THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut National Polytechnique de Toulouse (INP Toulouse)*

Présentée et soutenue le *11/12/2014* **par :**

El houcine BERGOU

Numerical methods for least squares problems with application to data assimilation

---

JURY

Luis Nunes Vicente	Professeur d'Université	Président du Jury
Serge Gratton	Professeur d'Université	Directeur de thèse
Philippe Toint	Professeur d'Université	Rapporteur
François Le gland	Professeur d'Université	Rapporteur
Jan Mandel	Professeur d'Université	Examinateur
Jean I. Tshimanga	Chercheur	Examinateur

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées
Unité de Recherche :
Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS)
Directeur de Thèse :
Prof. Serge GRATTON
Rapporteurs :
Prof. François LE GLAND et *Prof. Philippe TOINT*

Abstract

The Levenberg-Marquardt algorithm (LM) is one of the most popular algorithms for the solution of nonlinear least squares problems. Motivated by the problem structure in data assimilation, we consider in this thesis the extension of the LM algorithm to the scenarios where the linearized least squares subproblems, of the form $\min_{x \in \mathbf{R}^n} \|Ax - b\|^2$, are solved inexactly and/or the gradient model is noisy and accurate only within a certain probability.

Under appropriate assumptions, we show that the modified algorithm converges globally and almost surely to a first order stationary point. Our approach is applied to an instance in variational data assimilation where stochastic models of the gradient are computed by the so-called ensemble Kalman smoother (EnKS). A convergence proof in $L^{\mathbf{P}}$ of EnKS in the limit for large ensembles to the Kalman smoother is given. We also show the convergence of LM-EnKS approach, which is a variant of the LM algorithm with EnKS as a linear solver, to the classical LM algorithm where the linearized subproblem is solved exactly.

The sensitivity of the trucated sigular value decomposition method to solve the linearized subproblem is studied. We formulate an explicit expression for the condition number of the truncated least squares solution. This expression is given in terms of the singular values of A and the Fourier coefficients of b.

Keywords: Levenberg-Marquardt algorithm, least squares, random models, variational data assimilation, Kalman filter/smoother, ensemble Kalman filter/smoother, truncated singular value decomposition, condition number, perturbation theory.

Résumé

L'algorithme de Levenberg-Marquardt (LM) est parmi les algorithmes les plus populaire pour la résolution des problèmes des moindres carrés non linéaire. Motivés par la structure des problémes de l'assimilation de données, nous considérons dans cette thèse l'extension de l'algorithme LM aux situations dans lesquelles le sous problème linéarisé, qui a la forme $\min_{x\in\mathbf{R}^n} \|Ax-b\|^2$, est résolu de façon approximative, et/ou les données sont bruitées et précises qu'avec une certaine probabilité.

Sous des hypothèses appropriées, on montre que le nouvel algorithme converge presque sûrement vers un point stationnaire du premier ordre. Notre approche est appliquée à une instance dans l'assimilation de données variationnelles où les modèles aléatoires du gradient sont calculés par le lisseur de Kalman d'ensemble (EnKS). On montre la convergence dans $L^{\mathbf{P}}$ de l'EnKS vers le lisseur de Kalman, quand le nombre d'ensemble tend vers l'infini. On montre aussi la convergence de l'approche LM-EnKS, qui est une variante de l'algorithme de LM avec l'EnKS comme solveur linéaire, vers l'algorithme classique de LM où le sous problème est résolu de façon exacte.

La sensibilité de la méthode de décomposition en valeurs singulières tronquée est étudiée. Nous formulons une expression explicite pour le conditionnement de la solution des moindres carrés tronqués. Cette expression est donnée en termes de valeurs singulières de A et les coefficients de Fourier de b.

Mots clés: L'algorithme de Levenberg-Marquardt, Moindres carrés, modèles aléatoires, assimilation de données variationnelles, filtre/lisseur de Kalman, filtre/lisseur de Kalman d'ensemble, décomposition en valeurs singulières, conditionnement, théorie de perturbation.

Acknowledgements

I would like to express my special appreciation and thanks to my advisor **Serge Gratton**, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless.

I am equally grateful to **Jan Mandel**, **Jean Tshimanga Ilunga** and **Luis Nunes Vicente** from whom I learned so much. I appreciate their fruitful discussions which make my PhD experience productive and stimulating. The joy and enthusiasm they have for their research was contagious and motivational for me, even during tough times in the PhD pursuit. I would also like to thank **François Le Gland** and **Philippe Toint** for their acceptance to be the referees for my PhD. thanks to you.

I am also grateful to all of the **ALGO team members** at CERFACS for being with me during the past three years. Special thanks in particular to **Xavier Vasseur** and **Selime Gürol** for their help, advices, and encouragement. CERFACS administration would not be that efficient without **Brigitte Yzel**, **Michèle Campassens** and **Chantal Nasri**. Thanks to them for their permanent support in administrative procedures. They were always available to solve my problems with patience and smile.

My special thanks to my best friend **Diouane Youssef**, thanks for all the good and the bad times spent together. Thank you for the unforgettable and hard discussions that we had together. Very special thanks to **Nassira Akrad** for her never-ending support, trust, encouragement and understanding. My thanks go to my family and friends: **Ahmed**, **Boujemaa**, **Mohamed**, **Fatima**, **Aziz**, **Majid**, **Techfa**, **Omar**, **Ibrahim**, **Hamza**, **Monserat**, **Caroline**, **Julien**, **Abdelhadi**, **Mouloud**, **Adil**, **Azhar**, **Nabil**, **Bassam**, **Daoud**, **Hassan**, ...

Lastly, and most importantly, A special thanks to my parents. Words can not express how grateful I am to my dearest mother **Zahra Bergou** and my dearest father **Assou Bergou**. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

Contents

List of Figures

List of Tables

List of Algorithms

To my family.

Chapter 1

Introduction

There has been long interest in understanding random phenomena, and quantifying uncertainties in various scientific areas. For example, in meteorology, to predict the weather, Lewis F. Richardson in 1922 has proposed that it is possible by solving numerically the equations of the physical laws that govern the atmospheric motion [104]. In 1950 the first successful numerical prediction of the weather was performed by [23]. Since then, with the advent of electronic computers, the accuracy of numerical weather prediction models has improved steadily [102].

Uncertainties and randomness arise because models for real-world phenomena are too complicated to be described accurately. Therefore, it is necessary to make simplifications, and assumptions to find models which explain the main dynamical processes of the real-world phenomena. Once a simplified model is constructed, a random system state can be estimated by using techniques from estimation theory.

Estimation theory [48, 74, 108] is concerned with the determination of the best estimate of an unknown parameter vector of a random system, using the observations and the prior knowledge [28, 49] about the behavior of the system. An estimator takes a set of noisy observations, and uses a dynamical model (e.g. a linear predictive model) of the process (the models explaining the system motion) [122, 124] to estimate the unknown parameters. The estimation accuracy depends on the available information and on the efficiency of the estimator.

Usually the vector of unknowns to be estimated contains many parameters. The most usual ones concern the initial condition, parametric forcing, and functions modeling the errors on the model. In many fields such as geophysics and meteorology, the knowledge of an accurate initial condition is crucial for forecasting [98, 100]. The initial condition can not be fixed only using observations, because the measurements are generally

incomplete, sparse and local, and often only indirectly related to the model variables [83, 95]. Furthermore, each observation source has different error characteristics that depend on the properties of each instrument. Also a direct integration of the initial conditions using only the model may lead to a fast divergence [4, 57, 80]. Consequently, we can say that usually the observations alone, without a model, are not sufficient to characterize the system, whereas a model without any observations, does not provide sufficient information on the system. Thus, the best answer lies in combining both the observations and a model.

One of the methods to combine the information from the model and observations (to estimate the unknowns) is the Bayesian estimation [21, 89]. It is a framework for the formulation of statistical inference problems. In the prediction or estimation of a random process from a related observation signal, the Bayesian philosophy is based on combining the evidence contained in the signal with prior knowledge about the process by minimizing the so-called Bayes' risk function. Bayesian methodology includes the classical estimators such as maximum a posteriori (MAP) [51] , maximum-likelihood (ML) [32] and minimum mean square error (MMSE) [116].

The estimation process (the process used for combining the prior information and the observations) in meteorology and oceanography, is known as data assimilation [9, 20, 73]. This problem is often posed in one of these two ways: (i) *Variational methods*, such as 3 dimensional variational method (3DVAR) [19] and 4 dimensional variational method (4DVAR) [19, 114], construct least square estimates using two norms weighted by the inverse of the covariance matrices. The square error produced by the deviation from the original model state and observations is minimized using an iterative method. (ii) *Sequential methods*, such as Kalman filter/smoother [69, 120], extended Kalman filter [70], ensemble Kalman filter/smoother [42, 76] and particle filters [37, 53, 121]. These techniques solve the problem of assimilation sequentially, in the sense that they give an estimator at each time when new observations become available. These techniques use Bayesian inference.

Nowadays, 4DVAR is a worldwide dominant data assimilation method used in weather forecasting centers [36, 50, 62, 67, 101, 103]. 4DVAR attempts to reconcile model and observations variationally, by solving a weighted nonlinear least squares problem. The minimized objective function is the sum of the squares of the differences of the initial state from a known background state at the initial time and the differences of the values of observation operator and the observations at every given time point. In the weak-constraint 4DVAR [114], the model error is accounted by allowing the ending and starting state of the model at every given time point to be different, and adding to the least squares also the sums of the squares of those differences. The sums of the

squares are weighted by the inverses of the appropriate error covariance matrices, and a lot of work in the applications of 4DVAR goes into modeling those covariance matrices [26, 30, 34, 45, 96].

A widely used algorithm to solve 4DVAR problem, or more generally to solve any nonlinear least squares problem, is the Gauss-Newton algorithm [16], known in the data assimilation community under the name of incremental four dimensional variational method (Incremental 4DVAR) [27]. The Gauss-Newton algorithm relies on the approximate solution of a sequence of linear least squares subproblems in which the nonlinear least squares objective function is approximated by a quadratic function in the neighborhood of the current nonlinear iterate. However, it is well known that this simple variant of the Gauss-Newton algorithm does not ensure a monotonic decrease of the objective function. These problems arise, for example, in the case of highly nonlinear or very large residual problems [33, p. 225]. Hence the convergence of Gauss-Newton algorithm is not guaranteed [33, p. 225]. Handling this difficulty is typically achieved by using either line-search [33], trust-region [25], or Levenberg-Marquardt [79, 88, 92, 94] (also known as Levenberg-Morrison-Marquardt [25]) methods , which under appropriate assumptions, ensure global convergence to first order critical points. We consider the latter method in this thesis.

The Levenberg-Marquardt algorithm can be seen as a regularization of the Gauss-Newton algorithm. A regularization parameter is updated at every iteration and indirectly controls the size of the step, making Gauss-Newton globally convergent, i.e., convergent to stationarity independently of the starting point. We found that the regularization term added to Gauss-Newton maintains the structure of the linearized least squares subproblems arising in data assimilation, enabling us to use techniques like ensemble methods while simultaneously providing a globally convergent approach (see Chapters 4 and 5).

However, the use of ensemble methods, such as ensemble Kalman filter/smoother in data assimilation poses difficulties since it makes random approximations to the gradient. We thus propose and analyze a variant of the Levenberg-Marquardt method to deal with probabilistic gradient models (see Chapter 4). It is assumed that an approximation to the gradient is provided but it is only accurate with a certain probability. The knowledge of the probability of the error between the exact gradient and the model one can be used in our favor in the update of the regularization parameter. We show that using ensemble methods to solve 4DVAR linearized subproblem is equivalent to use the Levenberg-Marquardt method based on probabilistic models. Then, we illustrate

numerically our approach using as forecast models Lorenz 63 model [84], and the quasi-geostrophic model [44] (see Chapters 5 and 6). We investigate also in this thesis the asymptotic behavior of the new methods in the limit for large ensembles (see Chapter 7).

Having in mind the approximations and the errors in data, we consider as inexact the solution of the linearized least squares subproblem coming from each iteration of the Gauss-Newton or Levenberg-Marquardt methods. When the problem is ill-conditioned, a better solution of the subproblem, in the sense that it is less sensitive than the original one (the exact solution of the subproblem) to errors in data is obtained by truncating the original least squares solution. The Truncated Singular Value Decomposition (TSVD) [16] method is well known for these kind of problems. In this thesis we will study the sensitivity of the solution of a given subproblem (linear least squares problem) to perturbations in the data by computing the condition number of the truncated least squares solution (see Chapter 3).

This thesis is organized as follows: In Chapter 2, we present fundamental information that will be used as a reference for the other chapters. We start by giving an overview about some sequential methods for estimation theory, in particular Kalman filter/smoother, ensemble Kalman filter/smoother. Next, we present some methods for solving linear least squares problems, in particular, conjugate gradient method and the truncated singular value decomposition method. Finally, methods for solving nonlinear least squares problems will be presented, especially the Gauss-Newton and Levenberg-Marquardt methods.

In Chapter 3, a sensitivity analysis of the TSVD method will be studied. We will investigate an explicit expression of the condition number of the truncated least squares solution of $Ax = b$. The expression is given in terms of the singular values of A and the Fourier coefficient of b.

Chapter 4 gives an extension of the Levenberg-Marquardt method to the scenarios where the linearized least squares subproblems are solved inexactly and/or the gradient model is noisy and accurate only within a certain probability. We call this latter extension a Levenberg-Marquardt method based on probabilistic models. A proof of convergence to first order stationary point of new approach is given.

Chapter 5 presents the application of the approach proposed in Chapter 4 to data assimilation problems. We show that solving 4DVAR problem using ensemble Kalman smoother as linear solver is equivalent to approximating the gradients by random models. Moreover we illustrate numerically our approach using Lorenz 63 equations as a forecast model in 4DVAR problem.

In Chapter 6, we analyze the Levenberg-Marquardt method using ensemble Kalman smoother as linear solver to the filtering problems. We study the impact of different parameters on the iterations progress. We use two different forecast models in our experiments, namely Lorenz 63 model and quasi-geostropic model.

Chapter 7 studies the asymptotic behavior of some algorithms based on ensemble methods. We show the convergence of ensemble Kalman smoother in the limit for large ensembles to the Kalman smoother, and we show also the convergence of LM-EnKS Algorithm, which is a variant of the Levenberg-Marquardt algorithm with ensemble Kalman smoother as linear solver to the classical Levenberg-Marquart algorithm, where the linearized subproblem is solved exactly.

Finally, conclusions are drawn in Chapter 8, and future directions are discussed.

Contributions

The main contributions of this thesis are:

- to prove the global convergence of the Levenberg-Marquardt method with a fixed regularization parameter (see Theorem 2.1, in Chapter 2).
- to compute explicitly the condition number of the TSVD method (see Chapter 3). This work has been published in SIAM Journal on Matrix Analysis and Applications (SIMAX) [11].
- to derive an extension of the Levenberg-Marquardt method, to deal with the least squares problems where derivatives are random. We give an application of this new approach in data assimilation (see Chapters 4 and 5). This work is under revision at SIAM/ASA Journal on Uncertainty Quantification (JUQ) [12, 86].
- to illustrate numerically the new approaches and investigate the impact of different parameters on the iterations progress (see Chapters 4, 5 and 6) [86].
- to investigate the asymptotic behaviors of some ensemble based methods presented in Chapter 5 (see Chapter 7). This work is submitted for publication in SIAM/ASA Journal on Uncertainty Quantification (JUQ) [13].

Chapter 2

Background Material

This chapter consists of fundamental information that will be a reference for the following chapters. We give an overview about the estimation theory in Bayesian framework, and then we formulate the estimation problem as a least squares problem. After, solution methodologies are discussed, in particular the Newton and Gauss-Newton methods. We focus on the Gauss-Newton method as a solution algorithm, in which one solves a sequence of linear least squares subproblems. The Gauss-Newton method can be improved in terms of its convergence behavior by using trust-region strategies that we also outline in this chapter, by focusing especially on the well-known Levenberg-Marquardt method.

The solution of the linear least-squares subproblems arising in Gauss-Newton or Levenberg-Marquardt iteration can be found by solving the corresponding linear systems. Here, we present the singular value decomposition method, we give a small summary of the iterative methods, and present the conjugate gradient to solve those linear systems.

The reminder of this chapter is organized as follows, we begin by an overview about estimation theory, where we present the well known Kalman filter/smoother, ensemble Kalman filter/smoother methods. Next, we present a class of methods to solve linear least squares problem, especially singular value decomposition and the conjugate gradient methods. Finally, we present methods for solving nonlinear least squares problems, especially Gauss-Newton and Levenberg-Marquardt methods.

2.1 Estimation theory

Estimation theory is a branch of statistics that deals with the estimation of the values of an unknown parameter vector of a random system. These estimation is based on

the observations and the prior knowledge about the behavior of the system that have a random component. Bayesian theory is a framework for the formulation of estimation problems. It is based on combining the information contained in the observation with prior knowledge by minimizing the Bayes' risk function.

2.1.1 Concepts, notations and assumptions

This section presents the fundamental concepts, notations and assumptions that will be used in the dissertation:

- *True state*, or *truth* will refer to the unknown real (true) state of a given random system, which is usually random. We often look for models which somehow explain the physics, and behavior of the real problems.
- *The prior*, or the *background* will refer to the prior knowledge about the true state of a given random system, which contains the previous knowledge about the system (the knowledge about the behavior of the true state in the past).
- *Dynamical model*, or *forecast model* represents the physical laws that govern the system motion, it is imperfect, with errors arising from the approximate physics, parameterizations, and the discretization of an infinite dimensional dynamics into a numerical model.
- *The observations*, or *data* will refer to the information gathered while observing the behavior of the true state, obtained from measurements by instruments. These observations are generally incomplete and attached with errors coming from the instruments and the approximations.

The true state vector of a given system (or the vector of the unknowns of a given system) is denoted by x. The vector x_b denotes the prior about x and v_b is the error on the prior. We assume that the error on the prior is additive, i.e., x is related to x_b by:

$$x = x_b + v_b. \tag{2.1}$$

The error on the prior (v_b) is unknown because we do not know x. We assume that the prior is unbiased, i.e., the mean of the background error is equal to zero ($E(v_b) = 0$). We denote by B the error covariance matrix $\left(B = E\left(v_b v_b^\top\right)\right)$. We assume moreover that v_b is normally distributed, hence the probability density function of the random vector x is:

$$\mathbb{P}(x) = \frac{1}{(2\pi)^{n/2}|B|^{1/2}} \exp\left(-\frac{1}{2}(x - x_b)^\top B^{-1}(x - x_b)\right), \tag{2.2}$$

where n is the size of x, and $|B|$ is the determinant of the matrix B.

The observations in x are gathered into an observation vector, which we denote by y. These data are sometimes not directly related to the true state. The observation operator provides the link between x and the observations [83, 95]. We denote this operator by $\mathcal{H}$:

$$\mathcal{H} : \mathbf{R}^n \to \mathbf{R}^m.$$

This operator generates the values $\mathcal{H}(x)$ that the observations y would take in the absence of any error. In practice $\mathcal{H}$ is a nonlinear collection of interpolation operators from the model discretization to the observation points (the observation space), and conversions from model variables to the observed parameters.

The error on the observations is denoted by the vector w_o. These error is introduced by the interpolation operator, by the finite resolution of the model fields, and the instrumental errors. We assume that these error is additive, i.e., x is related to y by:

$$y = \mathcal{H}(x) + w_o. \tag{2.3}$$

We assume that the mean of the error w_o is equal to zero ($E(w_o) = 0$), and its covariance matrix is given by the symmetric positive definite matrix $R = E\left(w_o w_o^\top\right)$. In most cases the observation error covariance matrix is block-diagonal, or even diagonal, because usually it is assumed that there is no observation error correlations between independent observational networks, platforms or stations, and instruments, except in some special cases. We assume also that w_o is normally distributed, hence the probability density function of the observation knowing the real state x is:

$$\mathbb{P}(y|x) = \frac{1}{(2\pi)^{m/2}|R|^{1/2}} \exp\left(-\frac{1}{2}\left(y - \mathcal{H}(x)\right)^\top R^{-1}\left(y - \mathcal{H}(x)\right)\right), \tag{2.4}$$

where m is the size of y, and $|R|$ is the determinant of the matrix R.

2.1.2 Bayesian approach

Bayesian probability theory provides a mathematical framework for the computation of the probability of the state x knowing the data y, using probability. The foundations of Bayesian probability theory was laid down some 200 years ago based on the studies of Bayes, Price, and Laplace [14]. Bayes' rule state that the probability of x for given y is given by:

$$\mathbb{P}(x|y) = \frac{\mathbb{P}(y|x)\mathbb{P}(x)}{\mathbb{P}(y)}. \tag{2.5}$$

In (2.5), the term $\mathbb{P}(x)$ is the probability density function of the true state x prior. The term $\mathbb{P}(y|x)$ is called the likelihood function and it provides the probability of the observation y for a given true state x.

Different estimators produce different results depending on the estimation method, the observations and the influence of the prior information. Obviously, due to randomness of the observations, the same estimator would produce different results with different observations from the same process. Therefore an estimate is itself a random variable, it has a mean and a covariance, and it may be described by a probability density function. However, for most cases, an estimator is characterized in terms of its mean and its covariance matrix.

2.1.3 Best linear unbiased estimator (BLUE)

Best linear unbiased estimator gives the best linear guess to the value of x given the observed value y [1, 5]. We assume that the observation operator $\mathcal{H}$ is linear, in which case we denote it by the matrix H. We assume also that the errors v_b and w_o are independent. The mean x_{blue} of this estimator is a linear combination of the background and the observation:

$$x_{blue} = Lx_b + Ky, \tag{2.6}$$

where L and K are two matrices in $\mathbf{R}^{n\times n}$ and $\mathbf{R}^{n\times m}$ respectively. For completeness, major points in the development of the mean and covariance of the BLUE are derived here. From equations (2.1)-(2.3)-(2.6) we conclude that

$$v_{blue} = x - x_{blue} = Lv_b - Kw_o + (I - L - KH)x. \tag{2.7}$$

The BLUE is unbiased hence $I - L - KH = 0$, i.e., $L = I - KH$. The covariance matrix of v_{blue} can be obtained from equation (2.7) and the fact that the random vectors v_b and w_o are independent as follows:

$$\begin{aligned} P_{blue} &= E\left(v_{blue}v_{blue}^\top\right) = LBL^\top + KRK^\top \\ &= (I - KH)B(I - KH)^\top + KRK^\top. \end{aligned} \tag{2.8}$$

For the BLUE, the matrix K is chosen such that matrix P_{blue} has a minimum trace (which correspond to minimum of square of v_{blue}). We have

$$\delta\text{trace}(P_{blue}) = \text{trace}\left(-(\delta KH)B(I - KH)^\top - (I - KH)B(\delta KH)^\top + \delta KRK^\top + KR\delta K^\top\right).$$

Since B and R are symmetric matrices, and the trace is invariant by matrix transposition, we have:

$$\delta\text{trace}(P_{blue}) = 2\text{trace}\left(\left(-(I-KH)BH^\top + KR\right)\delta K^\top\right).$$

$\delta\text{trace}(P_{blue}) = 0$ for any $\delta K \neq 0$, if and only if:

$$K = BH^\top\left(R + HBH^\top\right)^{-1}. \tag{2.9}$$

This matrix is know in literature by *Kalman gain* or *optimal gain*. Substituting this K into the equation (2.8) gives:

$$P_{blue} \quad = \quad (I-KH)B, \tag{2.10}$$

and into equation (2.6) gives:

$$x_{blue} = x_b + K(y - Hx_b). \tag{2.11}$$

Note that, in the Gaussian case (when v_b and w_o are normally distributed), we can find the same values for x_{blue} and P_{blue}, using Bayes' rule (2.5) as follows: In this case x_{blue} coincide with the mean of $\mathbb{P}(x|y)$

$$x_{blue} = E(x|y) = \int x\mathbb{P}(x|y)dx.$$

From (2.2) and (2.4) we have:

$$\begin{aligned}
\mathbb{P}(x) &\propto \exp\left(-\frac{1}{2}(x-x_b)^\top B^{-1}(x-x_b)\right), \\
\mathbb{P}(y|x) &\propto \exp\left(-\frac{1}{2}\left(y-\mathcal{H}(x)\right)^\top R^{-1}\left(y-\mathcal{H}(x)\right)\right).
\end{aligned}$$

From one hand, Bayes' rule tells us:

$$\mathbb{P}(x|y) \propto \exp\left(-\frac{1}{2}\left((x-x_b)^\top B^{-1}(x-x_b) + \left(y-Hx\right)^\top R^{-1}\left(y-Hx\right)\right)\right).$$

On the other hand we have:

$$\mathbb{P}(x|y) \propto \exp\left(-\frac{1}{2}\left((x-x_{blue})^\top (P_{blue})^{-1}(x-x_{blue})\right)\right),$$

therefore, we obtain $\forall x \in \mathbf{R}^n$ that:

$$\begin{aligned}(x - x_{blue})^\top (P_{blue})^{-1}(x - x_{blue}) &= (x - x_b)^\top B^{-1}(x - x_b) \\ &+ (y - Hx)^\top R^{-1}(y - Hx), \\ x^\top (P_{blue})^{-1}x - 2x^\top (P_{blue})^{-1}x_{blue} + cst &= x^\top \left(B^{-1} + H^\top R^{-1}H\right)x \\ &- 2x^\top \left(B^{-1}x_b + H^\top R^{-1}y\right) + cst.\end{aligned}$$

From the latter equality, we obtain the system

$$\begin{cases} (P_{blue})^{-1} = B^{-1} + H^\top R^{-1}H, \\ (P_{blue})^{-1}x_{blue} = B^{-1}x_b + H^\top R^{-1}y. \end{cases} \tag{2.12}$$

From (2.12) and using Sherman–Morrison–Woodbury formula (see in the Appendix B) we obtain that:

$$\begin{aligned}P_{blue} &= \left(B^{-1} + H^\top R^{-1}H\right)^{-1} \\ &= B - BH^\top (HBH^\top + R)^{-1}HB \\ &= (I - KH)B, \text{ (the same value as in equation (2.10))}\end{aligned} \tag{2.13}$$

where

$$K = BH^\top \left(R + HBH^\top\right)^{-1} = \left(B^{-1} + H^\top R^{-1}H\right)^{-1} H^\top R^{-1}, \tag{2.14}$$

which is the same as in equation (2.9). Substituting equation (2.13) into equation (2.12) leads to:

$$\begin{aligned}x_{blue} &= (I - KH)B\left(B^{-1}x_b + H^\top R^{-1}Hy\right) \\ &= (I - KH)x_b + (I - KH)BH^\top R^{-1}y.\end{aligned} \tag{2.15}$$

We have

$$(I - KH)BH^\top R^{-1} = \left(B^{-1} + H^\top R^{-1}H\right)^{-1} H^\top R^{-1} = K, \tag{2.16}$$

thus, reporting (2.16) in (2.15) yields:

$$x_{blue} = x_b + K(y - Hx_b), \text{ (we obtain the same value as in equation (2.11)).}$$

2.2 Estimation using sequential techniques

In the previous section, we merely focused on the static case estimator (BLUE), in the sense that the evolution of x in time is not considered. We have derived the mean of

the BLUE, as well as its covariance, given some prior information (the background) and an observation. But for real system, our objective is to track the true state x over time. Hence we are interested in sequential estimators.

Sequential estimation methods introduce a new ingredient in the problem compared to static estimation: the dynamical model for the system state typically defined between two consecutive instants. These methods, as it is the case for the BLUE, use a probabilistic framework. Moreover they give estimates of the whole system state sequentially by propagating information in time. Let's consider a set of observations distributed over a given time interval. The subscripts will denote the quantities at any given observation time. The quantities x_k, y_k, w_k, $\mathcal{H}_k$ and R_k will denote the true state, the observation in x_k, the error on the observation, the observation operator and the covariance of the observation error respectively, at time k. We denote by p the number of time steps. Therefore:

$$y_k = \mathcal{H}_k(x_k) + w_k,\ w_k \sim N(0, R_k),\ k = 0, \ldots, p \tag{2.17}$$

$$x_0 = x_b + v_0,\ v_0 \sim N(0, B), \tag{2.18}$$

where the background is only defined at initial time. It is common to assume that the state x_{k+1} depends only on the state x_k but not on the previous ones, and observation y_k depends only on the state x_k according to the following scheme:

$$\begin{array}{ccccccccc} x_0 & \rightarrow & \ldots & \rightarrow & x_k & \rightarrow & x_{k+1} & \rightarrow & \ldots \\ & & & & \downarrow & & \downarrow & & \\ & & & & y_k & & y_{k+1} & & \end{array}$$

The objective of sequential filtering is to find the probability density function of $x_0, \ldots, x_k$ knowing the data set $y_0, \ldots, y_k$ (or at least to find the most likely state trajectories $x_0, \ldots, x_k$ knowing the data up to time k). The marginal density of x_k knowing the data set $y_0, \ldots, y_k$ is the known *filtering density*, and is often used for prediction purposes. Sequential methods estimate the latter density recursively in two steps: first the propagation step uses the dynamical model to determine the prior distribution which is the distribution of x_k knowing the data up to time $k-1$ (density of x_k knowing $y_0, \ldots, y_{k-1}$). Then a statistical analysis of the observation y_k enables to update the prior distribution and provides the posterior distribution (density of x_k knowing $y_0, \ldots, y_k$).

Since the state is changing over time, we will represent its evolution by assuming that there exists a model which represents the time evolution of x between time k and $k-1$. We denote this model by $\mathcal{M}_k$.

The errors in the model are denoted by v_k. These errors are introduced essentially by modelization of the system motion and the descritization. The state x_k is related to x_{k-1} by:

$$x_k = \mathcal{M}_k(x_{k-1}) + m_k + v_k,\ k = 1, \ldots, p, \tag{2.19}$$

where m_k is a deterministic vector. We assume that the error v_k has 0 mean ($E(v_k) = 0$), and that its covariance matrix is given by the symmetric positive definite matrix $Q_k = E\left(v_k {v_k}^\top\right)$. Moreover we assume that the random vectors $[v_k]_{k=1}^p$ are uncorrelated in time, i.e., $E\left(v_k v_l^\top\right) = 0, \forall k \neq l$. We assume also that the observation error vectors $[w_k]_{k=0}^p$ are uncorrelated in time, i.e., $E\left(w_k w_l^\top\right) = 0, \forall k \neq 0$, and that $E\left(w_k v_l^\top\right) = 0, \forall k,\ l$.

2.2.1 Kalman filter (KF)

First described by [71, 72], the KF is a simple recursive formula that implements the sequential estimation of x_k knowing the data $y_0, \ldots, y_k$, when the initial state and data distributions are independent, and the model and observation operators are linear. In the case of Gaussian errors (which is the case in this dissertation), the distributions of x_k knowing the data up to time $k-1$ or k are also Gaussian, therefore they can be represented uniquely by their means and covariances. The KF formula gives recursively the expectation of x_k knowing $y_0, \ldots, y_k$, $E\left(x_k | y_0, \ldots, y_k\right)$ and its covariance matrix $P\left(x_k | y_0, \ldots, y_k\right)$.

We denote by $x_{i|j}$ the expectation of x_i knowing $y_0, \ldots, y_j$, and by $P_{i|j}$ its covariance. For $k = 0$, if there is no observation in x_0 then $x_{0|0} = x_b$, and $P_{0|0} = B$. Otherwise $x_{0|0} = x_b + K_0(y_0 - H_0 x_b)$, and $P_{0|0} = (I - K_0 H_0)B$, where $K_0 = BH_0^\top (R_0 + H_0 B H_0^\top)^{-1}$. For $k = 1, \ldots, p$,

$$\begin{aligned}
x_{k|k-1} &= M_k x_{k-1|k-1} + m_k \text{ is the mean of } x_k \text{ given } y_0, \ldots, y_{k-1}, \\
P_{k|k-1} &= M_k P_{k-1|k-1} M_k^\top + Q_k \text{ is the covariance of } x_k \text{ given } y_0, \ldots, y_{k-1}, \\
K_k &= P_{k|k-1} H_k^\top \left(R_k + H_k P_{k|k-1} H_k^\top\right)^{-1} \text{ is the Kalman gain at time } k, \\
x_{k|k} &= x_{k|k-1} + K_k(y_k - H_k x_{k|k-1}), \\
P_{k|k} &= (I - K_k H_k)\, P_{k|k-1}.
\end{aligned}$$

We summarize the different steps of KF in Algorithm 2.1.

If the dimension of the state x_k is large, the covariance matrices $P_{k|k-1}$ and $P_{k|k}$ are very large symmetric matrices, hence storing and computing such matrices may be out of reach. To solve these problems, the idea is to use the ensemble methods.

Algorithm 2.1: Kalman filter algorithm

Initialization

Compute $x_{0|0}$ and $B_{0|0}$.

For $k = 1, 2, \ldots, p$,

1. Compute the prior mean and covariance at time k:

$$x_{k|k-1} = M_k x_{k-1|k-1} + m_k$$
$$P_{k|k-1} = M_k P_{k-1|k-1} M_k^\top + Q_k$$

2. Compute Kalman gain:

$$K_k = P_{k|k-1} H_k^\top \left(R_k + H_k P_{k|k-1} H_k^\top \right)^{-1}.$$

3. Compute the posterior mean and covariance at time k:

$$x_{k|k} = x_{k|k-1} + K_k(y_k - H_k X_{k|k-1}),$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1}.$$

2.2.2 Ensemble Kalman filter (EnKF)

The idea behind the EnKF is to use Monte Carlo samples and to use the corresponding empirical covariance matrix instead of the prediction covariance matrix $P_{k|k-1}$ [39–42, 76]. It was proposed by [39], and later amended by [22, 41, 42, 65]. The EnKF has proven to be very efficient on a large number of academic and operational problems. The EnKF is based on the concept of particles, a collection of state vectors, the members of the ensemble. Rather than propagating huge covariance matrices, the errors are emulated by scattered particles, a collection of state vectors whose variability is meant to be representative of the uncertainty of the system's state. The ensemble members index is denoted by l, it runs over $l = 1, \ldots, N$. In practice, given an ensemble $x^1_{k-1|k-1}, \ldots, x^N_{k-1|k-1}$ at time $k-1$, we build the ensemble at time k as follows:

$$x^l_{k|k-1} = M_k x^l_{k-1|k-1} + m_k + v^l_k, \; v^l_k \sim N(0, Q_k), \tag{2.20}$$

$$x^l_{k|k} = x^l_{k|k-1} + P^N_{k|k-1} H_k^\top \left(R_k + H_k P^N_{k|k-1} H_k^\top \right)^{-1} \left(y_k - w^l_k - H_k x^l_{k|k-1} \right), \; w^l_k \sim N(0, R_k). \tag{2.21}$$

In this above expression, $P^N_{k|k-1}$ is the covariance estimate from the ensemble $\left[x^l_{k|k-1} \right]_{l=1}^N$,

$$\begin{aligned} P^N_{k|k-1} &= \frac{1}{N-1} \sum_{l=1}^N \left(x^l_{k|k-1} - \frac{1}{N} \sum_{l=1}^N x^l_{k|k-1} \right) \left(x^l_{k|k-1} - \frac{1}{N} \sum_{l=1}^N x^l_{k|k-1} \right)^\top \\ &= \frac{1}{N-1} E_k E_k^\top, \end{aligned} \tag{2.22}$$

where

$$E_k = \left[e_k^1, \ldots, e_k^N\right], \; e_k^l = x_{k|k-1}^l - \frac{1}{N}\sum_{i=1}^{N} x_{k|k-1}^i, \; l = 1, \ldots, N.$$

Defining the matrix Z_k as:

$$Z_k = \left[z_k^1, \ldots, z_k^N\right], \; z_k^l = H_k x_{k|k-1}^l - \frac{1}{N}\sum_{i=1}^{N} H_k x_{k|k-1}^i, \; l = 1, \ldots, N. \tag{2.23}$$

Substituting equations (2.22) and (2.23) into equation (2.21) leads to:

$$x_{k|k}^l = x_{k|k-1}^l + \frac{E_k Z_k^\top}{N-1}\left(R_k + \frac{Z_k Z_k^\top}{N-1}\right)^{-1}\left(y_k - w_k^l - H_k x_{k|k-1}^l\right).$$

Using Sherman–Morrison–Woodbury formula we have that:

$$\left(R_k + \frac{Z_k Z_k^\top}{N-1}\right)^{-1} = R_k^{-1} - \frac{R_k^{-1} Z_k}{N-1}\left(I + \frac{Z_k^\top R_k^{-1} Z_k}{N-1}\right)^{-1} Z_k^\top R_k^{-1}, \tag{2.24}$$

and consequently,

$$x_{k|k}^l = x_{k|k-1}^l + \frac{E_k Z_k^\top R_k^{-1}}{N-1}\left[I - \frac{Z_k}{N-1}\left(I + \frac{Z_k^\top R_k^{-1} Z_k}{N-1}\right)^{-1} Z_k^\top R_k^{-1}\right]\left(y_k - w_k^l - H_k x_{k|k-1}^l\right).$$

The pseudo-code for the EnKF is given in Algorithm 2.2.

Notice that the i.i.d. random vectors $[v_k^l]_{l=1}^N$ are simulated here with the same statistics as the additive Gaussian noise v_k in the original state equation (2.19). The i.i.d. random vectors $[w_k^l]_{l=1}^N$ are simulated here with the same statistics as the additive Gaussian noise w_k in the original state equation (2.17). In the absence of observation in x_0, the initial ensemble $\left[x_{0|0}^l\right]_{l=1}^N$ is simulated as i.i.d. Gaussian random vectors with mean x_b and covariance B i.e., with the same statistics as the initial state x_0.

2.2.3 Kalman smoother (KS)

The KS [43, 90], is the recursion algorithm which gives the mean and covariance matrix of the joint state $x_0, \ldots, x_k$, knowing the complete set of observations $y_0, \ldots, y_k$ in the linear case. Denote by $x_{0:k}$ the joint state of $x_0, \ldots, x_k$, by $x_{0:k|j}$ the expectation of the joint state of $x_0, \ldots, x_k$ knowing the observations $y_0, \ldots, y_j$, and by $P_{0:k,0:k|j}$ its corresponding covariance. In

Algorithm 2.2: Ensemble Kalman filter algorithm

Initialization

Generate the initial ensemble $\left[x^1_{0|-1},\ldots,x^N_{0|-1}\right]=\left[x^l_{0|-1}\right]^N_{l=1}$, by sampling $x^l_{0|-1}\sim N\left(x_b,B\right)$, where $l=1,\ldots,N$ is the ensemble member index.

For $k=0,1,\ldots,p$

1. With $\left[x^l_{k|k-1}\right]^N_{l=1}$ already computed, Bayesian update for the observation:
   Compute the following quantities:

$$E_k=\left[e^1_k,\ldots,e^N_k\right],\ e^l_k=x^l_{k|k-1}-\frac{1}{N}\sum_{i=1}^{N}x^i_{k|k-1},\ l=1,\ldots,N.$$

$$Z_k=\left[z^1_k,\ldots,z^N_k\right],\ z^l_k=H_kx^l_{k|k-1}-\frac{1}{N}\sum_{i=1}^{N}H_kx^i_{k|k-1},\ l=1,\ldots,N$$

   Update step (correction step of the ensemble):

$$x^l_{k|k}=x^l_{k|k-1}+\frac{E_kZ_k^\top R_k^{-1}}{N-1}\left[I-\frac{Z_k}{N-1}\left(I+\frac{Z_k^\top R_k^{-1}Z_k}{N-1}\right)^{-1}Z_k^\top R_k^{-1}\right]$$
$$\left(y_k-w^l_k-H_kx^l_{k|k-1}\right),\ w^l_k\sim N\left(0,R_k\right). \tag{2.25}$$

2. While $k\leq p-1$, advance the ensemble members in time by applying the model M_{k+1} and sampling the model error:

$$x^l_{k+1|k}=M_{k+1}x^l_{k|k}+m_{k+1}+v^l_{k+1},\ v^l_{k+1}\sim N\left(0,Q_{k+1}\right) \tag{2.26}$$

(2.25) is evaluated as successive multiplications of a column vector by matrices and solving a system of the size equal to the number of ensemble members, rather than multiplying or inverting any large matrices.

the linear case the system of equations (2.17)-(2.18)-(2.19) is equivalent to the following system:

$$x_{0:k}=\begin{bmatrix}I_n & 0_n & \ldots & 0_n\\ 0_n & I_n & \vdots & \vdots\\ \vdots & \ddots & \ddots & 0_n\\ 0_n & \ldots & \ddots & I_n\\ 0_n & \ldots & 0_n & M_k\end{bmatrix}x_{0:k-1}+\begin{bmatrix}0\\ \vdots\\ m_k\end{bmatrix}+\begin{bmatrix}0\\ \vdots\\ v_k\end{bmatrix}\quad v_k\sim N(0,Q_k),\ k=1,\ldots,p \tag{2.27}$$

$$=\begin{bmatrix}I_{n(k-1)}\\ \tilde{M}_k\end{bmatrix}x_{0:k-1}+\begin{bmatrix}0\\ \vdots\\ m_k\end{bmatrix}+\begin{bmatrix}0\\ \vdots\\ v_k\end{bmatrix},$$

$$y_k=[0,\ldots,H_k]x_{0:k}+w_k\ w_k\sim N(0,R_k),\ k=0,\ldots,p \tag{2.28}$$
$$=\tilde{H}_kx_{0:k}+w_k,$$

where n is the size of x_k. The matrices I_n, and 0_n are respectively the identity matrix, and the null matrix of $\mathbf{R}^{n,n}$. The augmented matrices $\tilde{H}_k = [0, \ldots, H_k]$, and $\tilde{M}_k = [0, \ldots, M_k]$ are defined to maintain the correspondence with the filter equations. From these equations, we can derive the KS the same way as the KF:

$$
\begin{aligned}
x_{0:k|k-1} &= \begin{bmatrix} I_{n(k-1)} \\ \tilde{M}_k \end{bmatrix} x_{0:k-1|k-1} + \begin{bmatrix} 0 \\ \vdots \\ m_k \end{bmatrix} \\
&= \begin{bmatrix} x_{0:k-1|k-1} \\ M_k x_{k-1,k-1} + m_k \end{bmatrix}, \\
P_{0:k,0:k|k-1} &= \begin{bmatrix} I_{n(k-1)} \\ \tilde{M}_k \end{bmatrix} P_{0:k-1,0:k-1|k-1} \begin{bmatrix} I_{n(k-1)} \\ \tilde{M}_k \end{bmatrix}^\top + \begin{bmatrix} 0_{n(k-1)} & 0 \\ 0 & Q_k \end{bmatrix} \\
&= \begin{bmatrix} P_{0:k-1,0:k-1|k-1} & P_{0:k-1,0:k-1|k-1}\tilde{M}_k^\top \\ \tilde{M}_k P_{0:k-1,0:k-1|k-1} & \tilde{M}_k P_{0:k-1,0:k-1|k-1}\tilde{M}_k^\top + Q_k \end{bmatrix}, \\
K_k &= P_{0:k,0:k|k-1}\tilde{H}_k^\top (R_k + \tilde{H}_k P_{0:k,0:k|k-1}\tilde{H}_k^\top)^{-1} \\
&= P_{0:k,0:k|k-1}\tilde{H}_k^\top (R_k + H_k P_{k,k|k-1} H_k^\top)^{-1}, \\
x_{0:k|k} &= x_{0:k|k-1} + K_k(y_k - \tilde{H}_k x_{0:k|k-1}) = x_{0:k|k-1} + K_k(y_k - H_k x_{k|k-1}), \\
P_{0:k,0:k|k} &= (I_{nk} - K_k \tilde{H}_k) P_{0:k,0:k|k-1}.
\end{aligned}
$$

Algorithm 2.3 summarizes the steps of the KS.

Algorithm 2.3: Kalman smoother algorithm

Initialization

Compute $x_{0|0}$ and $B_{0|0}$.

For $k = 1, 2, \ldots, p$,

1. Compute the prior mean and covaraince at time k:

$$
\begin{aligned}
x_{0:k|k-1} &= \begin{bmatrix} x_{0:k-1|k-1} \\ M_k x_{k-1,k-1} + m_k \end{bmatrix}, \\
P_{0:k,0:k|k-1} &= \begin{bmatrix} P_{0:k-1,0:k-1|k-1} & P_{0:k-1,0:k-1|k-1}\tilde{M}_k^\top \\ \tilde{M}_k P_{0:k-1,0:k-1|k-1} & \tilde{M}_k P_{0:k-1,0:k-1|k-1}\tilde{M}_k^\top + Q_k \end{bmatrix}.
\end{aligned}
$$

2. Compute Kalman gain:

$$
K_k = P_{0:k,0:k|k-1}\tilde{H}_k^\top \left(R_k + H_k P_{k,k|k-1} H_k^\top\right)^{-1}.
$$

3. Compute the posterior mean and covariance at time k:

$$
\begin{aligned}
x_{0:k|k} &= x_{0:k|k-1} + K_k(y_k - H_k x_{k|k-1}), \\
P_{0:k,0:k|k} &= \left(I_{nk} - K_k \tilde{H}_k\right) P_{0:k,0:k|k-1}.
\end{aligned}
$$

2.2.4 Ensemble Kalman smoother (EnKS)

In the EnKS [42], the covariances are replaced by their approximations from the ensemble. Let

$$X_{0:k-1|k-1} = \left[\begin{bmatrix} x^1_{0|k-1} \\ \vdots \\ x^1_{k-1|k-1} \end{bmatrix}, \ldots, \begin{bmatrix} x^N_{0|k-1} \\ \vdots \\ x^N_{k-1|k-1} \end{bmatrix} \right] = \left[x^1_{0:k-1|k-1}, \ldots, x^N_{0:k-1|k-1} \right]$$

be an ensemble of N states over time up to $k-1$, conditioned on observations up to time $k-1$. Here, l is the ensemble member index. For $k = 0$, in the absence of observation in x_0, the ensemble $[x^l_{0|0}]^N_{l=1}$ are an i.i.d Gaussian random variables with the mean x_b and the covariance B. For $k = 1, \ldots, p$, we advance the model to time k by:

$$x^l_{k|k-1} = M_k x^l_{k-1|k-1} + m_k + v^l_k, \; v^l_k \sim N(0, Q_k), \; n = 1, \ldots, N,$$

we get the ensemble $X_{0:k|k-1}$ up to time k conditioned to observations up to time $k-1$,

$$X_{0:k|k-1} = \left[\begin{bmatrix} x^1_{0|k-1} \\ \vdots \\ x^1_{k|k-1} \end{bmatrix}, \ldots, \begin{bmatrix} x^N_{0|k-1} \\ \vdots \\ x^N_{k|k-1} \end{bmatrix} \right] = \left[x^1_{0:k|k-1}, \ldots, x^N_{0:k|k-1} \right].$$

Then, we incorporate the observation at time k, $y_k = \tilde{H}_k x_k + w_k$, $w_k \sim N(0, R_k)$ into the composite state the same way as for EnKF update:

$$\begin{bmatrix} x^l_{0|k} \\ \vdots \\ x^l_{k|k} \end{bmatrix} = \begin{bmatrix} x^l_{0|k-1} \\ \vdots \\ x^l_{k|k-1} \end{bmatrix} + P^N_{0:k,0:k|k-1} \tilde{H}_k^\top \left(R_k + \tilde{H}_k P^N_{0:k,0:k|k-1} \tilde{H}_k^\top \right)^{-1} \left(y_k - w^l_k - H_k x^l_{k|k-1} \right), \tag{2.29}$$

where $P^N_{0:k,0:k|k-1}$ is a covariance estimate from the ensemble $X_{0:k|k-1}$ and $w^l_k \sim N(0, R_k)$ is a random perturbation. The blocks of the sample covariance are: for $\ell, m = 0, \ldots, k$

$$\begin{aligned} P^N_{\ell,m|k-1} &= \frac{1}{N-1} \sum_{l=1}^N \left(x^l_{\ell|k-1} - \frac{1}{N} \sum_{i=1}^N x^i_{\ell|k-1} \right) \left(x^l_{m|k-1} - \frac{1}{N} \sum_{i=1}^N x^i_{m|k-1} \right)^\top \\ &= \frac{1}{N-1} E_\ell E_m^\top, \end{aligned} \tag{2.30}$$

where

$$E_\ell = \left[e^1_\ell, \ldots, e^N_\ell \right], \; e^l_\ell = x^l_{\ell|k-1} - \frac{1}{N} \sum_{i=1}^N x^i_{\ell|k-1}, \; l = 1, \ldots, N.$$

Substituting equations (2.30) and (2.23) into equation (2.29) leads to:

$$\begin{aligned}\begin{bmatrix} x^l_{0|k} \\ \vdots \\ x^l_{k|k} \end{bmatrix} &= \begin{bmatrix} x^l_{0|k-1} \\ \vdots \\ x^l_{k|k-1} \end{bmatrix} \\ &+ \begin{bmatrix} E_0 \\ \vdots \\ E_k \end{bmatrix} \frac{Z_k^\top}{N-1} \left(R_k + \frac{Z_k Z_k^\top}{N-1} \right)^{-1} \left(y_k - w_k^l - H_k x^l_{k|k-1} \right).\end{aligned} \tag{2.31}$$

Reporting (2.24) in the latter equality yields:

$$\begin{aligned}\begin{bmatrix} x^l_{0|k} \\ \vdots \\ x^l_{k|k} \end{bmatrix} &= \begin{bmatrix} x^l_{0|k-1} \\ \vdots \\ x^l_{k|k-1} \end{bmatrix} \\ &+ \begin{bmatrix} E_0 \\ \vdots \\ E_k \end{bmatrix} \frac{Z_k^\top R_k^{-1}}{N-1} \left[I - \frac{1}{N-1} Z_k \left(I + \frac{Z_k^\top R_k^{-1} Z_k}{N-1} \right)^{-1} Z_k^\top R_k^{-1} \right] \\ &\left(y_k - w_k^l - H_k x^l_{k|k-1} \right).\end{aligned}$$

The pseudo-code for the EnKS is given in Algorithm 2.4.

The (ensemble) Kalman filter (smoother) is originally based on a linear assumptions, meaning that the observation and the model operators are required to be linear. However, in some systems, these operators can be nonlinear. In this case there is variants of Kalman filter/smoother proposed to handle these problems such as extended and unscented Kalman filters [69, 70].

2.3 Estimation using optimization techniques

In the previous section, we presented the sequential method for the estimation. We presented, in particular the Kalman filter/smoother and their ensemble variants, which are derived under linearity assumption. But for real systems, these assumption is not always verified. In this case, usually the estimation problem is formulated as an optimization problem.

