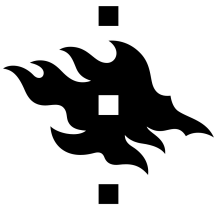
University of Helsinki
Faculty of Science
Department of Mathematics and Statistics



Master's thesis

Data assimilation using the Ensemble Adjustment Kalman filter with application to soil organic carbon modelling

Maisa Laine

Advisors: Toni Viskari and Eva Kisdi 2019

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Matematiikan ja tilastotieteen laitos

Tekijä — Författare — Author
Maisa Laine

Työn nimi — Arbetets titel — Title
Data assimilation using the Ensemble Adjustment Kalman filter with application to soil organic carbon modelling

Oppiaine — Läroämne — Subject		
Matematiikka		
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu -tutkielma	Toukokuu 2019	29 s.

Tiivistelmä — Referat — Abstract

Data assimilaatio on estimointi menetelmä, jolla voidaan yhdistää informaatiota useista eri lähteistä. Data assimilaatio menetelmien hyödyllisyys näkyy erityisesti kun yhdistetään epäsuoria havaintoja mallin tilaan.

Tässä tutkielmassa keskitytään sekventiaalisiin data assimilaatio menetelmiin, jotka pohjautuvat Kalman filter -menetelmään. Kalman filter -menetelmä johdetaan Bayesin kaavasta ja sen pohjalta esitellään ensemble-menetelmiä, jotka usein ovat laskennallisesti kevyempiä approksimaatiota Kalman filter -menetelmästä.

Tutkielmassa sovelletaan Ensemble Adjustment Kalman filter -menetelmään orgaanisen maahiilen hajoamista kuvaavaan Yasso-malliin. Yasson avulla mallinnetaan pitkäaikaista maahiiltä kuudelta eri pellolta. Ennusteita parannetaan data assimilaation avulla yhdistämällä ennusteeseen mittauksista saatu informaatio.

Avainsanat — Nyckelord — Keywords
Data assimilaatio, Kalman filter, Ensemble Adjustment Kalman filter

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpulan tiedekirjasto

Muita tietoja — Övriga uppgifter — Additional information

Contents

Notation

Many different notations have been used in literature with regard to data assimilation. In this thesis, we follow the notation proposed in [12], with only minor differences in order to make the notation as easy to follow as possible for the purposes of this thesis.

Symbols

$\boldsymbol{x}$	State vector
$\boldsymbol{y}$	Observation vector
$\boldsymbol{z}$	Joint state-observation vector
$\mathbf{M}$	Linear or tangent linear model operator
$\mathcal{M}$	Non-linear model operator
$\mathbf{H}$	Linear or tangent linear observation operator
$\mathcal{H}$	Non-linear observation operator
$\boldsymbol{\eta}$	Model error
$\boldsymbol{\epsilon}$	Observational error
$\boldsymbol{e}$	Estimation error
$\mathbf{Q}$	Model error covariance
$\mathbf{R}$	Observation error covariance
$\mathbf{P}$	Filter error covariance
$\mathbf{K}$	Gain matrix

Indices

$()^a$	Analysis
$()^f$	Forecast
$()_k$	At time t_k
$()_e$	Ensemble

Chapter 1

Introduction

Data assimilation is an estimation method for combining information of a system from multiple sources. Data assimilation can be used in a wide range of applications for several purposes ranging from interpolating sparse data to estimating model parameters. We focus on data assimilation with a goal in state estimation. Especially we concentrate in sequential data assimilation, which means that the previous state is used as prior information when new observations become available, as opposed to methods that use information from the whole time window. Common optimisation methods for calculating the estimate are to either minimise a cost function or to find a minimal variance solution for the problem. Similar results can be obtained using either one and both have their advantages. In this thesis we will discuss only the minimal variance optimisation.

In chapter 2 we will show how data assimilation is built on the Bayes' theorem, which provides a way of combining prior information of the system with observations, in order to calculate a new estimate, and builds a probability framework for it. In chapter 3 we will introduce the Kalman filter method, which is one of the most important statistical tools developed in the 20th century. Kalman filter is the basis for several different ensemble-based methods of which we introduce two in chapter 4. The first is the traditional Ensemble Kalman filter and the second, Ensemble Adjustment Kalman Filter, was developed based on the first. In our research problem we used the second filter.

After the theoretical part we will introduce the Yasso model for the decomposition of soil organic carbon. It is a simple linear model that requires an initial state and yearly weather conditions as input. We used observations of the amount of soil organic carbon in five fields across Europe measured as part of a long-term experiment [5] done in different climate conditions and soil types. In section 5.3 we will go through how the data was processed and introduce the DART software, which was used to run the Ensemble Adjustment Kalman filter algorithm. Finally the results are presented in chapter 6.

Chapter 2

Bayesian approach for data assimilation

Data assimilation is based on the Bayes' theorem. We start by looking at the Bayes' formula, which estimates a state x based on observations y and some previous information of the state

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)},$$

where the probability $p(x|y)$ is the posterior distribution for the state x given an observation y. Whereas the probability $p(x)$ is the prior distribution, which holds all the information we have about the state x before the assimilation. The probability $p(y|x)$ is the likelihood for the observation y, if the true state was x. The probability for the observation $p(y)$ works as a normalisation coefficient, $p(y) = \int p(y|x)p(x)\,dx$, which makes sure the above equation is indeed a probability distribution. In the following calculations we can drop the probability $p(y)$ since it doesn't depend on the state x, in which we are interested, and can thus be treated as a constant.

We will take a closer look at the Bayes' theorem in a simple scalar case and derive a basis for data assimilation following the deduction from [20]. We are interested in an unknown true state. The state x represents the true state but since we can not be certain what the true state is, we associate an uncertainty for x by assuming it to be a normally distributed scalar variable $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$, with mean μ_x and variance σ_x^2. By assuming the state to be a Gaussian variable we get a well known probability density function as our prior distribution $p(x)$. The probability for the observations $\boldsymbol{y} = (y_1, \ldots, y_n)$ conditioned on a realisation x of the state is

$$y_i|x \sim \mathcal{N}(x, \sigma^2),$$

with mean x and observational error variance σ^2. This is the likelihood distribution and

since it is also Gaussian we know its probability density function is

$$\begin{aligned} p(\boldsymbol{y}|x) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - x)^2\right\} \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x)^2\right\}. \end{aligned}$$

As mentioned above, the probability $p(\boldsymbol{y})$ doesn't depend on the state so we can be drop it from the Bayes' theorem. The posterior distribution can now be written as a product of the likelihood and the prior distribution

$$\begin{aligned} p(x|\boldsymbol{y}) &\propto p(\boldsymbol{y}|x)p(x) \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x)^2\right\} \exp\left\{-\frac{1}{2\sigma_x^2}(x - \mu_x)^2\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left(\sum_{i=1}^{n}\frac{(y_i - x)^2}{\sigma^2} + \frac{(x - \mu_x)^2}{\sigma_x^2}\right)\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left(\sum_{i=1}^{n}\frac{y_i^2 - 2xy_i + x^2}{\sigma^2} + \frac{x^2 - 2x\mu_x + \mu_x^2}{\sigma_x^2}\right)\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left[x^2\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right) - 2x\left(\sum_{i=1}^{n}\frac{y_i}{\sigma^2} + \frac{\mu_x}{\sigma_x^2}\right) + \sum_{i=1}^{n}\frac{y_i^2}{\sigma^2} + \frac{\mu_x^2}{\sigma_x^2}\right]\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right)\left[x^2 - 2x\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right)^{-1}\left(\sum_{i=1}^{n}\frac{y_i}{\sigma^2} + \frac{\mu_x}{\sigma_x^2}\right)\right.\right. \\ &\qquad \left.\left. + \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right)^{-1}\left(\sum_{i=1}^{n}\frac{y_i^2}{\sigma^2} + \frac{\mu_x^2}{\sigma_x^2}\right)\right]\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right)\left[x - \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right)^{-1}\left(\sum_{i=1}^{n}\frac{y_i}{\sigma^2} + \frac{\mu_x}{\sigma_x^2}\right)\right]^2\right\}, \end{aligned}$$

where the last term in the second to last formula doesn't depend on x so we can modify is to get the required form and the proportionality still holds.

