Bayesian inference in plant growth models for prediction and uncertainty assessment

Yuting Chen

ECOLE CENTRALE DES ARTS ET MANUFACTURES "ECOLE CENTRALE PARIS"

Thèse
présentée par Yuting CHEN
pour l'obtention du GRADE DE DOCTEUR

Spécialité: Mathématiques Appliquées
Laboratoire d'accueil: Mathématiques Appliquées aux Systèmes (MAS)

Inférence Bayésienne dans les Modèles de Croissance de Plantes pour la Prévision et la Caractérisation des Incertitudes

Bayesian Inference in Plant Growth Models for Prediction and Uncertainty Assessment

Soutenue le 27 juin 2014 N° d'ordre: 2014ECAP0040

Devant un jury composé de :

Mme. Florence D'ALCHÉ-BUC	Université d'Évry-Val d'Essonne	(Rapportrice)
M. Fabien CAMPILLO	INRIA	(Rapporteur)
M. Paul-Henry COURNÈDE	ECP	(Directeur de thèse)
M. Gilles FAŸ	ECP	(Président)
M. Jean-Louis FOULLEY	Université de Montpellier II	(Examinateur)
Mme. Martine GUÉRIF	INRA	(Examinatrice)
M. Samis TREVEZAS	ECP	(Examinateur)

Simplicity is the ultimate sophistication.

Leonardo da Vinci

It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.

Arthur Conan Doyle

Today's posterior distribution is tomorrow's prior.

David Lindley

Acknowledgement

This PhD thesis is the result of a challenging journey. It's with emotion and certainly a lot of sincerity that I would like to thank those from whom I have received tremendous help and those who encouraged and supported me to complete this endeavour.

First and foremost, my sincerest gratitude goes to my supervisor, Prof. Paul-Henry Cournède. The enthusiasm he has for his research has been an example for me, contagious and motivational even during tough times. Thank you for putting your trust in me and for giving me all the opportunities, guidance, patience and encouragement. I greatly appreciate your willingness to listen and I feel truly grateful to your expertise that allowed me to shape this dissertation. We have studied so many algorithms and tested so many hypotheses together! It was an honour and privilege to work under your supervision and surely this PhD would not have been achievable without you.

Furthermore, I'm indebted to Director of Research Fabien Campillo and Prof. Florence d'Alché-Buc for agreeing to be part of the reading committee and providing me insightful suggestions and advices, which allowed me to improve the manuscript and to enrich my research ideas. I'd like to extend my thanks and gratitude to the rest of my dissertation jury, Prof. Gilles Gaÿ, Director of Research Jean-Louis Foulley, Director of Research Martine Guérif and Dr. Samis Trevezas, for your invaluable feedback on my research, for being supportive of my work and for standing by me through this process.

This project could not come to fruition without the kindness of all the members of the group DigiPlante. They have contributed immensely to my personal and professional time at Centrale. The group has been a source of friendship as well as useful advices and collaborations. Undertaking this PhD would not have been easy without you guys. Particularly, I'm grateful for growing up with the other PhD students and engineers, Cédric, Chi-Thanh, Claire, Fenni, Gautier, Qiongli, Robert, Xiujuan and Zhongping who used to share the office with me, especially Charlotte and Marion who have acted as families to me and helped me with the last minute preparation. I have very much enjoyed exchanging ideas and sharing food with all of you and I can't forget those seminars and conferences that we attended together. Thank you, Robert, for those countless badminton training sessions and for taking the risk of missing your train and being there with me waiting for the liberation and the defence report. Thank you Samis, for all those emails that we've exchanged and all the fruitful discussions. You've always been so helpful and provided me with your assistance throughout my dissertation. Thank you, Véronique, for showing me how to make measurements of trees in China and for inviting me to supervise so many intriguing projects of fresh-year students. Moreover, I feel particularly grateful to the senior engineer of our group Benoît who often answered my emails and requests at late hours. Thank you for being patient and helping me to debug from time to time. I could not have been able to run so many tests of CPF without your help.

My deep appreciation also goes out to the other members of the MAS lab. Thank you Erick, Gilles (as well as Paul-Henry and Véronique) for according me this job of teaching assistant which allowed me to gain so much teaching skills and experiences

with constructive feedback. Many thanks to Annie and Sylvie who provided me with efficient help for all the administrative paperwork. I'm also thankful to the administrator of mesocenter Laurent who assisted me with parallel computation and the optimization of the resource of mesocenter.

Finally, none of this would have been possible without the love and understanding of my family. I'd like to say a heartfelt thank you to my parents who have raised me up and always believe in me and support me in all my pursuits. And most of all, I want to thank my husband Lionel, who has faithfully backed me up and encouraged me all along, especially during the final stage of this PhD. Thank you for being there for me.

This project has been undoubtedly an enormous test of commitment, patience and perseverance for me. Fortunately, I have not achieved this alone. Along the way, I have received various help in many ways from so many people whose names are not mentioned here. As a foreign student, I feel very lucky. I just want you to know that these contributions have not gone unnoticed but only acknowledged with genuinely appreciation and sincere gratitude.

Yuting C.

Abstract :

Plant growth models aim to describe plant development and functional processes in interaction with the environment. They offer promising perspectives for many applications, such as yield prediction for decision support or virtual experimentation in the context of breeding. This PhD focuses on the solutions to enhance plant growth model predictive capacity with an emphasis on advanced statistical methods. Our contributions can be summarized in four parts.

Firstly, from a model design perspective, the Log-Normal Allocation and Senescence (LNAS) crop model is proposed. It describes only the essential ecophysiological processes for biomass budget in a probabilistic framework, so as to avoid identification problems and to accentuate uncertainty assessment in model prediction.

Secondly, a thorough research is conducted regarding model parameterization. In a Bayesian framework, both Sequential Monte Carlo (SMC) methods and Markov chain Monte Carlo (MCMC) based methods are investigated to address the parameterization issues in the context of plant growth models, which are frequently characterized by nonlinear dynamics, scarce data and a large number of parameters. Particularly, when the prior distribution is non-informative, with the objective to put more emphasis on the observation data while preserving the robustness of Bayesian methods, an iterative version of the SMC and MCMC methods is introduced. It can be regarded as a stochastic variant of an EM type algorithm.

Thirdly, a three-step data assimilation approach is proposed to address model prediction issues. The most influential parameters are first identified by global sensitivity analysis and chosen by model selection. Subsequently, the model calibration is performed with special attention paid to the uncertainty assessment. The posterior distribution obtained from this estimation step is consequently considered as prior information for the prediction step, in which a SMC-based on-line estimation method such as Convolution Particle Filtering (CPF) is employed to perform data assimilation. Both state and parameter estimates are updated with the purpose of improving the prediction accuracy and reducing the associated uncertainty.

Finally, from an application point of view, the proposed methodology is implemented and evaluated with two crop models, the LNAS model for sugar beet and the STICS model for winter wheat. Some indications are also given on the experimental design to optimize the quality of predictions. The applications to real case scenarios show encouraging predictive performances and open the way to potential tools for yield prediction in agriculture.

Keywords: State space models; parameter estimation; data assimilation; uncertainty analysis; yield prediction; LNAS; STICS; dynamic crop model; Markov chain Monte Carlo methods; Sequential Monte Carlo methods; Regularized Particle Filter; Convolution Particle Filter; Adaptive Metropolis; Parallel and interacting Markov chain Monte Carlo; Differential Evolution Adaptive Metropolis; Ensemble Kalman Filter.

Résumé :

La croissance des plantes en interaction avec l'environnement peut être décrite par des modèles mathématiques. Ceux-ci présentent des perspectives prometteuses pour un nombre considérable d'applications telles que la prévision des rendements ou l'expérimentation virtuelle dans le contexte de la sélection variétale. Dans cette thèse, nous nous intéressons aux différentes solutions capables d'améliorer les capacités prédictives des modèles de croissance de plantes, en particulier grâce à des méthodes statistiques avancées. Notre contribution se résume en quatre parties.

Tout d'abord, nous proposons un nouveau modèle de culture (Log-Normal Allocation and Senescence ; LNAS). Entièrement construit dans un cadre probabiliste, il décrit seulement les processus écophysiologiques essentiels au bilan de la biomasse végétale afin de contourner les problèmes d'identification et d'accentuer l'évaluation des incertitudes.

Ensuite, nous étudions en détail le paramétrage du modèle. Dans le cadre Bayésien, nous mettons en œuvre des méthodes Monte-Carlo Séquentielles (SMC) et des méthodes de Monte-Carlo par Chaînes de Markov (MCMC) afin de répondre aux difficultés soulevées lors du paramétrage des modèles de croissance de plantes, caractérisés par des équations dynamiques non-linéaires, des données rares et un nombre important de paramètres. Dans les cas où la distribution a priori est peu informative, voire non-informative, nous proposons une version itérative des méthodes SMC et MCMC, approche équivalente à une variante stochastique d'un algorithme de type Espérance-Maximisation, dans le but de valoriser les données d'observation tout en préservant la robustesse des méthodes Bayésiennes.

En troisième lieu, nous soumettons une méthode d'assimilation des données en trois étapes pour résoudre le problème de prévision du modèle. Une première étape d'analyse de sensibilité permet d'identifier les paramètres les plus influents afin d'élaborer une version plus robuste de modèle par la méthode de sélection de modèles à l'aide de critères appropriés. Ces paramètres sélectionnés sont par la suite estimés en portant une attention particulière à l'évaluation des incertitudes. La distribution a posteriori ainsi obtenue est considérée comme information a priori pour l'étape de prévision, dans laquelle une méthode du type SMC telle que le filtrage par noyau de convolution (CPF) est employée afin d'effectuer l'assimilation de données. Dans cette étape, les estimations des états cachés et des paramètres sont mis à jour dans l'objectif d'améliorer la précision de la prévision et de réduire l'incertitude associée.

Finalement, d'un point de vue applicatif, la méthodologie proposée est mise en œuvre et évaluée avec deux modèles de croissance de plantes, le modèle LNAS pour la betterave sucrière et le modèle STICS pour le blé d'hiver. Quelques pistes d'utilisation de la méthode pour l'amélioration du design expérimental sont également étudiées, dans le but d'améliorer la qualité de la prévision. Les applications aux données expérimentales réelles montrent des performances prédictives encourageantes, ce qui ouvre la voie à des outils d'aide à la décision en agriculture.

Mots clés : modèle de Markov caché ; estimation paramétrique ; inférence Bayésienne ; assimilation de données ; analyse d'incertitude ; prévision de rendement ; LNAS ; STICS ; modèle de culture ; méthode de Monte Carlo par chaîne de Markov ; méthode de Monte-Carlo séquentielle ; filtre particulaire régularisé ; filtrage par noyaux de convolution ; Metropolis adaptatif ; méthode de Monte Carlo par chaînes de Markov en parallèle et en interaction ; évolution différentielle adaptative Metropolis ; filtre de Kalman d'ensemble.

Contents

CHAPTER 1

Introduction

PLANT growth modelling has become a key research subject in the field of agriculture, forestry and environmental science. In an interdisciplinary research landscape, the development of plant growth models has made considerable progress in the past two decades, thanks to the efforts made by biologists, mathematicians, computer scientists and the growing computation resources.

The main objective of plant growth modelling is to reproduce the growth of plants in interaction with the environment. It has a huge potential for many applications, such as to perform yield prediction in a qualitative and quantitative way for decision support, or by testing hypotheses and conducting virtual experiments to optimize the use of limited energy resources, which could otherwise take years in field conditions. All of these applications require plant growth models with accurate descriptions of the plant growth process as well as efficient mathematical and statistical methods to perform the parameterization and the prediction.

However, a major challenge for plant growth modelling is to keep up with the progress made in the related scientific domains (Fourcaud et al., 2008). This is especially difficult when it comes to the development of mathematical and statistical tools which play an essential role in a good practice of modelling, for most of the methods are developed outside of the plant growth model framework.

Therefore, this thesis encompasses the essential core steps of modelling, including model design, sensitivity analysis, model calibration (also known as parameterization), model selection, uncertainty analysis, model prediction and eventually design of experiments. The objective is to enhance the predictive capacity of plant growth models.

In the following, we first present the background of the prediction problem in plant growth modelling. An overview of the development of plant growth models and the statistical methods used is given. The problems of interest and the focus of this thesis are subsequently detailed.

1.1 Background and objectives

To improve or even to optimize the predictive capacity of plant growth models in various environmental contexts has been a long-standing challenge. A common idea is to enrich the mechanistic description of plant ecophysiology (Yin and Struik, 2010). With this purpose, particular efforts have been made to take into account abiotic stresses regarding temperature (*e.g.* Fowler et al. (2003)), water (*e.g.* Tardieu (2003)), or Nitrogen (*e.g.* (Bertheloot et al., 2011)). Some advanced agro-environmental models even aim at addressing the full diversity of environmental variations, like STICS (Brisson et al., 2003) or APSIM (Keating et al., 2003). However, the complexity of the interaction between processes can make the task rather difficult. Particularly in the case when several stresses are involved, it is delicate to describe the mixed effects of these combined stresses (Mittler, 2006). Hence, most models simply consider multiplicative stresses or a general stress value given by the maximal stress effects. As described by Yin and Struik (2010), the tendency is still to complicate the mechanistic description of biophysical processes, even by linking ecophysiology to "omics" sciences as an attempt for the full comprehension of the regulatory networks from which plant robustness and plasticity is supposed to emerge (Hirai et al., 2004), whilst a robust description appears difficult to achieve at the cell or tissue level.

This direction is clearly leading the way to great advances in research, especially in extending our understanding of how genotype leads to phenotype (Buck-Sorlin and Bachmann, 2000; Hammer et al., 2006; Yin and Struik, 2010). However, the objective of improving the prediction quality relies not only on providing an accurate prediction, but also on an appropriate assessment of the uncertainty, and further on reducing the uncertainty associated to the prediction.

It is obvious that the more complex the models are, the more troublesome their parameterization and the assessment of the estimate uncertainty become (Chen and Cournède, 2012; Ford and Kennedy, 2011), specifically due to the costly experimentation and the great number of unknown parameters to consider. This is thus an important drawback of this approach of building sophisticated models. Likewise, local environmental conditions (in terms of climatic and soil variables, as well as biotic stresses) and initial conditions in specific fields are also very delicate to characterize. Consequently, it may raise important issues regarding the identifiability of the parameters, the assessment of the confounding noises and the propagation of uncertainty and errors related to both parameters and inputs of these dynamic models. Failing to address these issues may finally result in poor predictions of plant-environment interactions in real situations, that is to say the opposite of the pursued objective.

Therefore, considering the lack of sufficient data assimilation approaches putting proper emphasis on the uncertainty assessment for prediction, the objective of this thesis is to propose a full methodology in a probabilistic framework for plant growth model prediction. This approach should be robust, efficient, adapted to the specific characteristics of plant growth models (nonlinear dynamics, restricted and irregular observation data) and provide a proper evaluation of the model uncertainties with the aim of reducing those associated to the prediction.

For this purpose, our research mainly focuses on three directions. The first concerns the design of an adequate model for prediction, then we are interested in the methods to perform a proper parameterization of the model with an emphasis on the reliable uncertainty assessment. Finally, the implementation of the data assimilation technique is investigated with the objective to improve the predictive capacity of plant growth models in general.

1.2 Problem formulation and contributions

1.2.1 Model design for prediction purpose

Pioneer studies carried out on data assimilation approach were aiming at specific farming conditions. A combination of a crop model and the data assimilation technique was used to update the model variables and / or parameters based on the observed data of early growth stages (Bouman, 1992; Delécolle et al., 1992; Maas, 1988; Moulin et al., 1998). This approach was particularly developed following the progress in deriving biophysical and biochemical canopy state variables from optical remote sensing (Dorigo et al., 2007), which could potentially give way to crop production forecast at large scales (Moran et al., 1997) and thus be considered as a tool for decision support (Gabrielle et al., 2002; Houlès et al., 2004).

The conventional strategy for prediction purpose is to employ reference models like SUCROS (Guérif and Duke, 1998, 2000; Launay and Guérif, 2005) or CERES (Dente et al., 2008) as the framework to integrate the remotely sensed observations to enhance the prediction quality. A few methods were developed under this perspective (see Dorigo et al. (2007) for a review). The forcing method consists in replacing a state variable of the model by the observed data, for instance the Leaf Area Index (LAI) in (Delécolle et al. (1992); Dente et al. (2008)). One important drawback is that generally a considerable part of the model state variables cannot be or are not observed and thus cannot be updated simultaneously at each time step. Moreover, the method does not take into account the observation error, which should not be neglected considering the general lack of accuracy of remote sensing data.

Another possibility is to use the available observation data to recalibrate some model parameters and / or initial states that may presumably vary with local conditions (Bouman, 1992; Guérif and Duke, 2000; Launay and Guérif, 2005). The main limitation of this method is that it requires sufficient data to perform the calibration, which is often costly and difficult to obtain. Besides, the global approach of this calibration step usually fails to capture and to maintain the system dynamics and thus is unable to improve the prediction accuracy.

On the other hand, it is noteworthy that in other research domains, data assimilation problems have been commonly reformulated and studied with a Bayesian probabilistic perspective, which allows the sequential estimation of model states and parameters simultaneously (Van Leeuwen and Evensen, 1996; Jazwinski, 1970) in the framework of generalized state-space models.

In the light of these applications, the first attempt to adapt a relatively simple

crop model into this perspective was made by Makowski et al. (2004). The method implementation relies on a probabilistic framework of crop model which is used to derive prior distributions of the model state variables and parameters at time steps with available observations while taking into account uncertainty in model prediction. Conditionally to the experimental observations and the observation error, posterior distributions are deduced according to Bayes' rule. An updated prediction of the model state variables can thus be inferred. The procedure is repeated at all measurement dates. Classical estimation methods used for this purpose are Ensemble Kalman Filter (see Evensen (2006) for the general presentation of the method, and Jones and Graham (2006) for an application in the context of crop models) or Particle Filter (see for example Kitagawa (1996) for the general concepts, and Naud et al. (2007) for an application in the context of crop models).

Nonetheless, one of the difficulties to implement this approach comes from the fact that it requires the plant growth model as well as the measurement model described in a probabilistic framework, as a hidden Markov model (Cappé et al., 2005). The classical and complex crop models (like STICS (Brisson et al., 1998), APSIM (Keating et al., 2003), CERES (Jones and Kiniry, 1986) ...) were not built in this perspective and their stochastic reformulation is therefore far from straightforward: the large number of involved processes may potentially lead to a drastic increase in the number of parameters to model process errors. One simple solution to circumvent this problem is to only consider observation errors (Guérif et al., 2006), but it may hinder a proper update of hidden state variables.

The above examples demonstrate the importance of having an appropriate model design to enhance the predicting performance. An adequate level of complexity may allow us to describe the principal plant development and functional process in interaction with the environment, while retaining the identifiability of the underlying parameters. Moreover, the model is better described in a probabilistic framework, so that the uncertainty evaluation can be carried out in an easier way.

By keeping these key points in mind, a new plant growth model describing biomass budget in crops is proposed in this thesis, with the particularity of being fully built in a probabilistic framework. Based on the analysis of Delécolle et al. (1992), we simplify the model description while retaining the major ecophysiological processes (at least in terms of Carbon economy): biomass production, biomass allocation, senescence and leaf surface development. The restriction of the model description to the fundamental processes allows an easier representation of the model errors without increasing significantly the number of parameters. The model is named LNAS (Log-normal allocation and senescence) in reference to the empirical functions of the (cumulative) Log-normal distribution employed to model allocation and senescence processes. Although the parameters of these functions are no longer supposed to be from genetic origin only, the adopted assumption is that the empirical functions describing the ecophysiological processes are flexible enough to adapt, and robust enough so that the mean prediction remains pertinent.

Currently the LNAS model is used to study simple crops such as sugar beet, maize, sunflower and wheat.

1.2.2 Model calibration and uncertainty assessment

The parametrization and the predictive capacity of plant growth models are considered to be complex and critical issues which are highly correlated.

Frequentist approaches

Facing limited and unevenly distributed measurements, important number of unknown parameters and complex interactions among compartments described in plant growth models, many studies are dedicated to tackle the parameterization problem in literature. Both frequentist and Bayesian approaches are considered. Yet most of the work put more emphasis on the model construction and the biological interpretation of the results, whereas the parameter estimation is usually performed with a simple frequentist approach with little attention to the uncertainty assessment. For instance, Zhan et al. (2003) and Guo et al. (2006) performed parameter estimation for the Greenlab models by minimizing an Ordinary Least-Squares criterion for applications to unbranched plants, such as cotton and maize, so is the case of de Reffye et al. (1999), who carried out a preliminary calibration for a hydraulic growth model for cotton. We could say that the objective in these studies is more about data fitting than parameter estimation.

However, with the increasing complexity of the considered models, nonlinear estimators are in great needs, as in the case of Hillier et al. (2005), who used a Generalized Nonlinear Least Square estimator with a maximum likelihood criteria and two Bootstrap methods to fit plant organ growth models. As a transition, Makowski et al. (2006) gave a general presentation of parameter estimation in crop models, in which the description of several more advanced estimation methods are presented including some basic Bayesian methods.

The first advantage of Bayesian methods is that they allow to take into account the prior knowledge. Indeed, although the experimental data are often restricted due to high costs, there is usually additional information available regarding the parameters of plants growth and development processes, specifically when they have clear biophysical interpretations. Hence, the estimation problem can be characterized as using experimental data to update the prior knowledge of the plant growth with the Bayesian philosophy. In this way, the weight of the observed data is limited and the estimation obtained is more interpretable.

Moreover, the Bayesian approaches provide a probability distribution as estimation for the unknown parameter which is opposite to the point estimation of the frequentist's philosophy. Therefore, not only the Bayesian approaches are capable of reconstructing the system by estimating simultaneously model parameters and unobservable state variables, but also they allow to evaluate the uncertainty related to the estimated model parameters and to identify the uncertainties stemming from other sources. In this way, the uncertainties can be handled properly and taken into account for the prediction process in order to achieve better prediction accuracy and reliable corresponding confidence intervals. These are very desirable features for estimation and prediction (data assimilation) problems in the context of plant growth models, which explain why sequential data assimilation techniques, especially the Bayesian filtering

methods have received considerable attention to deal with similar applications in other research areas (Anderson, 2001; Reichle et al., 2002; Mattern et al., 2013).

Yet despite all these desirable features, few efforts have been made to employ more sophisticated Bayesian approaches to solve plant growth models' parameterization problem. For example, Makowski et al. (2002) conducted a comparison between Generalized Likelihood Uncertainty Estimation method (GLUE) and a MCMC based method, the Metropolis-Hastings, for parameter estimation and Gaucherel et al. (2008) compared Particle filtering and a MCMC based method in the application to a process-based tree-growth model. Loi et al. (2011) introduced the first application of the filtering methods (Unscented Kalman Filter and Particle Filter) to a GreenLab model for Bayesian estimations. The comparison of these two methods can be found in (Loi, 2011).

