1) + \boldsymbol{C}_{\epsilon\epsilon}}\right), \tag{13.22}$$

which is identical to (3.15).

For state spaces where $n > 1$ the rank of the ensemble is reduced to one at the measurement locations, while the rows of $\boldsymbol{A}'$ corresponding to other grid points will generally not be parallel to $\boldsymbol{S}$ and the rank will be maintained. Note, however, that imposed spatial correlations will lead to poor conditioning of the ensemble at grid points close to the measurement location.

The update equation for symmetric square root scheme (13.11) includes the additional multiplication with $\boldsymbol{V}_2^{\mathrm{T}}$ and becomes

$$\begin{aligned}\boldsymbol{S}^{\mathrm{a}} &= \left(\sigma\sqrt{1-\sigma^2/\lambda}, 0, \ldots, 0\right)\boldsymbol{V}_2^{\mathrm{T}} \\ &= \sqrt{(1-\sigma^2/\lambda}\boldsymbol{S}.\end{aligned} \tag{13.23}$$

It is clear that the symmetric square root scheme is a symmetric contration of all the ensemble perturbations, where the zero mean of the perturbations is preserved.

13.1.6 Analysis update using a diagonal $\boldsymbol{C}_{\epsilon\epsilon}$

With more than one measurement the situation from the previous section changes but the same problem occur with the one-sided analysis equation (13.8) when $\boldsymbol{C}_{\epsilon\epsilon}$ is diagonal. We now consider the case with $1 < m < N$. Then the eigenvectors $\boldsymbol{Z}$, will be identical to the singular vectors $\boldsymbol{U}$, of $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$, and we can write

$$\boldsymbol{X}_2 = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{S} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}. \tag{13.24}$$

Thus, the singular value decomposition of $\boldsymbol{X}_2$ becomes again (13.6) but with

$$\boldsymbol{\Sigma}_2 = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{\Sigma}, \tag{13.25}$$

containing m nonzero elements on the diagonal, $\boldsymbol{V}_2 = \boldsymbol{V}$ and $\boldsymbol{U}_2 = \boldsymbol{I}$.

Then each of the m columns in $\boldsymbol{S}^{\mathrm{T}}$ will be contained in the space defined by the first m columns of $\boldsymbol{V}_2$. Thus, in the update, the first m ensemble perturbations will represent the analysis variance while the remainder will be zero.

The one-sided square root analysis scheme (13.8) results in an updated ensemble where the ensemble variance is reduced in directions defined by the rotation $\boldsymbol{V}_2$. In cases when $\boldsymbol{S}^{\mathrm{T}}$ is fully represented by a selection of singular vectors, as is the case when a single measurement is used and if m measurements are used with a diagonal $\boldsymbol{C}_{\epsilon\epsilon}$, then the ensemble variance at the measurement locations is represented by the first m ensemble members. This finding is consistent with the results from the Lorenz equations shown in Fig. 13.1.

13.2 Experiments

The impact of using the square root analysis scheme from the previous section will now be examined in some detail using the model and configuration from Sect. 11.7.

Experiment	N	β_{ini}	Residual	Std. dev.
F	100	1	0.69632	0.51328E-01
FS	100	1	0.68856	0.67178E-01
G	100	6	0.59581	0.39345E-01
GS	100	6	0.60496	0.40811E-01

Table 13.1. Summary of experiments. The first column is the experiment name, in the second column N is the ensemble size used, β_{ini} is a number which defines the size, $\beta_{\text{ini}}N$, of the start ensemble when using the improved sampling scheme from section 11.4 for generating the initial ensemble. The two last columns contain the average RMS errors of the 50 simulations in each experiment and the standard deviation of these

13.2.1 Overview of experiments

Four experiments are carried out as listed in Table 13.1. For each of the experiments, 50 EnKF simulations are performed to allow for a statistical comparison. In each simulation, the only difference is the random seed used. Thus, each simulation will have a different and random true state, first guess, initial ensemble, and set of measurements. The further details of the different experiments are as follows:

Exp. F is an experiment where a standard Monte Carlo ensemble is used for generating the 100 member initial ensemble without improved sampling. It is thus similar to and can be compared with *Exp. B* from Sect. 11.7.

Exp. FS is similar to *Exp. F* except that the mean preserving random rotation is used.

Exp. G is similar to *Exp. F* except that the initial ensemble is sampled from a start ensemble of 600 members as in *Exp. E* from Sect. 11.7. It examines the benefit of combined use of improved initial sampling and the square root algorithm.

Exp. GS is similar to *Exp. G* except that the mean preserving random rotation is used.

The analysis is computed from the square root implementation of the analysis scheme using the final update equation (13.13) with the update matrix defined by (13.17). As in Sect. 11.7 a full rank matrix $\boldsymbol{C} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}$ is assumed and inverted by computing the eigenvalue decomposition and using (13.3). The final update equation becomes

$$\begin{aligned}\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}\Big(&\mathbf{1}_N + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^{\mathrm{T}}\left(\boldsymbol{D} - \mathcal{M}\left[\boldsymbol{A}\right]\right)\mathbf{1}_N \\ &+ (\boldsymbol{I} - \mathbf{1}_N)\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}\Big).\end{aligned} \tag{13.26}$$

The residuals are computed as the Root Mean Square (RMS) errors of the difference between the estimate and the true solution taken over the complete

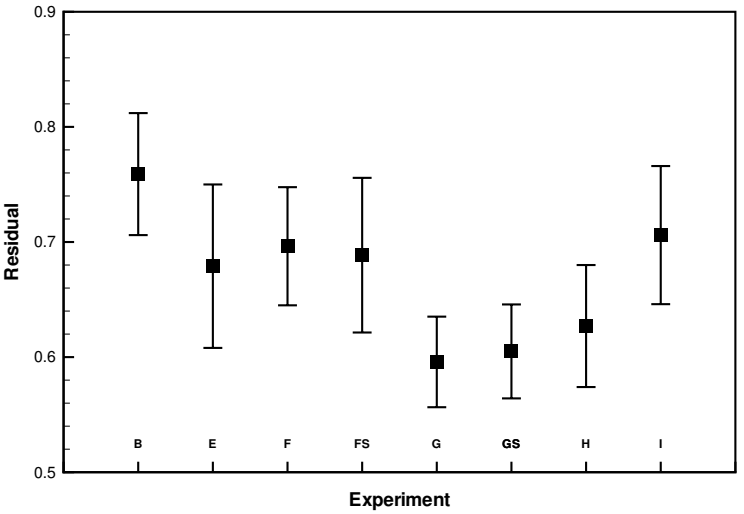

Fig. 13.2. Mean and standard deviation of the residuals from each of the experiments. Included are also *Exps. B,E,H* and *I* from Sect. 11.7

space and time domain. For each of the experiments we plot the mean and standard deviation of the residuals in Fig. 13.2.

The Table 13.2 gives the probabilities that the average residuals from the experiments are equal, as computed from the Student's t-test. Probabilities lower than, say 0.1, indicate statistically that the distributions from two experiments are significantly different.

It is also of interest to examine how well the predicted errors represent the actual residuals (RMS as a function of time). In the summary Figs. 13.3 we have plotted the average of the predicted errors from the 50 simulations as the thick full line. The thin full lines indicate the one standard deviation spread of the predicted errors from the 50 simulations. The average of the RMS errors from the 50 simulations is plotted as the thick dotted line, with associated one standard deviation spread shown by the dotted thin lines.

13.2.2 Impact of the square root analysis algorithm

The four experiments *Exps. F, FS, G* and *GS*, using the square root algorithm are compared with the results from the standard EnKF cases *Exps. B, E, H* and *I*, from Sect. 11.7. The *Exp. B* did not use improved sampling, *Exp. E* used improved sampling for the initial ensemble, *Exp. I* used improved sampling for the measurement perturbations, and *Exp. H* used improved sampling both for the initial ensemble and the measurement perturbations.

From the residuals plotted in Fig. 13.2, the random rotation does not seem to influence or degrade the results of the square root algorithm when used with

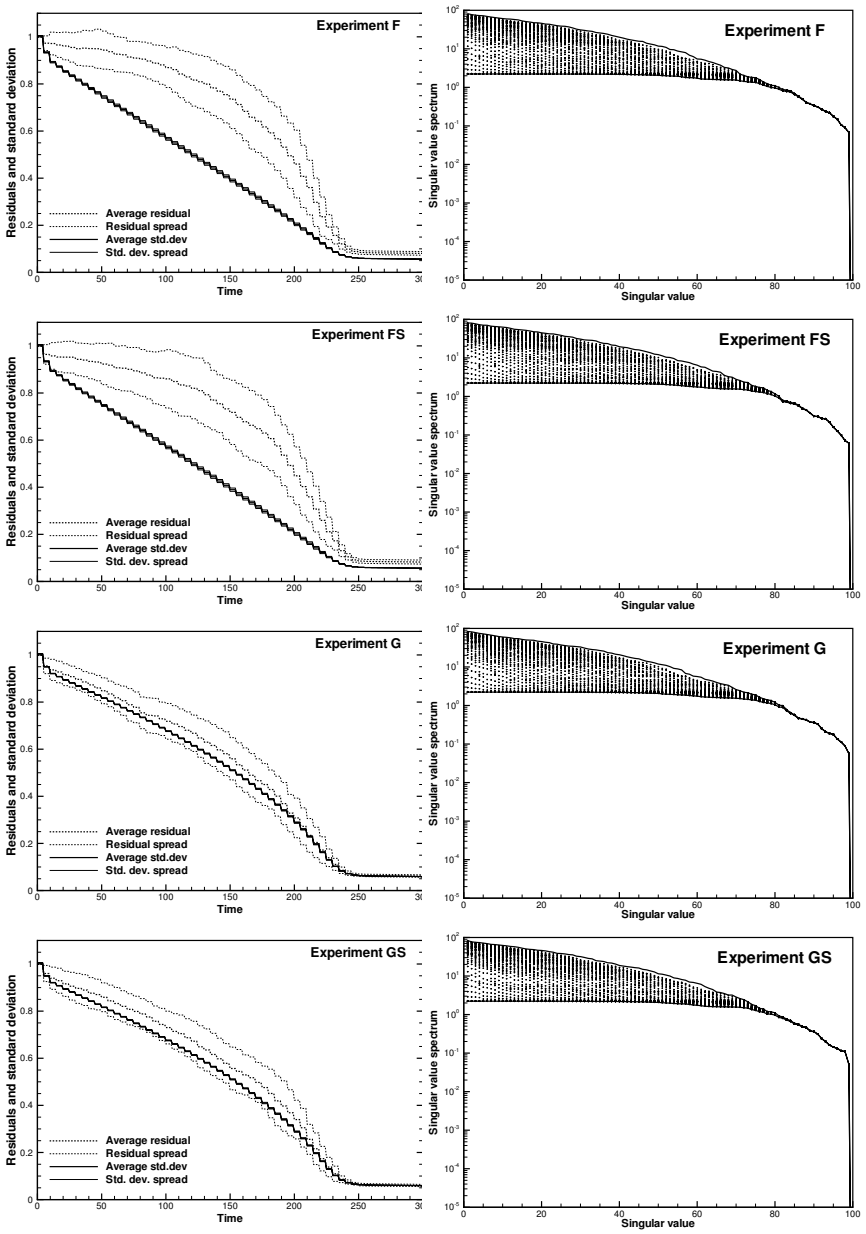

Fig. 13.3. Left column shows the time evolution for RMS residuals *(dashed lines)* and estimated standard deviations *(full lines)*. The thick lines show the means over the 50 simulations and the thin lines show the means plus/minus one standard deviation. The right column shows the time evolution of the ensemble singular value spectra for some of the experiments

Exp	*E*	*F*	*FS*	*G*	*GS*	*H*	*I*
B	0	0	0	0	0	0	0
E		**0.16**	**0.49**	0	0	0	0.04
F			**0.52**	0	0	0	**0.37**
FS				0	0	0	**0.17**
G					**0.26**	0	0
GS						0.02	0
H							0

Table 13.2. Statistical probability that two experiments provide an equal mean for the residuals as computed using the Student's t-test. A probability close to one indicates that it is likely that the two experiments provide distributions of residuals with similar mean

linear systems. The *Exps. F* and *FS* are very similar in performence, and so are *Exps. G* and *GS*. The time evolutions of the residuals in *Exps. F, FS, G* and *GS*, plotted in Fig. 13.3, show as with the standard EnKF, an underestimate of the residuals and otherwise behaviour that is similar to what was seen using the standard EnKF. However, there is a clear improvement in the cases *Exps. G* and *GS* where improved sampling is used for the initial ensemble.

It is of interest to examine how the rank and conditioning of the ensemble evolves in time and is impacted by the computation of the analysis. In the right column of Fig. 13.3 we have plotted the singular values for the ensemble at each analysis time. The initial singular spectrum is plotted as the upper thick line. Then the dotted lines indicate the reduction of the ensemble variance introduced at each analysis update, until the end of the experiment where the singular spectrum is given by the lower thick line. The initial spectra from *Exps. F* and *FS* are significantly different from the ones seen in the standard EnKF in Chap. 11. The square root scheme seems to lead to a reduction of variance that converges to a flat spectrum, which shows that the square root scheme weights the singular vectors more equally than the EnKF, which showed a tendency to loss of rank in the ensemble.

From Fig. 13.2 and Table 13.2 we observe that *Exps. F* and *FS* are similar in performence to the *Exps. E* and *I*. Further, the *Exps. G* and *GS* are superior to all the other experiments, but only slightly better than *Exp. H*. The square root scheme in *Exps. F* and *FS* provides an improvement to the standard EnKF in *Exp. B* and the results are similar to *Exp. I* where the EnKF is used with improved sampling for the measurement perturbations. When improved sampling is used for the initial ensemble, the square root scheme in *Exps. G* and *GS* provide the results with the lowest residuals, and slightly better than the standard EnKF with improved sampling of both the initial ensemble and the measurement perturbations in *Exp. H*.

14

Rank issues

It is in the previous chapters stated that the EnKF analysis scheme may have problems in cases where the number of measurements is larger than the number of members in the ensemble or when the matrix $\boldsymbol{C}$ for some reason has poor conditioning. In this chapter we will discuss these difficulties and propose algorithms that still makes it possible to use the EnKF analysis schemes in cases with poor conditioning. Thus, we provide an extended discussion of the rank problem as was introduced in *Evensen* (2004).

14.1 Pseudo inverse of $\boldsymbol{C}$

The matrix $\boldsymbol{C}$ that must be inverted in the analysis schemes is in (9.26) defined as

$$\boldsymbol{C} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}. \tag{14.1}$$

As in the previous chapters we define $\boldsymbol{S} = \mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}\prime}\big]$ as the measurements of the ensemble perturbations, and $\boldsymbol{C}_{\epsilon\epsilon}$ is the measurement error covariance matrix.

The analysis scheme for the EnKF is in (11.53) given as

$$\boxed{\begin{array}{l}\text{Standard EnKF analysis}\\ \boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\big(\boldsymbol{D} - \mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\Big),\end{array}} \tag{14.2}$$

with $\boldsymbol{D}$ being the ensemble of perturbed measurements.

In the square root scheme we compute the update of the mean which is derived from (14.2) by multiplication from the right by $\mathbf{1}_N$, where $\mathbf{1}_N$ is an N-dimensional quadratic matrix with all elements equal to $1/N$. Thus, we get the update for the mean (13.2), written as

$$\overline{\boldsymbol{A}}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\left(\mathbf{1}_N + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\big(\overline{\boldsymbol{D}} - \mathcal{M}\big[\overline{\boldsymbol{A}}^{\mathrm{f}}\big]\big)\right). \tag{14.3}$$

The perturbations are updated according to (13.12), i.e.

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_14,

$$\boldsymbol{A}^{\mathrm{a}\prime} = \boldsymbol{A}^{\mathrm{f}\prime}\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}, \tag{14.4}$$

which is derived from a factorization of (13.1), i.e.

$$\boldsymbol{A}^{\mathrm{a}\prime}\boldsymbol{A}^{\mathrm{a}\prime\mathrm{T}} = \boldsymbol{A}^{\mathrm{f}\prime}\left(\boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{S}\right)\boldsymbol{A}^{\mathrm{f}\prime\mathrm{T}}. \tag{14.5}$$

Equations (14.3) and (14.4) can be combined into one single equation, similar to (13.26), as

$$\boxed{\begin{aligned}&\text{Standard square root analysis}\\ &\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{1}_N + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\big(\boldsymbol{D} - \mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\boldsymbol{1}_N\\ &\qquad + (\boldsymbol{I} - \boldsymbol{1}_N)\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}\Big).\end{aligned}} \tag{14.6}$$

For the definition of the various matrices we refer to Chap. 13 where the square root scheme was derived.

It is seen that in both the EnKF and the square root algorithm we need to compute the inverse of $\boldsymbol{C}$. In the previous discussion an eigenvalue factorization is used when inverting $\boldsymbol{C}$. In cases where the dimension of $\boldsymbol{C}$ is large, or if nearly dependent measurements are assimilated, it is possible that $\boldsymbol{C}$ becomes numerically singular and the pseudo inverse $\boldsymbol{C}^+$ of $\boldsymbol{C}$ must be used. It is convenient to formulate the analysis schemes in terms of the pseudo inverse, since we have $\boldsymbol{C}^+ \equiv \boldsymbol{C}^{-1}$, when $\boldsymbol{C}$ is of full rank. The algorithm is then valid in the general case.

14.1.1 Pseudo inverse

The pseudo inverse of the quadratic matrix $\boldsymbol{C}$ with eigenvalue factorization

$$\boldsymbol{C} = \boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{Z}^{\mathrm{T}}, \tag{14.7}$$

is defined as

$$\boldsymbol{C}^+ = \boldsymbol{Z}\boldsymbol{\Lambda}^+\boldsymbol{Z}^{\mathrm{T}}. \tag{14.8}$$

The matrix $\boldsymbol{\Lambda}^+$ is diagonal and with $p = \mathrm{rank}(\boldsymbol{C})$ it is defined as

$$\mathrm{diag}(\boldsymbol{\Lambda}^+) = (\lambda_1^{-1}, \ldots, \lambda_p^{-1}, 0, \ldots, 0), \tag{14.9}$$

with the eigenvalues $\lambda_i \geq \lambda_{i+1}$.

The pseudo inverse has the following properties

$$\boldsymbol{C}\boldsymbol{C}^+\boldsymbol{C} = \boldsymbol{C}, \qquad \boldsymbol{C}^+\boldsymbol{C}\boldsymbol{C}^+ = \boldsymbol{C}^+, \tag{14.10}$$

$$(\boldsymbol{C}^+\boldsymbol{C})^{\mathrm{T}} = \boldsymbol{C}^+\boldsymbol{C}, \qquad (\boldsymbol{C}\boldsymbol{C}^+)^{\mathrm{T}} = \boldsymbol{C}\boldsymbol{C}^+. \tag{14.11}$$

Furthermore,

$$\boldsymbol{x} = \boldsymbol{C}^{+}\boldsymbol{b}, \tag{14.12}$$

is the least squares solution of the problem

$$\boldsymbol{C}\boldsymbol{x} = \boldsymbol{b}, \tag{14.13}$$

when $\boldsymbol{C}$ is singular.

14.1.2 Interpretation

It is useful to attempt an interpretation of the algorithm when using the pseudo inverse for $\boldsymbol{C}$. We start by storing the p nonzero elements of $\mathrm{diag}(\boldsymbol{\Lambda}^{+})$ on the diagonal of $\boldsymbol{\Lambda}_p^{-1} \in \Re^{p\times p}$, i.e.

$$\mathrm{diag}(\boldsymbol{\Lambda}_p^{-1}) = (\lambda_1^{-1}, \ldots, \lambda_p^{-1}). \tag{14.14}$$

We then define the matrix containing the first p eigenvectors in $\boldsymbol{Z}$ as $\boldsymbol{Z}_p = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_p) \in \Re^{m\times p}$. It is clear that the product $\boldsymbol{Z}_p\boldsymbol{\Lambda}_p^{-1}\boldsymbol{Z}_p^{\mathrm{T}}$ is the Moore-Penrose or pseudo inverse of the original matrix $\boldsymbol{C}$.

We now define the projected measurement operator $\widetilde{\mathcal{M}} \in \Re^{p\times n}$ as

$$\widetilde{\mathcal{M}} = \boldsymbol{Z}_p^{\mathrm{T}}\mathcal{M}, \tag{14.15}$$

the ensemble of p projected measurements

$$\widetilde{\boldsymbol{D}} = \boldsymbol{Z}_p^{\mathrm{T}}\boldsymbol{D}, \tag{14.16}$$

and the p projected measurements of the ensemble perturbations $\widetilde{\boldsymbol{S}} \in \Re^{p\times N}$, as

$$\widetilde{\boldsymbol{S}} = \boldsymbol{Z}_p^{\mathrm{T}}\mathcal{M}\big[\boldsymbol{A}'\big] = \widetilde{\mathcal{M}}\big[\boldsymbol{A}'\big] = \boldsymbol{Z}_p^{\mathrm{T}}\boldsymbol{S}. \tag{14.17}$$

This corresponds to the use of a measurement antenna which is oriented along the p dominant principal directions of $\boldsymbol{C}$ (see *Bennett*, 1992, Chap. 6). The analysis equation in the original EnKF analysis scheme then becomes

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \widetilde{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{\Lambda}_p^{-1}\big(\widetilde{\boldsymbol{D}} - \widetilde{\mathcal{M}}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\Big). \tag{14.18}$$

Thus, the analysis is just the assimilation of the p rotated and projected measurements in the space where $\widetilde{\boldsymbol{C}} = \boldsymbol{\Lambda}_p$ is diagonal.

14.1.3 Analysis schemes using the pseudo inverse of $\boldsymbol{C}$

The modification required for the EnKF and square root analysis schemes to use the pseudo inverse of $\boldsymbol{C}$ is minor. The same equations and derivation are used, it is only necessary to perform a truncation of the spectrum at the desired variance level, i.e. one need to decide how many eigenvalues to include and set the remainder to zero. Then $\boldsymbol{\Lambda}^{+}$ is defined and used instead of $\boldsymbol{\Lambda}^{-1}$.