Now the posterior is a Gaussian $x|\boldsymbol{y} \sim \mathcal{N}(\mu_{x|\boldsymbol{y}}, \sigma^2_{x|\boldsymbol{y}})$, where

$$\mu_{x|\boldsymbol{y}} = \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right)^{-1} \left(\sum_{i=1}^{n} \frac{y_i}{\sigma^2} + \frac{\mu_x}{\sigma_x^2}\right) \quad \text{and}$$
$$\sigma^2_{x|\boldsymbol{y}} = \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_x^2}\right)^{-1}.$$

The expected value for $x|\boldsymbol{y}$ is the mean $\mu_{x|\boldsymbol{y}}$, which can also be written as

$$\begin{aligned} \mathrm{E}(x|\boldsymbol{y}) &= \frac{\sigma^2\sigma_x^2}{\sigma^2 + n\sigma_x^2}\left(\frac{n}{\sigma^2}\bar{\boldsymbol{y}} + \frac{1}{\sigma_x^2}\mu_x\right) \\ &= w_{\boldsymbol{y}}\bar{\boldsymbol{y}} + w_{\mu_x}\mu_x. \end{aligned}$$

Here $\bar{\boldsymbol{y}} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is mean over the observations and $w_{\boldsymbol{y}} + w_{\mu_x} = 1$, when

$$w_{\boldsymbol{y}} = \frac{n\sigma_x^2}{\sigma^2 + n\sigma_x^2} \qquad \text{and} \qquad w_{\mu_x} = \frac{\sigma^2}{\sigma^2 + n\sigma_x^2}.$$

The posterior mean is the weighted average of the data mean and the prior mean. With some simple manipulation the posterior mean can also be written as

$$\begin{aligned} \mathrm{E}(x|\boldsymbol{y}) &= \mu_x + \frac{n\sigma_x^2}{\sigma^2 + n\sigma_x^2}(\bar{\boldsymbol{y}} - \mu_x) \\ &= \mu_x + K(\bar{\boldsymbol{y}} - \mu_x), \end{aligned} \tag{2.1}$$

in which the prior mean is adjusted towards the data mean according to a gain K. This is the basis for the Kalman filter method introduced in the next chapter.

Chapter 3

Kalman filter method

In this chapter we will first go through the Kalman filter method, which was introduced by *Rudolf Kalman* in 1960 in his paper *A New Approach to Linear Filtering and Prediction Problems* [14]. The other methods discussed in this thesis are based on Kalman filter so to understand them it is important to first understand how the Kalman filter works.

The Kalman filter is an elegant and simple method for estimating and updating the state variables as they move in time while giving a measure on the uncertainty of the estimates [11]. However the Kalman filter method is optimal, in a minimum variance sense, only when the operators are linear and the statistics Gaussian.

The model and observation operators are not always linear and this has motivated people to develop methods to extend it into more complicated situations. Later in this chapter we will introduce the extended Kalman filter which mimics the KF in a non-linear case and in the next chapter we will look at ensemble methods which offer ways to avoid the problems with large dimensions and non-linearity.

3.1 Kalman filter

Let the sequence of unknown true state vectors be $\boldsymbol{x}_k \in \mathbb{R}^m$, with a time index $k = 1, 2, \ldots$ denoting time t_k of an observation and let the observation vectors be $\boldsymbol{y}_k \in \mathbb{R}^n$. The evolution of the system is modelled iteratively with two equations

$$\boldsymbol{x}_k = \mathbf{M}_k \boldsymbol{x}_{k-1} + \boldsymbol{\eta}_k \quad \text{and} \tag{3.1}$$

$$\boldsymbol{y}_k = \mathbf{H}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k. \tag{3.2}$$

The linear model operator $\mathbf{M}_k \in \mathbb{R}^{m \times m}$ holds the information about how the model moves the state forward in time. Likewise the linear observation operator $\mathbf{H}_k \in \mathbb{R}^{n \times m}$

describes the measurement method and is a function from the model space to the observational space. The errors in the model, $\boldsymbol{\eta}_k \in \mathbb{R}^m$, and the observations, $\boldsymbol{\epsilon}_k \in \mathbb{R}^n$, are assumed to be independent white noise processes i.e. Gaussian with zero mean and covariance matrices $\mathbf{Q}_k \in \mathbb{R}^{m\times m}$ and $\mathbf{R}_k \in \mathbb{R}^{n\times n}$ respectively. This assumption is necessary for the optimal KF method but in practical problems it often doesn't hold.

The Kalman filter works in two steps; forecast and analysis step. The forecast step moves the state forward in time with the model operator

$$\boldsymbol{x}_k^f = \mathbf{M}_k \boldsymbol{x}_{k-1}^a, \tag{3.3}$$

and the analysis step updates the state given by the forecast step with the information from the observations

$$\boldsymbol{x}_k^a = \boldsymbol{x}_k^f + \mathbf{K}_k(\boldsymbol{y}_k - \mathbf{H}_k \boldsymbol{x}_k^f), \tag{3.4}$$

where the difference $\boldsymbol{y} - \mathbf{H}\boldsymbol{x}^f$, referred to as the innovation, describes the difference between the measurements and the projection of the state in the observational space.

Note that the forecast and analysis states are equivalent to the prior and posterior equations derived in the previous chapter and are often referred to as such. In both of these steps an error covariance matrix is computed to estimate the uncertainty of the estimate. The Kalman gain $\mathbf{K}$ is chosen to minimise the trace of the analysis error covariance matrix $\mathbf{P}^a$ defined below.

Theorem 3.1.1. *When the forecast and analysis states, $\boldsymbol{x}_k^f$ and $\boldsymbol{x}_k^a$, are defined as in equations* (3.3) *and* (3.4)*, the corresponding error covariances are given by*

$$\mathbf{P}_k^f = \mathbf{M}_k \mathbf{P}_{k-1}^a \mathbf{M}_k^T + \mathbf{Q}_k \quad \textit{and} \quad \mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^f,$$

and the optimal Kalman gain matrix is

$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k)^{-1}.$$

Proof. We follow the reasoning from [3] and start by writing the estimation errors for both steps as

$$\boldsymbol{e}_k^f = \boldsymbol{x}_k^f - \boldsymbol{x}_k \quad \text{and} \quad \boldsymbol{e}_k^a = \boldsymbol{x}_k^a - \boldsymbol{x}_k,$$

where $\boldsymbol{x}_k$ is the true state. Then by definition the error covariance matrices for the forecast and analysis states are

$$\mathbf{P}_k^f = \operatorname{cov}(\boldsymbol{e}_k^f) = \mathrm{E}(\boldsymbol{e}_k^f (\boldsymbol{e}_k^f)^T) \quad \text{and} \quad \mathbf{P}_k^a = \operatorname{cov}(\boldsymbol{e}_k^a) = \mathrm{E}(\boldsymbol{e}_k^a (\boldsymbol{e}_k^a)^T). \tag{3.5}$$

Let us look at the forecast error covariance first, in this proof we drop the time index k from the model and observational operators for simplicity but they can be added back without any change to the reasoning. When we substitute the forecast state $\boldsymbol{x}_k^f$ from equation (3.3) and the true state $\boldsymbol{x}_k$ from equation (3.1) into the above equation we get the desired form

$$\begin{aligned}
\mathbf{P}_k^f &= \mathrm{E}\left[\big(\mathbf{M}\boldsymbol{x}_{k-1}^a - \mathbf{M}\boldsymbol{x}_{k-1} + \boldsymbol{\eta}_k\big)\big(\mathbf{M}\boldsymbol{x}_{k-1}^a - \mathbf{M}\boldsymbol{x}_{k-1} + \boldsymbol{\eta}_k\big)^T\right] \\
&= \mathrm{E}\left[\big(\mathbf{M}(\boldsymbol{x}_{k-1}^a - \boldsymbol{x}_{k-1}) + \boldsymbol{\eta}_k\big)\big(\mathbf{M}(\boldsymbol{x}_{k-1}^a - \boldsymbol{x}_{k-1}) + \boldsymbol{\eta}_k\big)^T\right] \\
&= \mathrm{E}\Big[\mathbf{M}(\boldsymbol{x}_{k-1}^a - \boldsymbol{x}_{k-1})(\boldsymbol{x}_{k-1}^a - \boldsymbol{x}_{k-1})^T\mathbf{M}^T + \mathbf{M}(\boldsymbol{x}_{k-1}^a - \boldsymbol{x}_{k-1})\boldsymbol{\eta}_k^T \\
&\qquad + \boldsymbol{\eta}_k(\boldsymbol{x}_{k-1}^a - \boldsymbol{x}_{k-1})^T\mathbf{M}^T + \boldsymbol{\eta}_k\boldsymbol{\eta}_k^T\Big] \\
&= \mathbf{M}\mathbf{P}_{k-1}^a\mathbf{M}^T + \mathbf{Q}_k.
\end{aligned}$$

The last equality follows from the fact that the errors $\boldsymbol{\eta}_k$ are assumed to be zero mean Gaussian with covariance matrices $\mathbf{Q}_k$. We also needed to assume that the analysis errors $\boldsymbol{e}_{k-1}^a$ are independent of $\boldsymbol{\eta}_k$ for every $k = 1, 2, \ldots$.