Markov chain Monte Carlo based approaches

The Markov chain Monte Carlo (MCMC) based methods are very popular numerical approximations for Bayesian inference, which make the estimation of posterior distributions feasible in practice. However, some complex implementation issues can be encountered due to the specificity of plant growth models. First of all, the efficiency of the chain depends a lot on the proposal distribution. To tune the proposal distribution continuously, an adaptive scheme is proposed by Haario et al. (2001). Moreover, as pointed out by Geyer (1992), independent Markov chains may result in pseudo-convergence due to the multimodality of the target distribution. It seems that there is no satisfactory analytical tool to distinguish the pseudo-convergence from the real convergence, the only reliable way is to run the chain long enough, the strategy is thus known as "one long run". For this reason, the algorithm is very time consuming. Furthermore, if we aim at the joint estimation of both hidden states and unknown parameters, a poor mixing due to strong correlations between the two may result in very poor performance of the estimation (Liu et al., 1994; Roberts and Sahu, 1997). A possible solution to improve the mixture properties as well as the efficiency of the estimation is to make different copies of the same Markov chain and let them interact with each other, the strategy is known as "many short runs". Two well known algorithms developed in this perspective are the Interacting parallel MCMC (Campillo et al., 2009) and the Differential Evolution Adaptive Metropolis (DREAM) (Vrugt et al., 2009a). Therefore in this thesis, both strategies are investigated when applied to a plant growth model. In addition, a new Adapted Metropolis-within-Gibbs algorithm is proposed to cope with the mixing properties of the plant growth model in order to estimate both the hidden states and unknown parameters.

Note that given the iterative nature of MCMC-based methods, which means that they process all the observation data repeatedly, they are often regarded as *off-line* estimation approaches: when a new observation is available, the MCMC based methods have to run from the start again without being able to conserve the samples obtained formerly.

Sequential Monte Carlo approaches

Sequential Monte Carlo (SMC) methods are a set of simulation-based methods which

provide a convenient approach to compute the posterior distributions. The most representative methods of the SMC family are the filtering methods. Unlike MCMC-based approaches, filtering methods are performed in a sequential way. They are able to take into account the variation of parameters over time and carry out real-time updating, which is why they are known as *on-line* estimation methods.

The most efficient filtering method when dealing with linear systems is the Kalman filter (Kalman, 1960). To date, many efforts have been made to develop extensions of the Kalman filter for nonlinear systems. The most well known are the Extended Kalman Filter (EKF) (Evensen, 1994), the Unscented Kalman Filter (UKF) (Julier and Uhlmann, 1997; Quach et al., 2007) and the Ensemble Kalman Filter (EnKF) (Evensen, 1994). The EKF simply linearizes locally the model so that the traditional Kalman filter can be applied. However, when the nonlinearity is significant, it may cause divergence and the method proves to be no longer reliable. On the other hand, the UKF adopts deterministic sampling aiming at using a small set of discretely sampled points, known as sigma-points (Julier et al., 2000; Wan and Van Der Merwe, 2000), to get hold of the information of higher order for both the mean and the covariance matrix. But it performs poorly when confronting large scale nonlinear models because of its dependence on the Gaussian assumption and its limited sample size. The EnKF relies on normality assumptions in order to improve the accuracy of its estimates with a more important number of samples compared to the UKF. Both latter methods generalize to nonlinear systems in an elegant way without relying on the linearization required by the EKF.

Another important approach is the Particle Filtering (PF). Unlike the Kalman filter based methods, PF uses a set of randomly drawn samples with each an associated weight to represent the probability distribution of the hidden states and parameters conditioned on a series of observations. During decades, the development of PF-based methods has thrived since the Sequential Importance Sampling (SIS). The main weakness of SIS is the potential weight degeneracy (Arulampalam et al., 2002) and sample impoverishment (Gordon et al., 1993). Gordon et al. (1993); Kitagawa (1996) proposed a similar method, the Sequential Importance Resampling (SIR), by introducing the idea of resampling, which allows to overcome the degeneracy problem in most of the cases. Comparing to the Kalman-based filters, the PF-based methods have the advantage of being able to better predict nonlinear growth behaviours, for example, occasional skewness resulting from sudden or unusual climate changes.

Although the literature on the PF methods is considerably rich, the Convolution Particle Filter (CPF) (Campillo and Rossi, 2009; Musso and Oudjane, 1998; Rossi and Vila, 2006) which can be regarded as a generalization of the regularized particle filter proposed by Musso and Oudjane (1998), stands out for its attractive features regarding the challenges raised by parameter estimation and data assimilation of crop models. Firstly, the method is not only rather easy to adapt (with very few tuning parameters), but also robust in terms of convergence since it circumvents the classical problem of potential weight degeneracy in the SIR when the parameter identification issues occur. This property is valuable in classical field experimentations for which irregular or heterogeneous data are available. When these models are formalized as state-space hidden Markov models, the CPF can achieve a proper evaluation of model

uncertainty. Another interesting feature is that it works as well with deterministic models, which also makes the method straightforwardly adaptable to the classical and widely used crop models.

In this thesis, the above Bayesian approaches are used for model calibration of the LNAS model derived for sugar beet growth. Their performances are compared with both simulated data and real experimental data.

Iterative approaches

When the prior is non-informative and few observation data are available, it is preferable to put more weight on the data. Intuitively, an iterative approach is proposed in this thesis which can be regarded as a variant of Expectation-Maximization (EM) algorithm. The EM algorithm Dempster et al. (1977) is generally used to find the maximum-likelihood estimate of the parameters of an underlying distribution based on a dataset which is incomplete or has missing values, which corresponds to our case of rare and irregular experimental data. Some recent efforts to obtain appropriate stochastic EM-type algorithms for complex models arising in plant growth model applications can be found in Trevezas and Cournède (2013).

In contrast to classical stochastic variants (Celeux and Diebolt, 1985; Wei and Tanner, 1990; Delyon et al., 1999), in this thesis we present a simple way of turning a non-explicit M-step into an explicit one. Both MCMC based methods and SMC based methods are chosen to perform the E-step. We mention that until now the use of this type of Particle Filter was restricted to Bayesian type of estimation.

Consequently, the uncertainty related issues are mainly discussed, especially regarding the noise parameter calibration. The robustness of this iterative approach is examined with the application to LNAS model derived for sugar beet growth, based on both simulated data and real experimental data.

Implementation with PyGMAlion

All the methods and models tested in this thesis are implemented in the modelling platform PyGMAlion (Plant Growth Model Analysis, Identification and Optimization) (Cournède et al., 2013), which can be seen an attempt to promote a modelling framework and model analysis methods with the objectives of enhancing good modelling practices and increasing model design efficiency.

1.2.3 Model prediction with data assimilation

On-line estimation methods for model prediction

As stated previously, unlike the MCMC based methods, the filtering methods have the advantage of processing sequentially. This is a very attractive feature, for it enables the real-time updates for model prediction. New observed data can be assimilated to re-adjust the calibration of the model sufficiently without the need of eliminating all the previous estimations as the MCMC based methods do. In addition, both the model error and observation error can be quantified in the prediction process, which

is why the filtering methods are particularly suited to perform model prediction with uncertainty assessment.

Three-step approach for model prediction
In this thesis, we intend to provide a clear view about the numerical methods available to tackle the parameter estimation and yield prediction problem with both deterministic and stochastic plant growth models built with different perspectives, while putting a special emphasis on the uncertainty assessment.

Inspired by the various applications mentioned above, a full methodology to perform data assimilation with plant growth model is proposed in this thesis. A three-step data assimilation approach is proposed to address the model prediction problem :

- Parameter selection : the most influential parameters are identified by global sensitivity analysis and selected by model comparison.
- Model calibration : the unknown parameters are estimated either with a Bayesian perspective (with Sequential Monte Carlo (SMC) methods or Markov chain Monte Carlo (MCMC) methods), or with a frequentist perspective (with an iterative version of the previous Bayesian methods) from on a experimental dataset. The uncertainty analysis is as well conducted to identify various sources of noises and determine the confidence interval associated to the parameter estimates if the iterative approaches are employed. Although various estimation methods offer many options, according to the characteristic of the considered model, estimation strategies are summarized.
- Prediction with data assimilation : the filtering methods are carried out to benefit from the available data of early growth stages with regular updates of the state and parameter estimates obtained from the calibration step. The objective is to improve the prediction accuracy as well as to reduce the associated uncertainty. The prediction confidence interval can thus be derived from the posterior distribution.

This approach is applied to both the STICS model for winter wheat growth and the LNAS model for sugar beet growth.

1.3 Dissertation outline

In Chapter 2, we introduce a new model, the LNAS model, which is designed in the purpose of performing model prediction with an emphasis on uncertainty assessment. We also provide a brief overview of the main principles of the STICS model, which is one of the most classical crop models. It represents another modelling philosophy by enriching the description of growth processes.

As mentioned above, most of the estimation methods employed in the thesis are considered under a Bayesian framework. Therefore, basic notions and concepts of Bayesian inference are recalled in Chapter 3, some classical algorithms of both MCMC based approaches and SMC approaches are presented in chapter 4.

A methodological contribution of this thesis is detailed in Chapter 5, proposing a variant of the EM-type algorithm using SMC and MCMC based methods to perform the E-step. Chapter 6 describes the approaches adopted to perform uncertainty assessment for stochastic models Subsequently, the numerical implementation of the algorithms mentioned as well as the PyGMAlion modelling platform are presented in Chapter 7.

In Chapter 8, the Bayesian methods and the frequentist-based methods are applied to LNAS to perform parameter estimation with both simulated datasets and experimental dataset. The objective is to identify their performance with different configurations and in various modelling conditions. Note that both functional parameters and (observation and modelling) noise parameters are taken into account, and the corresponding implementation is detailed and estimation results are discussed.

Subsequently, a three-step approach is proposed in Chapter 9 to address prediction problems in plant growth models of different types. The approach is applied both to the stochastic LNAS model for sugar beet and to the deterministic STICS model for winter wheat. In order to test the predictive capacity of the model in various situations as well as its pertinence to evaluate the associated uncertainty, real experimental data are used for validation. An introduction to optimal design of experiments is given to select the best experimental configurations with the objective of improving the prediction quality while taking into account the experimental cost.

Finally, Chapter 10 summarizes all the results, the methods and the proposed selection strategies are discussed. Some perspective are presented and possible future research directions are outlined.

CHAPTER 2

Plant Growth Models

In this chapter, we present the two plant growth models studied in this thesis. Both models belong to the category of crop models. They describe the crop growth at compartment level, per unit of land surface area.

The LNAS model is developed as a simplification of the GreenLab model (de Reffye et al., 2003), but with a probabilistic perspective in the form of hidden Markov models. We highlight the fact that the LNAS model can also be derived at individual scale. In our study, it is specifically derived for the sugar beet growth.

On the contrary of the LNAS model, the second model considered is a classical one, the STICS model. It is deterministic and is well known for its detailed description of plant-environment interactions. In this study, it is derived for the winter wheat growth.

2.1 LNAS model of plant growth

For data assimilation purpose, a general scheme for agrosystem models was given by Delécolle et al. (1992) and adapted by Dorigo et al. (2007). They underlined the main ecophysiologcal processes and key variables to describe the interaction between plants and their environment. We adapt this scheme to the sugar beet case in Fig. 2.1. The principal processes they suggested to consider are crop development, light interception, biomass accumulation, biomass partitioning and senescence.

A specificity of the LNAS model is that the main growth processes underlined in Fig. 2.1 are considered as stochastic processes when it appears relevant. The version presented does not detail how nitrogen, temperature and water stress effects are taken into account in the model, classical functions borrowed from Sunflo (Lecoeur et al., 2011), STICS (Brisson et al., 2008) and Pilote (Mailhol et al., 1997) can be used, when necessary.

We believe that a realistic plant growth model should include the variability in-

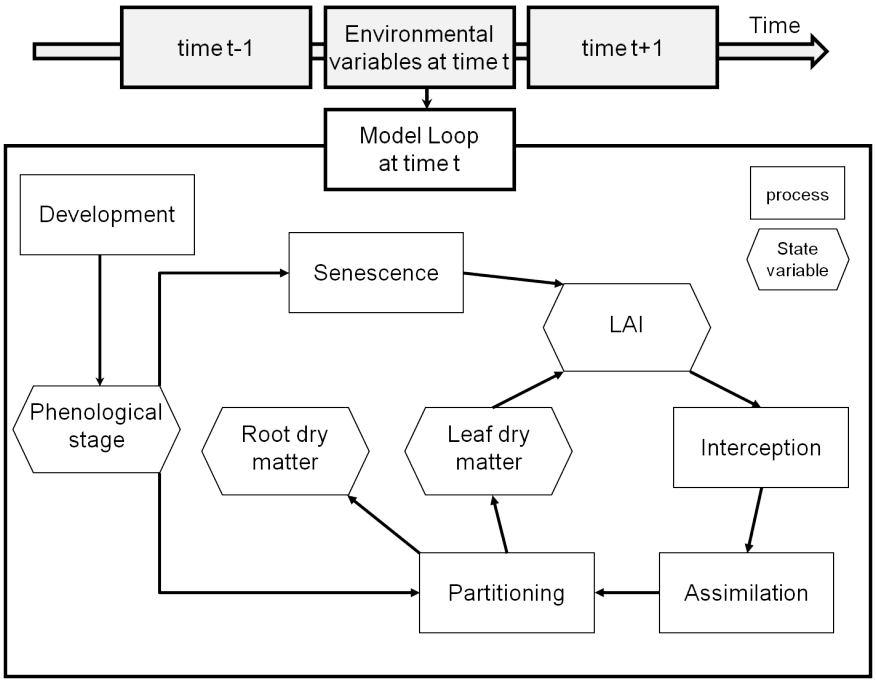


Figure 2.1: General scheme of an agroecosystem model for sugar beet (adapted from Delécolle et al. (1992) and Dorigo et al. (2007)).

volved in the way that the plant evolves (modelling noise, also known as *process noise* or *state noise*), as well as the variability regarding the estimation of the true value of the state of interest (observation noise) (de Valpine, 2002; Donnet et al., 2010), for there are multiple stochastic factors, including environmental noises and inter-individual variability, that are inherent in the plant functional growth processes. These noises are generally considered independent of the errors resulted from the sampling methods or from the observations (Dennis et al., 2006). Generally speaking, more and more models tend to take into account the observation noise (Hilborn and Ledbetter, 1979; Polacheck et al., 1993), but not the modelling noise, mostly for computation convenience. Nonetheless, if process noises are present in reality, then omitting them while building a model will lead to important bias in the likelihood function. Without modelling noise, the fit of the model is measured by only one trajectory (evolution of the state variables) that might have produced the data. This could result in bias in parameter estimation (Ludwig and Walters, 1981; de Valpine and Hastings, 2002; Schnute, 1995). By taking into account the modelling noise, an important range of trajectories that could have produced the data are incorporated in the fitting process. For the reasons mentioned above, it seems to be a more reasonable configuration for plant growth, which is why we opt for including both observation noise and modelling noise in our model.

Note that adaptations of the model are currently derived for maize, wheat and sunflower, and the equations can be adapted to other type of plant without difficulty by adding other types of compartments.

In the following, we present the equations of the Log-Normal Allocation and Senes-

cence (LNAS) model specifically derived for the sugar beet, per unit surface area, with two organ compartments: foliage and root system. The time step is one day and the environmental variables are daily averages.

Interception and assimilation: $Q(t)$ is the biomass production on day t per unit surface area ($g.m^{-2}$) which can be obtained by an adaptation of the Beer-Lambert law (Monteith, 1977): $\left(1 - e^{-\lambda \cdot Q_g(t)}\right)$ represents the fraction of intercepted radiation, with $Q_g(t)$ the total mass of green leaves on day t (in $g.m^{-2}$) and λ a parameter defined in $g^{-1}.m^2$. The biomass production of the whole plant is then deduced by multiplying the total amount of absorbed photosynthetically active radiation per unit surface area (PAR, in $MJ.m^{-2}$) and an energetic efficiency μ_a (in $g \cdot MJ^{-1}\cdot$):

$$Q(t) = \left(\mu_a \cdot PAR(t) \left(1 - e^{-\lambda Q_g(t)}\right)\right) \cdot (1 + \eta_Q(t)) \tag{2.1}$$

where we introduce the modelling noise $\eta_Q \sim \mathcal{N}(0, \sigma_Q^2)$. Since the characterization of the environmental variables and of the light interception is not accurately described by the Beer-Lambert law, the model noise appears relevant for the production equation. Despite the fact that $Q(t)$ should always be positive, we still use a normal law. With the multiplicative form and the levels of noise generally considered (inferior to 10%), there is no problem of positivity loss.

The parameter λ corresponds to $\lambda = k\,SLA$, where k is the Beer-Lambert extinction coefficient and SLA is the specific leaf area, so that the term $\left(1 - e^{-\lambda Q_g(t)}\right)$ can classically be rewritten as $\left(1 - e^{-k\,LAI}\right)$, with LAI the leaf area index. There is a slight difference in the formulation however, because we consider λ as an empirical constant parameter, whereas linking leaf mass to leaf surface via the SLA variable is not obvious since the SLA is known to vary during crop growth and within plants (see for example Jullien et al. (2009)), even though it is often regarded as constant in models. This simplification also allows us to avoid the differentiation between blades and petioles (which is not always easy from a botanical point of view in sugar beet), both constitute the leaf compartment.

Allocation to the foliage and root system compartments:

The description of the allocation process is a simplification of the GreenLab model (Yan et al., 2004; Guo et al., 2006) with the organ sink dynamics being described at compartment level: the proportion of biomass allocated to each compartment (foliage and root) is described by an empirical function γ:

$$Q_f(t+1) = Q_f(t) + \gamma(t) \cdot Q(t) \tag{2.2}$$
$$Q_r(t+1) = Q_r(t) + (1 - \gamma(t)) \cdot Q(t) \tag{2.3}$$

where

$$\begin{aligned}\gamma(t) &= \Gamma(t) \cdot (1 + \eta_\gamma(t)) \\ &= (\gamma_0 + (\gamma_f - \gamma_0) \cdot G_a(\tau(t))) \cdot (1 + \eta_\gamma(t))\end{aligned} \tag{2.4}$$

with $\tau(t)$ the thermal time, which corresponds to the accumulated daily temperature (above a threshold temperature, which is taken as 0 for the sugar beet (Lemaire et al., 2008)) since emergence day. G_a denotes the cumulative distribution function of a log-normal law parameterized by its median μ_γ and standard deviation σ_γ (for more explicit biological meanings of the parameters). γ_0 and γ_f correspond to the initial and final proportion of biomass allocated to the leaf compartment respectively (Figure 2.2). The modelling noise (process noise) is denoted by $\eta_\gamma(t) \sim \mathcal{N}(0, \sigma^2_{\gamma\gamma})$. The cumulative distribution of the log-normal law is chosen for its flexibility: it allows to reproduce dynamics similar to the sigmoid-type functions often employed to describe biological processes, while having the advantage to start with a null value on zero. The transformation chosen to obtain γ is inspired by the simulation of biomass allocation to root and leaf compartments of sugar beet described by SUCROS (Spitters et al., 1989) and GreenLab (Lemaire et al., 2008).

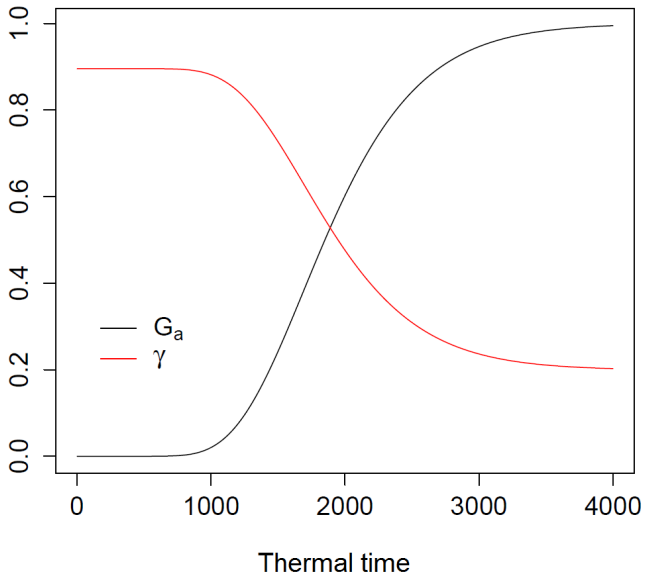


Figure 2.2: The cumulative distribution function G_a and the resulting allocation function γ, with the initial proportion of biomass allocated to the leaf compartment (γ_0) 0.9, and the final proportion of allocation (γ_f) 0.2.

The allocation strategy is very sensitive to environmental conditions, therefore we introduce a multiplicative perturbation, again following a Gaussian law.

Senescence: The senescent foliage mass Q_s is a proportion of the accumulated foliage mass given by the cumulative distribution of a log-normal law of median μ_s and standard deviation s_s:

$$Q_s(t) = G_s(\tau(t) - \tau_{sen})Q_f(t) \tag{2.5}$$

with τ_{sen} the thermal time at which the senescence process initiates. The green foliage mass Q_g can be hence obtained easily:

$$Q_g(t) = Q_f(t) - Q_s(t) \tag{2.6}$$

We choose a deterministic version of the senescence equation despite the strong variations that can characterize the process. As a matter of fact, the influence of senescence in the biomass budget is due to the decrease of photosynthetic foliage in Equation (2.1), adding a perturbation in the senescence mass either would be of second order or could be summarized in the modelling noise η_Q.

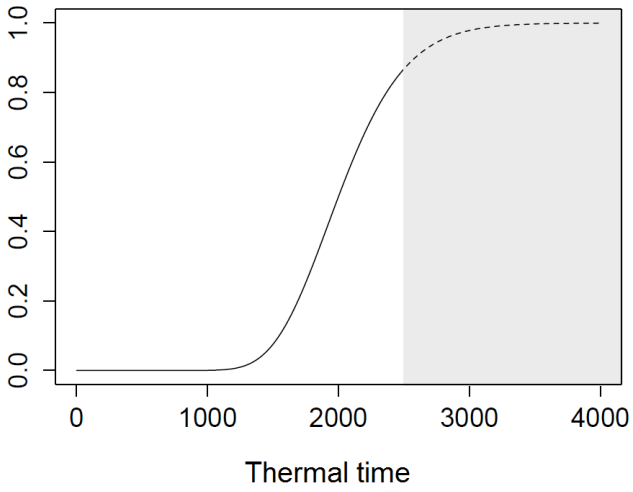


Figure 2.3: The evolution of the senescence function with the initial proportion of senescent biomass 0.0, and the final proportion 0.85. The gray part refers to the unobserved thermal time.

Note that both allocation and senescence processes are driven by the thermal time. The phenological stage in Fig. 2.1 is thus simply represented in our model by the course of the thermal time, from emergence.