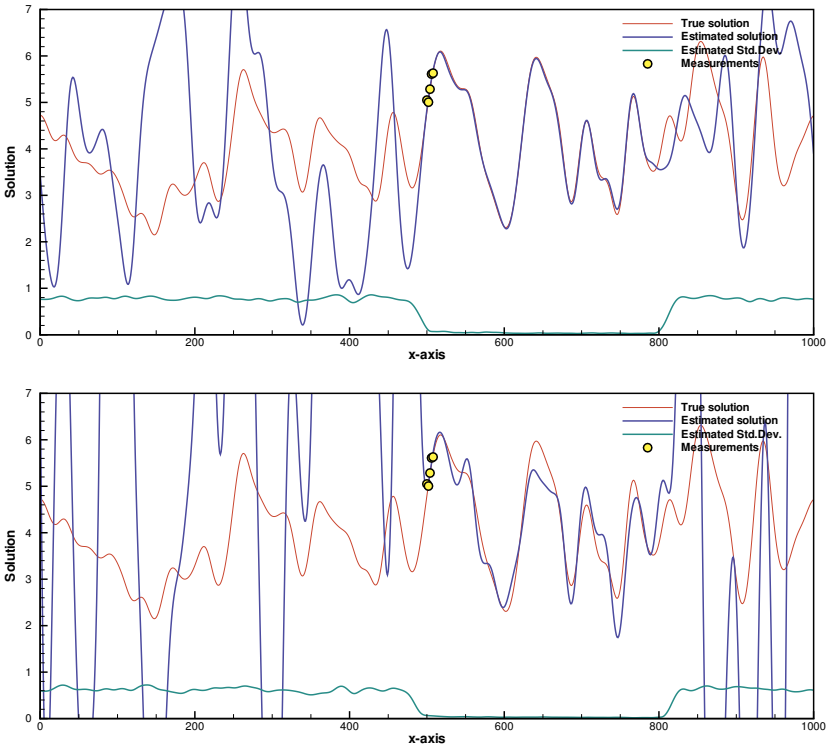

Fig. 14.1. Solution at the final time using the traditional EnKF analysis scheme with pseudo inversion of $\boldsymbol{C}$. The upper plot is the solution with truncation at 90% of the variance of the eigenvalue spectrum, while the lower plot is with truncation at 99.9% of the variance

14.1.4 Example

The advection example from Sects. 11.7 and 13.2 is now used to illustrate the importance of being able to handle a rank deficient $\boldsymbol{C}$. We first construct a case where we have five measurements located at neighbouring grid points. The measurement error covariance matrix is also nondiagonal, and it is assumed that the measurement errors are correlated with a Gaussian covariance function of de-correlation length equal to 20. This leads to a matrix $\boldsymbol{C}$ with a ratio of the largest over smallest eigenvalue of order 10^5. Thus, the conditioning of $\boldsymbol{C}$ is rather poor and the use of a pseudo inversion may be advantageous.

We now run two experiments similar to *Exp. E* from Sect. 11.7, and *Exp. G* from Sect. 13.2, and plot the solution at the final time $t = 300$, for different truncations of the eigenvalue spectrum. The results are plotted in respectively Figs. 14.1 and 14.2 for the traditional EnKF and the square root analysis

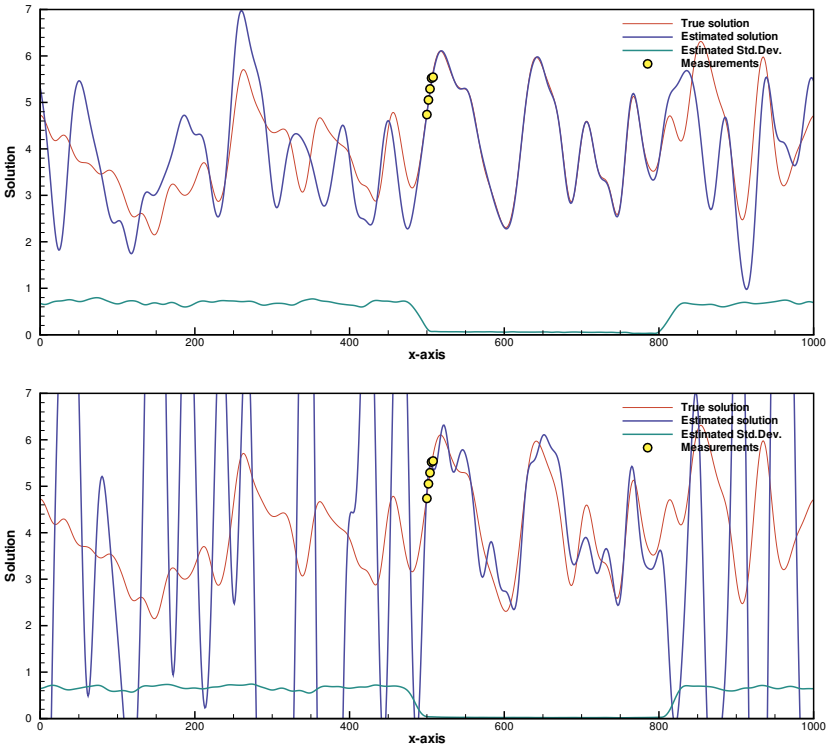

Fig. 14.2. Solution at the final time using the square root analysis scheme with pseudo inversion of $\boldsymbol{C}$. The upper plot is the solution with truncation at 90% of the variance of the eigenvalue spectrum, while the lower plot is with truncation at 99.9% of the variance

algorithms. It is seen that the inversion, using a truncation of the eigenvalue spectrum where 90% of the variance, corresponding to a single eigenvalue, is retained, leads to stable solutions. On the other hand, when the truncation is accounting for 99.9% of the variance, which retains four eigenvalues, both the traditional EnKF and square root scheme result in unstable inversions.

We now increase the number of measurements to 200, and use the same Gaussian error covariance matrix for the measurement errors. The results at $t = 25$, after 5 updates with measurements, are plotted in Fig. 14.3 for the traditional EnKF analysis and the square root analysis. In this case around 40 significant eigenvalues were included when a truncation at 99% of the variance was specified. It is clear that both schemes produce a stable inversion which is consistent with the measurements. For this case we also plotted the eigenvalue spectrum of $\boldsymbol{C}$ at each of the updates in Fig. 14.4. It is seen that there are around 40–50 significant eigenvalues for all the updates, and there is

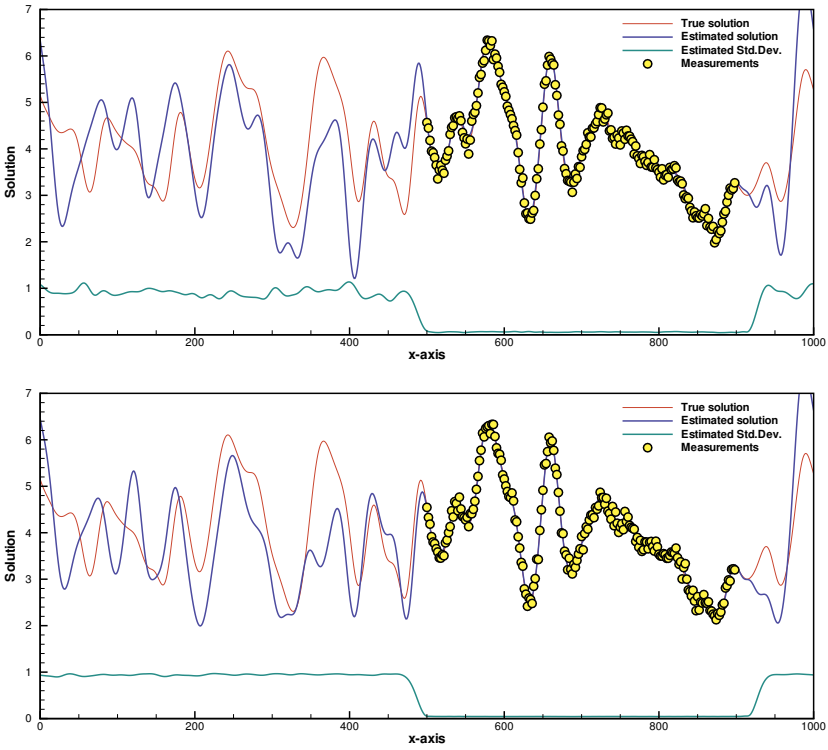

Fig. 14.3. Solution at time $t = 25$ using the EnKF scheme in the upper plot and square root scheme in the lower plot, and with a truncation accounting for 99% of the variance of the eigenvalue spectrum

a reduction of the variance for all of the significant eigenvalues, corresponding to the reduction of ensemble variance at the measurement locations.

Thus, it is clear that both the EnKF and the square root scheme can handle cases with dependent measurements and a larger number of measurements than ensemble members. Note that we may expect problems if the number of significant eigenvalues becomes larger than the number of ensemble members.

14.2 Efficient subspace pseudo inversion

In cases with many measurements the computational cost becomes large since Nm^2 operations are required to form the matrix $\boldsymbol{C}$ and the eigenvalue decomposition requires $\mathcal{O}(m^3)$ operations. An alternative inversion algorithm which reduces the factorization of the $m \times m$ matrix to a factorization of an $N \times N$ matrix is now presented. The algorithm computes the inverse in the

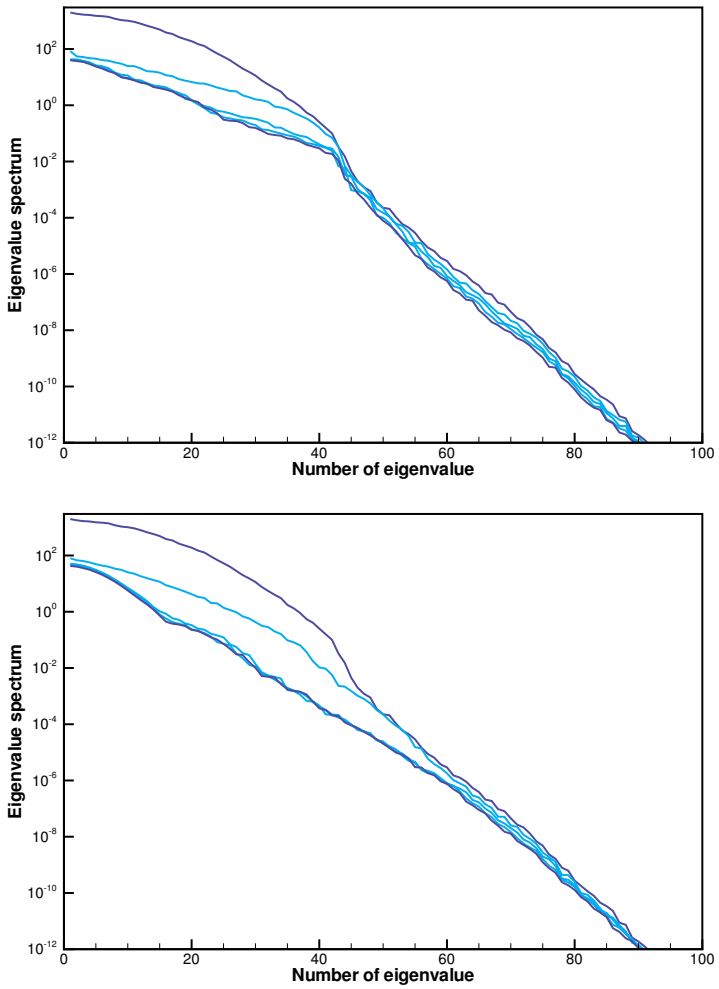

Fig. 14.4. Eigenvalue spectrum of $\boldsymbol{C}$ in the cases shown in Fig. 14.3. Results from the EnKF and square root schemes are shown in the upper and lower plots, respectively.

N-dimensional ensemble space rather than the m-dimensional measurement space.

14.2.1 Derivation of the subspace pseudo inverse

We start by assuming that $\boldsymbol{S}$ has rank $p \leq \min(m, N-1)$. The equality can be satisfied when the ensemble consists of linearly independent members and the measurement operator has full rank, i.e. the measurements are independent.

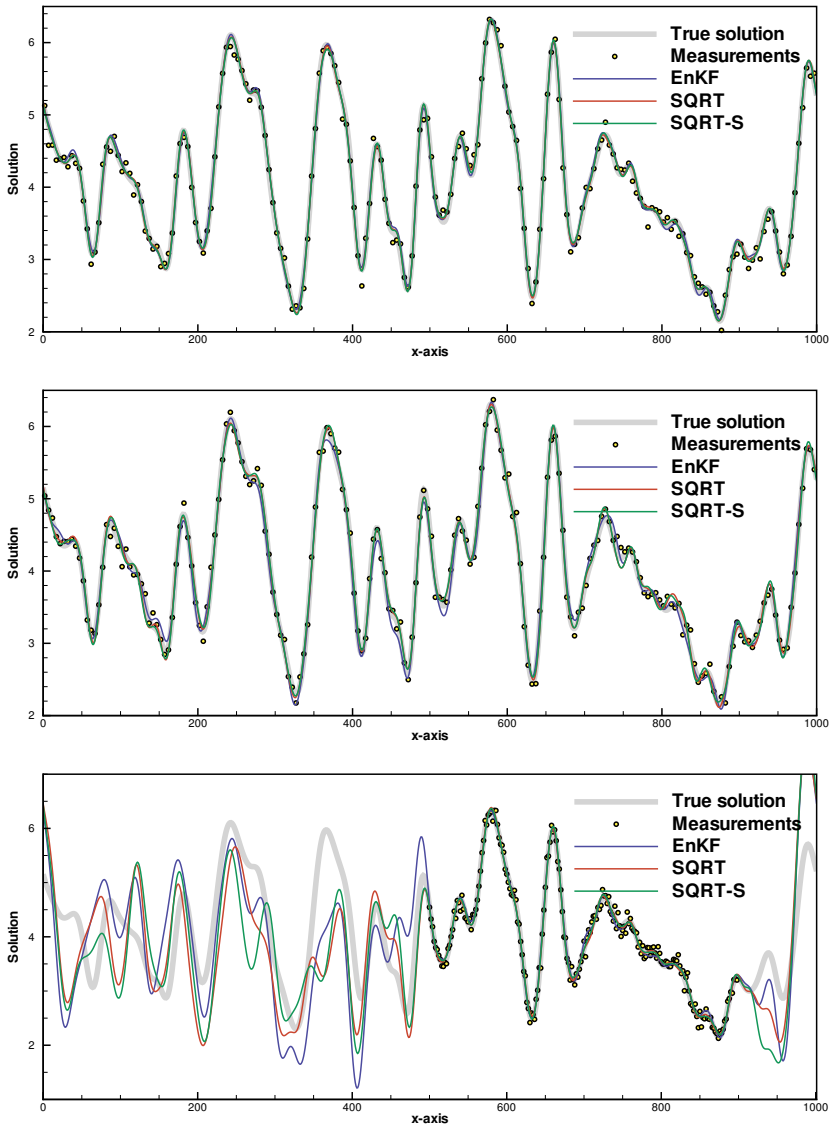

Fig. 14.5. Solution after 5 updates using the traditional EnKF, the square root scheme, and the new square root scheme with subspace projection of $\boldsymbol{C}_{\epsilon\epsilon}$ and pseudo inversion of $\boldsymbol{C}$. The upper plot shows the solution with uniform distribution of 200 measurements and a diagonal $\boldsymbol{C}_{\epsilon\epsilon}$. The middle plot is similar to the upper one but with a nondiagonal $\boldsymbol{C}_{\epsilon\epsilon}$. The lower plot has clustered the measurements and also uses a nondiagonal $\boldsymbol{C}_{\epsilon\epsilon}$

The SVD of $\boldsymbol{S}$ is

$$\boldsymbol{U}_0\boldsymbol{\Sigma}_0\boldsymbol{V}_0^{\mathrm{T}} = \boldsymbol{S}, \tag{14.19}$$

with $\boldsymbol{U}_0 \in \Re^{m\times m}$, $\boldsymbol{\Sigma}_0 \in \Re^{m\times N}$ and $\boldsymbol{V}_0 \in \Re^{N\times N}$. The SVD of $\boldsymbol{S}$ can be computed using $\mathcal{O}(6mN^2 + N^3)$ floating point operations when only the first N singular vectors are needed, and this is a significant saving when $m \gg N$. The subspace $\mathcal{S}$ is now defined by the first p singular vectors of $\boldsymbol{S}$ as contained in $\boldsymbol{U}_0$.

The pseudo inverse of $\boldsymbol{S}$ is defined as

$$\boldsymbol{S}^{+} = \boldsymbol{V}_0\boldsymbol{\Sigma}_0^{+}\boldsymbol{U}_0^{\mathrm{T}}, \tag{14.20}$$

where $\boldsymbol{\Sigma}_0^{+} \in \Re^{N\times m}$ is a diagonal matrix with elements defined as $\mathrm{diag}(\boldsymbol{\Sigma}_0^{+}) = (\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_p^{-1}, \ldots, 0)$. Thus, by computing the pseudo inversion in (14.20) it is also possible to use the algorithm with the number of measurements being less than $N-1$ and also with dependent measurements or dependent ensemble members.

We define $\tilde{\boldsymbol{I}}_p \in \Re^{m\times m}$, which has the first p diagonal elements equal to one and the remainder of the elements in the matrix are zero, from the matrix product $\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{+} = \tilde{\boldsymbol{I}}_p$.

Using the singular value decomposition (14.19) in the expression for $\boldsymbol{C}$, as defined in (14.1), we obtain

$$\begin{aligned}
\boldsymbol{C} &= \big(\boldsymbol{U}_0\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}\big) && (14.21)\\
&= \boldsymbol{U}_0\big(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{\mathrm{T}} + (N-1)\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0\big)\boldsymbol{U}_0^{\mathrm{T}} && (14.22)\\
&\approx \boldsymbol{U}_0\boldsymbol{\Sigma}_0\big(\boldsymbol{I} + (N-1)\boldsymbol{\Sigma}_0^{+}\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0\boldsymbol{\Sigma}_0^{+\mathrm{T}}\big)\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}} && (14.23)\\
&= \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)(\boldsymbol{S}\boldsymbol{S}^{+})\boldsymbol{C}_{\epsilon\epsilon}(\boldsymbol{S}\boldsymbol{S}^{+})^{\mathrm{T}}. && (14.24)
\end{aligned}$$

In (14.22) the matrix $\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0$ is the projection of the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$ onto the space spanned by the m singular vectors of $\boldsymbol{S}$, contained in the columns of $\boldsymbol{U}_0$.

Then in (14.23) we introduce an approximation by effectively multiplying $\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0$ from left and right by the matrix $\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{+} = \tilde{\boldsymbol{I}}_p \in \Re^{m\times m}$. Thus, we extract the part of $\boldsymbol{C}_{\epsilon\epsilon}$ contained in the subspace consisting of the p dominant directions in $\boldsymbol{U}_0$, i.e. the subspace $\mathcal{S}$.

The matrix $\boldsymbol{S}\boldsymbol{S}^{+} = \boldsymbol{U}_0\tilde{\boldsymbol{I}}_p\boldsymbol{U}_0^{\mathrm{T}}$ in (14.24) is a Hermitian and normal matrix. It is also an orthogonal projection onto $\mathcal{S}$. Thus, we essentially adopt a low-rank representation for $\boldsymbol{C}_{\epsilon\epsilon}$ which is contained in the same subspace as the ensemble perturbations in $\boldsymbol{S}$.

We use the expression for $\boldsymbol{C}$ as given in (14.23), i.e.

$$\boldsymbol{C} \approx \boldsymbol{U}_0\boldsymbol{\Sigma}_0(\boldsymbol{I} + \boldsymbol{X}_0)\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}}, \tag{14.25}$$

where we have defined

$$\boldsymbol{X}_0 = (N-1)\boldsymbol{\Sigma}_0^{+}\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0\boldsymbol{\Sigma}_0^{+\mathrm{T}}, \tag{14.26}$$

which is an $N \times N$ matrix with rank equal to p and it requires $m^2N + mN^2 + mN$ floating point operations to form it. We then proceed with an eigenvalue decomposition

$$\boldsymbol{Z}_1 \boldsymbol{\Lambda}_1 \boldsymbol{Z}_1^{\mathrm{T}} = \boldsymbol{X}_0, \tag{14.27}$$

where all matrices are $N \times N$, and insert this in (14.25) to get

$$\begin{aligned} \boldsymbol{C} &\approx \boldsymbol{U}_0 \boldsymbol{\Sigma}_0 (\boldsymbol{I} + \boldsymbol{Z}_1 \boldsymbol{\Lambda}_1 \boldsymbol{Z}_1^{\mathrm{T}}) \boldsymbol{\Sigma}_0^{\mathrm{T}} \boldsymbol{U}_0^{\mathrm{T}} \\ &= \boldsymbol{U}_0 \boldsymbol{\Sigma}_0 \boldsymbol{Z}_1 (\boldsymbol{I} + \boldsymbol{\Lambda}_1) \boldsymbol{Z}_1^{\mathrm{T}} \boldsymbol{\Sigma}_0^{\mathrm{T}} \boldsymbol{U}_0^{\mathrm{T}}. \end{aligned} \tag{14.28}$$

Now the pseudo inverse of $\boldsymbol{C}$ becomes

$$\begin{aligned} \boldsymbol{C}^{+} &\approx (\boldsymbol{U}_0 \boldsymbol{\Sigma}_0^{+\mathrm{T}} \boldsymbol{Z}_1)(\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-1} (\boldsymbol{U}_0 \boldsymbol{\Sigma}_0^{+\mathrm{T}} \boldsymbol{Z}_1)^{\mathrm{T}} \\ &= \boldsymbol{X}_1 (\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-1} \boldsymbol{X}_1^{\mathrm{T}}, \end{aligned} \tag{14.29}$$

where we have defined $\boldsymbol{X}_1 \in \Re^{m \times N}$ of rank $N - 1$ as

$$\boldsymbol{X}_1 = \boldsymbol{U}_0 \boldsymbol{\Sigma}_0^{+\mathrm{T}} \boldsymbol{Z}_1. \tag{14.30}$$

14.2.2 Analysis schemes based on the subspace pseudo inverse

By replacing $\boldsymbol{C}^{-1}$ in (14.2) with the pseudo inverse $\boldsymbol{C}^{+}$, from (14.29), we can easily compute the EnKF analysis using the subspace pseudo inversion by carrying out the matrix multiplications in

EnKF analysis by subspace pseudo inversion

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}} \left(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}} \boldsymbol{X}_1 (\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-1} \boldsymbol{X}_1^{\mathrm{T}} \left(\boldsymbol{D} - \mathcal{M}\left[\boldsymbol{A}^{\mathrm{f}} \right] \right) \right). \tag{14.31}$$

Similarly the square root algorithm uses (14.3) with $\boldsymbol{C}^{-1}$ replaced by $\boldsymbol{C}^{+}$ from (14.29),

$$\overline{\boldsymbol{A}}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}} \left(\boldsymbol{1}_N + \boldsymbol{S}^{\mathrm{T}} \boldsymbol{X}_1 (\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-1} \boldsymbol{X}_1^{\mathrm{T}} \left(\overline{\boldsymbol{D}} - \mathcal{M}\left[\overline{\boldsymbol{A}}^{\mathrm{f}} \right] \right) \right), \tag{14.32}$$

to compute the updated ensemble mean.