2.3.1 Maximum a posteriori estimator (MAP)

The maximum a posteriori estimator of the system of equations (2.17)-(2.18)-(2.19) is the maximum of the probability density function of $x_{0:k}$ knowing the data set $y_{0:k}$. Using Bayes rule

Algorithm 2.4: Ensemble Kalman smoother algorithm

Initialization

Generate the initial ensemble $\left[x^1_{0|-1}, \ldots, x^N_{0|-1}\right] = \left[x^l_{0|-1}\right]^N_{l=1}$, by sampling $x^l_{0|-1} \sim N(x_b, B)$, where $l = 1, \ldots, N$ is the ensemble member index.

For $k = 0, 1, \ldots, p$

1. With $\left[x^l_{0:k|k-1}\right]^N_{l=1}$ already computed, Bayesian update for the observation:
   Compute the following quantities:

$$E_\ell = \left[e^1_\ell, \ldots, e^N_\ell\right],\ e^l_\ell = x^l_{\ell|k-1} - \frac{1}{N}\sum_{i=1}^{N} x^i_{\ell|k-1},\ \ell = 1, \ldots, k,\ l = 1, \ldots, N,$$

$$Z_k = \left[z^1_k, \ldots, z^N_k\right],\ z^l_k = H_k x^l_{k|k-1} - \frac{1}{N}\sum_{i=1}^{N} H_k x^i_{k|k-1},\ l = 1, \ldots, N,$$

$$y_k{}^l = y_k - w^l_k - H_k x^l_{k|k-1},\ w^l_k \sim N(0, R_k),\ l = 1, \ldots, N.$$

   Update step (correction step of the ensemble):

$$\begin{bmatrix} x^l_{0|k} \\ \vdots \\ x^l_{k|k} \end{bmatrix} = \begin{bmatrix} x^l_{0|k-1} \\ \vdots \\ x^l_{k|k-1} \end{bmatrix} + \begin{bmatrix} E_0 \\ \vdots \\ E_k \end{bmatrix} \frac{Z_k^\top R_k^{-1}}{N-1}\left[I - \frac{1}{N-1} Z_k \left(I + \frac{Z_k^\top R_k^{-1} Z_k}{N-1}\right)^{-1} Z_k^\top R_k^{-1}\right] y^l_k. \tag{2.32}$$

2. While $k \leq p-1$, advance the ensemble members in time by applying the model M_{k+1} and sampling the model error:

$$x^l_{k+1|k} = M_{k+1} x^l_{k|k} + m_{k+1} + v^l_{k+1},\ v^l_{k+1} \sim N(0, Q_{k+1}), \tag{2.33}$$

(2.5), and the independence of the errors yield:

$$\begin{aligned}
\mathbb{P}(x_{0:p}|y_{0:p}) &= \mathbb{P}(x_0)\,\mathbb{P}(y_0|x_0)\prod_{k=1}^{p}\mathbb{P}(x_k|x_{k-1}, y_k) \\
&\propto \mathbb{P}(x_0)\prod_{k=0}^{p}\mathbb{P}(y_k|x_k)\prod_{k=1}^{p}\mathbb{P}(x_k|x_{k-1}) \\
&\propto \underbrace{e^{-\frac{1}{2}\|x_0 - x_b\|^2_{B^{-1}}}}_{\propto\ \mathbb{P}(x_0)}\prod_{k=0}^{p}\underbrace{e^{-\frac{1}{2}\|\mathcal{H}_k(x_k) - y_k\|^2_{R_k^{-1}}}}_{\propto\ \mathbb{P}\left(y_k|x_k\right)}\prod_{k=1}^{p}\underbrace{e^{-\frac{1}{2}\|x_k - \mathcal{M}_k(x_{k-1}) - m_k\|^2_{Q_k^{-1}}}}_{\propto\ \mathbb{P}(x_k|x_{k-1})} \\
&\propto e^{-\frac{1}{2}\|x_0 - x_b\|^2_{B^{-1}} - \frac{1}{2}\sum_{k=0}^{p}\|\mathcal{H}_k(x_k) - y_k\|^2_{R_k^{-1}} - \frac{1}{2}\sum_{k=1}^{p}\|x_k - \mathcal{M}_k(x_{k-1}) - m_k\|^2_{Q_k^{-1}}}.
\end{aligned}$$

Therefore, it can be easily seen that the MAP estimator is the solution of the following least squares problem:

$$\min_{x_0,\dots,x_p\in\mathbf{R}^n} \frac{1}{2}\left(\|x_0-x_b\|^2_{B^{-1}}+\sum_{k=0}^{p}\|\mathcal{H}_k(x_k)-y_k\|^2_{R_k^{-1}}+\sum_{k=1}^{p}\|x_k-\mathcal{M}_k(x_{k-1})-m_k\|^2_{Q_k^{-1}}\right). \tag{2.34}$$

The nonlinear least squares problem (2.34) is known as weak-constraint four dimensional variational problem (4DVAR). Originally in 4DVAR, $x_k=\mathcal{M}_k\left(x_{k-1}\right)$ i.e., the model $\mathcal{M}_k$ is supposed to be perfect; in this case (2.34) becomes:

$$\begin{aligned}\min_{x_{0:p}} \quad & \frac{1}{2}\left(\|x_0-x_b\|^2_{B^{-1}}+\sum_{k=0}^{p}\|\mathcal{H}_k(x_k)-y_k\|^2_{R_k^{-1}}\right)\\ & \text{subject to } x_k=\mathcal{M}_k(x_{k-1})\ \forall k=1,\dots,p.\end{aligned} \tag{2.35}$$

This latter problem is known as strong-constraint 4DVAR. In the case when $p=0$ (no evolution in time of the state), the problem (2.35) becomes:

$$\min_{x_0\in\mathbf{R}^n} \frac{1}{2}\left(\|x_0-x_b\|^2_{B^{-1}}+\|\mathcal{H}_0(x_0)-y_0\|^2_{R_0^{-1}}\right). \tag{2.36}$$

This problem is known as three dimensional variational problem (3DVAR).

Note that, since the distributions of the errors are Gaussian, hence in the linear case the maximum of $\mathbb{P}\left(x_{0:p}|y_{0:p}\right)$ coincide with its mean. In this case (the observations and model operators are linear, and the errors are Gaussian) the MAP estimator is equal to the KS mean:

$$E(x_{0:p}|y_{0:p})=\arg\min_{x_0,\dots,x_p} \frac{1}{2}\left(\|x_0-x_b\|^2_{B^{-1}}+\sum_{k=0}^{p}\|H_kx_k-y_k\|^2_{R_k^{-1}}+\sum_{k=1}^{p}\|x_k-M_kx_{k-1}-m_k\|^2_{Q_k^{-1}}\right). \tag{2.37}$$

2.3.2 The least squares problems

The main idea of least squares problem is to find the best model fit to the observed data in the sense that the sum of squared errors between the observed data and the model prediction is minimized. As in the case of MAP estimator, the aim is to find an estimate of the true state vector x that minimizes the sum of squares of errors (residuals). Therefore the least squares method seeks the solution by solving an optimization problem of the following form:

$$\min_{x\in\mathbf{R}^n} f(x)=\frac{1}{2}\|F(x)\|^2, \tag{2.38}$$

where $F:\mathbf{R}^n\to\mathbf{R}^q$ is the residual function. Usually the squares of the residuals are weighted by the inverse of the corresponding covariance matrices [112].

In the case of the system of equations (2.17)-(2.18)-(2.19), the residuals are:

$$\begin{aligned} v_0 &= x_0 - x_b, \\ v_k &= x_k - \mathcal{M}_k(x_{k-1}) - m_k, \text{ for } k = 1, \dots p, \\ w_k &= y_k - \mathcal{H}_k(x_k), \text{ for } k = 0, \dots p. \end{aligned}$$

For simplicity reasons from now on, unless we mention the contrary, it is assumed that the model $\mathcal{M}_k$ is perfect i.e., the residual $v_k = 0$, and we assume also that $m_k = 0$. The reader is invited to look for the case when this hypothesis alleviated in the Appendix A. Minimizing the sum of the squares of the residual vectors, weighted by the inverse of the corresponding covariance matrices, leads to the following nonlinear least squares problem:

$$\begin{aligned} \min_{x_{0:p}} \quad & \frac{1}{2}\left(\|x_0 - x_b\|^2_{B^{-1}} + \sum_{k=0}^{p} \|\mathcal{H}_k(x_k) - y_k\|^2_{R_k^{-1}} \right) \\ & \text{subject to } x_k = \mathcal{M}_k(x_{k-1}) \ \forall k = 1, \dots, p. \end{aligned} \tag{2.39}$$

This problem is the same as the problem (2.35) (strong-constraint 4DVAR). For convenience and simplicity, we will re-write the latter optimization problem as a non constraint problem and in a compact way. From $x_k = \mathcal{M}_k(x_{k-1})$ we obtain that:

$$x_k = \mathcal{M}_k o \mathcal{M}_{k-1} o \dots o \mathcal{M}_1(x_0) = \mathcal{M}_{k \leftarrow 0}(x_0),$$

where o denotes the composition operator, and $\mathcal{M}_{k \leftarrow 0}$ is the composition function of $\mathcal{M}_k, \dots, \mathcal{M}_1$. By using this notation, let us define:

$$y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_p \end{pmatrix}, \quad R = \begin{pmatrix} R_0 & 0_m & \dots & 0_m \\ 0_m & R_1 & 0_m & \dots \\ \vdots & 0_m & \ddots & 0_m \\ 0_m & \dots & 0_m & R_p \end{pmatrix}, \text{ and}$$

$$\mathcal{H}(x_0) = \begin{pmatrix} \mathcal{H}_0(x_0) \\ \mathcal{H}_1(\mathcal{M}_{1 \leftarrow 0}(x_0)) \\ \vdots \\ \mathcal{H}_p(\mathcal{M}_{p \leftarrow 0}(x_0)) \end{pmatrix}.$$

By using these definitions, the optimization problem (2.39) can be rewritten as:

$$\min_{x_0} f(x_0) = \frac{1}{2}\|F(x_0)\|^2 = \frac{1}{2}\left(\|x_0 - x_b\|^2_{B^{-1}} + \|y - \mathcal{H}(x_0)\|^2_{R^{-1}} \right), \tag{2.40}$$

where the control variable is only x_0. When there is no confusion, the index of x_0 is dropped. The residual function for the problem (2.40) is:

$$F(x) = \begin{pmatrix} B^{-1/2}(x - x_b) \\ R^{-1/2}(\mathcal{H}(x) - y) \end{pmatrix}, \tag{2.41}$$

which is a function from $\mathbf{R}^n \to R^q$, where $q = m(p+1)+n$.

2.3.3 Solving linear least squares problems

In the case when the function F in (2.38) is linear, there is a matrix $A \in \mathbf{R}^{q\times n}$ and a vector $b \in \mathbf{R}^q$ such that $F(x) = Ax - b$. Hence, in this case the problem (2.38) can be defined as:

$$\min_{x\in\mathbf{R}^n} \|Ax - b\|^2. \tag{2.42}$$

Let us denote the solution of the latter problem by x^*.

There are two basic classes of methods to find x^*. The first class is called direct methods. They theoretically give an exact solution to the problem up to round-of errors [16]. There are several direct methods for the resolution of (2.42) based on the matrix $A^\top A$ or A decomposition [29] such as (i) Cholesky decomposition which is suitable to the cases where the matrix $A^\top A$ is definite, (ii) for general matrix A ones of the most used methods are the QR decomposition which is known to be numerically stable, and the truncated singular value decomposition, which is suitable for the ill-conditioned problems. In this dissertation, we present the truncated singular value decomposition method (TSVD) (see section 2.3.3.1). The second class is represented by fixed point methods; and sometime, called iterative methods, which construct a series of approximations for the solution that (under some assumptions) converges to the solution of the problem (2.42) [16, 75, 106, 118]. These methods do not need to store the matrix A, but they need only the action of A and/or $A^\top$ on vectors (their product with a given vector). This makes these methods attractive for problems where A and/or $A^\top$ are only available by their action on vectors. In this thesis, we only give a brief overview of fixed point methods (see section 2.3.3.2). Finally, there is some methods on the borderline between the two classes; for example, projection methods based on Krylov subspaces [10, 105]. These methods are, sometime, considered as a class of iterative methods [106]. In this dissertation we focus on the known conjugate gradient method (see section 2.3.3.3).

2.3.3.1 Singular value decomposition method (SVD)

The singular value decomposition (SVD) method is based on a factorization of the matrix A [77, pages 18-22] [16, page 15]. Let us, assume that $\operatorname{rank}(A) = r^* \leq n$, then there are two orthogonal matrices $U = [u_1, \ldots, u_q] \in \mathbf{R}^{q\times q}$ and $V = [v_1, \ldots, v_n] \in \mathbf{R}^{n\times n}$, and a matrix

$$\Sigma = \begin{pmatrix} \Sigma_n \\ 0_{q-n} \end{pmatrix} \in \mathbf{R}^{q\times n}, \text{ where } \Sigma_n = \operatorname{diag}(\sigma_1, \ldots, \sigma_n),$$

such that:

$$A = U\Sigma V^\top,$$

is the *full* singular value decomposition of A. The nonnegative numbers σ_k, $k = 1, ..., n$, are the singular values of A given in descending order as:

$$\begin{cases} \sigma_1 \geq \ldots \geq \sigma_{r^*} > \sigma_{r^*+1} = \ldots \sigma_n = 0, \text{ if } r^* < n \\ \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0, \text{ if } r^* = n. \end{cases}$$

The column vectors $u_1, ..., u_q$, and $v_1, ..., v_n$, are called the left and the right singular vectors of A, respectively.

If $r^* = n$, the least squares problem (2.42) has a unique solution:

$$x^* = V\Sigma_n^{-1}U_n^\top b,$$

where U_n is formed from the first n columns of U. If $r^* < n$, the least squares problem (2.42) may have several solutions. In this case, it has the minimum 2-norm solution:

$$x^* = V_{r^*}\Sigma_{r^*}^{-1}U_{r^*}^\top b,$$

where Σ_{r^*} is the diagonal matrix consisting of the first r^* singular values of A in descending order, and U_{r^*} and V_{r^*} are formed from the first r^* columns of U and V, respectively. In some applications (e.g., problems arising from the discretization of an ill-posed problem), a better solution, in the sense that it is less sensitive than the original one to errors in the data (A, b), is obtained by a truncated least squares solution [16, page 100-103] of the form:

$$x_r = V_r\Sigma_r^{-1}U_r^\top b,$$

for some $r < r^*$, and where V_r, Σ_r, and U_r are defined as before but with r replacing r^*.

2.3.3.2 Fixed point methods

In this section, the subscript k denotes the iteration when using a fixed point method to solve the linear problem (2.42). To solve this problem at each iteration $k > 0$ an approximate solution of x^* is recurrently sought as:

$$x_{k+1} = x_k - F(A^\top Ax_k - A^\top b), \tag{2.43}$$

[16, pages 269-286] where F is a prescribed matrix related to the matrix A. F should be an approximation of the matrix $(A^\top A)^{-1}$. Using the fact that $A^\top Ax^* = A^\top b$ (x^* is a solution of the corresponding normal equation of (2.42)), we found:

$$\begin{aligned} x_{k+1} &= x_k - F(A^\top Ax_k - A^\top Ax^*) \\ &= x_k - FA^\top A(x_k - x^*). \end{aligned}$$

Hence

$$x_{k+1} - x^* = \left(I - FA^\top A\right)(x_k - x^*). \tag{2.44}$$

Therefore if we choose the matrix F such that:

$$\|I - FA^\top A\| < a,$$

where a is some constant in the interval $[0, 1)$, then we will have:

$$\|x_k - x^*\| \leq a^k \|x_0 - x^*\|,$$

hence the sequence x_k will converges to x^*.

There are many ways to construct the matrix F, and this leads to different resolution algorithms [16, 75, 106, 125]. Note that for $F = (A^\top A)^{-1}$, the solution is reached in one single iteration ($x_1 = x^*$).

2.3.3.3 Conjugate gradient method

Solving the problem (2.42) is equivalent to solving the corresponding normal equation:

$$A^\top Ax = A^\top b. \tag{2.45}$$

Krylov subspace methods have become a very useful tool for solving a linear equations of the form (2.45). These methods search for an approximate solution for a linear system (2.45) in a subspace $x_0 + K_l$ where x_0 is the initial guess, and K_l is the Krylov subspace defined as follows:

$$K_l = span\{r_0, A^\top Ar_0, \ldots, (A^\top A)^{l-1} r_0\},$$

where $r_0 = A^\top b - A^\top Ax_0$ and $l \in \mathbf{N}^*$. The subspace K_l is of dimension at most l. Moreover these methods seek an approximation by imposing the condition:

$$r_l = A^\top b - A^\top Ax_l \perp L_l,$$

where L_l is a subspace of dimension l. The different versions of Krylov subspace methods arise from different choices of the subspace L_l and from the ways in which the system is preconditioned. When the matrix $A^\top A$ is definite positive one of the most prominent Krylov method for solving the linear systems of the form (2.45) is the so called conjugate gradient method. It was originally proposed by [60]. The CG method converges in at most n iterations in exact arithmetic. CG method seeks x^* the solution of the linear system (2.45) by minimizing the following quadratic function:

$$\phi(x) = \frac{1}{2} x^\top A^\top Ax - x^\top A^\top b, \tag{2.46}$$

since $\nabla\phi(x) = A^\top Ax - A^\top b$ and $\nabla^2\phi(x) = A^\top A$ is symmetric positive definite, hence the solution x^* of the linear system (2.45) is equal to:

$$\arg \min_{x \in \mathbf{R}^n} \phi(x).$$

The CG method is a line search method with a special choice of directions. Given a current step approximation x_k to the minimum x^* and a direction p_k, then CG seeks $x_{k+1} = x_k + \alpha_k p_k$, where

$$\alpha_k = \arg\min_{\alpha \in \mathbf{R}} \phi(x_k + \alpha p_k).$$

It is easy to show that:

$$\alpha_k = \frac{p_k^\top r_k}{p_k^\top A^\top A p_k}, \text{ where, } r_k = A^\top b - A^\top A x_k.$$

The directions p_k are chosen recursively as follows:

$$p_0 = r_0 = A^\top b - A^\top A x_0 = -\nabla\phi(x_0).$$

$$p_k = r_k + \beta_k p_{k-1},$$

where β_k is chosen such that p_k and the previous directions are conjugate with respect to $A^\top A$, i.e.,

$$p_k^\top A^\top A p_i = 0, \forall i \leq k-1. \tag{2.47}$$

β_k that satisfies the property given in (2.47) can be given as:

$$\beta_k = \frac{r_k^\top r_k}{r_{k-1}^\top r_{k-1}} = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}, \forall k \geq 1.$$

Algorithm 2.5 gives the pseudo-code for the CG method.

Algorithm 2.5: Conjugate gradient algorithm

Initialization

Select x_0, the initial guess.

Compute $r_0 = A^\top b - A^\top A x_0$, $\rho_0 = r_0^\top r_0$, $p_0 = r_0$

For $k = 1, 2, \ldots, n$**:**

1. $q_k = A^\top A p_k$
2. $\alpha_k = \frac{\rho_k}{q_k^\top p_k}$
3. $x_{k+1} = x_k + \alpha_k p_k$
4. $r_{k+1} = r_k - \alpha_k q_k$
5. $\rho_{k+1} = r_{k+1}^\top r_{k+1}$
6. $\beta_{k+1} = \frac{\rho_{k+1}}{\rho_k}$
7. $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$

2.3.4 Solving nonlinear least squares problems

The nonlinear least squares are typically solved by the well-known line-search [33, p. 227] and trust-region strategies [25]. These methods are based on the Newton and quasi-Newton approaches with modifications that consider the special structure of the objective function f and of its derivatives [93, p. 247].

In this thesis we only present methods for solving the problem of finding a local minimizer for the function f. Several methods for nonlinear optimization are iterative: from a starting point x^0 the method produces a series of vectors x^1, x^2,... which converges to a local minimizer for the given function. In the case of several minimizers the result will depend on the starting point.

For each method, one step from the current iterate consists in finding a descent direction, and a step length giving the amount of the function decreasing. In this thesis we will present only some of this methods, especially those which are suitable to solve the nonlinear least squares problems. We will define each method in terms of the transition from a current iteration x^j to a new one x^{j+1}.

Before giving details of these methods, we first calculate the first and second order derivatives of the objective function in (2.40) which are needed by solution methods.

2.3.4.1 Computation of the derivatives

The optimization problem (2.40) can be viewed as a special case of an unconstrained optimization problem. It requires for its solution the computation of the values of objective function f and sometimes of its derivatives, in particular its gradient $\nabla f(x)$ (the first derivative) and its Hessian $\nabla^2 f(x)$ (the second derivative). We start this section by the computation of the first and second order derivatives of the objective function f defined in the problem (2.38), as a function of the vector function F and its derivatives. Then we compute explicitly the function in the least squares problem (2.40) derivatives.

Let us start with the gradient of f which is given by:

$$\nabla f(x) = J_F^\top(x) F(x),$$

where $J_F(x) \in \mathbf{R}^{q \times n}$ is the Jacobian of the function F on x defined as:

$$J_F(x) = \begin{pmatrix} \frac{\delta F_1(x)}{\delta x_1} & \cdots & \frac{\delta F_1(x)}{\delta x_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta F_q(x)}{\delta x_1} & \cdots & \frac{\delta F_q(x)}{\delta x_n} \end{pmatrix} = \begin{pmatrix} \nabla F_1(x)^\top \\ \vdots \\ \nabla F_q(x)^\top \end{pmatrix}.$$

Hence

$$\nabla f(x) = \sum_{i=1}^{q} F_i(x) \nabla F_i(x). \tag{2.48}$$

The Jacobian of the function defined in (2.41) is:

$$J_F(x) = \begin{pmatrix} B^{-1/2} \\ R^{-1/2}\mathcal{H}'(x) \end{pmatrix}. \tag{2.49}$$

Therefore the gradient of the objective function defined in (2.40) is

$$\nabla f(x) = \left(B^{-1}(x - x_b) + \mathcal{H}'(x)^\top R^{-1}(\mathcal{H}(x) - y)\right). \tag{2.50}$$

Now, we compute the expression of the Hessian of the function f. From (2.48), we have $\nabla f(x) = \sum_{i=1}^{q} F_i(x)\nabla F_i(x)$, hence

$$\begin{aligned} \nabla^2 f(x) &= \sum_{i=1}^{q} \nabla F_i(x)\nabla F_i(x)^\top + F_i(x)\nabla^2 F_i(x) \\ &= J_F(x)^\top J_F(x) + \sum_{i=1}^{q} F_i(x)\nabla^2 F_i(x) \\ &= B^{-1} + \mathcal{H}'(x)^\top R^{-1}\mathcal{H}'(x) + S(x), \end{aligned} \tag{2.51}$$

where $S(x) = \sum_{i=1}^{q} F_i(x)\nabla^2 F_i(x)$. Note that, in the expression of $\nabla^2 f(x)$, only $S(x)$ is depending on the second derivative of the function F.

2.3.4.2 Newton method

The Newton method finds the roots of a given nonlinear equation [123]. We know from the first optimization necessary condition that the minimizer of the problem (2.40) is a solution of the equation:

$$\nabla f(x) = 0. \tag{2.52}$$

To find a solution of the equation (2.52), the Newton method solves at each iteration the following subproblem:

$$\nabla f(x^j) + \nabla^2 f(x^j)(x^{j+1} - x^j) = 0. \tag{2.53}$$

In the case when $\nabla^2 f(x^j)$ is positive definite, which is the case that we will assume here, the solution of the latter equation can be found by minimizing the quadratic function:

$$m(x^{j+1}) = f(x^j) + (x^{j+1} - x^j)^\top \nabla f(x^j) + \frac{1}{2}(x^{j+1} - x^j)^\top \nabla^2 f(x^j)(x^{j+1} - x^j). \tag{2.54}$$

The quadratic function m is the second order Taylor approximation of the function f in the neighborhood of the iterate x^j. Minimizing the model m (or equivalently solving the equation (2.53)) gives

$$x^{j+1} = x^j - (\nabla^2 f(x^j))^{-1}\nabla f(x^j). \tag{2.55}$$

Usually, x^{j+1} is not computed by inverting the matrix $\nabla^2 f(x^j)$. Rather, given x^j, $\nabla f(x^j)$ is computed and the linear equation:

$$\nabla^2 f(x^j)s = -\nabla f(x^j), \tag{2.56}$$

is solved for the step s^j. Then (2.55) simply says that $x^{j+1} = x^j + s^j$. In the case when $\nabla^2 f(x^j)$ is not definite positive, the Newton method alone may not work. Under the assumption that the Hessian of the function f is Lipchitz continuous in a neighborhood of a solution x^*, the Newton method works well, and converge quadratically when the starting point (x^0) is close enough to a local minimum [93].

For the nonlinear least squares (2.40), substituting the objective function, its gradient and its Hessian values in (2.54) and (2.55) leads to:

$$\begin{aligned} m(x^{j+1}) &= \frac{1}{2}\left(\|s_b^j\|^2_{B^{-1}} + \|d_j\|^2_{R^{-1}}\right) - (x^{j+1} - x^j)^\top B^{-1} s_b^j \\ &- (x^{j+1} - x^j)^\top \mathbf{H}_j R^{-1} d_j \\ &+ \frac{1}{2}(x^{j+1} - x^j)^\top (B^{-1} + \mathbf{H}_j{}^\top R^{-1}\mathbf{H}_j + S(x^j))(x^{j+1} - x^j). \\ x^{j+1} &= x^j + (B^{-1} + \mathbf{H}_j{}^\top R^{-1}\mathbf{H}_j + S(x^j))^{-1} B^{-1} s_b^j \\ &+ (B^{-1} + \mathbf{H}_j{}^\top R^{-1}\mathbf{H}_j + S(x^j))^{-1} \mathbf{H}_j^\top R^{-1} d_j, \end{aligned}$$

and the substitution in (2.56) gives:

$$(B^{-1} + \mathbf{H}_j{}^\top R^{-1}\mathbf{H}_j + S(x^j))s = B^{-1} s_b^j + \mathbf{H}_j{}^\top R^{-1} d_j, \tag{2.57}$$

where $\mathbf{H}_j = \mathcal{H}'(x^j)$ is the linear tangent of the operator $\mathcal{H}$ on x^j, $d_j = y - \mathcal{H}(x^j)$, and $s_b^j = x_b - x^j$ (this quantity is the background on the step at iteration j). The pseudo-code for the Newton method is given in Algorithm 2.6.

Algorithm 2.6: Newton algorithm

Initialization

Select x^0, the initial iterate.

For $j = 0, 1, 2, \ldots$

1. Solve the linear system (2.56), and let s^j denote such a solution.
2. Compute $x^{j+1} = x^j + s^j$.

2.3.4.3 Gauss-Newton method

The Gauss–Newton algorithm is the same as Newton one where the second-order term in $\nabla^2 f(x)$ (in equation 2.51) is discarded, i.e., the term $S(x)$ is set to zero. Therefore the Gauss-Newton step s^j is defined to be the solution of the following linear system:

$$J_F(x^j)^\top J_F(x^j) s \quad = \quad -\nabla f(x^j) = -J_F(x^j)^\top F(x^j) \tag{2.58}$$

and correspondingly, the Gauss–Newton iterate is $x^{j+1} = x^j + s^j$. One motivation for the Gauss-Newton approach is the fact that the term $S(x)$ vanishes for zero residual problems (the case

when the function F is equal to zero at the minimum) and therefore might be negligible for small residual problems.

Another interpretation of the Gauss-Newton method is that, it is an iterative procedure where at each point x^j, a step is computed as a solution of the linearized least squares subproblem:

$$\min_{s\in\mathbf{R}^n} \frac{1}{2}\|F(x^j) + J_F(x^j)s\|^2.$$

The subproblem has a unique solution if $J_F(x^j)$ has full column rank, (in this case this solution is equal to the solution of the linear system in (2.58)).

For the nonlinear least squares (2.40), by linearization we consider the following subproblem:

$$\min_{s\in\mathbf{R}^n} m(x^j + s) = \frac{1}{2}\left(\|s - s_b^j\|^2_{B^{-1}} + \|d_j - \mathbf{H}_j s\|^2_{R^{-1}}\right). \tag{2.59}$$

The gradient, and the Hessian of the function m defined on the previous subproblem are:

$$\nabla m(x^j + s) = B^{-1}(s - s_b^j) + \mathbf{H}_j^\top R^{-1}(\mathbf{H}_j s - d_j), \tag{2.60}$$

$$\nabla^2 m(x^j + s) = B^{-1} + \mathbf{H}_j^\top R^{-1}\mathbf{H}_j. \tag{2.61}$$

Since $\nabla^2 m(x^j + s)$ is definite positive (because B^{-1} is definite positive and $\mathbf{H}_j^\top R^{-1}\mathbf{H}_j$ is semi definite positive), the solution of the equation $\nabla m(x^j + s) = 0$ is a solution of the subproblem (2.59), and is equal to:

$$s^j = (B^{-1} + \mathbf{H}_j^\top R^{-1}\mathbf{H}_j)^{-1}\left(B^{-1}(x_b - x^j) + \mathbf{H}_j^\top R^{-1} d_j\right). \tag{2.62}$$

Once again to compute s^j, usually we solve the following equation:

$$(B^{-1} + \mathbf{H}_j^\top R^{-1}\mathbf{H}_j)s = \left(B^{-1}s_b^j + \mathbf{H}_j^\top R^{-1} d_j\right). \tag{2.63}$$

Solving the latter problem using the conjugate gradient method is known as the *primal approach*. For some problems, like those solved daily in weather prediction systems, the state dimension n is larger then the observations dimension m, typically $n \sim 10^7$ and $m \sim 10^5$ [19]. In this case, a significant reduction in the computational cost is possible by rewriting the problem (2.63) in the m-dimentional space related to the observations as follows: From (2.13) we have:

$$(B^{-1} + \mathbf{H}_j^\top R^{-1}\mathbf{H}_j)^{-1} = (I - K\mathbf{H}_j)B,$$

where $K = B\mathbf{H}_j^\top(R + \mathbf{H}_j B\mathbf{H}_j^\top)^{-1}$ hence

$$s^j = (I - K\mathbf{H}_j)B\left(B^{-1}s_b^j + \mathbf{H}_j^\top R^{-1} d_j\right) \tag{2.64}$$

$$= (I - K\mathbf{H}_j)s_b^j + (I - K\mathbf{H}_j)B\mathbf{H}_j^\top R^{-1} d_j, \tag{2.65}$$

from (2.16) we have $(I - K\mathbf{H}_j)B\mathbf{H}_j^\top R^{-1} = K$, hence

$$s^j = s_b^j + K\left(d_j - \mathbf{H}_j s_b^j\right).$$

We can rewrite the latter equality as:

$$s^j = s_b^j + B\mathbf{H}_j^\top w^j, \tag{2.66}$$

where w^j is the solution of:

$$(R + \mathbf{H}_j B \mathbf{H}_j^\top) w^j = (d_j - \mathbf{H}_j s_b^j). \tag{2.67}$$

Note that the dimension of the vector w^j is m. Solving the problem (2.67) using conjugate gradient method and then retrieve s^j using the equation (2.66), is the so-called *dual approach* [2, 109]. In practice one of the most important methods to solve the dual problem is the RPCG method [56].

Algorithm 2.7: Gauss-Newton algorithm

Initialization

Select x^0, the initial iterate.

For $j = 0, 1, 2, \ldots$

1. Solve the linear system (2.63), and let s^j denote such a solution.
2. Compute $x^{j+1} = x^j + s^j$.

2.3.4.4 Globalization methods

The algorithms discussed above are locally convergent, in the sense that the convergence holds only if the starting point is near to a local minimum, [93]. The algorithms may fail when the initial iterate is not near the minimum. The reasons for this failure, are that (i) the directions, sometimes are not a descent directions for the function f and that even when a search direction is a direction of decrease of f, (ii) the length of the step is not controlled and can be too long or too small over iterations. Hence, taking a Newton, or Gauss-Newton step can lead to an increase in the function which causes the divergence of the iterations. The globalization methods address this problem, and ensure the convergence to a stationary point of the considered problem, independently from the starting point. This is done by controlling the length of the step at each iteration. Note that, these methods are not algorithms for global optimization (to find the global minimum). When these algorithms are applied to problems with many local minima, the results of the iteration (the local minimum found) may depend in the starting point.

Damped Gauss-Newton method

The damped Gauss-Newton method [16, p. 343], [33, p. 227] uses a line search strategy in the Gauss-Newton method. It can be shown that whenever the Jacobian of F, $J_F(x^j)$ has full

column rank and the gradient of f, $\nabla f(x^j)$ is nonzero, the Gauss-Newton step s^j is in a descent direction [93, p. 254] for the objective function f.

A line search strategy in a Gauss-Newton method leads to the damped Gauss-Newton method which generates the iterates as follows:

$$x^{j+1} = x^j + \alpha_j s^j$$

where $\alpha_j > 0$ is the step length, the optimal α_j is the one which verifies the solution of:

$$\min_{\alpha \in \mathbf{R}^+} f(x^j + \alpha s^j) = \frac{1}{2}\|F(x^j + \alpha s^j)\|^2. \tag{2.68}$$

For some functions, it may be expensive to find the solution of the latter problem. In this situations an inexact line search method is used, and the length step is asked to verifies only some conditions. For more details on the line search strategy we refer to [93, Chapter 3] [16, p. 344-346], and [33, p. 116-129].

Under suitable assumptions, the damped Gauss-Newton method is convergent even on large-residual problems or highly nonlinear problems [16, 33].

Levenberg-Marquardt method

The Levenberg-Marquardt algorithm [79, 88, 92, 94] is a regularization of the Gauss-Newton method. A regularization parameter is updated at every iteration and indirectly controls the size of the step, making Gauss-Newton globally convergent.

The Levenberg-Marquardt method was developed especially to deal with the rank deficiency of $J_F(x^j)$ and to provide a globalization strategy for Gauss-Newton. It solves at each iteration a subproblem of the following form:

$$\begin{aligned} \min_{s \in \mathbf{R}^n} \; m_j(x^j + s) &= \frac{1}{2}\|F(x^j) + J_F(x^j)s\|^2 + \frac{1}{2}\gamma_j^2\|s\|^2, \\ &= \frac{1}{2}\left\| \begin{pmatrix} F(x^j) \\ 0 \end{pmatrix} + \begin{pmatrix} J_F(x^j) \\ \gamma_j I \end{pmatrix} s \right\|^2, \end{aligned} \tag{2.69}$$

where γ_j is an appropriately chosen regularization parameter. Several strategies were developed to update γ_j. The latter subproblem, is the same as the Gauss-Newton subproblem for which we add a regularization term $\frac{1}{2}\gamma_j^2\|s\|^2$. This term controls the direction of minimization and the step length.

$$\begin{aligned} \nabla m_j(x^j + s) &= (J_F(x^j)^\top J_F(x^j) + \gamma_j^2 I)s + J_F(x^j)^\top F(x^j), \\ \nabla^2 m_j(x^j + s) &= J_F(x^j)^\top J_F(x^j) + \gamma_j^2 I. \end{aligned}$$

If $\gamma_j > 0$ then $\nabla^2 m_j(x^j + s)$ is definite positive, hence the unique solution of the subproblem (2.69) is the solution of:

$$(J_F(x^j)^\top J_F(x^j) + \gamma_j^2 I)s = -J_F(x^j)^\top F(x^j), \tag{2.70}$$

The Levenberg-Marquardt method can be seen as precursor of the trust-region method in the sense that it seeks to determine when the Gauss-Newton step is applicable (in which case the regularization parameter is set to zero) or when it should be replaced by a slower but safer gradient or steepest descent step (corresponding to a sufficiently large regularization parameter). The comparison with trust regions can also be drawn by looking at the square of the regularization parameter as the Lagrange multiplier of a trust-region subproblem of the form $\min_{s\in\mathbf{R}^n}\ (1/2)\|F(x^j) + J_F(x^j)s\|^2$ s.t. $\|s\| \leq \delta_j$, and in fact it was suggested by [91] to update the regularization parameter γ_j similarly to trust-region radius δ_j. For this purpose, one considers the ratio between the actual reduction $f(x^j) - f(x^j + s^j)$ attained in the objective function and the reduction $m_j(x^j) - m_j(x^j + s^j)$ predicted by the model, given by:

$$\rho_j \ = \ \frac{f(x^j) - f(x^j + s^j)}{m_j(x^j) - m_j(x^j + s^j)}.$$

Then, if ρ_j is larger than a given small constant, the step is accepted and γ_j is possibly decreased (corresponding to 'δ_j is possibly increased'). Otherwise the step is rejected and γ_j is increased (corresponding to 'δ_j is decreased'). Algorithm 2.8 gives the pseudo-code of the Levenberg-Marquardt method.

Algorithm 2.8: Levenberg-Marquardt algorithm

Initialization

Choose the constants $\eta_1 \in (0, 1)$, $\gamma_{\min} > 0$, and $\lambda > 1$. Select x^0 and $\gamma_0 \geq \gamma_{\min}$.

For $j = 0, 1, 2, \ldots$

1. Solve (or approximately solve) (2.69), and let s^j denote such a solution.
2. Compute $\rho_j = \frac{f(x^j) - f(x^j + s^j)}{m_j(x^j) - m_j(x^j + s^j)}$.

   If $\rho_j \geq \eta_1$, then set $x^{j+1} = x^j + s^j$ and $\gamma_{j+1} = \max(\gamma_j, \gamma_{\min})$.

   Otherwise, set $x^{j+1} = x^j$ and $\gamma_{j+1} = \lambda\gamma_j$.

This algorithm enables a global convergence disregarding the starting point of the iterations, at the cost of one more function evaluation per iteration. This algorithm still globally convergent when we maintain the regularization parameter γ fix over iterations and large enough. In this latter case, Algorithm 2.8 becomes Algorithm 2.9.

Algorithm 2.9: Levenberg-Marquardt algorithm with fixed regularization

Initialization

Choose the constants $\eta_1 \in (0,1)$, and $\gamma > 0$. Select x^0.

For $j = 0, 1, 2, \ldots$

1. Solve (or approximately solve) (2.69), and let s^j denote such a solution.
2. Compute $x^{j+1} = x^j + s^j$.

In the case when γ varies over iterations, by increasing it for unsuccessful iterations and decreasing it for successful iterations (Algorithm 2.8), we refer for the proof of global convergence to [94, Corollary 2.1 p.7]. In the case when γ is maintained constant over iterations i.e., the case of Algorithm 2.9, we still have the global convergence for γ large enough. In the following theorem we give the proof of Algorithm 2.9 global convergence.

Theorem 2.1. *Under the assumption that the function f Hessian is bounded, i.e., it exist κ_H such that:*

$$\|\nabla^2 f(x)\| \leq \kappa_H, \forall x \in \mathbf{R}^n,$$

then any finite limit point x^* of the sequence (x^j) generated by Algorithm 2.8 is a stationary value of f.

Proof. The proof still the same as in [94], the only change is the proof of [94, Theorem 2.2 p.5]. To prove the latter theorem, it is enough to show that there exists γ^* such that for any $\gamma \geq \gamma^*$, $\rho_j \geq \eta$ with $0 < \eta < 1$.

Let

$$J_F(x^j) = U(x^j)\Sigma(x^j)V(x^j)^\top$$

be the singular value decomposition of the function F Jacobian on x^j,

$$\Sigma(x^j) = \text{diag}(\sigma_1(x^j), ..., \sigma_n(x^j))$$

where $\sigma_1(x^j) \geq \sigma_2(x^j) ... \geq \sigma_n(x^j)$ are the singular values of $J_F(x^j)$. For the proof we will assume in addition that there exist $\sigma > 0$ and $\epsilon > 0$ such that $\forall\ x^j,\ \sigma_1(x^j) \leq \sigma$, and $\sigma_n(x^j) \geq \epsilon$.

For the following we omit the parameter x^j, the index j of m_j and the index F of J_F. s^* is a solution of:

$$(J^\top J + \gamma^2 I)s = -J^\top F.$$

We have

$$\begin{aligned} m(x + s^*) - f(x) &= \frac{1}{2} {s^*}^\top (J^\top J + \gamma^2 I)s^* + {s^*}^\top J^\top F \\ &= -\frac{1}{2} {s^*}^\top J^\top F + {s^*}^\top J^\top F = \frac{1}{2} {s^*}^\top J^\top F, \end{aligned}$$

and from Taylor expansion:

$$f(x+s^*) = f(x) + {s^*}^\top J^\top F + {s^*}^\top H_f s^*,$$

where H_f is the function f Hessian on some point $\bar{s}$, hence

$$\begin{aligned} 1-\frac{\rho}{2} &= \frac{f(x)+f(x+s^*)-2m(x+s^*)}{2(f(x)-m(x+s^*))} \\ &= \frac{2f(x)+{s^*}^\top J^\top F+\frac{1}{2}{s^*}^\top H_f s^*}{-{s^*}^\top J^\top F} \\ &= \frac{\frac{1}{2}{s^*}^\top H_f s^*}{-{s^*}^\top J^\top F}, \end{aligned}$$

therefore

$$\begin{aligned} \left|1-\frac{\rho}{2}\right| &\leq -\frac{\kappa_H\|s^*\|^2}{{s^*}^\top J^\top F} \\ &= \frac{\kappa_H\|V\Sigma^2V^\top+\gamma^2I)^{-1}V\Sigma U^\top F\|^2}{F^\top U\Sigma V^\top(V\Sigma^2V^\top+\gamma^2I)^{-1}V\Sigma U^\top F} \\ &\leq \frac{\kappa_H\left(\max\limits_{i\in\{1,\dots,n\}}\left(\frac{\sigma_i}{\sigma_i^2+\gamma^2}\right)\right)^2}{\min\limits_{i\in\{1,\dots,n\}}\left(\frac{\sigma_i^2}{\sigma_i^2+\gamma^2}\right)}, \end{aligned}$$

The two functions $x \to \frac{x^2}{x^2+\gamma^2}$, and $x \to \frac{x}{x^2+\gamma^2}$ are increasing on the domain $[0,\gamma]$. By taking $\gamma \geq \sigma$, we have:

$$\left|1-\frac{\rho}{2}\right| \leq \frac{\kappa_H\left(\frac{\sigma_1}{\sigma_1^2+\gamma^2}\right)^2}{\frac{\sigma_n^2}{\sigma_n^2+\gamma^2}} \tag{2.71}$$

$$\leq \frac{\kappa_H(\frac{\sigma}{\sigma^2+\gamma^2})^2}{\frac{\epsilon^2}{\epsilon^2+\gamma^2}}. \tag{2.72}$$

From (2.72) follows that if:

$$\kappa_H\left(\frac{\sigma}{\sigma^2+\gamma^2}\right)^2 \leq \left(1-\frac{\eta}{2}\right)\frac{\epsilon^2}{\epsilon^2+\gamma^2} \tag{2.73}$$

then we have $|1-\frac{\rho}{2}| \leq (1-\frac{\eta}{2})$ which implies that $\rho \geq \eta$. Furthermore (2.73) is equivalent to

$$0 \leq \gamma^4 + \left(2\sigma^2 - \frac{\sigma^2\kappa_H}{\epsilon^2\left(1-\frac{\eta}{2}\right)}\right)\gamma^2 + \sigma^4 - \frac{\sigma^2\kappa_H}{\left(1-\frac{\eta}{2}\right)}.$$

Defining

$$\Delta = \left(2\sigma^2 - \frac{\sigma^2\kappa_H}{\epsilon^2\left(1-\frac{\eta}{2}\right)}\right)^2 - 4\left(\sigma^4 - \frac{\sigma^2\kappa_H}{\left(1-\frac{\eta}{2}\right)}\right),$$

if $\Delta < 0$, then (2.73) holds for any $\gamma \geq \gamma^* = \sigma$, if on the contrary $\Delta \geq 0$, then (2.73) holds for any

$$\gamma \geq \gamma^* = \max \left(\sigma, \sqrt{\frac{\left| 2\sigma^2 - \frac{\sigma^2 \kappa_H}{\epsilon^2 \left(1 - \frac{\eta}{2}\right)} \right| + \sqrt{\Delta}}{2}} \right).$$

□

In this chapter, we tried to give an overview about some sequential methods for estimation theory, and some methods for solving least squares problems. A detailed description of these methods can also be found in [16, 33, 41, 42]. This chapter was introduced as background material to what comes next regarding our main contributions. The next chapter will detail our main first contribution where we study the sensitivity of the TSVD method to perturbations in the data.

Chapter 3

Sensitivity of the truncated singular value decomposition method

Perturbation analysis is the study of the sensitivity of the solution of a given problem to perturbations in the data. The concept of condition number allows one to assess the sensitivity of the solution. Sensitivity and conditioning theory has been applied to many fundamental problems of linear algebra, such as linear systems, linear least squares, or eigenvalue problems [16, 52, 55, 61, 110]. In this chapter, we extend the approach to the truncated singular value decomposition (TSVD) solution to linear least squares problems.

As presented in Section 2.3.3.1, the minimum 2-norm solution of the liner least squares problem (2.42) is $x^* = V_{r^*}\Sigma_{r^*}^{-1}U_{r^*}^\top b$, where $r^* = \text{rank}(A) \leq n$, Σ_{r^*} is the diagonal matrix consisting of the first r^* singular values of A in descending order, and U_{r^*} and V_{r^*} are formed from the first r^* columns of U and V, respectively. A better solution, in the sense that it is less sensitive than the original one to errors in the data (A, b), is obtained by a truncated least squares solution of the form:

$$x_r = V_r\Sigma_r^{-1}U_r^\top b, \tag{3.1}$$

for some $r < r^*$, and where V_r, Σ_r, and U_r are defined as before but with r replacing r^*. It turns out that, if $\hat{U}_r$ and $\hat{V}_r$ are any orthonormal bases for range (U_r) and range (V_r), then:

$$x_r = \hat{V}_r(\hat{U}_r^\top A\hat{V}_r)^{-1}\hat{U}_r^\top b.$$

Now let A and b be perturbed to yield $\tilde{A} = A + E$ and $\tilde{b} = b + f$, and let $\tilde{U}_r$ and $\tilde{V}_r$ form a pair of bases for the left and right singular subspaces associated with the r first singular values of $\tilde{A}$. The corresponding truncated least squares solution of the perturbed problem is then:

$$\tilde{x}_r = \tilde{V}_r(\tilde{U}_r^\top \tilde{A}\tilde{V}_r)^{-1}\tilde{U}_r^\top b. \tag{3.2}$$

Now it turns out that if the Fréchet derivative, x_r', of the function x_r exists then we have:

$$\tilde{x}_r = x_r + x_r'.(E, f) + \mathrm{o}(\|(E, f)\|).$$

Here, $x_r'.(E, f)$ is the application of a linear operator to (E, f). Given a norm on (E, f), call it $\|.\|_{(\alpha,\beta)}$, the *condition number* of x_r is defined to be the operator norm:

$$\|\|x_r'\|\|_{(\alpha,\beta),2} = \max_{[\alpha E,\, \beta f]\neq 0} \frac{\|x_r' \cdot (E, f)\|_2}{\|[E,\ f]\|_{(\alpha,\beta)}}.$$

The particular norm we use in this chapter is defined by:

$$\|(E, f)\|_{(\alpha,\beta)} = \sqrt{\alpha^2\|E\|_F^2 + \beta^2\|f\|_F^2},$$

where $\|.\|_F$ is the usual Frobenius norm and $\alpha \in]0, +\infty[$, $\beta \in]0, +\infty[$. Note that the purpose of the norm $\|.\|_{(\alpha,\beta)}$ is to tag the contributions of perturbations of A and b in the condition number, see [54].

The purpose of this chapter is to exhibit the square of the condition number of x_r as the 2-norm of a symmetric nonnegative matrix Δ that can be formed from the singular values of A, and the Fourier coefficients given by the entries of $U^\top b$. This chapter is organized as follows. In Section 3.1, we state preliminary results based on results from [111]. Section 3.2 is devoted to an expression for the first-order expansion of x_r with respect to the data (A, b). The main result of this section is the matrix representation for the corresponding Fréchet derivative leading to the formula for the condition number of x_r using the singular values of A and the Fourier coefficients of b. We give the upper and lower bounds of this quantity and perform some numerical tests to validate our analysis by comparing it with results of a finite difference approach in Section 3.3.

3.1 Preliminary results

It will be worthwhile to define the following matrix partitions:

$$V = [V_r, V_\perp] \in \mathbf{R}^{n\times n}, \quad U = [U_r, U_\perp] \in \mathbf{R}^{q\times q}, \quad \Sigma = \begin{bmatrix} \Sigma_r & \\ & \Sigma_\perp \end{bmatrix} \in \mathbf{R}^{q\times n},$$

where

$$\begin{aligned} V_r &\in \mathbf{R}^{n\times r}, \qquad V_\perp \in \mathbf{R}^{n\times(n-r)}, \qquad U_r \in \mathbf{R}^{q\times r}, \qquad U_\perp \in \mathbf{R}^{q\times(q-r)}, \\ \Sigma_r &= \mathrm{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbf{R}^{r\times r}, \quad \Sigma_\perp = \begin{bmatrix} \mathrm{diag}(\sigma_{r+1}, \ldots, \sigma_n) \\ 0 \end{bmatrix} \in \mathbf{R}^{(q-r)\times(n-r)}. \end{aligned}$$

Furthermore, we define matrices $E_{rr} = U_r^\top E V_r$, $E_{r\perp} = U_r^\top E V_\perp$, $E_{\perp r} = U_\perp^\top E V_r$, and $E_{\perp\perp} = U_\perp^\top E V_\perp$, and vectors $b_r = U_r^\top b$, $b_\perp = U_\perp^\top b$, and $f_r = U_r^\top f$. Finally, we shall denote by I_r, I_{q-r} and I_{n-r} the identity matrices of order r, $q-r$, and $n-r$, respectively.