Next we wish to find the analysis error covariance. We begin by taking a look at the analysis state (3.4), which can be written as

$$\boldsymbol{x}_k^a = \boldsymbol{x}_k^f + \mathbf{K}_k(\mathbf{H}\boldsymbol{x}_k + \boldsymbol{\epsilon}_k - \mathbf{H}\boldsymbol{x}_k^f),$$

when we substitute the observation vector from equation (3.2) into it. We use this form of the analysis state in the equation for the analysis estimation error and get

$$\begin{aligned}
\boldsymbol{e}_k^a &= \boldsymbol{x}_k^f + \mathbf{K}_k(\mathbf{H}\boldsymbol{x}_k - \mathbf{H}\boldsymbol{x}_k^f + \boldsymbol{\epsilon}_k) - \boldsymbol{x}_k \\
&= \mathbf{K}_k(-\mathbf{H}(\boldsymbol{x}_k^f - \boldsymbol{x}_k) + \boldsymbol{\epsilon}_k) + (\boldsymbol{x}_k^f - \boldsymbol{x}_k) \\
&= \mathbf{K}_k(\boldsymbol{\epsilon}_k - \mathbf{H}\boldsymbol{e}_k^f) + \boldsymbol{e}_k^f.
\end{aligned}$$

Now we can use the above equality and substitute it into equation (3.5) for the analysis error covariance

$$\begin{aligned}
\mathbf{P}_k^a &= \mathrm{E}\left[(\mathbf{K}_k(\boldsymbol{\epsilon}_k - \mathbf{H}\boldsymbol{e}_k^f) + \boldsymbol{e}_k^f)(\mathbf{K}_k(\boldsymbol{\epsilon}_k - \mathbf{H}\boldsymbol{e}_k^f) + \boldsymbol{e}_k^f)^T\right] \\
&= \mathrm{E}\left[\boldsymbol{e}_k^f(\boldsymbol{e}_k^f)^T - (\mathbf{K}_k\mathbf{H}\boldsymbol{e}_k^f)(\mathbf{K}_k\mathbf{H}\boldsymbol{e}_k^f)^T + \mathbf{K}_k\boldsymbol{\epsilon}_k(\mathbf{K}_k\boldsymbol{\epsilon}_k)^T\right] \\
&= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^f(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T, \qquad (3.6)
\end{aligned}$$

where the second equality follows from the assumption that the estimate errors are unbiased and so have expected value of zero.

For the final part of the proof we need some properties from matrix differential calculus [16]. For any matrices $\mathbf{B}$ and $\mathbf{C}$ of appropriate sizes it holds that

$$\frac{\partial}{\partial \mathbf{C}} \operatorname{Tr}(\mathbf{C}^T\mathbf{B}) = \mathbf{B} \quad \text{and} \quad \frac{\partial}{\partial \mathbf{C}} \operatorname{Tr}(\mathbf{C}\mathbf{B}\mathbf{C}^T) = \mathbf{C}\mathbf{B}^T + \mathbf{C}\mathbf{B},$$

where the partial derivative with respect to a matrix is defined elementwise.

Lastly we wish to find the optimal Kalman gain $\mathbf{K}$ which minimises the trace of the analysis error covariance $\mathbf{P}^a$. The reason we wish to minimise the trace is because it contains the estimation error variances for the components of the state and we are looking for the minimal variance solution. To do this we will take the derivative of the trace of the analysis error covariance with respect to the gain

$$\begin{aligned}\frac{\partial}{\partial \mathbf{K}_k} \operatorname{Tr}(\mathbf{P}_k^a) &= \frac{\partial}{\partial \mathbf{K}_k} \operatorname{Tr}\left(\mathbf{P}_k^f - \mathbf{K}_k\mathbf{H}\mathbf{P}_k^f - \mathbf{P}_k^f\mathbf{H}^T\mathbf{K}_k^T + \mathbf{K}_k(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})\mathbf{K}^T\right) \\ &= -2(\mathbf{H}\mathbf{P}_k^f)^T + 2\mathbf{K}_k(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R}),\end{aligned}$$

where we used the fact that $\mathbf{P}^f$ is symmetric. From this we get the optimal Kalman gain matrix by setting the above result to zero

$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^f\mathbf{H}^T + \mathbf{R})^{-1},$$

which we then substitute into the equation (3.6) for the analysis error covariance and get the desired form

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^f.$$

□

The error covariance matrices $\mathbf{P}^f$ and $\mathbf{R}$ give us a way of deciding how much trust we put on the forecast state and observations. The balance between trusting the observations versus the forecast is determined by the Kalman gain. If $\mathbf{R}$ is large we consider the observations to be untrustworthy and the new analysis state will stay close to the forecast state. Also $\mathbf{P}^a$ will be close to $\mathbf{P}^f$. On the other hand, if $\mathbf{R}$ is small we consider the observation to be good and the analysis state will be close to the projection of the observation to the model space. In this case $\mathbf{P}^a$ will be clearly smaller than $\mathbf{P}^f$.

However, $\mathbf{P}^f$ affects the gain in the opposite direction. If $\mathbf{P}^f$ is small we we trust the estimate a lot and the observations need to be really good to have a considerable effect on the analysis state. And similarly is $\mathbf{P}^f$ is large we rely on the observations even if they aren't that good.

We talked about covariance matrices as being large or small since it it easier to imagine all the variances increasing or decreasing. However it should be noted that we have a variance for each element of the state and observation vector and it is their relative size to the corresponding element that determines how we update that particular element.

3.2 Extended Kalman filter

As stated before, the Kalman filter is optimal only when the operators are linear and the statistics Gaussian. The need for the linear operators is based on the fact that we need linear algebra to work with the covariance matrices, linearity also preserves the Gaussianity. The Extended Kalman filter method, EKF, can be used if the operators are non-linear. [4] In EKF the true state, the observation vector, the forecast and analysis state are defined using the non-linear operators

$$\begin{aligned}
\boldsymbol{x}_k &= \mathcal{M}_k \boldsymbol{x}_{k-1} + \boldsymbol{\eta}_k, \\
\boldsymbol{y}_k &= \mathcal{H}_k \boldsymbol{x}_k + \boldsymbol{\epsilon}_k, \\
\boldsymbol{x}_k^f &= \mathcal{M}_k \boldsymbol{x}_{k-1}^a \qquad \text{and} \\
\boldsymbol{x}_k^a &= \boldsymbol{x}_k^f + \mathbf{K}_k(\boldsymbol{y}_k - \mathcal{H}_k \boldsymbol{x}_k^f).
\end{aligned}$$

However in the covariance matrices and the Kalman gain the model and observational operators need to be linear. This is achieved by replacing the non-linear operators with their tangent linear approximations i.e. taking the first order derivative of the operators with respect to the state

$$\mathbf{M}_k = \frac{\partial \mathcal{M}_k \boldsymbol{x}_{k-1}^a}{\partial \boldsymbol{x}} \qquad \text{and} \qquad \mathbf{H}_k = \frac{\partial \mathcal{H}_k \boldsymbol{x}_k^f}{\partial \boldsymbol{x}}.$$

The EKF works only for reasonably small breaches of linearity. For example, if the time step is long enough for the model's non-linearity to show, the forecast error covariance may be underestimated and this can cause filter divergence, which will be discussed in section 4.2. [3]

Chapter 4

Data assimilation using ensemble methods

In this chapter we introduce two ensemble methods based on the Kalman filter. They can be used when the system is, for instance, computationally too large for the traditional Kalman filter method to work. The advantage of these methods is that we avoid having to store and develop the full covariance matrices by replacing the forecast error covariance matrix with sample covariance at each time step.

Ensemble methods are widely used and the most common ones have software platforms accessible to anyone. This was one of the reasons we decided to use an ensemble method. Although our model is simple enough for the EKF method to work as well. Another reason for using an ensemble method was that it would have been extremely hard to estimate the initial error covariance for our model.

In all ensemble-based method an initial ensemble needs to be created. This is done by adding perturbations to a best-guess estimate of the state based on its uncertainty. Luckily small differences in the perturbations done to create the initial ensemble don't affect the results much over a long enough time period. [8] Over time, when we've run several iterations, the covariances develop and become more accurate than after the first iteration.

4.1 Ensemble Kalman filter

The Ensemble Kalman filter, EnKF, was first introduced in 1994 by *Geir Evensen* [7] and has been modified several times for example in 1998 by *Burgers et al.* [6] and *Evensen* in 2003 [8]. The EnKF method has been applied to a large range of problems for example in ocean and atmospheric science applications and is still widely used in many fields.

We start by initialising an ensemble of states $\{\boldsymbol{x}_i^f\}_{i=1,\dots,d}$, for which we have a mean and covariance by definition

$$\overline{\boldsymbol{x}}^f := \frac{1}{d}\sum_{i=1}^{d} \boldsymbol{x}_i^f \qquad \text{and} \qquad \mathbf{P}_e^f := \mathrm{E}[(\boldsymbol{x}^f - \overline{\boldsymbol{x}}^f)(\boldsymbol{x}^f - \overline{\boldsymbol{x}}^f)^T]. \tag{4.1}$$

For the EnKF method to hold the observations need to be treated as random variables. Whenever observations become available an ensemble of observations $\boldsymbol{y}_i = \boldsymbol{y} + \epsilon_i$ is drawn, where $\epsilon_i \sim \mathcal{N}(0, \mathbf{R})$. The ensemble observational error covariance matrix is $\mathbf{R}_e = \mathrm{E}(\epsilon\epsilon^T)$.