Using the expression (14.5) together with the pseudo inverse from (14.29) we can derive the update equation for the analysis perturbations in the square root scheme

$$\begin{aligned} \boldsymbol{A}^{\mathrm{a}\prime} \boldsymbol{A}^{\mathrm{a}\prime\mathrm{T}} &= \boldsymbol{A}^{\mathrm{f}\prime} \left(\boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}} \boldsymbol{C}^{+} \boldsymbol{S} \right) \boldsymbol{A}^{\mathrm{f}\prime\mathrm{T}} \\ &= \boldsymbol{A}^{\mathrm{f}\prime} \left(\boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}} \boldsymbol{X}_1 (\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-1} \boldsymbol{X}_1^{\mathrm{T}} \boldsymbol{S} \right) \boldsymbol{A}^{\mathrm{f}\prime\mathrm{T}} \\ &= \boldsymbol{A}^{\mathrm{f}\prime} \Big(\boldsymbol{I} - \left((\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-\frac{1}{2}} \boldsymbol{X}_1^{\mathrm{T}} \boldsymbol{S} \right)^{\mathrm{T}} \left((\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-\frac{1}{2}} \boldsymbol{X}_1^{\mathrm{T}} \boldsymbol{S} \right) \Big) \boldsymbol{A}^{\mathrm{f}\prime\mathrm{T}} \\ &= \boldsymbol{A}^{\mathrm{f}\prime} \left(\boldsymbol{I} - \boldsymbol{X}_2^{\mathrm{T}} \boldsymbol{X}_2 \right) \boldsymbol{A}^{\mathrm{f}\prime\mathrm{T}}, \end{aligned} \tag{14.33}$$

where we have defined $\boldsymbol{X}_2$ as

$$\boldsymbol{X}_2 = (\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-\frac{1}{2}} \boldsymbol{X}_1^{\mathrm{T}} \boldsymbol{S} = (\boldsymbol{I} + \boldsymbol{\Lambda}_1)^{-\frac{1}{2}} \boldsymbol{Z}_1^{\mathrm{T}} \check{\boldsymbol{I}}_p \boldsymbol{V}_0^{\mathrm{T}}, \tag{14.34}$$

which also has rank equal to p. We then end up with the final update equation (14.4) by following the derivation defined in (13.6–13.7).

Equations (14.32) and (14.4) can be combined into one single equation, similar to (14.6), as

$$\boxed{\begin{aligned}&\text{SQRT analysis by subspace pseudo inversion}\\ &\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{1}_N + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{X}_1(\boldsymbol{I}+\boldsymbol{\Lambda}_1)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\big(\boldsymbol{D} - \mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\boldsymbol{1}_N\\ &\qquad + (\boldsymbol{I}-\boldsymbol{1}_N)\boldsymbol{V}_2\sqrt{\boldsymbol{I} - \boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}\Big).\end{aligned}} \tag{14.35}$$

It is clear that, for $m > p$, this subspace algorithm will be an approximation except for some special cases. First, if $\boldsymbol{C}_{\epsilon\epsilon}$ is diagonal, then the matrix $\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$ and $\boldsymbol{C}$ will have the same eigenvectors as defined by $\boldsymbol{U}_0$, thus there is no approximation involved. On the other hand if $\boldsymbol{C}_{\epsilon\epsilon}$ is nondiagonal the eigenvectors will differ and the projection onto the $\mathcal{S}$-space eliminates the part of $\boldsymbol{C}$ which is orthogonal to the $\mathcal{S}$-space. Fortunately, in many applications this is a modest approximation.

Interestingly, the update of the perturbations in the square root algorithm does not suffer from this approximation since $\boldsymbol{C}^{-1}$ is already projected onto the $\mathcal{S}$-space through the matrix product $\boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{S}$ in (14.5).

14.2.3 An interpretation of the subspace pseudo inversion

A simple interpretation of the subspace pseudo inversion for the case when $m \gg N$ is given by *Skjervheim et al.* (2006). We start by computing the singular value factorization of $\boldsymbol{S}$ as in (14.19), and realize that $\boldsymbol{\Sigma}_0$ is diagonal and only the first $p \leq N-1$ singular values are larger than zero, i.e. the rank of $\boldsymbol{S}$ equals p. We then write the EnKF analysis scheme (14.2), with $\boldsymbol{D}' = \big(\boldsymbol{D} - \mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)$, as

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\Big(\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}\Big)^{-1}\boldsymbol{D}'\Big) \tag{14.36}$$

$$= \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\Big(\boldsymbol{U}_0\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}} + (N-1)\boldsymbol{C}_{\epsilon\epsilon}\Big)^{-1}\boldsymbol{D}'\Big) \tag{14.37}$$

$$= \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\Big(\boldsymbol{U}_0\big(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{\mathrm{T}} + (N-1)\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0\big)\boldsymbol{U}_0^{\mathrm{T}}\Big)^{-1}\boldsymbol{D}'\Big) \tag{14.38}$$

$$= \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{U}_0\Big(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{\mathrm{T}} + (N-1)\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0\Big)^{-1}\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{D}'\Big) \tag{14.39}$$

$$= \boldsymbol{A}^{\mathrm{f}}\Big(\boldsymbol{I} + \widehat{\boldsymbol{S}}^{\mathrm{T}}\Big(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{\mathrm{T}} + (N-1)\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_0\Big)^{-1}\widehat{\boldsymbol{D}}'\Big). \tag{14.40}$$

Here we have defined the rotated operators

$$\widehat{\boldsymbol{D}}' = \boldsymbol{U}_0^{\mathrm{T}} \boldsymbol{D}', \tag{14.41}$$

$$\widehat{\mathcal{M}} = \boldsymbol{U}_0^{\mathrm{T}} \mathcal{M}, \tag{14.42}$$

$$\widehat{\boldsymbol{S}} = \boldsymbol{U}_0^{\mathrm{T}} \boldsymbol{S} = \widehat{\mathcal{M}} \boldsymbol{A}^{\mathrm{f}\prime}. \tag{14.43}$$

It is clear that the original assimilation of m measurements in (14.36) is equivalent to the assimilation of m rotated measurements in (14.40), where the rotation is defined such that the matrix product $\widehat{\boldsymbol{S}}\widehat{\boldsymbol{S}}^{\mathrm{T}}$ becomes diagonal.

We now take this one step further and define the projection operator $\boldsymbol{U}_{0p} = \boldsymbol{S}\boldsymbol{S}^{+}$ which consists of the first p columns of $\boldsymbol{U}$. We can then define the projections

$$\widehat{\boldsymbol{D}}_p' = \boldsymbol{U}_{0p}^{\mathrm{T}} \boldsymbol{D}', \tag{14.44}$$

$$\widehat{\mathcal{M}}_p = \boldsymbol{U}_{0p}^{\mathrm{T}} \mathcal{M}, \tag{14.45}$$

$$\widehat{\boldsymbol{S}}_p = \boldsymbol{U}_{0p}^{\mathrm{T}} \boldsymbol{S} = \widehat{\mathcal{M}}_p \boldsymbol{A}^{\mathrm{f}\prime}, \tag{14.46}$$

all of dimension $\Re^{p \times N}$, and in addition we define $\boldsymbol{\Sigma}_{0p} \in \Re^{p \times p}$ to hold the p significant singular values on the diagonal. We can then write an approximate EnKF analysis equation as

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}} \left(\boldsymbol{I} + \widehat{\boldsymbol{S}}_p^{\mathrm{T}} \left(\boldsymbol{\Sigma}_{0p} \boldsymbol{\Sigma}_{0p}^{\mathrm{T}} + (N-1) \boldsymbol{U}_{0p}^{\mathrm{T}} \boldsymbol{C}_{\epsilon\epsilon} \boldsymbol{U}_{0p} \right)^{-1} \widehat{\boldsymbol{D}}_p' \right). \tag{14.47}$$

It is left as an exercise to show that this equation is identical to (14.31). Thus, we can interpret the subspace EnKF analysis scheme as the assimilation of a set of measurements after they have been projected onto the subspace $\mathcal{S}$ as defined by the first p singular vectors of $\boldsymbol{S}$. This projection allows us to assimilate very large data sets to a low cost in a stable algorithm. However, one can imagine cases where the subspace $\mathcal{S}$ is too small to properly represent the measurements. This problem can be resolved by either using a larger ensemble size or one may use a local analysis update as will be discussed in the Appendix.

14.3 Subspace inversion using a low-rank $\boldsymbol{C}_{\epsilon\epsilon}$

With large data sets one will have to generate and store the measurement error covariance matrix, $\boldsymbol{C}_{\epsilon\epsilon} \in \Re^{m \times m}$, and multiply it with the singular vectors in $\boldsymbol{U}_0$ at the cost of Nm^2 floating point operations. In the EnKF we have simulated measurement perturbations that reflect the error statistics of the measurement errors. It is clear that given the measurement perturbations we can use these to represent a low-rank approximation of the measurement error covariance matrix.

14.3.1 Derivation of the pseudo inverse

We now replace $\boldsymbol{C}_{\epsilon\epsilon}$ with a low-rank version $\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}} = \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}/(N-1)$, in (14.24) to get

$$\begin{aligned}\boldsymbol{C} &= \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}} \\ &\approx \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + (\boldsymbol{S}\boldsymbol{S}^{+})\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}(\boldsymbol{S}\boldsymbol{S}^{+}) \\ &= \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}} + \widehat{\boldsymbol{E}}\widehat{\boldsymbol{E}}^{\mathrm{T}},\end{aligned} \tag{14.48}$$

where $\widehat{\boldsymbol{E}} = (\boldsymbol{S}\boldsymbol{S}^{+})\boldsymbol{E}$ is the projection of $\boldsymbol{E}$ onto the first p singular vectors in $\boldsymbol{U}_0$, with p still being the rank of $\boldsymbol{S}$. When we project $\boldsymbol{E}$ onto $\mathcal{S}$ we reject all possible contributions in $\mathcal{S}^{\perp}$, and we can only account for the measurement variance contained in $\mathcal{S}$.

Replacing $\boldsymbol{C}_{\epsilon\epsilon}$ with $\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}/(N-1)$ in (14.23) we get

$$\boldsymbol{C} \approx \boldsymbol{U}_0\boldsymbol{\Sigma}_0\big(\boldsymbol{I} + \boldsymbol{\Sigma}_0^{+}\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{U}_0\boldsymbol{\Sigma}_0^{+\mathrm{T}}\big)\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}} \tag{14.49}$$

$$= \boldsymbol{U}_0\boldsymbol{\Sigma}_0\big(\boldsymbol{I} + \boldsymbol{X}_0\boldsymbol{X}_0^{\mathrm{T}}\big)\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}}, \tag{14.50}$$

where we have defined

$$\boldsymbol{X}_0 = \boldsymbol{\Sigma}_0^{+}\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{E}, \tag{14.51}$$

which is an $N \times N$ matrix with rank equal to $N-1$ and it requires $mN^2 + N^2$ floating point operations to form it. The approximate equality sign introduced in (14.49) denotes that all components in $\boldsymbol{E}$ contained in $\mathcal{S}^{\perp}$ are removed.

We then proceed with a singular value decomposition

$$\boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^{\mathrm{T}} = \boldsymbol{X}_0, \tag{14.52}$$

where all matrices are $N \times N$, and insert this in (14.50) to get

$$\begin{aligned}\boldsymbol{C} &\approx \boldsymbol{U}_0\boldsymbol{\Sigma}_0(\boldsymbol{I} + \boldsymbol{U}_1\boldsymbol{\Sigma}_1^2\boldsymbol{U}_1^{\mathrm{T}})\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}} \\ &= \boldsymbol{U}_0\boldsymbol{\Sigma}_0\boldsymbol{U}_1(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2)\boldsymbol{U}_1^{\mathrm{T}}\boldsymbol{\Sigma}_0^{\mathrm{T}}\boldsymbol{U}_0^{\mathrm{T}}.\end{aligned} \tag{14.53}$$

Now the pseudo inverse of $\boldsymbol{C}$ becomes

$$\begin{aligned}\boldsymbol{C}^{+} &\approx (\boldsymbol{U}_0\boldsymbol{\Sigma}_0^{+\mathrm{T}}\boldsymbol{U}_1)(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2)^{-1}(\boldsymbol{U}_0\boldsymbol{\Sigma}_0^{+\mathrm{T}}\boldsymbol{U}_1)^{\mathrm{T}} \\ &= \boldsymbol{X}_1(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2)^{-1}\boldsymbol{X}_1^{\mathrm{T}},\end{aligned} \tag{14.54}$$

where we have defined $\boldsymbol{X}_1 \in \Re^{m\times N}$ of rank $N-1$ as

$$\boldsymbol{X}_1 = \boldsymbol{U}_0\boldsymbol{\Sigma}_0^{+\mathrm{T}}\boldsymbol{U}_1. \tag{14.55}$$

14.3.2 Analysis schemes using a low-rank $\boldsymbol{C}_{\epsilon\epsilon}$

By replacing $\boldsymbol{C}^{-1}$ in (14.2) with the pseudo inverse $\boldsymbol{C}^{+}$, from (14.54), we can easily compute the EnKF analysis using the subspace pseudo inversion by carrying out the matrix multiplications in

$$\boxed{\begin{array}{l}\text{EnKF subspace analysis with low-rank } \boldsymbol{C}_{\epsilon\epsilon} \\ \boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\left(\boldsymbol{I} + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{X}_1\big(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2\big)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\big(\boldsymbol{D} - \mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\right).\end{array}} \tag{14.56}$$

Similarly the square root algorithm uses (14.3) with $\boldsymbol{C}^{-1}$ replaced by $\boldsymbol{C}^{+}$, from (14.54),

$$\overline{\boldsymbol{A}}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\left(\boldsymbol{1}_N + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{X}_1\big(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2\big)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\big(\overline{\boldsymbol{D}} - \mathcal{M}\big[\overline{\boldsymbol{A}}^{\mathrm{f}}\big]\big)\right), \tag{14.57}$$

to compute the updated ensemble mean.

Using the expression (14.54) for the inverse in (14.5) we get the following derivation of the perturbation updates in the square root analysis scheme,

$$\begin{aligned}\boldsymbol{A}^{\mathrm{a}\prime}\boldsymbol{A}^{\mathrm{a}\prime\mathrm{T}} &= \boldsymbol{A}'\left(\boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}}\boldsymbol{C}^{+}\boldsymbol{S}\right)\boldsymbol{A}'^{\mathrm{T}} \\ &= \boldsymbol{A}'\left(\boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}}\boldsymbol{X}_1\big(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2\big)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{S}\right)\boldsymbol{A}'^{\mathrm{T}} \\ &= \boldsymbol{A}'\Big(\boldsymbol{I} - \Big(\big(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2\big)^{-\frac{1}{2}}\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{S}\Big)^{\mathrm{T}} \\ &\qquad \Big(\big(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2\big)^{-\frac{1}{2}}\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{S}\Big)\Big)\boldsymbol{A}'^{\mathrm{T}} \\ &= \boldsymbol{A}'\left(\boldsymbol{I} - \boldsymbol{X}_2^{\mathrm{T}}\boldsymbol{X}_2\right)\boldsymbol{A}'^{\mathrm{T}},\end{aligned} \tag{14.58}$$

where we have defined $\boldsymbol{X}_2$ as

$$\boldsymbol{X}_2 = \big(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2\big)^{-\frac{1}{2}}\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{S} = \big(\boldsymbol{I} + \boldsymbol{\Sigma}_1^2\big)^{-\frac{1}{2}}\boldsymbol{U}_1^{\mathrm{T}}\tilde{\boldsymbol{I}}_p\boldsymbol{V}_0^{\mathrm{T}}. \tag{14.59}$$

We then end up with the same final update equation (14.4) by following the derivation defined in (13.6–13.7).

Thus, we have replaced the explicit factorization of $\boldsymbol{C} \in \Re^{m\times m}$, with an SVD of $\boldsymbol{S} \in \Re^{m\times N}$, and this is a significant saving when $m \gg N$. Further, by using a low-rank version for $\boldsymbol{C}_{\epsilon\epsilon}$ we replace the matrix multiplication $\boldsymbol{\Sigma}_0^{+}\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}$ in (14.23) with the less expensive $\boldsymbol{\Sigma}_0^{+}\boldsymbol{U}_0^{\mathrm{T}}\boldsymbol{E}$. Thus, there are none matrix operations that requires $\mathcal{O}(m^2)$ floating point operations in the new algorithm.

Equations (14.57) and (14.4) can be combined into one single equation, similar to (14.35), as

SQRT subspace analysis with low-rank $\boldsymbol{C}_{\epsilon\epsilon}$

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}}\Big(\mathbf{1}_N + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{X}_1(\boldsymbol{I}+\boldsymbol{\Sigma}_1^2)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\big(\boldsymbol{D}-\mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\mathbf{1}_N + (\boldsymbol{I}-\mathbf{1}_N)\boldsymbol{V}_2\sqrt{\boldsymbol{I}-\boldsymbol{\Sigma}_2^{\mathrm{T}}\boldsymbol{\Sigma}_2}\boldsymbol{V}_2^{\mathrm{T}}\boldsymbol{\Theta}^{\mathrm{T}}\Big). \tag{14.60}$$

Note that if we set $\boldsymbol{\Lambda}_1 = \boldsymbol{\Sigma}_1^2$ in (14.56) and (14.60) these equations becomes identical to respectively (14.31) and (14.35). Similarly, by replacing the expressions $\boldsymbol{X}_1(\boldsymbol{I}+\boldsymbol{\Sigma}_1^2)^{-1}\boldsymbol{X}_1^{\mathrm{T}}$ and $\boldsymbol{X}_1(\boldsymbol{I}+\boldsymbol{\Lambda}_1)^{-1}\boldsymbol{X}_1^{\mathrm{T}}$ in these equations, with $\boldsymbol{Z}\boldsymbol{\Lambda}^{+}\boldsymbol{Z}^{\mathrm{T}}$ or $\boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^{\mathrm{T}}$ they become identical to the analysis equations (14.2) and (14.6).

14.4 Implementation of the analysis schemes

For the practical implementation we first note that we can choose from three different algorithms when computing the pseudo inverse of $\boldsymbol{C}$. We can use a standard pseudo inversion based on an eigenvalue decomposition of $\boldsymbol{C}$, or we can use the subspace pseudo inversion with either a full measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$, or with a low-rank representation of the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}^{\mathrm{e}} = \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}/(N-1)$. From the standard eigenvalue factorization we obtain $\boldsymbol{Z}$ and $\boldsymbol{\Lambda}$. For the two subspace algorithms we obtain $\boldsymbol{X}_1$ and either $(\boldsymbol{I}+\boldsymbol{\Sigma}_1^2)$ or $(\boldsymbol{I}+\boldsymbol{\Lambda}_1^2)$.

Thereafter, we can choose between the computation of a traditional EnKF analysis or a square root analysis. Each of these schemes requires the evaluation of the matrix multiplied with $\boldsymbol{A}$ in one of (14.2), (14.31) or (14.56) for the EnKF and one of (14.6), (14.35) or (14.60) for the square root algorithm. The final multiplication with $\boldsymbol{A}$ to compute the updated ensemble is the same for all of the algorithms.

Thus, it is clear that it is possible to combine all of these algorithms into one efficient routine where the user can choose between different pseudo inversions and analysis schemes. In this routine one should also include specific code for handling the case with a single observation where a scalar inverse can be used. Note also that in the EnKF with few observations, it is more efficient to reorder the matrix multiplications and rewrite (14.2) as

$$\boldsymbol{A}^{\mathrm{a}} = \boldsymbol{A}^{\mathrm{f}} + \Big(\boldsymbol{A}^{\mathrm{f}}\boldsymbol{S}^{\mathrm{T}}\Big)\Big(\boldsymbol{C}^{-1}\big(\boldsymbol{D}-\mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)\Big). \tag{14.61}$$

The standard analysis scheme needs to compute a matrix multiplication for the final update which requires nN^2 floating point operations. When $n > m$ this becomes the most expensive computation in the analysis scheme. Note also that, in the standard scheme, mN^2 operations are required when $\boldsymbol{S}^{\mathrm{T}}$ is multiplied with the $m \times N$ matrix $\boldsymbol{C}^{-1}\big(\boldsymbol{D}-\mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)$.

However, with few observations it is more efficient to first compute the product $\boldsymbol{A}^{\mathrm{f}}\boldsymbol{S}^{\mathrm{T}}$, which requires nmN floating point operations. The additional multiplication with the matrix $\boldsymbol{C}^{-1}\big(\boldsymbol{D}-\mathcal{M}\big[\boldsymbol{A}^{\mathrm{f}}\big]\big)$ requires another nmN operations. Thus, when $2nmN < (n+m)N^2$ this procedure is more efficient. For the assimilation of a single observation this reduces the computation by a factor $N/2$.

14.5 Rank issues related to the use of a low-rank $\boldsymbol{C}_{\epsilon\epsilon}$

It has recently been shown by *Kepert* (2004) that the use of an ensemble representation $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ for $\boldsymbol{C}_{\epsilon\epsilon}$, in some cases leads to a loss of rank in the ensemble when $m > N$. The rank problem may occur both using the EnKF analysis scheme with perturbation of measurements and using the square root algorithm. However, it is not obvious that the case with $m > N$ and the use of a low-rank representation $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ of $\boldsymbol{C}_{\epsilon\epsilon}$, should pose a problem. After all, the final coefficient matrix which is multiplied with the ensemble forecast to produce the analysis, is an $N \times N$ matrix.

The following will revisit the analysis by *Kepert* (2004) and extend it to a more general situation. Further, it will be shown that the rank problem can be avoided when the measurement perturbations, used to represent the low-rank measurement error covariance matrix, are sampled under specific constraints.

The EnKF analysis equation (14.2) can be rewritten as

$$\begin{aligned}\boldsymbol{A} = \overline{\boldsymbol{A}} &+ \boldsymbol{A}'\boldsymbol{S}^{\mathrm{T}}\big(\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}+\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}\big)^{+}\big(\overline{\boldsymbol{D}}-\mathcal{M}\big[\overline{\boldsymbol{A}}^{\mathrm{f}}\big]\big)\\ &+ \boldsymbol{A}' + \boldsymbol{A}'\boldsymbol{S}^{\mathrm{T}}\big(\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}+\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}\big)^{+}(\boldsymbol{E}-\boldsymbol{S}),\end{aligned} \tag{14.62}$$

where the first line is the update of the mean and the second line is the update of the ensemble perturbations. Thus, for the standard EnKF is suffices to show that $\mathrm{rank}(\boldsymbol{W}) = N-1$ to conserve the full rank of the state ensemble, with $\boldsymbol{W}$ defined as

$$\boldsymbol{W} = \boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}}\big(\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}+\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}\big)^{+}(\boldsymbol{S}-\boldsymbol{E}). \tag{14.63}$$

Similarly, for the square root algorithm $\boldsymbol{W}$ is redefined from (14.5) as

$$\boldsymbol{W} = \boldsymbol{I} - \boldsymbol{S}^{\mathrm{T}}\big(\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}+\boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}\big)^{+}\boldsymbol{S}. \tag{14.64}$$

We consider the case where $m > N-1$ that is shown to cause problems in *Kepert* (2004). Define $\boldsymbol{S} \in \Re^{m\times N}$ with $\mathrm{rank}(\boldsymbol{S}) = N-1$, where the columns of $\boldsymbol{S}$ span a subspace $\mathcal{S}$ of dimension $N-1$. Further, we define $\boldsymbol{E} \in \Re^{m\times q}$ with $\mathrm{rank}(\boldsymbol{E}) = \min(m, q-1)$, where $\boldsymbol{E}$ contains an arbitrary number q, of measurement perturbations.