The operator $\mathrm{vec}\,(\cdot)$ and the Kronecker product $\otimes$ will be of a particular importance in the sequel. The $\mathrm{vec}\,(\cdot)$ operator stacks the columns of the matrix argument into one long vector. For any matrices B and C, the matrix $B \otimes C = (b_{ij}C)$. It is enough for our purpose to recall the following properties concerning these operators[1]. For any matrices B, X and C having compatible dimensions with respect to the involved products, we have:

$$\mathrm{vec}\,(BXC) = (C^\top \otimes B)\,\mathrm{vec}(X), \tag{3.3}$$

$$\mathrm{vec}(X^\top) = \Psi_{(q,n)}\mathrm{vec}(X), \quad \text{for all} \quad X \in \mathbf{R}^{q\times n}, \tag{3.4}$$

where $\Psi_{(q,n)} \in \mathbf{R}^{qn\times qn}$ is the permutation matrix defined by:

$$\Psi_{(q,n)} = \sum_{i=1}^{q}\sum_{j=1}^{n} L_{ij} \otimes L_{ij}^\top.$$

Here each $L_{ij} \in \mathbf{R}^{q\times n}$ has entry 1 in position (i,j) and all other entries are zero.

The following assumption will be of a particular importance in what follows.

Assumption 3.1.1. Let

$$\gamma = \|(E_{\perp r}^\top, E_{r\perp})\|_F.$$

suppose that:

$$\delta = |\sigma_r - \sigma_{r+1}| - \|E_{rr}\|_2 - \|E_{\perp\perp}\|_2 > 0,$$

and assume that:

$$\gamma/\delta < \quad 1/2.$$

Roughly speaking, the statement of Assumption 3.1.1 is that the existence of a gap between σ_r and $\sigma_{r+1} > 0$ is required and that $\|E\|_2$ must be small enough compared to this gap.

Now, we state and adapt results from [111] to our context in the following two theorems.

Theorem 3.1. *[111, Theorem 6.4]. Let an $q \times n$ perturbation matrix E be given and partition $U^\top EV$ with respect to $U = [U_r, U_\perp]$ and $V = [V_r, V_\perp]$ in the form:*

$$U^\top EV = \begin{pmatrix} E_{rr} & E_{r\perp} \\ E_{\perp r} & E_{\perp\perp} \end{pmatrix}.$$

Then under Assumption 3.1.1, there are matrices $Q \in \mathbf{R}^{(q-r)\times r}$ and $P \in \mathbf{R}^{(n-r)\times r}$ satisfying:

$$\|(Q^\top, P^\top)\|_F < 2\frac{\gamma}{\delta} < 1$$

such that: range$(V_r + V_\perp P)$ and range$(U_r - U_\perp Q)$ form a pair of singular subspaces for $\tilde{A} = A + E$.

[1] We refer to [64, Chapter 4] for further properties of these operators.

Among other things, the theorem above tells us that Q and P approach 0 as E approaches 0. Other useful results related to the ones above are given in the following theorem (See again [111] and [110, p. 266]).

Theorem 3.2. *Suppose Assumption 3.1.1 holds. Then there exist matrices $Q \in \mathbf{R}^{(q-r)\times r}$ and $P \in \mathbf{R}^{(n-r)\times r}$ such that:*

$$\tilde{U}_r = (U_r - U_\perp Q)(I + Q^\top Q)^{-1/2}, \quad \tilde{U}_\perp = (U_r Q^\top + U_\perp)(I + QQ^\top)^{-1/2}, \tag{3.5}$$

$$\tilde{V}_r = (V_r + V_\perp P)(I + P^\top P)^{-1/2}, \quad \tilde{V}_\perp = (-V_r P^\top + V_\perp)(I + PP^\top)^{-1/2}, \tag{3.6}$$

with $\tilde{U}_r^\top \tilde{A}\tilde{V}_\perp = 0$ and $\tilde{U}_\perp^\top \tilde{A}\tilde{V}_r = 0$. Furthermore, $\tilde{U} = [\tilde{U}_r, \tilde{U}_\perp] \in \mathbf{R}^{q\times q}$ and $\tilde{V} = [\tilde{V}_r, \tilde{V}_\perp] \in \mathbf{R}^{n\times n}$ are orthogonal matrices.

Since the overall aim of this investigation is to derive the condition number as the norm of the Fréchet derivative of x_r, our intermediate goal will be to write a first-order expansion of (3.2) in terms of quantities in (3.5) and (3.6) and then replace Q and P with their respective first-order expansions with respect to E. The next theorem exploits (3.5) and (3.6) together with properties of singular decomposition to establish these expansions.

Theorem 3.3. *Suppose that $\sigma_r - \sigma_{r+1} > 0$. Then the first-order expansions for Q and P are given by:*

$$\begin{aligned}\text{vec}\,(\mathrm{Q}^\top) = \\ -\left(I_{q-r} \otimes \Sigma_r^2 - (\Sigma_\perp \Sigma_\perp^\top) \otimes I_r\right)^{-1} [I_{q-r} \otimes \Sigma_r, \Sigma_\perp \otimes I_r] \begin{bmatrix} \Psi_{(q-r,r)}(V_r^\top \otimes U_\perp^\top) \\ V_\perp^\top \otimes U_r^\top \end{bmatrix} \text{vec}(\mathrm{E}) \\ + \mathrm{o}(\|E\|),\end{aligned} \tag{3.7}$$

$$\begin{aligned}\text{vec}\,(\mathrm{P}) = \\ \left(\Sigma_r^2 \otimes I_{n-r} - I_r \otimes (\Sigma_\perp^\top \Sigma_\perp)\right)^{-1} \left[I_r \otimes \Sigma_\perp^\top, \Sigma_r \otimes I_{n-r}\right] \begin{bmatrix} V_r^\top \otimes U_\perp^\top \\ \Psi_{(r,n-r)}\,(V_\perp^\top \otimes U_r^\top) \end{bmatrix} \text{vec}(\mathrm{E}) \\ + \mathrm{o}(\|E\|).\end{aligned} \tag{3.8}$$

Proof. In agreement with:

$$U^\top A V = \begin{bmatrix} U_r^\top A V_r & U_r^\top A V_\perp \\ U_\perp^\top A V_r & U_\perp^\top A V_\perp \end{bmatrix} = \begin{bmatrix} \Sigma_r & 0 \\ 0 & \Sigma_\perp \end{bmatrix} \in \mathbf{R}^{q\times n}, \tag{3.9}$$

together with the results of Theorem 3.2, we have:

$$\begin{aligned} U^\top \tilde{A} V &= \begin{bmatrix} U_r^\top (A+E) V_r & U_r^\top (A+E) V_\perp \\ U_\perp^\top (A+E) V_r & U_\perp^\top (A+E) V_\perp \end{bmatrix} \\ &\overset{\text{def}}{=} \begin{bmatrix} \Sigma_r + E_{rr} & E_{r\perp} \\ E_{\perp r} & \Sigma_\perp + E_{\perp\perp} \end{bmatrix}, \end{aligned} \tag{3.10}$$

$$\tilde{U}^\top \tilde{A} \tilde{V} = \begin{bmatrix} \tilde{U}_r^\top \tilde{A}\tilde{V}_r & \tilde{U}_r^\top \tilde{A}\tilde{V}_\perp \\ \tilde{U}_\perp^\top \tilde{A}\tilde{V}_r & \tilde{U}_\perp^\top \tilde{A}\tilde{V}_\perp \end{bmatrix} = \begin{bmatrix} \star & 0 \\ 0 & \star \end{bmatrix}. \tag{3.11}$$

If we substitute (3.5)-(3.6) into the extra-diagonal blocks of (3.11) (that are zero), we obtain:

$$\begin{aligned}-(&QU_r^\top AV_r + QU_r^\top AV_\perp P + QU_r^\top EV_r + QU_r^\top EV_\perp P\\ &\qquad -U_\perp^\top AV_r - U_\perp^\top AV_\perp P - U_\perp^\top EV_r - U_\perp^\top EV_\perp P) = 0,\\ -(&U_r^\top AV_r P^\top - U_r^\top AV_\perp + U_r^\top EV_r P^\top - U_r^\top EV_\perp\\ &+U_\perp^\top AV_r P^\top - Q^\top U_\perp^\top AV_\perp + Q^\top U_\perp^\top EV_r P^\top - Q^\top U_\perp^\top EV_\perp) = 0.\end{aligned}$$

Furthermore, using relations (3.9) and (3.10) and after rearranging terms, we obtain (see also [111, equation 6.2]) the pair of quadratic matrix equations:

$$Q(\Sigma_r + E_{rr}) + (\Sigma_\perp + E_{\perp\perp})P = -E_{\perp r} - QE_{r\perp}P, \tag{3.12}$$
$$P(\Sigma_r + E_{rr}^\top) + (\Sigma_\perp^\top + E_{\perp\perp}^\top)Q = E_{r\perp}^\top + PE_{\perp r}^\top Q, \tag{3.13}$$

where unknowns are Q and P. We retain only first-order terms[2] in $\|E\|$ in (3.12) and (3.13) leading to:

$$Q\Sigma_r + \Sigma_\perp P = -E_{\perp r} + \mathrm{o}(\|E\|), \tag{3.14}$$
$$P\Sigma_r + \Sigma_\perp^\top Q = E_{r\perp}^\top + \mathrm{o}(\|E\|), \tag{3.15}$$

from which we obtain the system

$$Q = -\Sigma_\perp P\Sigma_r^{-1} - E_{\perp r}\Sigma_r^{-1} + \mathrm{o}(\|E\|), \tag{3.16}$$
$$P = -\Sigma_\perp^\top Q\Sigma_r^{-1} + E_{r\perp}^\top\Sigma_r^{-1} + \mathrm{o}(\|E\|), \tag{3.17}$$

by a post-multiplication of both equations (3.14) and (3.15) by Σ_r (which exists because $\sigma_1 \geq , \ldots, \geq \sigma_r > \sigma_{r+1} \geq 0$). Replacing P in (3.16) by the right hand side of (3.17), and conversely, replacing Q in (3.17) by the right hand side of (3.16) we have:

$$Q = -\Sigma_\perp(-\Sigma_\perp^\top Q\Sigma_r^{-1} + E_{r\perp}^\top\Sigma_r^{-1})\Sigma_r^{-1} - E_{\perp r}\Sigma_r^{-1} + \mathrm{o}(\|E\|), \tag{3.18}$$
$$P = -\Sigma_\perp^\top(-\Sigma_\perp P\Sigma_r^{-1} - E_{\perp r}\Sigma_r^{-1})\Sigma_r^{-1} + E_{r\perp}^\top\Sigma_r^{-1} + \mathrm{o}(\|E\|). \tag{3.19}$$

Post-multiplying (3.18) and (3.19) by Σ_r^2, and rearranging terms yields:

$$\Sigma_r^2 Q^\top - Q^\top\Sigma_\perp\Sigma_\perp^\top = -E_{r\perp}\Sigma_\perp^\top - \Sigma_r E_{\perp r}^\top + \mathrm{o}(\|E\|), \tag{3.20}$$
$$P\Sigma_r^2 - \Sigma_\perp^\top\Sigma_\perp P = \Sigma_\perp^\top E_{\perp r} + E_{r\perp}^\top\Sigma_r + \mathrm{o}(\|E\|). \tag{3.21}$$

According to property (3.3), equations (3.20) and (3.21) may be rewritten as:

$$\begin{aligned}\left(I_{q-r}\otimes\Sigma_r^2 - (\Sigma_\perp\Sigma_\perp^\top)\otimes I_r\right)\mathrm{vec}(Q^\top) &= -\mathrm{vec}\left(E_{r\perp}\Sigma_\perp^\top + \Sigma_r E_{\perp r}^\top\right) + \mathrm{o}(\|E\|)\\ &= -\left[I_{q-r}\otimes\Sigma_r, \Sigma_\perp\otimes I_r\right]\begin{bmatrix}\mathrm{vec}\,(E_{\perp r}^\top)\\ \mathrm{vec}\,(E_{r\perp})\end{bmatrix} + \mathrm{o}(\|E\|),\end{aligned}$$

[2] This is why the terms $PE_{rr}^\top$, $E_{\perp\perp}^\top Q$, $PE_{\perp r}^\top Q$, QE_{rr}, $E_{\perp\perp}P$ and $QE_{r\perp}P$ do no longer appear.

$$\begin{aligned}\left(\Sigma_r^2 \otimes I_{n-r} - I_r \otimes (\Sigma_\perp^\top \Sigma_\perp)\right) \mathrm{vec}(P) &= \mathrm{vec}\left(\Sigma_\perp^\top E_{\perp r} + E_{r\perp}^\top \Sigma_r\right) + \mathrm{o}(\|E\|) \\ &= \left[I_r \otimes \Sigma_\perp^\top, \Sigma_r \otimes I_{n-r}\right] \begin{bmatrix} \mathrm{vec}\,(E_{\perp r}) \\ \mathrm{vec}\,(E_{r\perp}^\top) \end{bmatrix} + \mathrm{o}(\|E\|).\end{aligned}$$

One can replace $\mathrm{vec}(E_{\perp r}^\top)$ and $\mathrm{vec}(E_{r\perp}^\top)$ by $\Psi(q-r,r)\mathrm{vec}(E_{\perp r})$ and $\Psi(r,n-r)\mathrm{vec}\,(E_{r\perp})$, respectively, based on the property (3.4). Note that $\left(I_{q-r} \otimes \Sigma_r^2 - (\Sigma_\perp \Sigma_\perp^\top) \otimes I_r\right)$ and $\left(\Sigma_r^2 \otimes I_{n-r} - I_r \otimes (\Sigma_\perp^\top \Sigma_\perp)\right)$ are diagonal matrices of order $(q-r)r$ and $(n-r)r$, respectively. In addition, their diagonal entries are strictly positive since $\sigma_r > \sigma_{r+1}$. Hence, their inverses exist. To conclude the proof, observe that:

$$\begin{aligned}\mathrm{vec}\,(E_{\perp r}) &= (V_r^\top \otimes U_\perp^\top)\,\mathrm{vec}\,(E), & \mathrm{vec}\,(E_{r\perp}) &= (V_\perp^\top \otimes U_r^\top)\,\mathrm{vec}\,(E), \\ \mathrm{vec}\,(E_{\perp\perp}) &= (V_\perp^\top \otimes U_\perp^\top)\,\mathrm{vec}\,(E), & \mathrm{vec}\,(E_{rr}) &= (V_r^\top \otimes U_r^\top)\,\mathrm{vec}\,(E).\end{aligned}$$

□

In what follows, we use the results in Theorem 3.2 to introduce the first-order expansion for x_r around (A,b) in terms of the partitioned singular value decomposition matrices of A, the perturbation matrix E, the vector b, and the perturbation vector f.

3.2 The condition number

The continuity and the differentiability of x_r rely on the fact that one supposes that there is a gap between σ_r and σ_{r+1}, that is $\sigma_r - \sigma_{r+1} > 0$. Consider the following counter-example. Let

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad E = \begin{pmatrix} \epsilon^2 \sin(\frac{1}{\epsilon}) & 0 \\ 0 & -\epsilon^2 \sin(\frac{1}{\epsilon}) \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad f = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

We take $r = 1$. Thus

$$\tilde{x}_r = \begin{cases} \frac{1}{1+\epsilon^2 \sin(\frac{1}{\epsilon})} e_1, & \text{if } \sin(\frac{1}{\epsilon}) > 0, \\ \frac{1}{1-\epsilon^2 \sin(\frac{1}{\epsilon})} e_2, & \text{if } \sin(\frac{1}{\epsilon}) < 0, \\ x_r & \text{if } \sin(\frac{1}{\epsilon}) = 0. \end{cases}$$

where $e_1 = (1,0)^\top$ and $e_2 = (0,1)^\top$ are the canonical vectors of $\mathbf{R}^2$. The above counter-example shows that the unit-vector of $\tilde{x}_r$ fluctuates between e_1 and e_2 as ϵ tends to 0. In this case x_r is not continuous, and a fortiori not differentiable, around A. We know from Theorem 3.2 that the singular values of $\tilde{A}$ are the disjoint union of the singular values of $\tilde{U}_r^\top \tilde{A} \tilde{V}_r$ and those of $\tilde{U}_\perp^\top \tilde{A} \tilde{V}_\perp$. To define $\tilde{x}_r$ by (3.2) it is required that the r *leading singular values* of $\tilde{A}$ be those of $\tilde{U}_r^\top \tilde{A} \tilde{V}_r$. This is achieved if $\sigma_r - \sigma_{r+1} > 0$ and E, sufficiently small[3].

Now, let us state the following lemma.

[3] Observe that in the presence of a gap $\sigma_r - \sigma_{r+1} > 0$, the bases of the involved singular subspaces of $\tilde{A}$ tend continuously to those of A as E tends 0.

Lemma 3.4. *Suppose $\sigma_r - \sigma_{r+1} > 0$. Then the first-order expansion of x_r can be written in the form:*

$$\begin{aligned}\tilde{x}_r &= x_r + V\begin{bmatrix} I_r \\ 0 \end{bmatrix}\Sigma_r^{-1} f_r - V\begin{bmatrix} I_r \\ 0 \end{bmatrix}\Sigma_r^{-1}Q^\top b_\perp \\ &\quad + V\begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix}P\Sigma_r^{-1}b_r - V\begin{bmatrix} I_r \\ 0 \end{bmatrix}\Sigma_r^{-1}E_{rr}\Sigma_r^{-1}b_r + \mathrm{o}(\|[E,f]\|). \qquad (3.22)\end{aligned}$$

Proof. We insert Equations (3.5) and (3.6) in the expression (3.2) to yield:

$$\begin{aligned}\tilde{x}_r &= (V_r + V_\perp P)((U_r - U_\perp Q)^\top(A+E)(V_r + V_\perp P))^{-1}(U_r - U_\perp Q)^\top\tilde{b} \\ &= (V_r + V_\perp P)(\Sigma_r^{-1} - \Sigma_r^{-1}U_r^\top E V_r \Sigma_r^{-1})(U_r - U_\perp Q)^\top\tilde{b} + \mathrm{o}(\|[E,f]\|),\end{aligned}$$

where we used the following result concerning a perturbation of the inverse of a matrix $(F+G)^{-1} = F^{-1} - F^{-1}GF^{-1} + \mathrm{o}(\|G\|)$, see [110, p. 131]. Developing this equation and recalling that $E_{rr} \stackrel{\text{def}}{=} U_r^\top E V_r$ gives, after rearranging terms,

$$\begin{aligned}\tilde{x}_r &= x_r + V_r\Sigma_r^{-1}U_r^\top f - V_r\Sigma_r^{-1}Q^\top U_\perp^\top b + V_\perp P\Sigma_r^{-1}U_r^\top b - V_r\Sigma_r^{-1}E_{rr}\Sigma_r^{-1}U_r^\top b \\ &\qquad + \mathrm{o}(\|[E,f]\|) \\ &= x_r + V_r\Sigma_r^{-1}f_r - V_r\Sigma_r^{-1}Q^\top b_\perp + V_\perp P\Sigma_r^{-1}b_r - V_r\Sigma_r^{-1}E_{rr}\Sigma_r^{-1}b_r + \mathrm{o}(\|[E,f]\|).\end{aligned}$$

From the properties

$$VV^\top = I,\ V^\top V_r = \begin{bmatrix} I_r \\ 0 \end{bmatrix} \text{ and } V^\top V_\perp = \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix},$$

we have:

$$\begin{aligned}\tilde{x}_r &= x_r + VV^\top V_r\Sigma_r^{-1}f_r - VV^\top V_r\Sigma_r^{-1}Q^\top b_\perp \\ &\qquad + VV^\top V_\perp P\Sigma_r^{-1}b_r - VV^\top V_r\Sigma_r^{-1}E_{rr}\Sigma_r^{-1}b_r + \mathrm{o}(\|[E,f]\|),\end{aligned}$$

which implies (3.22). □

Now, we are ready to give the expression of the matrix x_r' that represents the Fréchet derivative of x_r, with respect to the data (A,b). The expression is given in terms of the singular value decomposition information of A and the vector b. For that, we simply use results in Theorem 3.3 to einate Q and P from (3.22).

Proposition 3.5. *Suppose that $\sigma_r - \sigma_{r+1} > 0$. Then the application*

$$x_r \ : \ \left(\mathbf{R}^{q\times n}, \mathbf{R}^q\right) \longrightarrow \mathbf{R}^n \ : \ (A,b) \longrightarrow x_r$$

is a differentiable function of (A,b). In addition, we have

$$\tilde{x}_r = x_r + x_r'\begin{bmatrix} \alpha\,\mathrm{vec(E)} \\ \beta\, f \end{bmatrix} + \mathrm{o}(\|[E,\ f]\|),$$

with

$$x'_r = V \left[\frac{1}{\alpha} M,\ \frac{1}{\beta} \begin{pmatrix} \Sigma_r^{-1} & \\ & 0 \end{pmatrix} \right] W \in \mathbf{R}^{q \times (qn+q)}. \tag{3.23}$$

Here, W is an orthogonal matrix defined by

$$W = \begin{bmatrix} V_r^\top \otimes U_\perp^\top & 0 \\ V_\perp^\top \otimes U_r^\top & 0 \\ V_r^\top \otimes U_r^\top & 0 \\ V_\perp^\top \otimes U_\perp^\top & 0 \\ 0 & U^\top \end{bmatrix} \in \mathbf{R}^{(qn+q)\times(qn+q)},$$

and M is the partitioned matrix given by:

$$M = \begin{bmatrix} R_r & S_r & -T_r & 0 \\ R_\perp & S_\perp & 0 & 0 \end{bmatrix} \in \mathbf{R}^{n \times (qn)},$$

with

$$\begin{aligned}
R_r &= (b_\perp^\top \otimes \Sigma_r^{-1}) \left(I_{q-r} \otimes \Sigma_r^2 - (\Sigma_\perp \Sigma_\perp^\top) \otimes I_r \right)^{-1} (I_{q-r} \otimes \Sigma_r) \Psi_{(q-r,r)}, && (3.24)\\
S_r &= (b_\perp^\top \otimes \Sigma_r^{-1}) \left(I_{q-r} \otimes \Sigma_r^2 - (\Sigma_\perp \Sigma_\perp^\top) \otimes I_r \right)^{-1} (\Sigma_\perp \otimes I_r), && (3.25)\\
R_\perp &= \left((b_r^\top \Sigma_r^{-1}) \otimes I_{n-r} \right) \left(\Sigma_r^2 \otimes I_{n-r} - I_r \otimes (\Sigma_\perp^\top \Sigma_\perp) \right)^{-1} (I_r \otimes \Sigma_\perp^\top), && (3.26)\\
S_\perp &= \left((b_r^\top \Sigma_r^{-1}) \otimes I_{n-r} \right) \left(\Sigma_r^2 \otimes I_{n-r} - I_r \otimes (\Sigma_\perp^\top \Sigma_\perp) \right)^{-1} (\Sigma_r \otimes I_{n-r}) \Psi_{(r,n-r)}, && (3.27)\\
T_r &= \left(b_r^\top \Sigma_r^{-1} \right) \otimes \Sigma_r^{-1}. && (3.28)
\end{aligned}$$

The dimensions of these matrices are given in the following:

$$R_r,\ S_r \in \mathbf{R}^{r \times (q-r)r}, \qquad R_\perp, S_\perp \in \mathbf{R}^{(n-r)\times(q-r)r}, \qquad \text{and } T_r \in \mathbf{R}^{r \times r^2}.$$

Proof. Consider the quantities in (3.22). Using the properties of the vec operator applied to a vector, we obtain:

$$\begin{bmatrix} I_r \\ 0 \end{bmatrix} \Sigma_r^{-1} E_{rr} \Sigma_r^{-1} b_r = \begin{bmatrix} (b_r^\top \Sigma_r^{-1}) \otimes \Sigma_r^{-1} \\ 0 \end{bmatrix} \operatorname{vec}(E_{rr}) = \begin{bmatrix} T_r \\ 0 \end{bmatrix} (V_r^\top \otimes U_r^\top) \operatorname{vec}(E).$$

Taking the expressions for $\operatorname{vec}(Q^\top)$ and $\operatorname{vec}(P)$ given in (3.7) and (3.8), we have:

$$\begin{aligned}
\begin{bmatrix} I_r \\ 0 \end{bmatrix} \Sigma_r^{-1} Q^\top b_\perp &= \begin{bmatrix} b_\perp^\top \otimes \Sigma_r^{-1} \\ 0 \end{bmatrix} \operatorname{vec}(Q^\top) \\
&= -\begin{bmatrix} R_r & S_r \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^\top \otimes U_\perp^\top \\ V_\perp^\top \otimes U_r^\top \end{bmatrix} \operatorname{vec}(E) + \mathrm{o}(\|[E, f]\|),
\end{aligned}$$

$$\begin{aligned}\begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} P\Sigma_r^{-1} b_r &= \begin{bmatrix} 0 \\ (b_r^\top \Sigma_r^{-1}) \otimes I_{n-r} \end{bmatrix} \mathrm{vec}\,(P) \\ &= \begin{bmatrix} 0 & 0 \\ R_\perp & S_\perp \end{bmatrix} \begin{bmatrix} V_r^\top \otimes U_\perp^\top \\ V_\perp^\top \otimes U_r^\top \end{bmatrix} \mathrm{vec}\,(E) + \mathrm{o}(\|[E, f]\|).\end{aligned}$$

Injecting these quantities in (3.22) results in:

$$\begin{aligned}\tilde{x}_r = x_r + V \begin{bmatrix} \Sigma_r^{-1} \\ 0 \end{bmatrix} U^\top f + V \begin{bmatrix} R_r & S_r & -T_r & 0 \\ R_\perp & S_\perp & 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^\top \otimes U_\perp^\top \\ V_\perp^\top \otimes U_r^\top \\ V_r^\top \otimes U_r^\top \\ V_\perp^\top \otimes U_\perp^\top \end{bmatrix} \mathrm{vec}\,(E) \\ + \mathrm{o}(\|[E, f]\|),\end{aligned}$$

from which the results are derived. □

We can now establish the expression of the x_r condition number. We know by definition that:

$$\||x_r'|\|_{(\alpha,\beta),2} = \max_{[\alpha E,\, \beta f] \neq 0} \frac{\|x_r' \cdot (E, f)\|_2}{\|\mathrm{vec}\,[E,\ f]\|_{(\alpha,\beta)}}.$$

Thus, from (3.23) we conclude that the exact condition number of x_r is

$$\||x_r'|\|_{(\alpha,\beta),2} = \lambda_{\max}^{1/2}(\Delta), \tag{3.29}$$

where

$$\Delta \stackrel{\mathrm{def}}{=} V^\top x_r'(x_r')^\top V = \frac{1}{\alpha^2} M M^\top + \frac{1}{\beta^2} \begin{pmatrix} \Sigma_r^{-2} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{n \times n}.$$

It remains to show how Δ can be expressed with the singular values of A and the Fourier coefficients given by the elements of $U^\top b$.

Proposition 3.6. *Assume that the singular values of the matrix A are such that:*

$$\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \geq \sigma_n \geq 0.$$

Then

$$\Delta = \begin{pmatrix} \frac{1}{\alpha^2}\Delta_{rr} + \frac{1}{\beta^2}\Sigma_r^{-2} & \frac{1}{\alpha^2}\Gamma_{\perp r}^\top \\ \frac{1}{\alpha^2}\Gamma_{\perp r} & \frac{1}{\alpha^2}\Delta_{\perp\perp} \end{pmatrix},$$

where

$$\begin{aligned}\Delta_{rr} &= \mathrm{diag}\left(\sum_{\mathrm{k}=1}^{\mathrm{r}} \frac{\theta_{\mathrm{k}}^2}{\sigma_{\mathrm{k}}^2 \sigma_{\mathrm{t}}^2} + \sum_{\mathrm{k}=\mathrm{r}+1}^{\mathrm{n}} (\pi_{\mathrm{k}}^{(\mathrm{t})})^2 \frac{\sigma_{\mathrm{k}}^2 + \sigma_{\mathrm{t}}^2}{\sigma_{\mathrm{t}}^2} \theta_{\mathrm{k}}^2 + \sum_{\mathrm{k}=\mathrm{n}+1}^{\mathrm{q}} \frac{\theta_{\mathrm{k}}^2}{\sigma_{\mathrm{t}}^4} \right), \quad 1 \leq \mathrm{t} \leq \mathrm{r}, \\ \Delta_{\perp\perp} &= \mathrm{diag}\left(\sum_{\mathrm{k}=1}^{\mathrm{r}} (\pi_{\mathrm{k}}^{(\mathrm{t})})^2 \frac{\sigma_{\mathrm{k}}^2 + \sigma_{\mathrm{t}}^2}{\sigma_{\mathrm{k}}^2} \theta_{\mathrm{k}}^2 \right), \qquad \mathrm{r}+1 \leq \mathrm{t} \leq \mathrm{n},\end{aligned}$$

$$\begin{aligned}\Gamma_{\perp r} &= R_\perp R_r^\top + S_\perp S_r^\top \\ &= 2\begin{pmatrix} (\pi_{r+1}^{(1)})^2\frac{\sigma_{r+1}}{\sigma_1}\theta_1\theta_{r+1} & (\pi_{r+1}^{(2)})^2\frac{\sigma_{r+1}}{\sigma_2}\theta_2\theta_{r+1} & \cdots & (\pi_{r+1}^{(r)})^2\frac{\sigma_{r+1}}{\sigma_r}\theta_r\theta_{r+1} \\ (\pi_{r+2}^{(1)})^2\frac{\sigma_{r+2}}{\sigma_1}\theta_1\theta_{r+2} & (\pi_{r+2}^{(2)})^2\frac{\sigma_{r+2}}{\sigma_2}\theta_2\theta_{r+2} & \cdots & (\pi_{r+2}^{(r)})^2\frac{\sigma_{r+2}}{\sigma_r}\theta_r\theta_{r+2} \\ \vdots & \vdots & \ddots & \vdots \\ (\pi_n^{(1)})^2\frac{\sigma_n}{\sigma_1}\theta_1\theta_n & (\pi_n^{(2)})^2\frac{\sigma_n}{\sigma_2}\theta_2\theta_n & \cdots & (\pi_n^{(r)})^2\frac{\sigma_n}{\sigma_r}\theta_r\theta_n \end{pmatrix},\end{aligned}$$

with $(\theta_1,\ldots,\theta_q) = b^\top U$, *and* $\pi_k^{(t)} = \frac{1}{\sigma_t^2-\sigma_k^2}$, *with either* $t = 1,\ldots,r$ *and* $k = r+1,\ldots,n$ *or* $k = 1,\ldots,r$ *and* $t = r+1,\ldots,n$.

Moreover, the quantity $\pi_k^{(t)}$ *is well defined, since whenever it appears,* $\sigma_t^2 - \sigma_k^2 \neq 0$ *holds.*

Proof. First we consider the $n \times n$ symmetric matrix

$$MM^\top = \begin{bmatrix} R_rR_r^\top + S_rS_r^\top + T_rT_r^\top & -R_rR_\perp^\top - S_rS_\perp^\top \\ -R_\perp R_r^\top - S_\perp S_r^\top & R_\perp R_\perp^\top + S_\perp S_\perp^\top \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \Delta_{rr} & \Gamma_{\perp r} \\ \Gamma_{r\perp} & \Delta_{\perp\perp} \end{bmatrix}.$$

Exploiting their structure, we can write the matrices (3.24)-(3.28) as:

$$R_r = \left[\theta_{r+1}(\Sigma_r^2-\sigma_{r+1}^2I_r)^{-1},\ldots,\theta_n(\Sigma_r^2-\sigma_n^2I_r)^{-1},\theta_{n+1}\Sigma_r^{-2},\ldots,\theta_q\Sigma_r^{-2}\right]\Psi_{(q-r,r)}, \quad (3.30)$$

$$S_r = \left[\theta_{r+1}\sigma_{r+1}\Sigma_r^{-1}(\Sigma_r^2-\sigma_{r+1}^2I_r)^{-1},\ldots,\theta_n\sigma_n\Sigma_r^{-1}(\Sigma_r^2-\sigma_n^2I_r)^{-1},0,\ldots,0\right], \quad (3.31)$$

$$R_\perp = \left[\theta_1\sigma_1^{-1}(\sigma_1^2I_{n-r}-\Sigma_\perp^\top\Sigma_\perp)^{-1}\Sigma_\perp^\top,\ldots,\theta_r\sigma_r^{-1}(\sigma_r^2I_{n-r}-\Sigma_\perp^\top\Sigma_\perp)^{-1}\Sigma_\perp^\top\right], \quad (3.32)$$

$$S_\perp = \left[\theta_1(\sigma_1^2I_{n-r}-\Sigma_\perp^\top\Sigma_\perp)^{-1},\ldots,\theta_r(\sigma_r^2I_{n-r}-\Sigma_\perp^\top\Sigma_\perp)^{-1}\right]\Psi_{(r,n-r)}, \quad (3.33)$$

$$T_r = \left[\theta_1\sigma_1^{-1}\Sigma_r^{-1},\ldots,\theta_r\sigma_r^{-1}\Sigma_r^{-1}\right]. \quad (3.34)$$

In (3.30), the first of the two factors,

$$\left[\theta_{r+1}(\Sigma_r^2-\sigma_{r+1}^2I_r)^{-1},\ldots,\theta_n(\Sigma_r^2-\sigma_n^2I_r)^{-1},\theta_{n+1}\Sigma_r^{-2},\ldots,\theta_q\Sigma_r^{-2}\right], \quad (3.35)$$

is a $1\times(q-r)$ partitioned matrix. Its blocks consist of r-order diagonal matrices. Recall that the second factor in (3.30) is:

$$\Psi_{(q-r,r)} = \sum_{i=1}^{q-r}\sum_{j=1}^{r} L_{ij}\otimes L_{ij}^\top, \quad (3.36)$$

where $L_{ij} = e_ie_j^\top \in \mathbf{R}^{(q-r)r}$ with $e_i \in \mathbf{R}^{(q-r)}$ and $e_j \in \mathbf{R}^r$. Observe that $L_{ij}\otimes L_{ij}^\top$ is an $(q-r)\times r$ partitioned matrix where each block has r rows and $q-r$ columns. Furthermore, it has the block $L_{ij}^\top$ in position i,j and 0 in the remaining blocks. The multiplication of the partitioned matrices (3.35) and (3.36) results in the $1\times r$ partitioned matrix

$$R_r = \sum_{i=1}^{q-r}\sum_{j=1}^{r}\left[\theta_{r+1}(\Sigma_r^2-\sigma_{r+1}^2I_r)^{-1},\ldots,\theta_n(\Sigma_r^2-\sigma_n^2I_r)^{-1},\theta_{n+1}\Sigma_r^{-2},\ldots,\theta_q\Sigma_r^{-2}\right]L_{ij}\otimes L_{ij}^\top,$$

whose block j can be written as:

$$\sum_{i=1}^{n-r} \theta_{r+i}(\Sigma_r^2 - \sigma_{r+i}^2 I_r)^{-1} L_{ij}^\top + \sum_{i=n-r+1}^{q-r} \theta_{r+i}\Sigma_r^{-2} L_{ij}^\top.$$

Consequently, multiplying $R_\perp$ and R_r block by block yields:

$$\begin{aligned} R_\perp R_r^\top &= \sum_{j=1}^{r} \theta_j \sigma_j^{-1}(\sigma_j^2 I_{n-r} - \Sigma_\perp^\top \Sigma_\perp)^{-1}\Sigma_\perp^\top \sum_{i=1}^{n-r} L_{ij}\theta_{r+i}(\Sigma_r^2 - \sigma_{r+i} I_r)^{-1} \\ &\quad + \sum_{j=1}^{r} \theta_j \sigma_j^{-1}(\sigma_j^2 I_{n-r} - \Sigma_\perp^\top \Sigma_\perp)^{-1}\Sigma_\perp^\top \sum_{i=n-r+1}^{q-r} L_{ij}\theta_{r+i}\Sigma_r^{-2}. \end{aligned} \tag{3.37}$$

Since $\Sigma_\perp^\top \sum_{i=n-r+1}^{q-r} e_i = 0$, one has $\Sigma_\perp^\top \sum_{i=n-r+1}^{q-r} L_{ij} = \Sigma_\perp^\top \sum_{i=n-r+1}^{q-r} e_i e j^\top = 0$ and hence the last term in (3.37) vanishes. Thus

$$R_\perp R_r^\top = \sum_{i=1}^{n-r}\sum_{j=1}^{r} \theta_{r+i}\theta_j \sigma_j^{-1}(\sigma_j^2 I_{n-r} - \Sigma_\perp^\top \Sigma_\perp)^{-1}\Sigma_\perp^\top e_i e_j^\top (\Sigma_r^2 - \sigma_{r+i}^2 I_r)^{-1}.$$

A direct computation gives:

$$\theta_{r+i}\theta_j\sigma_j^{-1}(\sigma_j^2 I_{n-r} - \Sigma_\perp^\top \Sigma_\perp)^{-1}\Sigma_\perp^\top e_i = \begin{cases} \frac{\theta_{r+i}\theta_j\sigma_{r+i}}{\sigma_j(\sigma_j^2 - \sigma_{r+i}^2)} e_i & i = 1, \ldots, n-r, \\ 0 & i = n-r+1, \ldots, q-r, \end{cases}$$

where $e_i \in \mathbf{R}^{(q-r)}$ in the left-hand side and $e_i \in \mathbf{R}^{(n-r)}$ on the right-hand side. Then from

$$e_j^\top (\Sigma_r^2 - \sigma_{r+i}^2 I_r)^{-1} = \frac{1}{(\sigma_j^2 - \sigma_{r+i}^2)} e_j^\top,$$

where $e_j \in \mathbf{R}^r$ on both of the equation sides, we deduce that:

$$\begin{aligned} R_\perp R_r^\top &= \sum_{i=1}^{n-r}\sum_{j=1}^{r} \frac{1}{(\sigma_j^2 - \sigma_{r+i}^2)^2}\frac{\sigma_{r+i}}{\sigma_j}\theta_{r+i}\theta_j e_i e_j^\top, \\ &= \sum_{i=1}^{n-r}\sum_{j=1}^{r} (\pi_{r+i}^{(j)})^2 \frac{\sigma_{r+i}}{\sigma_j}\theta_{r+i}\theta_j e_i e_j^\top \quad \in \mathbf{R}^{(q-r)\times r}, \end{aligned}$$

with $\pi_{r+i}^{(j)} = \frac{1}{(\sigma_j^2 - \sigma_{r+i}^2)}$. In the same manner we can compute and show that $S_\perp S_r^\top$ is equivalent to $R_\perp R_r^\top$.

The remaining blocks in $MM^\top$ are computed by performing the block matrix-matrix multiplications. So,

$$\begin{aligned} R_r R_r^\top = \sum_{k=r+1}^{n} \theta_k^2(\Sigma_r^2 - \sigma_k^2 I_r)^{-2} + \sum_{k=n+1}^{q} \theta_k^2 \Sigma_r^{-4} &= \operatorname{diag}\left(\sum_{k=r+1}^{n} (\pi_k^{(t)})^2\theta_k^2 + \sum_{k=r+1}^{q}\frac{\theta_k^2}{\sigma_t^4}\right), \\ S_r S_r^\top = \sum_{k=r+1}^{n} \theta_k^2\sigma_k^2\Sigma_r^{-2}(\Sigma_r^2 - \sigma_k^2 I_r)^{-2} &= \operatorname{diag}\left(\sum_{k=r+1}^{n} (\pi_k^{(t)})^2\frac{\sigma_k^2}{\sigma_t^2}\theta_k^2\right), \\ T_r T_r^\top = \sum_{k=1}^{r}\frac{\theta_k^2}{\sigma_k^2}\Sigma_r^{-2} &= \operatorname{diag}\left(\sum_{k=1}^{r}\frac{\theta_k^2}{\sigma_k^2\sigma_t^2}\right), \end{aligned}$$

for $t = 1, \ldots, r$.

$$\begin{aligned} R_\perp R_\perp^\top = \sum_{k=1}^{r} \frac{\theta_k^2}{\sigma_k^2} \Sigma_\perp^\top \Sigma_\perp (\sigma_k^2 I_{n-r} - \Sigma_\perp^\top \Sigma_\perp)^{-2} &= \operatorname{diag}\left(\sum_{k=1}^{r} (\pi_k^{(t)})^2 \frac{\sigma_t^2}{\sigma_k^2} \theta_k^2 \right), \\ S_\perp S_\perp^\top = \sum_{k=1}^{r} \theta_k^2 (\sigma_k^2 I_{n-r} - \Sigma_\perp^\top \Sigma_\perp)^{-2} &= \operatorname{diag}\left(\sum_{k=1}^{r} (\pi_k^{(t)})^2 \theta_k^2 \right), \end{aligned}$$

for $t = 1, \ldots, n-r$.

Putting the above results together yields the result. □

Let us point out the fact that an early result in [54], when $r = n$, that is *when we do not perform truncation* (i.e. we assume that A is a full rank matrix), is a particular case of the results above. In fact, in this case, Δ becomes diagonal and simplifies to:

$$\Delta_{rr} = \operatorname{diag}\left(\sum_{k=1}^{n} \frac{\theta_k^2}{\sigma_k^2 \sigma_t^2} + \sum_{k=n+1}^{q} \frac{\theta_k^2}{\sigma_t^4} \right) = \operatorname{diag}\left(\frac{1}{\sigma_t^2} \left(\sum_{k=1}^{n} \frac{\theta_k^2}{\sigma_k^2} + \sum_{k=n+1}^{q} \frac{\theta_k^2}{\sigma_t^2} \right) \right),$$

for $t = 1, \ldots, n$. This implies the result given in [54], that is:

$$\begin{aligned} |||x_r'|||_{(\alpha,\beta),2} &= \sqrt{\frac{1}{\alpha^2} \frac{1}{\sigma_{min}^2} \left[\sum_{k=1}^{n} \left(\frac{\theta_k}{\sigma_k} \right)^2 + \frac{1}{\sigma_{min}^2} \sum_{k=n+1}^{q} \theta_k^2 \right] + \frac{1}{\beta^2} \frac{1}{\sigma_{min}^2}} \\ &= \|A^\dagger\| \sqrt{\frac{1}{\alpha^2} \left(\|x\|^2 + \|A^\dagger\|^2 \|r\|^2 \right) + \frac{1}{\beta^2}}, \end{aligned}$$

where $A^\dagger$ denotes the Moore-Penrose inverse (see [64, p. 421]) of A, and x denotes the solution of the linear least squares problem associated with A and b.

Looking at the general result of Proposition 3.6, we see that the quantities (scalars) involved in the computation of the x_r condition number are nothing but the singular values σ_k of A and the components θ_k of b along singular vectors u_k. Finally, observe that the critical gap is $\sigma_r - \sigma_{r+1}$.

3.3 Upper and lower bounds for the condition number and numerical illustrations

The matrix Δ is of order n. Computing its largest eigenvalue may be achieved using standard eigenvalue procedures like the power method [99] or the Lanczos algorithm [52]. We mention here a possible use of the Gershgorin circle, see [64, Theorem 8.1.22], to obtain an estimate for $\kappa_{\phi_r}(A, b)$ where n is large. We recall this theorem in the following:

Theorem 3.7. *Assume $C = [c_{ij}] \in \Re^{n \times n}$. Then*

$$\min_{1 \le i \le n} \sum_{j=1}^{n} |c_{ij}| \le \rho(C) \le \max_{1 \le i \le n} \sum_{j=1}^{n} |c_{ij}|.$$

Proof. See [63, Theorem 8.1.22] for the proof. □

We now describe comparative numerical tests carried out in MATLAB. We took pairs (A, b) from the regularization tools package[4] by P. C. Hansen [59]. We arbitrarily choose values of q, n and r. To validate the expression of the exact condition number, we use the numerical derivative code[5] authored by John D'Errico and called "jacobianest.m" of an analytically supplied function $f : z \to x$ to estimate the corresponding Jacobian at a given particular point z. The code "jacobianest.m" uses a centered finite differences approach with Romberg extrapolation to improve the estimates to sixth order. For our purpose, we have to formally recast $\phi_{r(A,b)}$ as $f : z = \text{vec}([A, b]) \to x_r$ prior the use of "jacobianest.m", and then compute the 2-norm of the estimated Jacobian. In all tests we set $\alpha = \beta = 1$.

Table 3.1 and figure 3.1 display the exact condition number, an estimate of the condition number produced with "jacobianest.m", and an upper and a lower bounds. Values of q, n and r are also supplied. The results show how the derived expression of the exact condition fits the finite difference estimate. We also see that the upper bound is sharp for the selected pairs (A,b) whereas the lower bound is very pessimistic.

problem	cond(x_r) from 3.6	fin. diff. estim. value	upper bnd	lower bnd	q	n	r
baart	7.156e+3	7.087e+3	7.157e+3	4.967e-5	20	20	5
blur	2.516e+1	2.516e+1	2.706e+1	7.898e+1	16	16	6
derive	1.698e+3	1.698e+3	1.764e+3	1.144e+1	12	12	10
foxgood	2.896e+1	2.896e+1	2.897e+1	3.415	20	20	2
heat	4.486e+1	4.478e+1	4.694e+1	3.306	12	12	10
i_laplace	1.448e+4	1.367e+4	1.449e+4	2.457	20	20	7
parallax	1.412e+5	1.411e+5	1.417e+5	1.711e+1	26	12	10
phillips	5.731e+1	5.731e+1	5.734e+1	5.547e-5	12	12	10
shaw	1.044e+3	1.044e+3	1.045e+3	1.201	12	12	8
spikes	8.178e+2	8.178e+2	8.179e+2	0	12	12	4
full	1.032e+1	1.032e1	1.832e+1	1.627	16	12	8
ursell	3.716e+5	3.716e+5	3.724e+5	1.660e+2	20	20	3
wing	3.429e+6	3.010e+6	3.430e+6	2.549	20	20	5

TABLE 3.1: The exact value of cond(x_r) using the expression in Proposition 3.6, the finite difference estimate value using "jacobianest" and the upper and lower bound of cond(x_r) for 13 problems.

The main contribution of this chapter was to study the sensitivity of the solution of a given linear least squares problem to perturbations in the data, by computing the condition number of the truncated least squares solution. We anticipate that the obtained formula for these condition number will stimulate research in several directions. In the next chapter, we present a variant of Levenberg-Marquardt algorithm to solve nonlinear least squares problems for which the exact gradient is not available or expensive to compute, and replaced by a random models.

[4] see http://www2.imm.dtu.dk/~pch/Regutools/

[5] see http://www.mathworks.com/matlabcentral/fileexchange/13490-automatic-numerical-differentiation

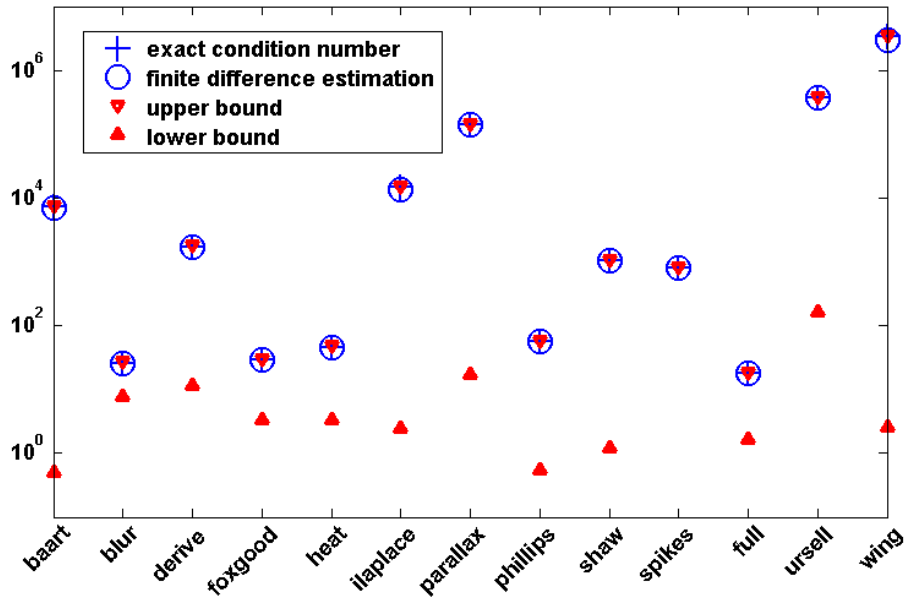


FIGURE 3.1: The exact value of $\text{cond}(x_r)$ using the expression in Proposition 3.6, the finite difference estimate value using "jacobianest" and the upper and lower bound of $\text{cond}(x_r)$ for 13 problems.

Chapter 4

Probabilistic methods for least squares problems

In this chapter, we are concerned with a class of nonlinear least squares problems for which the exact gradient is not available or expensive to compute and replaced by a probabilistic or random model. Problems of this nature arise in several important practical contexts. One example is variational modeling for meteorology, such as 3DVAR and 4DVAR. Here, ensemble methods, like EnKF and EnKS are used to approximate the data arising in the solution of the corresponding linearized least squares subproblem in a way where the true gradient is replaced by an approximated stochastic gradient model [126]. Other examples appear in the broad context of derivative-free optimization problems [24] where models of the objective function evaluation may result from, a possibly random, sampling procedure [6].

As explained in Section 2.3.4.4, the Levenberg-Marquardt algorithm is a regularization of the Gauss-Newton method. A regularization parameter is updated at every iteration and indirectly controls the size of the step, making Gauss-Newton globally convergent. The regularization term added to Gauss-Newton maintains the structure of the linearized least squares subproblems arising in data assimilation, enabling us to use techniques like ensemble methods while simultaneously providing a globally convergent approach. But, the use of ensemble methods makes random approximations to the gradient. We thus propose and analyze, in this chapter, a variant of the Levenberg-Marquardt method to deal with probabilistic gradient models.

We organize this chapter as follows: the new Levenberg-Marquardt method based on probabilistic gradient models is described in Section 4.1. Section 4.2 addresses the inexact solution of the linearized least squares subproblems arising in Levenberg-Marquardt. We cover essentially two possibilities: conjugate gradient and any generic inexact solution of the corresponding normal equations. In Section 4.3, we show that the whole approach is globally convergent to first order critical points, in the sense that a subsequence of the "true" objective function gradients goes to zero with probability one. The proposed approach is numerically illustrated in Section 4.4 with a simple problem, artificially modified to create (i) a scenario where the model gradient is a Gaussian perturbation of the exact gradient, and (ii) a scenario case where to compute the

model gradient both exact/approximated gradient routines are available but the exact one (seen as expensive) is called only with a certain probability.