The ensemble Kalman gain matrix is defined similarly as before but using the ensemble statistics

$$\mathbf{K}_e = \mathbf{P}_e^f \mathbf{H}^T (\mathbf{H}\mathbf{P}_e^f \mathbf{H}^T + \mathbf{R}_e)^{-1}.$$

Each of the ensemble members is updated separately according to the Kalman filter analysis step, where the observational operator can be non-linear,

$$\boldsymbol{x}_i^a = \boldsymbol{x}_i^f + \mathbf{K}_e(\boldsymbol{y}_i - \mathcal{H}_k \boldsymbol{x}_i^f),$$

from which we can compute the ensemble mean

$$\overline{\boldsymbol{x}^a} = \overline{\boldsymbol{x}}^f + \mathbf{K}_e(\overline{\boldsymbol{y}} - \mathcal{H}_k \overline{\boldsymbol{x}}^f),$$

this is how the mean was defined in the method though it is only valid when $\mathcal{H}_k$ is not too strongly non-linear. The analysis state and its mean are then used to compute analysis the error covariance matrix

$$\begin{aligned}
\mathbf{P}_e^a &= \mathrm{E}[(\boldsymbol{x}^a - \overline{\boldsymbol{x}}^a)(\boldsymbol{x}^a - \overline{\boldsymbol{x}}^a)^T] \\
&= (I - \mathbf{K}_e\mathbf{H})\mathbf{P}_e^f(I - \mathbf{H}^T\mathbf{K}_e^T) + \mathbf{K}_e\mathbf{R}_e\mathbf{K}_e^T \\
&= \mathbf{P}_e^f - \mathbf{K}_e\mathbf{H}\mathbf{P}_e^f - \mathbf{P}_e^f\mathbf{H}^T\mathbf{K}_e^T + \mathbf{K}_e(\mathbf{H}\mathbf{P}_e^f\mathbf{H}^T + \mathbf{R}_e)\mathbf{K}_e^T \\
&= (I - \mathbf{K}_e\mathbf{H})\mathbf{P}_e^f,
\end{aligned}$$

which is now defined analogous to the Kalman filter method but with the ensemble statistics. Lastly, the next forecast is computed from the obtained analysis states

$$\boldsymbol{x}_i^f = \mathcal{M}_k \boldsymbol{x}_i^a,$$

and the forecast statistics are calculated as in equations (4.1) using the new ensemble states.

4.2 Filter divergence

Filter divergence is a phenomenon that may happen if some of the errors are not included in the error covariances. When that happens the forecast error covariance gets repeatedly underestimated and the filter starts to trust its own estimates more and more and the observations have less and less effect on the estimate. Different error sources that are hard to incorporate into the covariances are, for example, the model and sampling error or errors from linear approximations.

Let us look at the error covariance matrices in the traditional Kalman filter.

$$\mathbf{P}^f = \mathbf{M}\mathbf{P}^a\mathbf{M}^T + \mathbf{Q} \quad \leftrightarrow \quad \mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f$$

The point of the analysis is to improve the forecast and it always gives smaller or equal uncertainty, i.e. $\mathbf{P}^a \leq \mathbf{P}^f$. This holds because $\mathbf{KH}$ can only have values between zero and one. The uncertainty for $\mathbf{P}^f$ is predicted from the previous analysis uncertainty using $\mathbf{M}$ with additional model error $\mathbf{Q}$. The effect $\mathbf{M}$ has on $\mathbf{P}^a$ depends on the application but it rarely increases the covariance enough on its own to obtain balance between $\mathbf{P}^f$ and $\mathbf{P}^a$ over time. Hence if $\mathbf{Q}$ is underestimated the filter diverges. However in practice estimating $\mathbf{Q}$ is hard. The model error consists of uncertainties in the model structure, inputs and parameters and there is no way of knowing the error caused by the model structure exactly.

In the ensemble methods $\mathbf{P}^f$ is calculated from the sample variance. The effect of the model on slightly different states within the ensemble is used to estimate the model uncertainties. This method still often underestimates the error covariances and we are faced with filter divergence. This is again due to the fact that we are unable to estimate the model error. Next we discuss some methods to handle filter divergence.

Randomising the observations in the EnKF method is one way of reducing filter divergence since it introduces an extra source of error to the covariance matrices. EnKF was first introduced without the perturbations, but later the method was corrected in [6] because without randomising the observations the analysis error covariance matrix would be underestimated and it wouldn't be equivalent to the Kalman filer theory. This is due to the observational error covariance $\mathbf{R}$ not having a counterpart in the ensemble method without perturbing the observations, all the states would be updated using the same measurement and this would cause correlations between them. [6]

One way to counteract filter divergence is to inflate the filter error covariance matrix $\mathbf{P}$ by multiplying it with a constant factor slightly greater than 1 before or after analysis. Inflation methods can also be additive or applied to the ensemble members instead of the covariances. There are also several different adaptive inflation methods where the factor is a variable. [3] In our application we will apply a simple constant inflation factor to the forecast error covariance. Inflation is quite an artificial method for reducing filter

divergence and there is no exact way of choosing the inflation factor. Nevertheless it is an effective and transparent way of controlling filter divergence, which makes it widely used.

4.3 Ensemble Adjustment Kalman filter

The Ensemble Adjustment Kalman filter, EAKF, was first introduced by Jeffrey Anderson in 2001 [1] as an alternative to EnKF. Both of these ensemble methods preserve information about the structure and higher-order moments of the prior ensemble. In EAKF this is done in the analysis step with a matrix $\mathbf{A}$, which we will talk about at the end of this section.

In this method we define a joint state-observation vector $\boldsymbol{z} \in \mathbb{R}^{m+n}$ as

$$\boldsymbol{z} = (\boldsymbol{x}, \mathcal{H}\boldsymbol{x}),$$

which contains the state and the projection of the state to the observational space. One advantage of this method is that there are no restrictions on $\mathcal{H}$. Same holds for the EnKF even though it isn't as transparent. The projection to the observational space can be obtained from the joint state vector with a linear operator $\mathbf{H} \in \mathbb{R}^{n\times(m+n)}$ which is defined as $\mathbf{H}[j, j+m] = 1, j = 1, \ldots, n$, and zero everywhere else. Now it holds that $\mathbf{H}\boldsymbol{z} = \mathcal{H}\boldsymbol{x}$. This linear $\mathbf{H}$ is used when calculating the mean and covariance for the joint state vector.

First we will draw an ensemble of joint state vectors $\boldsymbol{z}_e^f$ as our initial forecast ensemble. The initial ensemble is drawn from a Gaussian distribution with mean equal to our best-guess estimate for the state and variance determined based on its uncertainty. From this ensemble we get the sample mean $\overline{\boldsymbol{z}}_e^f$ and covariance $\mathbf{P}_e^f$. The forecast ensemble should be approximately Gaussian and can be represented by the calculated mean and variance, $\boldsymbol{z}^f \sim \mathcal{N}(\overline{\boldsymbol{z}}_e^f, \mathbf{P}_e^f)$.

Now that we have the forecast ensemble we want to update it based on information from the observations. The covariance and mean for the probability $p(\boldsymbol{z}^a) = p(\boldsymbol{z}^f|\boldsymbol{y})$ are given by

$$\mathbf{P}_e^a = [(\mathbf{P}_e^f)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]^{-1} \tag{4.2}$$

and

$$\overline{\boldsymbol{z}}_e^a = \mathbf{P}_e^a[(\mathbf{P}_e^f)^{-1}\overline{\boldsymbol{z}}_e^f + \mathbf{H}^T\mathbf{R}^{-1}\boldsymbol{y}]. \tag{4.3}$$

These can be derived from the Bayes' theorem in the same way we did in chapter 2. The likelihood for the observations given the joint state vector is $p(\boldsymbol{y}|\boldsymbol{z}^f) \sim \mathcal{N}(\mathbf{H}\boldsymbol{z}^f, \mathbf{R})$. We can now calculate conditional probability $p(\boldsymbol{z}^f|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{z}^f)p(\boldsymbol{z}^f) \propto \exp\left\{-\frac{1}{2}\mathbf{U}\right\}$, where

$$\begin{aligned}\mathbf{U} &= (\boldsymbol{y} - \mathbf{H}\boldsymbol{z}^f)^T\mathbf{R}^{-1}(\boldsymbol{y} - \mathbf{H}\boldsymbol{z}^f) + (\boldsymbol{z}^f - \overline{\boldsymbol{z}}_e^f)^T(\mathbf{P}_e^f)^{-1}(\boldsymbol{z}^f - \overline{\boldsymbol{z}}_e^f)\\ &= \boldsymbol{y}^T\mathbf{R}^{-1}\boldsymbol{y} - \boldsymbol{y}^T\mathbf{R}^{-1}\mathbf{H}\boldsymbol{z}^f - (\boldsymbol{z}^f)^T\mathbf{H}^T\mathbf{R}^{-1}\boldsymbol{y} + (\boldsymbol{z}^f)^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\boldsymbol{z}^f\\ &\quad + (\boldsymbol{z}^f)^T(\mathbf{P}_e^f)^{-1}\boldsymbol{z}^f - (\boldsymbol{z}^f)^T(\mathbf{P}_e^f)^{-1}\overline{\boldsymbol{z}}_e^f - \overline{\boldsymbol{z}}_e^T(\mathbf{P}_e^f)^{-1}\boldsymbol{z}^f + (\overline{\boldsymbol{z}}_e^f)^T(\mathbf{P}_e^f)^{-1}\overline{\boldsymbol{z}}_e^f.\end{aligned}$$