As in *Kepert* (2004) one can define the matrix $\boldsymbol{Y} \in \Re^{m\times(N+q)}$ as

$$\boldsymbol{Y} = (\boldsymbol{S}, \boldsymbol{E}), \tag{14.65}$$

and the matrix $\boldsymbol{C}$ becomes

$$\boldsymbol{C} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}, \tag{14.66}$$

with rank

$$p = \mathrm{rank}(\boldsymbol{Y}) = \mathrm{rank}(\boldsymbol{C}). \tag{14.67}$$

Dependent on the definition of $\boldsymbol{E}$ we have $\min(m, N-1) \leq p \leq \min(m, N+q-2)$. One extreme is the case where $q \leq N$ and $\boldsymbol{E}$ is fully contained in $\mathcal{S}$, in which case we have $p = N-1$.

The case considered in *Kepert* (2004) is another extreme which has $q = N$, and $p = \min(m, 2N-2)$. This corresponds to a situation which is likely to occur when $\boldsymbol{E}$ is sampled randomly and includes components along $N-1$ directions in $\mathcal{S}^{\perp}$.

We define the SVD of $\boldsymbol{Y}$ as

$$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{Y}, \tag{14.68}$$

with $\boldsymbol{U} \in \Re^{m\times m}$, $\boldsymbol{\Sigma} \in \Re^{m\times(N+q)}$ and $\boldsymbol{V} \in \Re^{(N+q)\times(N+q)}$.

The pseudo inverse of $\boldsymbol{Y}$ is defined as

$$\boldsymbol{Y}^{+} = \boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^{\mathrm{T}}, \tag{14.69}$$

where $\boldsymbol{\Sigma}^{+} \in \Re^{(N+q)\times m}$ is a diagonal matrix with the diagonal defined as $\mathrm{diag}(\boldsymbol{\Sigma}^{+}) = (\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_p^{-1}, 0, \ldots, 0)$.

Both the equations for $\boldsymbol{W}$ in (14.63) and (14.64) can be rewritten in a form similar to what was used by *Kepert* (2004). Introducing the expressions (14.68) and (14.69) in (14.64), and defining $\boldsymbol{I}_N$ to be the N-dimensional identity matrix, we get

$$\begin{aligned}
\boldsymbol{W} &= \boldsymbol{I}_N - (\boldsymbol{I}_N, \boldsymbol{0})\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}})^{+}\boldsymbol{Y}(\boldsymbol{I}_N, \boldsymbol{0})^{\mathrm{T}} \\
&= \boldsymbol{I}_N - (\boldsymbol{I}_N, \boldsymbol{0})\boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}^{+\mathrm{T}}\boldsymbol{\Sigma}^{+}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{I}_N, \boldsymbol{0})^{\mathrm{T}} \\
&= (\boldsymbol{I}_N, \boldsymbol{0})\boldsymbol{V}\left\{\boldsymbol{I}_{N+q} - \begin{pmatrix} \boldsymbol{I}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}_{N+q}\right\}\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{I}_N, \boldsymbol{0})^{\mathrm{T}} \\
&= (\boldsymbol{I}_N, \boldsymbol{0})\boldsymbol{V}\begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{N+q-p} \end{pmatrix}_{N+q}\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{I}_N, \boldsymbol{0})^{\mathrm{T}}.
\end{aligned} \tag{14.70}$$

The similar expression for $\boldsymbol{W}$ in (14.63) is obtained by replacing the matrix, $(\boldsymbol{I}_N, \boldsymbol{0}) \in \Re^{N\times(N+q)}$, with $(\boldsymbol{I}_N, -\boldsymbol{I}_N, \boldsymbol{0}) \in \Re^{N\times(N+q)}$.

We need the $N+q$ matrix in (14.70) to have a rank of at least $N-1$ to maintain the rank of the updated ensemble perturbations. Thus, we require that $N+q-p \geq N-1$ and get the general condition

$$p \leq q+1. \tag{14.71}$$

With $q = N$ this condition requires $p \leq N+1$. This is only possible when all singular vectors of $\boldsymbol{E}$, except two, are contained in $\mathcal{S}$. Thus, it is clear that

a low-rank representation of $\boldsymbol{C}_{\epsilon\epsilon}$ using N measurement perturbations $\boldsymbol{E}$, can be used as long as the selected perturbations do not increase the rank of $\boldsymbol{Y}$ to more than $N+1$.

It is also clear that if the constrained low-rank representation $\boldsymbol{E} \in \Re^{m\times N}$, is unable to properly represent the real measurement error covariance, it is possible to increase the number of perturbations to an arbitrary number $q > N$ as long as the rank p satisfies the condition (14.71).

In *Kepert* (2004) it was assumed that the rank $p = 2N-2$. That is, $\boldsymbol{E}$ has components in $N-1$ directions of $\mathcal{S}^{\perp}$. Then, clearly, the condition (14.71) is violated and this results in a loss of rank. It was shown that this problem can be resolved using a full rank measurement error covariance matrix (corresponding to the limiting case when $q \geq m+1$). Then, $p = \text{rank}(\boldsymbol{Y}) = \text{rank}(\boldsymbol{C}^{\text{e}}_{\epsilon\epsilon}) = m$ and the condition (14.71) is always satisfied.

As an example, assume now that we have removed r columns from the matrix $\boldsymbol{E} \in \Re^{m\times(q=m+1)}$. We then get the reduced $\boldsymbol{E} \in \Re^{m\times(q=m+1-r)}$ of rank equal to $m-r$. In this situation we can consider two cases. First, if the removed perturbations are also fully contained in $\mathcal{S}$, then the removal does not lead to a reduction of p which still equals m. In this case we can write the condition (14.71), for $r \leq N-1$, as

$$p = m \leq m+2-r, \tag{14.72}$$

which is violated for $r > 2$. Secondly, assume that the removed perturbations are fully contained in $\mathcal{S}^{\perp}$. Then the rank p will be reduced with r and we write the condition (14.71) as

$$p = m-r \leq m+2-r. \tag{14.73}$$

We can continue to remove columns of $\boldsymbol{E}$ contained in $\mathcal{S}^{\perp}$, without violating the condition (14.71), until there are only $N-1$ columns left in $\boldsymbol{E}$, all contained in $\mathcal{S}$.

From this discussion, it is clear that we need the measurement error perturbations to explain variance within $\mathcal{S}$. Note that the subspace pseudo inversion schemes automatically projects the measurement error covariance matrix or the measurement perturbations onto $\mathcal{S}$.

14.6 Experiments with $m \gg N$

The following experiments are performed to evaluate the properties of the analysis schemes in the case where $m \gg N$. An experimental setup, similar to the advection example from Sect. 4.1.3, is used. However, now 500 measurements are assimilated in each update step. Thus, there is a measurement at every second grid point. The measurements have correlated errors of decorrelation length equal to 20 m. The error variance of the measurements is

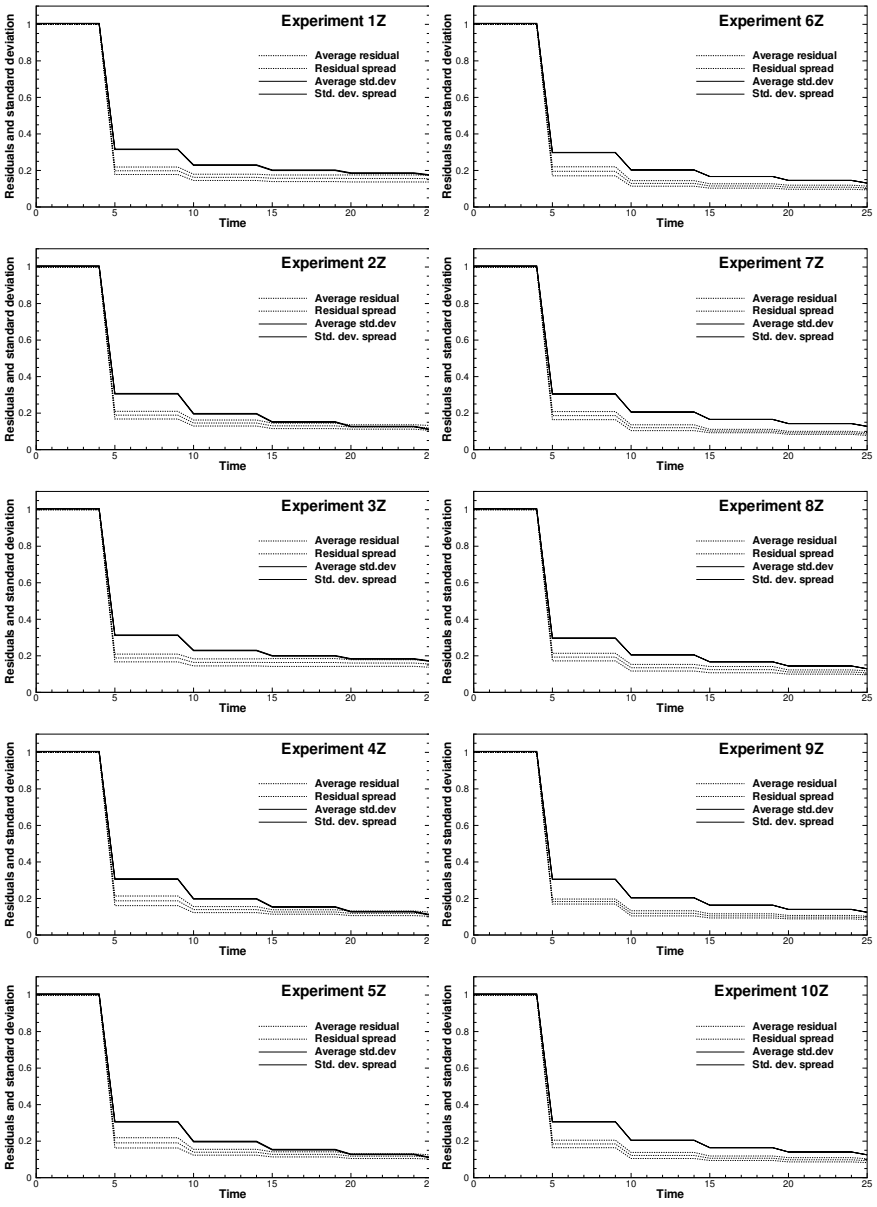

Fig. 14.6. Time evolution for RMS residuals *(dotted lines)* and estimated standard deviations *(full lines)* for all 50 simulations in the respective experiments

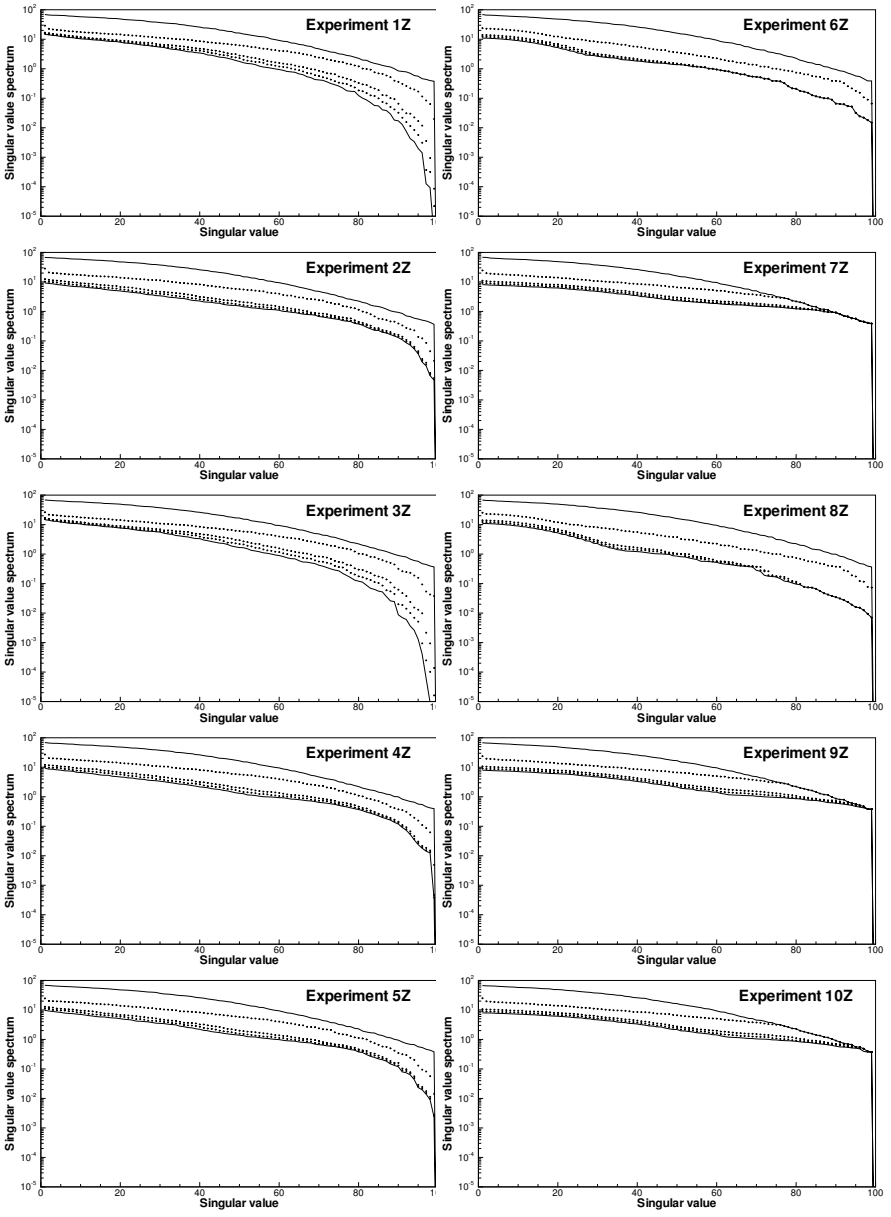

Fig. 14.7. Time evolution of the ensemble singular value spectra for some of the experiments

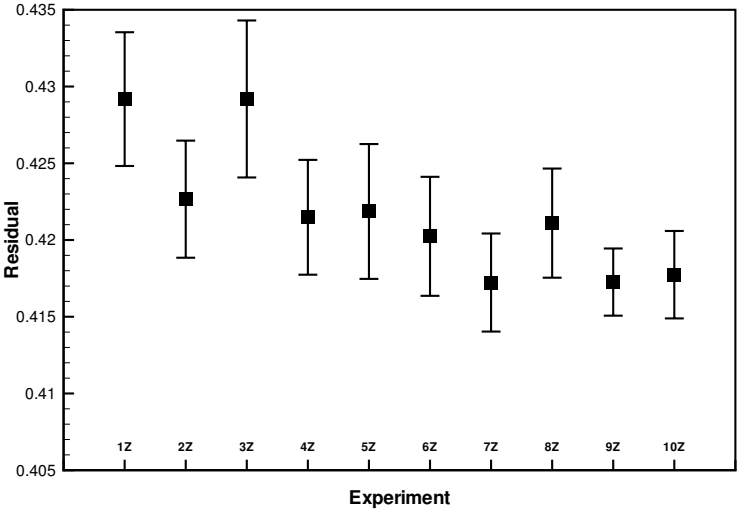

Fig. 14.8. Average residual and standard deviation for the 10 cases

set to 0.09, corresponding to a standard deviation of 0.30 and the number of assimilation steps is 5.

Ten experiments, which differ in the choice of analysis scheme (EnKF or SQRT) and inversion algorithm for $\boldsymbol{C}$, are run. In addition, both an exact and a low-rank representation of $\boldsymbol{C}_{\epsilon\epsilon}$ are used. The experiments are summarized in Table 14.1 where EnKF and SQRT denote the analysis scheme used. EIGC denote the inversion algorithm based on the eigenvalue factorization from Sect. 14.1, SUBC denote the subspace inversion discussed in Sect. 14.2, and SUBE means the subspace inversion using the measurement perturbations rather than the full measurement error covariance matrix, as presented in Sect. 14.3. In the different experiments we have specified either a full rank measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$, or a low-rank version defined as $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon} = \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}/(N-1)$.

It is straight-forward to sample normal correlated perturbations for each element of $\boldsymbol{E}$ with the correct statistics. This sampling is performed by using the same sampling scheme as is used to generate the initial ensemble and then measuring each member to create the columns in $\boldsymbol{E}$. In all the experiments we use improved sampling of order six for the initial ensemble and order four for the measurement perturbations.

Note that $\boldsymbol{E}$ is sampled with rank equal to $N-1$. When projected onto $\boldsymbol{U}_{0p}$, i.e. the sub-space $\mathcal{S}$ spanned by the first p singular vectors in $\boldsymbol{U}_0$, we are not guaranteed that the rank of $\boldsymbol{U}^{\mathrm{T}}_{0p}\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\boldsymbol{U}_{0p}$ or $\boldsymbol{U}_{0p}\boldsymbol{E}$ is equal to $N-1$. If $\boldsymbol{E}$ has columns which are orthogonal to $\boldsymbol{U}_{0p}$, these do not contribute when projected onto $\boldsymbol{U}_{0p}$. This corresponds to the assimilation of perfect measurements and

Exp. 1Z	EnKF	EIGC	$\boldsymbol{C}_{\epsilon\epsilon}$	*Exp. 6Z*	SQRT	EIGC	$\boldsymbol{C}_{\epsilon\epsilon}$
Exp. 2Z	EnKF	EIGC	$\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$	*Exp. 7Z*	SQRT	EIGC	$\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$
Exp. 3Z	EnKF	SUBC	$\boldsymbol{C}_{\epsilon\epsilon}$	*Exp. 8Z*	SQRT	SUBC	$\boldsymbol{C}_{\epsilon\epsilon}$
Exp. 4Z	EnKF	SUBC	$\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$	*Exp. 9Z*	SQRT	SUBC	$\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$
Exp. 5Z	EnKF	SUBE	$\boldsymbol{E}$	*Exp. 10Z*	SQRT	SUBE	$\boldsymbol{E}$

Table 14.1. List of experiments. See explanation in text

will lead to a corresponding loss of rank in the updated ensemble. We did not experience this to be a problem in the present experiments.

The use of a low-rank representation for $\boldsymbol{C}_{\epsilon\epsilon}$ is valid, and if $\boldsymbol{U}_{0p}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\boldsymbol{U}_{0p} = \boldsymbol{U}_{0p}^{\mathrm{T}}\boldsymbol{C}_{\epsilon\epsilon}\boldsymbol{U}_{0p}$, the results will be the same as the results obtained using a full rank $\boldsymbol{C}_{\epsilon\epsilon}$. This equality is nearly satisfied here since the random sampling of $\boldsymbol{E}$ used the same correlation functions as was used to generate the initial ensemble. Probably, in this case, the use of a diagonal error covariance matrix would be more difficult to represent properly by a low-rank random ensemble of smooth members.

It is also clear that the projection of $\boldsymbol{C}_{\epsilon\epsilon}$ onto the $\mathcal{S}$-space may lead to a lower measurement variance than specified in the full rank $\boldsymbol{C}_{\epsilon\epsilon}$, thus there may be a need to rescale $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ to avoid over-fitting the data, in which case the EnKF will predict too low estimated standard deviations.

As before we have run 50 assimilation simulations for each experiment to be able to give a statistical comparison of results between the different experiments. The time evolution of the residuals and singular spectra are presented in Figs. 14.6 and 14.7. It is clear that the residuals are rather similar for all the experiments, which appear to provide consistent solutions.

In Fig. 14.8 we have plotted the mean and standard deviation of the residuals as predicted by the 50 assimilation simulations in each experiment. We observe that the square root experiments have lower residuals than the standard EnKF experiments. In addition, the exeperiments using the exact measurement error covariance matrix, i.e. *Exps. 1, 3, 5* and *7* have poorer performence than the corresponding experiments where the measurement perturbations are used to represent the error covariance. This result may be linked to the discussion on the use of an ensemble based measurement error covariance matrix in the derivation of the update equations for the EnKF in Section 14.7.

The two EnKF *Exps. 1* and *2* provide statistically similar results as do the three EnKF *Exps. 2, 4* and *5*. Similarly the two square root *Exps. 6* and *8* are statistically indistinguisable as are the three square root *Exps. 7, 9* and *10*. Thus, the different inversion schemes do not seem to influence the results and may be used independently.

The experiments using the SQRT scheme seem to do a slightly better job than those using the EnKF in this experiment. All experiments were rerun starting from different random seeds and this confirmed the results.

Exp	*2Z*	*3Z*	*4Z*	*5Z*	*6Z*	*7Z*	*8Z*	*9Z*	*10Z*
1Z	0.96	0.27	0.71	0.35	0	0.02	0	0	0
2Z		0.23	0.72	0.31	0	0.01	0	0	0
3Z			0.50	0.85	0	0.10	0	0	0
4Z				0.61	0	0.04	0	0	0
5Z					0	0.07	0	0	0
6Z						0	0.57	0.10	0.11
7Z							0	0	0
8Z								0.02	0.02
9Z									0.78

Table 14.2. Statistical probability that two experiments provide an equal mean for the residuals as computed using the Student's t-test. A probability close to one indicates that it is likely that the two experiments provide distributions of residuals with similar mean

From the previous theoretical analysis, the new low-rank square root scheme introduces an approximation by projecting the measurements onto the $\mathcal{S}$ sub-space, and it was seen that this approximation both stabilises the computation of the analysis and also makes it computationally more efficient. However, when a low-rank $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ is used, a scheme is required for the proper sampling of measurement perturbations in $\mathcal{S}$.

14.7 Validity of analysis equation

The analysis scheme for the EnKF is derived in Sect. 4.3, and we find the well known result for the analyzed error covariance matrix in Eq. (4.41), i.e.,

$$\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{a}} = \left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}. \tag{14.74}$$

The current equation is derived under the assumption of an infinite ensemble of realizations, and zero correlation between the measurement perturbations and ensemble of model anomalies. When a finite ensemble is used, an additional correction term that represents the cross correlations between measurement perturbations and the model anomalies arise in the equation. In addition, it is seen below that an additional error term is introduced if an exact measurement error covariance matrix is used in the Kalman gain.