Observations: As described in the next paragraph, the general state space models considered in this study describe observation equations in addition to the state equations. In our study, the observation variables potentially available from field measurements for data assimilation are:

$$Y(t) = \begin{pmatrix} Q_g(t) \cdot (1 + \xi_g(t)) \\ Q_r(t) \cdot (1 + \xi_r(t)) \end{pmatrix} \tag{2.7}$$

with measurement noises: $\xi_g(t)) \sim \mathcal{N}(0, \sigma_g^2)$, and $\xi_r(t) \sim \mathcal{N}(0, \sigma_r^2)$.

We remark that in the LNAS model, multiplicative Gaussian noise structure is adopted for both the modelling and observation noises. This choice is justified since the scale of measurement varies greatly across time (from 0.01g to 250g for one plant), an additive noise would have clearly led to some difficulties regarding the risk of positivity

loss or a misrepresentation of the complex measurement process (plants cutting, drying, weighing, averaging on large samples). With a multiplicative form of the noise and with the standard deviation of the modelling (respectively observation) noise usually around 5% (resp. 10% for observation noise) and always below 15% (resp. 20%), the probability for a random variable following a Gaussian distribution $\mathcal{N}(0, 0.2^2)$ to be inferior to -1 (in order to cause positive loss) is 2.9 10^{-7}. Therefore, the probability of obtaining a negative value is extremely small. There are undoubtedly other advantages of choosing a Gaussian distribution, for instance the symmetry (which is surely shared by many other distributions), the simplifications in maximum likelihood inference, the easy simulation process and its robustness.

However, other distributions could have been possible such as the log-normal distribution, specifically to prevent positivity loss. Both log additive or log multiplicative noise can be considered.

Another important point is that the proposed framework would still be applicable when satellite image data are used for LAI evaluation. In that case, LAI should be specified clearly as a state variable in the system (with an explicit formulation of the SLA variable), and a corresponding observation function should be defined.

2.2 STICS model of plant growth

The STICS model (Brisson et al., 2003) focuses on the crop-soil system and has already been applied to various crops. It is divided into several modules with each representing different plant growth mechanisms (Brisson et al., 2008). Among them, the development module is in charge of the evolution of the LAI and root compartment, and in the meantime defines the harvested organ filling phase. Fig. 2.4 illustrates the main processes involved in the STICS model adapted to the winter wheat crop.

Contrary to LNAS which is based on a biomass budget, the growth in STICS is driven by an empirical law for the LAI growth. Three phases are involved, the first phase (from emergence to the maximal LAI point) is approached by a logistic function with the hypothesis that the ratio of biomass between leaves and stems is constant, followed by a stabilized phase and a senescent phase of linearly decreasing LAI. Several stress factors limit the potential daily increase in LAI. The daily biomass production is then computed as a quadratic function of the intercepted radiation, given by the Beer-Lambert law. Hence, crop total biomass results from the accumulation of the daily biomass increase, and the final grain biomass is obtained through a harvest index.

In our study, the field experimentations were conducted without Nitrogen stress, but in light water stress conditions. Therefore, the soil characteristics are taken into account to compute the water balance of the plant-soil-atmosphere system and thus to estimate several water stress indices impacting plant growth at different levels. For this purpose, the water contents in three soil layers are calculated.

In consequence, the above model can be divided into two sub-models, one concerning the plant system with state variables at time t denoted $X_p(t)$ and the other about the soil system with state variables denoted $X_s(t)$. Plant growth is described by the

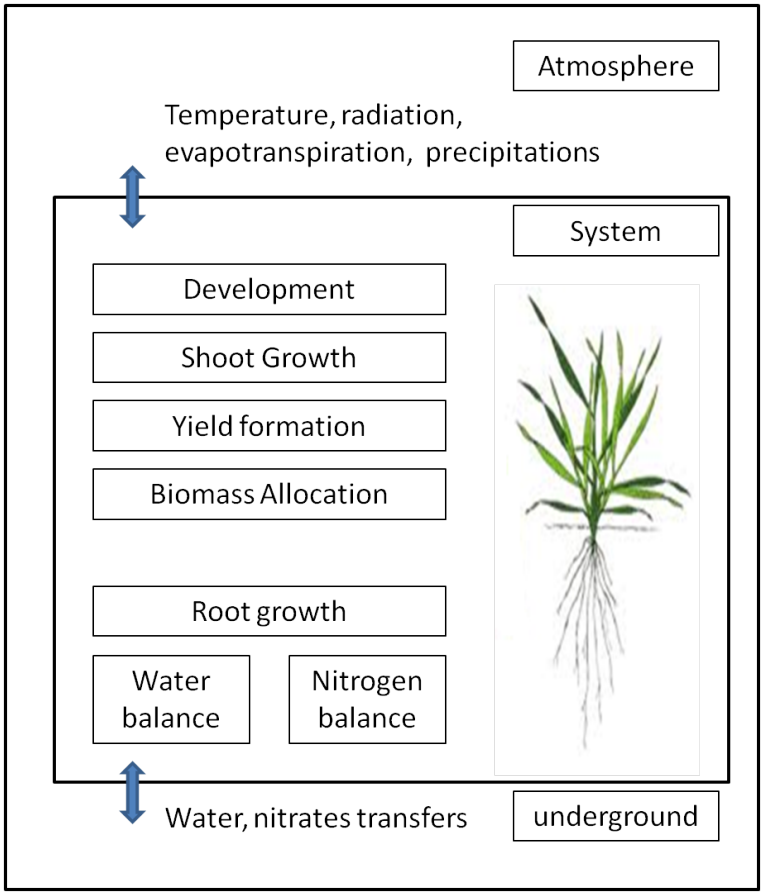


Figure 2.4: General scheme of the STICS model for winter wheat.

function f_p. The root compartment growth is directly affected by the soil temperature and water content. Likewise, the soil water content determines several water stress indices impacting LAI development, biomass production and senescence. On the other hand, water transfers including evaporation and plant transpiration are calculated in the soil system by the function f_s.

$$\begin{cases} X_p(t+1) = f_p\left(X_p(t), X_s(t), E_p(t), \Theta_p\right) \\ X_s(t+1) = f_s\left(X_s(t), X_p(t), E_s(t), \Theta_s\right) \end{cases} \tag{2.8}$$

The details of the equations can be found in Brisson et al. (2008).

Given the complexity of the STICS model, a large number of parameters are involved, some of which may be species dependent or genotype dependent.

CHAPTER 3

Preliminary - Estimation Theory

In parallel with theoretical advances, there has been a rise in the development of computationally efficient approximation models to address the increasing complexities of modern data analysis. With the availability of cheap and flexible computing power, simulation based Monte Carlo methods have gained great importance in probability and statistics. Mathematically based on the law of large numbers and conceptually based on a frequentist notion of probability, a lot of problems can be surmounted through approximation schemes with important number of simulations.

Before presenting the principles algorithms of the MCMC-based methods and the SMC methods in a general state-space setting, in this chapter, a number of important concepts to be used in the other sections are presented. First, we outline the fundamental difference between the frequentist and the Bayesian philosophies. A general probabilistic model for both frequentist and Bayesian inference objectives is described. Then, problems associated with the posterior computation which motivate the standard approximation methods are briefly discussed. Subsequently, the Monte Carlo methods are introduced. Some commonly used estimators are presented. Finally, we review some basic elements regarding the Markov chain principals and its behaviour to prepare for the use of Markov chains Monte Carlo methods on general state-spaces presented in the next chapter.

3.1 Estimation paradigms

Generally speaking, there are several streams of thought in statistical analysis (such as parametric or nonparametric, inferential or exploratory, robust or nonrobust), one of the most salient is frequentist or Bayesian (Robert, 2001). Their common objective is to explore a phenomenon based on observed data, like in the case of plant growth

modelling. The calibration of a given model is carried out based on past observations with the purpose of improving the understanding of the plant growth mechanisms and hopefully to be used as predictive tools for future applications. However, the two categories of the methods are based on different paradigms.

According to the frequentist approach, the studied events are assumed to have statistical stability, therefore, the parameter is supposed to have a true value which is fixed. A random confidence interval is usually provided, a 95% confidence interval indicates that if the experiment is repeated a large number of times, 95% of the confidence intervals calculated would contain the true value of the parameter.

On the other hand, in the Bayesian framework, the understanding of probability and of inference is conceptually different. A probability distribution can be associated with the parameter to express one's uncertainty and belief before taking into account the data [1]. The estimation consists in updating the previous (prior) knowledge of the studied event based on new observations. In some situations, a credible interval may be provided as a probabilistic region around the parameter. In the contrast of the frequentist confidence interval, it results from a probability distribution (posterior) which incorporates contextual problem-specific information given by the prior knowledge.

With that being said, generally for plant growth models, the limited observations contain frequently important noises, hence the importance of prior information and the advantage of Bayesian approaches.

Moreover, when it comes to the estimation of unknown parameters of a given model, frequentist methods rely on estimators whose properties are derived from distributions of repeated samples whereas Bayesian methods provide distribution of parameters conditional on the actual dataset observed. Given the settings and the objectives of plant growth modelling, one may come to the conclusion that the Bayesian approaches are more suitable by nature to address the issues of uncertainty assessment in model calibration and model prediction. As a result, in this thesis the Bayesian approaches are mainly investigated. Nevertheless, we also provide some results given by the frequentist approach in comparison, since in some situations, frequentist methods derive estimators based on the maximum likelihood function which offers similar estimation results compared to the Bayesian approach.

In the following, Bayesian inference framework is formally formulated and some frequently used estimators, both Bayesian and frequentist are presented for point estimation. First, we introduce the general state-space models.

3.2 General State-Space Models

For parameter estimation and sequential data assimilation in crop models, we rely on the statistical framework provided by the discrete nonlinear general state-space model. Here we first introduce the general formulation of a dynamic state-space model (Kalman, 1960; Harrison and West, 1989; Hamilton, 1994; Doucet et al., 2001) which

1. Parameters of prior distributions are known as *hyperparameters*, so as to distinguish from those of the model.

is usually defined by a state function and an observation function, as follows:

$$\begin{cases} X(t+1) = f_t\left(X(t), E(t), \Theta, \eta(t)\right), \\ \\ Y(t) = g_t\left(X(t), \Theta, \xi(t)\right). \end{cases} \tag{3.1}$$

The evolution equation is embodied in the function f_t, which is time dependent. $\{X(t)\}_{t\in\mathbb{N}}$ represents the state variables at time t, $X(t) \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$. Θ denotes the parameter vector, $\Theta \in \mathcal{P} \subseteq \mathbb{R}^{d_\Theta}$, and $E(t)$ represents the system input vector in a controlled environment, $E(t) \in \mathcal{E} \subseteq \mathbb{R}^{d_e}$. The modelling noise is represented with the random variables $\eta(t)$ (corresponding to model imperfections or uncertainty in the model inputs). The observation equation incorporates observations on the state variables of interest. $Y(t)$ is the output vector which is related to the state variable vector $X(t)$ through the function g_t. $\{Y(t)\}_{t\in\mathbb{N}^*}$ consists of state variables that can be observed experimentally (outcome) and usually differ from $X(t)$ (for instance, biomasses of some plant organs can be measured while the daily biomass production cannot), $Y(t) \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$. Measurement noises are denoted by $\xi(t)$. Here, we separate the noise parameters and the model parameters (also known as functional parameters), for they play different roles with respect to likelihood computation. $\{\eta(t)\}_{t\in\mathbb{N}}$ and $\{\xi(t)\}_{t\in\mathbb{N}}$ are considered as sequences of independent and identically distributed random variables.

Note that only in rare occasions (Makowski et al., 2004; Chen and Cournède, 2012; Trevezas and Cournède, 2013), plant models were built by really taking into account modelling and measurement noises. Models are generally written as deterministic dynamic systems. Of course, such deterministic models can still be represented with (3.1), the stochastic variables being zero with probability 1.

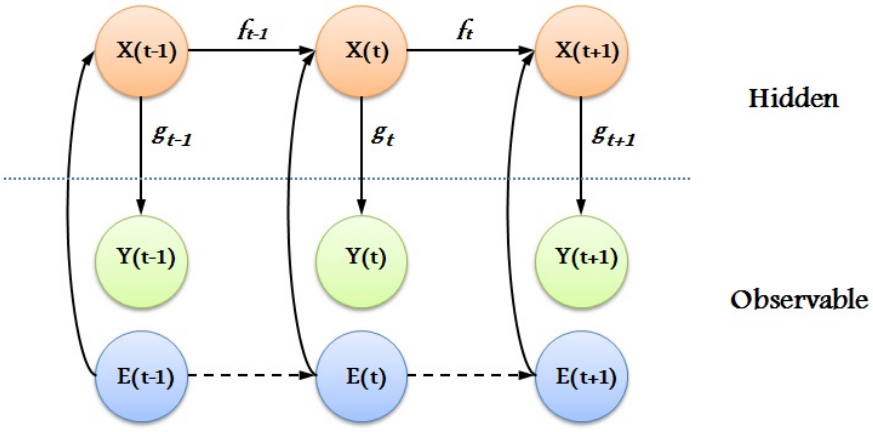


Figure 3.1: Generic state-space model.

Furthermore, since experimental data tend to be limited due to high costs, observations are only available at irregular times. Let $(t_1, t_2, ...t_N)$ be the N measurement time steps. For all $n \in \mathbb{N}$, we set: $x_n := X(t_n)$, $y_n := Y(t_n)$ and we denote $y_{1:n} := (y(t_1), y(t_2), \ldots, y(t_n))$.

If we reformulate the state-space dynamic model in a Bayesian perspective as a hidden Markov Model, it can be described as:

$$\begin{cases} x_0 & \sim p_\Theta(x_0) & \text{initial density,} \\ x_n & \sim p_\Theta(x_n|x_{n-1}) & \text{transition density,} \\ y_n & \sim p_\Theta(y_n|x_n) & \text{measurement density.} \end{cases} \tag{3.2}$$

with $\Theta \in \mathcal{P}$ and $n \geq 0$.

More discussions about the state space model and Hidden Markov model can be found in Jazwinski (1970); Robert and Casella (1999); Robert (2001); Kaipio and Somersalo (2005). The Bayesian approach for parameter estimation mainly consists of updating prior knowledge of parameter Θ, $\pi(\Theta)$, based on the observational evidence to a posterior distribution $\pi(\Theta|y_{1:n})$. The probability is thus interpreted as a conditional measure of uncertainty.

Before proceeding to Bayesian inference and Bayesian prediction, we first recall some basics of Bayesian statistics.

3.2.1 Bayesian statistics

To perform parameter estimation for plant growth models in a Bayesian framework, we intend to compute the posterior distributions for the parameters. However, complex models usually involve high dimensionality as well as nonlinearity, and the lack of data often occurs. These difficulties preclude analytical solutions: the posterior distributions are either impossible to obtain explicitly or difficult to sample from directly. Therefore, an approximation can be considered. Three types of problems inherently linked to the Bayesian statistics can be encountered: normalization, marginalization and expectation.

For the sake of simplicity, let Θ denote the unknown parameters, $\Theta \in \mathcal{P} \subseteq \mathbb{R}^{d_\Theta}$, and y the outcome variable $y \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$.

• Normalization
Given the prior $\pi(\Theta)$ and the likelihood function $\mathcal{L}(\Theta|y) = p(y|\Theta)$ which assesses the probability of obtaining y if Θ is the true value of the parameter, the posterior $\pi(\Theta|y)$ can be obtained according to Bayes' rule (Bayes, 1763) :

$$\pi(\Theta|y) = \frac{p(y|\Theta)\pi(\Theta)}{p(y)} = \frac{p(y|\Theta)\pi(\Theta)}{\int_{\mathcal{P}} p(y|\Theta)\pi(\Theta)d\Theta} \tag{3.3}$$

as the product of the prior and the likelihood divided by a normalizing factor, for which complex integrations are needed over the whole parameter space.

• Marginalization
Given the joint posterior (Θ, x), with $x \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$, the marginal posterior can be expressed as:

$$\pi(\Theta|y) = \int_{\mathcal{X}} p(\Theta, x|y)dx. \tag{3.4}$$

We show in the following sections (*e.g.* Sections 4.1.3, 4.3.3 and 5.2.1) that the marginalization plays an important role in Bayesian inference.

• **Expectation**
Given the conditional pdf, we are often interested in computing some averaged functions of Θ :

$$\mathbb{E}_{\pi(\Theta|y)}[f(\Theta)] = \int_{\mathcal{P}} f(\Theta)\pi(\Theta|y)d\Theta. \tag{3.5}$$

For instance, if f is an identity function, then we obtain $\mathbb{E}(\Theta|y)$.

Since the explicit computation of these integrals is generally intractable, we are led to consider approximate solutions based on Monte Carlo methods, which consist of sampling random variables from probability distributions so as to perform the integration implicitly.

3.3 Monte Carlo methods

One of the main numerical issues in statistical analysis concerns integration problems, which generally emerge with the likelihood approach and the Bayesian approach. Monte Carlo methods offer a way to evaluate integrals.

3.3.1 Classical Monte Carlo integration

Let π_u define the un-normalized density of interest on a measurable state space $\mathcal{X} \subset \mathbb{R}^d$, with respect to d-dimensional Lebesgue measure, which satisfies $0 < \int_{\mathcal{X}} \pi_u < \infty$. Let $\mathcal{B}$ denote the Borel σ-algebra associated to $\mathcal{X}$. We thus have the probability measure $\pi(\cdot)$ on $\mathcal{X}$, for $A \in \mathcal{B}(\mathcal{X})$:

$$\pi(A) = \frac{\int_A \pi_u(x)dx}{\int_{\mathcal{X}} \pi_u(x)dx}. \tag{3.6}$$

We wish to estimate the expectation of the function f: $\mathcal{X} \to \mathbb{R}$ with respect to $\pi(\cdot)$:

$$\pi(f) = \mathbb{E}_\pi[f(X)] = \frac{\int_{\mathcal{X}} f(x)\pi_u(x)dx}{\int_{\mathcal{X}} \pi_u(x)dx}, \tag{3.7}$$

for which direct evaluation of integration could be infeasible. Thus, by using the classical Monte Carlo method, if we have $\{X_i, i = 1, \ldots, n\}$ i.i.d. samples generated from $\pi(\cdot)$, then the following estimator is available :

$$\hat{\pi}(f) := \frac{1}{n}\sum_{i=1}^{n} f(X_i), \tag{3.8}$$

since

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} f(X_i) \longrightarrow \quad \mathbb{E}_\pi[f(X_i)] \quad a.s. \tag{3.9}$$

Particularly, in the case when $\pi(f^2) < \infty$, $\mathrm{Var}(\hat{\pi}(f))$ can be estimated according to the Central Limit Theorem, the speed of the convergence v_n can accordingly be assessed :

$$\mathrm{Var}(\hat{\pi}(f)) = \frac{1}{n}\frac{\int_{\mathcal{X}}(f(x) - \mathbb{E}_\pi[f(X)])^2\pi_u(x)dx}{\int_{\mathcal{X}}\pi_u(x)dx}, \tag{3.10}$$

$$\hat{v}_n = \frac{1}{n}\sum_{i=1}^{n}(f(Xi) - \hat{\pi}(f))^2, \tag{3.11}$$

and

$$\lim_{n\to\infty} \frac{\hat{\pi}(f) - \pi(f)}{\sqrt{\hat{v}_n}} \longrightarrow \mathcal{N}(0,1). \tag{3.12}$$

It means that the unbiased estimator has a standard deviation of order $O(1/\sqrt{n})$, which can lead to the construction of confidence bounds and convergence test.

Note that it is not necessary to draw samples directly from the distribution f to approximate the integral, especially when it is not easy to do so, some other techniques like *importance sampling* could also be employed (Ripley, 1988; Robert and Casella, 1999). It is also one of the most popular techniques to reduce the variance of the estimate. The idea is to represent the integration by an expectation of weighted samples.

3.3.2 Importance Sampling

The main idea is to evaluate the equation (5.8) (i.e. approximate the expected mean) based on i.i.d. samples $\{X_i, i = 1, \ldots, n\}$, generated from an arbitrary chosen distribution $g(\cdot)$. The expectation of f can be reformulated as follows:

$$\mathbb{E}_\pi[f(X)] = \frac{\int_{\mathcal{X}} f(x)\frac{\pi_u(x)}{g(x)}g(x)dx}{\int_{\mathcal{X}}\frac{\pi_u(x)}{g(x)}g(x)dx}, \tag{3.13}$$

which can be approximated and estimated by

$$\hat{\mathbb{E}}_\pi[f(X)] = \frac{\frac{1}{n}\sum_{i=1}^{n} f(X_i)\frac{\pi_u(X_i)}{g(X_i)}}{\frac{1}{n}\sum_{i=1}^{n}\frac{\pi_u(X_i)}{g(X_i)}}. \tag{3.14}$$

If we define

$$w(X_i) = \frac{\frac{\pi_u(X_i)}{g(X_i)}}{\sum_{j=1}^{n}\frac{\pi_u(X_j)}{g(X_j)}}, \tag{3.15}$$

a possible Monte Carlo estimate of the integration of $\mathbb{E}_\pi[f(X)]$ can be obtained :

$$\hat{\mathbb{E}}_\pi[f(X)] = \sum_{i=1}^{n} w(X_i) f(X_i). \tag{3.16}$$

The proposal density $g(\cdot)$ is chosen so as to be easy to sample from. g is sometimes referred to as *importance function*, and $w(x)$ as *importance weight*. Thus, $\{X_i, w(X_i)\}_{i=1}^n$ can be regarded as giving an approximation to $\pi_u(\cdot)$.

The advantage of introducing the proposal density is that it permits us to sample random variables from densities different from the original one. A noteworthy point is that the importance function is supposed to mimic the target distribution, in a way that it should be positive where the target distribution is positive, and particularly also be able to capture the peak of the target distribution.

Note that importance sampling is a general Monte Carlo integration method which brings considerable improvements compared to the classic Monte Carlo method. The estimate provided is consistent. However, the proposal distribution should have heavy tail in order to be insensitive to the outliers (Robert and Casella, 1999). Moreover, the method is computationally expensive for recursive estimation in state space models, since each time when there is a new data available, one needs to recompute the importance weights over the entire state space (Doucet et al., 2001). In the literature, many variations of importance sampling exist to overcome this problem. One of the most popular methods which generally works well for identification problems in dynamic systems, is sequential importance sampling. This method is shortly reviewed in Section 4.3.4.

3.4 Estimators and optimization criteria

The objective of Bayesian inference is to use prior knowledge, quantitatively and qualitatively, to infer the conditional probability given limited observations. Once the posterior distribution is constructed, an estimator is required. Here are some estimators and the associated criteria that can be used to measure the optimality.