Next we gather the terms with $\boldsymbol{z}^f$ and denote $\boldsymbol{v} = (\mathbf{P}_e^f)^{-1}\overline{\boldsymbol{z}}_e^f + \mathbf{H}^T\mathbf{R}^{-1}\boldsymbol{y}$ so that we can write $\mathbf{U}$ as

$$\begin{aligned}\mathbf{U} &= (\boldsymbol{z}^f)^T(\mathbf{P}^a)^{-1}\boldsymbol{z}^f - \boldsymbol{v}^T\boldsymbol{z}^f - (\boldsymbol{z}^f)^T\boldsymbol{v} + \{\boldsymbol{y}^T\mathbf{R}^{-1}\boldsymbol{y} + (\overline{\boldsymbol{z}}_e^f)^T(\mathbf{P}_e^f)^{-1}\overline{\boldsymbol{z}}_e^f\} \\ &= (\boldsymbol{z}^f - \overline{\boldsymbol{z}}_e^a)^T(\mathbf{P}^a)^{-1}(\boldsymbol{z}^f - \overline{\boldsymbol{z}}_e^a) + \{\boldsymbol{y}^T\mathbf{R}^{-1}\boldsymbol{y} + (\overline{\boldsymbol{z}}_e^f)^T(\mathbf{P}_e^f)^{-1}\overline{\boldsymbol{z}}_e^f - \boldsymbol{v}^T\mathbf{P}^a\boldsymbol{v}\},\end{aligned}$$

where the terms in the curly brackets do not contain $\boldsymbol{z}^f$ and can thus be treated as constants. From this form it is easy to see that equations (4.2) and (4.3) hold.

To get from the forecast ensemble to the analysis ensemble with the above statistics EAKF uses the following equation

$$\boldsymbol{z}_i^a = \mathbf{A}^T(\boldsymbol{z}_i^f - \overline{\boldsymbol{z}}_e^f) + \overline{\boldsymbol{z}}_e^a,$$

where the index $i = 1, \ldots, d$ denotes the members of the ensemble. The matrix $\mathbf{A} \in \mathbb{R}^{(m+n)\times(m+n)}$ is such that the sample covariance of the updated ensemble is identical to (4.2) and the sample mean is equal to (4.3). The matrix $\mathbf{A}$ is calculated using several different rotation and scaling matrices and the derivation for it can be found in the appendix A of [1].

The analysis step, specifically the matrix $\mathbf{A}$, does many things to the prior ensemble. First the prior ensemble in rotated into a coordinate system, where $\mathbf{P}^f$ and $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ become diagonal. Next this diagonal $\mathbf{P}^f$ is scaled to be an identity matrix and the same scaling is applied to the diagonal $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. In this transformed space the mean $\overline{\boldsymbol{z}}_e^a$ is calculated. Then the analysis ensemble can be obtained by shifting the transformed ensemble to the new mean $\overline{\boldsymbol{z}}_e^a$ and contracting it along the transformed coordinate axes according to the transformed $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. Finally the respective inverse transformations are applied to the transformed ensemble to return to the original state space. [9]

Chapter 5

Application to soil organic carbon modelling

In this chapter we will introduce the Yasso model, our observations and the DART software. Yasso is a model for the decomposition of soil organic carbon, later referred to as SOC. Our research project focused on enhancing the estimates of the Yasso15 model which is the latest version of the Yasso model. [13] The observations are from different sites across Europe, where the amount of organic carbon in the soil was measured regularly for several decades. Lastly we will discuss what the necessary computations were and how the data assimilation was run with the software platform DART.

5.1 Yasso model

Soil organic carbon can be divided into five pools based on decomposition rate; acid soluble (A), water soluble (W), ethanol soluble (E), non-soluble compounds (N) and humus (H). The decomposition of the first four may result in the formation of carbon dioxide or compounds of another pool. This is demonstrated in the figure 5.1, where the most likely results of decomposition for each pool are denoted by arrows.

The pools decompose at their own rate, which depends on temperature and precipitation. Also in the case of woody litter, such as stems and stumps, size affects the decomposition rate. The humus pool is the slowest to decompose and stores the long-term SOC, which was the focus of the experiments from which we got the measurements introduced in the next section.

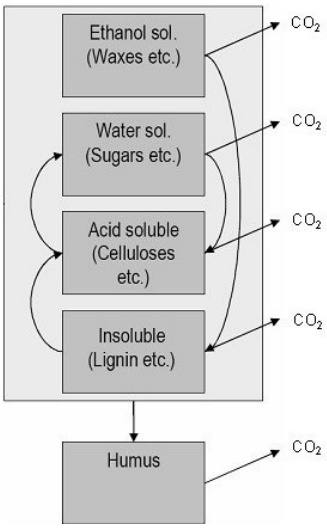


Figure 5.1: Decomposition of the SOC pools.

The model state vector $\boldsymbol{x}(t) = [x_A, x_W, x_E, x_N, x_H]^T$ denotes the mass of each pool and the change of the state in time is modelled as

$$\frac{d\boldsymbol{x}(t)}{dt} = \mathbf{D}\boldsymbol{x}(t) + \boldsymbol{b}(t), \tag{5.1}$$

where $\mathbf{D}$ describes how SOC decomposes and moves between the pools or leaves the system as carbon dioxide due to soil respiration. The last term $\boldsymbol{b}(t)$ is possible litter input which enters the system. The above equation is a first order non-homogeneous linear differential equation and we will solve it later. In the solution we will consider $\mathbf{D}$ and $\boldsymbol{b}$ to be constants. This can be done because Yasso is run one year at a time and for this time period $\mathbf{D}$ and $\boldsymbol{b}$ are determined and constant. The matrix $\mathbf{D}$ actually depends on environmental conditions and is defined as

$$\mathbf{D} = \begin{bmatrix} -1 & p_{WA} & p_{EA} & p_{NA} & 0 \\ p_{AW} & -1 & p_{EW} & p_{NW} & 0 \\ p_{AE} & p_{WE} & -1 & p_{NE} & 0 \\ p_{AN} & p_{WN} & p_{EN} & -1 & 0 \\ p_H & p_H & p_H & p_H & -1 \end{bmatrix} \mathbf{G},$$

where p_{ih} is the the relative flow from pool i to pool h and $i, h \in \{A, W, E, N, H\}$. The humus pools differs from the others. There is no flow from the humus pools to the other pools, $p_{Hh} = 0$. Also the relative flow from every other pool to humus is the same, $p_{iH} = p_H$. The matrix $\mathbf{G} = \mathrm{diag}(g_A, g_W, g_E, g_N, g_H)$ contains the decomposition rates g_i for the pools, which are defined as

$$g_i = \frac{\alpha_i}{J} \sum_{j=1}^{J} \exp(\beta_{i1} T_j + \beta_{i2} T_j^2)(1 - \exp(\gamma_i P))u(s), \tag{5.2}$$

where $\boldsymbol{T} \in \mathbb{R}^J$ is temperature and P is annual precipitation. The parameter α_i is a decomposition rate for a pool without considering temperature and precipitation, β_{i1} and β_{i2} are temperature dependencies and γ_i is a precipitation dependence. The decomposition slows down the larger the litter is due to the fact that it takes more time for the bacteria to reach the inside of the litter. This is described by $u(s) = \min((1 + \phi_1 s + \phi_2 s^2)^r, 1)$, where ϕ_1, ϕ_2 and r are parameters and s is the diameter of the woody litter.

The temperature vector $\boldsymbol{T}$ is a sinusoidal that takes into account the seasonal variations during one year. It is calculated from an annual mean T_m and an amplitude defined as $T_a = \frac{1}{2}(T_{max} - T_{min})$, where the maximum and minimum temperatures are the mean of the warmest and coldest months of the year.