We now define a Kalman gain that is based on a measurement error covariance matrix $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ represented in terms of an ensemble of measurement perturbations

$$\boldsymbol{K}_{\mathrm{e}} = \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}}\Big(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\Big)^{-1}. \tag{14.75}$$

The error covariance update is then derived as

$$
\begin{aligned}
\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{a}} &= \overline{\left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{a}}}\right)\left(\boldsymbol{\psi}^{\mathrm{a}} - \overline{\boldsymbol{\psi}^{\mathrm{a}}}\right)^{\mathrm{T}}} \\
&= \overline{\Big(\big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\big(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\big) + \boldsymbol{K}_{\mathrm{e}}\big(\boldsymbol{d} - \overline{\boldsymbol{d}}\big)\Big)} \\
&\quad \overline{\Big(\big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\big(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\big) + \boldsymbol{K}_{\mathrm{e}}\big(\boldsymbol{d} - \overline{\boldsymbol{d}}\big)\Big)^{\mathrm{T}}} \\
&= \big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\overline{\big(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\big)\big(\boldsymbol{\psi}^{\mathrm{f}} - \overline{\boldsymbol{\psi}^{\mathrm{f}}}\big)^{\mathrm{T}}}\big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)^{\mathrm{T}} \\
&\qquad + \boldsymbol{K}_{\mathrm{e}}\overline{\big(\boldsymbol{d} - \overline{\boldsymbol{d}}\big)\big(\boldsymbol{d} - \overline{\boldsymbol{d}}\big)^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&\qquad + 2\big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\overline{\big(\boldsymbol{\psi} - \overline{\boldsymbol{\psi}}\big)\big(\boldsymbol{d} - \overline{\boldsymbol{d}}\big)^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&= \big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\big(\boldsymbol{I} - \boldsymbol{M}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}}\big) + \boldsymbol{K}_{\mathrm{e}}\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&\qquad + 2\big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\boldsymbol{C}_{\psi\epsilon}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&= \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} - \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&\qquad + \boldsymbol{K}_{\mathrm{e}}\big(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\big)\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&\qquad + 2\big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\boldsymbol{C}_{\psi\epsilon}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\right)\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \\
&\qquad + 2\big(\boldsymbol{I} - \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\big)\boldsymbol{C}_{\psi\epsilon}\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}}.
\end{aligned}
\tag{14.76}
$$

Thus, (14.76) implies that EnKF in the limit of an infinite ensemble size gives the same result as the KF. It is assumed that the distributions used to generate the model-state ensemble and the observation ensemble are independent. Using a finite ensemble size, neglecting the cross-term introduces sampling errors.

As previously pointed out in Chap. 4, the derivation (14.76) shows that the observations $\boldsymbol{d}$ must be treated as random variables to introduce the measurement error covariance matrix $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ into the expression. That is,

$$
\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon} = \overline{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}} = \overline{\big(\boldsymbol{d} - \overline{\boldsymbol{d}}\big)\big(\boldsymbol{d} - \overline{\boldsymbol{d}}\big)^{\mathrm{T}}}. \tag{14.77}
$$

Note that the use of an ensemble representation of the measurement error covariance matrix leads to an exact cancellation in the second last line in (14.76), since we can write

$$
\begin{aligned}
&\boldsymbol{K}_{\mathrm{e}}\Big(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\Big)\boldsymbol{K}_{\mathrm{e}}^{\mathrm{T}} \\
&\quad = \boldsymbol{K}_{\mathrm{e}}\Big(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\Big)\Big(\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}\boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\Big)^{-1}\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \\
&\quad = \boldsymbol{K}_{\mathrm{e}}\boldsymbol{M}\left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}}.
\end{aligned}
\tag{14.78}
$$

If a full-rank measurement error covariance matrix is used in the Kalman Gain (14.75), then (14.78) is only approximately true with a finite ensemble size and gives rise to an additional error term.

Thus, we conclude that the use of a low-rank measurement error covariance matrix, represented by the measurement perturbations, when computing the Kalman gain, reduces the sampling errors in EnKF. The remaining sampling errors come from neglecting the cross-correlation term between the measurements and the forecast ensemble, which is nonzero with a final ensemble size, and from the approximation of the state error covariance matrix using a finite ensemble size.

The above derivation assumes that the inverse in the Kalman gain (14.75) exists. However, the derivation also holds when the matrix in the inversion is of low rank, for example, when the number of measurements is larger than the number of realizations and the low-rank $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$ is used. The inverse in (14.75) can then be replaced with the pseudoinverse, and we can write the Kalman gain as

$$\boldsymbol{K}_{\mathrm{e}} = \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} \left(\boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right)^{+}. \tag{14.79}$$

When the matrix in the inversion is of full rank, (14.79) becomes identical to (14.75). Using (14.79) the expression (14.78) becomes

$$\begin{aligned}
&\boldsymbol{K}_{\mathrm{e}} \left(\boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right) \boldsymbol{K}^{\mathrm{T}}_{\mathrm{e}} \\
&\quad = \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} \left(\boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right)^{+} \left(\boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right) \\
&\qquad\qquad \left(\boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right)^{+} \boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \\
&\quad = \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} \left(\boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \boldsymbol{M}^{\mathrm{T}} + \boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}\right)^{+} \boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}} \\
&\quad = \boldsymbol{K}_{\mathrm{e}} \boldsymbol{M} \left(\boldsymbol{C}^{\mathrm{e}}_{\psi\psi}\right)^{\mathrm{f}},
\end{aligned} \tag{14.80}$$

where we have used the property $\boldsymbol{Y}^{+} = \boldsymbol{Y}^{+}\boldsymbol{Y}\boldsymbol{Y}^{+}$ of the pseudoinverse.

It should be noted that the EnKF analysis scheme is approximate in the sense that non-Gaussian contributions in the predicted ensemble are not properly taken into account. In other words, the EnKF analysis scheme does not solve the Bayesian update equation for non-Gaussian pdfs. On the other hand, the EnKF analysis scheme is not just a re-sampling of a Gaussian posterior distribution. Only the updates defined by the right hand side of (4.37), which are added to the prior non-Gaussian ensemble, are linear. Thus, the updated ensemble inherits many of the non-Gaussian properties from the forecast ensemble. In summary, we have a computationally efficient analysis scheme where we avoid re-sampling of the posterior.

14.8 Summary

A comprehensive analysis is given on the use of the EnKF and square root analysis schemes when used with large data sets. It is seen that the inver-

sion of $\boldsymbol{C}$ may become poorly conditioned, and a pseudo inversion may be required. The analysis schemes are reformulated using a standard pseudo inversion based on an eigenvalue factorization of $\boldsymbol{C}$ followed by a truncation of the eigenvalue spectrum to only account for the significant eigenvalues. This algorithm seems to work well in many cases. However, when the number of measurements becomes large it is inefficient, since a matrix of dimension $m \times m$ needs to be factorized at a cost proportional to $\mathcal{O}(m^3)$.

An alternative pseudo inversion is derived where the measurements are projected onto a sub-space $\mathcal{S}$, spanned by the measured ensemble perturbations. It is seen that approach may introduce an approximation in some cases. In particular, if the measurement error covariance matrix is diagonal then the eigenvectors of $\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$ and $\boldsymbol{C}$ are identical and there is no approximation introduced. On the other hand, if $\boldsymbol{C}_{\epsilon\epsilon}$ is nondiagonal the eigenvectors will differ and the projection onto the $\mathcal{S}$-space eliminates the part of $\boldsymbol{C}$ that is orthogonal to the $\mathcal{S}$-space. Fortunately, this is mostly noise in many applications.

The sub-space pseudo inversion can be computed at a cost of $\mathcal{O}(Nm^2)$ which is a significant saving when $m \gg N$. However, it is also seen that a further speedup is possible if a low-rank representation is used for the measurement error covariance matrix. In particular, if we write the measurement error covariance matrix as $(N-1)\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon} = \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}$, and represent it by the measurement perturbations $\boldsymbol{E}$, it is possible to compute the analysis without forming $\boldsymbol{C}^{\mathrm{e}}_{\epsilon\epsilon}$. This approach further reduces the cost of the inversion to be proportional to $\mathcal{O}(N^2m)$, and the algorithm allows us to compute the analysis update using very large data sets. An important point is that the measurement perturbations must be sampled to span $\mathcal{S}$ to avoid a loss of rank in the updated ensemble.

15

Spurious correlations, localization, and inflation

The use of a finite ensemble size to approximate the error covariance matrix introduces sampling errors that are seen as spurious correlations over long spatial distances or between variables known to be uncorrelated. The spurios correlations imply that variables that are supposed to be uncorrelated with an observation, experience a small unphysical update. Over time and with many data, the spurious updates may cancel out and the drift in the mean may be negligible. However, with each spurious update there is an associated reduction of ensemble variance and over time the ensemble variance may significantly underestimate the true variance. This problem is present in all EnKF applications and can lead to filter divergence. On the other hand, the consistency of the updated variance improves when a larger ensemble is used.

In the following we will first examine and demonstrate the impact of the spurious correlations in a simple example. Thereafter we will look at two approaches for minimizing the impact of the spurious updates, i.e., ensemble inflation and localization.

15.1 Spurious correlations

The following example, which is based on the linear advection case from Fig. 4.2, illustrates the variance reduction resulting from spurious correlations.

The ensemble of model states are stored in $\boldsymbol{A} \in \Re^{n \times N}$. An additional ensemble $\boldsymbol{B} \in \Re^{n_{\text{rand}} \times N}$ is generated, where each row contains random samples from a Gaussian distribution with mean equal to zero and variance equal to one, and the entries in different rows are sampled independently. Thus, $\boldsymbol{B}$ is the ensemble matrix for a state vector of independent variables with zero mean and unit variance. At analysis times we compute the updates

$$\begin{pmatrix} \boldsymbol{A}^{\text{a}} \\ \boldsymbol{B}^{\text{a}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^{\text{f}} \\ \boldsymbol{B}^{\text{f}} \end{pmatrix} \boldsymbol{X}. \tag{15.1}$$

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_15,

The predicted ensemble $\boldsymbol{A}^{\mathrm{f}}$ is the result of the ensemble integration using the advection model, while $\boldsymbol{B}^{\mathrm{f}}$ does not evolve according to any dynamical equation. Thus, at an update time $\boldsymbol{B}^{\mathrm{f}}$ equals $\boldsymbol{B}^{\mathrm{a}}$ from the previous update time. The update matrix $\boldsymbol{X}$ can be defined from any of the analysis equations in Chap. 14.

Since the correlations between $\boldsymbol{B}$ and the predicted measurement perturbations $\boldsymbol{S}$ become zero in the limit of an infinite ensemble size, it follows that

$$\lim_{N\to\infty} \frac{\boldsymbol{B}\boldsymbol{S}^{\mathrm{T}}}{N-1} = \boldsymbol{0}. \tag{15.2}$$

However, due to the finite ensemble size, (15.2) cannot be exactly satisfied, and $\boldsymbol{B}^{\mathrm{a}}$ experiences a small update and associated reduction of variance through the update in (15.1).

As in the advection example, at every analysis step we compute the matrix $\boldsymbol{X}$ based on the four measurements, and then apply it to $\boldsymbol{B}$ according to (15.1).

The variance reduction resulting from the spurious correlations is illustrated in Fig. 15.1. This plot shows the decrease of the average variance of the random ensemble $\boldsymbol{B}$, resulting from EnKF with 100 and 250 realizations, and from the symmetric square root scheme using 100 realizations. The value $n_{\mathrm{rand}} = 100$ is found to be sufficient, when using 100 realizations, to obtain a consistent result that is independent of the random sampling of $\boldsymbol{B}$.

EnKF with 100 realizations is repeated 5 more times using different random seeds to verify that the result is independent of the seed. A nearly linear decrease of variance is obtained during the first 50 updates, while for the final 12 updates the decrease is lower. The reason for the lower error variance reduction in the final part of the experiment is that the information assimilated at one measurement location propagates to the next measurement location during 50 updates. Thus, after 50 updates the ensemble variance is lower at the measurement locations, and the relative weight on the data compared to the prediction is decreased. EnKF with 250 realizations experiences a significantly lower impact from spurious correlations, as expected.

The square root scheme is slightly less influenced by the spurious correlations, and an explanation can be that the measurement perturbations in the EnKF update increases the strength of the update of individual realizations and thus amplifies the impact of the spurious correlations.

In many dynamical systems, the variance decrease caused by spurious correlations may be masked by strong dynamical instabilities. The impact of the spurious correlations may then be less significant. On the other hand, in parameter-estimation problems, the spurious correlations clearly lead to an underestimate of the ensemble variance of the parameters.

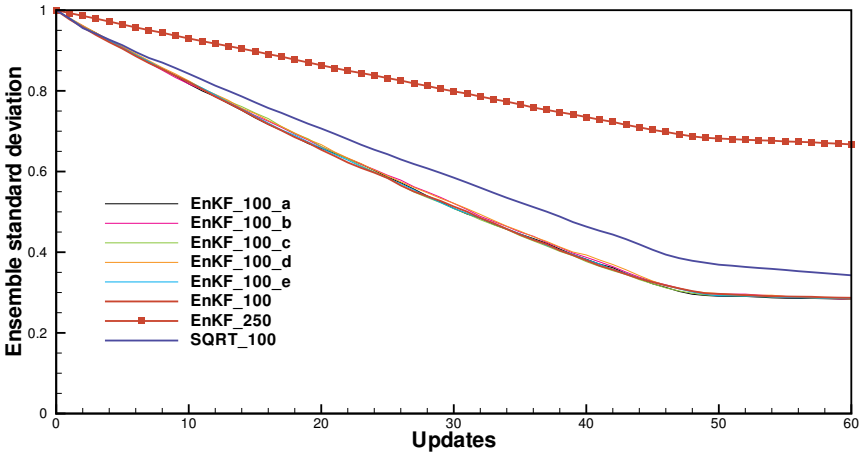

Fig. 15.1. Variance reduction of a random ensemble due to spurious correlations, as a function of analysis updates. EnKF with 100 realizations is compared with EnKF with 250 realizations as well as the square root scheme using 100 realizations. EnKF with 100 realizations is repeated using different seeds to ensure that the results are consistent.

15.2 Inflation

A covariance inflation procedure (*Anderson and Anderson*, 1999b) can be used to counteract the variance reduction observed due to the impact of spurious correlations, as well as other effects leading to underestimation of the ensemble variance. The impact of ensemble size on noise in distant covariances is examined in *Hamill et al.* (2001), while the impact of using an "inflation factor" as discussed in *Anderson and Anderson* (1999b) is evaluated. The inflation factor is used to replace the forecast ensemble according to

$$\boldsymbol{\psi}_j = \rho(\boldsymbol{\psi}_j - \overline{\boldsymbol{\psi}}) + \overline{\boldsymbol{\psi}}, \tag{15.3}$$

with ρ slightly greater than one (typically 1.01). The inflation procedure is also used in *Pham* (2001), where EnKF is examined in an application with the Lorenz attractor, and results are compared with those obtained from different versions of the singular evolutive extended Kalman (SEEK) filter and a particle filter. In *Pham* (2001), ensembles with very few members are used, which favors methods like the SEEK where the "ensemble" of EOFs is selected to best represent the model attractor.

Several approaches adaptively estimate an optimal inflation parameter. In *Wang and Bishop* (2003) the covariance inflation is estimated based on the sequence of innovation statistics, while in *Anderson* (2007a) a method is presented that is based on augmenting the inflation parameter to the model state where it is updated as a parameter in the EnKF analysis computations.

Online estimation of the inflation parameter is also studied in *Li et al.* (2009) together with the simultaneous estimation of observation errors. It is found that the estimation of inflation alone does not work properly without accurate observation error statistics, and vice versa.

Clearly, the inflation parameter becomes a tuning parameter and optimally it is best estimated adaptively. The need for inflation depends on the use of a local versus global analysis scheme, and the use of a local scheme can to a large extent reduce the need for an additional inflation.

Anderson (2009a) proposes a method for adaptively estimating a spatially and temporally varying inflation parameter using a Bayesian algorithm. The algorithm is recursive and updates the inflation parameter with time. *Sacher and Bartello* (2008) discuss the sampling errors in EnKF and proposes an analytical expression for the optimal covariance inflation method which depends on the Kalman gain, the analyzed variance, and the number of realizations.

15.3 An adaptive covariance inflation method

Here we describe an alternative Monte Carlo approach for estimating the inflation coefficient needed to compensate for the variance reduction resulting from spurious correlations. In the spurious correlation example, as presented in Fig. 15.1, an independent ensemble is used to quantify the variance reduction due to spurious correlations. A simple algorithm for correcting the analyzed ensemble perturbations in each analysis step goes as follows.

At each analysis time we generate the additional ensemble matrix $\boldsymbol{B}^{\mathrm{f}}$ with random normally distributed numbers, such that the mean in each row is exactly zero, and the variance is exactly equal to one. We thus sample the matrix randomly from $\mathcal{N}(0,1)$. Then, for each row, first subtract any nonzero mean, then compute the standard deviation and scale all entries by it. Then, compute the analysis update according to (15.1). For each row in $\boldsymbol{B}^{\mathrm{a}}$, compute the standard deviation. The inflation factor ρ is then defined as one over the average of the standard deviations from each row in $\boldsymbol{B}^{\mathrm{a}}$. The accuracy of the estimated inflation factor depends on the number of realizations used as well as the number of rows in $\boldsymbol{B}$. It is expected that with a low number of realizations additional rows in $\boldsymbol{B}$ might compensate for the sampling errors when computing the inflation factor.

This algorithm provides a good first approximation of the inflation factor needed to counteract variance reduction due to long-range spurious correlations resulting from sample noise. The estimated inflation factor depends on the number of realizations used, the number of measurements, and the strength of the update determined by the innovation vector and both the predicted and measurement error covariance matrices. A question remains, as to whether the inflation is best applied equally for the whole model state, including at the measurement locations.

15.4 Localization

We now discuss the use of localization to reduce spurious correlations. Two classes of localization methods are currently used, namely, covariance localization and local updating.

In *Houtekamer and Mitchell* (2001) the ensemble covariance matrix is multiplied with a specified correlation matrix through a Schur product (entry-wise multiplication). The specified correlation functions are defined with local support and thus effectively truncate the long-range spurious correlations produced by the limited ensemble size. Covariance localization is used in *Bishop et al.* (2001), *Hamill et al.* (2001), *Whitaker and Hamill* (2002), and *Anderson* (2003).

We can assume that only measurements located within a certain distance from a gridpoint impact the analysis in that gridpoint. This assumption allows for an algorithm where the analysis is computed gridpoint by gridpoint, and only a subset of observations, located near the current gridpoint, is used in each local analysis. This approach is used in *Haugen and Evensen* (2002), *Brusdal et al.* (2003), and *Evensen* (2003), and is also the approach used in the local EnKF in *Ott et al.* (2004). In addition to reducing the impact of long-range spurious correlations, the localization methods make it simpler to handle large data sets where the number of measurements is much greater than the number of ensemble realizations.

Another reason for computing the local analysis is the fact that EnKF is computed in a space spanned by the ensemble members. This subspace may be rather small compared to the total dimension of the model state. Computing the analysis gridpoint by gridpoint implies that, for each gridpoint, a small model state is solved for in a relatively large ensemble space. The analysis then results from a different combination of ensemble members for each gridpoint, and the analysis scheme is allowed to reach solutions not originally represented by the ensemble. In many applications the local analysis scheme significantly reduces the impact of a limited ensemble size and allows for the use of EnKF with high-dimensional model systems.

The degree of approximation introduced by the local analysis depends on the range of influence defined for the observations. In the limit that this range becomes sufficiently large to include all of the data, the solution for all the gridpoints becomes identical to the standard global analysis. The range parameter must be tuned and should be large enough to include the information from measurements that contribute significantly, but small enough to eliminate the spurious impact of remote measurements.

The local analysis algorithm goes as follows. We first construct the input matrices to the global EnKF, that is, the measured ensemble perturbations $\boldsymbol{S}$, the innovations $\boldsymbol{D}'$, and either the measurement perturbations $\boldsymbol{E}$ or the measurement error covariance matrix $\boldsymbol{C}_{\epsilon\epsilon}$. We then loop through the model grid, and, for each gridpoint, for example, (i, j) for a two-dimensional model, we extract the rows from these matrices corresponding to measurements that

are used in the current update, and then compute the matrix $\boldsymbol{X}_{(i,j)}$ that defines the update for gridpoint (i, j).

The analysis at gridpoint (i, j) becomes

$$\boldsymbol{A}^{\mathrm{a}}_{(i,j)} = \boldsymbol{A}_{(i,j)}\boldsymbol{X}_{(i,j)} \tag{15.4}$$

$$= \boldsymbol{A}_{(i,j)}\boldsymbol{X} + \boldsymbol{A}_{(i,j)}(\boldsymbol{X}_{(i,j)} - \boldsymbol{X}), \tag{15.5}$$

where $\boldsymbol{X}$ is the global solution, while $\boldsymbol{X}_{(i,j)}$ becomes the solution for a local analysis corresponding to gridpoint (i, j) where only the nearest measurements are used in the analysis. Thus, it is possible to compute the global analysis first, and then add the corrections from the local analysis if these effects are significant.

The quality of the EnKF analysis is connected to the ensemble size used. We expect that, to achieve the same quality of the result, a larger ensemble is needed for the global analysis than the local analysis. In the global analysis, a large ensemble is needed to properly explore the state space and to provide a consistent result that is as good as the local analysis. Note also that the use of a local analysis scheme is likely to introduce non-dynamical modes, although the amplitudes of these modes are small if a large enough influence radius is used when selecting measurements. We also refer to the discussions on localization and filtering of long-range correlations by *Mitchell et al.* (2002).

15.5 Adaptive localization methods

In adaptive localization methods, the assimilation system itself is used to determine the localization strategy. Such algorithms are useful since the dynamical covariance functions change in space and time, and the spurious correlations depend on the ensemble size. Thus, every assimilation problem and ensemble size requires a separate tuning of the localization parameters.

The hierarchical approach in *Anderson* (2007b) uses several small ensembles to explore the need for using localization in the analysis. This approach uses a Monte Carlo method based on splitting the ensemble into several small ensembles to assess the sampling errors and the spurious correlations. This method is a statistically consistent approach to the problem. However, the localization is optimized for a small ensemble and may become suboptimal when used with the full ensemble including all realizations.

An alternative localization method in *Bishop and Hodyss* (2007) is based on the online computation of a flow-dependent moderation function that is used to damp long-range and spurious correlations. This method is named SENCORP for "smoothed ensemble correlations raised to a power". The idea is that the moderation functions can be generated from a smoothed covariance function, which, when raised to a power, damps small correlations.