4.1 The Levenberg-Marquardt method based on probabilistic gradient models

We have seen in Section 2.3.4.4 the classical version of Levenberg-Marquardt algorithm, where it is supposed that the derivatives of the functions f and F are available. In this section we are interested in the case where we do not have exact values for the Jacobian $J_F(x^j)$ and the gradient $\nabla f(x^j) = J_F(x^j)^\top F(x^j)$, (of the model defined in (2.69) $m_j(x^j + s)$ at $s = 0$), but rather approximations which we will denoted by J_{m_j} and g_{m_j}. We are further interested in the case where these model approximations are built in some random fashion. We will then consider random models of the form M_j where g_{M_j} and J_{M_j} are random variables, and use the notation $m_j = M_j(\omega_j)$, $g_{m_j} = g_{M_j}(\omega_j)$, and $J_{m_j} = J_{M_j}(\omega_j)$ for their realizations. Note that the randomness of the models turns also random the current point $x^j = X^j(\omega_j)$ and the current regularization parameter $\gamma_j = \Gamma_j(\omega_j)$ generated by the corresponding optimization algorithm.

Thus, the model:

$$\begin{aligned} m_j(x^j + s) - m_j(x^j) &= \frac{1}{2}\|F_{m_j} + J_{m_j}s\|^2 + \frac{1}{2}\gamma_j^2\|s\|^2 - \frac{1}{2}\|F_{m_j}\|^2 \\ &= g_{m_j}^\top s + \frac{1}{2}s^\top \left(J_{m_j}^\top J_{m_j} + \gamma_j^2 I\right) s \end{aligned}$$

is a realization of:

$$M_j(X^j + s) - M_j(X^j) = g_{M_j}^\top s + \frac{1}{2}s^\top \left(J_{M_j}^\top J_{M_j} + \Gamma_j^2 I\right) s.$$

Note that we subtracted the order zero term to the model to avoid unnecessary terminology. Our subproblem becomes then just:

$$\min_{s\in\mathbf{R}^n} \; m_j(x^j + s) - m_j(x^j) = g_{m_j}^\top s + \frac{1}{2}s^\top \left(J_{m_j}^\top J_{m_j} + \gamma_j^2 I\right) s. \tag{4.1}$$

We will now impose that the gradient models g_{M_j} are accurate with a certain probability regardless of the history $M_1, \ldots, M_{j-1}$. The accuracy is defined in terms of a multiple of the inverse of the square of regularization parameter (as it happens in [6] for trust-region methods based on probabilistic models where it is defined in terms of a multiple of the trust-region radius). As we will see later in the convergence analysis (since the regularization parameter is bounded from below), one can demand less here and consider just the inverse of a positive power of the regularization parameter.

Assumption 4.1.1. Given constants $\alpha \in (0, 2]$, $\kappa_{eg} > 0$, and $p \in (0, 1]$, the sequence of random gradient models $\{g_{M_j}\}$ is (p)-probabilistically κ_{eg}-first order accurate, for corresponding

sequences $\{X^j\}$, $\{\Gamma_j\}$, if the events

$$S_j \;=\; \left\{ \|g_{M_j} - J(X^j)^\top F(X^j)\| \;\leq\; \frac{\kappa_{eg}}{\Gamma_j^\alpha} \right\}$$

satisfy the following submartingale-like condition

$$p_j^* \;=\; \mathbb{P}(S_j | \mathcal{F}_{j-1}^M) \;\geq\; p, \tag{4.2}$$

where $\mathcal{F}_j^M = \sigma(M_0, \ldots, M_{j-1})$ is the σ-algebra generated by $M_0, \ldots, M_{j-1}$.

Correspondingly, a gradient model realization g_{m_j} is said to be κ_{eg}-first order accurate if:

$$\|g_{m_j} - J(x^j)^\top F(x^j)\| \;\leq\; \frac{\kappa_{eg}}{\gamma_j^\alpha}.$$

The version of Levenberg-Marquardt that we will analyze and implement takes a successful step if the ratio ρ_j between actual and predicted reductions is sufficiently positive (condition $\rho_j \geq \eta_1$ below). In such cases, and now deviating from classical Levenberg-Marquardt and following [6], the regularization parameter γ_j is increased if the size of the gradient model is not of the order of the inverse of γ_j (condition $\|g_{m_j}\| < \eta_2/\gamma_j^2$ below). Another relevant distinction is that we necessarily decrease γ_j in successful iterations when $\|g_{m_j}\| \geq \eta_2/\gamma_j^2$. The algorithm is described below and generates a sequence of realizations for the above mentioned random variables.

Algorithm 4.1: Levenberg-Marquardt based on probabilistic gradient models

Initialization

Choose the constants $\eta_1 \in (0,1)$, $\eta_2, \gamma_{\min} > 0$, $\lambda > 1$, and $0 < p_{\min} \leq p_{\max} < 1$.
Select x_0 and $\gamma_0 \geq \gamma_{\min}$.

For $j = 0, 1, 2, \ldots$

1. Solve (or approximately solve) (4.1), and let s^j denote such a solution.
2. Compute $\rho_j = \frac{f(x^j) - f(x^j + s^j)}{m_j(x^j) - m_j(x^j + s^j)}$.
3. Make a guess p_j of the probability p_j^* given in (4.2) such that $p_{\min} \leq p_j \leq p_{\max}$.
   If $\rho_j \geq \eta_1$, then set $x_{j+1} = x^j + s^j$ and

$$\gamma_{j+1} = \begin{cases} \lambda \gamma_j & \text{if } \|g_{m_j}\| < \eta_2/\gamma_j^2, \\ \max\left\{ \frac{\gamma_j}{\lambda \frac{1-p_j}{p_j}}, \gamma_{\min} \right\} & \text{if } \|g_{m_j}\| \geq \eta_2/\gamma_j^2. \end{cases}$$

   Otherwise, set $x^{j+1} = x^j$ and $\gamma_{j+1} = \lambda \gamma_j$.

If exact gradients are used (in other words, if $g_{M_j} = J(X^j)^\top F(X^j)$), then one always has:

$$p_j^* \;=\; \mathbb{P}\left(0 \leq \frac{\kappa_{eg}}{\Gamma_j^\alpha} \,\middle|\, \mathcal{F}_{j-1}^M\right) \;=\; 1,$$

and the update of γ in successful iterations reduces to $\gamma_{j+1} = \max\{\gamma_j, \gamma_{\min}\}$ (when $\|g_{m_j}\| \geq \eta_2/\gamma_j^2$), as in the more classical deterministic-type Levenberg-Marquardt methods. In general one should guess p_j based on the knowledge of the random error occurred in the application context. It is however pertinent to stress that the algorithm runs for any guess of $p_j \in (0, 1]$ such that $p_j \in [p_{\min}, p_{\max}]$.

4.2 Inexact solution of the linearized least squares subproblems

Step 1 of Algorithm 4.1 requires the approximate solution of subproblem (4.1). As in trust-regions methods, there are different techniques to approximate the solution of this subproblem yielding a globally convergent step, and we will discuss three of them in this section. For the purposes of global convergence it is sufficient to compute a step s^j that provides a reduction in the model as good as the one produced by the so-called Cauchy step (defined as the minimizer the model along the negative gradient or steepest descent direction $-g_{m_j}$).

4.2.1 A Cauchy step

The Cauchy step is defined by minimizing $\phi(t) = m_j(x^j - tg_{m_j})$ when $t > 0$. We have:

$$\begin{aligned}
\phi(t) &= m_j(x^j - tg_{m_j}) = m_j(x^j) - t\|g_{m_j}\|^2 + \frac{t^2}{2} g_{m_j}^\top \left(J_{m_j}^\top J_{m_j} + \gamma_j^2 I\right) g_{m_j}, \\
\phi'(t) &= -\|g_{m_j}\|^2 + t g_{m_j}^\top \left(J_{m_j}^\top J_{m_j} + \gamma_j^2 I\right) g_{m_j}, \\
\phi''(t) &= g_{m_j}^\top \left(J_{m_j}^\top J_{m_j} + \gamma_j^2 I\right) g_{m_j} > 0 \;\; \left(\text{ if } g_{m_j} \neq 0\right).
\end{aligned}$$

Therefore the minimizer of ϕ is:

$$t^c = \frac{\|g_{m_j}\|^2}{g_{m_j}^\top (J_{m_j}^\top J_{m_j} + \gamma_j^2 I) g_{m_j}}.$$

Thus the Cauchy step is equal to:

$$s^{j^c} \;=\; -\frac{\|g_{m_j}\|^2}{g_{m_j}^\top (J_{m_j}^\top J_{m_j} + \gamma_j^2 I) g_{m_j}} g_{m_j}. \tag{4.3}$$

The corresponding Cauchy decrease on the model is:

$$\begin{aligned} m_j(x^j) - m_j(x^j + s^{j^c}) &= \frac{\|g_{m_j}\|^2}{g_{m_j}^\top (J_{m_j}^\top J_{m_j} + \gamma_j^2 I) g_{m_j}} \|g_{m_j}\|^2 - \frac{1}{2} \frac{\|g_{m_j}\|^4}{g_{m_j}^\top (J_{m_j}^\top J_{m_j} + \gamma_j^2 I) g_{m_j}} \\ &= \frac{1}{2} \frac{\|g_{m_j}\|^4}{g_{m_j}^\top (J_{m_j}^\top J_{m_j} + \gamma_j^2 I) g_{m_j}}. \end{aligned}$$

Since $g_{m_j}^\top (J_{m_j}^\top J_{m_j} + \gamma_j^2 I) g_{m_j} \leq \|g_{m_j}\|^2 (\|J_{m_j}\|^2 + \gamma_j^2)$, we conclude that:

$$m_j(x^j) - m_j(x^j + s^{j^c}) \geq \frac{1}{2} \frac{\|g_{m_j}\|^2}{\|J_{m_j}\|^2 + \gamma_j^2}.$$

The Cauchy step (4.3) is cheap to calculate as it does not require any system solve. Moreover, the Levenberg-Marquardt method will be globally convergent if it uses a step that attains a reduction in the model as good as a multiple of the Cauchy decrease. Thus we will impose the following assumption on the step calculation:

Assumption 4.2.1. For every step j and for all realizations m_j of M_j,

$$m_j(x^j) - m_j(x^j + s^j) \geq \frac{\theta_{fcd}}{2} \frac{\|g_{m_j}\|^2}{\|J_{m_j}\|^2 + \gamma_j^2}$$

for some constant $\theta_{fcd} > 0$.

4.2.2 A truncated-CG step

Despite providing a sufficient reduction in the model and being cheap to compute, the Cauchy step is a particular form of steepest descent, which can perform poorly regardless of the step length. One can see that the Cauchy step depends on $J_{m_j}^\top J_{m_j}$ only in the step length. Faster convergence can be expected if the matrix $J_{m_j}^\top J_{m_j}$ influences also the step direction.

Since the Cauchy step is the first step of the conjugate gradient method when applied to the minimization of the quadratic $m_j(x^j + s) - m_j(x^j)$, it is natural to propose to run CG further and stop only when the residual becomes relatively small. Since CG generates iterates by minimizing the quadratic over nested Krylov subspaces, and the first subspace is the one generated by g_{m_j} (see, e.g., [93, Theorem 5.2]), the decrease attained at the first CG iteration (i.e., by the Cauchy step) is kept by the remaining.

4.2.3 A step from inexact solution of normal equations

Following the spirit of what was done by [31], where the authors propose to approximately solve the linearized subproblem in the Newton method. We propose another possibility to approximately solve subproblem (4.1) by applying some iterative solver (not necessarily CG) to the solution of the normal equations:

$$\left(J_{m_j}^\top J_{m_j} + \gamma_j^2 I\right) s^j = -g_{m_j}.$$

An inexact solution $s^{j\,in}$ is then computed such that:

$$\left(J_{m_j}^\top J_{m_j} + \gamma_j^2 I\right) s^{j\,in} \;=\; -g_{m_j} + r_j \tag{4.4}$$

for a relatively small residual r_j satisfying $\|r_j\| \leq \epsilon_j \|g_{m_j}\|$. For such sufficiently small residuals we can guarantee Cauchy decrease.

Assumption 4.2.2. For some constants $\beta_{in} \in (0,1)$ and $\theta_{in} > 0$, suppose that $\|r_j\| \leq \epsilon_j \|g_{m_j}\|$ and

$$\epsilon_j \;\leq\; \min\left\{\frac{\theta_{in}}{\gamma_j^\alpha}, \sqrt{\beta_{in}\frac{\gamma_j^2}{\|J_{m_j}\|^2+\gamma_j^2}}\right\}.$$

Note that we only need the second above bound on ϵ_j to prove the desired Cauchy decrease. The first above bound will be used later, in the convergence analysis.

Lemma 4.1. *Under Assumption 4.2.2, an inexact step $s^{j\,in}$ of the form (4.4) achieves Cauchy decrease and it satisfies Assumption 4.2.1 with $\theta_{fcd} = 2(1-\beta_{in})$.*

Proof. In the proof we will omit the indices j. One has:

$$\begin{aligned} m(x) - m(x+s^{in}) \;&=\; -g_m^\top s^{in} - \frac{1}{2}(-g_m + r)^\top s^{in} \;=\; -\frac{1}{2}(g_m + r)^\top s^{in} \\ &=\; \frac{1}{2}(g_m - r)^\top (J_m^\top J_m + \gamma^2 I)^{-1}(g_m + r). \end{aligned}$$

Since $J_m^\top J_m$ is positive semidefinite:

$$r^\top (J_m^\top J_m + \gamma^2 I)^{-1} r \;\leq\; \frac{\|r\|^2}{\gamma^2} \;\leq\; \frac{\epsilon^2 \|g_m\|^2}{\gamma^2}$$

and

$$(g_m)^\top (J_m^\top J_m + \gamma^2 I)^{-1} g_m \;\geq\; \frac{\|g_m\|^2}{\|J_m\|^2 + \gamma^2}.$$

Thus, using Assumption 4.2.2, we conclude that:

$$\begin{aligned} m(x) - m(x+s^{in}) \;&\geq\; \left(\frac{1}{\|J_m\|^2+\gamma^2} - \frac{\epsilon^2}{\gamma^2}\right)\|g_m\|^2 \\ &\geq\; \frac{2(1-\beta_{in})}{2}\frac{\|g_m\|^2}{\|J_m\|^2+\gamma^2}. \end{aligned}$$

□

4.3 Global convergence to first order critical points

We start by proving that two terms, that later will appear in the difference between the actual and predicted decreases, have the right order accuracy in terms of γ_j.

Lemma 4.2. *For the three steps proposed (Cauchy, truncated CG, and inexact normal equations), one has that:*

$$\|s^j\| \;\leq\; \frac{2\|g_{m_j}\|}{\gamma_j^2}$$

and

$$|{s^j}^\top(\gamma_j^2 s^j + g_{m_j})| \;\leq\; \frac{4\|J_{m_j}\|^2\|g_{m_j}\|^2 + 2\theta_{in}\|g_{m_j}\|^2}{\min\{1,\gamma_{\min}^{2-\alpha}\}\gamma_j^{2+\alpha}}.$$

(Assumption 4.2.2 is assumed for the inexact normal equations step $s^j = {s^j}^{in}$.)

Proof. We will omit the indices j again in the proof.

If $s = s^c$ is the Cauchy point, since $J_m^\top J_m$ is positive semidefinite, $\|g_m^\top(J_m^\top J_m + \gamma^2 I)g_m\| \geq \gamma^2\|g_m\|^2$ and we have that $\|s^c\| \leq \|g_m\|/\gamma^2$. To prove the second inequality:

$$\begin{aligned}(s^c)^\top(\gamma^2(s^c) + g_m) &= \frac{\gamma^2\|g_m\|^6}{((g_m)^\top(J_m^\top J_m + \gamma^2 I)g_m)^2} - \frac{\|g_m\|^4}{(g_m)^\top(J_m^\top J_m + \gamma^2 I)g_m} \\ &= -\frac{\|g_m\|^4(g_m)^\top J_m^\top J_m g_m}{((g_m)^\top(J_m^\top J_m + \gamma^2 I)g_m)^2},\end{aligned}$$

and then using a similar argument and $\gamma \geq \gamma_{\min}$:

$$|(s^c)^\top(\gamma^2(s^c) + g_m)| \;\leq\; \frac{\|J_m\|^2\|g_m\|^2}{\gamma^4} \;\leq\; \frac{4\|J_m\|^2\|g_m\|^2 + 2\theta_{in}\|g_m\|^2}{\min\{1,\gamma_{\min}^{2-\alpha}\}\gamma^{2+\alpha}}.$$

If $s = s^{cg}$ is obtained by truncated CG, then there exists an orthogonal matrix V with first column given by $-g_m/\|g_m\|$ and such that:

$$s^{cg} \;=\; V\left(V^\top(J_m^\top J_m + \gamma^2 I)V\right)^{-1}V^\top g_m \;=\; V\left(V^\top J_m^\top J_m V + \gamma^2 I\right)^{-1}\|g_m\|e_1,$$

where e_1 is the first vector of the canonical basis of $\mathbf{R}^n$. From the positive semidefiniteness of $V^\top J_m^\top J_m V$, we immediately obtain $\|s^{cg}\| \leq \|g_m\|/\gamma^2$. To prove the second inequality we apply the Sherman–Morrisson–Woodbury formula, to obtain:

$$s^{cg} \;=\; V\left(\frac{1}{\gamma^2}I - \frac{1}{\gamma^4}(J_mV)^\top\left(I + \frac{(J_mV)(J_mV)^\top}{\gamma^2}\right)^{-1}(J_mV)\right)\|g_m\|e_1.$$

Since $Ve_1 = -g_m/\|g_m\|$,

$$\gamma^2 s^{cg} + g_m \;=\; -\frac{1}{\gamma^2}V(J_mV)^\top\left(I + \frac{(J_mV)(J_mV)^\top}{\gamma^2}\right)^{-1}(J_mV)\|g_m\|e_1.$$

Now, from the fact that $(J_mV)(J_mV)^\top/\gamma^2$ is positive semidefinite, the norm of the inverse of $I + (J_mV)(J_mV)^\top/\gamma^2$ is no greater than one, and thus (since V is orthogonal):

$$\|\gamma^2 s^{cg} + g_m\| \;\leq\; \frac{\|J_m\|^2\|g_m\|}{\gamma^2}.$$

Finally (recalling $\gamma \geq \gamma_{\min}$),

$$\begin{aligned}|(s^{cg})^\top(\gamma^2(s^{cg}) + g_m)| &\leq \|s^{cg}\|\|\gamma^2 s^{cg} + g_m\| \leq \frac{\|J_m\|^2\|g_m\|^2}{\gamma^4} \\ &\leq \frac{4\|J_m\|^2\|g_m\|^2 + 2\theta_{in}\|g_m\|^2}{\min\{1,\gamma_{\min}^{2-\alpha}\}\gamma^{2+\alpha}}.\end{aligned}$$

If $s = s^{in}$ is an inexact solution of the normal equations, and the residual satisfies Assumption 4.2.2, $\|s^{in}\| \leq (\|g_m\| + \|r\|)/\gamma^2 \leq 2\|g_m\|/\gamma^2$. Applying the Sherman–Morrisson–Woodbury formula:

$$s^{in} = \left(\frac{1}{\gamma^2}I - \frac{1}{\gamma^4}J_m^\top\left(I + \frac{J_m J_m^\top}{\gamma^2}\right)^{-1} J_m\right)(-g_m + r).$$

Thus,

$$\gamma^2 s^{in} + g_m = -\frac{1}{\gamma^2}J_m^\top\left(I + \frac{J_m J_m^\top}{\gamma^2}\right)^{-1} J_m(-g_m + r) + r,$$

Using the fact that the norm of the inverse above is no greater than one, Assumption 4.2.2, and $\gamma \geq \gamma_{\min}$:

$$\begin{aligned} |(s^{in})^\top(\gamma^2(s^{in}) + g_m)| &\leq \|s^{in}\|\|\gamma^2 s^{in} + g_m\| \\ &\leq \frac{4\|J_m\|^2\|g_m\|^2}{\gamma^4} + \frac{2\theta_{in}\|g_m\|^2}{\gamma^{2+\alpha}} \\ &\leq \frac{4\|J_m\|^2\|g_m\|^2 + 2\theta_{in}\|g_m\|^2}{\min\{1, \gamma_{\min}^{2-\alpha}\}\gamma^{2+\alpha}}. \end{aligned}$$

□

We proceed by stating the conditions required for global convergence.

Assumption 4.3.1. The function f is continuously differentiable in an open set containing $L(x_0) = \{x \in \mathbf{R}^n : f(x) \leq f(x_0)\}$ with Lipschitz continuous gradient on $L(x_0)$ and corresponding constant $\nu > 0$.

The Jacobian model is uniformly bounded, i.e., there exists $\kappa_{Jm} > 0$ such that $\|J_{m_j}\| \leq \kappa_{Jm}$ for all j.

The next result is a classical one and essentially says that the actual and predicted reductions match each other well for a value of the regularization parameter γ_j sufficiently large relatively to the size of the gradient model (which would correspond to say for a sufficiently small trust-region radius in trust-region methods).

Lemma 4.3. *Let Assumption 4.3.1 hold. Let also Assumption 4.2.2 hold for the inexact normal equations step $s^j = s^{j^{in}}$. If x^j is not a critical point of f and the gradient model g_{m_j} is κ_{eg}-first order accurate, and if:*

$$\gamma_j \geq \left(\frac{\kappa_j}{1-\eta_1}\right)^{\frac{1}{\alpha}} \text{ with } \kappa_j = \left(1 + \frac{\kappa_{Jm}^2}{\gamma_{\min}^2}\right)\frac{2\nu + \frac{2\kappa_{eg}}{\|g_{m_j}\|} + 2\theta_{in} + 8\kappa_{Jm}^2}{\min\{1, \gamma_{\min}^{2-\alpha}\}\theta_{fcd}},$$

then $\rho_j \geq \eta_1$.

Proof. Again we omit the indices j in the proof. Making a Taylor expansion:

$$\begin{aligned} 1-\frac{\rho}{2} &= \frac{m(x)-f(x)+f(x+s)-m(x+s)+m(x)-m(x+s)}{2[m(x)-m(x+s)]} \\ &= \frac{s^\top J(x)^\top F(x)+R-s^\top g_m-s^\top(J_m^\top J_m+\gamma^2 I)s-s^\top g_m}{2[m(x)-m(x+s)]} \\ &= \frac{R+(J(x)^\top F(x)-g_m)^\top s-s^\top(J_m^\top J_m)s-s^\top(\gamma^2 s+g_m)}{2[m(x)-m(x+s)]}, \end{aligned}$$

where $R \leq \nu\|s\|^2/2$.

Now, using Lemma 4.2, Assumptions 4.2.1 and 4.3.1, and $\gamma \geq \gamma_{\min}$:

$$\begin{aligned} 1-\frac{\rho}{2} &\leq \frac{\frac{\nu}{2}\|s\|^2+\frac{\kappa_{eg}}{\gamma^\alpha}\|s\|+\|J_m\|^2\|s\|^2-s^\top(\gamma^2 s+g)}{\frac{\theta_{fcd}\|g_m\|^2}{\|J_m\|^2+\gamma^2}} \\ &\leq \frac{\frac{2\nu\|g_m\|^2}{\gamma^4}+\frac{2\kappa_{eg}\|g_m\|}{\gamma^{2+\alpha}}+\frac{4\kappa_{Jm}^2\|g_m\|^2}{\gamma^4}+\frac{4\kappa_{Jm}^2\|g_m\|^2+2\|g_m\|^2\theta_{in}}{\min\{1,\gamma_{\min}^{2-\alpha}\}\gamma^{2+\alpha}}}{\frac{\theta_{fcd}\|g_m\|^2}{\gamma^2(\|J_m\|^2/\gamma_{\min}^2+1)}} \\ &\leq \frac{\left(1+\frac{\kappa_{Jm}}{\gamma_{\min}^2}\right)\left(2\nu+\frac{2\kappa_{eg}}{\|g_m\|}+2\theta_{in}+8\kappa_{Jm}^2\right)}{\min\{1,\gamma_{\min}^{2-\alpha}\}\theta_{fcd}\gamma^\alpha} \leq \frac{\kappa}{\gamma^\alpha} \leq 1-\eta_1. \end{aligned}$$

We have thus proved that $\rho \geq 2\eta_1 > \eta_1$. □

One now establishes that the regularization parameter goes to infinity, which corresponds to say in [6] that the trust-region radius goes to zero.

Lemma 4.4. *Let the second part of Assumption 4.3.1 hold (the uniform bound on J_{m_j}). For every realization of the Algorithm 4.1, $\lim_{j\to\infty} \gamma_j = \infty$.*

Proof. If the result is not true, then there exists a bound $B > 0$ such that the number of times that $\gamma_j < B$ happens is infinite. Because of the way γ_j is updated one must have an infinity of iterations such $\gamma_{j+1} \leq \gamma_j$, and for these iterations one has $\rho_j \geq \eta_1$ and $\|g_{m_j}\| \geq \eta_2/B^2$. Thus,

$$\begin{aligned} f(x_j)-f(x^j+s^j) &\geq \eta_1[m_j(x_j)-m_j(x^j+s^j)] \\ &\geq \eta_1\left(\frac{\theta_{fcd}}{2}\frac{1}{\|J_m\|^2+\gamma^2}\right)\|g_{m_j}\|^2 \\ &\geq \frac{\eta_1\theta_{fcd}}{2(\kappa_{Jm}^2+B^2)}\left(\frac{\eta_2}{B^2}\right)^2. \end{aligned}$$

Since f is bounded from below by zero, the number of such iterations can not be infinite, and hence we arrived at a contradiction. □

Now, if we assume that the gradient models are (p_j)-probabilistically κ_{eg}-first order accurate, we can show our main global convergence result. First we will state an auxiliary result from the literature that will be useful for the analysis (see [38, Theorem 5.3.1] and [38, Exercise 5.3.1]).

Lemma 4.5. *Let G_j be a submartingale, in other words, a set of random variables which are integrable ($E(|G_j|) < \infty$) and satisfy $E(G_j|\mathcal{F}_{j-1}) \geq G_{j-1}$, for every j, where $\mathcal{F}_{j-1} =$*

$\sigma(G_0, \ldots, G_{j-1})$ is the σ-algebra generated by $G_0, \ldots, G_{j-1}$ and $E(G_j|\mathcal{F}_{j-1})$ denotes the conditional expectation of G_j given the past history of events $\mathcal{F}_{j-1}$.

Assume further that there exists $M > 0$ such that $|G_j - G_{j-1}| \leq M < \infty$, for every j. Consider the random events $C = \{\lim_{j\to\infty} G_j$ exists and is finite$\}$ and $D = \{\limsup_{j\to\infty} G_j = \infty\}$. Then $P(C \cup D) = 1$.

Theorem 4.6. *Let Assumption 4.3.1 hold. Let also Assumption 4.2.2 hold for the inexact normal equations step $s^j = s^{j\,in}$.*

Suppose that the gradient model sequence $\{g_{M_j}\}$ is (p_j)-probabilistically κ_{eg}-first order accurate for some positive constant κ_{eg} (Assumption 4.1.1). Let $\{x^j\}$ be a sequence of random iterates generated by Algorithm 4.1. Then almost surely:

$$\liminf_{j\to\infty} \|\nabla f(x^j)\| \;=\; 0.$$

Proof. The proof follows the same lines as [6, Theorem 4.2]. Let

$$W_j \;=\; \sum_{i=0}^{j} \left(\frac{1}{p_i} 1_{S_i} - 1\right),$$

where S_i is as in Assumption 4.1.1. Recalling $p_j^* = \mathbb{P}(S_j|\mathcal{F}_{j-1}^M) \geq p_j$, we start by showing that $\{W_j\}$ is a submartingale:

$$E(W_j|\mathcal{F}_{j-1}^M) \;=\; W_{j-1} + \frac{1}{p_j}\mathbb{P}(S_j|\mathcal{F}_{j-1}^M) - 1 \;\geq\; W_{j-1}.$$

Moreover, $\min\{1, 1/p_j - 1\} \leq |W_j - W_{j-1}| \leq \max\{(1-p_j)/p_j, 1\} \leq \max\{1/p_j, 1\} = 1/p_j$. Since $0 < p_{\min} \leq p_j \leq p_{\max} < 1$, one has $0 < \min\{1, 1/p_{\max} - 1\} \leq |W_j - W_{j-1}| \leq 1/p_{\min}$. Thus, from $0 < \min\{1, 1/p_{\max} - 1\} \leq |W_j - W_{j-1}|$, the event $\{\lim_{j\to\infty} W_j$ exists and is finite$\}$ has probability zero, and using Lemma 4.5 and $|W_j - W_{j-1}| \leq 1/p_{\min}$, one concludes that $\mathbb{P}(\limsup_{j\to\infty} W_j = \infty) = 1$.

Suppose there exist $\epsilon > 0$ and j_1 such that, with positive probability, $\|\nabla f(x^j)\| \geq \epsilon$ for all $j \geq j_1$. Let now $\{x^j\}$ and $\{\gamma_j\}$ be any realization of $\{x^j\}$ and $\{\Gamma_j\}$, respectively, built by Algorithm 4.1. By Lemma 4.4, there exists j_2 such that: $\forall j \geq j_2$

$$\gamma_j \;>\; b_\epsilon \;=\; \max\left\{\left(\frac{2\kappa_{eg}}{\epsilon}\right)^{\frac{1}{\alpha}}, \left(\frac{2\eta_2}{\epsilon}\right)^{\frac{1}{2}}, \lambda^{\frac{p-1}{p}}\gamma_{\min}, \left(\frac{\kappa_\epsilon}{1-\eta_1}\right)^{\frac{1}{\alpha}}\right\} \tag{4.5}$$

where

$$\kappa_\epsilon \;=\; \left(1 + \frac{\kappa_{Jm}^2}{\gamma_{\min}^2}\right) \frac{2\nu + \frac{4\kappa_{eg}}{\epsilon} + 2\theta_{in} + 8\kappa_{J_m}^2}{\min\{1, \gamma_{\min}^{2-\alpha}\}\theta_{fcd}}.$$

For any $j \geq j_0 = \max\{j_1, j_2\}$ two cases are possible.

If $1_{S_j} = 1$, then, from (4.5),

$$\|g_{m_j} - J(x^j)^\top F(x^j)\| \;\leq\; \frac{\kappa_{eg}}{\gamma_j^\alpha} \;<\; \frac{\epsilon}{2},$$

yielding $\|g_{m_j}\| \geq \epsilon/2$. From (4.5) we also have that $\|g_{m_j}\| \geq \epsilon/2 \geq \eta_2/\gamma_j^2$. On the other hand, Lemma 4.3, (4.5), and $\|g_{m_j}\| \geq \epsilon/2$ together imply that $\rho_j \geq \eta_1$. Hence, from this and Step 3 of Algorithm 4.1, the iteration is successful. Also, from $\|g_{m_j}\| \geq \eta_2/\gamma_j^2$ and (4.5) (note that $(1-x)/x$ is decreasing in $(0,1]$), γ is updated in Step 3 as:

$$\gamma_{j+1} \;=\; \frac{\gamma_j}{\lambda^{\frac{1-p_j}{p_j}}}.$$

Let now B_j be a random variable with realization $b_j = \log_\lambda(b_\epsilon/\gamma_j)$. In the case $1_{S_j} = 1$,

$$b_{j+1} \;=\; b_j + \frac{1-p_j}{p_j}.$$

If $1_{S_j} = 0$, then $b_{j+1} \geq b_j - 1$, because either $\gamma_{j+1} \leq \gamma_j$ therefore $b_{j+1} \geq b_j$ or $\gamma_{j+1} = \lambda\gamma_j$ therefore $b_{j+1} \geq b_j - 1$. Hence $B_j - B_{j_0} \geq W_j - W_{j_0}$, and from $\mathbb{P}(\limsup_{j\to\infty} W_j = \infty) = 1$ one obtains $\mathbb{P}(\limsup_{j\to\infty} B_j = \infty) = 1$ which leads to a contradiction with the fact that $B_j < 0$ happens for all $j \geq j_0$ with positive probability. □

4.4 A numerical illustration

The main concern in the application of Algorithm 4.1 is to ensure that the gradient model is (p_j)-probabilistically accurate (i.e., $p_j^* \geq p_j$, see Assumption 4.1.1) or at least to find a lower bound $p_{\min} > 0$ such that $p_j^* \geq p_{\min}$. However, one can, in some situations, overcome these difficulties such as in the cases where the model gradient (i) is a Gaussian perturbation of the exact one, or (ii) results from using either the exact one (seen as expensive) or an approximation. In the former case we will consider a run of the algorithm under a stopping criterion of the form $\gamma_j > \gamma_{\max}$.

4.4.1 Gaussian noise

At each iteration of the algorithm, we consider an artificial random gradient model, by adding to the exact gradient an independent Gaussian noise, more precisely we have $g_{M_j} = J(x^j)^\top \nabla F(x^j)$ $+\varepsilon_j$ where $(\varepsilon_j)_i \sim N(0, \sigma_{j,i}^2)$, for $i = 1, \ldots, n$. Let Σ_j be a diagonal matrix with diagonal elements $\sigma_{j,i}$, $i = 1, \ldots, n$. It is known that:

$$\|\Sigma_j \varepsilon_j\|^2 \;=\; \sum_{i=1}^{n} \left(\frac{(\varepsilon_j)_i}{\sigma_{j,i}}\right)^2 \;\sim\; \chi_2(n),$$

where $\chi_2(n)$ is the chi-2 distribution with n degrees of freedom. To be able to give an explicit form of the probability of the model being κ_{eg}-first order accurate, for a chosen $\kappa_{eg} > 0$, we assume also that the components of the noise are identically distributed, that is $\sigma_{j,i} = \sigma_j$, $\forall i \in \{1, \ldots, n\}$. Because of the way in which γ_j is updated in Algorithm 4.1, it is bounded by $\lambda^j\gamma_0$, and thus $\Gamma_j \leq \min\{\lambda^j\gamma_0, \gamma_{\max}\}$, where $\gamma_{\max}$ is the constant used in the stopping criterion.

One therefore has:

$$\begin{aligned} p_j^* &= \mathbb{P}\left(\|g_{M_j} - J(x^j)^\top F(x^j)\| \leq \frac{\kappa_{eg}}{\Gamma_j^\alpha} \middle| F_{j-1}^M \right) \\ &\geq \mathbb{P}\left(\|\Sigma_j \varepsilon_j\|^2 \leq \left(\frac{\kappa_{eg}}{\sigma_j \min\{\lambda^j \gamma_0, \gamma_{\max}\}^\alpha} \right)^2 \middle| F_{j-1}^M \right). \end{aligned}$$

Using the Gaussian nature of the noise ε_j and the fact that it is independent from the filtration F_{j-1}^M, we obtain:

$$p_j^* \geq CDF_{\chi^2(n)}^{-1}\left(\left(\frac{\kappa_{eg}}{\sigma_j \min\{\lambda^j \gamma_0, \gamma_{\max}\}^\alpha} \right)^2 \right) \stackrel{\text{def}}{=} \tilde{p}_j. \tag{4.6}$$

where $CDF_{\chi^2(n)}$ is the cumulative density function of a chi-squared distribution with n degrees of freedom.

The numerical illustration was done with the following nonlinear least squares problem defined using the well-known Rosenbrock function:

$$f(x,y) = \frac{1}{2}\left(\|x-1\|^2 + 100\|y-x^2\|^2 \right) = \frac{1}{2}\|F(x,y)\|^2.$$

The minimizer of this problem is $(x^*, y^*)^\top = (1,1)^\top$.

Algorithm 4.1 was initialized with $x_0 = (1.2, 0)^\top$ and $\gamma_0 = 1$. The algorithmic parameters were set to $\eta_1 = \eta_2 = 10^{-3}$, $\gamma_{\min} = 10^{-6}$, and $\lambda = 2$. The stopping criterion used is $\gamma_j > \gamma_{\max}$ where $\gamma_{\max} = 10^6$. We used $\alpha = 1/2$, $\sigma_j = \sigma = 10$ $\forall j$, and $\kappa_{eg} = 100$ for the random gradient model.

Figure 4.1 depicts the average, over 60 runs of Algorithm 4.1, of the objective function values, the absolute errors of the iterates, and the percentages of successful iterations, using, across all iterations, the three choices $p_j = 1$, $p_j = \tilde{p}_j$, and $p_j = p_{\min}$. In the last case, $p_{\min}$ is an underestimation of p_j^* given by:

$$p_{\min} = CDF_{\chi^2(n)}^{-1}\left(\left(\frac{\kappa_{eg}}{\sigma \gamma_{\max}^\alpha} \right)^2 \right) = 5 \cdot 10^{-3}.$$

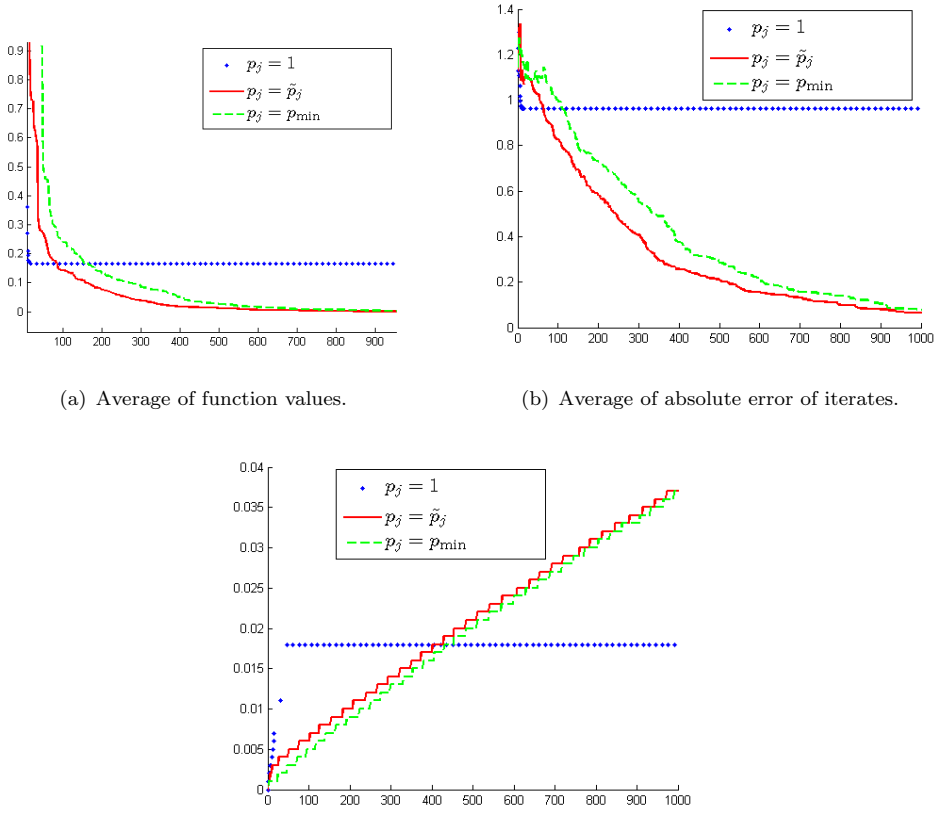
The final objective function values and the relative final errors are shown in Table 4.1 for the first three runs of the algorithm. One can see that the use of $p_j = \tilde{p}_j$ leads to a better performance than $p_j = p_{\min}$ (because $\tilde{p}_j \geq p_{\min}$ is a better bound for p_j^* than $p_{\min}$ is).

In the case where $p_j = 1$, Algorithm 4.1 exhibits a performance worse than for the two other choices of p_j. The algorithm stagnated after some iterations, and could not approximate the minimizer with a descent accuracy. In this case, γ_j is increasing along the iterations, and thus it becomes very large after some iterations while the step $s^j \sim 1/\gamma_j^2$ becomes very small.

Other numerical experiments (not reported here) have shown that, when the error on the gradient is small ($\sigma \ll 1$), the two versions $p_j = \tilde{p}_j$ and $p_j = 1$ give almost the same results, and this is

run number	1	2	3
$\|(x,y)-(x^*,y^*)\|/\|(x^*,y^*)\|$ $(p_j=1)$	1.0168	0.3833	0.7521
$f(x,y)$ $(p_j=1)$	0.5295	0.0368	1.47
$\|(x,y)-(x^*,y^*)\|/\|(x^*,y^*)\|$ $(p_j=\tilde{p}_j)$	0.0033	0.0028	0.0147
$f(x,y)$ $(p_j=\tilde{p}_j)$	2.6474e-6	1.9778e-6	4.3548e-5
$\|(x,y)-(x^*,y^*)\|/\|(x^*,y^*)\|$ $(p_j=p_{\min})$	0.1290	0.1567	0.0068
$f(x,y)$ $(p_j=p_{\min})$	0.0036	0.0059	9.1426e-6

TABLE 4.1: For three different runs of Algorithm 4.1, the table shows the values of the objective function and relative error of the solution found for the three choices $p_j = 1$, $p_j = \tilde{p}_j$, and $p_j = p_{\min} = 5 \cdot 10^{-3}$.



(a) Average of function values.

(b) Average of absolute error of iterates.

(c) Average percentage of successful iterations.

FIGURE 4.1: Average results of Algorithm 4.1 for 60 runs when using probabilities $p_j = 1$ (dotted line), $p_j = \tilde{p}_j$ (solid line), and $p_j = p_{\min}$ (dashed line).

consistent with the theory because when $\sigma \to 0$, from (4.6),

$$\tilde{p}_j \;\to\; CDF^{-1}_{\chi^2(n)}(\infty) \;=\; 1.$$

Note that, on the other extreme, when the error on the gradient is big ($\sigma \gg 1$), version $p_j = \tilde{p}_j$ approaches version $p_j = p_{\min}$ since $\tilde{p}_j \;\simeq\; p_{\min}$.

4.4.2 Expensive gradient case

Let us assume that, in practice, for a given problem, one has two routines for gradient calculation. The first routine computes the exact gradient and is expensive. The second routine is less expensive but computes only an approximation of the gradient. The model gradient results from a call to either routine. In this section, we propose a technique to choose the probability of calling the exact gradient which makes our approach applicable.

Algorithm 4.2: Algorithm to determine when to call the exact gradient

Initialization

Choose the constant $p_{\min} \in (0, 1)$ ($p_{\min}$ is the lower bound of all the probabilities p_j^*).

For a chosen probability $\bar{p}_j$ such that $\bar{p}_j \geq p_{\min}$

1. Sample a random variable $U \sim \mathcal{U}([0, 1/\bar{p}_j])$, independently from F_{j-1}^M, and $\mathcal{U}([0, 1/\bar{p}_j])$ is the uniform distribution on the interval $[0, 1/\bar{p}_j]$.

   1.1 If $U \leq 1$, compute g_{M_j} using the routine which gives the exact gradient.

   1.2 Otherwise, compute g_{M_j} using the routine which gives an approximation of the exact gradient.

Lemma 4.7. *If we use Algorithm 4.2 to compute the model gradient at the j-th iteration of Algorithm 4.1, then we have $p_j^* \geq \bar{p}_j \geq p_{\min}$.*

Proof. By using inclusion of events, we have that:

$$\begin{aligned} p_j^* &= \mathbb{P}\left(\|g_{M_j} - J(x^j)^\top F(x^j)\| \leq \frac{\kappa_{eg}}{\Gamma_j^\alpha} \middle| F_{j-1}^M \right) \\ &\geq \mathbb{P}\left(\|g_{M_j} - J(x^j)^\top F(x^j)\| = 0 \middle| F_{j-1}^M \right) \end{aligned}$$

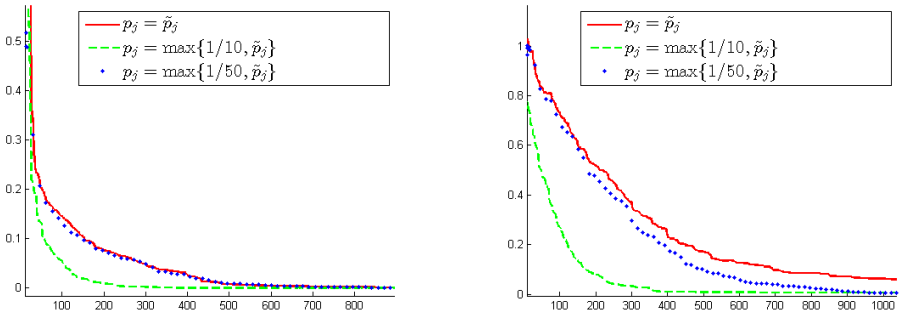
and from Algorithm 4.2 we conclude that:

$$\mathbb{P}\left(\|g_{M_j} - J(x^j)^\top F(x^j)\| = 0 \middle| F_{j-1}^M \right) \geq \mathbb{P}(U \leq 1) = \frac{1}{1/\bar{p}_j},$$

and thus $p_j^* \geq \frac{1}{1/\bar{p}_j} \geq p_{\min}$. □

For the experiments we use the same test function and the same parameters as in Section 4.4.1. In Step 1.2 of Algorithm 4.2, we set the model gradient g_{M_j} to the exact gradient of the function plus a Gaussian noise sampled from $N(0, 10I)$. Across all iterations, we use Algorithm 4.2 to compute g_{M_j} with the three following choices of $\bar{p}_j$:

- $\bar{p}_j = 1/10$, i.e., at iteration j the model gradient coincides with the exact gradient with probability equal to $\bar{p}_j = 1/10$. Moreover, we have $p_j^* \geq \tilde{p}_j$, where $\tilde{p}_j$ is the same as in (4.6), and thus one can choose $p_j = \max\{1/10, \tilde{p}_j\}$.
- $\bar{p}_j = 1/50$, with the same analysis as before and one can choose $p_j = \max\{1/50, \tilde{p}_j\}$.
- $\bar{p}_j \simeq 0$ ($\bar{p}_j = 10^{-10}$ in the experiment below), i.e., at iteration j the probability that the model gradient coincides with the exact gradient is very small. Thus one can choose $p_j = \tilde{p}_j$.



(a) Average of function values.

(b) Average of absolute error of iterates.

FIGURE 4.2: Average results of Algorithm 4.1 for 60 runs when using probabilities $p_j = \tilde{p}_j$ (solid line), $p_j = \max\{1/10, \tilde{p}_j\}$ (dotted line), and $p_j = \max\{1/50, \tilde{p}_j\}$ (dashed line).

Figure 4.2 depicts the average of the function values and the absolute error of the iterates over 60 runs of Algorithm 4.1 when using the three choices of the probability p_j. As expected, the better the quality of the model is the more efficient the Algorithm 4.1 is (less iterations are needed to 'converge' in the sense of sufficiently reducing the objective function value and absolute error). We can clearly see that Algorithm 4.1 using the models for which $p_j = \max\{1/10, \tilde{p}_j\}$ provides a better approximation to the minimizer of the objective function than using the models for which $p_j = \max\{1/50, \tilde{p}_j\}$, and this latter one is better than the case when $p_j = \tilde{p}_j$.

The main contribution of this chapter was to propose a variant of Levenberg-Marquardt method to deal with the nonlinear least squares problems for which the exact gradient is not available and we have only a probabilistic models. We illustrated our new approach with a basic numerical application using Rosenbrock function. In the next chapter, we present the application of our approach to data assimilation problems, more precisely we will show that solving 4DVAR problem using EnKS as linear solver is equivalent to approximating derivatives in random fashion. Then we give a variant of algorithm 4.1 to solve the 4DVAR problem while ensuring the global convergence. Moreover we illustrate numerically our approach using Lorenz 63 equations as a forecast model in 4DVAR problem.

Chapter 5

Probabilistic methods for 4DVAR problems (ensemble based methods)

The aim of this chapter is to present the application of the approach developed in the previous chapter to data assimilation problems (4DVAR). We will show that solving 4DVAR problem, using EnKS as linear solver, is equivalent to approximating derivatives in random fashion, which renders our approach sound in the case of hybridization of 4DVAR and ensemble-based methods. Combinations of ensemble (EnKF/EnKS and their variants) and variational (3DVAR/4DVAR) approaches have become of considerable recent interest in data assimilation [17, 18, 58, 107, 119, 126]. In [126] and [107], it was proposed to use gradient methods in the span of the ensemble to solve the 3DVAR problem. In [17, 107] the authors propose to add regularization and use ensemble method approaches to minimize the nonlinear objective function over linear combinations of the ensemble. The authors in [81, 82] combine ensembles with strong-constraint 4DVAR and perform the minimization in the observation space. The proposed approach in [18] extends the method of [17] to strong-constraint 4DVAR, and the authors scale the ensemble to approximate the derivatives (tangent operators) as in [107]. They call their approach bundle variant which is the same as using finite differences to approximate derivatives. Here, we use the same technique to approximate the derivatives, and we propose also an other different implementation which relies on the scaling of the ensemble, but different from the bundle variant. The approach proposed in [18] nests the minimization loop for the strong-constraint 4DVAR objective function inside the ensemble and performs the minimization over the span of the ensemble, rather than nesting ensemble as a linear solver inside the 4DVAR minimization loop over the full state space as we will present here.

In this chapter, we propose to use the EnKS as linear least squares solver for weak-constraint 4DVAR problem (2.34). The ensemble approach is naturally parallel over the ensemble members. The proposed approach uses finite differences from the ensemble or scale the covariances to avoid using tangent and adjoint operators. We will present a version of Levenberg-Marquardt method,

to solve the general nonlinear least squares (2.34), and to use EnKS to approximate the solution of the linearized subproblem. The method that we will present is suitable for the large dimension problems. The method needs only a matrix vector products.

The reminder of this chapter is organized as follows, we begin by explaining how to approximate the solution of the linearized subproblem arising in Levenberg-Marquardt method using the ensembles (see Section 5.1). Next, we show that solving 4DVAR problem using EnKS as linear solver is equivalent to consider subproblems with random gradients. Finally, we give a numerical illustration using Lorenz 63 equations as a forecast model in 4DVAR problem (see section 5.2).