The equation (5.2) for temperature, precipitation and size dependence was obtained in articles [17], [18] and [19]. In [17] the equation for temperature dependence is chosen empirically from six different models. Similarly in [18] it was described how the precipitation

dependence was chosen from different alternatives so that it had the highest probability with respect to measurements. Lastly the size dependence term $u(s)$ was obtained in [19] by considering three different models of which this form gave the highest posterior probability using Bayesian model comparison theory. The model parameters we used had been calibrated using experimental data and Bayesian inference by utilising MCMC simulation algorithm.[13] [10]

When the parameters, initial state $\boldsymbol{x}_0$, and the environmental conditions are determined and for a constant litter input $\boldsymbol{b}$ we can solve the differential equation (5.1) for $\boldsymbol{x}(t)$. The solution for this type of ODE, with constant $\mathbf{D}$ and $\boldsymbol{b}$, is

$$\begin{aligned}
\boldsymbol{x}(t) &= e^{\mathbf{D}(t-t_0)}\boldsymbol{x}_0 + e^{\mathbf{D}t}\int_{t_0}^{t} e^{-\mathbf{D}q}\,dq\boldsymbol{b} \\
&= e^{\mathbf{D}t}\boldsymbol{x}_0 + e^{\mathbf{D}t}\int_{0}^{t} e^{-\mathbf{D}q}\,dq\boldsymbol{b} \\
&= e^{\mathbf{D}t}\boldsymbol{x}_0 + e^{\mathbf{D}t}(-\mathbf{D}^{-1}e^{-\mathbf{D}t} + \mathbf{D}^{-1})\boldsymbol{b} \\
&= e^{\mathbf{D}t}\boldsymbol{x}_0 - \mathbf{D}^{-1}\boldsymbol{b} + \mathbf{D}^{-1}e^{-\mathbf{D}t}\boldsymbol{b} \\
&= \mathbf{D}^{-1}[e^{\mathbf{D}t}(\mathbf{D}\boldsymbol{x}_0 + \boldsymbol{b}) - \boldsymbol{b}],
\end{aligned}$$

where we defined $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ and $t_0 = 0$. We also used the fact that every matrix commutes with its exponential, i.e. $e^{\mathbf{A}t}\mathbf{A} = \mathbf{A}e^{\mathbf{A}t}$. The above equation is used as the model operator $\mathcal{M}$ in the data assimilation.

5.2 Observational data

The data we used to test the data assimilation is from a long-term vegetation-free experiments where the decay of SOC was monitored for decades after all inputs of carbon had stopped. The time it takes for a carbon atom bounded to a plant to return to the atmosphere is called turnover time. The study focused on carbon that has a turnover time of centuries or more. The duration of the experiments was short compared to the turnover time and this was taken into consideration when estimating the uncertainties for the data. The sites were kept vegetation-free for at least 25 years and the amount of SOC was measured regularly. [5] Note that our model state consists of the five pools whereas our observations are their sum, so the model state is not directly observed, which is often the case.

The sites of the experiment were located in Europe in different climate conditions and had different soil types. We used measurements from five different locations of which one had two fields so six in total.

Site	Years	Duration in years	Number of observations
Askov B3	1956-1987	31	30
Askov B4	1956-1987	31	29
Grignon	1959-2008	49	11
Rothamsted	1959-2008	49	7
Ultuna	1956-2007	51	18
Versailles	1928-2008	80	9

In addition to these measurements we had daily weather data from each of these sites for the duration of the experiment. From the weather data we got the annual precipitation and minimum, maximum and mean temperatures for each year, which were given as input to the model.

5.3 Computations and the DART toolbox

The software platform DART was designed for running different types of data assimilation problems. A user can implement their own types of observations and model into DART and then run data assimilation using a filter of their choosing e.g. EAKF or EnKF. [2] We used a previous version of DART called Classic, and not the newest version Manhattan.

Both DART and Yasso are coded using FORTRAN so adding Yasso into DART required just a few lines of code when reading and writing the states in and out of the model. From the input file Yasso reads the current state and the length of the time step. Yasso and DART handle the state data files in different format so we also needed to add a code to convert the data in to the right format between DART and Yasso.

Before running our assimilation we needed to define new types of observations in DART and tell DART how to handle them. This was made quite easy and required only simple additions to a few files and changes in some settings. Before the observations could be added they were written in a file format that DART can read.

We got the initial states for the sites from [15], where the history of each site was taken into account when estimating the carbon in each pool at the beginning of the experiments. The initial ensemble of states was also created with R by taking the initial state for the pools as the mean and with a 10% variance. The distribution for the initial ensemble for each site can be seen on figure 5.2.

DART reads the start time from the initial ensemble and checks if there is an observation at that time. If there is, the observation is used to compute a new analysed ensemble. This is then give to the model which moves the states forward in time one by one until the time of the next observation. This is repeated until the last observation. We ran the assimilation with inflation factors ranging from 1.0 to 1.5, with which the forecast states

were multiplied. In the results in the next chapter we will show the assimilations done without inflation or with an inflation factor of 1.25.

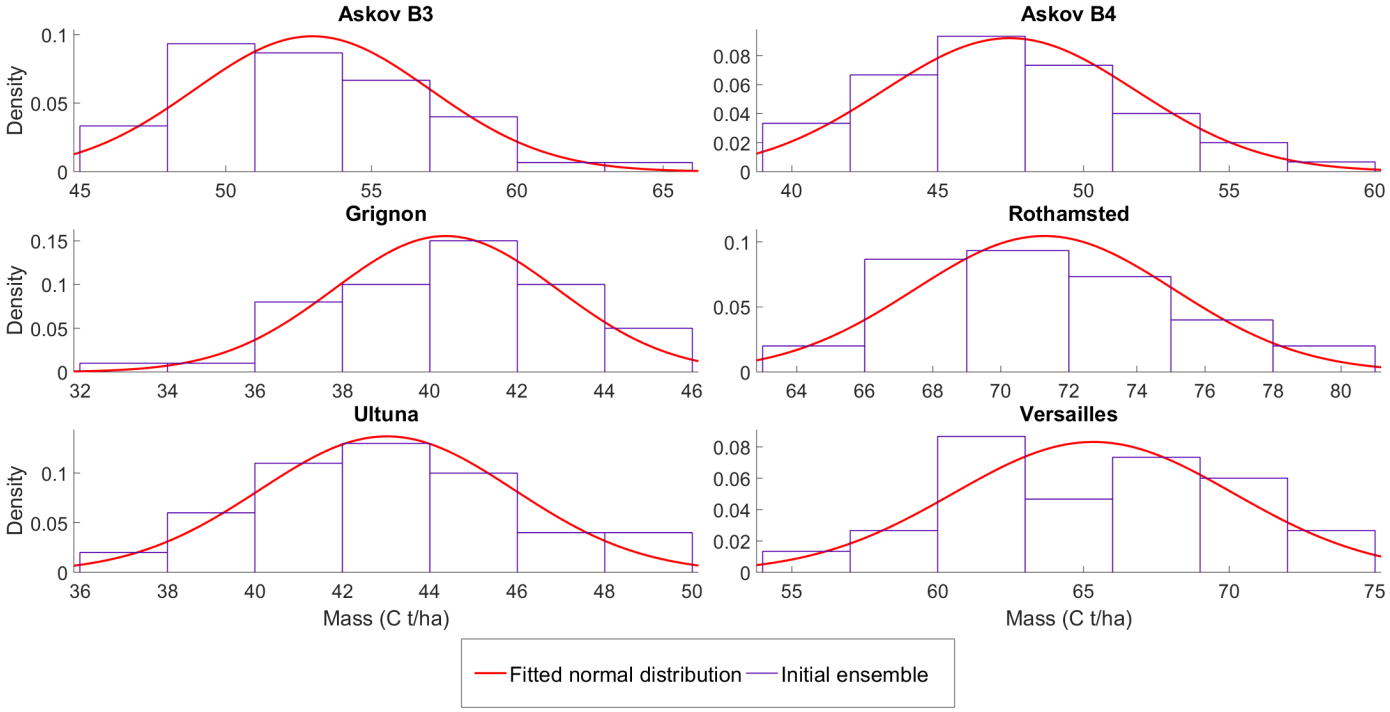


Figure 5.2: The distribution of the initial ensembles for each site

Chapter 6

Results and discussion

Here we present the results we got from running our data assimilation problem with DART using the EAKF filter. The results were plotted using MATLAB. The different sites, and the measurements made in them during the experiments, can be seen in figures 6.1 and 6.2. In all the figures we can also see the model prediction without data assimilation for comparison.

In figure 6.1 we have plotted the forecast state at each time step, i.e. when all the observations before that time have been assimilated into the estimation. This demonstrates how the filter works over a long period of time with several observations and at each point we see the forecast that our model gave for that time. The figure shows assimilations that were done with no inflation and with an inflation factor of 1.25.

We notice that in both of the Askov fields there is a systematic change in the observations during the experiment, in field B3 in 1977 and in B4 in 1966. We also see how filter divergence affects the estimate without inflation after a few observations. Even after the systematic change in the measurements the analysis trusts the previous forecast more than the observations. Whereas the estimation with inflation factor 1.25 corrects towards the new trend in the observations.

In Grignon and Rothamsted we have fewer observation than in Askov. In both we see an improvement that is quite consistent with and without inflation. In Ultuna we see a clear correction at the third observation because we are looking at the forecast states. The measurement error in Ultuna is also smaller than in the other sites and the variance in both of the estimates stays very small after the first three observations. The initial ensemble in Versailles, that was seen in figure 5.2, seems have a bimodal rather than a normal distribution. We ran the computations also with an initial ensemble that has clearly normally distributed and this didn't have any significant effect on the results. So even though the initial ensemble was bimodal by chance the ensemble method gave stable results.