In *Fertig et al.* (2007) a local analysis method handles measurements that are integral parameters of the model state. In this case it is not easy to use

distance based localization. Instead an alternative algorithm is used to select the measurements to be used in an update of the variables at a particular gridpoint, where only the measurements that are significantly correlated with the model variables in the particular gridpoint are assimilated.

Thus, while traditional localization methods are distance based, *Anderson* (2007b), *Bishop and Hodyss* (2007), and *Fertig et al.* (2007) discuss adaptive localization methods where the assimilation system determines whether correlations are significant or spurious, and whether a particular measurement shall be used in the update of a particular model variable. The further development of adaptive localization methods is important for many applications where distance-based methods are less suitable.

Finally, it is not clear how the local analysis scheme is best implemented in EnKS. One approach is to define the local analysis to use only measurements in a certain space-time domain, taking into account the propagation of information in the model together with the time scales of the model. In *Khare et al.* (2008) EnKS is used with a high-dimensional atmospheric circulation model. The impact of spurious correlations related to the lag time in a lagged EnKS is studied, and it is pointed out that the lagged implementation facilitates localization in time.

15.6 A localization and inflation example

The advection model is used to examine the impact of inflation and localization in the EnKF. It is in all previous examples found that the EnKF with global updates leads to an underestimate of the ensemble variance irrespective of the kind of analysis scheme used, as is seen in Figs. 11.4, 11.5, 13.3, and 14.6. We will now repeat the EnKF case from *Exp. B* discussed in Chap. 11, where the standard EnKF scheme is used to compute the updates and standard sampling is used for the initial ensemble, but introducing different localization and inflation schemes. When using the advection model there are no model errors or dynamical instabilities and the only cause for ensemble collapse are the spurious correlations introduced by using a limited ensemble size.

As an initial test of the impact of inflation we tried a range of inflation parameters on the advection example with global analysis updates, using both the EnKF and the SQRT schemes. The results are plotted in Fig. 15.2 where we show the average residual over the 50 last timesteps as a function of the inflation parameter for 10 assimilation experiments initialized with different random seeds. The upper plot shows the results from the EnKF experiments while the middle plot shows the corresponding results from the SQRT scheme. In the lower plot we give the best constant inflation parameters for the 10 EnKF and SQRT experiments. In the EnKF, an inflation parameter in the range from 1.028 to 1.045 seems to give the best result with the average best inflation parameter equal to 1.034. When using the SQRT scheme the

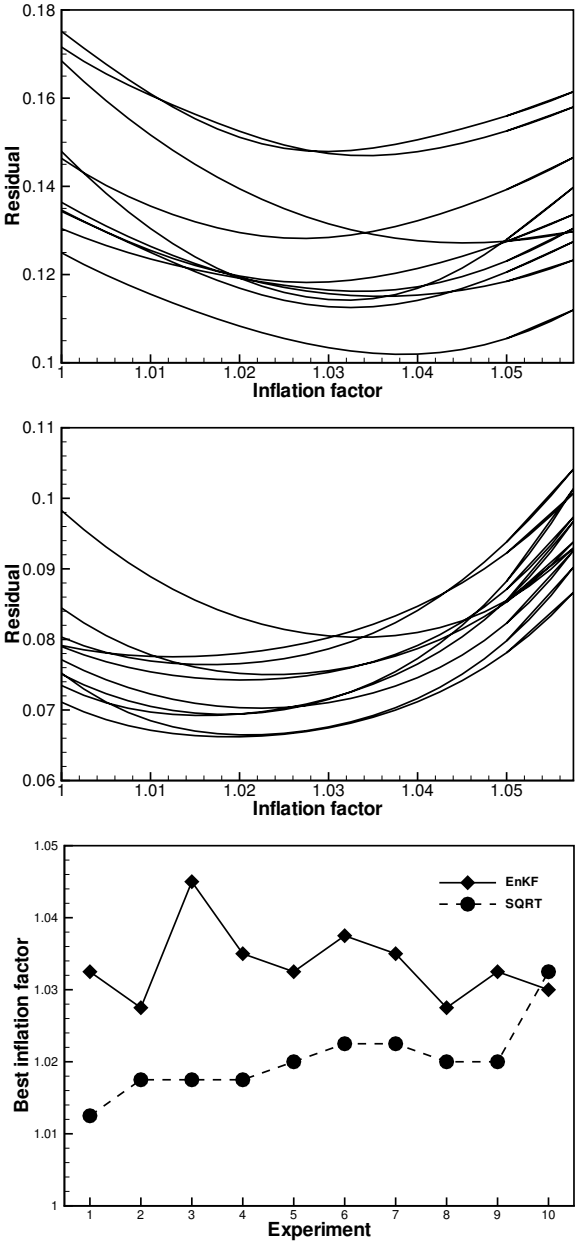

Fig. 15.2. Residual as a function of inflation parameter for 10 experiments with different random seeds. The upper plot is for the EnKF while the middle plot shows the corresponding result using the SQRT filter. The lower plot gives the best inflation parameter for the 10 experiments with different random seeds, plotted both for the EnKF and the SQRT filter.

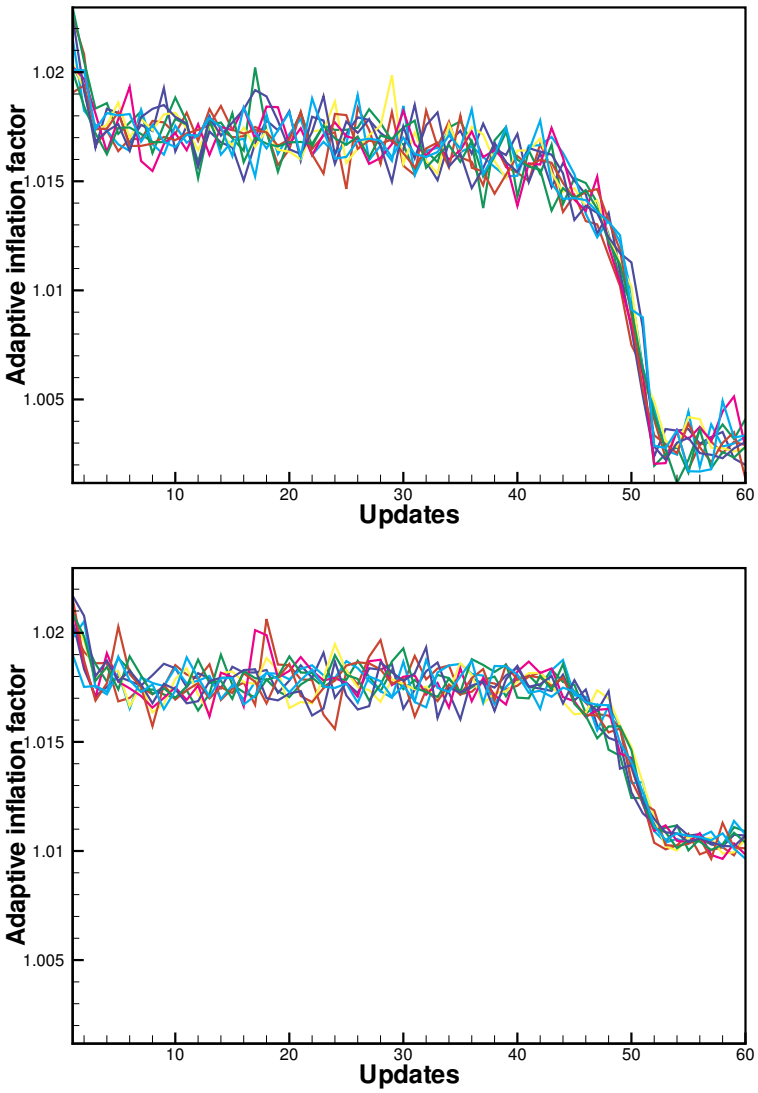

Fig. 15.3. The estimated adaptive inflation parameter as a function of updates for the 10 experiments. The upper plot is for the EnKF while the lower plot shows the results from the SQRT scheme.

corresponding range of best inflation parameters is from 1.013 to 1.033, with an average of 1.020. If we use the average residual over the whole time interval as our measure for the impact of inflation, we actually find that the results are better with a weak deflation, rather than using inflation. This result may

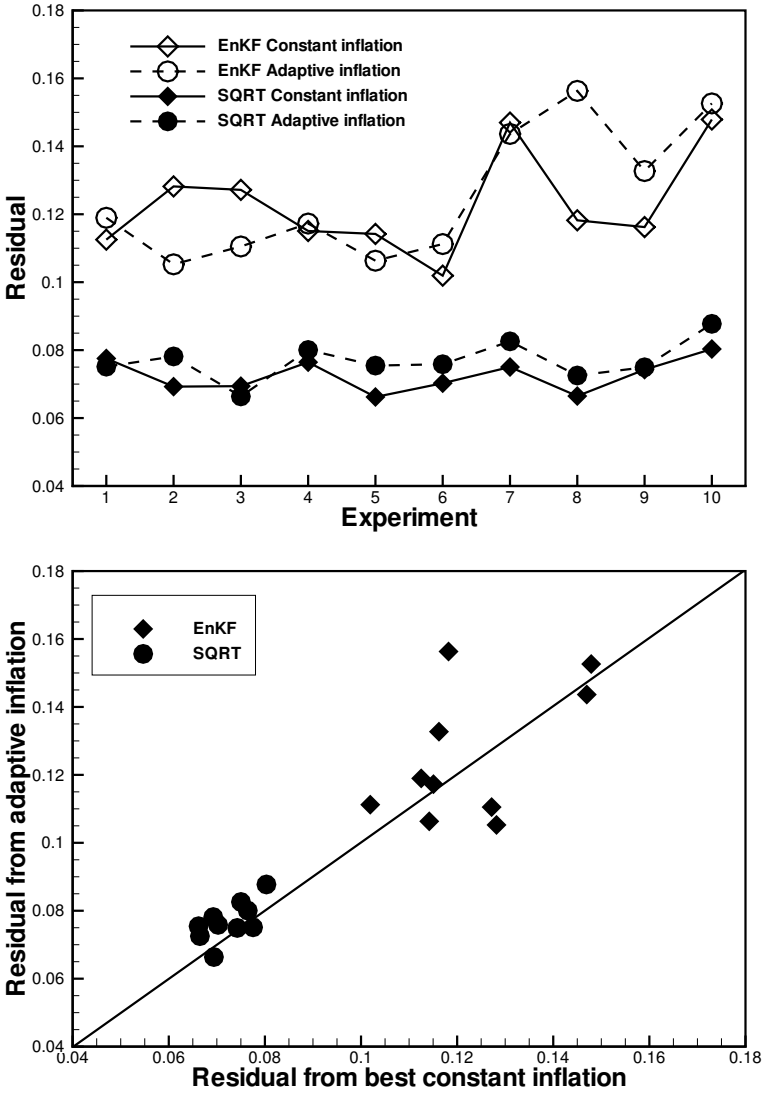

Fig. 15.4. Residual using an adaptive inflation parameter versus the inflation parameter for the 10 different experiments.

be specific for this particular example and we choose to use the residual at the final part of the simulation as the measure, since we want the filter to converge and reduce the residual over time.

Fig. 15.3 shows the adaptive inflation parameters from the 10 EnKF experiments in the upper plot and for the 10 SQRT experiments in the lower plot.

It is clear that the inflation factors are consistent in between the experiments with different random seed. For both the EnKF and the SQRT scemes, the qualitative evolution of the inflation parameter with time are similar. There is a reduction of the inflation after the first update and a further reduction after 50 updates when information from one measurement location has reached the location of the next measurement. Until the 50th update the inflation is slightly less in EnKF than in the SQRT scheme, while for the last 50 updates the EnKF scheme has a much lower inflation factor than the SQRT scheme. The reduced inflation factor after 50 updates is due the reduced innovation at this time. For the SQRT scheme, the inflation factor is contained in the range of best constant inflation factors, while for EnKF the adaptive inflation factor is slightly below the corresponding range of best constant inflation factors.

In Fig. 15.4 we plot the residuals from the different assimilation experiments, using both the best constant inflation and the adaptive inflation. It is seen that the SQRT scheme results in lower overall residuals for all the experiments when compared with the EnKF. It is also clear that the adaptive inflation results in residuals that are very similar and matching those from the best constant inflation. This result is positive with respect to using the adaptive inflation, since we cannot really be certain that we use the best constant inflation in real applications.

It is stressed that there is no history in the adaptive inflation parameter. The inflation parameter is computed only from the predicted variance, the ensemble size implicitly through the spurious correlations, the number and location of the measurements, and the measurement innovations. Furthermore, the adaptive inflation can only help avoiding ensemble collapse caused by spurious correlations. An additional underestimation of variance caused by an inability of the ensemble to represent the true state will not be corrected for by the adaptive inflation.

On the other hand, the use of localization makes it possible to search for solutions not contained in the original ensemble. In the examples to follow we examine the impact of using inflation as well as the distance based and adaptive localization. The following experiments are run:

Exp. B is the EnKF reference case using the global EnKF analysis scheme. The case is a rerun of the original *Exp. B* from Chap. 11, and the results are not identical to the original experiment, probably both due to the use of a different random seed, and the fact that the code has been updated since the original experiment was run. The major difference is that the final residual is slightly higher in the rerun.

Exp. BI is identical to *Exp. B* but uses the adaptive inflation discussed above.

Exp. F250 is identical to *Exp. B* but uses an ensemble size of 250 realizations and the SQRT analysis scheme.

Exp. F250I is identical to *Exp. BI* but uses an ensemble size of 250 realizations and the SQRT analysis scheme

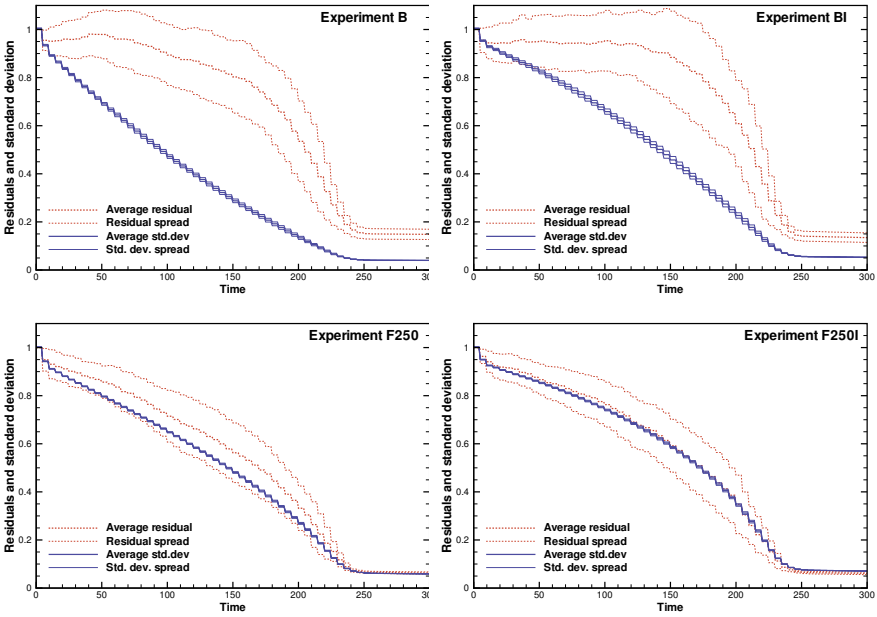

Fig. 15.5. Time evolution for RMS residuals *(dotted lines)* and estimated standard deviations *(full lines)* for all 50 simulations in the respective experiments

Exp. BL is similar to *Exp. B* but a traditional distance based local analysis scheme is used to compute the update. In this experiment only measurements located within a distance of two characteristic length scales are used in the update at a particular gridpoint.

Exp. BLS is similar to *Exp. BL* but an additional smoothing by a Shapiro filter is applied to all realizations after each update.

Exps. BLA20, BLA25 and *BLA30* use adaptive localization where a truncation level at a correlation of respectively 0.20, 0.25, and 0.30 is used when selecting the measurements to retain in the update of a particular gridpoint.

Exps. BLA20S, BLA25S and *BLA30S* are similar to *Exps. BLA20, BLA25* and *BLA30* but an additional smoothing by a Shapiro filter is applied to all realizations after each update.

In Figs. 15.5, 15.6 we show the residuals as a function of time for the different experiments. In *Exp. B* we note that there is a large mismatch between the predicted errors and the mean of the squared residuals. The underestimation of the ensemble variance is caused partly by the variance reduction introduced by the spurious correlations and partly by the inability of the 100 member ensemble to properly represent the true solution. From *Exp. BI* it is clear that the use of the adaptive inflation only leads to a partial improvement in the

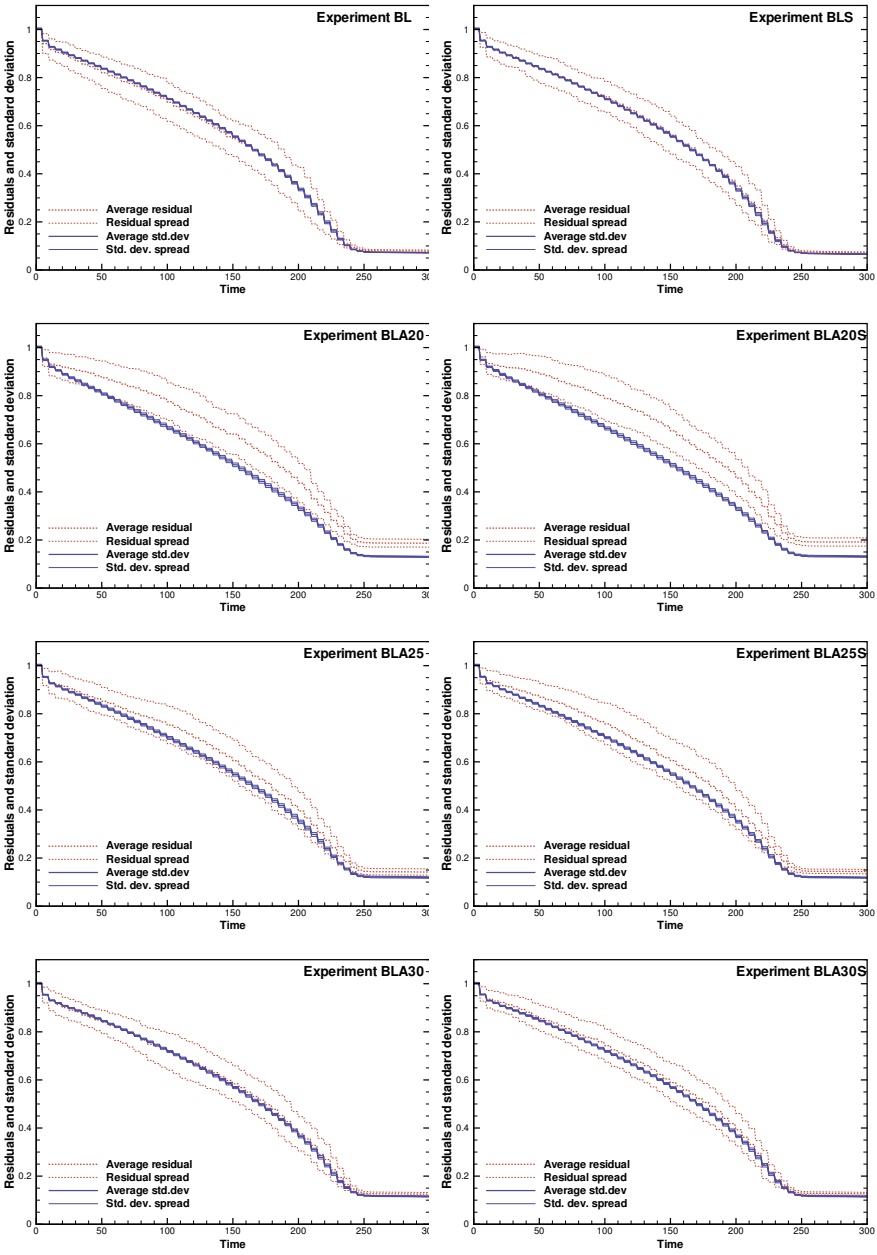

Fig. 15.6. Time evolution for RMS residuals *(dotted lines)* and estimated standard deviations *(full lines)* for all 50 simulations in the experiments with localization.

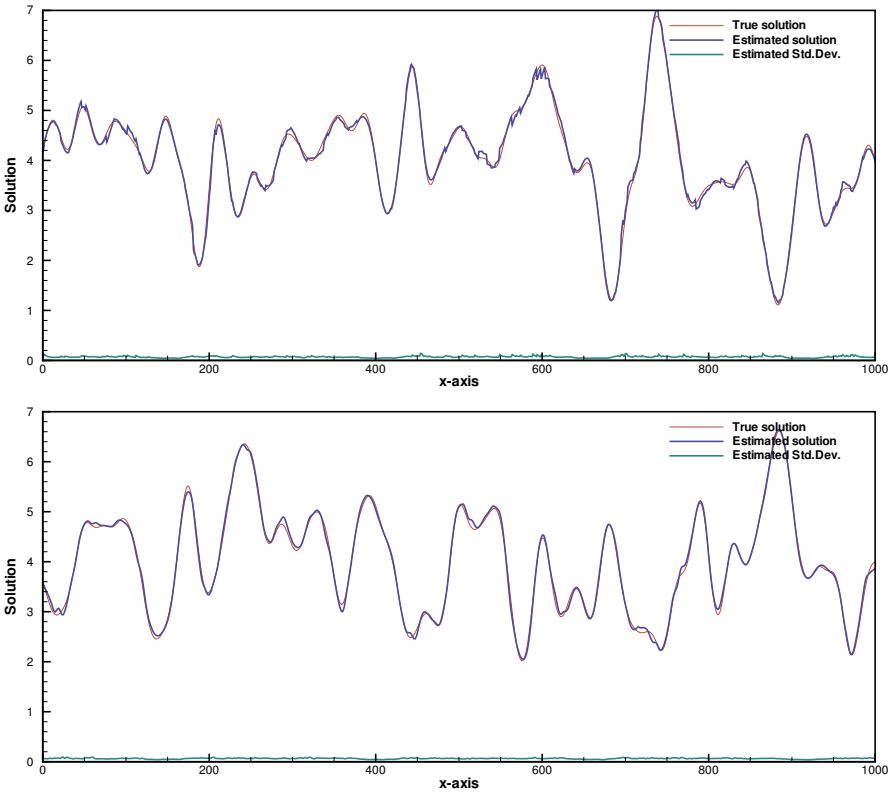

Fig. 15.7. The upper plot shows the final estimate from one of the EnKF simulations in *Exp. BL*, while the lower plot shows the corresponding result for *Exp. BLS*.

overall representation of the errors during the final part of the experiment. The estimated errors are slightly larger, and the residuals at the final time are slightly reduced, but one cannot say that the inflation has fixed the problem of under-representation of the error variance. *Exps. F250* and *F250I* repeats the *Exps. B* and *BI* but using a larger ensemble size of 250 members and the SQRT analysis scheme as in *Exp. F* from Chap. 13 to avoid any impact of measurement perturbations. The large size of the ensemble ensures that the full solution space is well represented by the ensemble. In this case the adaptive inflation leads to an ensemble variance that is fairly close to and consistent with the true residuals. Thus, the impact of spurious correlations is corrected for by the adaptive inflation.