5.1 4DVAR by ensemble Kalman smoother

We recall the least squares problem to be solved:

$$\begin{aligned}\min_{x_0,\ldots,x_p\in\mathbf{R}^n} f(x_0,\ldots,x_p) &= \frac{1}{2}\left(\|x_0-x_b\|^2_{B^{-1}}+\sum_{k=1}^{p}\|x_k-\mathcal{M}_k(x_{k-1})\|^2_{Q_k^{-1}}\right. \\ &+ \left.\sum_{k=0}^{p}\|\mathcal{H}_k(x_k)-y_k\|^2_{R_k^{-1}}\right).\end{aligned} \tag{5.1}$$

5.1.1 Levenberg-Marquardt and Ensemble Kalman smoother method (LM-EnKS)

When applying the Gauss-Newton algorithm to solve the problem in (5.1). This latter problem is solved iteratively by linearization. At iteration j, one solves the following linearized subproblem for the increments $\delta x^j_{0:p}$:

$$\begin{aligned}\min_{\delta x^j_{0:p}}\frac{1}{2}&\left(\left\|x_0^j+\delta x_0^j-x_b\right\|^2_{B^{-1}}+\sum_{k=1}^{p}\left\|x_k+\delta x_k^j-\mathcal{M}_k\left(x_{k-1}^j\right)-\mathcal{M}_k'\left(x_{k-1}^j\right)\delta x_{k-1}^j\right\|^2_{Q_k^{-1}}\right. \\ &+\left.\sum_{k=0}^{p}\left\|y_k-\mathcal{H}_k\left(x_k^j\right)-\mathcal{H}_k'\left(x_k^j\right)\delta x_k^j\right\|^2_{R_k^{-1}}\right).\end{aligned} \tag{5.2}$$

This is known in data assimilation community as the incremental approach [8, 27, 115]. For the following we omit the index j. Denote:

$$\begin{aligned}&\delta x_b = x_b - x_0,\ m_k = \mathcal{M}_k\left(x_{k-1}\right)-x_k,\ \boldsymbol{M}_k=\mathcal{M}_k'\left(x_{k-1}\right),\ k=1,\ldots,p,\\ &d_k = y_k-\mathcal{H}_k\left(x_k\right),\ k=1,\ldots,p,\ \boldsymbol{H}_k=\mathcal{H}_k'\left(x_k\right),\ k=0,\ldots,p,\end{aligned}$$

and write the auxiliary linear least squares problem (5.2) as:

$$\min_{\delta x_{0:p}}\left(\|\delta x_0-\delta x_b\|^2_{B^{-1}}+\sum_{k=1}^{p}\|\delta x_k-\boldsymbol{M}_k\delta x_{k-1}-m_k\|^2_{Q_k^{-1}}+\sum_{k=0}^{p}\|d_k-\boldsymbol{H}_k\delta x_k\|^2_{R_k^{-1}}\right). \tag{5.3}$$

The function minimized in (5.3) is the same as the one minimized in the KS (the function minimized to find the KS mean) [8]. Hence the solution of (5.3) is then the mean of the smoothing

problem whose evolution is given by:

$$\begin{aligned} \delta x_0 &= \delta x_b + v_b,\ v_b \sim N(0, B), && (5.4)\\ \delta x_k &= \boldsymbol{M}_k \delta x_{k-1} + m_k + v_k,\ v_k \sim N(0, Q_k),\ k = 1, \dots, p, && (5.5)\\ d_k &= \boldsymbol{H}_k \delta x_k + w_k,\ w_k \sim N(0, R_k),\ k = 0, \dots, p. && (5.6) \end{aligned}$$

The Gauss-Newton method with the KS as a linear solver (solver used to solve the subproblem (5.3) at each iteration) is known as the iterated Kalman smoother [8, 46].

We have seen that Gauss-Newton method may diverge, and convergence to a stationary point of (5.1) can be recovered by a control of the step $\delta x_{0:p}$. The Levenberg-Marquardt method (see section 2.3.4.4) controls the step $\delta x_{0:p}$ by adding term of the form $\gamma^2 \|\delta x_{0:p}\|$, or in this section more generally term of the form $\gamma^2 \|\delta x_{0:p}\|^2_{S^{-1}_{0:p}}$, where $S_{0:p}$ is a symmetric positive definite matrix. These term controls the step size as well as rotates the step direction towards the steepest descent, and obtain the Levenberg-Marquardt method $x_{0:p} \leftarrow x_{0:p} + \delta x_{0:p}$, where $\delta x_{0:p}$ is the minimizer of:

$$\begin{aligned} m_j(x_{0:p} + \delta x_{0:p}) = \frac{1}{2} \Bigg(& \|\delta x_0 - \delta x_b\|^2_{B^{-1}} + \sum_{k=1}^{p} \|\delta x_k - \boldsymbol{M}_k \delta x_{k-1} - m_k\|^2_{Q_k^{-1}} \\ & + \sum_{k=0}^{p} \|d_k - \boldsymbol{H}_k \delta x_k\|^2_{R_k^{-1}} \quad + \gamma^2 \sum_{k=0}^{p} \|\delta x_k\|^2_{S_k^{-1}} \Bigg). \end{aligned} \tag{5.7}$$

Similarly as in [68], we interpret the regularization terms:

$$\gamma^2 \sum_{k=0}^{p} \|\delta x_k\|^2_{S_k^{-1}} = \sum_{k=0}^{p} \|\delta x_k\|^2_{(\gamma^{-2} S_k)^{-1}}$$

in (5.7) as arising from additional independent observations:

$$0 = \delta x_k + e_k,\ e_k \sim N\left(0, \gamma^{-2} S_k\right),\ k = 0, \dots, p.$$

Hence the solution of the subproblem (5.7) is equal to the mean of the smoothing problem whose evolution is given by:

$$\begin{aligned} \delta x_0 &= \delta x_b + v_b,\ v_b \sim N(0, B),\\ \delta x_k &= \boldsymbol{M}_k \delta x_{k-1} + m_k + v_k,\ v_k \sim N(0, Q_k),\ k = 1, \dots, p,\\ d_k &= \boldsymbol{H}_k \delta x_k + w_k,\ w_k \sim N(0, R_k),\ k = 0, \dots, p, && (5.8)\\ 0 &= \delta x_k + e_k,\ e_k \sim N\left(0, \gamma^{-2} S_k\right),\ k = 0, \dots, p. && (5.9) \end{aligned}$$

To approximately solve the subproblem (5.7), we propose to use the EnKS as a linear solver instead of the KS (which solve exactly the subproblem and needs the tangent and adjoint operators).

Since in EnKS the state covariance determines the spread of the ensemble, and we may want to work with ensembles with a very small spread to avoid linearization by tangent operators, we

use covariances scaled by a parameter $t > 0$:

$$\frac{1}{2}\left(\|\delta x_0 - \delta x_b\|^2_{(tB)^{-1}} + \sum_{k=1}^{p} \|\delta x_k - \boldsymbol{M}_k \delta x_{k-1} - m_k\|^2_{(tQ_k)^{-1}} \right.$$
$$\left. + \sum_{k=0}^{p} \|d_k - \boldsymbol{H}_k \delta x_k\|^2_{(tR_k)^{-1}} \quad + \gamma^2 \sum_{k=0}^{p} \|\delta x_k\|^2_{(tS_k)^{-1}} \right). \tag{5.10}$$

The value of the parameter t does not change the minimizer of (5.7), but it will affect the operation of the EnKS.

The minimizer $\delta x_{0:p}$ of (5.10) (or equivalently (5.7)) is then the mean of the smoothing problem whose evolution is given by:

$$\begin{aligned}
\delta x_0 &= \delta x_b + v_b,\ v_b \sim N(0, tB), \\
\delta x_k &= \boldsymbol{M}_k \delta x_{k-1} + m_k + v_k,\ v_k \sim N(0, tQ_k),\ k = 1, \ldots, p \\
d_k &= \boldsymbol{H}_k \delta x_k + w_k,\ w_k \sim N(0, tR_k),\ k = 0, \ldots, p, \qquad (5.11) \\
0 &= \delta x_k + e_k,\ e_k \sim N\left(0, \frac{t}{\gamma^2} S_k\right),\ k = 0, \ldots, p. \qquad (5.12)
\end{aligned}$$

From section 2.2.4, and following the spirit of Algorithm 2.4, the EnKS method, with scaled covariances, to approximate the minimizer of (5.10) gives:

1. Generate the initial ensemble $\left[\delta x^1_{0|-1}, \ldots, \delta x^N_{0|-1}\right] = \left[\delta x^l_{0|-1}\right]$, by sampling $\delta x^l_{0|-1} \sim N(\delta x_b, tB)$, where $l = 1, \ldots, N$ is the ensemble member index.
2. For $k = 0, 1, \ldots, p$
   (a) With $\left[\delta x^l_{0:k|k-1}\right]$ already computed, Bayesian update for the observation (5.11):

$$E_\ell = \left[e^1_\ell, \ldots, e^N_\ell\right],\ e^l_\ell = \delta x^l_{\ell|k-1} - \frac{1}{N}\sum_{i=1}^{N} \delta x^i_{\ell|k-1},\ \ell = 0, \ldots, k,\ l = 1, \ldots, N$$
$$Z_k = \left[z^1_k, \ldots, z^N_k\right],\ z^l_k = \boldsymbol{H}_k \delta x^l_{k|k-1} - \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{H}_k \delta x^i_{k|k-1},\ l = 1, \ldots, N \tag{5.13}$$
$$D_k = \left[d^1_k, \ldots, d^N_k\right],\ d^l_k = d_k - w^l_k - \boldsymbol{H}_k \delta x^l_{k|k-1},\ w^l_k \sim N\left(0, tR_k\right),\ l = 1, \ldots, N, \tag{5.14}$$

$$P^N_{0:k,0:k|k-1} = \frac{1}{N-1} \begin{bmatrix} E_0 \\ \vdots \\ E_k \end{bmatrix} \left[E_0^\top \ldots E_k^\top\right].$$

Then we get the update formula:

$$\begin{aligned}\begin{bmatrix} \delta x^l_{0|k} \\ \vdots \\ \delta x^l_{k|k} \end{bmatrix} &= \begin{bmatrix} \delta x^l_{0|k-1} \\ \vdots \\ \delta x^l_{k|k-1} \end{bmatrix} + P^N_{0:k,0:k|k-1} \boldsymbol{H}_k^\top (tR_k + \boldsymbol{H}_k P^N_{0:k,0:k|k-1} \boldsymbol{H}_k^\top)^{-1} \\ &\quad \left(d_k - w^l_k - \boldsymbol{H}_k \delta x^l_{k|k-1}\right) \qquad (5.15) \\ &= \begin{bmatrix} \delta x^l_{0|k-1} \\ \vdots \\ \delta x^l_{k|k-1} \end{bmatrix} + \begin{bmatrix} E_0 \\ \vdots \\ E_k \end{bmatrix} \frac{Z_k^\top t^{-1} R_k^{-1}}{N-1} \\ &\quad \left[I - \frac{1}{N-1} Z_k \left(I + \frac{Z_k^\top t^{-1} R_k^{-1} Z_k}{N-1} \right)^{-1} Z_k^\top t^{-1} R_k^{-1} \right] d^l_k.\end{aligned}$$

(b) Bayesian update for the regularization (5.12) is similar but simpler, taking the identity for $\boldsymbol{H}_k$, 0 for d_k, and $\frac{t}{\gamma^2} S_k$ for R_k. It might be implemented by a call to the same subroutine as for the Bayesian update for the observation y_k.

$$\begin{aligned} E^{\mathbf{1}}_\ell &= \left[e^{\mathbf{11}}_\ell, \ldots, e^{N\mathbf{1}}_\ell \right], \; e^{l\mathbf{1}}_\ell = \delta x^{l\mathbf{1}}_{\ell|k} - \frac{1}{N} \sum_{i=1}^N \delta x^{i\mathbf{1}}_{\ell|k}, \; \ell = 0, \ldots, k, \; l = 1, \ldots, N \\ Z^{\mathbf{1}}_k &= \left[z^{\mathbf{11}}_k, \ldots, z^{N\mathbf{1}}_k \right], \; z^{l\mathbf{1}}_k = \delta x^{l\mathbf{1}}_{k|k} - \frac{1}{N} \sum_{i=1}^N \delta x^{i\mathbf{1}}_{k|k}, \; l = 1, \ldots, N \\ D^{\mathbf{1}}_k &= \left[d^{\mathbf{11}}_k, \ldots, d^{N\mathbf{1}}_k \right], \; d^{l\mathbf{1}}_k = w^{l\mathbf{1}}_k - \delta x^{l\mathbf{1}}_{k|k}, \; w^{l\mathbf{1}}_k \sim N\left(0, \frac{t}{\gamma^2} S_k\right), \; l = 1, \ldots, N. \end{aligned}$$

$$P^N_{0:k,0:k|k} = \frac{1}{N-1} \begin{bmatrix} E^{\mathbf{1}}_0 \\ \vdots \\ E^{\mathbf{1}}_k \end{bmatrix} \left[E^{\mathbf{1}\top}_0 \ldots E^{\mathbf{1}\top}_k \right].$$

We get the update formula:

$$\begin{aligned}\begin{bmatrix} \delta x^{l\mathbf{1}}_{0|k} \\ \vdots \\ \delta x^{l\mathbf{1}}_{k|k} \end{bmatrix} &= \begin{bmatrix} \delta x^l_{0|k} \\ \vdots \\ \delta x^l_{k|k} \end{bmatrix} + P^N_{0:k,0:k|k} \left(\frac{t}{\gamma^2} S_k + P^N_{0:k,0:k|k} \right)^{-1} \left(w^{l\mathbf{1}}_k - \delta x^l_{k|k} \right) \\ &= \begin{bmatrix} \delta x^l_{0|k} \vdots \\ \delta x^l_{k|k} \end{bmatrix} + \begin{bmatrix} E^{\mathbf{1}}_0 \\ \vdots \\ E^{\mathbf{1}}_k \end{bmatrix} \frac{Z_k^{\mathbf{1}\top} \frac{\gamma^2}{t} S_k^{-1}}{N-1} \qquad (5.16) \\ &\quad \left[I - \frac{1}{N-1} Z^{\mathbf{1}}_k \left(I + \frac{Z_k^{\mathbf{1}\top} \frac{\gamma^2}{t} S_k^{-1} Z^{\mathbf{1}}_k}{N-1} \right)^{-1} Z_k^{\mathbf{1}\top} \frac{\gamma^2}{t} S_k^{-1} \right] d^{l\mathbf{1}}_k,\end{aligned}$$

(c) While $k \leq p-1$, advance the ensemble members in time by applying the linearized model $\boldsymbol{M}_{k+1}$ and sampling the model error:

$$\delta x^l_{k+1|k} = \boldsymbol{M}_{k+1}\delta x^{l\mathbf{1}}_{k|k} + m_{k+1} + v^l_{k+1},\ v^l_{k+1} \sim N\left(0, tQ_{k+1}\right). \tag{5.17}$$

3. The approximation of the minimizer of (5.7) (or equivalently the minimizer of (5.10)) is $\frac{1}{N}\sum_{l=1}^{N} \delta x^{l\mathbf{1}}_{0:p|p}$.

5.1.2 Finite differences and the fully nonlinear method

The derivatives $\boldsymbol{M}_k = \mathcal{M}'_k(x_{k-1}),\ k \in \{1,\dots,p\}$, and $\boldsymbol{H}_k = \mathcal{H}'_k(x_k),\ k \in \{0,\dots,p\}$, of the operators $\mathcal{M}_k$ and $\mathcal{H}_k$ only occur in the evaluation of matrix-vector products. Thus, we can replace the derivatives by finite differences involving only the evaluation of the original operators, obviating the need for tangential operators. Substituting

$$\mathcal{M}'_k\left(x_{k-1}\right)\delta x^{l\mathbf{1}}_{k-1|k-1} \approx \frac{\mathcal{M}_k\left(x_{k-1} + \tau\delta x^{l\mathbf{1}}_{k-1|k-1}\right) - \mathcal{M}_k\left(x_{k-1}\right)}{\tau} \text{ with } \tau > 0, \tag{5.18}$$

in (5.17) gives:

$$\begin{aligned}\delta x^l_{k|k-1} \quad = \quad & \frac{\mathcal{M}_k\left(x_{k-1} + \tau\delta x^{l\mathbf{1}}_{k-1|k-1}\right) - \mathcal{M}_k\left(x_{k-1}\right)}{\tau} + \left[\mathcal{M}_k\left(x_{k-1}\right) - x_k\right] + v^l_k, \\ & l = 1,\dots,N,\end{aligned} \tag{5.19}$$

and substituting

$$\mathcal{H}'_k\left(x_k\right)\delta x^l_{k|k-1} \approx \frac{\mathcal{H}_k\left(x_k + \tau\delta x^l_{k|k-1}\right) - \mathcal{H}_k\left(x_k\right)}{\tau} \tag{5.20}$$

in (5.13) and (5.14) gives:

$$\begin{aligned} z^l_k &= \frac{\mathcal{H}_k\left(x_k + \tau\delta x^l_{k|k-1}\right) - \mathcal{H}_k\left(x_k\right)}{\tau} - \frac{1}{N}\sum_{i=1}^{N}\frac{\mathcal{H}_k\left(x_k + \tau\delta x^i_{k|k-1}\right) - \mathcal{H}_k\left(x_k\right)}{\tau} \\ &= \frac{1}{\tau}\left[\mathcal{H}_k\left(x_k + \tau\delta x^l_{k|k-1}\right) - \frac{1}{N}\sum_{i=1}^{N}\mathcal{H}_k\left(x_k + \tau\delta x^i_{k|k-1}\right)\right],\ l = 1,\dots,N, \end{aligned} \tag{5.21}$$

$$d^l_k = \left[y_k - \mathcal{H}_k\left(x_k\right)\right] - w^l_k - \frac{\mathcal{H}_k\left(x_k + \tau\delta x^l_{k|k-1}\right) - \mathcal{H}_k\left(x_k\right)}{\tau},\ l = 1,\dots,N. \tag{5.22}$$

When $\tau \to 0$, the resulting method is asymptotically equivalent to the method with the derivatives (see chapter 7 for the proof). In the case when $\tau = 1$, and $t = 1$ we recover the standard EnKS as presented in [43]. Indeed, (5.19) becomes:

$$\begin{aligned}\delta x^l_{k|k-1} &= \mathcal{M}_k\left(x_{k-1} + \delta x^{l\mathbf{1}}_{k-1|k-1}\right) - \mathcal{M}_k\left(x_{k-1}\right) + \left[\mathcal{M}_k\left(x_{k-1}\right) - x_k\right] + v^l_k \\ &= \mathcal{M}_k\left(x_{k-1} + \delta x^{l\mathbf{1}}_{k-1|k-1}\right) - x_k + v^l_k.\end{aligned}$$

Noting that $x_{k-1} + \delta x^{\boldsymbol{l1}}_{k-1|k-1} = x^{\boldsymbol{l1}}_{k-1|k-1}$, (5.19) becomes:

$$x^l_{k|k-1} = \mathcal{M}_k \left(x^{\boldsymbol{l1}}_{k-1|k-1} \right) + v^l_k, \tag{5.23}$$

which is exactly the same as advancing the ensemble member l in the usual way, as in [43].

Similarly, noting that $x_k + \delta x^l_{k|k-1} = x^l_{k|k-1}$, (5.22) becomes with $\tau = 1$:

$$\begin{aligned} z^l_k &= \mathcal{H}_k \left(x_k + \delta x^l_{k|k-1} \right) - \mathcal{H}_k \left(x_{k|k-1} \right) - \frac{1}{N} \sum_{i=1}^{N} \left[\mathcal{H}_k \left(x_k + \delta x^i_{k|k-1} \right) - \mathcal{H}_k \left(x_{k|k-1} \right) \right] \\ &= \mathcal{H}_k \left(x^l_{k|k-1} \right) - \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}_k \left(x^i_{k|k-1} \right), \end{aligned} \tag{5.24}$$

and (5.21) becomes:

$$\begin{aligned} d^l_k &= \left[y_k - \mathcal{H}_k \left(x_k \right) \right] - w^l_k - \mathcal{H}_k \left(x_k + \delta x^l_{k|k-1} \right) + \mathcal{H}_k \left(x_k \right) \\ &= y_k - w^l_k - \mathcal{H}_k \left(x^l_{k|k-1} \right), \end{aligned} \tag{5.25}$$

which is the same as presented in [43].

Note that, in the case when $\tau \neq 1$, the dynamical model operator is evaluated two times (we need to evaluate it in two different points) to approximate the derivatives in EnKS. However in the case when $\tau = 1$, one evaluation of this operator is needed. In addition when the covariances are scaled with $t > 0$ so small, $\tau = 1$ may be sufficient to approximate well the derivatives. Therefore, we conclude that the cost in term of the model operator evaluations is less in the case where the covariances are scaled with small t ($t << 1$) than in the case where the covariances are not scaled.

The pseudo-code for the Levenberg-Marquardt method, using EnKS as linear solver, to solve 4DVAR problem (5.1) is given in Algorithm 5.1.

5.2 LM-EnKS and Levenberg-Marquardt based on probabilistic models

In Algorithm 5.1, we use EnKS method to approximate the subproblem solution. When the ensemble size is infinite the method gives the exact solution of the linearized subproblem (see Chapter 7), hence the use of a finite ensemble can be seen in turn as an approximation of derivatives. In this section we will quantify probabilistically the error between the derivatives approximation made when using EnKS and the exact derivatives. Then we give a version of Algorithm 4.1 for the solution of the 4DVAR problem (5.1) when using EnKS as the linear solver, and adding the regularization.

Algorithm 5.1: Levenberg-Marquardt EnKS algorithm

Initialization

Choose the constants $\eta \in (0,1)$, $\gamma_{\min}, \gamma_{max} > 0$,$\tau \in (0,1]$, $t \in (0,1]$ and $\lambda > 1$. Select $x^0_{0:p}$ and $\gamma_0 \in [\gamma_{\min}, \gamma_{\max}]$. Choose all the parameters related to solving the 4DVAR problem (5.1) using EnKS as the linear solver.

For $j = 0, 1, 2, \ldots,$ **and while** $\gamma_j \leq \gamma_{max}$

1. Compute the increment ensemble $[\delta x^l_{0:p|p}]_{l=1}^N$ using (5.16), and by approximating the derivatives as explained in section 5.1.2. Let
$$s^j_{0:p} = \frac{1}{N}\sum_{l=1}^{N} \delta x^l_{0:p|p}.$$
2. Compute $\rho_j = \frac{f(x^j_{0:p}) - f(x^j_{0:p}+s^j_{0:p})}{m_j(x^j_{0:p}) - m_j(x^j_{0:p}+s^j_{0:p})}$, where f is the nonlinear least squares model in (5.1) and m_j is the model in (5.7).
3. If $\rho_j \geq \eta_1$, then set $x^{j+1}_{0:p} = x^j_{0:p} + s^j_{0:p}$ and $\gamma_{j+1} = \max(\gamma_j, \gamma_{\max})$. Otherwise, set $x^{j+1}_{0:p} = x^j_{0:p}$ and $\gamma_{j+1} = \lambda\gamma_j$.

For simplicity, we now rewrite the linear system (5.4)-(5.6) as:

$$\delta X = \delta X_b + V,\ V \sim N(0, B_V), \tag{5.26}$$
$$D = H\delta X + W,\ W \sim N(0, R), \tag{5.27}$$

where

$$\begin{aligned}
\delta X &= [\delta x_0; \cdots; \delta x_p] \text{ is the joint state of the states } \delta x_0, \ldots, \delta x_p,\\
D &= [d_0; d_1; \cdots; d_p],\\
\delta X_b &= [\delta x_b; \boldsymbol{M}_1\delta x_b + m_1; \boldsymbol{M_2}(\boldsymbol{M}_1\delta x_b + m_1) + m_2; \cdots; \boldsymbol{M}_p(\cdots \boldsymbol{M}_1\delta x_b + m_1 \cdots) + m_p],\\
\mathbf{H} &= \mathrm{diag}(\boldsymbol{H}_0, \ldots, \boldsymbol{H}_p) \text{ is the joint observation operator},\\
V &= [v_b; \boldsymbol{M}_1 v_b + v_1; \boldsymbol{M}_2(\boldsymbol{M}_1 v_b + v_1) + v_2; \cdots; \boldsymbol{M}_p(\cdots \boldsymbol{M}_1 v_b + v_1 \cdots) + v_p],\\
B_V &= \mathrm{cov}(V),\ W = [w_0; w_1; \cdots; w_p],\ \text{and}\ R = \mathrm{cov}(W).
\end{aligned}$$

To simplify it even more, we make the change of variables $U = \delta X - \delta X_b$, and then (5.26)–(5.27) becomes:

$$U \sim N(0, B_V) \tag{5.28}$$
$$D - \mathbf{H}\delta X_b = \mathbf{H}U + W,\ W \sim N(0, R), \tag{5.29}$$

and the linear least squares problem (5.3) becomes, with $u = \delta X - \delta X_b$:

$$\min_{u \in \mathbf{R}^{n(p+1)}} \frac{1}{2} \left(\|u\|^2_{B_V^{-1}} + \|D - \mathbf{H}\delta X_b - \mathbf{H}u\|^2_{R^{-1}} \right). \tag{5.30}$$

To approximate the solution of the problem (5.30), we use a centered EnKS, and approximate the derivatives by finite differences. We explain in the following how to built this approximation. Let us denote by l the ensemble members index, running over $l = 1, \ldots, N$, where N is the ensemble size. We sample an ensemble $\left[\tilde{U}^l_{0:p}\right]_{l=1}^N$ from $N(0, B_V)$ as follows:

We sample $\left[v_b^l\right]_{l=1}^N$ according to $N(0, B)$, $\left[v_1^l\right]_{l=1}^N$ according to $N(0, Q_1)$, ..., $\left[v_p^l\right]_{l=1}^N$ according to $N(0, Q_p)$, and then we set $\left[\tilde{U}^l_{0:p}\right]_{l=1}^N$ as follows:

For $l \in \{1, \ldots, N\}$, $\tilde{U}_0^l = v_b^l$, $\tilde{U}_1^l = \boldsymbol{M}_1 v_b^l + w_1^l$, ..., $\tilde{U}_p^l = \boldsymbol{M}_p(\cdots \boldsymbol{M}_1 v_b^l + v_1^l \cdots) + v_p^l$.

When there is no confusion on the notation we omit the subscripts. Let

$$\bar{\tilde{U}}_{0:p} = \frac{1}{N}\sum_{l=1}^N \tilde{U}^l_{0:p} \text{ and } B^N = \frac{1}{N-1}\sum_{l=1}^N \left(\tilde{U}^l_{0:p} - \bar{\tilde{U}}_{0:p}\right)\left(\tilde{U}^l_{0:p} - \bar{\tilde{U}}_{0:p}\right)^\top$$

be the empirical mean and covariance of the ensemble $\tilde{U}^l_{0:p}$, respectively. One has:

$$B^N = CC^\top, \text{ where } C = \frac{1}{\sqrt{N-1}}\left[\tilde{U}^1 - \bar{\tilde{U}}, \tilde{U}^2 - \bar{\tilde{U}}, \ldots, \tilde{U}^N - \bar{\tilde{U}}\right].$$

We then build the centered ensemble $\left[U^l\right]_{l=1}^N = \left[\tilde{U}^l - \bar{\tilde{U}}\right]_{l=1}^N$. Note that the empirical mean of the ensemble $\left[U^l\right]_{l=1}^N$ is equal to zero and that its empirical covariance matrix is B^N.

Now one generates the ensemble $\left[U^l_{0:p|p}\right]_{l=1}^N$ as follows:

$$U^l_{0:p|p} = U^l_{0:p} + K^N\left(D - W^l - \mathbf{H}\delta X_b\right),\ l = 1, \ldots, N, \tag{5.31}$$

where W^l is sampled from $N(0, R)$, and

$$K^N = B^N\mathbf{H}^\top\left(R + \mathbf{H}B^N\mathbf{H}^\top\right)^{-1}.$$

In practice, as we already explained in section 2.2.2, the empirical covariance matrix B^N is never computed or stored since to compute the matrix products $B^N\mathbf{H}^\top$ and $\mathbf{H}B^N\mathbf{H}^\top$ only matrix-vector products are needed:

$$\begin{aligned}
B^N\mathbf{H}^\top &= \frac{1}{N-1}\sum_{l=1}^N U^l {U^l}^\top \mathbf{H}^\top = \frac{1}{N-1}\sum_{l=1}^N U^l h_l^\top, \\
\mathbf{H}B^N\mathbf{H}^\top &= \mathbf{H}\frac{1}{N-1}\sum_{l=1}^N U^l {U^l}^\top \mathbf{H}^\top = \frac{1}{N-1}\sum_{l=1}^N h_l h_l^\top, \\
K^N &= \frac{1}{N-1}\sum_{l=1}^N U^l h_l^\top \left(R + \frac{1}{N-1}\sum_{l=1}^N h_l h_l^\top\right)^{-1},
\end{aligned}$$

where $h_l = \mathbf{H}U^l = \left[\boldsymbol{H}_0 U_0^l; \cdots; \boldsymbol{H}_p U_p^l\right]$.

Let $\bar{U}$ and $\bar{W}$ denote the empirical mean of the ensembles $U_{0:p|p}^l$ and W^l, respectively. One has from (5.31):

$$\bar{U} = K^N \left(D - \mathbf{H}\delta X_b - \bar{W}\right). \tag{5.32}$$

5.2.1 The linearized least squares subproblem arising in EnKS

$\bar{U}$ is equal to the KS mean for the smoothing problem whose evolution is given by:

$$\begin{aligned} \tilde{U} &\sim N(0, B^N), \\ \tilde{D} &= \mathbf{H}\tilde{U} + \tilde{W},\ \tilde{W} \sim N(0, R), \ \text{ where } \ \tilde{D} = D - \mathbf{H}\delta X_b - \bar{W}. \end{aligned} \tag{5.33}$$

Hence, for a large N (such that B^N is invertible), $\bar{U}$ is the solution of the following linear least squares problem:

$$\min_{u \in \mathbf{R}^{n(p+1)}} \ \frac{1}{2}\left(\|u\|^2_{(B^N)^{-1}} + \|\mathbf{H}u - \tilde{D}\|^2_{R^{-1}}\right). \tag{5.34}$$

From the above derivation, we conclude that when we use the EnKS (until now with exact derivatives) to approximate the solution of the linearized subproblem (5.3), what is obtained is the solution of the linear least squares problem (5.34). The least squares model in (5.34) can be seen, in turn, as a realization of the following stochastic model:

$$\frac{1}{2}\left(\|u\|^2_{\mathcal{B}^{-1}} + \|\mathbf{H}u - \tilde{\mathcal{D}}\|^2_{R^{-1}}\right), \tag{5.35}$$

where $\mathcal{B}^{-1}$ and $\tilde{\mathcal{D}}$ are random variables, with realizations $\left(B^N\right)^{-1}$ and $\tilde{D}$, respectively. Hence approximating the solution of the linearized subproblem (5.3) using EnKS, is the same as finding a minimizer of a realization of the quadratic random model (5.35). This method which approximates the solution of the linearized subproblem (5.3) using EnKS may diverge. Convergence to a stationary point of (5.1) can be recovered by controlling the size of the step, and one possibility way to do so is to consider the application of the Levenberg-Marquardt method as in Algorithm 2.8. At each step, a regularization term is then added to the model in (5.34):

$$m(x+u) \ = \ \frac{1}{2}\left(\|u\|^2_{(B^N)^{-1}} + \|\mathbf{H}u - \tilde{D}\|^2_{R^{-1}} + \gamma^2\|u\|^2\right), \tag{5.36}$$

which corresponds to adding a regularization term to the model (5.35):

$$M(x+u) \ = \ \frac{1}{2}\left(\|u\|^2_{\mathcal{B}^{-1}} + \|\mathbf{H}u - \tilde{\mathcal{D}}\|^2_{R^{-1}} + \Gamma^2\|u\|^2\right). \tag{5.37}$$

We now provide the details about the solution of (5.36). For this purpose let

$$P^N \ = \ (I - K^N\mathbf{H})B^N.$$

Note that by using the Sherman–Morrison–Woodbury formula one has:

$$P^N \;=\; \left((B^N)^{-1} + \mathbf{H}^\top R^{-1}\mathbf{H}\right)^{-1},$$

in other words, P^N is the inverse of the Hessian of model in (5.34).

Proposition 5.1. *The minimizer of the model (5.36) is* $u^* = \bar{U} - P^N(P^N + (1/\gamma^2)I_n)^{-1}\bar{U}$.

Proof. Since $\bar{U}$ is the solution of problem (5.34), a Taylor expansion around $\bar{U}$ of the model in (5.34) gives:

$$\frac{1}{2}\left(\|u\|^2_{(B^N)^{-1}} + \|\mathbf{H}u - \tilde{D}\|^2_{R^{-1}}\right) \;=\; \frac{1}{2}\left(\|\bar{U}\|^2_{(B^N)^{-1}} + \|\mathbf{H}\bar{U} - \tilde{D}\|^2_{R^{-1}} + \|u - \bar{U}\|^2_{(P^N)^{-1}}\right).$$

Hence, the minimizer of the model (5.36) is the same as the minimizer of

$$\frac{1}{2}\left(\|\bar{U}\|^2_{(B^N)^{-1}} + \|\mathbf{H}\bar{U} - \tilde{D}\|^2_{R^{-1}} + \|u - \bar{U}\|^2_{(P^N)^{-1}} + \gamma^2\|u\|^2\right).$$

and thus given by:

$$u^* \;=\; \left((P^N)^{-1} + \gamma^2 I\right)^{-1}(P^N)^{-1}\bar{U}. \tag{5.38}$$

By using the Sherman–Morrison–Woodbury formula, one has:

$$\left((P^N)^{-1} + \gamma^2 I\right)^{-1} \;=\; P^N - P^N\left(P^N + (1/\gamma^2)I_n\right)^{-1}P^N,$$

which together with (5.38) concludes the proof. □

5.2.2 A derivative-free LM-EnKS

The linearized model and observation operators appear only when acting on a given vector, and therefore they could be efficiently approximated by finite differences (the same way as in Section 5.1.2). The linearized observation operator $\boldsymbol{H}_k = \mathcal{H}_k^{'}(x_k),\ k \in \{0,\ldots,p\}$, appears in the action on the ensemble members and can be approximated by:

$$\boldsymbol{H}_k\delta x_k \;=\; \mathcal{H}_k^{'}(x_k)\delta x_k \;\simeq\; \frac{\mathcal{H}_k\left(x_k + \tau\delta x_k\right) - \mathcal{H}_k\left(x_k\right)}{\tau},$$

where $\tau > 0$ is a finite differences parameter. The linearized model $\boldsymbol{M}_1 = \mathcal{M}_1^{'}(x_0)$ occurs in the action on the vector δx_b, $\boldsymbol{M}_2 = \mathcal{M}_2^{'}(x_1)$ occurs in the action on the vector $\boldsymbol{M}_1\delta x_b$, and so on for $\boldsymbol{M}_3,\ldots,\boldsymbol{M}_p$, and (just for the first two terms) the finite difference approximations are:

$$\begin{aligned}
\boldsymbol{M}_1\delta x_b &= \mathcal{M}_1^{'}(x_0)\delta x_b \;\simeq\; \frac{\mathcal{M}_1(x_0 + \tau\delta x_b) - \mathcal{M}_1(x_0)}{\tau}\\
\boldsymbol{M}_2(\boldsymbol{M}_1\delta x_b + m_1) &= \mathcal{M}_2^{'}(x_1)(\boldsymbol{M}_1\delta x_b + m_1) \;\simeq\; \frac{\mathcal{M}_2\left(x_1 + \tau(M_1\delta x_b + m_1)\right) - \mathcal{M}_2(x_1)}{\tau}\\
&\simeq\; \frac{\mathcal{M}_2\left(x_1 + \mathcal{M}_1(x_0 + \tau\delta x_b) - \mathcal{M}_1(x_0) + \tau m_1\right) - \mathcal{M}_2(x_1)}{\tau}.
\end{aligned}$$

Since our approach is derivative free, we replace all the derivatives of the model and of the observation operators by approximation by finite differences. The quantities using derivatives become then:

$$\begin{aligned}
\hat{h}_l &= \left[\frac{\mathcal{H}_0\left(x_0+\tau U_0^l\right)-\mathcal{H}_0\left(x_0\right)}{\tau};\cdots;\frac{\mathcal{H}_p\left(x_p+\tau U_p^l\right)-\mathcal{H}_p\left(x_p\right)}{\tau}\right] \simeq h_l,\\
\hat{K}^N &= \frac{1}{N-1}\sum_{l=1}^{N} U^l\hat{h}_l^\top\left(R+\frac{1}{N-1}\sum_{l=1}^{N}\hat{h}_l\hat{h}_l^\top\right)^{-1} \simeq K^N,\\
\delta\hat{X}_b &= \left[\delta x_b;\frac{\mathcal{M}_1\left(x_0+\tau\delta x_b\right)-\mathcal{M}_1\left(x_0\right)}{\tau}+m_1;\cdots\right] \simeq \delta X_b, \qquad (5.39)\\
\hat{\mathbf{H}}\delta X_b &= \left[\frac{\mathcal{H}_0\left(x_0+\tau\delta x_b\right)-\mathcal{H}_0\left(x_0\right)}{\tau};\cdots\right] \simeq \mathbf{H}\delta X_b,\\
\hat{U} &= \hat{K}^N(D-\hat{\mathbf{H}}\delta X_b-\bar{V}) \simeq \bar{U},\\
\hat{P}^N &= B^N-\hat{K}^N\frac{1}{N-1}\sum_{l=1}^{N}\hat{h}_l{U^l}^\top \simeq P^N,\\
\hat{u}^* &= \hat{U}-\hat{P}^N\left(\hat{P}^N+(1/\gamma^2)I_n\right)^{-1}\hat{U} \simeq u^*. \qquad (5.40)
\end{aligned}$$

Since $\hat{u}^*$ is an approximation to u^* using finite differences for derivatives, there exists a constant $\zeta>0$, which depends on the second derivatives of the model and observation operators, such that $\|e\|\leq\zeta\tau$, where $e=u^*-\hat{u}^*$. Moreover, from (5.36), u^* is the solution of the normal equations:

$$\left(\left(B^N\right)^{-1}+\mathbf{H}^\top R^{-1}\mathbf{H}+\gamma^2 I\right)u^* = \mathbf{H}^\top R^{-1}\tilde{D},$$

where $H^\top R^{-1}\tilde{D}=\nabla m(x)=g_m$, and thus:

$$\left(\left(B^N\right)^{-1}+\mathbf{H}^\top R^{-1}\mathbf{H}+\gamma^2 I\right)\hat{u}^* = g_m-\left(\left(B^N\right)^{-1}+\mathbf{H}^\top R^{-1}\mathbf{H}+\gamma^2 I\right)e,$$

and so $\hat{u}^*$ can be seen as an inexact solution of the normal equations, with a residual equal to:

$$r = -\left((B^N)^{-1}+\mathbf{H}^\top R^{-1}\mathbf{H}+\gamma^2 I\right)e.$$

We have seen that the solution of the normal equations can be inexact as long as Assumption 4.2.2 is met. The residual r is then required to satisfy $\|r\|\leq\epsilon\|g_m\|$, for some $\epsilon>0$, to fulfill the global convergence requirements of our Levenberg-Marquardt approach, and for this purpose we need the following assumption.

Assumption 5.2.1. The Jacobian of the observation operators $\mathcal{H}_k$, $k=0,\ldots,p$, are uniformly bounded, i.e., there exists $\kappa_H>0$ such that $\|\mathcal{H}_k'(x_k)\|\leq\kappa_H$ for all $k\in\{0,..,p\}$ and for all iterations j.

Proposition 5.2. *Under Assumption 5.2.1. If the finite differences parameter τ is such that*

$$\tau \leq \frac{\epsilon\|g_m\|}{\zeta\left(\|(B^N)^{-1}\|+\kappa_H^2\|R^{-1}\|+\gamma^2\right)}, \qquad (5.41)$$

then $\|r\|\leq\epsilon\|g_m\|$.

Proof. We have:

$$\begin{aligned} \|r\| &\leq \left\|(B^N)^{-1} + \mathbf{H}^\top R^{-1}\mathbf{H} + \gamma^2 I\right\| \|e\| \\ &\leq \left(\|(B^N)^{-1}\| + \kappa_H^2 \|R^{-1}\| + \gamma^2\right) \zeta\tau \;\leq\; \epsilon\|g_m\|. \end{aligned}$$

□

We note that the iteration index j has been omitted from the notation of this section until now. In fact, the point x has been denoting the iterate x^j. Now, from (5.37) the gradient of the stochastic model is $g_{M_j} = -\mathbf{H}^\top R^{-1}\tilde{\mathcal{D}}$ and from (5.30) the exact gradient of the function to minimized in problem (2.38) is $-\mathbf{H}^\top R^{-1}(D - \mathbf{H}\delta X_b)$. Thus,

$$p_j^* \;=\; \mathbb{P}\left(\|\mathbf{H}^\top R^{-1}(D - \mathbf{H}\delta X_b - \tilde{\mathcal{D}})\| \leq \frac{\kappa_{eg}}{\Gamma_j^\alpha} \middle| \mathcal{F}_{j-1}^{\tilde{M}} \right).$$

But we know that $D - \mathbf{H}\delta X_b - \tilde{\mathcal{D}} = \bar{V} = (1/N)\sum_{i=1}^N V_i$, where V_i are i.i.d. and follow $N(0, R)$, and thus $D - \mathbf{H}\delta X_b - \tilde{\mathcal{D}} \sim N(0, R/N)$ and $R^{-1}(D - \mathbf{H}\delta X_b - \tilde{\mathcal{D}}) \sim \frac{R^{-1/2}}{\sqrt{N}} N(0, I)$. Thus

$$\begin{aligned} p_j^* &\geq \mathbb{P}\left(\frac{\kappa_H \|R^{-1/2}\|}{\sqrt{N}} \|N(0,I)\| \leq \frac{\kappa_{eg}}{\Gamma_j^\alpha} \middle| \mathcal{F}_{j-1}^{\tilde{M}} \right) \\ &= \mathbb{P}\left(\|N(0,I)\| \leq \frac{\kappa\sqrt{N}}{\Gamma_j^\alpha} \middle| \mathcal{F}_{j-1}^{\tilde{M}} \right), \end{aligned}$$

where $\kappa = \frac{\kappa_{eg}}{\kappa_H \|R^{-1/2}\|}$. Since $\Gamma_j \leq \min\{\lambda^j \gamma_0, \gamma_{\max}\}$,

$$p_j^* \;\geq\; CDF_{\chi^2(m)}^{-1}\left(\left(\frac{\kappa\sqrt{N}}{\min\{\lambda^j\gamma_0, \gamma_{\max}\}^\alpha} \right)^2 \right) \;\stackrel{\text{def}}{=}\; \tilde{p}_j, \tag{5.42}$$

where $m = \sum_{k=0}^p m_k$, m_k is the size of y_k, and $\gamma_{\max}$ is the tolerance used in the stopping criterion.

Note that $\lim_{N\to\infty} \tilde{p}_j = 1$, thus $\lim_{N\to\infty} p_j^* = 1$, and hence when $N \to \infty$ the gradient approximation using ensemble converges almost surely to the exact gradient.

We are now ready to propose a version of Algorithm 2.8 for the solution of the 4DVAR problem (5.1) when using EnKS as the linear solver, and adding the regularization.

5.2.3 Computational experiments with Lorenz 63 as forecast model

The twin experiment technique is used to evaluate the performance of the of Algorithm 5.2. It can be described as follows: an integration of the model is chosen as the true state, meaning that an initial true state is fixed (truth_0), and then we integrate it over time using the model to obtain the true state at each time k (truth_k). Then, we build the observations y_k by applying the observation operator $\mathcal{H}_k$ to the truth and by adding a Gaussian perturbation $N(0, R_k)$.

Algorithm 5.2: Levenberg-Marquardt based on probabilistic gradient models for data assimilation 4DVAR problem

Initialization

Choose the constants $\eta_1 \in (0,1)$, $\eta_2, \gamma_{\min}, \gamma_{max} > 0$, and $\lambda > 1$. Select x_0 and $\gamma_0 \in [\gamma_{\min}, \gamma_{\max}]$. Choose all the parameters related to solving the 4DVAR problem (5.1) using EnKS as the linear solver.

For $j = 0, 1, 2, \ldots$

1. Choose τ satisfying (5.41). Compute the increment $\hat{u}^*$ using (5.40) and set $\delta x^* = \hat{u}^* + \delta\hat{X}_b$, where $\delta\hat{X}_b$ is computed as in (5.39). Let $s^j = \delta x^*$.

2. Compute $\rho_j = \frac{f(x^j) - f(x^j + s^j)}{m_j(x^j) - m_j(x^j + s^j)}$, where f is the nonlinear least squares model in (5.1) and m_j is the model (5.36).

3. If $\rho_j \geq \eta_1$, then set $x^{j+1} = x^j + s^j$ and

$$\gamma_{j+1} = \begin{cases} \lambda\gamma_j & \text{if } \|g_{m_j}\| < \eta_2/\gamma_j^2, \\ \max\left\{ \frac{\gamma_j}{\lambda \frac{1-p_j}{p_j}}, \gamma_{\min} \right\} & \text{if } \|g_{m_j}\| \geq \eta_2/\gamma_j^2, \end{cases}$$

   where $p_j = \tilde{p}_j$ is computed as in (5.42).
   Otherwise, set $x^{j+1} = x^j$ and $\gamma_{j+1} = \lambda\gamma_j$.

Similarly, the background x_b is sampled from the Gaussian distribution with the mean equal to the initial conditions and the covariance matrix B. Finally we try to retrieve the truth using the observations and the background.

We consider as model in the problem (2.38), Lorenz 63 equations [84], a simple dynamical model with chaotic behavior. The Lorenz equations are given by the nonlinear system:

$$\frac{dx}{dt} = -\sigma(x - y), \quad \frac{dy}{dt} = \rho x - y - xz, \text{ and } \frac{dz}{dt} = xy - \beta z,$$

where $x = x(t)$, $y = y(t)$, $z = z(t)$, and σ, ρ, β are parameters. The state at time t is $X_t = (x(t), y(t), z(t))^\top \in \mathbf{R}^3$. This nonlinear system is discretized using a fourth-order Runge-Kutta method. The parameters σ, ρ, β are chosen as 10, 28, and 8/3 respectively.

The initial truth is set to $(1,1,1)^\top$ and the truth at time k to $\text{truth}_k = \mathcal{M}(\text{truth}_{k-1}) + v_k$, where v_k is sampled from $N(0, Q_k)$ and $\mathcal{M}$ is the model obtained by discretization of Lorenz 63 system. The model error covariance is given by $Q_k = \sigma_q^2 I$ where $\sigma_q = 10^{-4}$. The background mean x_b is sampled from $N(\text{truth}_0, B)$. The background covariance is $B = \sigma_b^2 I$, where $\sigma_b = 1$. The time step is chosen as $dt = 0.11$. The time windows length is $p = 40$. The observation operator is $\mathcal{H}_k = 10I$. At each time i, the observations are constructed as follows: $y_k = \mathcal{H}_k(\text{truth}_k) + w_k$, where w_k is sampled from $N(0, R)$, $R = \sigma_r^2 I$, and $\sigma_r = 1$.

The size of the ensemble used is $N = 400$. Following the spirit of Proposition 4.2.2, the finite difference parameter is set as:

$$\tau_j \;=\; \min\left\{10^{-3}, \frac{\epsilon_j \|g_{m_j}\|}{\zeta\left(\|(B^N)^{-1}\| + \kappa_H^2 \|R^{-1}\| + \gamma_j^2\right)}\right\},$$

where the value of 1 is chosen for the unknown constants ζ and κ_H (see Assumption 5.2.1). In this experimental framework, the model gradient is given by $g_{m_j} = -H^\top R^{-1}\tilde{D} = 10\tilde{D}$, where $\tilde{D}$ is computed according to (5.33). Then, following the spirit of Assumption 4.2.2, ϵ_j is chosen as:

$$\epsilon_j \;=\; \min\left\{\frac{\theta_{in}}{\gamma_j^\alpha}, \sqrt{\beta_{in}\frac{\gamma_j^2}{\kappa_{Jm}^2 + \gamma_j^2}}\right\},$$

where $\beta_{in} = 1/2$, $\theta_{in} = 1$, and $\alpha = 0.5$. The unknown constant κ_{Jm} (see Assumption 4.3.1) is set to 1.

The basic algorithmic parameters are set to $\eta_1 = \eta_2 = 10^{-6}$, $\gamma_{\min} = 10^{-5}$, $\gamma_{\max} = 10^6$, and $\lambda = 8$. The initial regularization parameter is $\gamma_0 = 1$. Finally, we set $\kappa = 1$ in the calculation of $\tilde{p}_j$ given in (5.42).

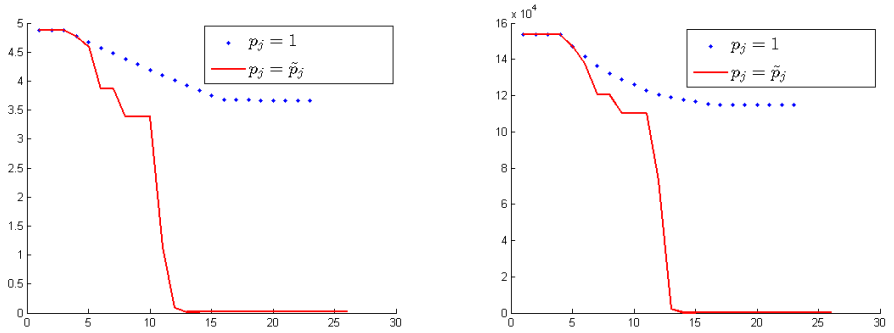
In order to measure the quality of the solutions we use as performance metric the Root Mean Square Error (RMSE), which is defined as follows:

$$\text{RMSE} \;=\; \frac{1}{p}\sum_{k=0}^{p}\text{RSE}_k,$$

where RSE_k is the Root Squares Error at time k given by

$$\text{RSE}_k \;=\; \sqrt{\frac{1}{n}(\text{truth}_k - x_k)^\top(\text{truth}_k - x_k)},$$

where truth_k is the true vector state at time k and x_k is the estimator of the state computed using the algorithm.



(a) The RMSE. (b) Objective function values.

FIGURE 5.1: Results of one run of Algorithm 5.2 when using probabilities $p_j = 1$ (dotted line) and $p_j = \tilde{p}_j$ (solid line).

Figures 5.1(a) and 5.1(b) show, respectively, the RMSE and the objective function values, for one run of Algorithm 5.2, using the choices $p_j = \tilde{p}_j$ and $p_j = 1$ (One run shows well the behavior of the algorithm on this problem and there was no need to take averages over several runs). As it can be seen from these plots, 40 iterations were enough for Algorithm 5.2 using $p_j = \tilde{p}_j$ to reduce the RMSE from 4.88 to 0.019. But when $p_j = 1$ is used, the same 40 iterations were not enough to drive the RMSE to the same value. These results illustrate the importance of using probability $p_j = \tilde{p}_j$ to update the regularization parameter γ.

In this chapter we have explained how to use EnKS to approximately solve the linearized least square subproblem when using Levenberg-Marquardt method to solve 4DVAR problem (see Algorithm 5.1). Moreover, we have shown that using EnKS to solve 4DVAR linearized subproblem is equivalent to use the Levenberg-Marquardt method based on probabilistic models. Numerically, we have illustrated the importance of using probability p_j to increase the performance of the new method. In the next chapter, we present more numerical experiments to investigate the impact of the other parameters, namely the ensemble size (parameter N), the finite differences parameter (τ) and the covariances scale parameter (t).