1. **Minimum mean square error (MMSE)**
   The point estimation of Θ can be found by minimizing the prediction or filtering error. The MMSE estimator thus aims to obtain the conditional mean $\mathbb{E}[\Theta|y_{0:n}] = \int_{\mathcal{P}} \theta p(\theta|y_{0:n}) \lambda(d\theta)$, with λ the Lebesgue measure. It can be defined as :

$$\hat{\Theta}^{MMSE} = \underset{\Theta \in \mathcal{P}}{\operatorname{argmin}}\ \mathbb{E}(\|\Theta - \hat{\Theta}\|^2 | y_{0:n}) \tag{3.17}$$

2. **Maximum *a posteriori* (MAP)**
   The point estimation of Θ can as well be obtained by choosing the value of Θ

at which $\mathcal{L}(\Theta|y_{0:n})$ attains its maximum (mode) in the posterior density. The Maximum *a posteriori* (MAP) estimator is defined as :

$$\begin{aligned}\hat{\Theta}^{MAP} &= \underset{\theta\in\mathcal{P}}{\operatorname{argmax}} \;\; p(\theta|y_{0:n}) \\ &= \underset{\theta\in\mathcal{P}}{\operatorname{argmax}} \;\; p(y_{0:n}|\theta)p(\theta).\end{aligned} \tag{3.18}$$

3. **Maximum likelihood (ML)**
   If the prior is non-informative, the MAP estimator can be further simplified and as a result, be generalized to the maximum likelihood estimator (MLE):

$$\hat{\Theta}^{ML} = \underset{\theta\in\mathcal{P}}{\operatorname{argmax}} \;\; p(y_{0:n}|\theta). \tag{3.19}$$

   Notice that the range of MLE should coincide with the range of the parameter by construction. Under fairly general conditions (see Casella and Lehmann (1998)), the MLE is converging almost surely to the true value of the parameter. However, according to Berger and Wolpert (1984), it is situated at the fringe of the Bayesian paradigm.

N.B.

- We highlight that the above criteria and estimators are valid for state estimations as well as for parameter estimations.

- Both MMSE and MAP estimators require the computation of the posterior density distribution. However, MAP has the advantage of not requiring the calculation of the denominator (integration), which makes it less computational expensive. On the other hand, one of its drawbacks is that in high-dimensional space, high probability density could belong to a narrow distribution and thus does not imply high probability mass. Thus, the width of the mode is also very important to take into account. Moreover, in the case of a real parameter, it has no justification in terms of minimization of a risk function (Heinrich, 2014). Therefore, the MMSE estimator is selected for the Bayesian approaches presented in this thesis.

In the meantime, under very simple assumptions on the error model, the computation of maximum likelihood can also be implemented through the minimization of sums of square residuals, which introduces the **Least Squares** (LS) estimator. For nonlinear functions, it can be formulated as :

$$\hat{\Theta}^{LS} = \underset{\Theta\in\mathcal{P}}{\operatorname{argmin}} \sum_{i=0}^{n} \|y_i - g(\Theta)\|^2. \tag{3.20}$$

Regarding the estimation of model state variables, the generalized least squares estimator is often employed.

3.4.1 Generalized Least Squares Estimator

The classical parameter estimation technique Generalized Least Squares (GLS) is traditionally used when the measurement errors have unequal variance or are correlated. Generally, modelling errors are not considered, and the dynamics of the model is not taken into account in the error model. If we denote $\tilde{Y}(\Theta)$ the full output vector of a deterministic model with parameter vector Θ as in the equation (3.1), Y the corresponding experimental data, $\tilde{Y}(\Theta), Y \in \mathcal{Y} \subseteq \mathbb{R}^{d_y}$ and $\Theta \in \mathcal{P} \subseteq \mathbb{R}^{d_\Theta}$, we assume:

$$Y = \tilde{Y}(\Theta) + \xi$$

with ξ an additive measurement error. If $\mathrm{Var}(Y|\Theta) = \Sigma$ is known, the GLS estimator is given by:

$$\hat{\Theta}_{GLS} = \underset{\Theta \in \mathcal{P}}{\operatorname{argmin}} \left(Y - \tilde{Y}(\Theta)\right)^T \Sigma^{-1} \left(Y - \tilde{Y}(\Theta)\right). \tag{3.21}$$

If the model is linear, $\tilde{Y}(\Theta) = A\Theta$ with A a $n_y \times n_\Theta$ matrix, we can deduce (see for example Rao (1973)) that $\hat{\Theta}_{GLS}$ has a Gaussian vector of variance:

$$\mathrm{Var}\left(\hat{\Theta}_{GLS}\right) = \left(A^T \Sigma^{-1} A\right)^{-1}.$$

If the model is moderately nonlinear (see for example Wu et al. (2010) which gives some ways to characterize the nonlinearity of plant growth models), an approximation is given by:

$$\mathrm{Var}\left(\hat{\Theta}_{GLS}\right) = \left(\frac{\partial \tilde{Y}}{\partial \theta}\left(\hat{\Theta}_{GLS}\right)^T \Sigma^{-1} \frac{\partial \tilde{Y}}{\partial \theta}\left(\hat{\Theta}_{GLS}\right)\right)^{-1}. \tag{3.22}$$

When Σ is unknown, yet can be expressed as a function of Θ, $\Sigma(\Theta)$, then the strategy is to implement an iterative method with the multi-stage Aitken estimator (Taylor, 1977). Given Σ^0, $\hat{\Theta}^{(1)}$ can be obtained, which in turn can be used to update the estimation of Σ to obtain $\hat{\Sigma}^{(1)}$. The new estimate can be used to compute $\hat{\Theta}^{(2)}$ according to equation (3.21). This alternating estimation scheme can be iterated until both estimates stabilize. A very useful example is given by Taylor (1977) considering the case when Y can be gathered into q groups, $q \in \mathbb{N}^*$, with Y_i the error terms in group i, $(1 \leq i \leq q)$, supposed to have common unknown variance Θ_i. Σ is assumed given by a diagonal matrix:

$$\Sigma = \begin{pmatrix} \alpha_1 I_{Y_1} & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 I_{Y_2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \alpha_{q-1} I_{Y_{q-1}} & 0 \\ 0 & \cdots & 0 & 0 & \alpha_q I_{Y_q} \end{pmatrix}$$

with I_k, the identity matrix of rank k. Note that this method may induce rearranging the data in the $\tilde{Y}$ vector by grouping the data of the same type.

The 2-stage estimator proposed by Taylor is described as follows. From an algorithmic point of view, there are two stages in the estimation process:

(1) α_i is estimated in the first place as the variance of all experimental data in group i to give the first estimate $\hat{\Sigma}^{(1)}$ of Σ. We then obtain:

$$\hat{\Theta}_{2SA}^{(1)} = \underset{\Theta \in \mathcal{P}}{\operatorname{argmin}} \left(Y - \tilde{Y}(\Theta)\right)^T \left(\hat{\Sigma}^{(1)}\right)^{-1} \left(Y - \tilde{Y}(\Theta)\right).$$

(2) We then estimate α_i with:

$$\hat{\alpha_i} = \frac{1}{Y_i - n_\Theta} \left(Y_i - \tilde{Y}_i\left(\hat{\Theta}_{2SA}^{(1)}\right)\right)^T \left(Y_i - \tilde{Y}_i\left(\hat{\Theta}_{2SA}^{(1)}\right)\right),$$

to obtain $\hat{\Sigma}^{(2)}$ and the final estimator which is given by

$$\hat{\Theta}_{2SA} = \underset{\Theta \in \mathcal{P}}{\operatorname{argmin}} \left(Y - \tilde{Y}(\Theta)\right)^T \left(\hat{\Sigma}^{(2)}\right)^{-1} \left(Y - \tilde{Y}(\Theta)\right).$$

Finally the variance of $\hat{P}_{2SA}$ is approximated with Equation (3.22) (using $\hat{\Sigma}^{(2)}$).

N.B.

- This strategy is widely used in the application of GreenLab model calibration (Cournède et al., 2011).
- Note that when the underlying model is incorrect, other methods such as the Huber Sandwich Estimator (Huber, 1967) can be considered to provide a robust estimation of the variance of the MLE.

3.5 Markov chain Monte Carlo methods

MCMC has become a standard technique commonly used as a tool to generate large samples in order to estimate unknown characteristics of a target distribution. It implies building a Markov chain on a state space $\mathcal{X}$, produced by a discrete-time stochastic process $(X_n)_{n\in\mathbb{N}}$, to represent the posterior distribution. However, before implementation and interpretation, particular attention should be paid to the conditions and assumptions that need to be verified to establish the central limit theorem for Markov chains. In this section, some basic concepts of Markov chains which we need for the application in this thesis are briefly recalled.

More details regarding stochastic processes and Markov chains' properties can be found in (Robert et al., 1999; Robert and Casella, 1999; Jones and Hobert, 2001; Roberts and Tweedie, 2008).

3.5.1 Markov chains

Suppose $(X_n)_{n\in\mathbb{N}}$ is a homogeneous Markov chain with values in a general state space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. Although $\mathcal{X}$ is a general set, in practice, we are most likely considering d-dimensional Euclidean space, so that $\mathcal{X} = \mathbb{R}^d$, or some countable or uncountable

subset of $\mathbb{R}^d$. In order to define probabilities, $\mathcal{B}(\mathcal{X})$ is employed to denote a countably generated σ-algebra on $\mathcal{X}$: when $\mathcal{X} = \mathbb{R}^d$, $\mathcal{B}(\mathcal{X})$ will be taken as the Borel σ-algebra; when $\mathcal{X}$ is countable, $\mathcal{B}(\mathcal{X})$ contains all subsets; otherwise it may remain arbitrary.

Let $P(x, dy)$ be the transition kernel associated to the homogeneous Markov chain (X_n). For $x \in \mathcal{X}$, $A \in \mathcal{B}(\mathcal{X})$ and $i \in \mathbb{N}$:

$$P(x, A) = Pr(X_{i+1} \in A | X_i = x).$$

For all $n \in \mathbb{N}$, we define $P^n(x, dy)$ as the n-step Markov transition kernel, then for $x \in \mathcal{X}$, $A \in \mathcal{B}(\mathcal{X})$ and $i \in \mathbb{N}$:

$$P^n(x, A) = Pr(X_{n+i} \in A | X_i = x).$$

Since in our application context, $\mathcal{X} \subset \mathbb{R}^d$, we assume that for all $x \in \mathcal{X}$, the probability measure $P(x, \cdot)$ has a conditional density $k(\cdot|x)$ with respect to the Lebesgue measure,

$$P(x, A) = \int_A k(u|x)du.$$

The density k is referred to as the Markov transition density. If there exists a density π that satisfies :

$$\pi(x) = \int_{\mathcal{X}} k(x|y)\pi(y)dy, \tag{3.23}$$

then $\pi(\cdot)$ is called a *stationary distribution*, also known as *invariant distribution* of the Markov chain (X_n). It is easy to see that if we can simulate $X_0 \sim \pi(\cdot)$, then (X_n) is a sequence of dependent samples from $\pi(\cdot)$.

Definition 3.5.1 *A Markov chain transition kernel P is said to be reversible with respect to a probability distribution $\pi(\cdot)$ on $\mathcal{X}$, if:*

$$\pi(dx)P(x, dy) = \pi(dy)P(y, dx), \quad \textit{for all } x, y \in \mathcal{X}. \tag{3.24}$$

Definition 3.5.2 *A Markov chain transition kernel P is said to be **π-irreducible** if for any $x \in \mathcal{X}$ and any set A, $A \in \mathcal{B}(\mathcal{X})$ with $\pi(A) > 0$, there exists an n that satisfies:*

$$P^n(x, A) > 0.$$

If (X_n) is a Markov chain associated with a π-irreducible transition kernel, then the chain can be qualified as a π-irreducible Markov chain.

If the chain is π-irreducible, it means for all the points in the considered state space, there is a positive probability that we may reach any set which has positive π-probability. This definition assures that those sets that are π-positive are always reachable by the chain with positive probability, independently of the initial point, as a result, the chain cannot be reduced into separated pieces.

Another important property of Markov chains is the *periodicity*.

Definition 3.5.3 *If for a π-irreducible Markov transition kernel P, there exists an integer d and a collection of disjoint sets $A_1, \ldots, A_d \in \mathcal{B}(\mathcal{X})$ such that for each $x \in A_j$, $P(x, A_j) = 1$ for $j = 1, \ldots, d-1$, and for each $x \in A_d$, $P(x, A_1) = 1$, then P is said to be **periodic**. Otherwise, P is said to be **aperiodic**.*

In other words, if the state space can be partitioned in such ways that the cyclic behavior is introduced, then P is periodic. Otherwise, if such partition does not exist, then P is aperiodic. For countable space, being aperiodic indicates that the return to a visited state may occur at irregular times for the chain. In similar ways, if a Markov chain (X_n) is associated with a periodic (resp. aperiodic) transition kernel, then the chain is called periodic (resp. aperiodic).

Definition 3.5.4 *If (X_n) is a π-irreducible Markov chain with π the stationary distribution, then X is defined to be **recurrent** if for any set Λ with $\pi(\Lambda) > 0$:*

$$Pr(X_n \in A i.o. \mid X_1 = x) > 0 \quad \textit{for all } x,$$
$$Pr(X_n \in A i.o. \mid X_1 = x) = 1 \quad \textit{for } \pi\textit{-almost all } x.$$

The chain is called **Harris recurrent** if $Pr(X_n \in A\text{i.o.} \mid X_1 = x) = 1$ for all x.

For countable space, a π-irreducible Markov chain is defined as *recurrent* if the probability of returning to a visited state is 1. Additionally, if $\pi(\cdot)$ is a probability distribution, then the chain is *positive recurrent*.

If a Markov chain is π-irreducible, aperiodic and positive Harris recurrent, then it is **Harris ergodic**.

However, in practice, it is often impossible to draw samples directly from $\pi(\cdot)$ which motivates the use of MCMC. Consequently, it is important to state same conditions under which the chain can be described as "converged" to the desired stationary distribution :

$$\|P^n(x, \cdot) - \pi(\cdot)\| := \sup_{A \in \mathcal{B}} |P^n(x, A) - \pi(A)|.$$

Proposition 3.5.1 *Suppose that a Markov chain transition kernel P is π-irreducible, which has an invariant distribution $\pi(\cdot)$. Then P is positive recurrent and π is the unique invariant distribution of P. If P is an aperiodic as well, then for π-almost all x,*

$$\|P^n(x, \cdot) - \pi(\cdot)\| \to 0 \quad \textit{as } n \to \infty.$$

In the case that P is Harris recurrent, the convergence occurs for all x.

The proof of proposition 3.5.1 is provided by Athreya et al. (1996).

Markov Chain Monte Carlo methods aim to construct a Markov chain with random samples which has a unique stationary distribution as its posterior distribution to estimate certain characteristics of the parameter's real distribution. To be assured that the estimate of the integration converges to the right value as the number of samples approaches infinity, the Strong Law of Large Numbers is needed. That is to say, we are under the assumption that the random variables should be sampled

independently from the same distribution. If a Markov chain has converged to a stationary distribution, we may assure that the samples are produced from the same distribution, but they are still correlated. The property of ergodicity makes the SLLN hold also for correlated samples.

Theorem 3.5.2 *(Strong Law of Large Numbers)*
If (X_n) is a Harris ergodic Markov chain on a state space $\mathcal{X} \subset \mathbb{R}^d$ and has an invariant distribution $\pi(\cdot)$, then $\forall f$ verifies $\int_{\mathcal{X}} |f(x)|\pi(x)dx < \infty$,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} f(X_i) = \int_{\mathcal{X}} f(x)\pi(x)dx \quad a.s. \tag{3.25}$$

for any initial distribution.

However, to assure that the Central Limit Theorem (CLT) holds for a Markov chain, some stronger regularity conditions are needed. More precisely, the chain is required to be **geometrically ergodic** or **uniformly ergodic**, which depends on the moment condition.

Definition 3.5.5 *If (X_n) is a Harris ergodic Markov chain which has an invariant distribution $\pi(\cdot)$. The chain is defined as **geometrically ergodic** if there exists a constant $0 < t < 1$ and a function $M : X \mapsto \mathbb{R}^+$ such that for any $x \in \mathcal{X}$:*

$$\|P^n(x, \cdot) - \pi(\cdot)\| \leq M(x)t^n$$

*for $n \in \mathbb{N}$. If there is a bounded M which satisfies this condition, then (X_n) is **uniformly ergodic**.*

Here we state three different sets of conditionals for Markov chain CLT.

Theorem 3.5.3 *Suppose (X_n) is a Harris ergodic Markov chain on $\mathcal{X}$ which has an invariant distribution $\pi(\cdot)$. Let $f : X \mapsto \mathbb{R}$ be a Borel function. If one of the following conditions is assumed:*

1. *(Doukhan et al. 1994) (X_n) is geometrically ergodic, and $\mathbb{E}_\pi[f^2(X)(log^+|f(X)|)] < \infty$;*
2. *(Roberts and Rosenthal 1997) (X_n) is geometrically ergodic and reversible, and $\mathbb{E}_\pi[f^2(X)] < \infty$;*
3. *(Ibragimov and Linnik, 1971) (X_n) is uniformly ergodic and $\mathbb{E}_\pi[f^2(X)] < \infty$.*

Then, for any initial distribution, as $n \to \infty$, and assuming $X_1 \sim \pi(\cdot)$:

$$\sqrt{n}(\mathbb{E}_\pi[f(X)] - \bar{f}_n) \xrightarrow{d} \mathcal{N}(0, \sigma_f^2)$$

where

$$\sigma_f^2 = \mathrm{Var}_\pi(f(X_1)) + 2\sum_{i=2}^{\infty} \mathrm{Cov}_\pi(f(X_1), f(X_i)). \tag{3.26}$$

Further discussions concerning the convergence issues related to MCMC based methods can be found in Ibragimov and Linnik (1971); Rosenthal (1995); Robert et al. (1999); Robert and Casella (1999); Jones and Hobert (2001) and Roberts and Tweedie (2008).

CHAPTER 4

Numerical Approaches for Bayesian Estimation

BAYESIAN inference, formal and informal, has found widespread applications. Although Monte Carlo methods are finely suited to generate samples from the posterior distribution, they remain generally inefficient when complex models with multimodality problems are presented. This motivates the development of Markov chain Monte Carlo (MCMC) methods, which aim at visiting solutions with stable frequency stemming from a stationary distribution. In parallel, inspired from the fact that the observations arrive sequentially in time, the idea of performing inference online seems to be very attractive and convenient in real world applications, such as aircraft tracking, noisy signal identification, or stock market prediction. Based on this idea, the approaches developed are known as Sequential Monte Carlo (SMC) methods. Instead of having one simulation creating trajectories repeatedly, like MCMC-based methods do, one may imagine SMC methods as a large number of simulations evolving simultaneously and those who fit best the observations are sequentially selected.

In this chapter, simulation based methods, including both the MCMC-based methods and the SMC methods are investigated using the sampling and integration techniques presented in Section 3.3. Three MCMC-based algorithms are investigated and some important issues often raised in implementation are subsequently addressed. The aim is to provide some options to the parameterization problem of dynamic state-space system in plant growth modelling. We are particularly convinced that with proper estimation methods, even for misspecified models the impact could be limited and contextual.

Note that in this chapter, we only investigate the Bayesian numerical methods for functional parameter estimation while considering the noise parameters of the state-space model known. The issues related to noise parameters estimation and uncertainty assessment are further addressed in Chapter 6.

The chapter is organized as follows: in the first part, MCMC-based methods which

are known as *offline* Bayesian estimation methods are introduced. The two most famous MCMC-based algorithms, Metropolis-Hastings (MH) and Gibbs sampler (GS) are presented. They can be employed as basic tools to construct ergodic Markov chains. Then, some latest MH and GS based algorithms, involving techniques like "one long run" or "many short runs" are detailed. In order to meet the demand of parameter estimation purpose in state-space plant growth models, an adaptive scheme is proposed in this thesis. In the second part, some SMC methods which represent *online* Bayesian estimation methods are reviewed. The general filtering recursion framework is revisited, followed by a brief presentation of Kalman based filtering methods. Sequential Important Sampling and Resampling algorithms are reviewed and two recent particle filtering based algorithms Regularized Particle Filter and Convolution Particle Filter are presented. They will be further studied in comparison with the MCMC based methods when applied to the LNAS plant growth model in Chapter 8.

4.1 MCMC-based algorithms

In a short period of time, the Markov chain Monte Carlo (MCMC) integration methods, especially the Metropolis-Hastings algorithm Metropolis et al. (1953); Hastings (1970) and the Gibbs sampling algorithm (Geman and Geman, 1984; Gelfand et al., 1990) have emerged as popular tools in the field of Bayesian analysis to perform analysis for complex statistical models. Appropriately defined and implemented, the MCMC-based methods allow us to evaluate complex and high-dimensional integrals with the purpose of obtaining posterior distributions for unknown parameter, missing data and hidden variables, by sampling values from an ergodic Markov chain.

Excellent tutorial concerning the implementation and the methodology is provided by Robert and Casella (1999) and Robert and Casella (2010).

4.1.1 Metropolis-Hastings algorithm

Based on the accept-reject methodology, the Metropolis-Hastings algorithm was first introduced by Hastings (1970) as a generalized version of the Metropolis algorithm, described in Metropolis et al. (1953). It offers a way to obtain a sequence of samples from a probability distribution for which direct sampling is infeasible. The resulting sequence can be used to approximate the target distribution or to compute an integral. For each iteration of the algorithm, a new candidate point is drawn from a proposal distribution, and the decision is made based on the acceptance probability computed by taking into account both the actual candidate and the last accepted candidate.

Suppose that π_u defines the un-normalized target density on a state space $\mathcal{X} \subset \mathbb{R}^d$ as in (3.6), with respect to the d-dimensional Lebesgue measure. Suppose also that it is possible to sample from a Markov chain with transition density $q(\cdot|x)$, namely the *proposal distribution* or the *instrumental distribution*, with respect to the d-dimensional Lebesgue measure. The Metropolis-Hastings algorithm associated with the target distribution and the proposal distribution produces a Markov chain in the following way:

Algorithm : Metropolis-Hastings

- **Initialization**
- Choose initial values for X_0.
- **Iteration**
- For $i = 1, \ldots, n$:
  1. Sample $Z_i \sim q(\cdot|X_{i-1})$.
  2. Set

$$X_i = \begin{cases} Z_i & \text{with probability } \alpha(X_{i-1}, Z_i), \\ X_{i-1} & \text{with probability } 1 - \alpha(X_{i-1}, Z_i), \end{cases} \tag{4.1}$$

  where

$$\alpha(x, z) = \min\left(1, \frac{\pi_u(z)}{\pi_u(x)} \frac{q(x|z)}{q(z|x)}\right). \tag{4.2}$$

In the case that $\pi_u(x)q(z|x) = 0$, to avoid ambiguity, we set $\alpha(x, z) = 0$ (Roberts and Rosenthal, 2004).

Remark that since only the ratio of densities $\pi_u(z)/\pi_u(x)$ is required, the normalizing constant $\int_{\mathcal{X}} \pi_u(x)dx$ does not need to be computed, which alleviates the normalization problem described in Section 3.2.1.