The distance-based localization used in *Exp. BL* results in a significant improvement in the residuals compared to *Exps. B* and *BI* and the results are as good as for the *Exp. FI*. The actual residuals matches well the estimated variance and the residuals are significantly reduced. The localization allows

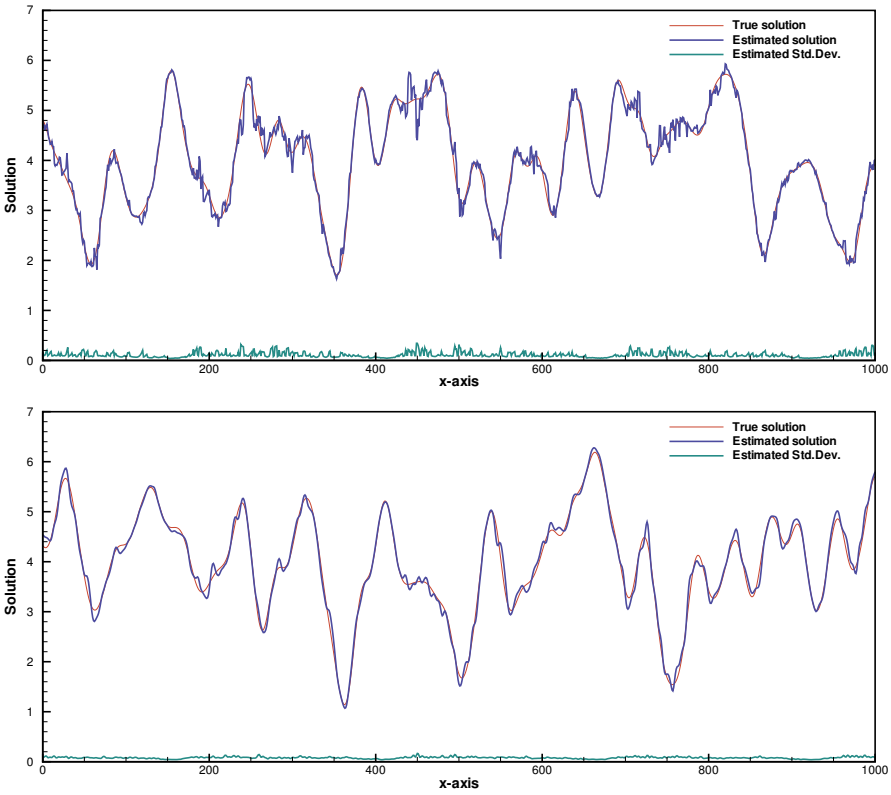

Fig. 15.8. The upper plot shows the final estimate from one of the EnKF simulations in *Exp. BLA25*, while the lower plot shows the corresponding result for *Exp. BLA25S*.

for the solution to be found outside the original space spanned by the initial ensemble and it is now possible to properly represent the true solution by the updated ensemble members. In the current example, only measurements located within a distance equal to two characteristic lengths from a particular gridpoint is used to update the gridpoint.

In the upper plot of Fig. 15.7 we show the final estimated solution in one EnKF experiment using the distance based localization as in *Exp. BL*. It is seen that the localization introuces some small scale noise in the estimate. This noise might be reduced by using a sufficiently large influence radii for the measurements, but with a limited ensemble size there will always be some noise introduced into the estimate when a distance based localization method is used without any smoothing. In the current model with exact advection there is no dissipation or diffusion and once introduced the noise is retained in the solution. The noise does not result in any nummerical problems for this

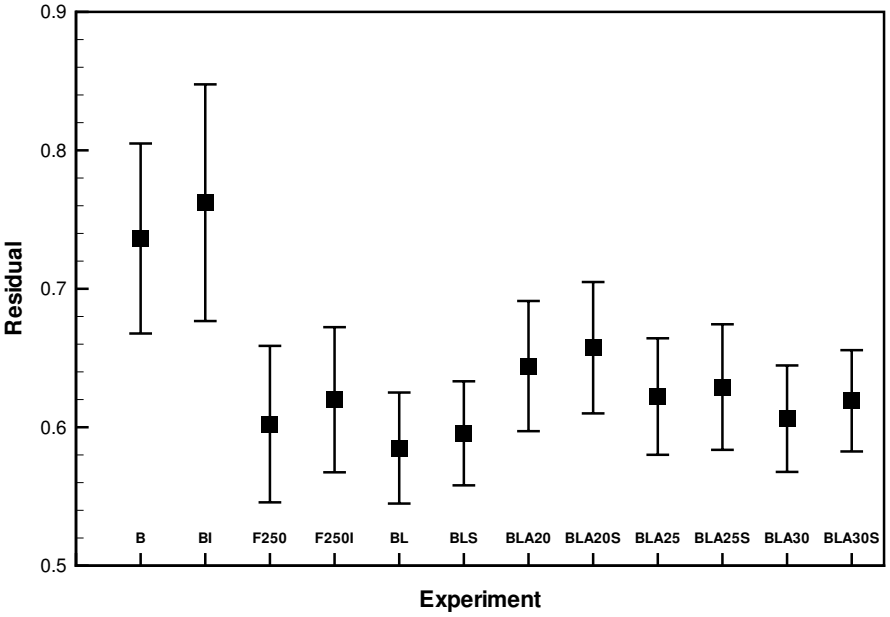

Fig. 15.9. Average residual and standard deviation from the experiments.

particular model, but for a more realistic and nonlinear model some kind of smoothing needs to be introduced. The smoothing may be implicit diffusion in the numerical schemes or an explicit filtering of the solution using for example the Shapiro filter. In *Exp. BLS* we have repeated *Exp. BL* but applied a second order Shapiro filter on each realization after each update step. It is clear from Figs. 15.6 and 15.7 that the application of the Shapiro filter removes the small scale noise without significantly impacting the residuals.

The adaptive localization, which is based on a truncation using the correlation functions, also provides a significant improvement to the results when examining the residuals. The improvement is nearly as good as those obtained using the distance based localization. It appears that a truncation at a correlation around 0.30 gives the optimal result, which is in agreement with the results from *Fertig et al.* (2007). However, a closer examination of the upper plot in Fig. 15.8, which show the final estimated solution in one EnKF experiment using the adaptive localization in *Exp. BLA25*, shows that the adaptive localization introduces more noise in the estimate than is seen in the example with distance based localization. On the other hand, the lower plot shows that the use of the Shapiro filter effectively filter away the noise and makes the adaptive localization an alterntive to consider, in particular for models where the influence regions for the measurements are poorly known.

Finally, in Fig. 15.9 we have plotted the averaged residuals for all the experiments. It is not clear if the average over the whole time period is the best measure, since the impact of the early large residuals will dominate the result. On the other hand, we plot the average residuals here as well since they are also plotted for the experiments in the previous chapters. The results can be qualitatively derived from the residuals plotted in Figs. 15.5 and 15.6, and we find that the average residuals increase when we introduce inflation in *Exp. B*. The final residuals in *Exp. BI* are lower and in better consistence with the estimated residuals. In *Exp. F250I* we find that the introduction of inflation leads to a residual that is in good agreement with the estimated errors, even though the average residuals are slightly increased. The use of distance based localization in *Exp. BL* results in the lowest average residuals of all the experiments. The use of the Shapiro filter only slightly impacts the results in *Exp. BLS* but also leads to more physically acceptable realizations without discontinuities. For the adaptive localization, it seems that a fairly strong truncation at a correlation of 0.30 gives the best result, and clearly in this case the smoothing of the realizations should be included.

16

An ocean prediction system

The ocean modelling community has been in the forefront when it comes to developing advanced data assimilation systems and taking these into use in real applications. This chapter will briefly present one such system, named TOPAZ, forming the North Atlantic and Arctic component of the European "MERSEA" integrated system, and being one of the contributors to the international Global Ocean Data Assimilation Experiment (GODAE). The system is based on the latest scientific developments in terms of ocean modelling with the Hybrid Coordinate Ocean Model (HYCOM) and data assimilation with the EnKF.

16.1 Introduction

The need for high quality predictions of marine parameters has been well identified. During recent years, offshore oil-exploration activities have expanded off the continental shelfs to deeper waters. Drilling and production of oil and gas at depths of 2000 meters or more are ongoing at several locations, and the Arctic Shelf contains considerable gas resources in ice-covered areas. This has introduced a need for real time forecasts of oceanic currents and sea-ice which in some cases may have severe impact on the safety related to drilling, production and critical operations. In addition, sustainable exploitation of marine resources through commercial fisheries and fish farming are becoming increasingly important. Fisheries management systems will benefit from accurate prediction of marine parameters such as nutrient and plankton concentrations, and this will lead to more accurate monitoring and prediction of fish stocks. Thus, there are needs for operational monitoring and prediction of both physical and biological marine parameters.

An ocean data assimilation system allows for the integration of remotesensing and in situ observations of ocean, ice, biological, and chemical variables, with coupled marine ecosystem (*Natvik and Evensen*, 2003a,b) and ice-ocean general circulation models (*Brusdal et al.*, 2003, *Lisæter et al.*, 2003).

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_16,

This integration can best be done using advanced data assimilation techniques. In the ocean community there has been a strong focus on the development and implementation of consistent data assimilation techniques that can be used with primitive equation models and also models of the marine ecosystem. Further, the real time processing and flow of observational data have now been developed to a degree where both satellite and in situ data are available in near real time. Several ocean forecasting systems are exploiting this real time flow of observed information in data assimilation systems and provide operational ocean forecasts.

The TOPAZ system consists of the HYCOM ocean model (*Bleck*, 2002) which has been coupled to two different sea-ice models, one is a simple model for ice-thickness and ice-concentration while the other is multi-category sea-ice model which represents ice-thickness distributions. Further, four ecosystem models of increasing complexity have been integrated in the system.

The TOPAZ system has been developed to meet the needs from future users of marine parameters. The system development has been supported by two previous European Commission funded projects, i.e. the DIADEM and TOPAZ projects, and current work is aimed at integration into the European MERSEA system within the MERSEA Integrated Project. TOPAZ results are displayed on the web-page `http://topaz.nersc.no` as well as validation statistics against in situ data provided by the Coriolis center.

16.2 System configuration and EnKF implementation

The model domain used for the TOPAZ prediction system is shown in Fig. 16.1. The grid is created using a conformal mapping of the poles to two new locations using the algorithm outlined in *Bentsen et al.* (1999). The horizontal model resolution varies from 11 km in the Arctic to 18 km near the Equator.

The TOPAZ system has a huge state vector consisting of 79.6 million variables just for the physical ocean parameters. The inclusion of the marine ecosystem multiplies the number of unknowns by a factor 2 to 3, depending on the ecosystem model formulation used. The system uses 100 members in the ensemble, thus the computational cost of running the system is 100 times the cost of running a single model. Fortunately, the members evolve completely independently of each other and the new parallel clusters with multiple CPUs are very well tailored to this kind of application. Clearly, to a similar computational cost it is possible to run a single model with quadruple resolution. On the other hand, we then lose the opportunity to update this single model consistently with the observations, and simplified and less consistent assimilation schemes need to be used. We would also lose the possibility to generate error estimates for the predictions.

The number of observations assimilated is huge. It consists of satellite observed sea level anomalies merged from four satellites (ERS2, Jason1, EN-

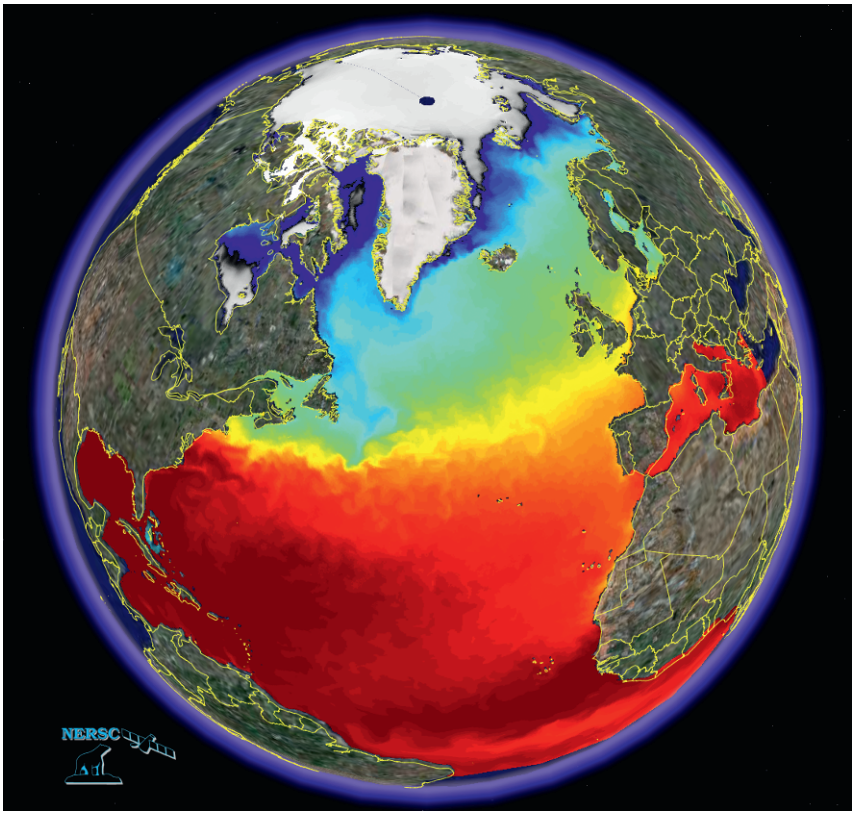

Fig. 16.1. Surface temperature and sea ice concentrations in the North Atlantic and Arctic Ocean with the TOPAZ system as viewed in Google Earth.

VISAT and GEOSAT follow on), available from Collecte Localisation Satellites (CLS) on a grid containing 100 000 observations in the North Atlantic at each assimilation cycle. In addition, TOPAZ assimilates 40 000 gridded ice concentration data from SSM/I and 8000 sea surface temperature observations (Reynolds SST), still with relatively low resolution (120 km at the Equator). When higher resolution products (25 km) will be available from the Medspiration project the number of SST data assimilated will increase to around 200 000 observations depending on cloud coverage.

Clearly, it is a challenge to represent the solution search space for such a large state vector and when assimilating this many measurements using only a limited ensemble size. It is possible to use the sophisticated analysis schemes discussed in the previous chapters, but for this particular system a slight modification is required. In *Haugen and Evensen* (2002), *Brusdal et al.* (2003), *Evensen* (2003), *Ott et al.* (2004) an algorithm named "local analysis" was used. This is a rather simple approach where the analysis update is com-

puted grid point by grid point, and using only observations located within a certain distance from the grid point, see Chap. 15 for a detailed discussion. In an ocean model it is convenient to consider this as an update of grid column by grid column since the depth is much less than the horizontal scale of the model.

The local analysis is spatially discontinuous and the updated ensemble members may not represent solutions of the original model equations, but the deviation should not be to large as long as the range of influence is large enough. In addition the updated ensemble members are not represented in the space spanned by the predicted ensemble. In fact, the use of an update matrix which varies smoothly throughout the grid effectively reduces the dimension of the problem. That is, in an ocean model where we update the solution grid column by grid column, we are solving many small problems instead of one large. In the TOPAZ system the number of unknowns in each grid column is of the same order as the number of ensemble members (113 for 22 hybrid vertical layers), as well as the number of local observations (50 at most).

The quality of the EnKF analysis is clearly connected to the ensemble size used. We expect that a larger ensemble is needed for the global analysis than the local analysis to achieve the same quality of the result. That is, in the global analysis a large ensemble is needed to properly explore the state space and to provide a consistent result for the global analysis. We expect this to be application dependent. Note also that the use of a local analysis scheme is likely to introduce non-dynamical modes, although the amplitudes of these will be small if a large enough influence radius is used when selecting measurements. In dynamical models with large state spaces, the local analysis allows for the computation of a realistic analysis result while still using a relatively small ensemble of model states. This also relates to the discussions on localization and filtering of long range correlations by *Mitchell et al.* (2002).

The TOPAZ system is run every week and produces two weeks forecasts. The propagation and analysis steps are orchestrated by a collection of scripts in the following way: every Tuesday the observations are collected and the analysis is run sequentially for each observed variable[1], then a single member forecast is run until the two-weeks forecast, initialized by the ensemble average, the whole ensemble is then propagated by the model with perturbed forcing fields (winds and thermodynamic forcing). The communication between the analysis and propagation steps is done by files so that both executables are distinct and mostly independent. This allows separate upgrades of the model and analysis codes. The propagation step requires 1200 CPU hours per week but is "embarrassingly parallel" and the hundred independent jobs are easily patched into the supercomputer idle time. TOPAZ runs on the super-

[1] This is meant to avoid scaling issues when assimilating different types of observations and it is in theory correct in the Gaussian case as all statistics (mean and variance-covariance) are updated by each observation set. This is not the case with OI-type of methods because the background covariance remains unchanged.

computing facilities of Parallab at the University of Bergen that are shared with many other users but the privileges required by the operational system are relatively small and do not represent a nuisance to other users.

The single member forecast dumps boundary conditions for nested models. Running an ensemble forecast is also possible starting from the latest analysis ensemble.

16.3 Nested regional models

To meet the end users needs of high resolution accurate information, regional models with very high resolution are embedded into the TOPAZ system in the target areas where mesoscale processes must be properly resolved. The nested models depend on the basin-scale model but the global system is not dependent on the regional models, thus each nested system can be tuned on purpose to satisfy one application without disturbing the globality of the system.

With the inclusion of a nesting capability and the assimilation of both in situ data and data from a variety of satellite sensors, the TOPAZ system constitutes a state of the art and flexible operational ocean prediction system. The model system has been designed to be easily extensible to other geographical areas including the global domain and it allows for nesting of an arbitrary number of regional high resolution models with arbitrary orientation and horizontal resolution.

Regional high-resolution models covering the Gulf of Mexico, the North Sea and the Barents Sea are currently receiving boundary conditions from TOPAZ and are run in real time. The Gulf of Mexico model uses data assimilation based on the ensemble OI method presented in Appendix A.4. It is used to predict the location of the Loop Current and the formation and propagation of rings in the Gulf of Mexico, and thus provides valuable information related to deep water drilling and oil production facilities in the Gulf of Mexico.

The only observations assimilated in the regional model are the sea surface heights from satellite altimeters, that are available with three days delay. The data assimilation is therefore performed one week back in time and assimilates gridded maps that are representative of a weekly average. The model is then integrated over the past week and two weeks forecast in the future. When necessary, e.g. when the situation is particularly dynamic, the nested system can be updated twice a week, independently from the updates of the outer model.

Figure 16.2 shows the observed limits of the Loop Current and two rings in the Northern and Northwestern Gulf of Mexico. The loop current and its detached rings may have large velocities, and when these exceed 1.5 m/s, the security of the staff and equipment is threatened and many operations have to be postponed, causing major financial losses.

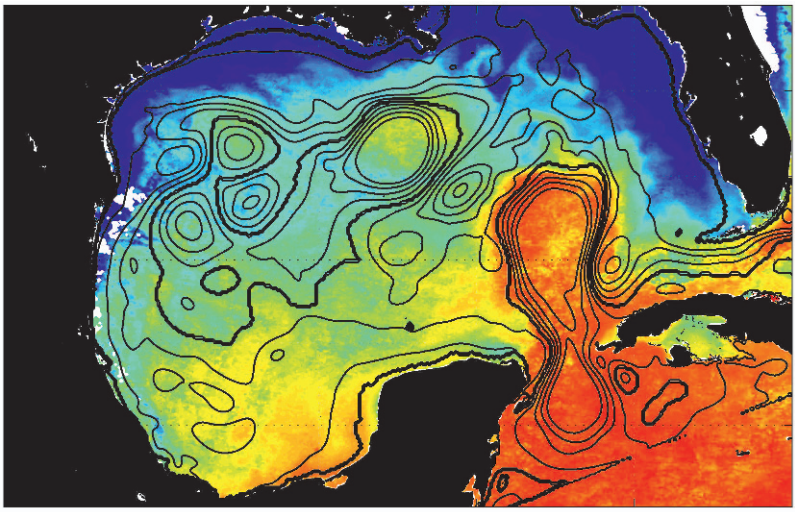

Fig. 16.2. Predicted sea-surface heights (*isolines*) overlaid a map of satellite observed sea surface temperatures (not yet assimilated) for the Gulf of Mexico on 29$^{\text{th}}$ March 2006, showing accurate positioning of the Loop Current and a detached ring. Red colours indicate high temperatures and the blue colours denote cold water.

The model nowcast (i.e. estimate at the current date) represents well the Loop Current and the two detached rings and agree well with the measured current directions, but some inaccuracies remain in the locations and extents of these features. We expect that the remaining errors are not far from being irreducible with respect to the chaotic behaviour of the small scales features, their representation by the model and in the observations. The next major improvement of the user product would therefore be a probabilistic forecast based on an ensemble. It would indicate the areas where the forecast can be given with some confidence and those where the situation is too chaotic to be predicted.

16.4 Summary

The real time operation of the system has proved to be feasible and relies on the availability of remote sensing products in near real time, and atmospheric forcing fields from the meteorological forecasting centers. The forecasts of eddies in the Gulf of Mexico have been presented to potential users in the offshore oil industry by Ocean Numerics Ltd., revealing their strong interest in the way the problem is tackled and providing useful feedback for the future product developments. Oil companies have also invested into the Barents Sea

high-resolution model which is nested into the TOPAZ system in the perspective of offshore exploration and production in the ice-covered Shtokman field. The latter system provides information on ice-ocean conditions and will be the basis for an ice and iceberg forecasting system.

There is now a strong consensus in the offshore industry, within funding agencies and among ocean researchers, on the need for development of operational ocean prediction systems. It is expected that several such systems will be established in the near future, covering the global ocean and providing valueable information about the state of the ocean both to commercial users and the public.

17

Estimation in an oil reservoir simulator

The EnKF has recently been taken into use with simulation models for oil and gas reservoirs, with the purpose of estimating poorly known parameters and to improve the predictive capability of the models. There are economical benefits of obtaining a model which best possible represents the reservoir. Optimally, it could be used for predicting the future production and to assist in the planning of new production and injection wells. A better model also provides insight and understanding regarding the properties of the reservoir.

Parameter estimation in reservoir simulation models is often named "history matching" by reservoir engineers, and the purpose is to find model parameters that result in simulations which better match the production history. History matching has traditionally been considered as a manual process where the engineer wisely tunes parameters and the impact is examined through model simulations.