Chapter 6

Numerical experiments

In Chapters 4 and 5, we gave a variant of Levenberg-Marquardt algorithm to deal with the case where the linearized subproblem is solved inexactly and the gradient model is noisy. We already provide some numerical tests, especially to illustrate the importance of using the probability p_j to update the regularization parameter γ. In this chapter, we give more numerical experiments to investigate the impact of other parameters. We give tests with simple version of Algorithm 5.1, where we maintain the regularization parameter fix (we do not update the parameter γ over iterations). In this case Algorithm 5.1 becomes Algorithm 6.1.

Algorithm 6.1: Levenberg-Marquardt EnKS method with fixed regularization

Initialization

Choose the constants $\tau \in (0, 1]$, $t \in (0, 1]$, N and $\gamma \geq 0$. Select $x^0_{0:p}$. Choose all the parameters related to solving the 4DVAR problem (5.1) using EnKS as the linear solver.

For $j = 0, 1, 2, \ldots$

1. Compute the increment ensemble $[\delta x^l_{0:p|p}]_{l=1}^N$ using (5.16) and the approximation of the derivatives as explained in section 5.1.2. Let $s^j_{0:p} = \frac{1}{N}\sum_{l=1}^N \delta x^l_{0:p|p}$.
2. Set $x^{j+1}_{0:p} = x^j_{0:p} + s^j_{0:p}$.

We organize this chapter as follows: in Section 6.1, we give some results where the regularization is not necessary to guarantee the convergence. We investigate the impact on the progress of the iterations of the following parameters, (i) the ensemble size (parameter N), (ii) the finite differences parameter (τ), (iii) and the covariance scale parameter (t). Lorenz 63 equations system is used as a forecast model. Section 6.2 is devoted to experiments where the regularization is necessary to guarantee the convergence (Gauss-Newton method without control of the step is not sufficient to ensure the convergence). We analyze the impact of the regularization parameter

(γ) on the progress of the iterations. The tests in these section are performed using the quasi geostrophic model as forecast model.

6.1 Numerical experiments using Lorenz 63 equations

In this section, experiments are performed by using the classical Lorenz 63 system [84] as the forecast model. We show an example without model error (strong-constraint 4DVAR problem), where convergence is achieved with $\gamma = 0$. There is no need for regularization to converge for this example (Gauss-Newton approach with EnKS as linear solver).

In this section, we first explain the experiments set up. Then we analyze the impact of the parameters: N, τ and t on the progress of the iterations.

6.1.1 Experiments set up

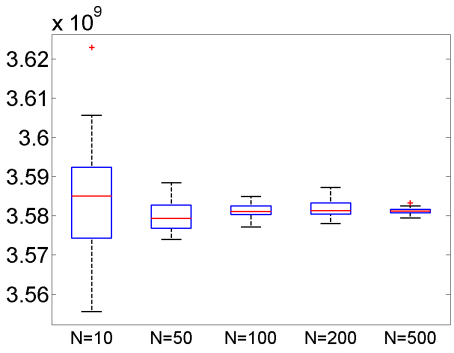
In the problem (2.38), we consider Lorenz 63 equations as forecast model, (the description of this model is already given in section 5.2.3). The twin experiment technique is used to evaluate the performance of the Algorithm 6.1. The initial truth is set to $\text{truth}_0 = [1, 1, 1]^\top$ and the truth at time k to $\text{truth}_k = \mathcal{M}(\text{truth}_{k-1})$, where $\mathcal{M}$ is the model obtained by discretization of Lorenz 63 system. The background mean x_b is sampled from $N(\text{truth}_0, B)$. The background covariance is $B = I_3$. The time step is chosen as $dt = 0.05$. The number of time steps is $p = 40$. The observation operator is $\mathcal{H}_k(x, y, z) = \left(x^3, y^3, z^3\right)$. At each time k, the observations are built as follows: $y_k = \mathcal{H}_k(\text{truth}_k) + v_k$, where v_k is sampled from $N(0, R)$ with $R = I_3$.

6.1.2 The ensemble size impact on the iteration progress

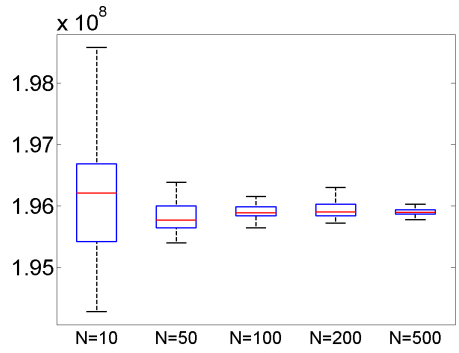
In this section, we investigate the influence of the ensemble size used to approximate the solution of the linearized subproblem, on the iteration progress. We fix the finite differences parameter τ to 10^{-6}, and the covariance scale parameter t to 1. Since the results depend on the ensemble generated at each iteration, the results we report here are averaged over 30 experiments. For each of these experiments, at each iteration, Algorithm 6.1 produces an ensemble to approximate the solution of the linearized subproblem.

Figures 6.1 and 6.2 show the box plots [1] of objective function values for eight iterations of Algorithm 6.1. The plots of the first figure correspond to 4 first iterations respectively. In the second figure, the plots correspond to the iterations, 5, 6, 7, and 8 respectively. In each plot, the first column corresponds to the results when $N = 10$, the second is for $N = 50$, the third is for $N = 100$, the fourth is for $N = 200$, and the last column is for $N = 500$.

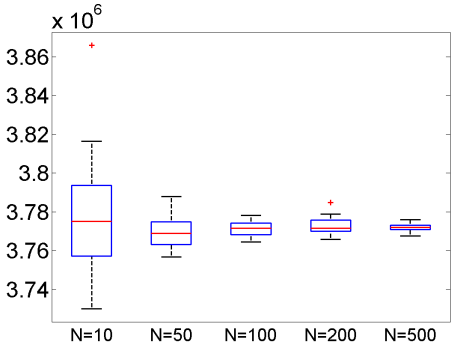
[1] It is a matlab function, where in each box the central curve presents the median (red curve), the edges are the 25th and 75th percentiles (blue curve), the whiskers extend to the most extreme data points the algorithm (matlab algorithm) considers to be not outliers (black curve), and the outliers are plotted individually (red dots).



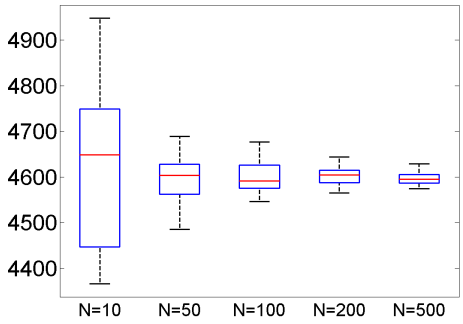
(a) Objective function value for the first iteration.



(b) Objective function value for the second iteration.



(c) Objective function value for the third iteration.

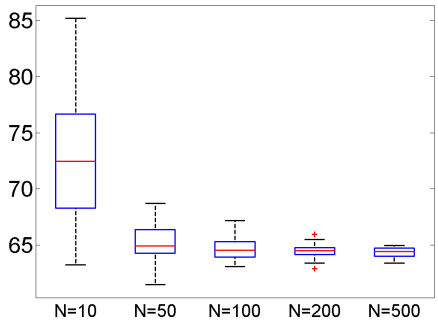


(d) Objective function value for the fourth iteration.

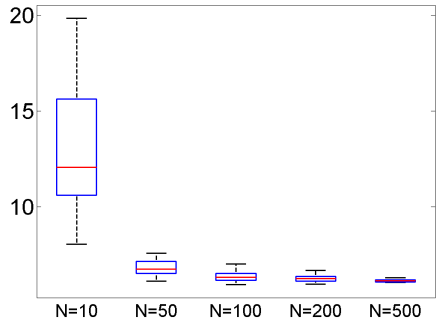
FIGURE 6.1: Box plots of objective function values for the first 4 iterations. In each plot, the first column corresponds to the results when $N = 10$, the second is for $N = 50$, the third is for $N = 100$, the fourth is for $N = 200$, and the last column is for $N = 500$.

From Figures 6.1 and 6.2 we see clearly, as expected, that increasing values of N provides better results (in terms of the decrease in the objective function). The standard deviation of the ensemble of runs (the length of the boxes) decreases when N increases. However, when we are interested only by the median of the different runs, we observe that for the first four iterations of the algorithm, the objective function value is almost the same for different values of N. But after the fifth iteration, the median of the objective function decreases when N increases. So we conclude that when the current iteration is "far" from the objective function minimum, on average, the ensemble size has not a significant influence. However, when the current iteration is "near" to the minimum, then the larger N is, the better results will be. In the first 4 iterations, we observe that for some runs, the smaller N is, the better reduction of the objective function will be. Hence from the previous analysis we conclude that an adaptive ensemble size over iteration can be a better choice than fixed N for all iterations: to choose small N for the first iteration and to increase it over iterations.

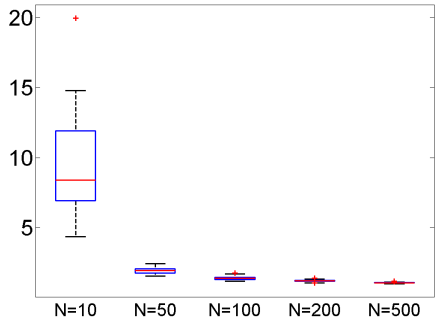
Figures 6.3 and 6.4 show the box plots of relative gradients for the first eight iterations of Algorithm 6.1. The plots of the first figure correspond to the four first iterations respectively. In the second figure, the plots correspond to the iterations, 5, 6, 7, and 8 respectively. In each plot,



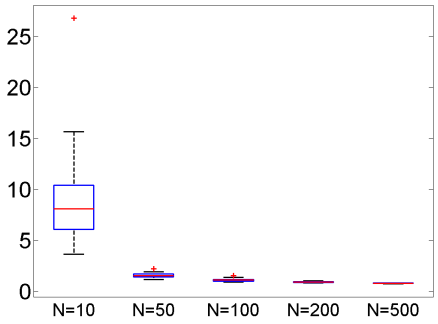
(a) Objective function value for the fifth iteration.



(b) Objective function value for the sixth iteration.



(c) Objective function value for the seventh iteration.



(d) Objective function value for the eighth iteration.

FIGURE 6.2: Box plots of objective function values for the 5th, 6th, 7th, and 8th iterations.

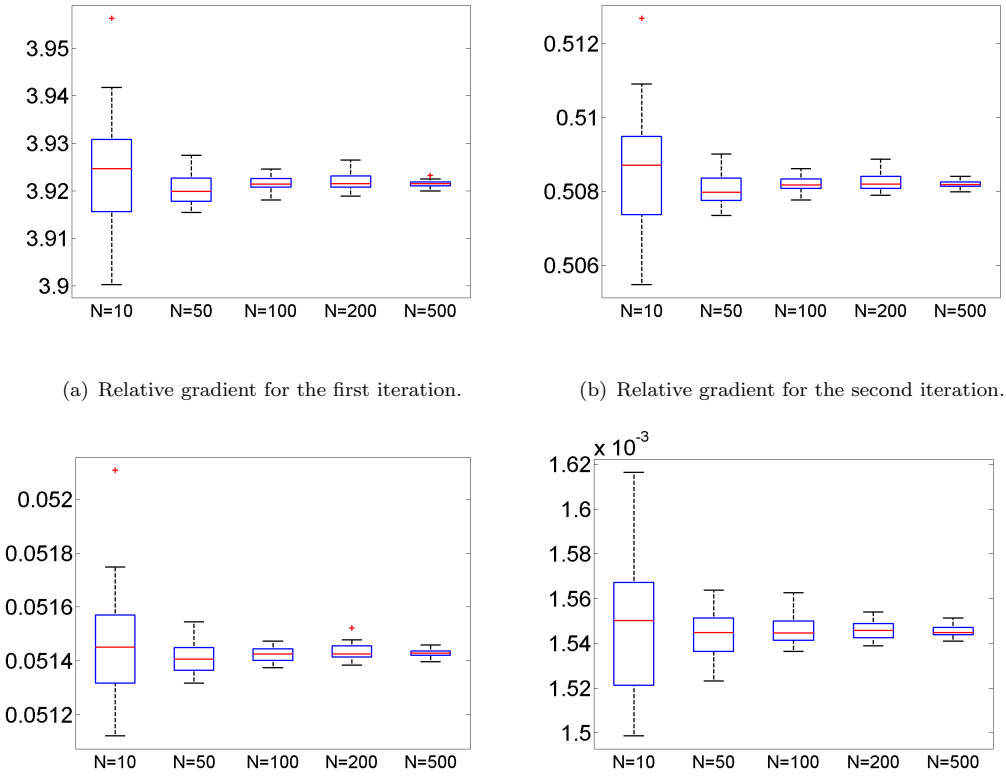
the first column corresponds to the results when $N = 10$, the second is for $N = 50$, the third is for $N = 100$, the fourth is for $N = 200$, and the last column is for $N = 500$. These figures confirm our previous analysis about the impact of the parameter N.

The mean and the standard deviation over different runs of the objective function and relative gradient are summarized in tables in Appendix C.

6.1.3 The impact of finite differences parameter along the iterations

In this section, we investigate the influence of the finite differences parameter used to approximate the derivatives of the model and observation operators. We fix the covariance scale parameter t to 1, and ensemble size N to 50. The results, we report here, are averaged over 30 experiments.

Tables 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, and 6.7 represent summary of results using Algorithm 6.1 with the following choices for the parameter τ: 1, 0.1, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6} respectively. These Tables show the mean and the standard deviation of the objective function and relative gradient for eight iterations.



(a) Relative gradient for the first iteration.

(b) Relative gradient for the second iteration.

(c) Relative gradient for the third iteration.

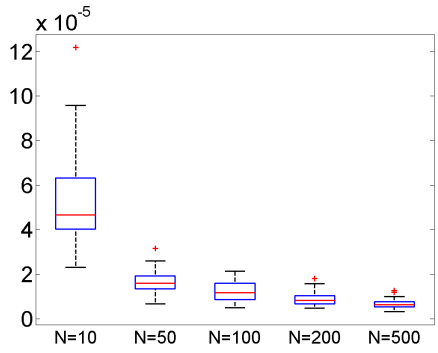
(d) Relative gradient for the fourth iteration.

FIGURE 6.3: Box plots of relative gradient for the first 4 iterations.

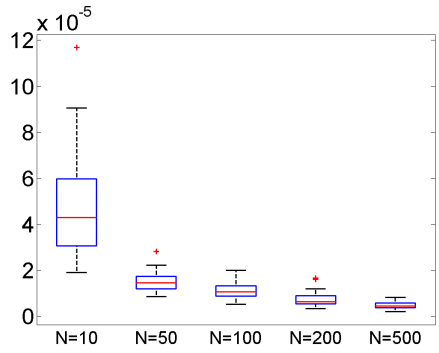
Figures 6.5 and 6.6 show the box plots of objective function for the first eight iterations of Algorithm 6.1. The plots of the first figure correspond to the four first iterations respectively. In the second figure, the plots correspond to the iterations, 5, 6, 7, and 8 respectively. In each plot, the first column corresponds to the results when $\tau = 10^{-2}$, the second is for $\tau = 10^{-3}$, the third is for $\tau = 10^{-4}$, the fourth is for $\tau = 10^{-5}$, and the last column is for $\tau = 10^{-6}$.

These tables show the impact of the parameter τ on the progress of iterations. For $\tau = 1$ (when we use the classical non linear EnKS), the results are almost the same after the first iteration, in this case the iterations do not improve the results. However, when $\tau \leq 0.1$ the objective function is decreasing over iterations.

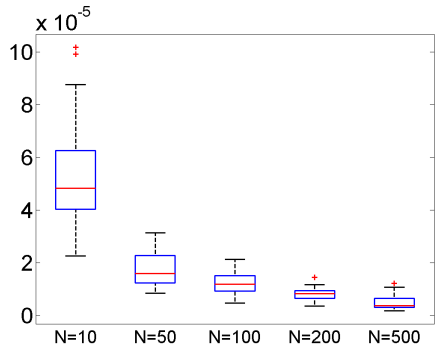
For small values of τ, for example, in Tables 6.3 and 6.4, we see that when $\tau \leq 10^{-2}$ few iterations were enough to reduce significantly the objective function. But for $\tau = 0.1$, the algorithm needs more iterations to reduce the objective function significantly. When $\tau = 10^{-4}$, the results are slightly different than the results with $\tau = 10^{-5}$ or 10^{-6}. So for these experiments, we conclude that it is better to choose $\tau \leq 10^{-4}$, such that the results will be less sensitive to the value of τ. This value is problem dependent, so for other experiences maybe a smaller τ will be needed. In practice, to avoid divergence due to the finite differences, it is better to choose τ as small as



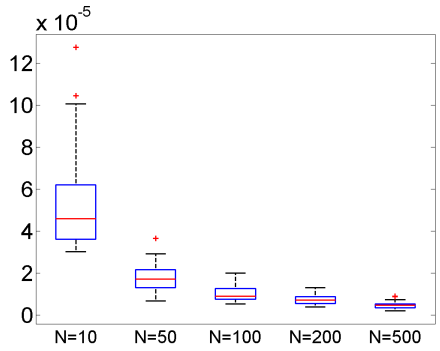
(a) Relative gradient for the fifth iteration.



(b) Relative gradient for the sixth iteration.



(c) Relative gradient for the seventh iteration.



(d) Relative gradient for the eighth iteration.

FIGURE 6.4: Box plots of relative gradient for the 5th , 6th,7th and 8th iterations.

possible, and since the computers use finite-precision arithmetic, we need to be careful to the effects of computer rounding.

We can see also from these tables, that for the first iteration, the best decrease in objective function is obtained when $\tau = 1$, and the worst decrease is obtained for $\tau = 10^{-6}$ (the bigger τ is, the better decrease in objective function will be). And from Figures 6.5 and 6.6 we see that for the first four iterations the bigger τ is, the better results will be, but for the iterations 5, 6, 7, and 8 we see that, the smaller τ is, the better results will be. Hence, an adaptive τ over iterations can be a good choice than fixed τ for all iterations: To choose big τ ($\tau = 1$) for the first iteration and to decrease it over iterations. Exploration of the best strategy to choose τ over iterations will be studied in the future works.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$1.02003e+6$	760713	0.0254455	0.00793264
2	$1.31874e+6$	904111	0.028162	0.00855416
3	$1.32354e+6$	769817	0.0284948	0.00676967
4	$1.38256e+6$	$1.46461e+6$	0.0279326	0.01112
5	$1.54959e+6$	$1.17558e+6$	0.0292484	0.0100845
6	$1.34157e+6$	988026	0.0275389	0.00930916
7	$2.05108e+6$	$2.02847e+6$	0.032617	0.0130256
8	$1.47114e+6$	$1.31421e+6$	0.0285715	0.0109438

TABLE 6.1: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $\tau = 1$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$1.39475e+9$	$1.02545e+7$	1.94175	0.0104526
2	$5.26613e+7$	551084	0.223874	0.00140712
3	414255	15901.7	0.0153886	0.000395556
4	5699.8	410.231	0.00117148	0.000542535
5	1299.63	315.227	0.00127304	0.000425505
6	830.148	130.175	0.00118449	0.000252579
7	826.846	133.989	0.00128004	0.000224837
8	847.404	162.952	0.00126899	0.000294887

TABLE 6.2: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $\tau = 0.1$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.21852e+9$	$3.84072e+6$	3.61684	0.00327528
2	$1.70111e+8$	250978	0.464039	0.000439236
3	$2.98839e+6$	7613.99	0.0454189	$6.15652e-5$
4	3266.88	44.8316	0.00120926	$1.28007e-5$
5	89.2153	2.95203	0.000119746	$3.21321e-5$
6	17.0808	2.27432	0.000122451	$3.17617e-5$
7	10.7502	2.00966	0.000123399	$2.70921e-5$
8	10.8172	1.88677	0.000122659	$2.7123e-5$

TABLE 6.3: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $\tau = 10^{-2}$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.54264e+9$	$3.99354e+6$	3.88933	0.00332865
2	$1.93129e+8$	265209	0.503535	0.000447233
3	$3.68603e+6$	7814.41	0.0507985	$5.75722e-5$
4	4431.52	41.6994	0.00150852	$8.92524e-6$
5	65.6978	1.45526	$2.26206e-5$	$8.20163e-6$
6	6.93278	0.428038	$1.92633e-5$	$6.6285e-6$
7	1.88476	0.254633	$1.73697e-5$	$6.35718e-6$
8	1.68046	0.213557	$2.17494e-5$	$9.8257e-6$

TABLE 6.4: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $\tau = 10^{-3}$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.57725e+9$	$3.66682e+6$	3.91813	0.00303631
2	$1.95616e+8$	249033	0.507715	0.000416979
3	$3.76302e+6$	6988.06	0.0513628	$5.07704e-5$
4	4581.31	45.7803	0.00154127	$9.35865e-6$
5	65.4442	1.45785	$1.987e-5$	$9.61086e-6$
6	6.844	0.482017	$1.54439e-5$	$6.27586e-6$
7	1.89082	0.249112	$1.73318e-5$	$5.58338e-6$
8	1.63813	0.306168	$1.54461e-5$	$4.786e-6$

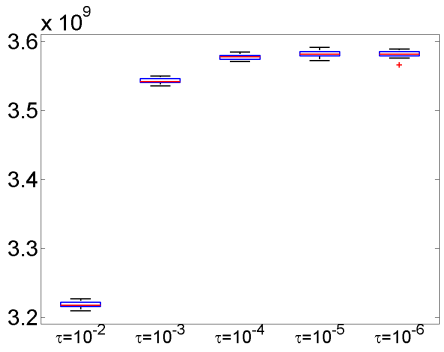
TABLE 6.5: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $\tau = 10^{-4}$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.58192e+9$	$4.39612e+6$	3.92203	0.00365915
2	$1.95938e+8$	297734	0.508257	0.000499714
3	$3.77314e+6$	8958.34	0.0514367	$6.52323e-5$
4	4594.81	38.0969	0.00154488	$7.73641e-6$
5	65.4126	1.55834	$1.97131e-5$	$7.6434e-6$
6	6.8555	0.421578	$1.61269e-5$	$6.16052e-6$
7	1.80078	0.250871	$1.42417e-5$	$5.08144e-6$
8	1.54713	0.227356	$1.56828e-5$	$5.60583e-6$

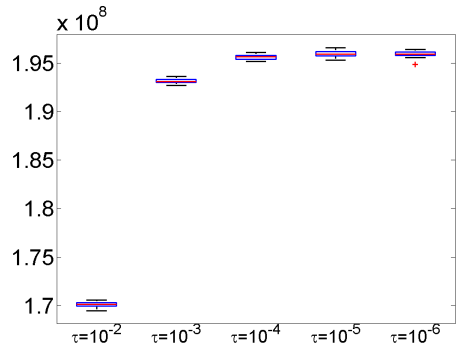
TABLE 6.6: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $\tau = 10^{-5}$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.58166e+9$	$4.51102e+6$	3.92179	0.00374781
2	$1.95928e+8$	301738	0.508239	0.000506715
3	$3.7728e+6$	8519.76	0.0514343	$6.22944e-5$
4	4598.51	47.7101	0.00154509	$9.46065e-6$
5	65.2617	1.45906	$1.79196e-5$	$7.04679e-6$
6	6.92311	0.509595	$1.83612e-5$	$9.08593e-6$
7	1.72074	0.222652	$1.63607e-5$	$6.36998e-6$
8	1.64117	0.213949	$1.78754e-5$	$7.12538e-6$

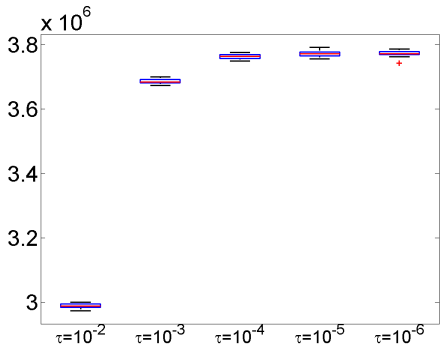
TABLE 6.7: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $\tau = 10^{-6}$. This results are based on 30 runs of the algorithm.



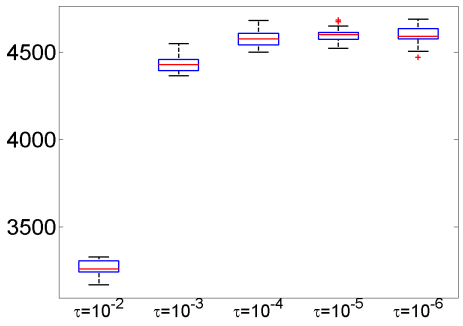
(a) Objective function value for the first iteration.



(b) Objective function value for the second iteration.

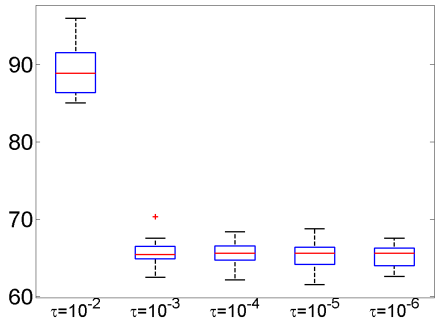


(c) Objective function value for the third iteration.

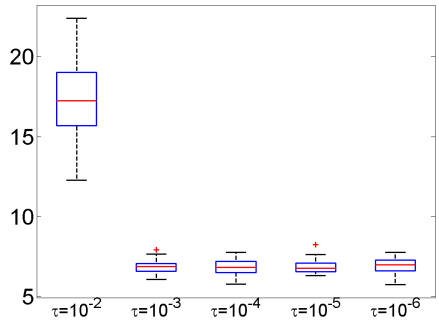


(d) Objective function value for the fourth iteration.

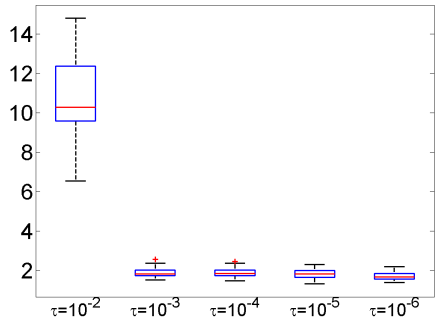
FIGURE 6.5: Box plots of objective function values for the first 4 iterations. In each plot, the first column corresponds to the results when $\tau = 10^{-2}$, the second is for $\tau = 10^{-3}$, the third is for $\tau = 10^{-4}$, the fourth is for $\tau = 10^{-5}$, and the last column is for $\tau = 10^{-6}$.



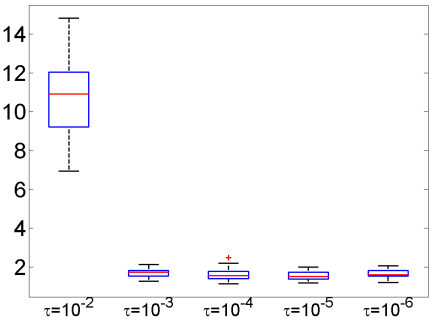
(a) Objective function value for the fifth iteration.



(b) Objective function value for the sixth iteration.



(c) Objective function value for the seventh iteration.



(d) Objective function value for the eighth iteration.

FIGURE 6.6: Box plots of objective function values for the 5th, 6th, 7th, and 8th iterations. In each plot, the first column corresponds to the results when $\tau = 10^{-2}$, the second is for $\tau = 10^{-3}$, the third is for $\tau = 10^{-4}$, the fourth is for $\tau = 10^{-5}$, and the last column is for $\tau = 10^{-6}$

6.1.4 The impact of the covariance scale parameter along iterations

In this section, we investigate the influence of the covariance scale parameter (parameter t) used to scale the covariances on the iteration progress. As we explained in Section 5.1.1, the covariance determines the spread of the ensemble, and in purpose to avoid linearization by tangent operators, we had to work with ensembles with a very small spread. In the following experiments we do not use linearization of the model and observation operators, hence we set the finite differences parameter τ to 1. The ensemble size is chosen to be $N = 50$. The results are averaged over 30 experiments.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$1.28257e+6$	$1.23853e+6$	0.0270522	0.0109531
2	$1.10873e+6$	916209	0.0251437	0.00896673
3	$1.51647e+6$	$1.49452e+6$	0.0290045	0.0119987
4	$1.39313e+6$	$1.27483e+6$	0.0279534	0.010177
5	$1.51007e+6$	$1.21716e+6$	0.0292823	0.0108102
6	$1.39358e+6$	880561	0.0288245	0.00842026
7	$1.95103e+6$	$4.04489e+6$	0.0301669	0.0221282
8	$1.20969e+6$	779758	0.0274561	0.00760138

TABLE 6.8: The mean and the standard deviation of the objective function values and relative gradient over iterations, for t = 1. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	5733.16	19422.4	0.00161204	0.00133425
2	1451.47	1138.93	0.0012582	0.000400696
3	1208.99	396.589	0.00124296	0.000366118
4	1951.53	2175.58	0.0013618	0.000526474
5	2044.8	4057.06	0.00144038	0.000594717
6	1476.91	1402.97	0.00126713	0.000420628
7	1869.56	1229.01	0.00156065	0.000479531
8	1230.38	489.404	0.00133016	0.000369764

TABLE 6.9: The mean and the standard deviation of the objective function values and relative gradient over iterations, for t = 0.1. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	96236.1	54838.6	0.00356334	0.00141698
2	18.7648	13.5008	0.000167082	$7.54836e-5$
3	15.3066	6.5091	0.00016295	$7.96217e-5$
4	15.9289	8.13209	0.000170817	$8.1496e-5$
5	14.3342	5.64091	0.000138005	$6.90142e-5$
6	16.3813	10.5751	0.000176655	0.000111055
7	13.6446	5.73428	0.000138049	$7.15604e-5$
8	14.2618	5.64293	0.00014546	$7.48936e-5$

TABLE 6.10: The mean and the standard deviation of the objective function values and relative gradient over iterations, for t = 10^{-2}. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	151509	30308.4	0.00499104	0.000712935
2	323.352	1157.41	$9.7606e-5$	$9.18185e-5$
3	5.70462	0.99866	$6.62323e-5$	$1.01845e-5$
4	5.45984	0.833337	$6.82585e-5$	$1.27266e-5$
5	5.37228	1.0759	$6.99485e-5$	$2.20055e-5$
6	5.47419	0.980439	$6.59529e-5$	$1.17609e-5$
7	5.40448	0.906405	$6.64815e-5$	$1.36321e-5$
8	5.35449	1.00148	$6.56933e-5$	$1.48703e-5$

TABLE 6.11: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $t = 10^{-3}$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	156532	6830.53	0.00505647	0.000158234
2	950.31	1941.53	0.000149884	0.000145801
3	13.5984	31.4542	$6.31974e-5$	$5.56812e-6$
4	5.64767	0.521936	$6.34469e-5$	$3.45804e-6$
5	5.40699	0.547723	$6.22896e-5$	$4.15821e-6$
6	5.43251	0.477186	$6.31823e-5$	$2.65772e-6$
7	5.37252	0.39704	$6.26536e-5$	$2.62201e-6$
8	5.40146	0.439204	$6.22815e-5$	$2.39527e-6$

TABLE 6.12: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $t = 10^{-4}$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	158242	2263.76	0.00508601	$5.21562e-5$
2	813.719	1756.22	0.000138571	0.000135563
3	17.6345	50.6745	$6.4255e-5$	$7.59141e-6$
4	5.39086	0.319794	$6.24925e-5$	$1.58549e-6$
5	5.3323	0.275657	$6.24319e-5$	$1.60708e-6$
6	5.32189	0.337935	$6.20656e-5$	$2.27654e-6$
7	5.23654	0.330453	$6.17784e-5$	$2.00154e-6$
8	5.3509	0.349742	$6.24633e-5$	$1.93566e-6$

TABLE 6.13: The mean and the standard deviation of the objective function values and relative gradient over iterations, for $t = 10^{-5}$. This results are based on 30 runs of the algorithm.

Iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	157885	1981.7	0.00507934	$4.59814e-5$
2	1206.74	1975.32	0.000172241	0.00015554
3	18.5641	48.9945	$6.40279e-5$	$6.60551e-6$
4	5.58651	0.483219	$6.27524e-5$	$1.95824e-6$
5	5.40551	0.258106	$6.2646e-5$	$1.51741e-6$
6	5.32878	0.280484	$6.2203e-5$	$1.93783e-6$
7	5.39498	0.259749	$6.27501e-5$	$1.68833e-6$
8	5.41447	0.308099	$6.30845e-5$	$1.70115e-6$

TABLE 6.14: The mean and the standard deviation of the objective function values and relative gradient over iterations, for t = 10^{-6}. This results are based on 30 runs of the algorithm.

Tables 6.8, 6.9, 6.10, 6.11, 6.12, 6.13 and 6.14 represent summary of the results when using Algorithm 6.1 with the following choices for the convariance scale parameter t: 1, 0.1, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, and 10^{-6} respectively. These tables show the objective function and the relative gradient mean and standard deviation for eight iterations.

These tables show the impact of the parameter t on the iteration progress. For $t = 1$, the results are almost the same after the first iteration (the same conclusion as for the case when $\tau = 1$ and $t = 1$ in the previous section). In this case the iterations do not improve the results. However when $t \leq 0.1$ the objective function is decreasing over iterations. The smaller t is, the better results will be. For small values of t, for example, in Tables 6.10 and 6.11 we see that when $t \leq 10^{-2}$ few iterations were enough to reduce significantly the objective function. But for $t = 0.1$, the algorithm needs more iterations to reduce the objective function significantly. These results illustrate the importance of scaling the covariances with small t.

6.2 Numerical tests using Quasi Geostrophic model (QG)

In this section, we show an example with model error, and where the regularization is necessary to guarantee the convergence. We will analyze the impact of the regularization parameter (parameter γ) used in Algorithm 6.1 approach.

We start by introducing the qg model [44], which will be used as dynamical model. Then we describe the experiments set up. Finally, we present the results when using Algorithm 6.1. We present the results for the following different choices of regularization parameter $\gamma = 0$ (no regularization used), 0.001, 0.1, 1, 10, 100, 500, 1000.

6.2.1 Model description

The model description follows the ECMWF technical report [47].

The two-layer qg model represents quasi-geostrophic flow in a cyclic channel. The equations of the two-level model are given by [44] (see also [97]), and are expressed in terms of non-dimensionalized variables:

$$\frac{Dq_1}{Dt} = \frac{Dq_2}{Dt} = 0 \tag{6.1}$$

where $\frac{D}{Dt}$ denotes the total derivative, and q_1 and q_2 denote the quasi-geostrophic potential vorticity [35] on the upper and lower layers respectively. For each quantity the subscript 1 will refer to the upper layer and 2 to the lower layer. The equations in (6.1) correspond to conservation of potential vorticity. The quantities q_1 and q_2 satisfy also the following equations:

$$q_1 = \nabla^2\psi_1 - F_1(\psi_1 - \psi_2) + \beta y, \tag{6.2}$$

$$q_2 = \nabla^2\psi_2 - F_2(\psi_2 - \psi_1) + \beta y + R_s, \tag{6.3}$$

where ψ denotes stream function, ∇^2 is the two dimensional Laplacian, β is the northward derivative of the Coriolis parameter, and R_s is the heating. The two parameters F_1 and F_2 are used to couple the two layers:

$$F_1 = \frac{f_0^2 L^2}{D_1 g \Delta\theta/\bar{\theta}} \text{ and } F_2 = \frac{f_0^2 L^2}{D_2 g \Delta\theta/\bar{\theta}}.$$

L is a typical length scale. D_1 and D_2 are the depths of the upper and lower layers respectively. f_0 is the Coriolis parameter at the southern boundary and β_0 is its northward derivative. g is the acceleration due to gravity , $\Delta\theta$ is the difference in potential temperature across the layer interface, and $\bar{\theta}$ is the mean potential temperature. We define $\bar{U}$ a typical velocity. We denote by $\tilde{t}$, $\tilde{x}$, $\tilde{y}$, $\tilde{u}$, and $\tilde{v}$ the dimensional quantities corresponding to time, spatial coordinates and velocities respectively. The non-dimensional corresponding quantities are defined by:

$$t = \tilde{t}\frac{\bar{U}}{L},\ x = \frac{\tilde{x}}{L},\ y = \frac{\tilde{y}}{L},\ u = \frac{\tilde{u}}{\bar{U}},\ v = \frac{\tilde{v}}{\bar{U}},\ \beta = \beta_0\frac{L^2}{\bar{U}}.$$

For experiments described in this dissertation, we have set:

$$L = 10^6 m,\ \bar{U} = 10 m s^{-1},\ f_0 = 10^{-4} s^{-1},\ \beta_0 = 1.5 \times 10^{-11} s^{-1} m^{-1},$$

$$g = 10 m s^{-2},\ D_1 = 6000 m,\ D_2 = 4000 m,\ \frac{\Delta\theta}{\bar{\theta}} = 0.1.$$

These parameters are used also to define the true evolution of the system (truth).

The model variables (stream function, potential vorticity and wind components) are defined on a rectangular grid of dimension $n_x \times n_y$. In this experiments we choose $n_x = 40$ and $n_y = 20$, with a dimensional grid spacing of $300km$ in both the north-south and east-west directions. The model state is only the values of stream function over the grid. The potential vorticity and wind components are diagnostic quantities and they can be calculated from stream function. They do not form part of the control variable. The dimension of the state vector of the model (stream function) is thus 1600 ($2 \times n_x \times n_y$).

The time-stepping consists of a semi-Lagrangian advection of potential vorticity, followed by an

inversion of the potential vorticity equation to determine stream function and velocity components. The interpolation to the departure point is bi-cubic. A 1-hour time step was used for all the experiments presented here.

The equations are solved on a domain which is cyclic in the zonal direction, hence the potential vorticity equations can be decoupled. The meridional wind is equal to zero on the northern and southern boundaries and the stream function in this boundaries is chosen by the user. The choice of the stream function in the boundaries is equivalent to choose the mean zonal wind on each layer . In this experiments, the mean wind was $40ms^{-1}$ in the upper layer and $10ms^{-1}$ in the lower layer.

Potential vorticity is discretized using a standard five-point finite-difference representation of the Laplacian. It is inverted by applying ∇^2 to equation (6.2) and subtracting F_1 times equation (6.3) and F_2 times equation (6.2) to give:

$$\nabla^2 q_1 - F_2 q_1 - F_1 q_2 = \nabla^2(\nabla^2\psi_1) - (F_1 + F_2)\nabla^2\psi_1. \tag{6.4}$$

This latter equation is a two-dimensional Helmholtz equation, which can be solved for $\nabla^2\psi_1$. The Laplacian can then be inverted to determine ψ_1 . After determining ψ_1 and $\nabla^2\psi_1$, the stream function on level 2 can be determined by substitution into equation (6.2). Solution of the Helmholtz equation and inversion of the Laplacian are achieved using an FFT-based method. Applying a Fourier transform in the east-west direction to equation (6.4) gives a set of independent equations for each wave number.

6.2.2 Experiments set up

The initial states for the two sets of integration were constructed by taking a sequence of states from an unperturbed truth run (the truth), and adding perturbations drawn from a multivariate Gaussian distribution with zero mean and covariance matrix constructed from a large sample of errors in three-hour forecasts made by a version of the model with layer depths fixed to $D_1 = 5500m$ and $D2 = 4500m$ for the upper and lower layer respectively.

The truth was generated from a model with layer depths of $D_1 = 6000m$ and $D_2 = 4000m$, and the time step was set to $300s$ whereas the assimilating model had layer depths of $D_1 = 5500m$ and $D2 = 4500m$, and time step was set to $3600s$. these changes on the layer depths and time step provides a source of model error.

For all the experiments presented here, observations of non-dimensional stream function, vector wind and wind speed were taken from a truth of the model at 100 points randomly distributed over both levels. Observations were made every 12 hours. We note that the number of observations used in an analysis cycle is much smaller than the number of degrees of freedom of the model. Observation errors were assumed to be independent from each others and uncorrelated in time. The standard deviations were chosen to be equal to 0.4 for stream function observation error, 0.6 for wind and 1.2 for wind speed. The observations operator is the bi-linear interpolation of the model fields to horizontal observation locations.

The background error covariance matrix (B matrix) and the model error covariances (matrices Q_k) used in these experiments correspond to isotropic, homogeneous correlations of stream function in the horizontal, with Gaussian spacial structure, and with constant vertical correlation over the grid. These matrices are characterized by theirs standard deviations, theirs vertical correlations and theirs horizontal length scale. For the matrix B the standard deviation in this experiments is 0.8. The vertical correlation is equal to 0.2 and the horizontal length scale is equal to $10^6 m$. For the matrices Q_k the standard deviation in this experiments is 0.2. The vertical correlation is equal to 0.5 and the horizontal length scale is equal to $2 \times 10^6 m$.

We used an analysis windows of 10 days, with two sub-windows of 5 days ($p = 2$). For testing codes we used the ECMWF framework named Object-Oriented Prediction System (OOPS) [113].

6.2.3 Numerical results

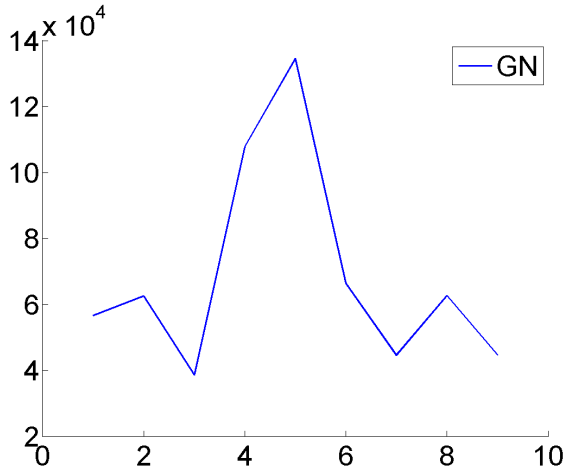
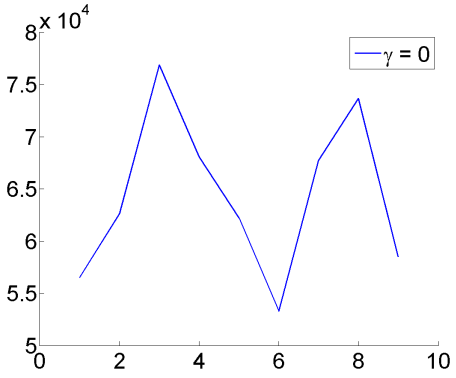


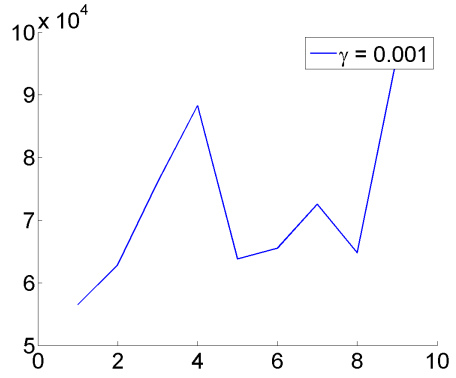
FIGURE 6.7: Objective function values for eight iterations when using Gauss-Newton algorithm (Algorithm 2.7), the subproblem is solved exactly at each iteration.

The proposed method (Algorithm 6.1) uses the sample covariance from the ensemble which can be suboptimal as a result of small ensemble size. The most common algorithms for dealing with these deficiency are inflation [3] and covariance localization [66]. Here, we do not want to use these techniques that would mask some of the properties of the proposed method, hence the ensemble size is chosen to be large $N = 3000$. Nevertheless, one can contemplate building local versions of the method similarly to what was done by [66] (Local Transform Kalman Filter (LETKF)).

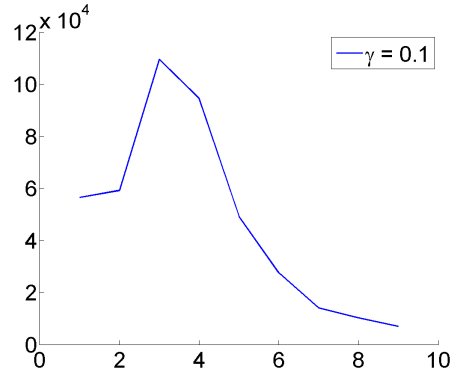
Figure 6.7 shows the objective function over iterations, when using Gauss-Newton method (these objective function values are summarized in Table C.6 in Appendix C). We see, clearly, that the Gauss-Newton algorithm does not converge. The objective function is not monotonically decreasing over iterations. The objective function is oscillating by increasing and decreasing along iterations. Therefore for this example Gauss-Newton approach without regularization is diverging.



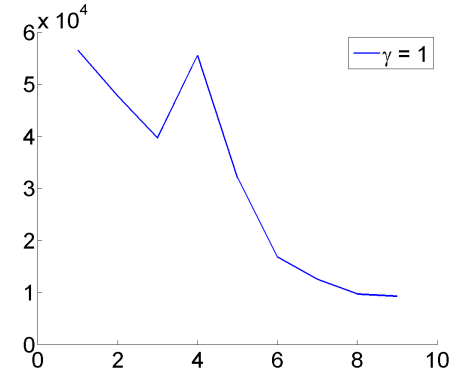
(a) Objective function values over iterations for $\gamma = 0$.



(b) Objective function values over iterations for $\gamma = 0.001$.



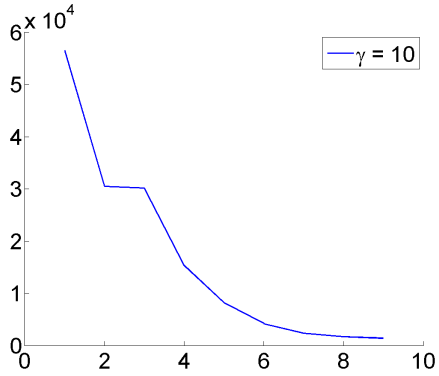
(c) Objective function values over iterations for $\gamma = 0.1$.



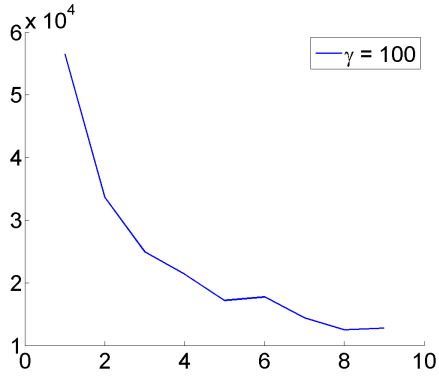
(d) Objective function values over iterations for $\gamma = 1$.

FIGURE 6.8: Objective function values for eight iterations for the following choices of γ: 0, 0.001, 0.1, 1.

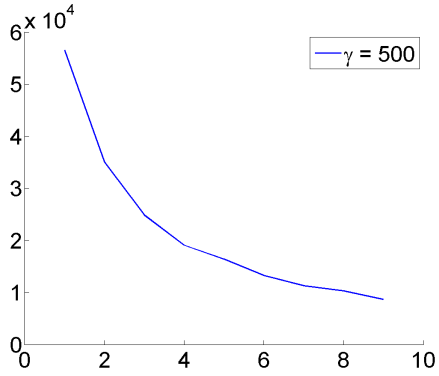
Figures 6.8(a), 6.8(b), 6.8(c), 6.8(d), 6.9(a), 6.9(b), 6.9(c), and 6.9(d) show the objective function values over 8 iterations for the following choices of regularization parameter: $\gamma = 0$, 0.001, 0.1, 1, 10, 100, 500, 1000, respectively. From these figures, we see that: for $\gamma = 0$, as expected, Algorithm 6.1 is diverging (since we do not use regularization and we only approximate the linearized subproblem using an ensemble). For small values of γ (in this experiments for $\gamma = 0.001$, 0.1 or 1) the objective function is not monotonically decreasing. In the case where $\gamma = 0.001$, the algorithm is still diverging even if the regularization is used. Hence small values of regularization are not enough to control well the step size, in the sense that the objective function does not decrease monotonically over iterations. However, when $\gamma \geq 10$ the objective function is decreasing over iterations, for example when $\gamma = 10$ the objective function decreases monotonically from 56508.9 (at the first iteration) to 1367.02 after eight iterations. Moreover, the decrease in the objective function depends on γ, the best decrease in this experiment is obtained for $\gamma = 10$. For big values of γ ($\gamma \geq 100$) the objective function is decreasing, as expected, but the decrease in the objective function is less than one attained using $\gamma = 10$. We conclude that when the regularization is used to ensure convergence: (i) for small values of regularization, the method can still diverging, (ii) and for big values of regularization the objective function decreases but slowly (and may be a lot of iterations will be needed to attain some predefined



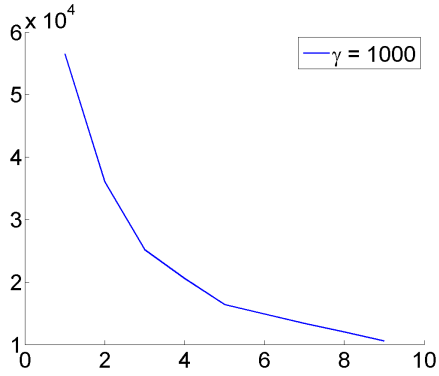
(a) Objective function values over iterations for $\gamma = 10$.



(b) Objective function values over iterations for $\gamma = 100$.



(c) Objective function values over iterations for $\gamma = 500$.



(d) Objective function values over iterations for $\gamma = 1000$.

FIGURE 6.9: Objective function values for eight iterations for the following choices of γ: 10, 100, 500, 1000.

decrease). Therefore the regularization parameter should not be neither "very small" nor "very big". An adaptive γ over iterations (As proposed in Algorithm 5.1) can be a good compromise, in the sense (i) to increase γ when the objective function increases (ii) and to decrease γ when the objective function decreases.

Tables C.7 and C.8 in appendix C summarize the objective function values over the iterations for the different choices of the regularization parameter.

In this chapter we have analyzed numerically the impact of several parameters arising in Algorithm 5.1. We have used two different forecast models in our experiments, namely Lorenz 63 model and the quasi-geostropic model. In the next chapter, we study the asymptotic behavior of Algorithm 5.1, as the finite differences parameter goes to zero and/or the ensemble size goes to infinity.