An important remark is that although $q(\cdot|x)$ does not need to be related to our target density π_u, the choice of this proposal density may influence crucially the efficiency of the method. There exist many possible ways to choose the proposal density, for instance:

Independence sampler
When the instrumental density q does not depend on the previous accepted candidate X_{i-1}, i.e. $q(z|x) = q(z)$, the acceptance probability can be written as :

$$\alpha(x, z) = \min\left(1, \frac{\pi_u(z)}{\pi_u(x)} \frac{q(x)}{q(z)}\right), \tag{4.3}$$

where $\pi_u(z)/q(z)$ can be regarded as the importance weight function used in importance sampling. Indeed, the basic ideas of independent sampler and importance sampler (Section 3.3.2) are quite similar, both of them attribute massive probability to points with large weights, however in different ways: the importance sampler selects those points more often than usual, whereas the independence sampler remains at those points for a long time.

Symmetric Random walk Metropolis Algorithm (RWM)
Another choice for the instrumental law is random walk with $q(z|x) = q_{RW}(z - x)$

and we assume that $q_{RW}(w) = q_{RW}(-w)$. Therefore the acceptance probability can be simplified:

$$\alpha(x, z) = \min\left(1, \frac{\pi_u(z)}{\pi_u(x)}\right). \tag{4.4}$$

All candidates that improve π_u are accepted. By applying the random walk-type of proposal distribution, we are supposed to approach the zone containing the mode of π_u more often than to move away from it (see Figure 4.1).

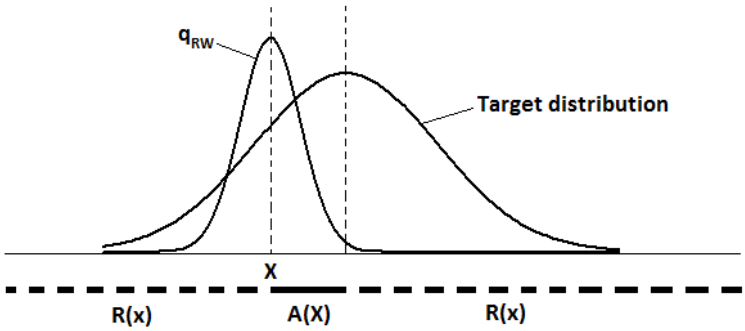


Figure 4.1: The region of acceptance (A(x)) and the region of rejection (R(x)) for the random walk proposal (q_{RW}) in Metropolis Hastings algorithm.

An advantage of RWM is that the computation of π_u is only involved in the accept-reject mechanism, hence the algorithm is relatively easy to implement, which is probably why it is considered as one of the most popular MCMC-based methods.

There are many possible choices for q_{RW}, including uniform distributions on the unit circle, Student's t-distributions, or multivariate normal distributions, and the choices of q_{RW} is crucial to determine the performance of the method (the convergence speed, the precision of the estimator, etc.). A classical choice is the multivariate normal distribution with zero-mean and covariance matrix Σ. It is well known that either too small or too large variances for the proposal distribution of the random walk will cause highly correlated Markov chains and therefore result in imprecise estimates. For instance, if the variance is too small, then almost all the proposed values are accepted. As a consequence, the difference between the two accepted values are often very small which leads to the slow exploration of the state space. Conversely, if the variance is too large, the proposals are often rejected and the chain has to stay for a long time at the same point. Thus, to tune the covariance matrix Σ of the proposal distribution to a proper scale is absolutely essential for a successful implementation of RWM.

Note that to illustrate the performance of the method, the acceptance rate could be monitored, as illustrated by Figure 4.2. When the acceptance rate is too large, it indicates that the chain may move too slowly on the surface of the model space f, except in the case when f is almost flat, for which a high acceptance rate is acceptable, otherwise, it is highly possible that some parts of the domain can be missed. On the other hand, if the acceptance rate is too low, it suggests that the chain may move too fast and have missed an isolated mode of f.

In the meantime, another variable, the autocovariance can as well be used to monitor the convergence and the mixing of the chain. The autocovariance is defined as the covariance between equidistant states in a chain which reaches its real-valued stationary distribution. The term $\mathrm{Cov}_\pi(f(X_i), f(X_{i+k}))$ presented in Equation (3.26) of Theorem 3.5.3 is known as the *lag-k autocovariance* of the function f, and accordingly $\frac{\mathrm{Cov}_\pi(f(X_i),f(X_{i+k}))}{\mathrm{Var}_\pi(f(X_i))}$ is called the *lag-k autocorrelation* of the function f.

In the following toy example given in Robert and Casella (2010), three choices of δ, $\delta \in \mathbb{R}$ for the random walk proposal distribution are made to estimate the mean of a standard normal distribution $\mathcal{N}(0, 1)$.

$$Y_i = X_{i-1} + \epsilon_i, \quad \epsilon_i \sim \mathcal{U}[-\delta, \delta]$$

All the three chains are monitored by both the acceptance rate and the autocorrelation function.

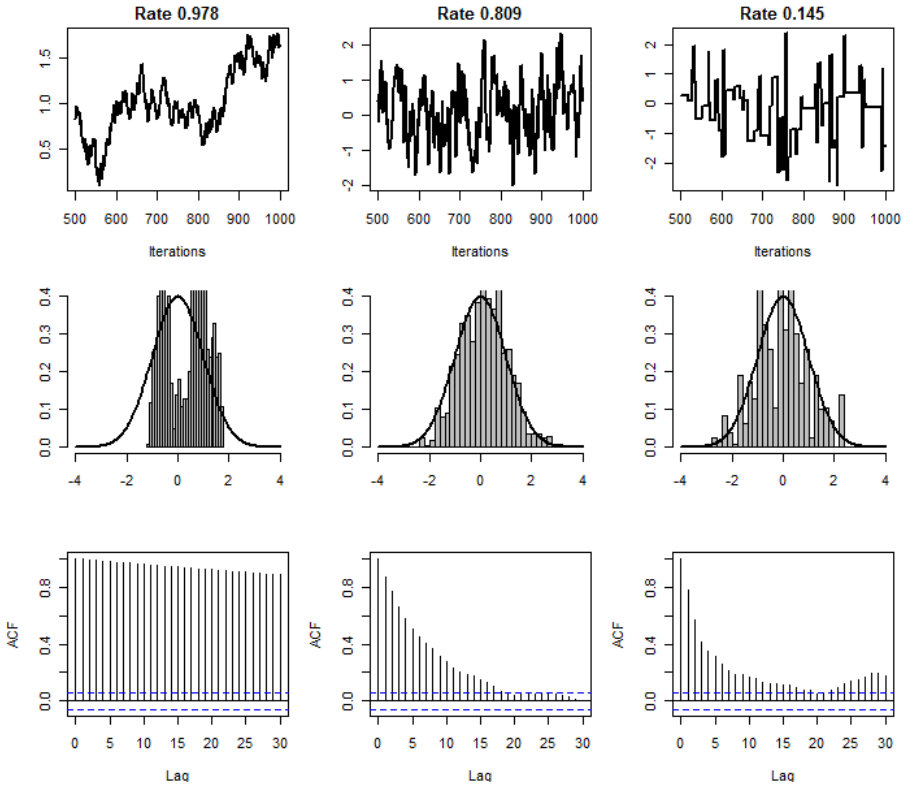


Figure 4.2: Three different proposals and resulting chains. The evolution of the estimates, the histogram representing the distribution of the estimates and the autocorrelation function are presented for each choice of δ. Left panel: $\delta = 0.1$. Middle panel: $\delta = 1$. Right panel: $\delta = 10$.

As demonstrated in Figure 4.2, the first choice $\delta = 0.1$ yields in significant autocorrelations, which indicates that the chain has not converged after 5000 iterations. The acceptance rate is extremely high, which confirms the drawback of too small steps of the RWM formerly stated. In the case of $\delta = 10$, the autocorrelations are less remarkable, but the chain still converges more slowly than the second case ($\delta = 1$). Hence,

in this particular example, $\delta = 1$ with the median acceptance rate is proved to be the most efficient among the three. The autocorrelations decay to nearly zero. However, we also notice that $\delta = 10$ which gives the lowest acceptance rate performs better than a high acceptance rate provided by $\delta = 0.1$.

It is demonstrated that in the case of Random Walk MH with multivariate normal proposal, and a large dimensional context ($d \to \infty$), the optimum is achieved when the overall acceptance rate (without the burn-in part) of the chain is close to 0.234 (Roberts et al.; Roberts and Rosenthal; Bédard, 2008), and the variance of the proposal distribution is $(2.38^2/d)\Sigma_\pi$ (Gelman et al., 1996), with d the dimension of $\mathcal{X}$ and Σ_π the covariance matrix of the target distribution. To estimate Σ_π, a few adaptive algorithms have been developed in the last two decades, the most famous one is the Adaptive Metropolis-Hastings proposed by Haario et al. (1999, 2001).

Adaptive Metropolis

To avoid the time-consuming manual tuning of the proposal distribution, meanwhile to optimize the choice of Gaussian RWM proposals aiming at a more rapid convergence, Haario et al. (2001) have proposed the *Adaptive Metropolis* to update continuously Σ_π (learn on the fly). Inspired by the Robbins-Monro update scheme (Robbins and Monro, 1951), the stochastic approximation framework is proposed as follows :

Algorithm : Adaptive Metropolis

- **Initialization**
- Choose initial values for X_0, μ_0 and Σ_0.
- **Iteration**
- For $i = 1, \ldots, n$:

1. Sample $Z_i \sim \mathcal{N}(X_{i-1}, \lambda\Sigma_{i-1})$, with $\lambda = 2.38^2/d$.
2. Set
$$X_i = \begin{cases} Z_i & \text{with probability } \alpha(X_{i-1}, Z_i), \\ X_{i-1} & \text{with probability } 1 - \alpha(X_{i-1}, Z_i), \end{cases}$$
where $\alpha(x, z)$ is given by (4.2).
3. Update
$$\begin{aligned} \mu_i &= \mu_{i-1} + \gamma_i(X_i - \mu_{i-1}), \\ \Sigma_i &= \Sigma_{i-1} + \gamma_i((X_i - \mu_{i-1})(X_i - \mu_{i-1})^T - \Sigma_{i-1}). \end{aligned} \tag{4.5}$$

In this algorithm, γ_i is a deterministic, non-increasing sequence of stepsizes which satisfies $\sum_{i=1}^{\infty} \gamma_i = \infty$ and $\sum_{i=1}^{\infty} \gamma_i^{1+\nu} < \infty$ for some $\nu > 0$. The objective is to make sure that the variations introduced by the adaptive scheme vanish as the iteration

$i \to \infty$, so that π-ergodicity of the chain is guaranteed. One classical choice is $1/i$, with i denoting the current iteration number.

In literature, many adaptive schemes have been proposed which appear to enhance significantly the acceptance rate of the MCMC based methods. However, prudence must be taken to make sure that the adaptation does not damage the detailed balance and the overall ergodicity of the chain, for it is obvious that the adaptive scheme has modified the nature of the chain, the process is no longer Markovian because of the dependence on the history of the chain. Only in some cases, it can be proved that the algorithm still has appropriate ergodicity properties which ensure the correct generation of samples from the target distribution. For further ergodicity investigation, please refer to Haario et al. (2006).

Moreover, there may still remain some difficulties during the exploration of the target distribution. In the particular plant growth models studied in this thesis, we are facing multi-dimensional vectors with various scales. When the initial values of Σ_π are poorly chosen, in a way that $\lambda\Sigma_{i-1}$ can be too large in some directions or too small in all the directions, it results in poor exploration of the target distribution due to the localised proposals. This motivates the introduction of a non-increasing function for the scaling parameter λ in a global adaptive scaling framework, which initially takes the value of $2.38^2/d$ (Andrieu and Thoms, 2008).

Algorithm : Adaptive Metropolis with global multivariate adaptive scaling

---

- **Initialization**

- Choose initial values for X_0, μ_0, Σ_0 and λ_0.

- **Iteration**

- For $i = 1, \ldots, n$:

  1. Sample $Z_i \sim \mathcal{N}(X_{i-1}, \lambda_{i-1}\Sigma_{i-1})$.
  2. Set
  $$X_i = \begin{cases} Z_i & \text{with probability } \alpha(X_{i-1}, Z_i) \\ X_{i-1} & \text{with probability } 1 - \alpha(X_{i-1}, Z_i), \end{cases}$$
  where $\alpha(x, z)$ is given by (4.2).
  3. Update μ_i and Σ_i as in (4.5).
  4. Update λ_i :
  $$\log \lambda_i = \log \lambda_{i-1} + \gamma_i \left(\alpha(X_{i-1}, Z_i) - \alpha_*\right), \tag{4.6}$$
  with α_* the target acceptance rate.

Note that this algorithm is proved to be quite useful in the early stages of the computation. It improves the performance of Adaptive Metropolis in practice. Nevertheless, it is unlikely to make significant difference in a long run. The interest lies in the case of poor chosen Σ_0 which can be translated into either too small or too large acceptance probability.

It is shown that this algorithm can be further improved in the case when we customize λ, $\lambda \in \mathbb{R}$ for each component (parameter) of the state space $\mathcal{X}$ when updating the chain in an univariate fashion. To achieve this goal, we first revisit the Gibbs Sampler.

4.1.2 Gibbs Sampler

Introduced by Geman and Geman (1984) and developed by Gelfand et al. (1990), the Gibbs Sampler can be seen as a special case of MH transition (Robert et al., 1999) which decomposes a multi-dimensional simulation into a collection of single dimensional ones. For each step of the algorithm, only a single component of the state of the chain is updated, by keeping the others fixed and drawing form the conditional distribution of the component given the others. Suppose that the i-th component of x is to be updated, x^{-i} denotes the complementary components, a proposal distribution is thus chosen:

$$q(z|x) = \begin{cases} \pi(z^{(i)}|x^{(-i)}) & \text{if } x^{(-i)} = z^{(-i)}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

Theorem 4.1.1 *The Gibbs sampling method is equivalent to the composition of p Metropolis-Hastings algorithms, with acceptance probabilities equal to 1.*

Proof
The resulting acceptance probability is:

$$\begin{aligned} \alpha(x,z) &= \frac{\pi(z)q(x|z)}{\pi(x)q(z|x)} = \frac{\pi(z)/\pi(z^{(i)}|x^{(-i)})}{\pi(x)/\pi(x^{(i)}|z^{(-i)})} \\ &= \frac{\pi(z)/\pi(z^{(i)}|z^{(-i)})}{\pi(x)/\pi(x^{(i)}|x^{(-i)})} && \text{since } x^{(-i)} = z^{(-i)} \\ &= \frac{\pi(z^{(-i)})}{\pi(x^{(-i)})} = 1. \end{aligned}$$

□

To explain this proof, one can easily imagine that the only possible jump is the proposal z that matches x on all components other than the i-th, and it is accepted automatically.

Now suppose that $\pi_u(\cdot)$ is a d-dimensional un-normalised density on a state space $\mathcal{X}$ which is an open subset of $\mathbb{R}^d$, with $X = (X^{(1)}, \ldots, X^{(d)})$, the sampling proceeds by systematically updating each component in turn based on univariate *full conditional* distributions $\pi_1, \ldots, \pi_d$. For $j = 1, 2, \ldots, d$:

$$X^{(j)} \mid x^{(1)}, \ldots, x^{(j-1)}, x^{(j+1)}, \ldots, x^{(d)} \sim \pi_j(x^{(j)} \mid x^{(1)}, \ldots, x^{(j-1)}, x^{(j+1)}, \ldots, x^{(d)}).$$

Therefore, to update from X_{i-1} to X_i, the associated Gibbs sampling algorithm can be described as follows :

Algorithm : Gibbs Sampling

- Given $X_{i-1} = (x_{i-1}^{(1)}, \ldots, x_{i-1}^{(d)})$.
- Sample
  1. $X_i^{(1)} \sim \pi_1(x^{(1)} \mid x_{i-1}^{(1)}, \ldots, x_{i-1}^{(d)})$,
  2. $X_i^{(2)} \sim \pi_2(x^{(2)} \mid x_i^{(1)}, x_{i-1}^{(3)}, \ldots, x_{i-1}^{(d)})$,

  $\ldots$,

  d. $X_i^{(d)} \sim \pi_d(x^{(d)} \mid x_i^{(1)}, \ldots, x_i^{(d-1)})$.

Note that although the Gibbs sampler is considered as a special case of MH, it has some features that can be distinguished from MH and need to be taken care of during implementation:

- Since all the proposed values are accepted, the acceptance rate is equal to 1, which indicates that the optimal acceptance rates cannot be applied in this setting. The convergence criteria are therefore different from those of MH.
- The fact that the choice of instrumental distribution is limited implies that the prior knowledge and some analytical properties of $\pi_u(\cdot)$ are required when implementing the Gibbs sampler.
- If the transition kernel associated with the resulting chain is absolutely continuous with respect to the dominating measure, then the chain is irreducible, aperiodic and therefore ergodic. Detailed proof can be found in (Robert et al., 1999).

Thanks to Gibbs sampler, we are able to have the choice to sample from high dimensional distribution using lower (one) dimensional conditional distributions. Motivated by the fact that some of the conditional distributions might be difficult or even impossible to sample from, it gives rise to an extremely convenient and widely used algorithm known as *Metropolis-within-Gibbs* (Carter and Kohn, 1994), or *Hastings-within-Gibbs*. The idea is to hybrid the Metropolis Hastings and the Gibbs sampler, replacing some sampling steps from the conditional distribution $\pi_i(\cdot \mid x^{(-i)})$ in Gibbs sampler by a Metropolis step.

4.1.3 Metropolis-within-Gibbs algorithm

In this thesis, beside improving the performance of adaptive Metropolis, another motivation to employ the Metropolis-within-Gibbs algorithm comes from the fact that in state-space plant growth models, we are often interested in some hidden state variables other than the unknown parameters (also known as *latent variables*) (see Equation (3.1)), such as biomass of the root compartment for yield prediction. Even if the marginal density of the parameter vector Θ is our sole interest, since $\pi(\Theta|y_{1:t_{max}})$ are often analytically intractable or difficult to evaluate, the hidden state variables are often introduced to provide an analytical expression, and subsequently make the numerical implementation easier. Hence, a joint estimation of both unknown parameters and hidden states is required. Intuitively, given the target $p(\Theta, x_{1:t_{max}}|y_{1:t_{max}})$, we define the generic joint updates' scheme:

– Update $\Theta \sim \pi(\theta|x_{1:t_{max}}, y_{1:t_{max}})$.
– Update $X_{1:t_{max}} \sim \pi(x_{1:t_{max}}|\Theta, y_{1:t_{max}})$.

However, to obtain the full set of conditional distributions in order to update such time dependent variables simultaneously can be a troublesome task. Being able to break the hidden state vector into unidimensional (or lower dimensional) components can largely facilitate the update process of these hidden state variables.

Under these circumstances, the Metropolis-within-Gibbs appears to provide an appropriate framework, since it allows us not only to reduce the dimensionality of the parameter vector, separate and alternate the estimation of unknown parameters $\Theta = \{\theta^{(1)}, \ldots, \theta^{(d)}\}, d \in \mathbb{N}^*$ and the hidden state variables $X = \{x^{(1)}, \ldots, x^{(t_{max})}\}, t_{max} \in \mathbb{N}^*$, but also to simplify the update process by replacing the sampling step of Gibbs Sampling by a Metropolis-Hastings step. In this way, we no longer require to sample directly from the full set of conditional distributions which can be hard or impossible to obtain.

In the following, the adaptive Metropolis-within-Gibbs algorithm for state-space models is presented. It can be used to perform a numerical approximation by simulating an ergodic Markov chain $\{\theta_i^{(1:d)}, x_i^{(1:t_{max})}\}$ which admits $\pi(\Theta, x_{1:t_{max}}|y_{1:t_{max}})$ as its stationary distribution.

Algorithm : Adaptive Metropolis-within-Gibbs

- **Initialization**
- Choose initial values for $X_0 = (\theta_0^{(1)}, \ldots, \theta_0^{(d)}, x_0^{(1)}, \ldots, x_0^{(t_{max})})$.
- **Iteration**
- For $i = 1, \ldots, n$:
  - **Parameters**
  - For $j = 1, \ldots, d$:
    1. Sample $Z_i^{(j)} \sim q_j(\theta^{(j)} \mid \theta_i^{(1)}, \ldots, \theta_i^{(j-1)}, \theta_{i-1}^{(j+1)}, \ldots, \theta_{i-1}^{(d)}, x_{i-1}^{(1:t_{max})})$.
    2. Set
$$\theta_i^{(j)} = \begin{cases} Z_i^{(j)} & \text{with probability } \alpha_j(\theta_{i-1}^{(j)}, Z_i^{(j)}), \\ \theta_{i-1}^{(j)} & \text{with probability } 1 - \alpha_j(\theta_{i-1}^{(j)}, Z_i^{(j)}), \end{cases}$$
    where
$$\alpha_j(x, z) = \min\left(1, \frac{\pi_j(z)}{\pi_j(x)} \frac{q_j(x \mid z)}{q_j(z \mid x)}\right). \tag{4.8}$$
  - **Hidden states**
  - For $t = 1, \ldots, t_{max}$:
    1. Sample $Z_i^{(d+t)} \sim q_t(x^{(t)} \mid \theta_i^{(1:d)}, x_i^{(1)}, \ldots, x_i^{(t-1)}, x_{i-1}^{(t+1)}, \ldots, x_{i-1}^{(t_{max})})$.
    2. Set
$$x_i^{(t)} = \begin{cases} Z_i^{(d+t)} & \text{with probability } \alpha_{d+t}(x_{i-1}^{(t)}, Z_i^{(d+t)}), \\ x_{i-1}^{(t)} & \text{with probability } 1 - \alpha_{d+t}(x_{i-1}^{(t)}, Z_i^{(d+t)}), \end{cases}$$
    where $\alpha_{d+t}(x, z)$ can be obtained by (4.8).

Note that when the posterior distribution (target distribution) can be expressed in its analytical form, we need to determine if it is better to sample from the target distribution individually or in groups. The latter is known as *block updating*, in contrast to *componentwise sampling*. It refers to splitting a multivariate vector into groups and being sampled separately. Parameters which are more correlated should be grouped into the same block to retain their correlations instead of ignoring them as is the case in componentwise sampling. Moreover, there exist other advantages of the block updating, such that different sampling strategies can be used for each block, and that large dimensional covariance matrices can be reduced, which can be helpful for high dimensional models.

In this algorithm, the block update is adopted for the parameter Θ in which all of its components are updated simultaneously by using a multivariate proposal distribution. However, we may also choose to implement Adaptive Metropolis with componentwise updating. With such an update scheme, one $\lambda_i := (\lambda_i^1, \ldots, \lambda_i^d)$ for each parameter can be defined at iteration i, to tune the random walk step so as to speed up convergence. This approach is known as *Adaptive Metropolis with componentwise adaptive scaling* (Andrieu and Thoms, 2008).