Recently, there has been a growing interest in more mathematical and statistical methods for history matching. These involve both brute force direct minimization techniques and gradient methods based on the use of adjoints. Common for these is that they have all considered a pure parameter estimation problem, and not the combined parameter and state estimation problem as was advocated in the previous chapters.

An alternative approach based on the EnKF was proposed by *Nævdal et al.* (2003), where the reservoir model state and parameters were updated sequentially in time, using the information contained in pressure and rate measurements from production wells. There are now several groups continuing this work and below an application of the EnKF for history mathing in an oil reservoir model is discussed.

17.1 Introduction

An oil reservoir often consists of layers of sand and shale, each characterized by their respective porosity and permeability. The sands and shales are sed-

G. Evensen, *Data Assimilation*, 2nd ed., DOI 10.1007/978-3-642-03711-5_17,

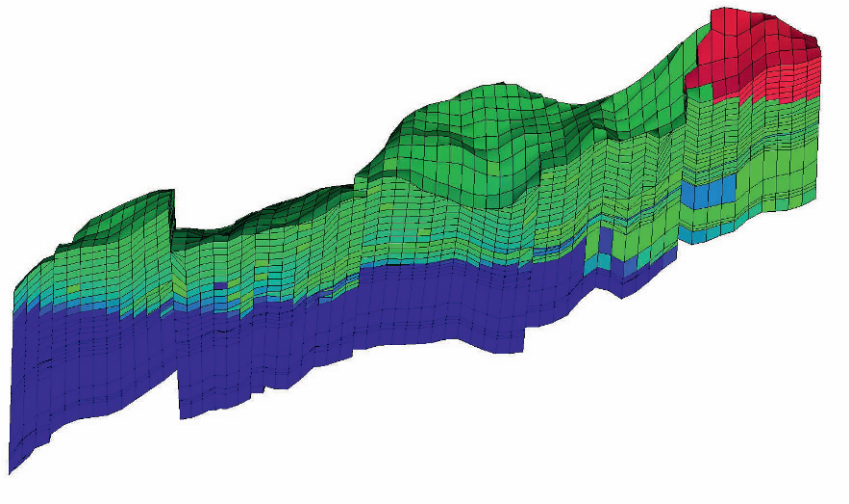

Fig. 17.1. Cross-section through the reservoir simulation model. Red colour represents gas, green denotes oil and blue is water

iments deposited on the seabed during different geological regimes, and are characterized by the porosity, $\phi(\boldsymbol{x})$, describing the fraction of a sand body which can accommodate fluids and the permeability, $\boldsymbol{k}_h(\boldsymbol{x})$, which describes how well fluids can flow in the reservoir. Normally the porosity of the reservoir sands is about 10–30 %, dependent on the grain size which varies for different depositional environments. The permeability is measured in a unit named Darcy, where 1 Darcy (D) is of order 10^{-12} m^2. Typical reservoirs have permeabilities in the range 0.1–10 D.

For reservoir sands to contain hydro-carbons, the permeable sands must be overlaid by an impermeable shale, or cap-rock, which prevents the oil and gas from escaping the reservoir. During geological time the sand layers fold and tilt, and faults may develop. The faults may become impermeable as well. Thus, the reservoir boundaries consist of the cap-rock and impermeable faults which enclose the oil and gas.

The density of gas is much less than the density of oil and water, and the mobility of gas is also much higher than for oil and water. Oil is also lighter than water and in a hydrostatic equilibrium we find gas overlaying oil and water below the oil. Fig. 17.1 shows a cross section of an oil reservoir in the North Sea. This reservoir is limited by an upper impermeable layer of shale and the horizontal extension is determined by two sealing faults. The depths of the gas-oil contact (GOC) and water-oil contact (WOC) are clearly identified. Note also the four faults located within the reservoir.

A reservoir simulation model describes the flow of oil, gas and water in the reservoir. The state vector in a reservoir model consists of the reservoir pressure, P, and saturations of water, gas and oil; S_{w}, S_{g} and S_{o}. The knowledge

of two saturations allows for the computation of the third one. In addition one often includes variables describing the amount of gas which is in a fluid state at reservoir conditions and which becomes gas at the surface, R_s, and also gas in the reservoir which condensates and becomes fluid at surface conditions, R_v. When a well is drilled into the reservoir and operated at a pressure lower than the reservoir pressure, this sets up gradients in the reservoir pressure and the reservoir fluids start flowing towards the well.

The reservoir model is coupled to a model describing the flow of fluids in the wells. There are both production wells where oil, gas and water are produced from the reservoir, and injection wells which are used to pump water, gass and sometimes other chemicals into the reservoir to maintain the reservoir pressure and to force the oil and gas towards the production wells. The wells are often controlled by valves at the surface which regulate the rate of flow in the well and thus the pressure in the well.

Recent studies with reservoir simulation models suggest that the EnKF can be used for improved reservoir management. This was first proposed by *Nævdal et al.* (2002, 2003) who used the EnKF in a simplified reservoir model to estimate the permeability of the reservoir. They showed that there could be a great benefit of using the EnKF to improve the model through parameter estimation, and that this could lead to improved predictions. These initial works have been followed by several more recent publications (see the listing in the Appendix). These have mostly considered simplified reservoirs and various test cases. The estimated parameters comprise porosity and permeability and the data assimilated have been well pressures and rates. An exception is *Skjervheim et al.* (2005) where seismic 4D-data were assimilated as well. In the next sections we describe an implementation of the EnKF with a reservoir simulator for a North Sea field example.

17.2 Experiment

It is clear that there are large uncertainties when it comes to defining the exact properties of the reservoir. Geologists and geophysicists start by estimating the location of the top of the reservoir. Then, using seismic data together with log-data from test wells, combined with a good geological understanding of the depositional processes, they develop a conceptual model for the layering of different sand types and shales in the reservoir. A structural geologist will analyse the presence of faults in the reservoir and develop a structural model. This will also be based on the relatively few test wells and the seismic data. Using data from the test wells one attempts to identify the locations of the fluid contacts, as well as the properties of the oil, gas and water in the reservoir. One can then build a set of initial models or realizations of the reservoir using various statistical simulation methods.

17.2.1 Parameterization

The first step in the history matching procedure is to identify the parameters which determines the uncertainty of the model and need to be estimated. We have now assumed that the structural model is fairly accurate, i.e. the locations of faults and layers in the model are reasonable. This may not be the case but it is currently not clear how the EnKF can be used to estimate structural parameters, since the update equation in the EnKF combines ensemble members, and these all need to be defined on the same numerical grid.

Fluid contacts

In the current application we have identified large initial uncertainties in the oil-water and gas-oil contacts, WOC and GOC. The reservoir consists of several compartments which are separated by more or less insulating faults. Unless we have vertical wells penetrating the contacts it is difficult to obtain good estimates of them. The depths of the contacts varies between different isolated regions and we only have information from wells drilled through a few of these. The initial uncertainty of the WOC had in some regions standard deviations of up to 30 m. Thus, a major set of parameters to be estimated is the WOC and GOC in the different regions of the model, since this determines the volume of oil in the reservoir as well as the optimal vertical location of horizontal production wells.

Fault transmissibilties

With a large number of faults and only few pressure measurements there is a large uncertainty in the assumed fault transmissibilities. Thus, we also include the set of transmissibilities, *multflt*, of the faults as parameters to be estimated.

Vertical layer transmissibilties

The vertical flow in the reservoir is normally determined by the vertical permeability. In the current experiment we set the vertical permeability equal to 10 % of the horizontal permeability which is included as a parameter to be estimated. Instead of estimating the vertical permeability directly we include a parameter, *multz*, which describes how well fluids will flow between model layers. This is a constant for each layer, which is multiplied with the vertical permeability to get the effective vertical communication between two layers. Some of the model layers are also assumed to be more or less impermeable for vertical flow and the estimates of *multz* should allow us to determine the layers with low vertical communication.

Porosity and permeability fields

We have also included the full three dimensional porosity and permeability fields, $\phi(\boldsymbol{x})$ and $\boldsymbol{k}_h(\boldsymbol{x})$, as variables to be estimated. The porosity is important to be able to estimate the volume of oil a part of the reservoir can contain, e.g. by increasing the porosity in a region we allow for more oil to be accommodated there. The permeability determines how well fluids are flowing through the reservoir and need to be adjusted to match the observed production rate as well as the timing of the water breakthrough.

17.2.2 State vector

For the combined parameter and state estimation problem we define the state vector to contain dynamic variables of the reservoir model, such as the pressure and saturations, and static variables as defined above. With the parameters included in this example the EnKF update of each ensemble member can be written in a simple form as

$$\underset{\text{Update}}{\begin{Bmatrix} P \\ S_{\mathrm{w}} \\ S_{\mathrm{g}} \\ R_{\mathrm{s}} \\ \boldsymbol{k}_h \\ \phi \\ multz \\ multflt \\ WOC \\ GOC \end{Bmatrix}_j} = \underset{\text{Forecast}}{\begin{Bmatrix} P \\ S_{\mathrm{w}} \\ S_{\mathrm{g}} \\ R_{\mathrm{s}} \\ \boldsymbol{k}_h \\ \phi \\ multz \\ multflt \\ WOC \\ GOC \end{Bmatrix}_j} + \sum_i \alpha_{ji} \underset{\text{Covariances}}{\begin{Bmatrix} C(P, d_i) \\ C(S_{\mathrm{w}}, d_i) \\ C(S_{\mathrm{g}}, d_i) \\ C(R_{\mathrm{s}}, d_i) \\ C(\boldsymbol{k}_h, d_i) \\ C(\phi, d_i) \\ C(multz, d_i) \\ C(multflt, d_i) \\ C(WOC, d_i) \\ C(GOC, d_i) \end{Bmatrix}_j}, \tag{17.1}$$

where j is a counter for the ensemble members and i is a counter for the measurements. The coefficients, α_{ji}, define the impact each measurement has on the update of the ensemble members.

It is seen that the different dynamic and static variables are updated by adding weighted covariances between the modelled measurements and the variables, one for each measurement. Note that both the state variables and the various parameters are updated simultaneously.

The reason why it is possible to update the parameters given only rate information from the wells, is that the rates are dependent on the properties of the reservoir as given by the parameter set defined above. Thus, there will exist correlations between reservoir properties and the observed production rates.

Considering that the porosity and permeability are defined as 3D fields with one unknown on each grid node, there is a large number of parameters to be estimated in the current system. However, the number of degrees of

freedom of the parameter space is much less than the actual number of parameters. The reason is that the porosity and permeability are smooth fields and do not consist of independent numbers in each grid node. The smoothness is prescribed from prior statistics through horizontal and vertical correlations which characterizes each depositional environment in the model. This effectively reduces the actual dimension of the problem and makes it tractable using a finite ensemble size in the EnKF.

In a particular application, where we are trying to estimate, e.g. the permeability, this implies that we can only expect to find corrections to the permeability estimates which can be represented in the space spanned by the initial permeability ensemble. This is, however, only a practical restriction since its impact can be reduced by either increasing the ensemble size or by chosing the initial ensemble wisely.

Another issue considers the scales which can be estimated for permeability. This is also clearly dependent on the initial choice of ensemble members. The "smoothness" of the members should be chosen to represent the true scales of the permeability field while keeping in mind that the limited number of wells and measurements certainly constrains the scales which can be resolved or estimated.

The model has about 82 000 active grid nodes, and the state vector then consists of 328 000 dynamic variables, 5 *WOC* and *GOC* contacts, 42 fault transmissibilities, 24 vertical multipliers, and 82 000 parameters for each of the porosity and permeability. An initial ensemble of 100 model states were generated.

Priors for the first guesses of the parameters are constructed based on the interpretation and information available from several data sources in the project. In particular the ensemble of contacts are simulated as independent numbers drawn from a Gaussian distribution with the mean equal to a best guess estimate and standard deviations of 20 m. Note that the contacts are only used initially to initialize the model, and then define the vertical saturation profile for each region. By including the contacts in the state vector, they will be updated in every assimilation step, although they are not used explicitly in the model but rather indirectly through the updates of the saturations. At the end of the assimilation experiment we have obtained improved estimates of the contacts, which can then be used in new model simulations or oil volume calculations.

The first guesses of the fault transmissibilities are set to either 1.0, 0.1 or 0.001 and with standard deviations of 20 %. This took into account knowledge about some of the faults that are known to be almost closed.

The vertical multipliers had first guesses equal to 1.0 except for three of the layers that were assumed to have low vertical permeability from the well-log data. Standard deviations were set to 10–20 %.

The porosity and permeability fields are simulated using the algorithm from Sect. 11.2, based on average values, uncertainties, and Gaussian vari-

ograms with horizontal and vertical de-correlation lengths, as specified from the geological interpretation of the reservoir.

17.3 Results

Initially we ran a pure ensemble integration of the prior ensemble. The spread of the results then provides an indication if the parameter space and the perturbations used, lead to a realistic representation of the uncertainty in the model predictions. In Fig. 17.2 we have plotted, as the red curves, the total accumulated oil production from the first 20 ensemble members together with the actual production. The upper plot shows the total accumulated field production while the middle and lower plots show the prediction of the accumulated production from the two individual production wells, P1 and P2. It is clear that the uncertainties in the initial parameter space leads to a large uncertainty in the model predictions. Without access to the production history it would not be possible to discriminate between the different realizations since all of them represent a statistically valid representation of the reservoir. From the individual wells it is also clear that there is a problem in the simulation of P1 where we have very little spread and much to large oil production. The simulation of P2 leads to a huge uncertainty, but it also captures the magnitude of the observed production.

In the EnKF experiment we have assimilated the production rates of oil (OPR), the gas-oil-ratio (GOR) and the water cut (WCT), from the two production wells. In the assimilation run we obtained rates of oil, water and gas which were in good agreement with the observations, as is expected since these are also the data assimilated. Another verification test was therefore performed. The ensemble of estimated parameters, i.e. porosity and permeability, fault and vertical multipliers, and the initial contacts, were all used in a new pure ensemble integration starting from time zero. The results from this simulation are plotted as the blue curves in Fig. 17.2. It is clear that the initially predicted uncertainties have been significantly reduced, and this must be attributed to the use of improved values of the static model parameters. Thus, we have successfully managed to compute improved estimates of a total of more than 164 000 poorly known model parameters.

The estimates of the porosity and permeability for one of the model layers are plotted in Fig. 17.3. The ensemble mean for the estimated porosity and permeability are given, respectively, in the upper and lower plots in the left column. It is clear that the estimated fields have developed clear and significant structures when compared with the first guess ensemble mean which was constant throughout the model layer. The standard deviations are reduced by approximately 25% during the assimilation updates.

Another test was also carried out where results were compared with a third production well, P3, which was excluded from the assimilation experiment. It was shown that the ensemble of improved model parameters resulted in

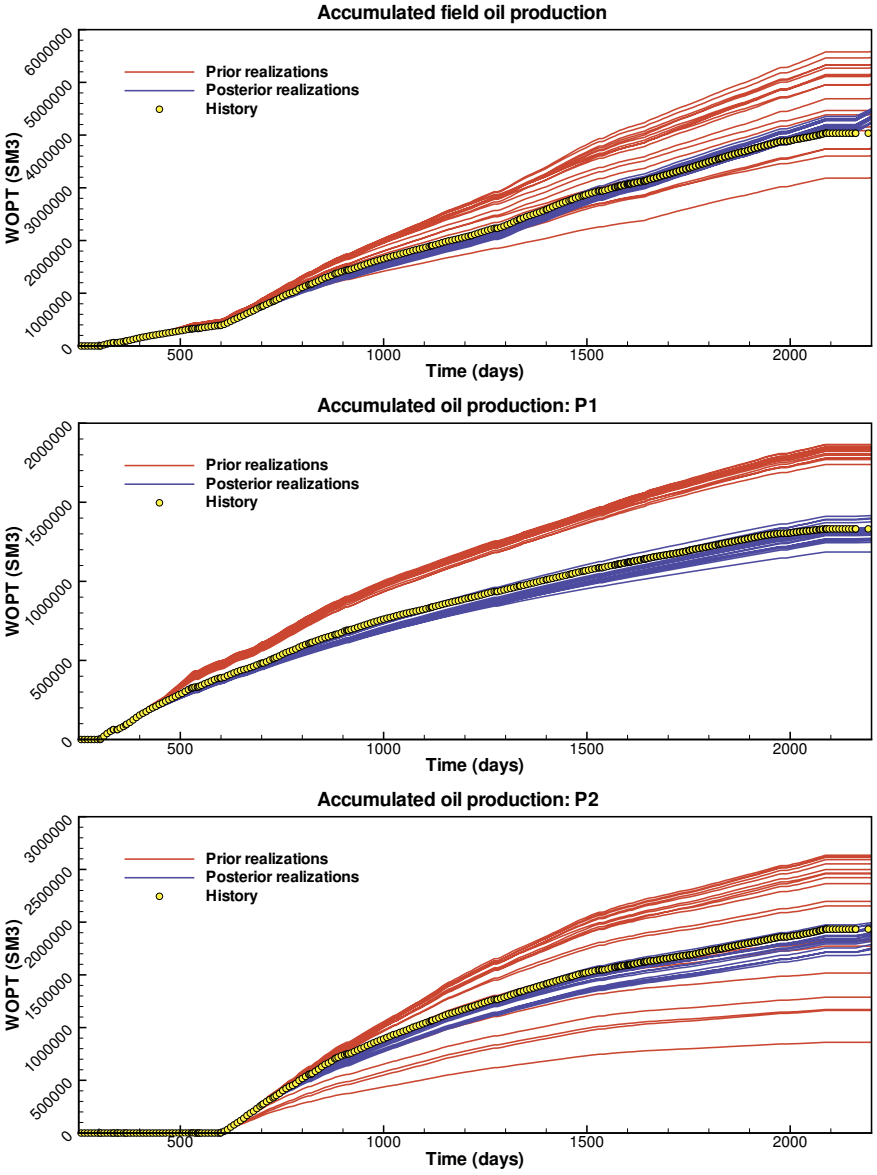

Fig. 17.2. Ensemble prediction based on initial ensemble of realizations. The total accumulated field oil production is shown in the upper plot. The middle and lower plots show the total accumulated oil production for the two wells

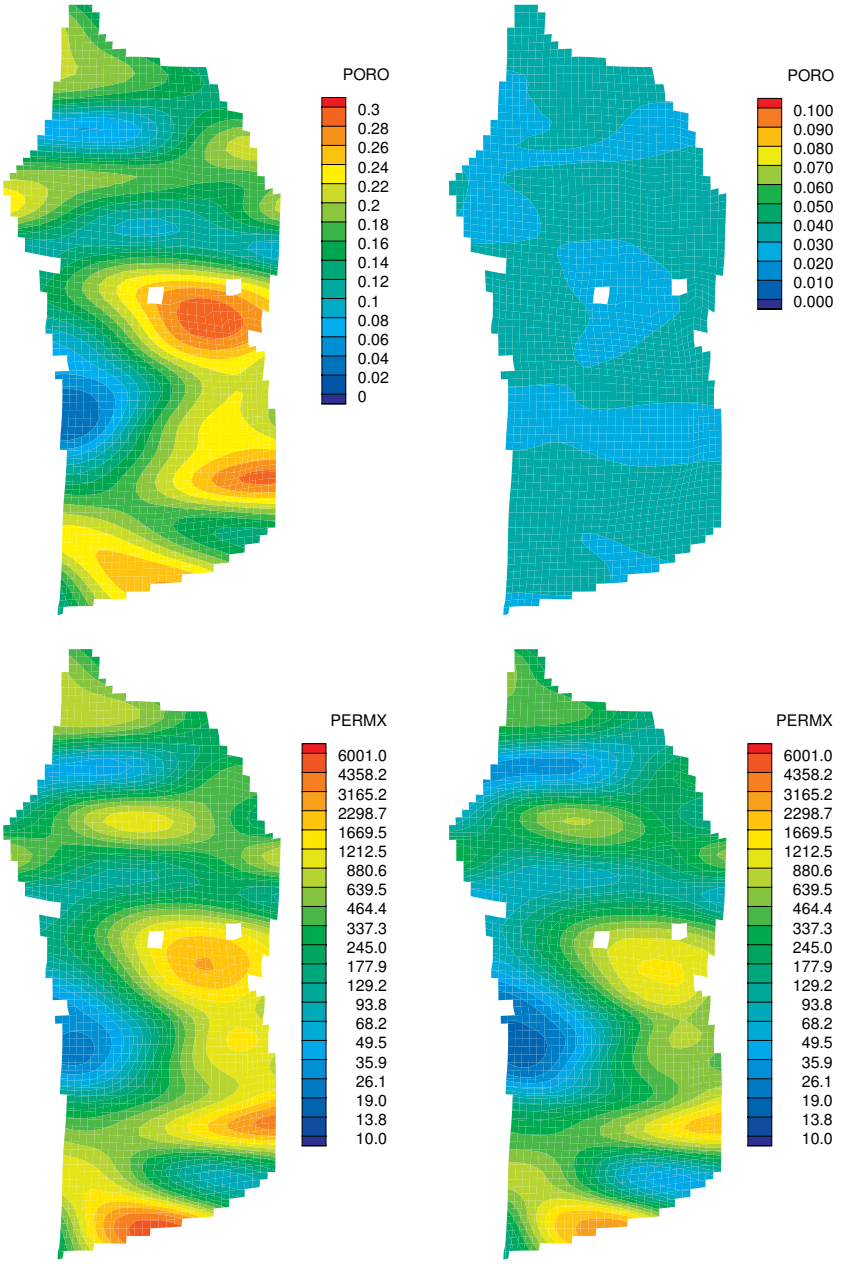

Fig. 17.3. Estimated porosity and permeability *(left column)* with standard deviations *(right column)* in one of the model layers

a significant improvement also for this well. This is an indication that the estimated model parameters are realistic and the improved realizations may then be used for the simulation and design of future wells.

17.4 Summary

The EnKF provides an ideal framework for real-time updating and prediction in reservoir simulation models. Every time new observations are available and are assimilated there is an improvement of the model parameters, and the associated model saturations and pressure. Thus, the analyzed ensemble provides optimal realizations which are conditioned on all previous data, and which can be used in a prediction of the future production. A single realization could be integrated forward in time starting from the ensemble mean or median, to obtain a quick forecast. Alternatively, the whole ensemble could be used in a forward integration to provide a future prediction with uncertainty estimates.

The EnKF has provided a tool for parameter estimation in cases with large number of poorly known parameters. It does not appear to suffer from the curse of dimensionality and multiple local minima, which have been observed in many other methods. This must be attributed to the sequential processing of observations, but also the fact that the EnKF also allows for model errors in addition to errors in the estimated parameters. Furthermore, the solution is searched for in the space spanned by the ensemble members rather than the high dimensional parameter space. Clearly, this approach should be examined in applications with other dynamical models as well.