Chapter 7

Towards a convergence theory of ensemble based methods

In Chapter 5, we have proposed to use EnKS as linear solver for 4DVAR problem (Algorithm 5.1). At each iteration we approximated the linearized subproblem solution by the empirical mean of an EnKS, where each ensemble member is considered as vector of $\mathbf{R}^n$, meaning that each vector is regarded as a sample point of a random vector. In this chapter we investigate a different way to interpret such algorithm, similarly as in [78, 87], each ensemble member is considered as random vector and not merely as vector of $\mathbf{R}^n$. In fact the elements of the EnKS can be seen as random vectors instead of realizations. Then an important question related to EnKF/EnKS and related ensemble methods is a law of large number-like theorem as the size of the ensemble grows to infinity. In [78, 87], it was proved that the ensemble mean and covariance of EnKF converge to those of the KF, as the number of ensemble members grows to infinity. The analysis in [87] relies on the fact that ensemble members are exchangeable and uses the uniform integrability theorem, which does not provide a rate of convergence; in [78] a stochastic inequalities for the random matrices and vectors are obtained with the classical rate $\frac{1}{\sqrt{N}}$. In this chapter we follow the spirit of the paper [87], and propose to extend the convergence to EnKS as the number N increases to infinity. The randomness of the elements of EnKS turns also random the current point of Algorithm 5.1 (at each iteration the solution of linearized subproblem is "a random vector"). We will investigate also the asymptotic behavior of this algorithm.

We start by recalling some definitions and preliminary results that will be useful in the following of the chapter (see Section 7.1). Then we show the convergence in $L^{\mathbf{p}}, \forall\ \mathbf{p} \in [1, \infty)$ of EnKS in the limit for large ensemble to the KS, in the sense that the ensemble mean and covariance constructed by EnKS method converge to the mean and covariance of the KS respectively (see Section 7.3). Finally, we show the convergence of Algorithm 5.1 iterations to their corresponding iterations in Algorithm 2.8. Since Algorithm 5.1 uses finite differences for derivatives approximation, (i) we start by showing the convergence on probability of its iterations to the iterations generated by the algorithm with exact derivatives as the finite differences parameter goes to zero,

(ii) then we prove the convergence in $L^{\mathbf{p}}$ of Algorithm 5.1 iterations as the size of the ensemble grows to infinity (see Section 7.4).

7.1 Basic concepts and preliminaries

This section consists of fundamental information that will be a reference for the sequel of the chapter. First, we recall definition of sequence of random vectors exchangeability, the notion of convergence in probability and in $L^{\mathbf{p}}$ of random elements. Then we present several lemmas that will be useful for the following of the chapter.

Definition 7.1. (Exchangeability of random vectors) A set of N random vectors $[x^1, \ldots, x^N]$ is exchangeable [7] if their joint distribution is invariant to a permutation of the indices; that is, for any permutation π of the numbers $1, \ldots, N$ and any Borel set B

$$\mathbb{P}\left(\left[x^{\pi(1)}, \ldots, x^{\pi(N)}\right] \in B\right) = \mathbb{P}\left(\left[x^1, \ldots, x^N\right] \in B\right).$$

Clearly an i.i.d sequence is exchangeable.

Definition 7.2. (Convergence in probability) A sequence $\left(x^k\right)$ of random elements converges in probability towards the random element x if $\forall\ \epsilon > 0$:

$$\lim_{k\to\infty} \mathbb{P}\left(\left\|x^k - x\right\| \geq \epsilon\right) = 0.$$

$$i.e. \forall\ \epsilon > 0,\ \forall\ \tilde{\epsilon} > 0,\ \exists\ k_0 \text{ such that } \forall\ k \geq k_0\ \mathbb{P}\left(\left\{\omega : \left\|x^k(\omega) - x(\omega)\right\| \leq \epsilon\right\}\right) \geq 1 - \tilde{\epsilon},$$

where $x(\omega)$ means a realization of random element x.

The concept of convergence in probability is extended in an obvious manner to the case when the random elements are indexed by $\tau > 0$. Then $x^\tau \to x$ in probability as $\tau \to 0$ means:

$$\forall\ \epsilon > 0,\ \forall\ \tilde{\epsilon} > 0,\ \exists\ \tau_0 > 0 \text{ such that } \forall\ \tau \leq \tau_0\ \mathbb{P}\left(\left\{\omega : \left\|x^\tau(\omega) - x(\omega)\right\| \leq \epsilon\right\}\right) \geq 1 - \tilde{\epsilon}.$$

If x is a random element (either vector or matrix), and $\|x\|$ is the usual Euclidean norm for vectors and spectral norm for matrices. For $\mathbf{p} \in [1, \infty)$, denote

$$\|x\|_{\mathbf{p}} = E\left(\|x\|^{\mathbf{p}}\right)^{1/\mathbf{p}}.$$

The space $L^{\mathbf{p}}$ (of vectors or matrices) consists of all random elements x such that $\|x\|_{\mathbf{p}} < \infty$. $\|.\|_{\mathbf{p}}$ is a pseudo norm of the space $L^{\mathbf{p}}$. Note that if the element x is deterministic

$$\|x\|_{\mathbf{p}} = E\left(\|x\|^{\mathbf{p}}\right)^{1/\mathbf{p}} = \left(\|x\|^{\mathbf{p}}\right)^{1/\mathbf{p}} = \|x\|.$$

Definition 7.3. (Convergence in $L^{\mathbf{p}}$) Given a real number $\mathbf{p} \geq 1$. A sequence $\left(x^k\right)$ of random elements converges in $L^{\mathbf{p}}$ (or in the $\mathbf{p}$-th mean) towards the random element x if the $\mathbf{p}$-th absolute

moments $E\left(\left\|x^k\right\|^{\mathbf{p}}\right)$ and $E\left(\|x\|^{\mathbf{p}}\right)$ of x^k and x exist, and

$$\lim_{k\to\infty} E\left(\left\|x^k - x\right\|^{\mathbf{p}}\right) = 0.$$

We state the following lemmas which will be used in this chapter.

Lemma 7.4. *If random elements $y^1, \ldots, y^N$ are exchangeable, and $z^1, \ldots, z^N$ are also exchangeable, and independent from $y^1, \ldots, y^N$, then $y^1 + z^1, \ldots, y^N + z^N$ are exchangeable.*

Lemma 7.5. *Suppose $X = \left[x^1, \ldots, x^N\right]$ and $Y = \left[y^1, \ldots, y^N\right]$ are exchangeable, the random elements X and Y are independent, and $z^k = F\left(y^1, \ldots, y^N, y^k, x^k\right)$ where F is measurable and permutation invariant in the first N arguments, then $Z = \left[z^1, \ldots, z^N\right]$ has exchangeable columns.*

For the proof of the previous two lemmas we refer to [87, lemma 1].

Lemma 7.6. *(Uniform integrability) If (x^k) is a bounded sequence in $L^{\mathbf{p}}$ and $x^k \to x$ in probability, then: $x^k \to x$ in L^q $\forall$ $q \in [1, \mathbf{p})$.*

Proof. The lemma follows from uniform integrability [15, page 338]. □

Lemma 7.7. *(Continuous mapping theorem) Let x^k be a sequence of random elements with values on a metric space $\mathcal{A}$, such that $x^k \to x$ in probability. Let f be a continuous function from $\mathcal{A}$ to another metric space $\mathcal{B}$. Then $f\left(x^k\right) \to f\left(x\right)$ in probability.*

Proof. For the proof we refer to [117, Theorem 2.3]. □

7.2 On the convergence of ensemble Kalman filter

For theoretical purposes, we define an auxiliary ensemble $U_{p|p} = [u^l_{p|p}]^N_{l=1}$ in the same way as the ensemble $X_{p|p} = [x_{p|p}]^N_{l=1}$ which is constructed by Algorithm 2.2, but in the recurrence to build $U_{k|k}$ we use the exact covariance matrix of $U_{k|k-1}$ instead of its empirical estimate. In fact for $k = 0$, $U_{0|0} = X_{0|0}$, and for $k = 1, \ldots, p$, we build $U_{k|k}$ as follows:

$$u^l_{k|k-1} = M_k u^l_{k-1|k-1} + m_k + v^l_k, \tag{7.1}$$

$$u^l_{k|k} = u^l_{k|k-1} + P_{k|k-1}H_k^\top \left(R_k + H_k P_{k|k-1} H_k^\top\right)^{-1} \left(y_k - w^l_k - H_k u^l_{k|k-1}\right), \quad l = 1, \ldots, N, \tag{7.2}$$

where $P_{k|k-1}$ is the exact covariance of $u^1_{k|k-1}$,

$$P_{k|k-1} = E\left[\left(u^1_{k|k-1} - E\left(u^1_{k|k-1}\right)\right)\left(u^1_{k|k-1} - E\left(u^1_{k|k-1}\right)\right)^\top\right],$$

and the random vectors $[v^l_k]^N_{l=1}$ and $[w^l_k]^N_{l=1}$ are the same as those used to build the ensemble $X_{k|k}$.

Note that the only difference between the two ensembles $X_{k|k}$ and $U_{k|k}$ is that for the construction of the first ensemble we use the empirical prediction covariance $P^N_{k|k-1}$ of the ensemble $X_{k|k-1}$, which depends on all ensemble members. Therefore, the members of the ensemble $[X^l_{k|k}]^N_{l=1}$ are in general dependent. However,

Lemma 7.8. *The members of the ensemble $U_{k|k}$ are i.i.d and the distribution of each $u^l_{k|k}$ is the same as the Kalman filter distribution, for any $k = 0, \ldots, p$,*

Proof. The proof is by induction and the same as in [87, Lemma 4], except we take the additional perturbation v^l_k into account. Since $[v^l_k]^N_{l=1}$ are Gaussian and independent of everything else by assumption, $[u^l_{k|k}]^N_{l=1}$ are independent and Gaussian. The forecast covariance $P_{k|k-1}$ is non-random matrix, and consequently, the step (7.2) is a linear transformation, which preserves the independence of the ensemble members and the Gaussianity of the distribution. The members of the ensemble $U_{k|k}$ have the same mean and covariance as given by the Kalman filter [22, eq. (15) and (16)]. The proof is completed by noting that a Gaussian distribution is determined by its mean and covariance. □

The large sample asymptotic behavior of the i.i.d. random vectors $[u^l_{k|k}]^N_{l=1}$ is "simple" to analyze, because of independence, but their covariance matrix $P_{k|k-1}$ is unknown in general, and so are the random vectors $[u^l_{k|k-1}]^N_{l=1}$ and $[u^l_{k|k}]^N_{l=1}$ themselves. In contrast, the random vectors in the EnKF $[x^l_{k|k-1}]^N_{l=1}$ and $[x^l_{k|k}]^N_{l=1}$ are dependent, because they all contribute to the empirical covariance matrix $P^N_{k|k-1}$, but their empirical covariance matrix can be easily computed, and so are the elements in the EnKF. In the following theorem we recall a result of convergence obtained in [78, 87] between the members of those ensembles ($X_{k|k-1}/X_{k|k}$ and $U_{k|k-1}/U_{k|k}$).

Theorem 7.9. *Let the random matrix defined for $k = 0, \ldots, p$, by*

$$\left[X_{k|k}; U_{k|k}\right] = \begin{bmatrix} X_{k|k} \\ U_{k|k} \end{bmatrix} = \begin{bmatrix} x^1_{k|k}, \ldots, x^N_{k|k} \\ u^1_{k|k}, \ldots, u^N_{k|k} \end{bmatrix}. \tag{7.3}$$

for each time step $k = 0, \ldots, p$, (7.3) has exchangeable columns. Moreover

$$\begin{aligned}
x^1_{k|k-1} &\rightarrow u^1_{k|k-1},\ x^1_{k|k} \rightarrow u^1_{k|k} \\
\frac{1}{N}\sum_{l=1}^{N} x^l_{k|k-1} &\rightarrow E(u^1_{k|k-1}),\ \frac{1}{N}\sum_{l=1}^{N} x^l_{k|k} \rightarrow E(u^1_{k|k}) \\
P^N_{k|k-1} &= \frac{1}{N-1}\sum_{l=1}^{N}\left(x^l_{k|k-1} - \frac{1}{N}\sum_{l=1}^{N} x^l_{k|k}\right)\left(x^l_{k|k-1} - \frac{1}{N}\sum_{l=1}^{N} x^l_{k|k}\right)^{\top} \rightarrow \\
P_{k|k-1} &= E\left[\left(u^1_{k|k-1} - E\left(u^1_{k|k-1}\right)\right)\left(u^1_{k|k-1} - E\left(u^1_{k|k-1}\right)\right)^{\top}\right],
\end{aligned}$$

in $L^{\mathbf{p}}$ as $N \rightarrow \infty, \forall\ \mathbf{p} \in [1, \infty)$.

Proof. The theorem is a simple extension of that of [87, Theorem 1], by adding the model error v^l_k in each step of the induction over k.

Notice that since (7.3) has exchangeable columns, and $x^1_{k|k} \to u^1_{k|k}$ in $L^{\mathbf{p}}$ for given $\mathbf{p} \in [1, \infty)$, we have the same convergence result for each member of the ensemble i.e, $\forall\ l \in \mathbb{N}^*,\ x^l_{k|k-1} \to u^l_{k|k-1}$ and $x^l_{k|k} \to u^l_{k|k}$ in $L^{\mathbf{p}}$ (the proof of convergence for other members is exactly the same by changing the superscript 1 by the member superscript). □

7.3 On the convergence of ensemble Kalman smoother

In this section, we extend the result of Theorem 7.9 in the previous section to the EnKS. We denote by $X_{0:p|p} = [x^l_{0:p|p}]_{l=1}^N$ an EnKS generated by Algorithm 2.4. Just as for EnKF, we construct an ensemble $U_{0:p|p} = [u^l_{0:p|p}]_{l=1}^N$ by induction on k as follows:

For $k = 0$, $U_{0|0} = X_{0|0}$, and for $k = 1, \dots, p$,

$$\begin{aligned} u^l_{k|k-1} &= M_k u^l_{k-1|k-1} + m_k + v^l_k, \\ u^l_{0:k|k} &= u^l_{0:k|k-1} + P_{0:k,0:k|k-1}\tilde{H}_k^\top \left(R_k + \tilde{H}_k P_{0:k,0:k|k-1}\tilde{H}_k^\top\right)^{-1} \left(y_k - w^l_k - H_k u^l_{k|k-1}\right), \\ & \quad l = 1, \dots, N. \end{aligned}$$

where $P_{0:k,0:k|k-1}$ is the exact covariance of $u^1_{0:k|k-1}$. The blocks of this covariance are, for $l, q = 0, \dots, k$,

$$P_{\ell,q|k-1} = E[(U^1_{\ell|k-1} - E(U^1_{\ell|k-1}))(U^1_{q|k-1} - E(U^1_{q|k-1}))^\top].$$

$[v^l_k]_{l=1}^N$ and $[w^l_k]_{l=1}^N$ are the same as those used to build the ensemble $X_{0:k|k}$.

As in the case of the filter, the members of the ensemble $U_{0:k|k}$ are i.i.d and their common distribution is the Kalman smoother distribution.

Since the Kalman smoother is nothing else than the Kalman filter for the composite state $X_{0:k}$, the same induction step as in Theorem 7.9 applies for each $k \in \{0, \dots, p\}$, and we have the following:

Theorem 7.10. *Let the random matrix defined for $k = 0, \dots, p$, by*

$$[X_{0:k|k}; U_{0:k|k}] = \begin{bmatrix} X_{0:k|k} \\ U_{0:k|k} \end{bmatrix} = \begin{bmatrix} x^1_{0:k|k}, \dots, x^N_{0:k|k} \\ u^1_{0:k|k}, \dots, u^N_{0:k|k} \end{bmatrix}. \tag{7.4}$$

for each time step $k = 0, \dots, p$, (7.4) has exchangeable columns. Moreover

$$\begin{aligned} x^1_{0:k|k-1} &\to u^1_{0:k|k-1},\ x^1_{0:k|k} \to u^1_{0:k|k} \\ \bar{x}_{0:k|k-1} &= \frac{1}{N}\sum_{l=1}^N x^l_{0:k|k-1} \to E\left(u^1_{0:k|k-1}\right),\ \bar{x}_{0:k|k} = \frac{1}{N}\sum_{l=1}^N x^l_{0:k|k} \to E\left(u^1_{0:k|k}\right) \\ P^N_{0:k,0:k|k} &\to P_{0:k,0:k|k}, \end{aligned}$$

in $L^{\mathbf{p}}$ as $N \to \infty, \forall\ \mathbf{p} \in [1, \infty)$.

7.4 On the convergence of LM-EnKS algorithm

The final aim of this section is to give the limit of each iteration of LM-EnKS algorithm (Algorithm 5.1), when the ensemble size goes to infinity and the finite differences parameter goes to zero. We give the proof in the simple case when the regularization parameter γ is fix over iterations and all the iterations are accepted. In this case, it is convenient to consider the joint observation at time k on the increment δx_k (the regularization observation and the observation y_k), instead of considering each observation alone. From Section 5.1.1 we have the observation on the increment is (equation 5.8):

$$d_k = \boldsymbol{H}_k \delta x_k + w_k,\ w_k \sim N(0, R_k)$$

and the new observation which arise from regularization is (equation 5.9):

$$0 = \delta x_k + e_k,\ e_k \sim N\left(0, \frac{S_k}{\gamma^2}\right).$$

Therefore the joint observation is:

$$\begin{bmatrix} d_k \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_k \\ I \end{bmatrix} \delta x_k + \begin{bmatrix} v_k \\ e_k \end{bmatrix},\ \begin{bmatrix} v_k \\ e_k \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{pmatrix} R_k & 0 \\ 0 & \frac{S_k}{\gamma^2} \end{pmatrix}\right). \tag{7.5}$$

If we denote $\begin{bmatrix} d_k \\ 0 \end{bmatrix}$, $\begin{bmatrix} \boldsymbol{H}_k \\ I \end{bmatrix}$, $\begin{bmatrix} v_k \\ e_k \end{bmatrix}$ and $\begin{pmatrix} R_k & 0 \\ 0 & \frac{S_k}{\gamma^2} \end{pmatrix}$ by $\tilde{d}_k$, $\tilde{\boldsymbol{H}}_k$, $\tilde{v}_k$ and $\tilde{R}_k$ respectively, then the equation (7.5) becomes simply:

$$\tilde{d}_k = \tilde{\boldsymbol{H}}_k \delta x_k + \tilde{v}_k,\ \tilde{v}_k \sim N\left(0, \tilde{R}_k\right).$$

Hence to avoid this new notations and without loss of generality, it is enough to do the analysis in the case when γ is equal to 0. Also, for the simplicity reason we assume that there is no observation in the state x_0.

Algorithm 7.1: Gauss-Newton algorithm to solve 4DVAR problem

Initialization

Select $x^0 = x^0_{0:p}$

For $j = 1, 2, \ldots$

$$x^j = \arg\min_{x_{0:p}} \frac{1}{2}\left(\|x_0 - x_b\|^2_{B^{-1}} + \sum_{k=1}^{p} \left\| x_k - \mathcal{M}_k(x^{j-1}_{k-1}) - \mathcal{M}'_k(x^{j-1}_{k-1})\left(x_{k-1} - x^{j-1}_{k-1}\right)\right\|^2_{Q_k^{-1}} \right.$$
$$\left. + \sum_{k=1}^{p} \left\| y_k - \mathcal{H}_k(x^{j-1}_k) - \mathcal{H}'_k(x^{j-1}_k)\left(x_k - x^{j-1}_k\right)\right\|^2_{R_k^{-1}} \right).$$

Algorithm 7.2: Gauss-Newton-EnKS with derivatives method

Initialization

Select $x^0 = x^0_{0:p}$ the same starting point as in Algorithm 7.1, and $N_0 \geq 2$ (set the initial ensemble $x^{0,l}_{0:p|p} = x^0,\ l = 1, \ldots, N_0$).

For $j = 0, 1, 2, \ldots$. Choose an ensemble size $N_{j+1} \geq 2$.

1. Generate the initial ensemble:

$$\delta x^{j+1,l}_{0|0} = x_b - x^j_0 + v^{j+1,l}_b,\ v^{j+1,l}_b \sim N(0, B),\ l = 1, \ldots, N.$$

2. For $k = 1, 2, \ldots, p$

   (a) With $\left[\delta x^{j+1,l}_{0:k-1|k-1}\right]_{l=1}^{N_{j+1}}$ already computed:

$$\begin{aligned} \delta x^{j+1,l}_{k|k-1} &= \mathcal{M}'_k\left(x^j_{k-1}\right)\delta x^{j+1,l}_{k-1|k-1} + \mathcal{M}_k\left(x^j_{k-1}\right) - x^j_{k-1} \\ &\quad + v^{j+1,l}_k,\ v^{j+1,l}_k \sim N(0, Q_k),\ l = 1, \ldots, N. \end{aligned} \tag{7.6}$$

   (b) Bayesian update for the observation:

$$\begin{aligned} \delta x^{j+1,l}_{0:k|k} &= \delta x^{j+1,l}_{0:k|k-1} + P^{N_{j+1}}_{0:k,0:k|k-1} \begin{bmatrix} 0 \\ \vdots \\ \mathcal{H}'^{\top}_k\left(x^j_k\right) \end{bmatrix} \\ &\quad \left(R_k + \mathcal{H}'_k\left(x^j_k\right) P^{N_{j+1}}_{k,k|k-1} \mathcal{H}'^{\top}_k\left(x^j_k\right)\right)^{-1} \left(y_k - \mathcal{H}_k\left(x^j_k\right)\right. \\ &\quad \left. - w^{j+1,l}_k - \mathcal{H}'_k\left(x^j_k\right)\delta x^{j+1,l}_{k|k-1}\right),\ w^{j+1,l}_k \sim N(0, R_k), \end{aligned} \tag{7.7, 7.8}$$

3. Set $x^{j+1,l}_{0:p|p} = x^j_{0:p} + \delta x^{j+1,l}_{0:p|p},\ l = 1, \ldots, N_{j+1}$. Then set

$$x^{j+1}_{0:p} = \bar{x}^{j+1,N_{j+1}}_{0:p|p} = \frac{1}{N_{j+1}} \sum_{l=1}^{N_{j+1}} x^{j+1,l}_{0:p|p}.$$

In this simple case, Algorithm 5.1 becomes Algorithm 7.3 (Gauss-Newton algorithm with EnKS as linear solver).

Algorithm 7.2 presents the version of Algorithm 7.3 where the derivatives arising in EnKS at each iteration are not approximated by finite differences. For the clarity reasons we will remind in details each step of the Algorithms 7.2 and 7.3, and also we will recall the Gauss-Newton algorithm (Algorithm 7.1) when used to solve the weak constraint 4DVAR problem:

Algorithm 7.3: Gauss-Newton-EnKS method

Initialization

Choose the constant $\tau \in (0,1]$, x^0 the same starting point as in Algorithm 7.1, and $N_0 \geq 2$ (set the initial ensemble $x_{0:p|p}^{0,l,\tau} = x^0,\ l = 1,\ldots,N_0$).

For $j = 0,1,2,\ldots$. Choose N_{j+1} the same as in Algorithm 7.2.

1. Generate the initial ensemble:

$$\delta x_{0|0}^{j+1,l,\tau} = x_b - x_0^j + v_b^{j+1,l},\ v_b^{j+1,l},\ l = 1,\ldots,N.$$

2. For $k = 1,2,\ldots,p$

   (a) With $\left[\delta x_{0:k-1|k-1}^{j+1,l,\tau}\right]_{l=1}^{N_{j+1}}$ already computed:

$$\begin{aligned}\delta x_{k|k-1}^{j+1,l,\tau} &= \frac{\mathcal{M}_k\left(x_{k-1}^j + \tau\delta x_{k-1|k-1}^{j+1,l,\tau}\right) - \mathcal{M}_k\left(x_{k-1}^j\right)}{\tau} + \mathcal{M}_k\left(x_{k-1}^j\right) \\ &\quad -x_{k-1}^j + v_k^{j+1,l},\ v_k^{j+1,l}.\end{aligned} \tag{7.9}$$

   (b) Bayesian update for the observation:

$$\begin{aligned}\delta x_{0:k|k}^{j+1,l,\tau} &= \delta x_{0:k|k-1}^{j+1,l,\tau} + \frac{1}{N_{j+1}-1} E_{0:k}^{j+1,\tau}\left(Z_{0:k}^{j+1,\tau}\right)^\top \\ &\quad \left(R_k + Z_k^{j+1,\tau}\left(Z_k^{j+1,\tau}\right)^\top\right)^{-1}\left(y_k - \mathcal{H}_k\left(x_k^j\right)\right. \\ &\quad \left. -w_k^{j,l} - \frac{\mathcal{H}_k\left(x_k^j + \tau\delta x_{k|k-1}^{j+1,l}\right) - \mathcal{H}_k\left(x_k^j\right)}{\tau}\right),\ \text{where}\end{aligned}$$

$$E_\ell^{j+1,\tau} = \left[e_\ell^{j+1,1,\tau},\ldots,e_\ell^{j+1,N_{j+1},\tau}\right],\ e_\ell^{j+1,l,\tau} = x_{\ell|k-1}^{j+1,l,\tau} - \frac{1}{N}\sum_{i=1}^{N} x_{\ell|k-1}^{j+1,i,\tau},$$

$$Z_\ell^{j+1,\tau} = \left[z_\ell^{j+1,1,\tau},\ldots,z_\ell^{j+1,N_{j+1},\tau}\right],\ z_\ell^{j+1,l,\tau} = \frac{\mathcal{H}_\ell\left(x_\ell^j + \tau e_\ell^{j+1,l,\tau}\right) - \mathcal{H}_\ell\left(x_\ell^j\right)}{\tau}$$

$$\ell = 0,\ldots,k,\ l = 1,\ldots,N_{j+1}.$$

3. Set $x_{0:p|p}^{j+1,l,\tau} = x_{0:p}^j + \delta x_{0:p|p}^{j+1,l,\tau},\ l = 1,\ldots,N_{j+1}$. Then set

$$x_{0:p}^{j+1} = \bar{x}_{0:p|p}^{j+1,N_{j+1},\tau} = \frac{1}{N_{j+1}}\sum_{l=1}^{N_{j+1}} x_{0:p|p}^{j+1,l,\tau}.$$

$$\min_{x_0,\ldots,x_p\in\mathbf{R}^n} f(x_0,\ldots,x_p) = \frac{1}{2}\left(\|x_0-x_b\|^2_{B^{-1}} + \sum_{k=1}^{p}\|x_k-\mathcal{M}_k(x_{k-1})\|^2_{Q_k^{-1}} + \sum_{k=1}^{p}\|y_k-\mathcal{H}_k(x_k)\|^2_{R_k^{-1}}\right).$$

Note that at each iteration j, Algorithms 7.3 and 7.2 generate ensembles of size N_j, and not necessary ensembles with the same size for all iterations. Moreover, for $k=0,\ldots,p$, the random vectors $[v_k^{j,l}]_{l=1}^N$ and $[w_k^{j,l}]_{l=1}^N$ are the same in the two Algorithms.

We summarize the differences between Algorithms 7.1, 7.2 and 7.3:

- The first one is the classical Gauss-Newton algorithm and it solves exactly the linearized subproblem (no approximation in the solution of the linearized subproblem).
- The second algorithm is called here the Gauss-Newton-EnKS with derivatives. It approximates the solution of the linearized subproblem using an EnKS, and it does not approximate the derivatives of the model and observation operators arising in EnKS (the approximation in the solution of the linearized subproblem arises only from the use of the ensembles).
- The third algorithm, is called here the Gauss-Newton-EnKS. It approximates the solution of the linearized subproblem using an EnKS without derivatives, meaning that it approximates derivatives of the model and observation operators arising in EnKS with finite differences as described in Section 5.1.2 (the approximation in the solution of the linearized subproblem arises from the use of the ensembles and the finite differences to approximate derivatives).

The goal of the following will be to find the limit of each Algorithm 7.3 iteration as $\tau\to 0$ and/or $\min\{N_1,\ldots,N_j\}\to\infty$ (equivalently $N_1\to\infty,\ldots,N_j\to\infty$).

For simplicity, in the following, when there is no confusion we drop the index j of N_j

7.4.1 Convergence when the finite differences parameter goes to zero

The aim of this section is to study the asymptotic behavior of Algorithms 7.3 as the finite differences parameter $\tau\to 0$. More precisely we show that when $\tau\to 0$, each Algorithm 7.3 iteration converges to its corresponding iteration of Algorithm 7.2 in probability.

We start by the following two technical lemmas which will be used later to prove the convergence.

Lemma 7.11. *Let (x_τ) and (y_τ) be 2 functions of $\tau>0$, f a function twice continuously differentiable, and $\lim_{\tau\to 0} x_\tau = x$ and $\lim_{\tau\to 0} y_\tau = y$. Then*

$$\frac{f(x_\tau+\tau y_\tau)-f(x_\tau)}{\tau}\to f'(x)y \quad \text{as } \tau\to 0.$$

$$\begin{aligned} \textit{i.e.,} \quad & \forall\ \epsilon >,\ \exists\ \tau_0 > 0,\ \eta > 0 \textit{ such that } \forall\ \tau \leq \tau_0,\ \|x_\tau - x\| \leq \eta,\ \textit{and } \|y_\tau - y\| \leq \eta \\ & \textit{implies } \left\| \frac{f\left(x_\tau + \tau y_\tau\right) - f\left(x_\tau\right)}{\tau} - f'(x)y \right\| \leq \epsilon. \end{aligned}$$

Proof. Let us define the following function:

$$t \to \phi(t) = f\left(x_\tau + t\tau y_\tau\right).$$

From Taylor expansion with integral remainder formula we have:

$$\phi(1) = \phi(0) + \phi'(0) + \int_0^1 (1-t)\phi''(t)dt. \tag{7.10}$$

One can, easily, shows that:

$$\begin{aligned} \phi'(0) &= \tau f'\left(x_\tau\right) y_\tau, & (7.11) \\ \phi''(t) &= \tau^2 y_\tau^\top f''\left(x_\tau + t\tau y_\tau\right) y_\tau. & (7.12) \end{aligned}$$

Substituting equations (7.11) and (7.12) into equation (7.10) gives:

$$f\left(x_\tau + \tau y_\tau\right) = f(x_\tau) + \tau f'\left(x_\tau\right) y_\tau + \tau^2 \int_0^1 (1-t) y_\tau^\top f''\left(x_\tau + t\tau y_\tau\right) y_\tau dt.$$

Therefore

$$\begin{aligned} \left\| \frac{f\left(x_\tau + \tau y_\tau\right) - f\left(x_\tau\right)}{\tau} - f'(x)y \right\| &\leq \left\| f'\left(x_\tau\right) y_\tau - f'(x)y \right\| \\ &+ \left\| \tau \int_0^1 (1-t) y_\tau^\top f''\left(x_\tau + t\tau y_\tau\right) y_\tau dt \right\|. \end{aligned}$$

Since the function f is twice continuously differentiable, $\lim_{\tau\to 0} x_\tau = x$ and $\lim_{\tau\to 0} y_\tau = y$ then:

$$\lim_{\tau\to 0} \left\| f'\left(x_\tau\right) y_\tau - f'(x)y \right\| + \left\| \tau \int_0^1 (1-t) y_\tau^\top f''\left(x_\tau + t\tau y_\tau\right) y_\tau dt \right\| = 0,$$

thus

$$\lim_{\tau\to 0} \frac{f(x_\tau + \tau y_\tau) - f(x_\tau)}{\tau} = f'(x)y.$$

□

The following lemma extends the result of the previous lemma (Lemma 7.11) to the case where (x_τ) and (y_τ) are functions of random vectors.

Lemma 7.12. *Let (x_τ) and (y_τ) be 2 functions of random vectors for $\tau > 0$, such that $\lim_{\tau\to 0} x_\tau = x$ and $\lim_{\tau\to 0} y_\tau = y$ in probability, and f is twice continuously differentiable. Then*

$$\frac{f(x_\tau + \tau y_\tau) - f(x_\tau)}{\tau} \to f'(x)y \textit{ in probability, as } \tau \to 0.$$

Proof. Let $\epsilon > 0$, $\tilde{\epsilon} > 0$. On one hand, we have

$$\lim_{\tau \to 0} x_\tau = x, \text{ and } \lim_{\tau \to 0} y_\tau = y$$

in probability, implies that $\forall \eta > 0, \ \exists \ \tau_\eta > 0$,

$$\forall \ \tau < \tau_\eta, \ \mathbb{P}(\Omega_{x,\eta}) \geq 1 - \tilde{\epsilon}/2 \text{ and } \mathbb{P}(\Omega_{y,\eta}) \geq 1 - \tilde{\epsilon}/2, \tag{7.13}$$

where

$$\Omega_{x,\eta} = \{\omega : \|x_\tau(\omega) - x(\omega)\| \leq \eta\} \text{ and } \Omega_{y,\eta} = \{\omega : \|y_\tau(\omega) - y(\omega)\| \leq \eta\}.$$

On the other hand, we have the function f is twice continuously differentiable therefore using Lemma 7.11, $\exists \ \tilde{\eta} > 0$, and $\tau_1 > 0$ such that for every ω and τ which verifies $\tau \leq \tau_1$, $\|x_\tau(\omega) - x(\omega)\| \leq \tilde{\eta}$, and $\|y_\tau(\omega) - y(\omega)\| \leq \tilde{\eta}$ implies

$$\left\| \frac{f\left(x_\tau(\omega) + \tau y_\tau(\omega)\right) - f\left(x_\tau(\omega)\right)}{\tau} - f'(x(\omega))y(\omega) \right\| \leq \epsilon.$$

Thus we have $\forall \tau \leq \min(\tau_{\tilde{\eta}}, \tau_1)$

$$\Omega_{x,\tilde{\eta}} \cap \Omega_{y,\tilde{\eta}} \subset \left\{ \omega : \ \left\| \frac{f\left(x_\tau(\omega) + \tau y_\tau(\omega)\right) - f\left(x_\tau(\omega)\right)}{\tau} - f'(x(\omega))y(\omega) \right\| \leq \epsilon \right\}.$$

Moreover, using (7.13), we have:

$$\mathbb{P}(\Omega_{x,\tilde{\eta}} \cap \Omega_{x,\tilde{\eta}}) = 1 - \mathbb{P}(\bar{\Omega}_{x,\tilde{\eta}} \cup \bar{\Omega}_{y,\tilde{\eta}}) \geq 1 - \mathbb{P}(\bar{\Omega}_{x,\tilde{\eta}}) - \mathbb{P}(\bar{\Omega}_{y,\tilde{\eta}}) \geq 1 - \tilde{\epsilon}, \text{ thus}$$

$$\mathbb{P}\left(\left\{ \omega : \ \left\| \frac{f\left(x_\tau(\omega) + \tau y_\tau(\omega)\right) - f\left(x_\tau(\omega)\right)}{\tau} - f'(x(\omega))y(\omega) \right\| \leq \epsilon \right\} \right) \geq 1 - \tilde{\epsilon}.$$

□

In the case where f is a vector function, the previous lemmas hold. For the proof it is enough to consider the previous proofs for each component of the function f.

Theorem 7.13. *Under the assumption that the model and observation operators, $\mathcal{M}_k$, and $\mathcal{H}_k$ are twice continuously differentiable for any $k = 1, \ldots, p$, then: when $\tau \to 0$, at each iteration j of Algorithm 7.3, $\forall \ l = 1, \ldots, N$, $x^{j,l,\tau}_{0:k|k} \to x^{j,l}_{0:k|k}$ in probability, where $x^{j,l,\tau}_{0:k|k}$ and $x^{j,l}_{0:k|k}$ are the l-th members of the ensembles generated at the j-th iteration of Algorithms 7.3 and 7.2 respectively.*

Proof. The proof is done by induction on j, let $l \in \{1, \ldots, N\}$, for $j = 0$, we have $x^{0,l,\tau} = x^{0,l} = x^0$. For $j \geq 1$, we use induction on time step k. For $k = 0$, we have $x^{j,l,\tau}_{0|0} = x_b + v^{j,l}_b$, hence

$$\lim_{\tau \to 0} x^{j,l,\tau}_{0|0} = x_b + v^l_b = x^{j,l}_{0|0} \text{ in probability.}$$

We recall that the j-th iterates generated by Algorithm 7.3 and 7.2 are equal to $\bar{x}^{j,N,\tau}_{0:p|p}$ and $\bar{x}^{j,N}_{0:p|p}$ respectively.

For $k = 1, \ldots, p$, we have from Lemma 7.12, and induction assumption when $\tau \to 0$

$$\frac{\mathcal{M}_k\left(\bar{x}^{j-1,N,\tau}_{k-1|p} + \tau\delta x^{j,l,\tau}_{k-1|k-1}\right) - \mathcal{M}_k\left(\bar{x}^{j-1,N,\tau}_{k-1|p}\right)}{\tau} \text{ converges in probability to}$$
$$\mathcal{M}'_k\left(\bar{x}^{j-1,N}_{k-1|p}\right)\delta x^{j,l}_{k-1|k-1}, \text{ and}$$
$$\frac{\mathcal{H}_k\left(\bar{x}^{j-1,N,\tau}_{k|p} + \tau\delta x^{j,l,\tau}_{k|k-1}\right) - \mathcal{H}_k\left(\bar{x}^{j-1,N,\tau}_{k|p}\right)}{\tau} \text{ converges in probability to}$$
$$\mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)\delta x^{j,l}_{k|k-1},$$

therefore, using continuous mapping theorem (Lemma 7.7) we conclude the following convergences in probability as $\tau \to 0$:

$$\begin{aligned} x^{j,l,\tau}_{k|k-1} &= \frac{\mathcal{M}_k\left(\bar{x}^{j-1,N,\tau}_{k-1|p} + \tau\delta x^{j,l,\tau}_{k-1|k-1}\right) - \mathcal{M}_k\left(\bar{x}^{j-1,N,\tau}_{k-1|p}\right)}{\tau} + \mathcal{M}_k\left(\bar{x}^{j-1,N,\tau}_{k-1|p}\right) + v^{j,l}_k \to \\ x^{j,l}_{k|k-1} &= \mathcal{M}'_k\left(\bar{x}^{j-1,N}_{k-1|p}\right)\delta x^{j,l}_{k-1|k-1} + \mathcal{M}_k\left(\bar{x}^{j-1,N}_{k-1|p}\right) + v^{j,l}_k. \end{aligned} \tag{7.14}$$

$$e^{j,l,\tau}_\ell = x^{j,l,\tau}_{\ell|k-1} - \frac{1}{N}\sum_{i=1}^N x^{j,i,\tau}_{\ell|k-1} \to e^{j,l}_\ell = x^{j,l}_{\ell|k-1} - \frac{1}{N}\sum_{i=1}^N x^{j,i}_{\ell|k-1} \quad \ell = 0, \ldots, k,\ l = 1, \ldots, N. \tag{7.15}$$

Using the latter convergence and Lemma 7.12 we conclude that:

$$z^{j,l,\tau}_\ell = \frac{\mathcal{H}_\ell\left(\bar{x}^{j-1,N,\tau}_{k|p} + \tau e^{j,l,\tau}_\ell\right) - \mathcal{H}_\ell\left(x^j_\ell\right)}{\tau} \to z^{j,l}_\ell = \mathcal{H}'_\ell\left(\bar{x}^{j-1,N}_{k-1|p}\right)e^{j,l}_\ell, \tag{7.16}$$
$$\ell = 0, \ldots, k,\ l = 1, \ldots, N.$$

Therefore using the convergences in (7.14), (7.15) and (7.16) and the continuous mapping theorem once more gives:

$$x^{j,l,\tau}_{0:k|k} =\!\!\to x^{j,l}_{0:k|k} \text{ in probability, as } \tau \to 0.$$

□

Corollary 7.14. *For any time index $k = 0, \ldots, p$ we have:*

$$x^{j,N,\tau}_{0:k} = \frac{1}{N}\sum_{l=1}^N x^{j,l,\tau}_{0:k|k} \to x^{j,N}_{0:k} = \frac{1}{N}\sum_{l=1}^N x^{j,l}_{0:k|k}, \text{ in probability as } \tau \to 0.$$

7.4.2 Convergence when the ensemble sizes go to infinity

In this section, we study the asymptotic behavior of Algorithm 7.2 when the ensemble sizes $\{N_0, \ldots, N_j\}$ go to infinity. We will show that each Algorithm 7.2 iteration converges to its corresponding iteration of Algorithm 7.1 in $L^{\mathbf{p}}$. We recall that for $k = 0, \ldots, p$, $X^j_{0:k|k} = [x^{j,l}_{0:k|k}]_{l=1}^{N_j}$ are denoting the ensembles generated by Algorithm 7.2 at iteration j. For theoretical purposes, we define by induction on j an ensemble $U^j_{0:k|k} = [u^{j,l}_{0:k|k}]_{l=1}^{N_j}$ of size N_j as follows:

1. For $j = 0$, we set $U^0_{0:k|k} = X^0_{0:k|k}$.
2. For $j = 1, 2, \ldots$
   (a) For $k = 0$, $u^{j,l}_{0|0} = x^j_{0|0}$, $\forall\ l = 1, \ldots, N_j$.
   (b) For $k = 1, \ldots, p$,

$$\begin{aligned} u^{j,l}_{k|k-1} &= \mathcal{M}'_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) u^{j,l}_{k-1|k-1} - \mathcal{M}'_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) E\left(u^{j-1,1}_{k-1|p}\right) \\ &+ \mathcal{M}_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) + v^{j,l}_k, \\ u^{j,l}_{0:k|k} &= u^{j,l}_{0:k|k-1} + P^j_{0:k,0:k|k-1}\tilde{\mathcal{H}}'^{\top}_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)\left(R_k + \tilde{\mathcal{H}}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)\right. \\ &\quad \left. P^j_{0:k,0:k|k-1}\tilde{\mathcal{H}}'^{\top}_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)\right)^{-1}\left(y_k - \mathcal{H}_k(E(u^{j-1,1}_{k|p})) - w^{j,l}_k\right. \\ &\quad \left. -\mathcal{H}'_k(E(u^{j-1,1}_{k|p}))u^{j,l}_{k|k-1} + \mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right) E\left(u^{j-1,1}_k\right)\right), \\ &\quad \forall\ l = 1, \ldots, N_j. \end{aligned}$$

where $P^j_{0:k,0:k|k-1}$ is the exact covariance matrix of $u^{j,1}_{0:k,0:k|k-1}$, and $\tilde{\mathcal{H}}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right) = \left[0, \ldots, \mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)\right]$.

Assumption 7.4.1. The model and observation operators, $\mathcal{M}_k$, and $\mathcal{H}_k$ are twice continuously differentiable and have at most polynomial growth at infinity, and their Jacobians have also at most polynomial growth at infinity. i.e. there exists $\kappa > 0$, and $s \geq 0$, such that $\|\mathcal{M}_k(x)\| \leq \kappa\left(1 + \|x\|^s\right)$, $\|\mathcal{M}'_k(x)\| \leq \kappa\left(1 + \|x\|^s\right)$, $\|\mathcal{H}_k(x)\| \leq \kappa\left(1 + \|x\|^s\right)$, and $\|\mathcal{H}'_k(x)\| \leq \kappa\left(1 + \|x\|^s\right)$ for all $k = 0, \ldots, p$, and x.
Note that we chose the same κ and the same s for all the operators to avoid unnecessary notations.

Each member of the ensemble $X^j_{0:k|k} = [x^{j,l}_{0:k|k}]_{l=1}^{N_j}$ generated by Algorithm 7.2 at iteration j is considered as a sequence of N_j (the ensemble size). For fixed member index l, and time step k we denote these sequence by $\left\{x^{j,l}_{0:k|k}\right\}_{N_j=2}^{\infty}$.

Lemma 7.15. *Let Assumption 7.4.1 holds. Then* $\left\{x^{j,l}_{0:k|k}\right\}_{N_j=2}^{\infty}$ *is bounded in* $L^{\mathbf{p}}$ *(independently from* N_j*,* l*,* $N_0, \ldots, N_{j-1}$*), for any* $\mathbf{p} \in [1, \infty)$*, any* $j \geq 0$*, any* $l \in \{1, \ldots, N_j\}$*, and any* $k = 0, \ldots, p$.

Proof. Let $\mathbf{p} \in [1, \infty)$, and $l \in \{1, \ldots, N_j\}$. The proof is done by induction on j, for $j = 0$, $x^{0,l} = x^0$ is bounded in $L^{\mathbf{p}}$. For $j > 0$, we proceed by induction on time step, for $k = 0$, $x^{j,l}_{0|0}$ is Gaussian, so $\left\{x^{j,l}_{0|0}\right\}_{N_j=2}^{\infty}$ is bounded in $L^{\mathbf{p}}$. For $k = 1, \ldots, p$, from (7.9) we conclude that:

$$\begin{aligned} \left\|x^{j,l}_{k|k-1}\right\|_{\mathbf{p}} \leq \left\|\mathcal{M}'_k\left(\bar{x}^{j-1,N}_{k-1|p}\right)\right\|_{2\mathbf{p}}\left(\left\|x^{j,l}_{k-1|k-1}\right\|_{2\mathbf{p}} + \left\|\bar{x}^{j-1,N}_{k-1|p}\right\|_{2\mathbf{p}}\right) \\ + \left\|\mathcal{M}_k\left(\bar{x}^{j-1,N}_{k-1|p}\right)\right\|_{\mathbf{p}} + \left\|v^{j,l}_k\right\|_{\mathbf{p}}. \end{aligned}$$

Under assumption (7.4.1), and the fact that $v_k^{j,l}$ is normally distributed, there exists a constant $C_{\mathbf{p}}$ such that:

$$\begin{aligned}\left\|x_{k|k-1}^{j,l}\right\|_{\mathbf{p}} \leq \kappa C_{\mathbf{p}}\left(1+\left\|\bar{x}_{k-1|p}^{j-1,N}\right\|_{2\mathbf{p}s}^{s}\right)\left(\left\|x_{k-1|k-1}^{j,l}\right\|_{2\mathbf{p}}+\left\|\bar{x}_{k-1|p}^{j-1,N}\right\|_{2\mathbf{p}}\right)\\ +\kappa C_{\mathbf{p}}\left(1+\left\|\bar{x}_{k-1|p}^{j-1,N}\right\|_{\mathbf{p}s}^{s}\right)+C_{\mathbf{p}},\end{aligned}$$

hence, using induction assumptions on j and k we have $\left\{x_{0:k|k-1}^{j,l}\right\}_{N_j=2}^{\infty}$ is bounded in $L^{\mathbf{p}}$. From equation (7.7) we conclude that:

$$\begin{aligned}\left\|x_{0:k|k}^{j,l}\right\|_{\mathbf{p}} \leq \left\|x_{0:k|k-1}^{j,l}\right\|_{\mathbf{p}}+\left\|P_{0:k,0:k|k-1}^{j,N}\right\|_{8\mathbf{p}}\left\|\mathcal{H}_k'^{\top}\left(\bar{x}_{k|p}^{j-1,N}\right)\right\|_{8\mathbf{p}}\\ \left\|\left(R_k+\mathcal{H}_k'\left(\bar{x}_{k|p}^{j-1,N}\right)P_{k,k|k-1}^{j,N}\mathcal{H}_k'^{\top}\left(\bar{x}_{k|p}^{j-1,N}\right)\right)^{-1}\right\|_{4\mathbf{p}}\left(\|y_k\|\right.\\ +\left\|\mathcal{H}_k\left(\bar{x}_{k|p}^{j-1,N}\right)\right\|_{2\mathbf{p}}+\left\|w_k^{j,l}\right\|_{2\mathbf{p}}\\ \left.+\left\|\tilde{\mathcal{H}}_k'\left(\bar{x}_{k|p}^{j-1,N}\right)\right\|_{4\mathbf{p}}\left(\left\|x_{k|k-1}^{j,l}\right\|_{4\mathbf{p}}+\left\|\bar{x}_{k|p}^{j-1,N}\right\|_{4\mathbf{p}}\right)\right).\end{aligned}$$

Since R_k is positive definite and $P_{0:k,0:k|k-1}^{j,N}$ is positive semi definite, hence

$$\left\|\left(R_k+\tilde{\mathcal{H}}_k'^{\top}\left(\bar{x}_{k|p}^{j-1,N}\right)P_{0:k,0:k|k-1}^{j,N}\tilde{\mathcal{H}}_k'\left(\bar{x}_{k|p}^{j-1,N}\right)\right)^{-1}\right\|_{4\mathbf{p}} \leq \left\|R_k^{-1}\right\|. \tag{7.17}$$

From [85, lemma 31] we have:

$$\left\|P_{0:k,0:k|k-1}^{j,N}\right\|_{8\mathbf{p}} \leq 2\left\|x_{0:k|k-1}^{j,1}\right\|_{16\mathbf{p}}^{2}. \tag{7.18}$$

From the inequalities (7.17) and (7.18), assumption 7.4.1, and the fact that $w_k^{j,l}$ is normally distributed there exists a constant $\tilde{C}_{\mathbf{p}}$ such that:

$$\begin{aligned}\left\|x_{0:k|k}^{j,l}\right\|_{\mathbf{p}} \leq \left\|x_{0:k|k-1}^{j,l}\right\|_{\mathbf{p}}+2\left\|x_{0:k|k-1}^{j,1}\right\|_{16\mathbf{p}}^{2}\kappa\tilde{C}_{\mathbf{p}}\left(1+\left\|\bar{x}_{k|p}^{j-1,N}\right\|_{8\mathbf{p}s}^{s}\right)\left\|R_k^{-1}\right\|\left(\|y_k\|\right.\\ +\kappa\tilde{C}_{\mathbf{p}}\left(1+\left\|\bar{x}_{k|p}^{j-1,N}\right\|_{2\mathbf{p}s}^{s}\right)+\tilde{C}_{\mathbf{p}}\\ \left.+\kappa\tilde{C}_{p}\left(1+\left\|\bar{x}_{k|p}^{j-1,N}\right\|_{4\mathbf{p}s}^{s}\right)\left(\left\|x_{k|k-1}^{j,l}\right\|_{4\mathbf{p}}+\left\|\bar{x}_{k|p}^{j-1,N}\right\|_{4\mathbf{p}}\right)\right),\end{aligned}$$

hence $\left\{x_{0:k|k}^{j,l}\right\}_{N_j=2}^{\infty}$ is bounded in $L^{\mathbf{p}}$. □

Theorem 7.16. *For each j, and $k=0,\ldots,p$,*

$$\left[X_{0:k|k}^{j};U_{0:k|k}^{j}\right]=\begin{bmatrix}X_{0:k|k}^{j}\\U_{0:k|k}^{j}\end{bmatrix}=\begin{bmatrix}x_{0:k|k}^{j,1},\ldots,x_{0:k|k}^{j,N}\\u_{0:k|k}^{j,1},\ldots,u_{0:k|k}^{j,N}\end{bmatrix}. \tag{7.19}$$

are exchangeable, and $x_{0:k|k}^{j,1}\to u_{0:k|k}^{j,1}$, $\bar{x}_{0:k|k}^{j,N}\to E\left(u_{0:k|k}^{j,1}\right)$, as $\min\{N_0,\ldots,N_j\}\to\infty$, in $L^{\mathbf{p}}$ $\forall p\in[1,\infty)$.