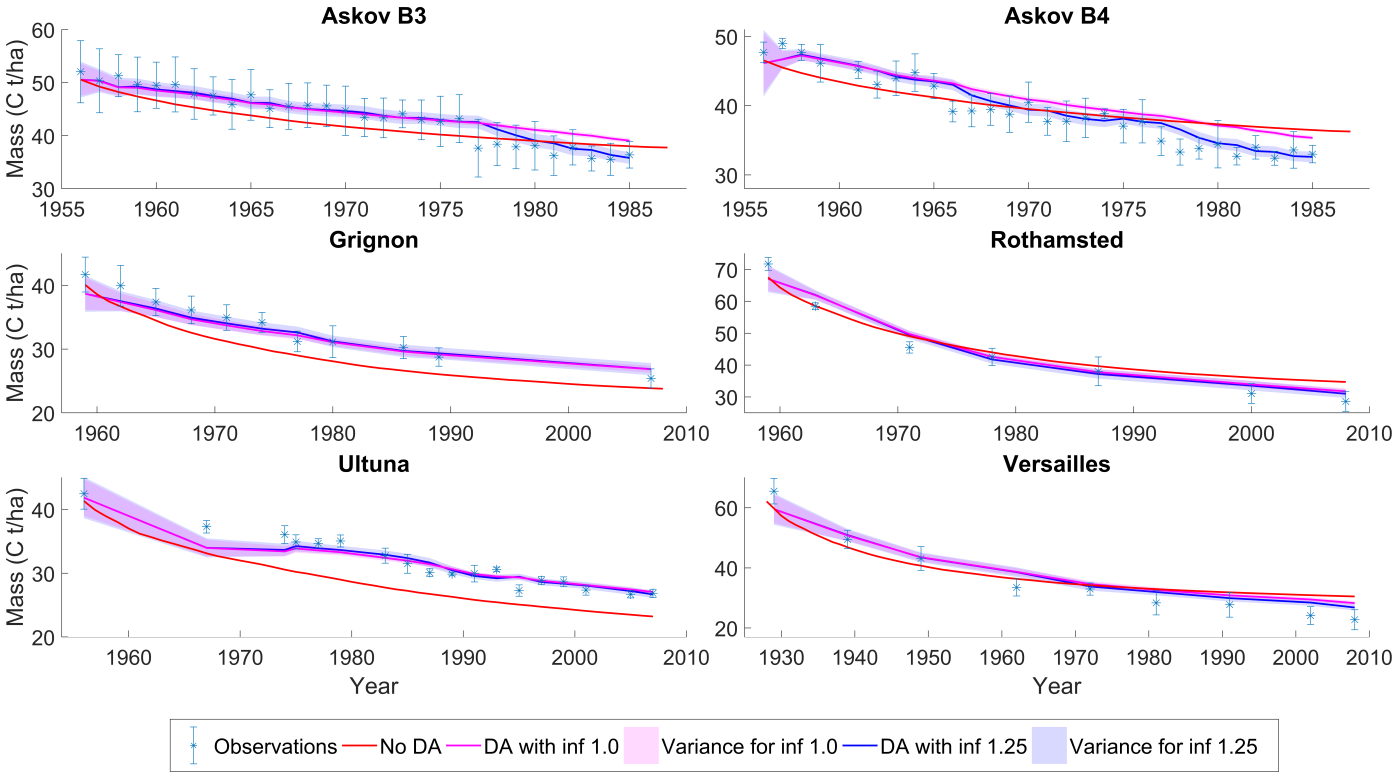


Figure 6.1: Forecast for the total amount of SOC in each year without inflation and with inflation factor 1.25

Next we took the first four analysis states, calculated with an inflation factor of 1.25, as initial states and ran the model onward, as can be seen in figure 6.2. With these four cases we can see how with just a few observations the filter enhances the future forecasts. We could have made these same estimates at the time of the fourth observation in each site.

Again we notice the systematic change in Askov. This time we assimilated observations only before this change so the model estimates follow the trend from the measurements before the change. After the change our estimates are clearly worse than the original model estimate.

In Grignon and Ultuna the model estimates are considerably closer to the measurements compared to the original estimate and we see clear improvement even after the second observation. Rothamsted has the fewest number of observations overall but they have a quite clear trend that we were able to follow after assimilating the first three observations. However in Versailles assimilating the first four measurements isn't enough to enhance the initial estimate. From these we can see that assimilation enhances the model estimate quite fast and just few observations are enough to have an effect.

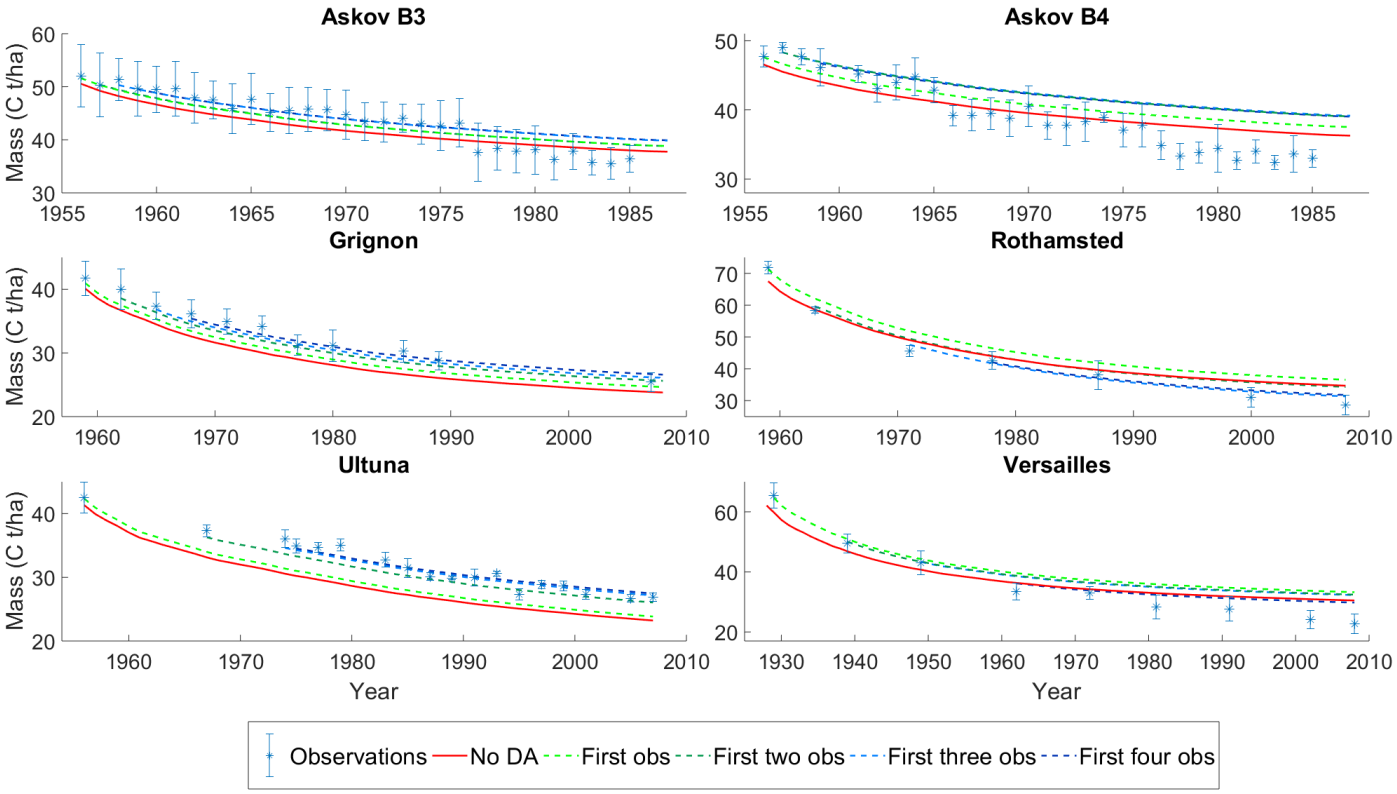


Figure 6.2: Estimates when assimilating the first four observations with inflation 1.25.