As for the hidden states' estimation, in this algorithm, we opt for a componentwise update. The advantage of this choice is that it is very easy to implement. A convenient choice of proposal distribution is the model transition density.

In the next section, further attention is paid to the joint update scheme of the parameters and the hidden state variables. In the case of plant growth models, some important issues may arise during the implementation regarding the mixing properties of the chain which may affect the efficiency of the estimates in a significant way. An adapted algorithm is thus proposed based on the previous Adaptive Metropolis-within-Gibbs algorithm.

4.1.4 Adapted Metropolis-within-Gibbs

Mixing property

The conditional joint updates' scheme enables joint estimation of both the parameter vector and the hidden state variables. As formerly mentioned, the natural approach is to update the parameter conditionally on the current value of the estimated hidden state trajectory. Yet the overall efficiency of the algorithm can be sabotaged if there is a strong correlation between the parameter vector Θ and the hidden state vector X (Liu et al., 1994; Roberts and Sahu, 1997), which is obviously an undesirable feature. In the plant growth models that we study, such a problem is frequently encountered. This model configuration known as *centered parameterisation* (Fearnhead, 2011), for which :

$$p(\Theta|x_{1:t_{max}}, y_{1:t_{max}}) \approx p(\Theta|x_{1:t_{max}}).$$

It is shown by (Roberts and Sahu, 1997) that for $f \in L^2$, the *Bayesian fraction of missing information* is defined by

$$\gamma_f = 1 - \frac{\mathbb{E}(\mathrm{Var}(f(\Theta)|x_{1:t_{max}}, y_{1:t_{max}})|y_{1:t_{max}})}{\mathrm{Var}(f(\Theta)|y_{1:t_{max}})}. \tag{4.9}$$

The convergence rate is thus $\sup_f \gamma_f$. When the value of $\gamma \approx 1$, it implies a poor mixing between Θ and X. It is due to the fact that the most of the variation of $f(\Theta)$ is explained by $x_{1:t_{max}}$ conditioned by the data $y_{1:t_{max}}$. Further investigation of this problem, particularly the convergence rate of such algorithms can be found in Liu (1994) and Roberts and Sahu (1997).

To understand the impact of strong correlation on the mixing property of the chain, one may imagine that large moves of Θ are more likely to be rejected since it is inconsistent with the value of $x_{1:t_{max}}$. One way to alleviate this is to consider the joint update of $(\Theta, x_{1:t_{max}})$ with a proposal distribution under the form (Fearnhead, 2011) :

$$q(\Theta', x'_{1:t_{max}}|\Theta, x_{1:t_{max}}) = q(\Theta'|\Theta)q(x'_{1:t_{max}}|\Theta'),$$

where $(\Theta', x'_{1:t_{max}})$ denotes the proposal. Note that $q(\Theta'|\Theta)$ allows to propose moves of reasonable sizes, and the proposed hidden states being consistent with Θ'. Indeed, the choice of the proposal for the hidden states is crucial. If the exploration of the state space $\mathcal{X}$ is not guided, the proposed trajectory can result in a very poor likelihood evaluation (a trajectory unlikely to happen in reality). In order to go through the state space in a more informative way, ideally, the proposal should take into account the observations $y_{1:t_{max}}$ and could be chosen as :

$$q(x'_{1:t_{max}}|\Theta') = p(x'_{1:t_{max}}|\Theta', y_{1:t_{max}}).$$

Nonetheless, such a proposal for the hidden states would necessitate computing and simulating from $\pi(x'_{1:t_{max}}|\Theta', y_{1:t_{max}})$, which is impossible for most real applications. When the density cannot be computed explicitly or if it is complicated to simulate from, $p(x'_{1:t_{max}}|\Theta')$ can be used as a reasonable compromise, which simplifies the computation in a considerable way and is very easy to simulate from. Thus, the acceptance probability α can be simplified and this value does not need to be evaluated:

$$\begin{aligned}
&\alpha(\{\Theta, x_{1:t_{max}}\}, \{\Theta', x'_{1:t_{max}}\}) \\
&= \min\left(1, \frac{\pi(\Theta', x'_{1:t_{max}}|y_{1:t_{max}})q(\Theta, x_{1:t_{max}}|\Theta', x'_{1:t_{max}})}{\pi(\Theta, x_{1:t_{max}}|y_{1:t_{max}})q(\Theta', x'_{1:t_{max}}|\Theta, x_{1:t_{max}})}\right) \\
&= \min\left(1, \frac{p(\Theta', x'_{1:t_{max}}|y_{1:t_{max}})}{p(\Theta, x_{1:t_{max}}|y_{1:t_{max}})} \frac{q(\Theta|\Theta')q(x_{1:t_{max}}|\Theta)}{q(\Theta'|\Theta)q(x'_{1:t_{max}}|\Theta')}\right) \\
&= \min\left(1, \frac{p(y_{1:t_{max}}|\Theta', x'_{1:t_{max}})p(x'_{1:t_{max}}|\Theta')p(\Theta')}{p(y_{1:t_{max}}|\Theta, x_{1:t_{max}})p(x_{1:t_{max}}|\Theta)p(\Theta)} \frac{q(\Theta|\Theta')p(x_{1:t_{max}}|\Theta)}{q(\Theta'|\Theta)p(x'_{1:t_{max}}|\Theta')}\right) \\
&= \min\left(1, \frac{p(y_{1:t_{max}}|\Theta', x'_{1:t_{max}})p(\Theta')}{p(y_{1:t_{max}}|\Theta, x_{1:t_{max}})p(\Theta)} \frac{q(\Theta|\Theta')}{q(\Theta'|\Theta)}\right).
\end{aligned} \tag{4.10}$$

Compared to the algorithms presented in the literature, such as the Monte Carlo within Metropolis and Grouped Independence MH (Beaumont, 2003), in this thesis, we propose to simplify the step of simulating a large number of X' from $p(x_{1:t_{max}}|\Theta')$ by simulating only once the hidden state variables. Thus, the efficiency of the resulting algorithm is based on the efficiency of the proposal distribution of Θ, as well as the closeness of $q(x'_{1:t_{max}}|\Theta')$ to $p(x'_{1:t_{max}}|\Theta', y_{1:t_{max}})$. When the modelling noises are relatively small, the proposed algorithm helps to reduce considerably the computational cost.

Furthermore, with the purpose of improving the estimation of the hidden states, we propose to perform additional Metropolis-within-Gibbs steps simulating X' from $p(x'_{1:t_{max}}|\Theta)$ repeatedly once the functional parameters have converged. This also allows to improve the estimation of the modelling noise parameters, the implementation details of which are described in Chapter 6.

We also remark that if Θ has a low dimension ($d \leq 6$ in our test case), a multivariate proposal may be preferred since its computational cost is more advantageous. Therefore, the resulting algorithm can be summarized as follows:

Algorithm : Adapted Metropolis-within-Gibbs

- **Initialization**
- Choose initial values for $\Theta_0 = \{\theta_0^{(1)}, \dots, \theta_0^{(d)}\}$ and $x_0 = \{x_0^{(1)}, \dots, x_0^{(t_{max})}\}$ with $x_0 = f(\Theta_0)$, where f is the transition density of the model.
- Choose initial values for μ_0, Σ_0 and λ_0.
- **Parameters**

  Iteration
  - For $i = 1, \dots, n_1$:
    1. Sample $\theta_i^{(1:d)'} \sim \mathcal{N}(\theta_{i-1}^{(1:d)}, \lambda_{i-1}\Sigma_{i-1})$.
    2. Sample $x_i^{(1:t_{max})'} \sim p(x^{1:t_{max}}|\theta_i^{(1:d)'})$.
    3. Set

$$\{\theta_i^{(1:d)}, x_i^{(1:t_{max})}\} = \begin{cases} \{\theta_i^{(1:d)'}, x_i^{(1:t_{max})'}\} & \text{with probability } \alpha_\theta(\theta_{i-1}^{(1:d)}, \theta_i^{(1:d)'}), \\ \{\theta_{i-1}^{(1:d)}, x_{i-1}^{(1:t_{max})}\} & \text{with probability } 1 - \alpha_\theta(\theta_{i-1}^{(1:d)}, \theta_i^{(1:d)'}), \end{cases}$$

       where $\alpha_\theta(\theta, \theta')$ is given by (4.10).
    4. Update μ_i and Σ_i as in (4.5).
    5. Update λ_i as in (4.6).

  Estimator
  - For a burn-in period $n_0 < n_1$:

$$\hat{\theta}_{n_1}^{(1:d)} = \frac{1}{n_1 - n_0} \sum_{k=n_0+1}^{n_1} \theta_k^{(1:d)},$$

$$\hat{\Sigma}_{n_1} = \frac{1}{(n_1 - n_0) - 1} \sum_{k=n_0+1}^{n_1} (\theta_k^{(1:d)} - \hat{\theta}_{n_1}^{(1:d)})^T (\theta_k^{(1:d)} - \hat{\theta}_{n_1}^{(1:d)}).$$

- **Hidden states**
  - **Iteration**
  - For $j = n_1 + 1, \ldots, n_2; n_2 > n_1 + 1$:
    - For $t = 1, \ldots, t_{max}$:

      1. Sample $x_j^{(t)'} \sim q_t(x^{(t)} \mid \hat{\theta}_{n_1}^{(1:d)}, x_j^{(1)}, \ldots, x_j^{(t-1)}, x_{j-1}^{(t+1)}, \ldots, x_{j-1}^{(t_{max})})$.

      2. Set

$$x_j^{(t)} = \begin{cases} x_j^{(t)'} & \text{with probability } \alpha_{d+t}(x_{j-1}^{(t)}, x_j^{(t)'}), \\ x_{j-1}^{(t)} & \text{with probability } 1 - \alpha_{d+t}(x_{j-1}^{(t)}, x_j^{(t)'}), \end{cases}$$

         where

$$\alpha_{d+t}(x_{j-1}^{(t)}, x_j^{(t)'}) = \min\left(1, \frac{\pi(x_j^{(t)'} | x_j^{(1:t-1)}, x_{j-1}^{(t+1:t_{max})}, \hat{\theta}_{n_1}^{(1:d)}, y^{1:t_{max}})}{\pi(x_{j-1}^{(t)} | x_j^{(1:t-1)}, x_{j-1}^{(t+1:t_{max})}, \hat{\theta}_{n_1}^{(1:d)}, y^{1:t_{max}})} \cdot \frac{q_t(x_{j-1}^{(t)} | \hat{\theta}_{n_1}^{(1:d)}, x_j^{(1:t-1)}, x_{j-1}^{(t+1:t_{max})})}{q_t(x_j^{(t)'} | \hat{\theta}_{n_1}^{(1:d)}, x_j^{(1:t-1)}, x_{j-1}^{(t+1:t_{max})})}\right). \quad (4.11)$$

N.B.

- In the hidden state estimation part, the target distribution for the hidden state $x^{(t)}$ at time t can be simplified according to the Markov chain property :

$$\begin{aligned} & \pi(x^{(t)} | x^{(1:t-1)}, x^{(t+1:t_{max})}, \theta^{(1:d)}, y^{1:t_{max}}) \\ \propto \quad & p(x^{(t)} | x^{(t-1)}) \;\; p(x^{(t+1)} | x^{(t)}) \;\; p(y^{(t)} | x^{(t)}). \end{aligned}$$

- As for the proposal distribution for the hidden state update of $x^{(t)}$, a convenient choice is to use the transition density of the model f. Since $x^{(t)}$ only depends on $\theta^{(1:d)}$ and on $x^{(t-1)}$ in the case of a state space model, the proposal distribution for the state variable $x^{(t)}$ can be simplified as :

$$q_t(x^{(t)} \mid \theta^{(1:d)}, x^{(1:t-1)}, x^{(t+1:t_{max})}) = p(x^{(t)} \mid \theta^{(1:d)}, x^{(t-1)}).$$

- Based on the previous simplifications, the acceptance ratio (4.11) can be described as:

$$\begin{aligned} & \alpha_{d+t}(x_{j-1}^{(t)}, x_j^{(t)'}) \\ = & \min\left(1, \frac{p(x_j^{(t)'} | x_j^{(t-1)}, \hat{\theta}_{n_1}^{(1:d)}) p(x_{j-1}^{(t+1)} | x_j^{(t)'}, \hat{\theta}_{n_1}^{(1:d)}) p(y^t | x_j^{(t)'}, \hat{\theta}_{n_1}^{(1:d)})}{p(x_{j-1}^{(t)} | x_j^{(t-1)}, \hat{\theta}_{n_1}^{(1:d)}) p(x_{j-1}^{(t+1)} | x_{j-1}^{(t)}, \hat{\theta}_{n_1}^{(1:d)}) p(y^t | x_{j-1}^{(t)}, \hat{\theta}_{n_1}^{(1:d)})} \frac{q_t(x_{j-1}^{(t)} | x_j^{(t-1)}, \hat{\theta}_{n_1}^{(1:d)})}{q_t(x_j^{(t)'} | x_j^{(t-1)}, \hat{\theta}_{n_1}^{(1:d)})}\right) \\ = & \min\left(1, \frac{p(x_{j-1}^{(t+1)} | x_j^{(t)'}, \hat{\theta}_{n_1}^{(1:d)}) p(y^{(t)} | x_j^{(t)'}, \tilde{\theta}_{j,t}^{(1:d)})}{p(x_{j-1}^{(t+1)} | x_{j-1}^{(t)}, \hat{\theta}_{n_1}^{(1:d)}) p(y^{(t)} | x_{j-1}^{(t)}, \tilde{\theta}_{j-1,t}^{(1:d)})}\right). \quad (4.12) \end{aligned}$$

- Instead of fixing $\theta^{(1:d)}$ at $\hat{\theta}_{n_1}^{(1:d)}$ when estimating the hidden state variables, an additional step of sampling $\tilde{\theta}_{j,t}^{(1:d)} \sim \mathcal{N}(\hat{\theta}_{n_1}^{(1:d)}, \hat{\Sigma}_{n_1})$ can be considered before drawing a new hidden state candidate. It allows to introduce more variability to the proposal. According to the Markov chain property :

$$\begin{aligned} & \pi(\theta^{(1:d)}, x^{(t)} | x^{(1:t-1)}, x^{(t+1:t_{max})}, y^{1:t_{max}}) \\ \propto \quad & p(\theta^{(1:d)}) \;\; p(x^{(t)} | x^{(t-1)}, \theta^{(1:d)}) \;\; p(x^{(t+1)} | x^{(t)}, \theta^{(1:d)}) \;\; p(y^{(t)} | x^{(t)}, \theta^{(1:d)}). \end{aligned}$$

In this way, we obtain a dual proposal as in the case for the parameter update, the acceptance ratio (4.11) however, remains the same :

$$\begin{aligned} & \alpha_{d+t}(\{x_{j-1}^{(t)}, \tilde{\theta}_{j-1,t}^{(1:d)}\}, \{x_j^{(t)'}, \tilde{\theta}_{j,t}^{(1:d)}\}) \\ = & \min\left(1, \frac{p(x_j^{(t)'}, \tilde{\theta}_{j,t}^{(1:d)} | x_j^{(t-1)}, x_{j-1}^{(t+1)}, y^{(t)})}{p(x_{j-1}^{(t)}, \tilde{\theta}_{j-1,t}^{(1:d)} | x_j^{(t-1)}, x_{j-1}^{(t+1)}, y^{(t)})} \frac{q(\tilde{\theta}_{j-1,t}^{(1:d)})}{q(\tilde{\theta}_{j,t}^{(1:d)})} \frac{q(x_{j-1}^{(t)} | x_j^{(t-1)}, \tilde{\theta}_{j-1,t}^{(1:d)})}{q(x_j^{(t)'} | x_j^{(t-1)}, \tilde{\theta}_{j,t}^{(1:d)})}\right) \\ = & \min\left(1, \frac{p(\tilde{\theta}_{j,t}^{(1:d)}) p(x_j^{(t)'} | x_j^{(t-1)}, \tilde{\theta}_{j,t}^{(1:d)}) p(x_{j-1}^{(t+1)} | x_j^{(t)'}, \tilde{\theta}_{j,t}^{(1:d)}) p(y^{(t)} | x_j^{(t)'}, \tilde{\theta}_{j,t}^{(1:d)})}{p(\tilde{\theta}_{j-1,t}^{(1:d)}) p(x_{j-1}^{(t)} | x_j^{(t-1)}, \tilde{\theta}_{j-1,t}^{(1:d)}) p(x_{j-1}^{(t+1)} | x_{j-1}^{(t)}, \tilde{\theta}_{j,t}^{(1:d)}) p(y^{(t)} | x_{j-1}^{(t)}, \tilde{\theta}_{j-1,t}^{(1:d)})}\right. \\ & \qquad \left. \cdot \frac{p(\tilde{\theta}_{j-1,t}^{(1:d)})}{p(\tilde{\theta}_{j,t}^{(1:d)})} \frac{p(x_{j-1}^{(t)} | x_j^{(t-1)}, \tilde{\theta}_{j-1,t}^{(1:d)})}{p(x_j^{(t)'} | x_j^{(t-1)}, \tilde{\theta}_{j,t}^{(1:d)})}\right) \\ = & \min\left(1, \frac{p(x_{j-1}^{(t+1)} | x_j^{(t)'}, \tilde{\theta}_{j,t}^{(1:d)}) p(y^{(t)} | x_j^{(t)'}, \tilde{\theta}_{j,t}^{(1:d)})}{p(x_{j-1}^{(t+1)} | x_{j-1}^{(t)}, \tilde{\theta}_{j,t}^{(1:d)}) p(y^{(t)} | x_{j-1}^{(t)}, \tilde{\theta}_{j-1,t}^{(1:d)})}\right). \end{aligned} \tag{4.13}$$

- Note also that if only the functional parameter estimation is desired, the hidden state estimation part can be omitted.

More discussion about the mixing property of state-space models can be found in Andrieu and Roberts (2009) and Andrieu et al. (2010).

Scarce observations

As mentioned previously, one important characteristic of the experimental data for plant growth models is that they are often limited and unevenly registered. Tanner and Wong (1987) point out that sometimes the posterior distributions are intractable because of the lack of some data, however MCMC can still be used in those situations. Therefore, the observation vector Y can be separated into two parts, $Y = (y, y')$, where y are the observed data and y' are the missing data. The current posterior distribution $\pi(\Theta|y)$ may appear to be muddled, while the "intended" posterior distribution $\pi(\Theta|Y)$ could be in a tractable form. According to the requirement of the Gibbs sampler, the full conditionals are accordingly:

– Update $\Theta \sim \pi(\Theta | x_{1:t_{max}}, y', y)$.

– Update $y' \sim \pi(y'|\Theta, x_{1:t_{max}}, y)$.
– Update $X \sim \pi(x_{1:t_{max}}|\Theta, y', y)$.

Note that $\pi(y'|\Theta, x_{1:t_{max}}, y) = \pi(y'|\Theta, x_{1:t_{max}})$. Consequently, the missing data y' can be treated as hidden states, also known as *latent variable*.

Given the fact that the missing observations y' are not really the objective of our estimation, in our implementation, the trajectory is divided into segments according to the observations y. The missing segment of the trajectory of Y is accepted or rejected according to the decision regarding the corresponding segment of x. More precisely, if t_i and t_{i+1} denote the time of two successive observations, then the proposal for the segments $x_{t_i+1:t_{i+1}}$ and $y'_{t_i+1:t_{i+1}-1}$ are accepted or rejected according to the decision for the proposal $x^{(t_{i+1})}$.

Further discussions about the general strategies regarding the missing data can be found in Smith and Roberts (1993) and Robert and Casella (1999). In the following, a series of key issues regarding the implementation of MCMC based methods are presented.

4.1.5 Implementation issues

To sum up, the implementation of an MCMC based method should involve constructing an ergodic Markov chain with transition kernel K which has an invariant distribution $\pi(\cdot)$.

However in practice, many issues have been raised during the implementation of MCMC, for instance, the burn-in period can be too long and potentially useful samples are wasted, or the chain could be stopped too early and miss some modes of the target distribution, or the estimation provided contains no information of the quality of the estimates. In this section, these problems are addressed and discussed.

Burn-in period

When applying an MCMC based method, it is highly possible that at the beginning of the chain, there is a period during which the algorithm has not converged to the stationary distribution and therefore should be discarded when computing the estimates. This period is known as *burn-in period*. The length of the burn-in period however, depends not only on the initial value of the chain X_0, but also on the shape of the target distribution.

To determine the length of the burn-in period, the intuitive way is to run the algorithm a few times with different initial values and to note the number of iterations needed before the chain converges to some kind of equilibrium. Thus the burn-in period is calculated in an empirically way based on these observations. As a matter of fact, the burn-in period is a method to find a good starting point, from which the chain is considered "converged" and the following iterations can be used for MCMC calculations.

There are other ways to find a good starting point, for example, to start the next run from the point where the last run ended, or to start at a mode of the equilibrium distribution obtained by other optimization methods. A Markov chain started anywhere near the center of the equilibrium distribution does not need burn-in. More discussion regarding the burn-in period issue can be found in Jones and Hobert (2004).

Up to now, we have reviewed some methods to construct a Markov chain which may produce an ergodic stationary distribution as its posterior distribution. In this case, as stated in Section 3.5.1, the average of the estimates converges towards a Gaussian distribution. However, since the samples are often not independent (i.e. correlation between consecutive iterations), one possible way to reduce the correlation is to take into account every k-th estimate of the chain, which is known as the *thinning* technique. Although the chain is no longer Markovian if treated this way, the subchains are still proved to converge to the correct distribution (Roberts and Rosenthal, 2004; Robert and Casella, 2010).

Furthermore, the precision of the mean estimate can as well be provided. In the following, the batch mean method is presented.

Precision estimation: Batch means

A batch consists of a subsequence of estimates given by consecutive iterations of the Markov chain $X_{k+1}, ..., X_{k+b}$, with batch length b. If the Markov chain reaches an ergodic and stationary distribution, then all batches with the same length should have the same joint distribution. Under regularity conditions (Theorem 3.5.3), the CLT can be applied.