Proof. Let $\mathbf{p} \in [1, \infty)$. We use the induction on j, for $j = 0$, we have $\forall\ l \in \{1, \dots, N\}$, and $k \in \{0, \dots, p\}$, $x^{0,l}_{0:k|k} = x^0_{0:k} = u^{0,l}_{0:k|k}$, therefore $\left[X^0_{0:k|k}; U^0_{0:k|k}\right]$ are exchangeable, $x^{0,1}_{0:k|k} \to u^{0,1}_{0:k|k}$, and $\bar{x}^{0,N}_{0:k|k} \to E\left(u^{0,1}_{0:k|k}\right)$ in $L^{\mathbf{p}}$, as $N_0 \to \infty$. For $j > 0$, we use the induction on time index k, for $k = 0$, $\left[x^{j,l}_{0|0}\right]_{l=1}^{N}$ are i.i.d and $x^{j,l}_{0|0} = u^{j,l}_{0|0}$, therefore $\left[X^j_{0|0}; U^j_{0|0}\right]$ are exchangeable, $x^{j,1}_{0|0} \to u^{j,1}_{0|0}$, and using Law of large numbers $\bar{x}^{j,N}_{0|0} \to E\left(u^{j,1}_{0|0}\right)$ as $N \to \infty$ in $L^{\mathbf{p}}$.

For $k = 1, \dots, p$, let $l \in \{1, \dots, N\}$, we have

$$
\begin{aligned}
\begin{bmatrix} x^{j,l}_{k|k-1} \\ u^{j,l}_{k|k-1} \end{bmatrix} &= \begin{bmatrix} \mathcal{M}'_k(\bar{x}^{j-1,N}_{k-1|p}) & 0 \\ 0 & \mathcal{M}'_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) \end{bmatrix} \begin{bmatrix} x^{j,l}_{k-1|k-1} \\ u^{j,l}_{k-1|k-1} \end{bmatrix} \\
&+ \begin{bmatrix} \mathcal{M}_k\left(\bar{x}^{j-1,N}_{k-1|p}\right) - \mathcal{M}'_k\left(\bar{x}^{j-1,N}_{k-1|p}\right)\bar{x}^{j-1,N}_{k-1|p} \\ \mathcal{M}_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) - \mathcal{M}'_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) E\left(u^{j-1,1}_{k-1|p}\right) \end{bmatrix} + \begin{bmatrix} v^{j,l}_k \\ v^{j,l}_k \end{bmatrix} \\
&= F^k\left(\bar{x}^{j-1,N}_{k-1|p}, \begin{bmatrix} x^{j,l}_{k-1|k-1} \\ u^{j,l}_{k-1|k-1} \end{bmatrix}, \begin{bmatrix} v^{j,l}_k \\ v^{j,l}_k \end{bmatrix}\right),
\end{aligned}
$$

where F^k is a measurable function.

The ensemble sample mean $\bar{x}^{j-1,N}_{k-1|p}$ is invariant to a permutation of ensemble members. The matrix $V^j_k = \left[v^{j,1}_k, \dots, v^{j,N}_k\right]$ is exchangeable $\left(\left[v^{j,l}_k\right]_{l=1}^{N}\right.$ are i.i.d$\left.\right)$. From the induction assumption on k, $\begin{bmatrix} X^j_{k-1|k-1} \\ U^j_{k-1|k-1} \end{bmatrix}$ is exchangeable, and it is also independent from $\begin{bmatrix} V^j_k \\ V^j_k \end{bmatrix}$, therefore $\begin{bmatrix} X^j_{k|k-1} \\ U^j_{k|k-1} \end{bmatrix}$ is exchangeable by Lemma 7.5. From induction assumption on j and k, $\bar{x}^{j-1,N}_{k-1|p} \to E\left(u^{j-1,1}_{k-1|p}\right)$, $x^{j,1}_{k-1|k-1} \to u^{j,1}_{k-1|k-1}$ in $L^{\mathbf{p}}$, as $\min\{N_0, \dots, N_j\} \to \infty$ and using continuous mapping theorem, we conclude that when $\min\{N_0, \dots, N_j\} \to \infty$

$$
\begin{aligned}
x^{j,1}_{k|k-1} &= \mathcal{M}'_k\left(\bar{x}^{j-1,N}_{k-1|p}\right) x^{j,1}_{k-1|k-1} + \mathcal{M}_k\left(\bar{x}^{j-1,N}_{k-1|p}\right) - \mathcal{M}'_i\left(\bar{x}^{j-1,N}_{k-1|p}\right)\bar{x}^{j-1,N}_{k-1|p} + v^{j,1}_k \to \\
u^{j,1}_{k|k-1} &= \mathcal{M}'_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) u^{j,1}_{k-1|k-1} + \mathcal{M}_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) - \mathcal{M}'_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) E\left(u^{j-1,1}_{k-1|p}\right) \\
&\quad + v^{j,1}_k,
\end{aligned}
$$

in probability. From Lemma 7.15, we have $\left\{x^{j,1}_{0:k|k-1}\right\}_{N=2}^{\infty}$ is bounded $L^{\mathbf{p}}$, therefore by using the uniform integrability theorem (Lemma 7.6) we can leverage the last convergence in probability to the convergence in $L^{\mathbf{p}}$.

We have also

$$\begin{aligned}\begin{bmatrix} x^{j,l}_{0:k|k} \\ u^{j,l}_{0:k|k} \end{bmatrix} &= \begin{bmatrix} x^{j,l}_{0:k|k-1} \\ u^{j,l}_{0:k|k-1} \end{bmatrix} + \begin{bmatrix} K^N_k & 0 \\ 0 & K^j_k \end{bmatrix} \\ &\left(\begin{bmatrix} y_k + \mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)\bar{x}^{j-1,N}_{k|p} - \mathcal{H}_k\left(\bar{x}^{j-1,N}_{k|p}\right) \\ y_k + \mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)E\left(u^{j-1,1}_{k|p}\right) - \mathcal{H}_k\left(E\left(u^{j-1,1}_{k|p}\right)\right) \end{bmatrix} \right. \\ &\left. - \begin{bmatrix} w^{j,l}_k \\ w^{j,l}_k \end{bmatrix} - \begin{bmatrix} \mathcal{H}'_k(\bar{x}^{j-1,N}_{k|p}) & 0 \\ 0 & \mathcal{H}'_k(E(u^{j-1,1}_{k|p})) \end{bmatrix} \begin{bmatrix} x^{j,l}_{k|k-1} \\ u^{j,l}_{k|k-1} \end{bmatrix} \right) \\ &= F^k\left(\bar{x}^{j-1,N}_{k|p}, P^N_{0:k,0:k|k-1}, \begin{bmatrix} x^{j,l}_{k|k-1} \\ u^{j,l}_{k|k-1} \end{bmatrix}, \begin{bmatrix} w^{j,l}_k \\ w^{j,l}_k \end{bmatrix}\right)\end{aligned}$$

where

$$K^N_k = \begin{bmatrix} P^N_{0,k|k-1}\mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)^\top \\ \vdots \\ P^N_{k,k|k-1}\mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)^\top \end{bmatrix} \left(R_k + \mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)P^N_{k,k|k-1}\mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)^\top\right)^{-1},$$

$$K^j_k = \begin{bmatrix} P^j_{0,k|k-1}\mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)^\top \\ \vdots \\ P^j_{k,k|k-1}\mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)^\top \end{bmatrix} \left(R_k + \mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)P^j_{k,k|k-1}\mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)^\top\right)^{-1}$$

and F^k is a measurable function.

The ensemble sample mean $\bar{x}^{j-1,N}_{k|p}$, and the ensemble sample covariance $P^N_{0:k,0:k|k-1}$ are invariant to a permutation of ensemble members, $W^j_k = \left[w^{j,1}_k, \ldots, w^{j,N}_k\right]$ is exchangeable $\left(\left[w^{j,l}_k\right]^N_{l=1}$ are i.i.d$\right)$, we have also $\begin{bmatrix} X^j_{k|k-1} \\ U^j_{k|k-1} \end{bmatrix}$ is exchangeable, and it is independent from $\begin{bmatrix} W^j_k \\ W^j_k \end{bmatrix}$, therefore $\begin{bmatrix} X^j_{0:k|k} \\ U^j_{0:k|k} \end{bmatrix}$ is exchangeable by Lemma 7.5. We have $\bar{x}^{j-1,N}_{k|p} \to E\left(u^{j-1,1}_{k|p}\right)$, $x^{j,1}_{0:k|k-1} \to u^{j,1}_{0:k|k-1}$ in $L^{\mathbf{p}}$. From [87, lemma 3] we have $P^N_{0:k,0:k|k-1} \to P^j_{0:k,0:k|k-1}$ in probability, therefore using continuous mapping theorem $K^N_k \to K^j_k$. From the fact that the convergence in $L^{\mathbf{p}}$ induce the convergence in probability, and using again the continuous mapping theorem we conclude that:

$$\begin{aligned} x^{j,1}_{0:k|k} &= x^{j,1}_{0:k|k-1} + K^N_k\left(y_k - \mathcal{H}_k\left(\bar{x}^{j-1,N}_{k|p}\right) - w^{j,1}_k - \mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)x^{j,1}_{k|k-1}\right. \\ &\quad \left. + \mathcal{H}'_k\left(\bar{x}^{j-1,N}_{k|p}\right)\bar{x}^{j-1,N}_{k|p}\right) \to \\ u^{j,1}_{0:k|k} &= u^{j,1}_{0:k|k-1} + K^j_k\left(y_k - \mathcal{H}_k\left(E\left(u^{j-1,1}_{k|p}\right)\right) - w^{j,1}_k - \mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)u^{j,1}_{k|k-1}\right. \\ &\quad \left. + \mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)E\left(u^{j-1,1}_{k|p}\right)\right), \end{aligned}$$

in probability, when $\min\{N_0, \ldots, N_j\} \to \infty$. Then we leverage the last convergence to the convergence in $L^{\mathbf{p}}$ using the uniform integrability theorem.

□

Lemma 7.17. *$E\left(u^{j,1}_{0:p|p}\right) = x^j$, where x^j is the j-th iterate generated by Algorithm (7.1).*

Proof. The proof is done by induction on j.

For $j = 0$, we have $\forall\ l \in \{1, \dots, N\}$, $u^{0,l}_{0:p|p} = x^0_{0:p|p} = x^0$, thus $E\left(u^{0,1}_{0:p|p}\right) = x^0$. For $j > 0$, we have

$$\begin{aligned} E\left(u^{j,1}_{0:p|p}\right) = \arg\min_{x_{0:p}} \frac{1}{2}\Big(&\|x_0 - x_b\|^2_{B^{-1}} \\ &+ \sum_{k=1}^{p} \left\| x_k - \mathcal{M}_k\left(E\left(u^{j-1,1}_{k-1|p}\right)\right) - \mathcal{M}'_k(E(u^{j-1,1}_{k-1|p}))\left(x_{k-1} - E\left(u^{j-1,1}_{k-1|p}\right)\right)\right\|^2_{Q_k^{-1}} \\ &+ \sum_{k=1}^{p} \left\| y_k - \mathcal{H}_k\left(E\left(u^{j-1,1}_{k|p}\right)\right) - \mathcal{H}'_k\left(E\left(u^{j-1,1}_{k|p}\right)\right)\left(x_k - E\left(u^{j-1,1}_{k|p}\right)\right)\right\|^2_{R_k^{-1}}\Big), \end{aligned}$$

and from the induction assumption on j we have $E\left(u^{j-1,1}_{0:p|p}\right) = x^{j-1}_{0:p}$, hence

$$\begin{aligned} E\left(u^{j,1}\right) = \arg\min_{x_{0:p}} \frac{1}{2}\Bigg(&\|x_0 - x_b\|^2_{B^{-1}} + \sum_{k=1}^{p} \left\| x_k - \mathcal{M}_k(x^{j-1}_{k-1}) - \mathcal{M}'_k(x^{j-1}_{k-1})\left(x_{k-1} - x^{j-1}_{k-1}\right)\right\|^2_{Q_k^{-1}} \\ &+ \sum_{k=1}^{p} \left\| y_k - \mathcal{H}_k(x^{j-1}_k) - \mathcal{H}'_k(x^{j-1}_k)\left(x_k - x^{j-1}_k\right)\right\|^2_{R_k^{-1}}\Bigg) = x^j. \end{aligned}$$

□

Corollary 7.18. *For each j,*

$$\lim_{\min\{N_1,\dots,N_j\}\to\infty}\left(\lim_{\tau\to 0} x^{j,N,\tau}\right) = x^j,$$

in probability where $x^{j,N,\tau}$ and x^j are the j-th iterates generated by Algorithms 7.3 and 7.1 respectively.

Proof. The proof follows immediately from Theorem 7.13, Theorem 7.16, and Lemma 7.17. □

In this chapter we have shown the convergence in $L^{\mathbf{p}}$ spaces of the empirical mean and covariance of EnKS to the KS mean and covariance in the limit for large ensemble size. We have shown also that each LM-EnKS iterate converges in probability to its corresponding iterate of Algorithm 7.1 as the finite differences parameter goes to zero and then the ensemble sizes go to infinity. We think that it is possible to obtain a stronger limit result, especially to leverage the convergences in probability to convergences in $L^{\mathbf{p}}$, and show the convergence rate of the algorithms following [78]. These convergences will be further explored in the future works.

Chapter 8

Conclusions and perspectives

The thesis concentrates on the numerical methods for least squares problems, in which the gradient model is expensive or noisy and accurate only within a certain probability. Within this study, a solution method based on a Gauss-Newton technique, made globally convergent with a trust-region strategy, is considered (Levenberg-Marquardt method). We have given an application in data assimilation of the new proposed method, and also we have studied the sensitivity of the linearized subproblem solution to data, when using the singular value decomposition method.

In this work, we have contributed to the research area of least squares problems by addressing the following challenges:

(i) Solving the problem of the determination of a closed formula for the condition number of the truncated singular value solution, in the case of ill-conditioned problems and/or when the data are noisy.

(ii) Giving a variant of the Levenberg-Marquardt method to the scenarios where the linearized least squares subproblems are solved inexactly and/or the gradient model is accurate only within a certain probability.

(iii) Proposing an application of the new variant of the Levenberg-Marquardt method in data assimilation framework.

(iv) Analyzing numerically the impact of different parameters arising in the Levenberg-Marquardt method, using EnKS as a linear solver, on the iteration progress.

(v) Studying the asymptotic behavior of each iteration of the Levenberg-Marquardt algorithm, in the case where we maintain the regularization parameter fixed and we use EnKS as a linear solver.

The challenge (i) was addressed in Chapter 3 by solving the problem of the determination of the condition number of the truncated singular value solution. The expression that has been found for this condition number relies on a singular value decomposition of the problem (see (3.29)). We anticipate that the proposed formula will therefore stimulate research in several

directions. Finding good estimates of the condition number using iterative techniques would, for instance, be of crucial relevance for large scale problems. From a theoretical point of view, we also believe that the condition number may bring new insight into the problem of the detection of the truncation index of the singular value decomposition. One of the topics of future research will be to explore this issue on practical problems.

The challenge (ii) was addressed in Chapter 4 by showing how to adapt the Levenberg-Marquardt method for nonlinear least squares problems to handle the cases where the gradient of the objective function is subject to noise or only is computed accurately within a certain probability. The gradient model was then considered random in the sense of being a realization of a random vector, and assumed first order accurate under some probability p_j^* (see (4.2)). Given the knowledge of a lower bound p_j for the probability p_j^* (see Assumption 4.1.1), and an approximate solution to the subproblem which achieves at least the Cauchy decrease on the model (see Assumption 4.2.1) we have shown how to update the regularization parameter of the method in such a way that the whole approach is almost surely globally convergent. We mean by the latter convergence that a subsequence of the true objective function gradients goes to zero with probability one. We have covered also the situation where the linearized least squares subproblems, arising in the Levenberg-Marquardt method, are solved inexactly. We covered essentially two possibilities: conjugate gradient and any generic inexact solution of the corresponding normal equations, which then encompasses a range of practical situations, from inexactness in linear algebra to inexactness in derivatives. This is particularly useful in the 4DVAR application to accommodate finite differences of the nonlinear operators involved. The main difficulty in the application of the new approach (Algorithm 2.8 in Chapter 4) is to ensure that the models are indeed (p_j)-probabilistically accurate, but we have presented a number of practical situations where this is achievable. It would be interesting to further explore the role of the probability p_j^* in the adaptation of the regularization parameter, and to seek better lower bounds of the probability p_j^* which may improve the convergence properties of the new approach.

The challenge (iii) was addressed in Chapter 5 by proposing to use ensemble methods, namely EnKS to approximate the subproblem solution arising when using Gauss-Newton or Levenberg-Marquardt methods to sovle the 4DVAR problem. The use of ensemble methods as a linear solver makes random approximations to the gradient. We thus showed how to adapt the approach of Levenberg-Marquardt based on random models method in this situation. We have shown that to solve the 4DVAR problem arising in data assimilation, in the framework of the application of the Levenberg-Marquardt method, when using the EnKS method for the formulation and solution of the corresponding linearized least squares subproblem, is equivalent to approximately solve a realization of a random model. We have also provided a lower bound p_j for the probability of first order accuracy (see 5.42), which renders our approach applicable and sound. We gave some numerical results to illustrate our approach by using Lorenz 63 equations as forecast model in the 4DVAR problem. Here also, it would be interesting to further investigate the better lower bounds to the probability p_j^* and to study the performance of our approach when applied to large and realistic data assimilation problems.

The challenge (iv) was addressed in Chapter 6 by giving numerical results to illustrate the LM-EnKS method (see Algorithm 5.1). The numerical experiments are done using two different

forecast models, namely Lorenz 63 model and the quasi-geostrophic model. We have analyzed mainly the impact of the following parameters which arise in the LM-EnKS algorithm:

- The ensemble size: we have shown that, in the average of several runs, for the first iterations (when the current iteration is "far" from the objective function minimum) this parameter has not a big impact on the iteration progress. However after some iterations (when the current iteration is "near" to the minimum), then the larger the ensemble size is, the better the results will be. Hence we believe that an adaptive ensemble size over iteration can be a better choice (than fixed one for all iterations). We mean by adaptive ensemble size to generate an ensemble with a small size in the first iteration and then to increase it over iterations.
- The finite differences parameter (τ): we have shown that, it is better for the first iteration to use the classical ensemble Kalman smoother as proposed in [43] to approximately solve the subproblem (which correspond to the choice of $\tau = 1$), and then to decrease the finite differences parameter to zero over iterations.
- The covariance scale parameter: we have shown that the scaling of the covariances is very important to speed up the decrease of the objective function over iterations. We have shown that, few iterations were enough to reduce significantly the objective function in the case where the covariances are scaled. But in the case where the covariances are not scaled, the algorithm needs more iterations to reduce the objective function significantly.

We conclude that the choice of the previous parameters is of crucial importance for the cost of the algorithm. One of the topics of future research will be to explore in more details the best strategies to adapt these parameters over iterations.

Finally, the challenge (v) was addressed in Chapter 7. The main results of these chapter show that:

- In the linear case, i.e., when the observation and the model operators are linear for any time step, the empirical mean and covariance of EnKS converge to the KS mean and covariance in the limit for large ensemble size in $L^{\mathbf{p}}$ for any $\mathbf{p} \in [1, \infty)$.
- In the nonlinear case, i.e., in the case where the observation and the model operators are not necessary linear, we have shown the convergence of LM-EnKS iterations (Algorithm 7.3) in the limit for large ensemble size. The convergence is in the sense that (i) each iterate generated by Algorithm 7.3 converges in probability to its corresponding iterate of Algorithm 7.2 as the finite differences parameter goes to zero (Algorithm 7.3 is asymptotically equivalent to the algorithm with derivatives as finite differences parameter goes to zero), (ii) and that each iterate generated by Algorithm 7.2 converges, in $L^{\mathbf{p}}$ for any $\mathbf{p} \in [1, \infty)$, to its corresponding iterate of Algorithm 7.1 (the classical Gauss-Newton algorithm).

These convergence issues, and more generally the asymptotic behavior of the ensemble based algorithms deserve further investigation. Here in the nonlinear case, we have given only the

limit in probability of each iteration of Algorithm 7.3 as the finite differences parameter goes to zero and the ensemble sizes go to infinity. One may try to prove stronger convergences, especially to leverage the convergences in probability to convergences in $L^{\mathbf{p}}$, and show the convergence rate of these algorithms following the spirit of [78].

There are some other general issues which worth further exploration. It would be interesting to further explore globalization strategies by developing algorithms that are similar in spirit to the classical trust regions approach [25, Chapter 6], and to extend the proposed algorithms to the case of constrained least squares problems. In both cases, we expect that the formulation of the subproblem using Lagrange multipliers is the key issue to obtain a robust algorithm.

Appendix A

Derivatives of weak constraints 4DVAR problem

When we alleviate the assumptions that the model $\mathcal{M}_k$ is perfect (i.e., the residual $v_k \neq 0$), and that $m_k = 0$. The least-squares problem (2.39) becomes:

$$\min_{x_{0:p}} \quad \frac{1}{2}\left(\|x_0 - x_\mathrm{b}\|^2_{B^{-1}} + \sum_{k=0}^{p} \|\mathcal{H}_k(x_k) - y_k\|^2_{R_k^{-1}} + \sum_{k=1}^{p} \|x_k - \mathcal{M}_k(x_{k-1}) - m_k\|^2_{Q_k^{-1}} \right).$$

The function $F : \mathbb{R}^{n(p+1)} \to \mathbb{R}^{(n+m)(p+1)}$, is defined by:

$$F(x_{0:p}) = \begin{pmatrix} B^{-1/2}(x_0 - x_\mathrm{b}) \\ Q_1^{-1/2}(\mathcal{M}_1(x_0) - x_1 + m_1) \\ \vdots \\ Q_p^{-1/2}(\mathcal{M}_p(x_{p-1}) - x_p + m_p) \\ R_0^{-1/2}(\mathcal{H}_0(x_0) - y_0) \\ \vdots \\ R_p^{-1/2}(\mathcal{H}_p(x_p) - y_p) \end{pmatrix}, \tag{A.1}$$

Note that:

$$\begin{aligned} F_{1:n}(x_{0:p}) &= B^{-1/2}(x_0 - x_\mathrm{b}), \\ F_{nk+1:n(k+1)}(x_{0:p}) &= Q_k^{-1/2}(\mathcal{M}_k(x_{k-1}) - x_k), \text{ for } k = 1, \ldots, p \\ F_{n(p+1)+mk+1:n(p+1)+m(k+1)}(x_{0:p}) &= R_k^{-1/2}(\mathcal{H}_k(x_k) - y_k), \text{ for } k = 0, \ldots, p \end{aligned}$$

where $F_{k:l}$ denotes the joint function of F_k, F_{k+1}, ..., F_l.

Computation of the derivatives

The Jacobibian of the function defined in (A.1) is:

$$
\begin{aligned}
J_F(x) &= \begin{pmatrix}
\frac{\delta F_1(x)}{\delta x_{0(1)}} & \cdots & \frac{\delta F_1(x)}{\delta x_{0(n)}} & \frac{\delta F_1(x)}{\delta x_{1(1)}} & \cdots & \frac{\delta F_1(x)}{\delta x_{p(n)}} \\
\frac{\delta F_2(x)}{\delta x_{0(1)}} & \cdots & \frac{\delta F_2(x)}{\delta x_{0(n)}} & \frac{\delta F_2(x)}{\delta x_{1(1)}} & \cdots & \frac{\delta F_2(x)}{\delta x_{p(n)}} \\
\vdots & \vdots & \ddots & \cdots & \vdots & \\
\frac{\delta F_q(x)}{\delta x_{0(1)}} & \cdots & \frac{\delta F_q(x)}{\delta x_{0(n)}} & \frac{\delta F_q(x)}{\delta x_{1(1)}} & \cdots & \frac{\delta F_q(x)}{\delta x_{p(n)}}
\end{pmatrix} = \begin{pmatrix} \nabla F_1(x)^\top \\ \vdots \\ \nabla F_q(x)^\top \end{pmatrix} \\
&= \begin{pmatrix}
J_{F_{1:n}}(x) \\
J_{F_{n+1:2n}}(x) \\
\vdots \\
J_{F_{np+1:n(p+1)}}(x) \\
J_{F_{n(p+1)+1:n(p+1)+m}}(x) \\
\vdots \\
J_{F_{(n+m-1)(p+1)+1:(n+m)(p+1)}}(x)
\end{pmatrix},
\end{aligned}
$$

where $x_{k(j)}$ denotes the j-eme component of the vector x_k. Hence the function F Jacobian is equal to:

$$
\begin{pmatrix}
B^{-1/2} & 0_n & 0_n & \dots & \dots & 0_n & 0_n \\
Q_1^{-1/2}\mathcal{M}_1'(x_0) & -Q_1^{-1/2} & 0_n & \dots & \dots & 0_n & 0_n \\
0_n & Q_2^{-1/2}\mathcal{M}_2'(x_1) & -Q_2^{-1/2} & 0_n & \ddots & \dots & 0_n \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0_n & \dots & \ddots & \dots & 0_n & Q_p^{-1/2}\mathcal{M}_p'(x_{p-1}) & -Q_p^{-1/2} \\
R_0^{-1/2}\mathcal{H}_0'(x_0) & 0_m & 0_m & \dots & \ddots & 0_m & 0_m \\
0_m & R_1^{-1/2}\mathcal{H}_1'(x_1) & 0_m & \ddots & \dots & \ddots & 0_m \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0_m & \dots & \ddots & \dots & 0_m & 0_m & R_p^{-1/2}\mathcal{H}_p'(x_p)
\end{pmatrix},
$$

and the gradient of the objective function defined in (A.1) is equal to:

$$
\nabla f(x) = \begin{pmatrix}
B^{-1}(x_0 - x_b) + \mathcal{M}_1'(x_0)^\top Q_1^{-1}(\mathcal{M}_1(x_0) - x_1 + m_1) \\
+\mathcal{H}_0'(x_0)^\top R_0^{-1}(\mathcal{H}_0(x_0) - y_0) \\
\mathcal{M}_2'(x_1)^\top Q_2^{-1}(\mathcal{M}_2(x_1) - x_2 + m_2) + Q_1^{-1}(x_1 - \mathcal{M}_1(x_0) - m_1) \\
+\mathcal{H}_1'(x_1)^\top R_1^{-1}(\mathcal{H}_1(x_1) - y_1) \\
\vdots \\
\mathcal{M}_p'(x_{p-1})^\top Q_p^{-1}(\mathcal{M}_p(x_{p-1}) - x_p + m_p) + Q_{p-1}^{-1}(x_{p-1} - \mathcal{M}_{p-1}(x_{p-2}) - m_{p-1}) \\
+\mathcal{H}_{p-1}'(x_{p-1})^\top R_{p-1}^{-1}(\mathcal{H}_{p-1}(x_{p-1}) - y_{p-1}) \\
Q_p^{-1}(x_p - \mathcal{M}_p(x_{p-1}) - m_p) + \mathcal{H}_p'(x_p)^\top R_p^{-1}(\mathcal{H}_p(x_p) - y_p)
\end{pmatrix}.
$$

Appendix B

Sherman–Morrison–Woodbury formula

Sherman-Morrison-Woodbury formula is useful when one want to update the inverse of a small rank adjustment of a given matrix.

Let $A \in \mathbf{R}^{n,n}$ be a non singular matrix having the inverse A^{-1}, and let $U \in \mathbf{R}^{n,r}$ and $V \in \mathbf{R}^{r,n}$ two matrices with $r \leq n$. If $I_r + V^\top A^{-1} U \in \mathbf{R}^{r,r}$ is invertible , then the matrix $A + UV^\top$ is invertible and

$$\left(A + UV^\top\right)^{-1} = A^{-1} - A^{-1}U\left(I_r + V^\top A^{-1}U\right)^{-1} V^\top A^{-1}.$$

Let $C \in \mathbf{R}^{r,r}$ be a non singular matrix having the inverse C^{-1}, if $C^{-1} + V^\top A^{-1} U$ is invertible, then the matrix $A + UCV^\top$ is invertible and

$$\left(A + UCV^\top\right)^{-1} = A^{-1} - A^{-1}U(V^\top A^{-1}U + C^{-1})^{-1}V^\top A^{-1}.$$

Appendix C

Test results

The mean and the standard deviation over different runs of the objective function and relative gradient, for different values of N

iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.58419e+9$	$1.37083e+7$	3.9239	0.0113705
2	$1.96093e+8$	902717	0.508512	0.00151073
3	$3.77692e+6$	27388.9	0.051462	0.000198448
4	4623.21	170.757	0.00154629	$3.24746e-5$
5	72.8078	5.86888	$5.28081e-5$	$2.21919e-5$
6	13.1061	3.17854	$4.88829e-5$	$2.3701e-5$
7	9.37865	3.60414	$5.32011e-5$	$2.02696e-5$
8	9.27288	4.58593	$5.41864e-5$	$2.3871e-5$

TABLE C.1: The mean and the standard deviation of the objective function values and relative gradient over iterations, for N = 10. This results are based on 50 runs of the algorithm.

The objective function values over iterations for different values of γ

iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.58001e+9$	$3.70922e+6$	3.92042	0.00307241
2	$1.95815e+8$	243755	0.508049	0.00040834
3	$3.77e+6$	7661.99	0.0514139	$5.56985e-5$
4	4592.79	50.8496	0.00154387	$1.04138e-5$
5	65.3008	1.67208	$1.64732e-5$	$5.35522e-6$
6	6.81256	0.421082	$1.49686e-5$	$4.19515e-6$
7	1.92003	0.228182	$1.74395e-5$	$6.5466e-6$
8	1.603	0.222602	$1.82193e-5$	$7.22388e-6$

TABLE C.2: The mean and the standard deviation of the objective function values and relative gradient over iterations, for N = 50. This results are based on 50 runs of the algorithm.

iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.58124e+9$	$1.84726e+6$	3.92146	0.00153481
2	$1.95902e+8$	126403	0.508196	0.000212041
3	$3.77161e+6$	3639.52	0.0514257	$2.6511e-5$
4	4601.22	31.7067	0.00154601	$6.19756e-6$
5	64.6354	1.01239	$1.23927e-5$	$4.38367e-6$
6	6.36547	0.283221	$1.11069e-5$	$3.77735e-6$
7	1.38782	0.134158	$1.19745e-5$	$3.78766e-6$
8	1.15234	0.1458	$1.01889e-5$	$3.89619e-6$

TABLE C.3: The mean and the standard deviation of the objective function values and relative gradient over iterations, for N = 100. This results are based on 50 runs of the algorithm.

iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.58161e+9$	$1.97718e+6$	3.92176	0.00164204
2	$1.95921e+8$	130025	0.508229	0.000218074
3	$3.77279e+6$	3974.62	0.0514344	$2.89634e-5$
4	4600.75	21.4993	0.00154601	$4.19961e-6$
5	64.4984	0.613937	$9.08844e-6$	$3.19868e-6$
6	6.25661	0.170825	$7.68826e-6$	$3.52966e-6$
7	1.18981	0.0685448	$7.9474e-6$	$2.42671e-6$
8	0.936584	0.0615694	$7.35678e-6$	$2.18636e-6$

TABLE C.4: The mean and the standard deviation of the objective function values and relative gradient over iterations, for N = 200. This results are based on 50 runs of the algorithm.

iter.	Obj. fun. mean	Obj. fun. std	Rel. grad. mean	Rel. grad. std
1	$3.58125e+9$	857555	3.92147	0.000716486
2	$1.95896e+8$	56306.8	0.508188	$9.48331e-5$
3	$3.77196e+6$	1807.75	0.0514284	$1.31439e-5$
4	4596.43	13.0518	0.0015452	$2.50977e-6$
5	64.3515	0.449598	$6.64223e-6$	$2.22441e-6$
6	6.12631	0.0651529	$4.63101e-6$	$1.46142e-6$
7	1.06267	0.0390599	$4.95395e-6$	$2.66412e-6$
8	0.827915	0.0169406	$4.87621e-6$	$1.71227e-6$

TABLE C.5: The mean and the standard deviation of the objective function values and relative gradient over iterations, for N = 500. This results are based on 50 runs of the algorithm.

Iter.	Obj. fun. when using incremental 4DVAR
0	56508.9
1	62500.3
2	38573.7
3	107781.0
4	134528.0
5	66415.5
6	44556.9
7	62627.7
8	44669.8

TABLE C.6: The objective function values over iterations when using Gauss-Newton algorithm (incremental 4DVAR algorithm).

Iter.	$\gamma = 0$	$\gamma = 0.001$	$\gamma = 0.1$	$\gamma = 1$
0	56508.9	56508.9	56508.9	56508.9
1	62824.3	59241.7	62624.6	47705.1
2	75969	109593	76888.8	39642.3
3	88286.7	94525.7	68068.9	55521
4	63846.3	49202.8	62213.3	31999.4
5	65510.3	27512.6	53272.8	16767.4
6	72536.9	14105.2	67739.8	12517.5
7	64758.1	10363.7	73657.4	9694.71
8	96220.3	6958.5	58497.4	9274.92

TABLE C.7: The objective function values over iterations for the following value of γ: 0, 0.001, 0.1, 1.

Iter.	$\gamma = 10$	$\gamma = 100$	$\gamma = 500$	$\gamma = 1000$
0	56508.9	56508.9	56508.9	56508.9
1	30470.7	33648.7	35025.3	36037.3
2	30138.6	24974.8	24916.5	25207.9
3	15324.9	21384.2	19093.2	20603.2
4	8183.56	17169.9	16413.2	16405.7
5	4177.05	17732.6	13289.7	14897.3
6	2339.36	14419.5	11317.1	13402.4
7	1667.51	12482.5	10300.8	12005.8
8	1367.02	12754.1	8666.3	10556.3

TABLE C.8: The objective function values over iterations for the following values of γ: 10, 100, 500, 1000.

Bibliography

[1] A. C. Aitken. On least squares and linear combinaitions of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1935.

[2] A. E. Akkraoui, P. Gauthier, S. Pellerin, and S. Buis. Intercomparison of the primal and dual formulations of variational data assimilation. *Quarterly Journal of The Royal Meteorological Society*, 134:1015–1025, 2008.

[3] J. L. Anderson and S. L. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127:2741–2758, 1999.

[4] G. Artana, A. Cammilleri, J. Carlier, and E. Mémin. Strong and weak constraint variational assimilations for reduced order fluid flow modeling. *Journal of Computational Physics*, 8:3264–3288, 2012.

[5] J. K. Baksalary and S. Puntanen. Characterizations of the best linear unbiased estimator in the general Gauss-Markov model with the use of matrix partial orderings. *Linear Algebra and its Applications*, 127:363–370, 1990.

[6] A. S. Bandeira, K. Scheinberg, and L. N. Vicente. Convergence of trust-region methods based on probabilistic models. Technical Report 13-11, Dept. Mathematics Univ. Coimbra, 2013.

[7] R. E. Barlow and T. Z. Irony. *Foundations of Statistical Quality Control*, volume 17 of *Lecture Notes–Monograph Series*. Institute of Mathematical Statistics, Hayward, CA, 1992.

[8] B. M. Bell. The iterated Kalman smoother as a Gauss-Newton method. *Society for Industrial and Applied Mathematics Journal on Optimization*, 4(3):626–636, 1994.

[9] A. F. Bennett. *Inverse Methods in Physical Oceanography*. Arnold and Caroline Rose Monograph Series of the American So. Cambridge University Press, 1992.

[10] M. Benzi and C. D. Meyer. A direct projection method for sparse linear systems. *Society for Industrial and Applied Mathematics, Journal of Scientific Computing*, 16(5):1159–1176, 1995.

[11] E. Bergou, S. Gratton, and J. Tshimanga. The exact condition number of the truncated singular value solution of a linear ill-posed problem. *Society for Industrial and Applied Mathematics, Journal on Matrix Analysis and Applications (SIMAX)*, 2014.

[12] E. Bergou, S. Gratton, and L. N. Vicente. Levenberg-Marquardt methods based on probabilistic gradient models and inexact subproblem solution, with application to data assimilation. Technical report, CERFACS, Toulouse, France, 2014.

[13] E. Bergou, J. Mandel, and S. Gratton. On the convergence of a non-linear ensemble Kalman smoother. Technical report, CERFACS, Toulouse, France, 2014.

[14] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley Series in Probability & Statistics. Wiley, 1994.

[15] P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, third edition, 1995.

[16] A. Bjorck. *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[17] M. Bocquet and P. Sakov. Combining inflation-free and iterative ensemble Kalman filters for strongly nonlinear systems. *Nonlinear Processes in Geophysics*, 19(3):383–399, 2012.

[18] M. Bocquet and P. Sakov. An iterative ensemble Kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, page in print, 2013.

[19] F. Bouttier and P. Courtier. Data assimilation concepts and methods. meteorological training course lecture series (ECMWF), 1999.

[20] F. Bouttier and P. Courtier. Data assimilation concepts and methods. meteorological training course lecture series (ECMWF), 2008.

[21] G. L. Bretthorst. An introduction to parameter estimation using bayesian probability theory, 1990.

[22] G. Burgers, P. J. van Leeuwen, and G. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126:1719–1724, 1998.

[23] J. Charney, R. Fjortoft, and J. V. Newman. Numerical integration of the barotropic vorticity equation. *Tellus*, 1950.

[24] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[25] R. A. Conn, I. N. Golub, and P. L. Toint. *Trust-region Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[26] M. Corazza, E. Kalnay, D. J. Patil, S. C. Yang, R. Morss, M. Cai, I. Szunyogh, B. R. Hunt, and J. A. Yorke. Use of the breeding technique to estimate the structure of the analysis "errors of the day". *Nonlinear Processes in Geophysics*, 10(3):233–243, 2003.

[27] P. Courtier, J. N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4D-VAR, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994.

[28] G. Cressman. An operational objective analysis system. *Monthly Weather Review*, 1959.

[29] T. A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[30] D. P. Dee and A. M. D. Silva. Data assimilation in the presence of forecast bias, 1997.

[31] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Serie B*, 39(1):1–38, 1977.

[33] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16)*. Society for Industrial and Applied Mathematics, 1996.

[34] J. Derber and F. Bouttier. A reformulation of the background error covariance in the ECMWF global data assimilation system. *Tellus A*, 51(2), 2011.

[35] B. Desjardins and E. Grenier. Derivation of quasigeostrophic potential vorticity equations. In *Jinqiao Duan and Bj Orn Schmalfuss*, pages 715–752, 1997.

[36] G. Desroziers, G. Hello, and J. N. Thépaut. A 4D-VAR reanalysis of the fastex. *Quarterly Journal of The Royal Meteorological Society*, 2003.

[37] A. Doucet and N. D. Freitas. Sequential Monte Carlo methods in practice. *Springer*, 2001.

[38] R. Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, fourth edition, 2010.

[39] G. Evensen. Sequential data assimilation with nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99 (C5)(10):143–162, 1994.

[40] G. Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation, 2003.

[41] G. Evensen. *Data Assimilation: the Ensemble Kalman Filter*. Springer, Berlin, 2007.

[42] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2nd edition, 2009.

[43] G. Evensen and P. J. van Leeuwen. An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6):1852–1867, 2000.

[44] C. Fandry and L. Leslie. A two-layer quasi-geostrophic model of summer trough formation in the australian subtropical easterlies. *Journal of the Atmospheric Sciences*, 41:807–817, 1984.

[45] M. Fisher and E. C. for Medium Range Weather Forecasts. *Estimation of Entropy Reduction and Degrees of Freedom for Signal for Large Variational Analysis Systems*. ECMWF technical memorandum. European Centre for Medium-Range Weather Forecasts, 2003.

[46] M. Fisher, M. Leutbecher, and G. A. Kelly. On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131(613, Part c):3235–3246, 2005.

[47] M. Fisher, Y. Trémolet, H. Auvinen, D. Tan, and P. Poli. Weak-constraint and long-window 4D-Var. Technical report, European Centre for Medium-Range Weather Forecasts, 2011.

[48] R. A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22:700–725, 1925.

[49] R. Franke, E. Barker, and J. Goerss. The use of observed data for the initial-value problem in numerical weather prediction. *Computers and Mathematics with Applications*, 16(1–2):169 – 184, 1988.

[50] P. Gauthier, M. Tanguay, S. Laroche, S. Pellerin, and J. Morneau. Extension of 3D-Var to 4D-Var: implementation of 4D-Var at the meteorological service of canada. *Monthly Weather Review*, 2007.

[51] J. L. Gauvain and C. H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994.

[52] G. H. Golub and C. F. V. Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[53] N. Gordon, D. Salmond, and A. Smith. Novel approach to non-linear/non-gaussian bayesian state estimation. *IEEE Processing-F*, 1993.

[54] S. Gratton. On the condition number of linear least squares problems in frobenius norm. *BIT Numerical Mathematics*, 36:523–530, 1995.

[55] S. Gratton, D. Titley-Peloquin, and J. Tshimanga. Sensitivity and conditioning of the truncated total least squares solution. *Society for Industrial and Applied Mathematics, Journal on Matrix Analysis and Applications*, 34(3):1257–1276, 2013.

[56] S. Gratton and J. Tshimanga. An observation-space formulation of variational assimilation using a restricted preconditioned conjugate gradient algorithm. Technical Report, CERFACS, Toulouse, France, 2010.

[57] A. Gronskis, D. Heitz, and E. Mémin. Inflow and initial conditions for direct numerical simulation based on adjoint data assimilation. *Journal of Computational Physics*, pages 480–497, 2013.

[58] T. M. Hamill and C. Snyder. A hybrid ensemble Kalman filter–3D variational analysis scheme. *Monthly Weather Review*, 128(8):2905–2919, 2000.

[59] P. C. Hansen. Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 46:187–194, 2007.

[60] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–436, 1952.

[61] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edition, 2002.

[62] Y. Honda, M. Nishijima, K. Koizumi, Y. Ohta, K. Tamiya, T. Kawabata, , and T. Tsuyuki. A pre-operational variational data assimilation system for a non-hydrostatic model sat the japan meteorological agency: Formulation and preliminary results. *Quarterly Journal of the Royal Meteorological Society*, 2007.

[63] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[64] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[65] P. Houtekamer and H. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126(3):796–811, 1998.

[66] B. R. Hunt. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D*, pages 112–126, 2007.

[67] M. Janisková, J. Thépaut, and J. Geleyn. Simplified and regular physical parametrizations for incremental four-dimensional variational assimilation. *Monthly Weather Review*, 1999.

[68] C. J. Johns and J. Mandel. A two-stage ensemble Kalman filter for smooth data assimilation. *Environmental and Ecological Statistics*, 15:101–110, 2008.

[69] J. Joseph and J. LaViola. A comparison of unscented and extended Kalman filtering for estimating quaternion motion, 2003.

[70] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, pages 182–193, 1997.

[71] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82, 1960.

[72] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Transactions of the ASME. Series D, Journal of Basic Engineering*, 83:95–107, 1961.

[73] E. Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, 2003.

[74] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Fundamentals of Statistical Signal Processing. Prentice-Hall PTR, 1998.

[75] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Number 16 in Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995.

[76] S. P. Khare, J. L. Anderson, T. J. Hoar, and D. Nychka. An investigation into the application of an ensemble Kalman smoother to high-dimensional geophysical systems. *Tellus A*, 60(1):97–112, 2008.

[77] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*, volume 15 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1995. Revised reprint of the 1974 original.

[78] F. Le Gland, V. Monbet, and V. D. Tran. Large sample asymptotics for the ensemble Kalman filter. In D. Crisan and B. Rozovskiĭ, editors, *The Oxford Handbook of Nonlinear Filtering*, pages 598–631. Oxford University Press, 2011.

[79] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[80] J. Lewis, S. Lakshmivarahan, and S. Dhall. *Dynamic Data Assimilation: A Least Squares Approach*. Number vol. 13 in Dynamic data assimilation: a least squares approach. Cambridge University Press, 2006.

[81] C. Liu, Q. Xiao, and B. Wang. An ensemble-based four-dimensional variational data assimilation scheme. Part I: Technical formulation and preliminary test. *Monthly Weather Review*, 136(9):3363–3373, 2008.

[82] C. Liu, Q. Xiao, and B. Wang. An ensemble-based four-dimensional variational data assimilation scheme. Part II: Observing system simulation experiments with Advanced Research WRF (ARW). *Monthly Weather Review*, 137(5):1687–1704, 2009.

[83] A. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 1986.

[84] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[85] J. Mandel. Introduction to infinitely-dimensional statistics, 2013.

[86] J. Mandel, E. Bergou, and S. Gratton. 4DVAR by ensemble Kalman smoother. arxiv, 2013.

[87] J. Mandel, C. Loren, and J. D. Beezley. On the convergence of the ensemble Kalman filter. *Applications of Mathematics*, 56:533–541, 2011.

[88] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Society for Industrial and Applied Mathematics J. Appl. Math.*, 11:431–441, 1963.

[89] C. E. Mehmet and L. Kurz. Robust locally optimal filters: Kalman and Bayesian estimation theory. *Information Sciences*, 92(1-4):1–32, 1996.

[90] R. Ménard and R. Daley. The application of Kalman smoother theory to the estimation of 4DVAR error statistics. *Tellus A*, 48(2):221–237, 1996.

[91] J. J. Moré. The Levenberg-Marquardt algorithm: implementations and theory. In G. A. Watson, editor, *Lecture Notes in Math.*, volume 360, pages 105–116. Springer-Verlag, Berlin, 1977.

[92] D. D. Morrison. Methods for nonlinear least squares problems and convergence proofs. In *Proceedings of the Seminar on Tracking Programs and Orbit Determination*, pages 1–9. Jet Propulsion Laboratory, Pasadena, CA, USA, 1960.

[93] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[94] M. R. Osborne. Nonlinear least squares–the Levenberg algorithm revisited. *Journal of the Australian Mathematical Society*, 19(3):343–357, 1976.

[95] J. Pailleux. A global variational assimilation scheme and its application for using tovs radiances. *In Proc. WMO International Symposium on Assimilation of observations in meteorology and oceanography Clermont-Ferrand France*, page 325–328, 1990.

[96] D. Parish and J. Derber. The national meteorological center's spectral statistical analysis system. *Monthly Weather Review*, 120, 1992.

[97] J. Pedlosky. *Geophysical Fluid Dynamics*. Springer, 1979.

[98] S. Peng and L. Xie. Effect of determining initial conditions by four-dimensional variational data assimilation on storm surge forecasting. *Ocean Model*, 14, 2006.

[99] E. B. Pitman. Power method, 1998.

[100] N. Privé and R. Errico. The role of model and initial condition error in numerical weather forecasting investigated with an observing system simulation experiment. *Tellus A*, 65(0), 2013.

[101] F. Rabier. *The ECMWF Operational Implementation of Four Dimensional Variational Assimilation Part I: Experimental Results with Simplified Physics*. ECMWF technical memorandum. European Centre for Medium-Range Weather Forecasts, 1999.

[102] F. Rabier. Overview of global data assimilation developments in numerical weather-prediction centres. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3215–3233, 2005.

[103] F. Rawlins, S. P. Ballard, K. J. Bovis, A. M. Clayton, D. Li, G. W. Inverarity, A. C. Lorenc, and T. J. Payne. The Met office global four-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 133(623):347–362, 2007.

[104] L. Richardson. *Weather Prediction by Numerical Process*. Cambridge, 1922.

[105] Y. Saad. Overview of Krylov subspace methods with applications to control problems, 1990.

[106] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[107] P. Sakov, D. S. Oliver, and L. Bertino. An iterative EnKF for strongly nonlinear systems. *Monthly Weather Review*, 140(6):1988–2004, 2012.

[108] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* A Wiley-interscience publication. Wiley, 1992.

[109] A. D. Silva, J. Pfaendtner, J. Guo, M. Sienkiewicz, and S. Cohn. Assessing the effects of data selection with dao's physical-space statistical analysis system. In *Assessing the Effects of Data Selection with DAO's Physical-space Statistical Analysis System*, pages 273–278. Monthly Weather Review, 1994.

[110] G. W. Stewart, , and J. Q. Sun. *Matrix Perturbation Theory.* Academic Press, 1990.

[111] G. W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *Society for Industrial and Applied Mathematics*, 15(4):727–764, 1973.

[112] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2004.

[113] Y. Trémolet. Object-oriented prediction system, 1990.

[114] Y. Trémolet. Model-error estimation in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 133(626):1267–1280, 2007.

[115] J. Tshimanga, S. Gratton, A. T. Weaver, and A. Sartenaer. Limited-memory preconditioners, with application to incremental four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 134(632):751–769, 2008.

[116] M. Tüchler, A. C. Singer, and R. Koetter. Minimum mean squared error equalization using a priori information. *IEEE Transactions on Signal Processing*, 50(3):673–683, 2002.

[117] A. W. van der Vaart. *Asymptotic Statistics.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2000.

[118] R. S. Varga. *Matrix Iterative Analysis.* Springer Series in Computational Mathematics. Springer, 2009.

[119] X. Wang. Incorporating ensemble covariance in the gridpoint statistical interpolation variational minimization: A mathematical framework. *Monthly Weather Review*, 138(7):2990–2995, 2010.

[120] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[121] C. K. Wikle and L. M. Berliner. A bayesian tutorial for data assimilation. physica d: Nonlinear phenomena. *Physica D: Nonlinear Phenomena*, pages 1–16, 2012.

[122] C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In *Learning and Inference in Graphical Models*, pages 599–621. Kluwer, 1997.

[123] T. J. Ypma. Historical development of the Newton-Raphson method. *Society for Industrial and Applied Mathematics*, 37(4):531–551, 1995.

[124] T. Zhang, A. Popescul, and B. Dom. Linear prediction models with graph regularization for web-page categorization. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD 06, pages 821–826, New York, NY, USA, 2006.

[125] X. Zou, I. M. Navon, M. Berger, K. H. Phua, T. Schlickii, and F. X. L. Dimet. Numerical experience with limited-memory quasi-newton methods and truncated newton methods. *Society for Industrial and Applied Mathematics, Journal on Optimization*, 1992.

[126] M. Zupanski. Maximum likelihood ensemble filter: Theoretical aspects. *Monthly Weather Review*, 133(6):1710–1726, 2005.