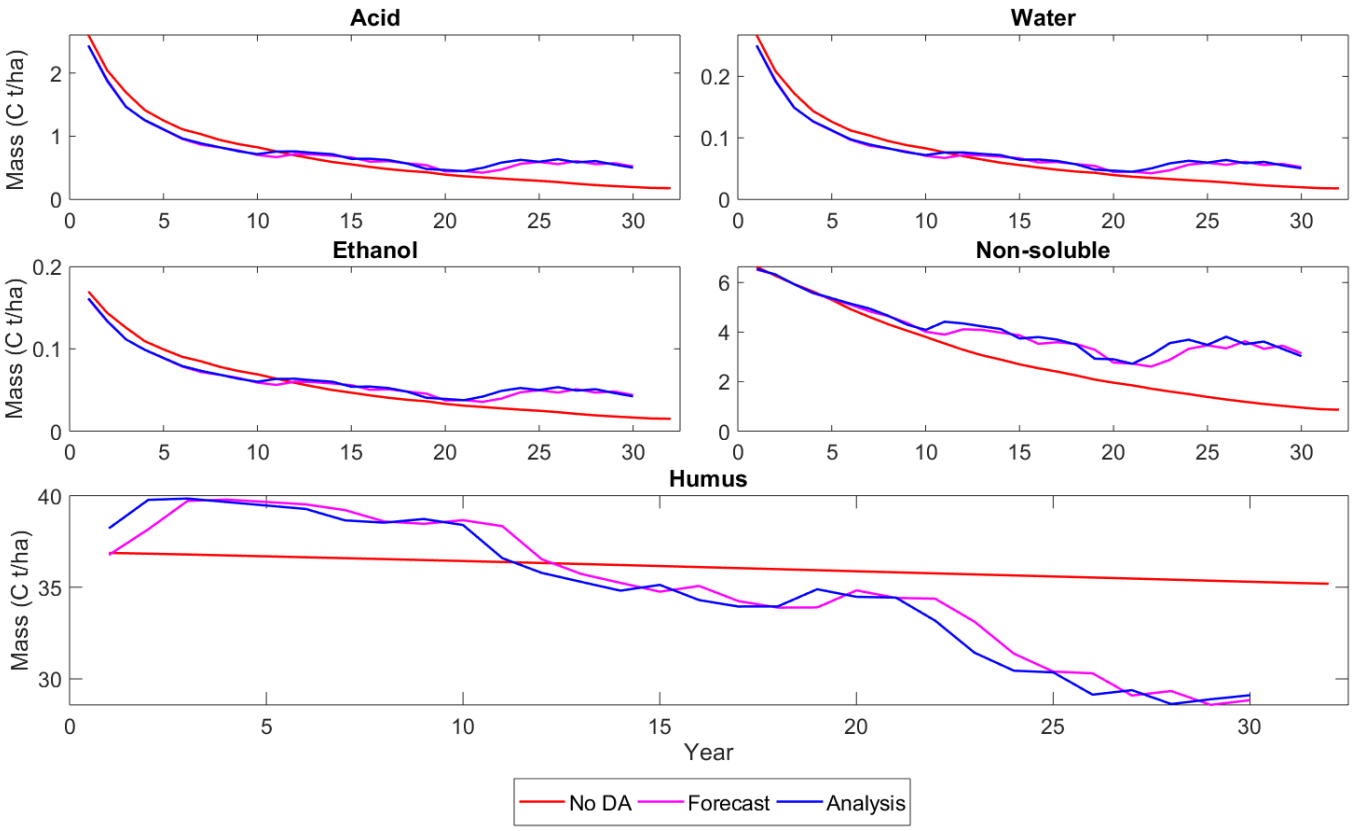


Figure 6.3: The estimates for the pools in Askov B4 with inflation 1.25

Next we were interested in seeing how data assimilation affected the model state, i.e. the five carbon pools, since the observations were only of the total amount of SOC. In figure 6.3, we take a look at how the soil organic carbon is distributed among the pools during assimilation with an inflation factor of 1.25 in Askov B4. In this figure we can see how the error covariance matrix affects all the pools during the assimilation according to the correlations between the pools. When the total amount of carbon decreases it is not taken evenly off the pools but we even see growth at times. This is something that we would expect to be happening in reality and it demonstrates that the model is able to represent the actual chemical reactions the way we understand them.

Chapter 7

Conclusion

In this thesis we have discussed data assimilation and derived the Kalman filter equations from the Bayes' theorem and proved that the error estimation process is optimal. We also discussed the Extended Kalman filter which provides a way to deal with non-linear systems. We introduced ensemble methods based on the Kalman filter and discussed filter divergence and ways to handle it.

We then introduced the Yasso model for soil organic carbon decomposition. We had observations form several bare fallow experiments and Yasso was used to estimate the amount of SOC in the fields during these experiments. The Ensemble Adjustment Kalman filter was used to enhance the estimates of the Yasso model with information from the observations. We were able improve the estimates with data assimilation and it also showed that the model understands the physical processes of SOC decomposition in a realistic way. The results were promising and further work should be done in more complex systems, e.g. with decaying plant litter in the system.

Bibliography

[1] Jeffrey L. Anderson. "An Ensemble Adjustment Kalman Filter for Data Assimilation". In: *Monthly Weather Review* 129.12 (2001), pp. 2884–2903. DOI: 10.1175/1520-0493(2001)129<2884:AEAKFF>2.0.CO;2.

[2] Jeffrey Anderson et al. "The Data Assimilation Research Testbed: A Community Facility". In: *Bulletin of the American Meteorological Society* 90.9 (2009), pp. 1283–1296. DOI: 10.1175/2009BAMS2618.1.

[3] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data Assimilation : Methods, Algorithms and Applications*. 2016.

[4] Johnathan M. Bardsley. *A Matrix Theoretic Derivation of the Kalman Filter*. URL: https://www.matrix-inst.org.au/wp_Matrix2016/wp-content/uploads/2018/05/Bardsley.pdf. (accessed: 12.09.2018).

[5] P. Barré et al. "Quantifying and isolating stable soil organic carbon using long-term bare fallow experiments". In: *Biogeosciences* 7.11 (2010), pp. 3839–3850. DOI: 10.5194/bg-7-3839-2010.

[6] Gerrit Burgers, Peter Jan van Leeuwen, and Geir Evensen. "Analysis Scheme in the Ensemble Kalman Filter". In: *Monthly Weather Review* 126.6 (1998), pp. 1719–1724. DOI: 10.1175/1520-0493(1998)126<1719:ASITEK>2.0.CO;2.

[7] Geir Evensen. "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In: *J. Geophys. Res* 99 (1994), pp. 10143–10162.

[8] Geir Evensen. "The Ensemble Kalman Filter: theoretical formulation and practical implementation". In: *Ocean Dynamics* 53.4 (Nov. 2003), pp. 343–367. ISSN: 1616-7228. DOI: 10.1007/s10236-003-0036-9.

[9] Gernot Geppert. "Analysis and application of the ensemble Kalman filter for the estimation of bounded quantities". PhD thesis. Jan. 2015. DOI: 10.17617/2.2161673.

[10] Heikki Haario et al. “DRAM: Efficient adaptive MCMC”. In: *Statistics and Computing* 16.4 (Dec. 2006), pp. 339–354. ISSN: 1573-1375. DOI: 10.1007/s11222-006-9438-0.

[11] J. Humpherys, P. Redd, and J. West. “A Fresh Look at the Kalman Filter”. In: *SIAM Review* 54.4 (2012), pp. 801–823. DOI: 10.1137/100799666.

[12] K. Ide et al. “Unified notation for data assimilation: Operational, sequential and variational”. In: *Journal of the Meteorological Society of Japan* 75.1B (1997), pp. 181–189.

[13] M. Järvenpää et al. “Soil carbon model Yasso15 - Bayesian calibration using worldwide litter decomposition and carbon stock data”. Manuscript in preparation.

[14] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Transactions of the ASME – Journal of Basic Engineering* 82 (Series D) (1960), pp. 35–45.

[15] L. Kulmala and J. Liski. “Bare fallow experiments highlight the importance of long-term history on soil carbon decomposition rate on agricultural lands.” In: *Proceedings of ‘The Centre of Excellence in Atmospheric Science (CoE ATM) – From Molecular and Biological processes to The Global Climate’* (2018), pp. 225–227. URL: http://www.atm.helsinki.fi/FAAR/reportseries/rs-215.pdf.

[16] K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. 2012. URL: http://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf. (accessed: 05.10.2018).

[17] Mikko Tuomi et al. “Heterotrophic soil respiration—Comparison of different models describing its temperature dependence”. In: *Ecological Modelling* 211.1 (2008), pp. 182–190. ISSN: 0304-3800. DOI: https://doi.org/10.1016/j.ecolmodel.2007.09.003.

[18] M. Tuomi et al. “Leaf litter decomposition—Estimates of global variability based on Yasso07 model”. In: *Ecological Modelling* 220.23 (2009), pp. 3362–3371. ISSN: 0304-3800. DOI: https://doi.org/10.1016/j.ecolmodel.2009.05.016.

[19] M. Tuomi et al. “Wood decomposition model for boreal forests”. In: *Ecological Modelling* 222.3 (2011), pp. 709–718. ISSN: 0304-3800. DOI: https://doi.org/10.1016/j.ecolmodel.2010.10.025.

[20] Christopher K. Wikle and L. Mark Berliner. “A Bayesian tutorial for data assimilation”. In: *Physica D: Nonlinear Phenomena* 230.1 (2007), pp. 1–16. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2006.09.017.