Suppose $X_1, \ldots, X_m$ is a Markov chain of m iterations with the burn-in period discarded. Let $f : X \mapsto \mathbb{R}$ be a Borel function. We define $\bar{f}_m = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$. The batch mean method can be describe as follows :

– **Non-overlapping batch mean** : the chain is divided into batches of equal size. $m = a_m b_m$. a_m denotes the number of the batches and b_m denotes the batch length. The mean estimates of a_m batches are defined as :

$$\bar{Y}_j = \frac{1}{b_m}\sum_{k=1}^{b_m} f(X_{(j-1)b_m+k}), \qquad \text{for } j = 1, \ldots, a_m.$$

The batch means estimates of σ_f^2 are therefore :

$$\hat{\sigma}^2_{BM} = \frac{b_m}{a_m - 1}\sum_{j=1}^{a_m}(\bar{Y}_j - \bar{f}_m)^2 \tag{4.14}$$

– **Overlapping batch mean** : with the $m - b_m + 1$ overlapping batches of length b_m, the mean estimates are given by

$$\bar{Y}_j = \frac{1}{b_m}\sum_{k=1}^{b_m} f(X_{j-1+k}), \qquad \text{for } j = 1, \ldots, m - b_m + 1.$$

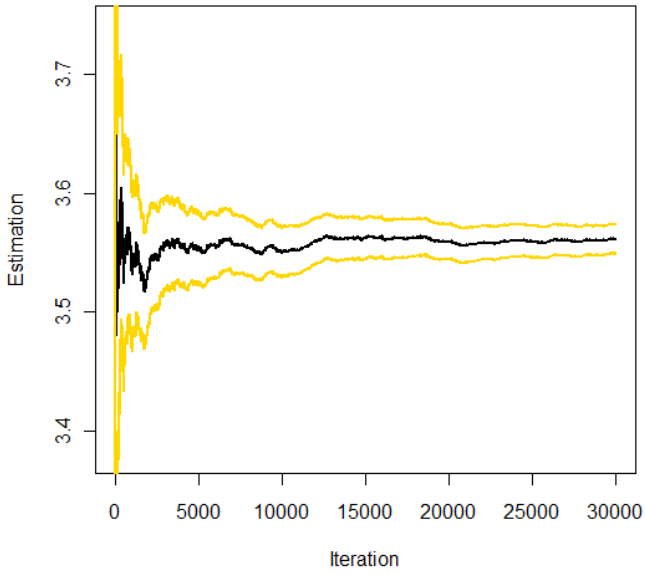


Figure 4.3: Variance estimation performed with batch means.

The batch means estimates of σ_f^2 are thus :

$$\hat{\sigma}^2_{OBM} = \frac{mb_m}{(n-b_m)(n-b_m+1)} \sum_{j=0}^{n-b_m} (\bar{Y}_j - \bar{f}_m)^2 \tag{4.15}$$

Batch means is an attractive method for its easy implementation. However, in literature, some authors warn that it should be used with caution since the estimator $\hat{\sigma}^2_{BM}$ is not consistent (Roberts, 1996). Flegal and Jones (2010) demonstrate that the overlapping batch means provide a consist estimator $\hat{\sigma}^2_{OBM}$ and are more stable than the non-overlapping batch means. On the other hand, Jones et al. (2006) argue that if the number of batches and their sizes are allowed to increase as the overall length of the chain increases, then by setting $b_m = \lfloor m^{\psi} \rfloor$ and $a_m = \lfloor m/b_m \rfloor$, $\hat{\sigma}^2_{BM} \to \sigma^2$ with probability 1 as $m \to \infty$. According to the regularity condition, the chain has to be geometrically ergodic (i.e. $\mathbb{E}_\pi |f|^{2+\delta+\epsilon} < 0$ for some strictly positive δ and strictly positive ϵ with $(1+\delta/2)^{-1} < \psi < 1$). Hence, $\psi = 1/2$ is a convenient choice ($b_m = \lfloor \sqrt{m} \rfloor$) in practice. Therefore, in this thesis, we consider the batch means with increasing batch length rather than the fixed length.

It is noteworthy that this approach allows us to obtain an estimation of the Monte Carlo Error (MCE) of the estimate $\bar{f}_m$: $\hat{\sigma}^2_f/m$, and from there, to form an asymptotically valid confidence interval:

$$[\bar{f}_m + t_{a_m-1}\frac{\hat{\sigma}_{BM}}{\sqrt{m}}; \bar{f}_m - t_{a_m-1}\frac{\hat{\sigma}_{BM}}{\sqrt{m}}].$$

where t_{a_m-1} is a quantile of Student's -t distribution with $a_m - 1$ degrees of freedom.

Note that the estimated Monte Carlo error can also be considered as a stopping criteria for the MCMC-based algorithm by putting a constraint on the precision of the estimate : limiting the confidence interval width. It can also be estimated by

performing a large number of the estimation with the same configuration, prior and initialization. Indeed, the stopping criteria is crucial, the most frequently asked question in practice is when the simulation can be ended. The Batch means is a popular method, yet it does not give any indication regarding the convergence of the chain. In the following, some convergence diagnostic related problems are discussed.

Convergence diagnostic and stopping rules

Undoubtedly, stopping rules and convergence diagnostic are both crucial and critical issues in the practice of MCMC-based methods. The objective is to decide whether the chain has reached its invariant distribution and whether the converged distribution is the target distribution. Regarding this issue, conflicting advices are often given in literature which makes the output analysis of MCMC a muddled area. The difficulty is that the convergence rate varies according to each algorithm and the specific target distribution. A chain may appear to have converged while it is not, as a result of poorly connected parts of the considered state space. The convergence is in fact conditioned on the initial value of the chain, which is known as *pseudo-convergence* or *pseudo-ergodicity*. To move from one part to another could take a surprisingly long time, especially in case of *multimodality* (Figure 4.4). As stated in Brooks and Roberts (1998), it seems that it is not possible to get any analytical tool or even stopping rules that could uniquely determine the necessary length of the run. Likewise, the MCMC based algorithm cannot be verified thoroughly whether the samples produced can truly characterize the target distribution. Hence, the proposed diagnostics are only used for educated guess regarding the convergence of the algorithm.

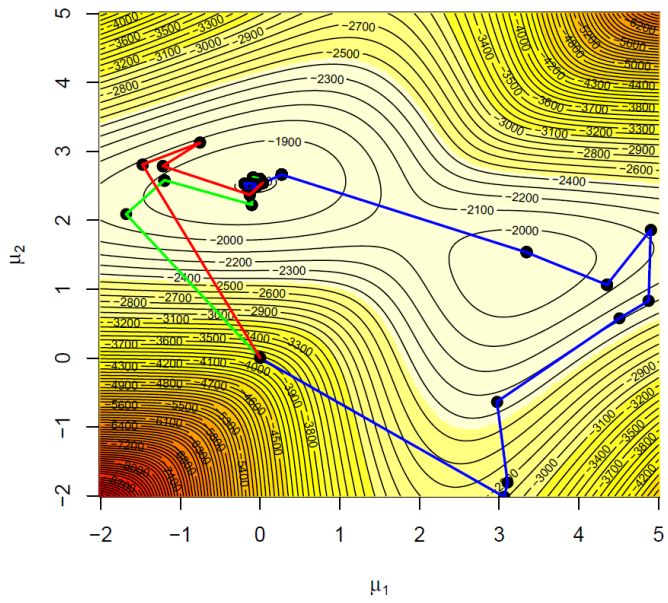


Figure 4.4: Multimodality: a contour plot of log-likelihood surface of a two-dimensional state-space and three independent runs (Robert and Casella, 2010).

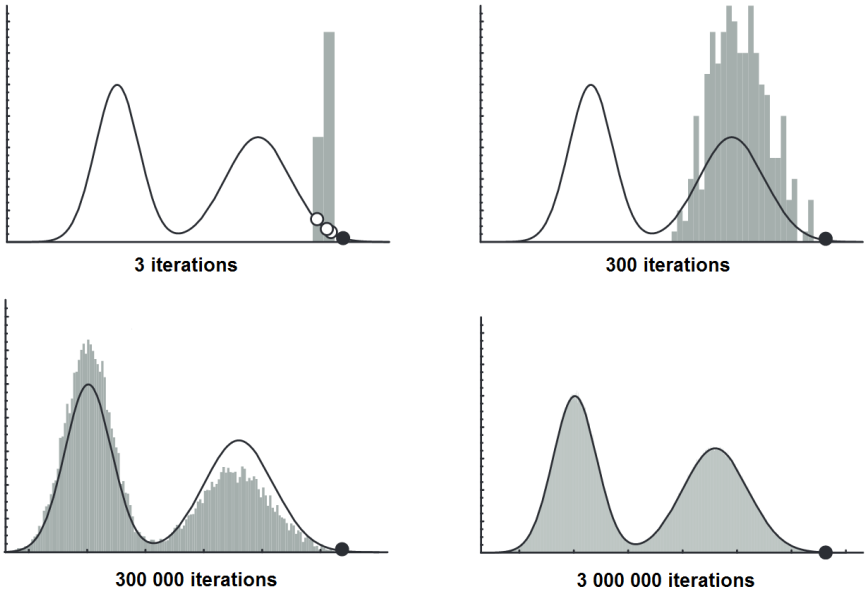


Figure 4.5: Histograms of runs with different numbers of iterations compared to the theoretical target distribution.

An easy way to verify the convergence is to run a few number of parallel chains with different initial values for each. With the growing iteration number, as illustrated by Figure 4.5, one may visually identify the period after which the chains start to converge, and if they converge to the same distribution.

On the other hand, monitoring the acceptance rate is also a very effective way to diagnostic the convergence. Although some optimal values are suggested in literature as mentioned above (0.234 for multidimensional problems and 0.44 for unidimensional problems). However, it is not always the case that 0.44 is the optimal, as demonstrated by Figure 4.2. The real optimal depends on the shape of the posterior distribution. Generally speaking, an acceptance rate between 0.1 and 0.5 can be considered as satisfactory.

As for the stopping rules, in this thesis, we propose a mixed strategy. Two criteria are considered, one to diagnostic the convergence of the chain and the other to claim the end of the run. Similar approaches based on the CLT are discussed in Chauveau and Diebolt (1999).

- Convergence diagnostic based on mean estimates

As we all know, the scaling nature of the Adaptive Metropolis changes the nature of the Markov chain, in the sense that current state of the chain depends not only on the one previous state, but on the history of the chain. Therefore, the first criterion is used to determine the end of the adaptive scheme (tuning of the random walk proposal) and define the burn-in period. In other words, once this criterion is satisfied, the covariance matrix Σ is fixed and no longer dependent on the evolution of the chain.

The criterion is grounded on a standard stopping rule computing the last a_n successive overlapped batch means with each containing b_n iterations (Booth and Hobert,

1999) with c_n iterations overlapped. $\forall i \in \{1, \ldots, d\}$, for $r = 1, \ldots, a_n$, if the relative changes in the batch means are sufficiently small :

$$\max_i \left(\frac{\mid \theta_{r+1}^{(i)} - \theta_r^{(i)} \mid}{\mid \theta_r^{(i)} \mid + \delta_1} \right) < \delta_2 \tag{4.16}$$

then the convergence can be claimed. Searle et al. (1992) suggest that $\delta_1 = 0.001$ and $\delta_2 = 0.0005$ are generally appropriate choices.

In the light of this basic criterion, a first convergence diagnostic can be provided, however, some parameters may exhibit a very slow first order convergence behaviour, especially when few data are available and the parameters are not sensitive to the model (see sensitivity analysis in Section 6.1). Thus, we propose to add another condition to this: the a_n successive relative changes should not be monotonic. In our implementation, $a_n = 4$, $b_n = 3000$ and $c_n = 1000$ seem to assure a good convergence behaviour.

- Stopping rule based on Monte Carlo Errors

If the previous convergence criterion is satisfied, the Monte Carlo Error can be used as the second condition to terminate the simulations. Indeed, asymptotic variance can also be used for MCMC convergence control Chauveau and Diebolt (2003). Theorem 3.5.3 shows that the precision of the estimates can be provided by assessing the Monte Carlo Error σ_f^2. It can be obtained by estimating the variance of the asymptotic distribution of $\hat{\pi}(f)$ with Batch means (Section 4.1.5).

$$\sigma_f^2 := \mathrm{Var}_\pi(f(X_1)) + 2 \sum_{i=2}^{\infty} \mathrm{Cov}_\pi(f(X_1), f(X_i))$$

as $n \to 0$. Therefore, in our implementation, a relative threshold of the estimation precision is defined for the all the parameters which translates as a constraint on the width of the confidence interval provided by the chain, so that the simulation continues if the interval is not narrow enough.

According to the implementation experience, to avoid a premature stop which would probably result in a poor estimate, a minimum number of the iterations is assured, which can be determined based on the number of parameters to estimate and the number of the available observations. Then, the proposing stopping criterion is checked every m_{BM} iterations (1000 or 5000 in our implementation) until it is satisfied for all the parameters and hidden states. This criterion can be described as follows :

$$\frac{\hat{\sigma}(\pi)}{\sqrt{m}} + p(m) \leq \epsilon,$$

where $p(m) = \epsilon I(m < m_{min})$ is the penalty term with fixed strictly positive m_{min} (10000 in our implementation), I is the indicator function on $\mathbb{Z}$ and ϵ is the threshold predefined according to the application.

More discussion regarding the stopping criteria and convergence diagnostics can be found in Jones and Hobert (2004), Chauveau and Diebolt (2003) and Roberts and Rosenthal (2004).

It is well known that when using MCMC-based methods, one may adopt either a sufficiently long run or many short runs as estimation strategies. In the following section, some algorithms based on multiple parallel MCMC are presented.

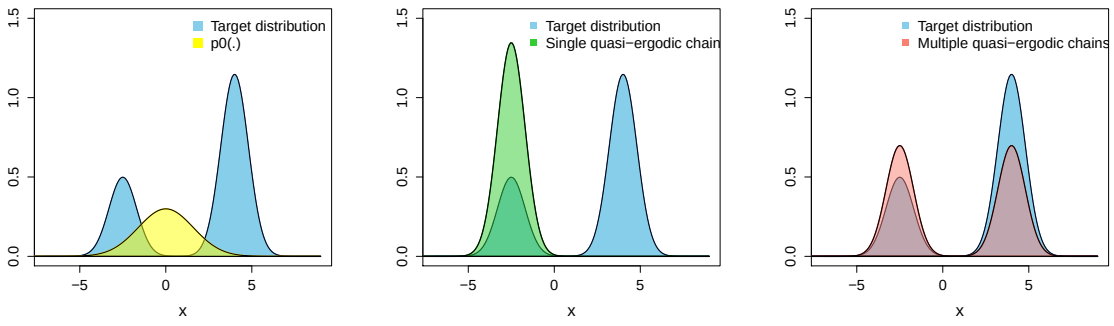
4.2 Parallel MCMC

In the literature, the idea of multiple short runs has been well developed, the motivation of which comes from the reasoning that a single chain is unable to efficiently cope with potentially complex posterior surface with multiple local optima. such as the work of Gilks et al. (1994) and Chauveau and Vandekerkhove (2002), who use multiple chains to explore the space, or some self-regenerative algorithms, such as Mykland et al. (1995), Gilks et al. (1998) and Sahu and Zhigljavsky (2003), in which mixture components are used. However, these approaches involve high computational cost which makes them rarely feasible to confront multi-dimensional applications. Recognizing these limits, the DiffeRential Evolution Adaptive Metropolis algorithm is proposed.

4.2.1 Differential Evolution Adaptive Metropolis

The DiffeRential Evolution Adaptive Metropolis (DREAM) algorithm is proposed by Vrugt et al. (2009a) based on the Differential Evolution-Markov chain (DE-MC) method (Ter Braak, 2006). The major idea is to run multiple sequences in parallel in an MCMC setting in order to hasten the convergence and to detect the multimodality problem in an efficient way if it is present in the target distribution.

In fact, in order to explore multimodal density in efficiently, a mode should be represented by at least a single chain, so that a second chain can be moved to it in accordance in the DREAM algorithm. This could be due to the *quasi-ergodicity*, which refers to isolated modes, or strong correlation between parameters.



(a) (b) (c)

Figure 4.6: Illustrations of quasi-ergodicity (Murray, 2010). (a): The target distribution and the prior distribution. (b): Estimation results from one quasi-ergodic Markov chain. (c): The samples result from multiple quasi-ergodic Markov chains.

Figure 4.6 gives two examples of quasi-ergodicity when multimodality is present in the target distribution. Figure 4.6(b) demonstrates the case when only one mode of the target distribution is detected if only one Markov chain is carried out with a limited length. It is highly possible that under multimodality target distribution, one Markov chain with limited length is conditional to its departure point. In the case that multiple chains are implemented as illustrated by Figure 4.6(c), we are more likely to detect both modes, however, it is still possible that their weight are not properly defined due to the small number of the chains which leads to the samples disproportionally being partitioned in the two modes.

The recommended configuration for the number of chains is $M = 2d$ (Ter Braak and Vrugt, 2008), with d the dimension of the target distribution initialised from over-dispersed distributions. Instead of running the chains independently as proposed by Gelman et al. (2004), jumps are made between chains by generating from a discrete proposal distribution to encourage the chains to learn from each other. The algorithm scales the orientation and tunes the proposal distribution by using an adaptive scheme (Adaptive Metropolis) and is designed to maintain the ergodicity by generating an exact approximation of the target distribution (posterior pdf) (Vrugt et al., 2009b).

More concretely, the chains are updated sequentially conditional to the other chains. A fixed multiple of the difference of the states belonging to randomly chosen pairs of other chains is taken into account in the proposal. Therefore, the DREAM can be seen as a single Markov chain on the state space $\mathcal{X}^{\mathcal{M}}$ (Ter Braak and Vrugt, 2008). The update of the i^{th} chain uses a mixture of kernels. Vrugt et al. (2009a) demonstrate that the Markov chain yielded by the DREAM algorithm is ergodic with unique stationary distribution of pdf $\pi(\cdot)^M$, with $\pi(\cdot)$ the original target distribution.

In the following, the DREAM algorithm is detailed and adaptations are made to fit the algorithm to the state-space model estimations.

Algorithm : DREAM

- **Initialization**
- For $i = 1, \ldots, M$:
  - Choose initial values for $X_{i,0} = (\theta_{i,0}^{(1)}, \ldots, \theta_{i,0}^{(d)}, x_{i,0}^{(1)}, \ldots, x_{i,0}^{(t_{max})})$ from the prior distributions $p_0(\cdot)$.
- **Parameters**
  Iteration
  - For $j = 1, \ldots, n_1$:
    1. Proposal step: for $i = 1, \ldots, M$:
       – Sample e and ϵ:

$$e \sim \mathcal{U}_d(-b, b) \qquad \epsilon \sim \mathcal{N}_d(0, b^*)$$

with $|b| < 1$ and b^* smaller than the width of the target distribution.

– Sample $\theta_{i,j}^{(1:d)'}$:

$$\theta_{i,j}^{(1:d)'} = \theta_{i,j-1}^{(1:d)} + (1_d + e_d)\gamma(\delta)\left(\sum_{k=1}^{\delta}\theta_{r(k),j-1}^{(1:d)} - \sum_{l=1}^{\delta}\theta_{r(l),j-1}^{(1:d)}\right) + \epsilon_d.$$

where δ denotes the number of pairs used for the candidate, $r(k), r(l) \in \{1, \ldots, M\}$, $r(k) \neq r(l) \neq i$, and $\gamma = 2.38/\sqrt{2\delta d}$.

– Sample $x_{i,j}^{(1:t_{max})'} \sim p(x^{1:t_{max}}|\theta_{i,j}^{(1:d)'})$.

2. Transition step: for $i = 1, \ldots, M$:

– Set

$$\{\theta_{i,j}^{(1:d)}, x_{i,j}^{(1:t_{max})}\} = \begin{cases} \{\theta_{i,j}^{(1:d)'}, x_{i,j}^{(1:t_{max})'}\} \text{ with probability } \alpha_\theta(\theta_{i,j-1}^{(1:d)}, \theta_{i,j}^{(1:d)'}), \\ \{\theta_{i,j-1}^{(1:d)}, x_{i,j-1}^{(1:t_{max})}\} \text{ with probability } 1 - \alpha_\theta(\theta_{i,j-1}^{(1:d)}, \theta_{i,j}^{(1:d)'}), \end{cases}$$

where $\alpha_\theta(\theta, \theta')$ is given by (4.10).

Estimator

– For a burn-in period $n_0 < n_1$:

$$\hat{\theta}_{n_1}^{(1:d)} = \frac{1}{M(n_1 - n_0)} \sum_{l=1}^{l=M} \sum_{k=n_0}^{n_1} \theta_{l,k}^{(1:d)},$$

$$\hat{\Sigma}_{n_1} = \frac{1}{M(n_1 - n_0) - 1} \sum_{l=1}^{l=M} \sum_{k=n_0}^{n_1} (\theta_{l,k}^{(1:d)} - \hat{\theta}_{n_1}^{(1:d)})^T (\theta_{l,k}^{(1:d)} - \hat{\theta}_{n_1}^{(1:d)})$$

- **Hidden states**

– For $i = 1, \ldots, M$:

– Sample $\tilde{\theta}_i^{(1:d)} \sim \mathcal{N}(\hat{\theta}_{n_1}^{(1:d)}, \hat{\Sigma}_{n_1})$.

– Sample ${x_{i,n_1}}^{(1:t_{max})} \sim p(x^{1:t_{max}}|\tilde{\theta}_i^{(1:d)})$.

Iteration

- For $j = n_1 + 1, \ldots, n_2; n_2 > n_1 + 1$:

- For $i = 1, \ldots, M$:

- For $t = 1, \ldots, t_{max}$:

1. Sample $x_{i,j}^{(t)'} \sim q_t(x^{(t)} \mid \tilde{\theta}_i^{(1:d)}, x_{i,j}^{(1)}, \ldots, x_{i,j}^{(t-1)}, x_{i,j-1}^{(t+1)}, \ldots, x_{i,j-1}^{(t_{max})})$.

2. Set

$$x_{i,j}^{(t)} = \begin{cases} x_{i,j}^{(t)'} & \text{with probability } \alpha_{d+t}(x_{i,j-1}^{(t)}, x_{i,j}^{(t)'}), \\ x_{i,j-1}^{(t)} & \text{with probability } 1 - \alpha_{d+t}(x_{i,j-1}^{(t)}, x_{i,j}^{(t)'}), \end{cases}$$

where $\alpha_{d+t}(x, y)$ can be obtained according to (4.13).

N.B.

- Note that here, we opt to keep independent chains for the estimation of the hidden state variables while the same between-chain communication can be implemented as for the estimation of the parameters. However, the comparison result suggests that it does not significantly improve the estimation probably due to the important time dependency among the hidden states.

- The resampling step for the parameters at the beginning of the hidden state estimation can be omitted, and therefore reduce the variability of the hidden state proposals.

- Another possible extension for the hidden state variable estimation is to transform the Gibbs sampling to importance sampling.

- The posterior distribution is built based on the pool of samples from all the chains except those of the burn-in period.

Unfortunately, in practice, we may run into convergence problems since the chains may not converge fast enough, therefore the parallel short runs will turn out to be parallel long runs and the algorithm may end up with computational complexity. In Geyer (1992) and Robert and Casella (1999) (Section 6.5), the authors argue that independent parallel chains could be a poor idea, for some of them may not converge. Therefore, one long chain could be preferred. To avoid this frequently occurred problem, another technique is developed, taking into account only the last estimates of an important number of chains. In this way, the parallel computing may reduce efficiently the computational time. The following algorithm, known as Interacting MCMC is one of the approaches that adopt this strategy.

4.2.2 Interacting Metropolis-within-Gibbs

As stated previously, the idea of multiple MCMC is to accelerate the state space in a more efficient way compared to single MCMC, and can be parallelised and distributed to multiple CPU to save computational time. It is a temping idea, despite the convergence of all the chains which is hard to achieve. Efforts have been made to encourage the exchange among chains (Chauveau and Vandekerkhove, 2002; Drugan and Thierens, 2005; Laskey and Myers, 2003) with the purpose of improving the mixing properties of the samples. A Population Monte Carlo framework was proposed by Cappé et al. (2004).

In the following Interacting MCMC algorithm introduced by Campillo et al. (2009), to concentrate the computational effort on the zone of interest, each individual chain proposes one candidate for all the chains. It is proved that the set of interacting chains has a unique stationary distribution with pdf $\pi(\cdot)^M$, with $\pi(\cdot)$ the original target distribution and M the number of parallel chains. Each of the M chains provides a sample to construct the posterior distribution (Campillo et al., 2009).

Algorithm : Interacting Metropolis-within-Gibbs

- **Initialization**

- For $i = 1, \ldots, M$:
  - Choose initial values for $X_{i,0} = (\theta_{i,0}^{(1)}, \ldots, \theta_{i,0}^{(d)}, x_{i,0}^{(1)}, \ldots, x_{i,0}^{(t_{max})})$ from the prior distributions $p_0(\cdot)$.

- **Parameters**
  Iteration
  - For $j = 1, \ldots, n_1$:
    - Proposal step: for $i = 1, \ldots, M$:
      1. Sample $\theta_{i,j}^{(1:d)'} \sim \mathcal{N}(\theta_{i,j-1}^{(1:d)}, \Sigma_0)$.
      2. Sample $x_{i,j}^{(1:t_{max})'} \sim p(x^{1:t_{max}} | \theta_{i,j}^{(1:d)'})$.
    - Selection step: for $i = 1, \ldots, M$:
      1. Compute $\alpha_\theta(\theta_{i,j-1}^{(1:d)}, \theta_{i,j}^{(1:d)'})$ as in (4.10).
      2. Set

$$\{\theta_{i,j}^{(1:d)}, x_{i,j}^{(1:t_{max})}\} \begin{cases} \{\theta_{1,j}^{(1:d)'}, x_{1,j}^{(1:t_{max})'}\} \text{ with probability } \frac{1}{M}\alpha_\theta(\theta_{i,j-1}^{(1:d)}, \theta_{1,j}^{(1:d)'}), \\ \ldots \\ \{\theta_{M,j}^{(1:d)'}, x_{M,j}^{(1:t_{max})'}\} \text{ with probability } \frac{1}{M}\alpha_\theta(\theta_{i,j-1}^{(1:d)}, \theta_{M,j}^{(1:d)'}), \\ \{\theta_{i,j-1}^{(1:d)}, x_{i,j-1}^{(1:t_{max})}\} \text{ with probability } 1 - \frac{1}{M}\sum_{k=1}^{M} \alpha_\theta(\theta_{i,j-1}^{(1:d)}, \theta_{k,j}^{(1:d)'}). \end{cases}$$

- **Hidden states**
  Iteration
  - For $j = n_1 + 1, \ldots, n_2; n_2 > n_1 + 1$:
    - For $t = 1, \ldots, t_{max}$:
      - Proposal step: for $i = 1, \ldots, M$:
        1. Sample $x_{i,j}^{(t)'} \sim q_t(x^{(t)} \mid \theta_{i,n_1}^{(1:d)}, x_{i,j}^{(1)}, \ldots, x_{i,j}^{(t-1)}, x_{i,j-1}^{(t+1)}, \ldots, x_{i,j-1}^{(t_{max})})$
        2. Compute $\alpha_x(x_{i,j-1}^{(t)}, x_{i,j}^{(t)'})$ according to (4.13).
      - Selection step: for $i = 1, \ldots, M$:
        Set

$$x_{i,j}^{(t)} = \begin{cases} x_{1,j}^{(t)'} & \text{with probability } \frac{1}{M}\alpha_x(x_{i,j-1}^{(t)}, x_{1,j}^{(t)'}), \\ \ldots & \\ x_{M,j}^{(t)'} & \text{with probability } \frac{1}{M}\alpha_x(x_{i,j-1}^{(t)}, x_{M,j}^{(t)'}), \\ x_{i,j-1}^{(t)} & \text{with probability } 1 - \frac{1}{M}\sum_{k=1}^{M} \alpha_x(x_{i,j-1}^{(t)}, x_{k,j}^{(t)'}). \end{cases}$$

Note that the transition kernel and the prior distribution are regarded as proposal distributions respectively for the joint estimation of the parameter vector and the hidden state variables.

The posterior distribution is build only based on the samples of the last iteration from all the chains.

More discussion about population based simulations for static inference can be found in Jasra et al. (2007) and Mengersen and Robert (2003).

4.2.3 Convergence criteria for multiple chains

Gelman-Rubin (Gelman, 1992)

For parallel chains, the Gelman-Rubin criterion is created based on the normal theory approximation to exact Bayesian posterior inference. The idea involves the estimation of the target distribution as a conservative Student-t distribution by using the last n iterations of the considering chains. Both the between-chain variance and the with-in chain variance are evaluated to compute a scale parameter, with the aim of monitoring the convergence.

To use this criterion, the assumption that the starting point is sampled from an appropriately over-dispersed prior distribution is made. More precisely, to carry out this evaluation of convergence, 7 steps are involved. Firstly, suppose that we simulate m independent chains of length n after the burn-in period. The Gelman and Rubin stopping criterion needs to compute the following elements, $\forall\theta^{(k)}$, $k \in \{1 \dots d\}$:

- $\hat{\theta}^{(k)}$, which is the estimator of the target mean $\mu_{\theta^{(k)}} = \int \theta^{(k)} P(\theta^{(k)}) d\theta^{(k)}$.

$$\hat{\theta}^{(k)} = \bar{\theta}^{(k)}_{\cdot\cdot}$$

- $B^{(k)}/n$, which denotes the variance between the m chain means, each based on n estimations of $\theta^{(k)}$.

$$B^{(k)}/n = \frac{1}{m-1} \sum_{i=1}^{m} (\bar{\theta}^{(k)}_{i\cdot} - \bar{\theta}^{(k)}_{\cdot\cdot})^2$$

- W, which is the average of the m within-chain variances.

$$W^{(k)} = \frac{1}{m} \frac{1}{(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{n} (\theta^{(k)}_{ij} - \bar{\theta}^{(k)}_{i\cdot})^2 = \frac{1}{m} \sum_{i=1}^{m} \left(s^{(k)}_i\right)^2$$

- $\left(\hat{\sigma}^{(k)}\right)^2$, which is the estimator of the target variance $\int (\theta^{(k)} - \mu_{\theta^{(k)}})^2 P(\theta^{(k)}) d\theta^{(k)}$.

$$\left(\hat{\sigma}^{(k)}\right)^2 = \frac{n-1}{n} W^{(k)} + \frac{1}{n} B^{(k)}$$

  Note that the variance estimator may overestimate the variance, if we assume that the prior distribution is over-dispersed. However, it remains unbiased if the prior distribution is the target distribution (under stationarity). The reason to explain the two components of $\left(\hat{\sigma}^{(k)}\right)^2$ is that W alone can under-estimate the overall variance, since the individual chains may do not have enough time to explore the whole target distribution, and therefore have less variability. When $n \to \infty$, the expectation of W approaches $\left(\sigma^{(k)}\right)^2$.

- $\sqrt{\hat{V}^{(k)}}$, as the scale of the estimated target distribution. The motivation is to be more tolerance (allowing more variability) by adopting an approximation of Student's t distribution for $\theta^{(k)}$.

$$\sqrt{\hat{V}^{(k)}} = \sqrt{\left(\hat{\sigma}^{(k)}\right)^2 + B^{(k)}/mn}$$

with $df = 2(\hat{V}^{(k)})^2/\hat{\text{Var}}(\hat{V}^{(k)})$ degrees of freedom, where

$$\hat{\text{Var}}(\hat{V}^{(k)}) = \left(\frac{n-1}{n}\right)^2 \frac{1}{m}\hat{\text{Var}}((s_i^{(k)})^2) + \left(\frac{m+1}{mn}\right)^2 \frac{2}{m-1}(B^{(k)})^2 + 2\frac{(m+1)(n-1)}{mn^2}\frac{n}{m}[\hat{\text{Cov}}((s_i^{(k)})^2, (\bar{\theta}_{i\cdot}^{(k)})^2) - 2\bar{\theta}_{\cdot\cdot}^{(k)}\hat{\text{Cov}}((s_i^{(k)})^2, \bar{\theta}_{i\cdot}^{(k)})]$$

– $\sqrt{\hat{R}^{(k)}}$, served as the convergence monitor, built by estimating the potential scale reduction.

$$\sqrt{\hat{R}^{(k)}} = \sqrt{\frac{\hat{V^{(k)}}}{W^{(k)}}\frac{df}{(df-2)}} \overset{n\to\infty}{\longrightarrow} 1$$

$\hat{R}^{(k)}$ is the ratio of the current estimate of variance $\hat{V^{(k)}}$ to the within-chain variance $W^{(k)}$ corrected by a factor account for the extra variance considered from the Student's t distribution. If $\sqrt{\hat{R}^{(k)}}$ is important, then further simulation may be needed to improve the inference based on the observations. Therefore, a threshold can be defined to indicate the convergence and stop the simulation based on $\sqrt{\hat{R}^{(k)}}$, which is usually close to 1. In our implementation, 1.0001 is chosen.

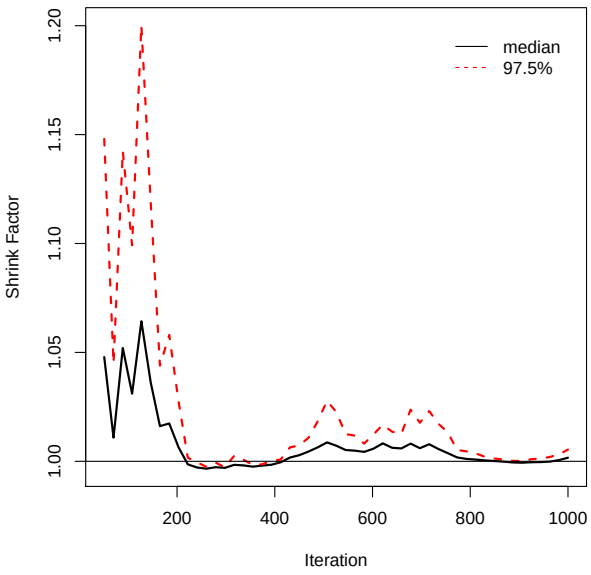


Figure 4.7: The evolution of shrink factor from the Gelman-Rubin criterion.

Note that another desirable function of this criterion due to its feature is that it could help us to detect the problem of multi-modal target distribution by producing a value of $\sqrt{\hat{R}^{(k)}}$ not decreasing to 1.

In our implementation, this criterion is used for the DREAM algorithm.

4.3 Sequential Monte Carlo methods

Within a Bayesian framework, Sequential Monte Carlo (SMC) methods as well as Monte Carlo Markov Chain (MCMC) based methods offer the opportunity to estimate the parameter distribution (posterior) based on the predefined knowledge (prior) (Gelman et al., 1995; Andrieu et al., 2001). However, unrealistic results might be obtained by MCMC based methods with a given inappropriate prior.

Moreover, despite all the advantage of MCMC-based methods, in practice, many problems require online estimation, they hence rely on recursive methods. Therefore, the filtering methods might be considered as a better choice. Traditional estimation methods for linear systems subject to normal noises are based on the Kalman Filter (Kalman, 1960; Kailath et al., 2000). However, to answer the growing complexity of the considering models often nonlinear, extensions of Kalman Filters, such as Extended Kalman Filter (Anderson and Moore, 1979; Sorenson, 1985), Unscented Kalman Filter (Julier and Uhlmann, 1997; Quach et al., 2007) and Ensemble Kalman filter (Evensen, 1994, 2006) are developed. Only in recent years that the growth in computational power has made computational intensive statistical methods feasible. More and more approaches appeared solving directly the estimation problem based on Monte Carlo techniques (Doucet et al., 2001), instead of using approximation to linearize the system. The main breakthrough came with the invention of particle filtering method (Gordon et al., 1993). In the meantime, new issues arise mainly in regards of uncertainty assessment.

In this thesis, the particle filter based methods are mainly analysed and applied, whereas the Kalman Filter based methods are served as comparison. In the following, we first introduce the context of Bayesian inference and describe the general filtering recursion. The algorithms of two Kalman based filters, the Unscented Kalman Filter and the Ensemble Kalman Filter are reviewed. Subsequently, two particle filter based algorithms, the Regularized Particle Filter and the Convolution Particle Filter are presented.

General Bayesian estimation problems

For $n \in \mathbb{N}$, the classical estimation problems can be divided into three categories:

- *Filtering* : it involves the extraction of information of a variable of interest X_n at time n from the data observed up to time n: $Y_{0:n}$.
- *Prediction* : it aims to derive information of a variable of interest $X_{n+\tau}$ at some time $n+\tau$ ($\tau \in \mathbb{N}^*$) in the future based on the observation up to time n.
- *Smoothing* : it consists using all the observations $Y_{0:n}$, even those after the time of interest n' ($n' \in \mathbb{N}, n' < n$) for the estimation of $X_{n'}$.

For filtering methods, the objective is to estimate jointly the unknown parameters and the hidden states. An augmented state vector $X_n^a = (X_n, \Theta_n)$ is consequently defined for each time step $n \in \{0, \ldots, t_{max}\}$ with the associated state space $\mathcal{X}^a \overset{\text{def}}{=} \mathcal{X} \times \mathcal{P}$. It contains X_n the true hidden state at time t_n and Θ_n the vector of unknown

parameters which can be considered as a random variable defined on the measurable space $(\mathcal{P}, B(\mathcal{P}))$ (see Quach et al. (2007) and d'Alché Buc and Brunel (2010) for examples). In the following, if X represents a random variable with values in $\mathcal{X}$, then for all $x \in \mathcal{X}$, $p(x)$ denotes the probability density of X on x.

A nature way to deal with the dynamic parameter vector Θ_n is to keep it being constant in time. That is to say: $\Theta_n = \Theta_{n+1}$. However, in Section 5.1, a new version of particle filter may enable us to modify Θ_n sequentially in time. Therefore, an artificial dynamic evolution is assigned to Θ_n :

$$\Theta_{n+1} = h_{n+1}(\Theta_n, \zeta_{n+1}), \qquad n \geq 0,$$

with a Borel function h_n and $\{\zeta_n\}_{n\geq 0}$ a sequence of i.i.d. random variables which is mutually independent of $\{\xi_n\}_{n\geq 0}$ and $\{\eta_n\}_{n\geq 0}$. For $n \in \mathbb{N}$, we assume that Θ_n admits a density with respect to the Lebesgue measure, denoted $p(\Theta_n)$.

Therefore, the description of state-space model (3.1) presented in Section 3.2 is slightly modified :

$$\begin{cases} X_{n+1}^a = f_{n+1}^a \left(X_n^a, E_n, \eta_{n+1}^a, \right), & n \geq 0, \\ Y_n = g_n \left(X_n^a, \Theta, \xi_n\right), n \geq 0, \end{cases} \tag{4.17}$$

with $\eta^a = (\eta, \zeta)$ and $f : \mathcal{X} \times \mathcal{P} \times \mathcal{E} \times \mathbb{R}^{d_{\eta_n}+d_{\zeta_n}} \mapsto \mathcal{X} \times \mathcal{P}$ which characterizes the evolution of X_n^a. We denote $\mathcal{X} \times \mathcal{P}$ by $\mathcal{X}^a$.

General filtering recursions

The filtering methods allow us to estimate the augmented hidden state variable X_n^a based on $Y_{0:n}$. An MMSE estimator of $\hat{X}_n^a$ conditional to $Y_{0:n}$ is often chosen and defined as:

$$\hat{X}_n^a = \mathbb{E}[X_n^a | Y_{0:n}] = \int_{\mathcal{X}^a} x_n^a p(x_n^a | y_{0:n}) \lambda(dx_n^a) = \int_{(\mathcal{X}^a)^{n+1}} f(x_{0:n}^a) p(x_{0:n}^a | y_{0:n}) \lambda(dx_{0:n}^a).$$

with λ the Lebesgue measure on $\mathcal{X}^a$. $p(x_n^a | y_{0:n})$ is the marginal density of $p(x_{0:n}^a | y_{0:n})$. Note that if only a parameterization is desired, it is possible to extract the estimation of the parameter vector Θ_n from the estimator $\hat{X}_n^a$.

The filtering process can be summarized by two steps, the *prediction*, also known as *time update*, and the *correction*, respectively known as *measurement update*. The first-order hidden Markov model is characterized by the transition density $p(x_n^a | x_{n-1}^a)$ corresponding to the state equation (Quach et al., 2007), and the observation density $p(y_n | x_n^a)$ corresponding to the observation equation and the initial density $p(x_0^a)$.

– Determinate the density distribution to predict the hidden state at time $n+1$ based on the former observations.

$$p(x_{n+1}^a | y_{0:n}) = \int_{\mathcal{X}^a} p(x_{n+1}^a | x_n^a) p(x_n^a | y_{0:n}) \lambda(dx_n^a). \tag{4.18}$$

– Correct the density distribution by taking into account the new observation y_{n+1}.

$$p(x^a_{n+1}|y_{0:n+1}) = \frac{p(x^a_{n+1}|y_{0:n})p(y_{n+1}|x^a_{n+1})}{\int_{\mathcal{X}^a} p(x^a_{n+1}|y_{0:n})p(y_{n+1}|x^a_{n+1})\lambda(dx^a_{n+1})}. \tag{4.19}$$

Kalman filter can be regarded as the most efficient filtering method for linear system. However, it cannot cope with nonlinear systems. To date, many efforts have been made to develop extension for Kalman filter in nonlinear systems, the most well known extensions remain to be the extended Kalman filter, the unscented Kalman filter and the ensemble Kalman filter. The Extended Kalman Filter (EKF) (Evensen, 1994) simply linearizes locally the model so that the tradition Kalman filter can be applied. However, it exhibits poor performance with highly nonlinear system, while both latter methods generalize elegantly to nonlinear systems which are free of the linearization required by the EKF. In the following, the Unscented Kalman Filter and ensemble Kalman filter are presented. First, we revisit the algorithm of the famous Kalman Filter.

4.3.1 Unscented Kalman Filter

Kalman Filter

Generally, there does not exist a finite dimensional solution to the Bayesian estimation problem. However, for a linear system subject to Gaussian noise, the solution can be formulated recursively using the Kalman Filter (Kalman, 1960), which can be regarded as the most efficient filtering method for linear systems.

Let us consider a linear dynamic system (4.17) with both f^a_n and g_n linear functions. Suppose the density $p(x^a_0)$ associated to X^a_0 is Gaussian. Since a Gaussian distribution can be characterized by its first two moments, mean and covariance matrix, the update of $p(x^a_n|y_{0:n})$ concerns only the density of the expectation $\hat{x}^a_{n|n}$ and the covariance matrix $\hat{\Sigma}^{x^a}_{n|n}$:

$$\begin{aligned} \hat{x}^a_{n|n} &= E[X^a_n|Y_{0:n}] \\ \hat{\Sigma}^{x^a}_{n|n} &= E[(X^a_n - \hat{x}^a_{n|n})^T(X^a_n - \hat{x}^a_{n|n})|Y_{0:n}]. \end{aligned} \tag{4.20}$$

In the following, we assume that η^a_n and ξ_n are white noises drawn from Gaussian distributions :

$$\eta^a_n \sim \mathcal{N}(0, J_n) \quad \text{and} \quad \zeta_n \sim \mathcal{N}(0, R_n).$$

For the sake of simplicity, we denote $f^a_{n+1}(X^a_n) = f^a_{n+1}(X^a_n, E_n, \eta^a_{n+1})$ and $g_{n+1}(X^a_{n+1}) = g_{n+1}(X^a_{n+1}, \xi_{n+1})$. The filtering process is thus carried out in two steps.

• **Prediction** (*time update*) : the aim is to determinate $p(x^a_{n+1}|y_{0:n})$ using the evolution equation f^a_{n+1} of the state-space model.

$$\hat{x}^a_{n+1|n} = E[X^a_{n+1}|Y_{0:n}] = E[f^a_{n+1}(X^a_n, \eta^a_{n+1})|Y_{0:n}].$$

Note that η^a_{n+1} is independent of $\{\eta^a_k\}_{k=1,\dots,n}$ and $\{\xi_k\}_{k=1,\dots,n+1}$, and therefore is independent of $Y_{0:n+1}$. Its expected value is determined by $\mathcal{N}(0, J_{n+1})$ and $p(x^a_n|y_{0:n})$.

$$\hat{\Sigma}^{x^a}_{n+1|n} = E[(X^a_{n+1} - \hat{x}^a_{n+1|n})^T(X^a_{n+1} - \hat{x}^a_{n+1|n})|Y_{0:n}].$$

In the same way, we can deduce the associate expectation of $p(y_{n+1}|y_{0:n})$:

$$\hat{y}_{n+1|n} = E[Y_{n+1}|Y_{0:n}] = E[g_{n+1}(X^a_{n+1})|Y_{0:n}], \tag{4.21}$$

since ξ_{n+1} and $Y_{0:n}$ are independent, the corresponding covariance matrix is therefore:

$$\begin{aligned}\hat{\Sigma}^{y}_{n+1|n} &= E[(Y_{n+1} - \hat{y}_{n+1|n})^T(Y_{n+1} - \hat{y}_{n+1|n})|Y_{0:n}] \\ &= E[(g_{n+1}(X^a_{n+1}) - \xi_{n+1} - \hat{y}_{n+1|n})^T(g_{n+1}(X^a_{n+1}) - \xi_{n+1} - \hat{y}_{n+1|n})|Y_{0:n}] + R_{n+1}.\end{aligned} \tag{4.22}$$

The correlation matrix between X^a_{n+1} and Y_{n+1} conditional by $Y_{0:n}$ can be calculated :

$$\hat{\Sigma}^{x^a y}_{n+1|n} = E[(X^a_{n+1} - \hat{x}^a_{n+1|n})^T(Y_{n+1} - \xi_{n+1} - \hat{y}_{n+1|n})|Y_{0:n}].$$

• **Correction** (*measurement update*) : the expectation and the covariance matrix associated to $p(x^a_{n+1}|y_{0:n+1})$ can be estimated by :

$$\begin{aligned}\hat{x}^a_{n+1|n+1} &= \hat{x}^a_{n+1|n} + K_{n+1}(y_{n+1} - \hat{y}_{n+1|n})^T \\ \hat{\Sigma}^{x^a}_{n+1|n+1} &= \hat{\Sigma}^{x^a}_{n+1|n} - K_{n+1}\hat{\Sigma}^{y}_{n+1|n}K^T_{n+1}.\end{aligned} \tag{4.23}$$

K_{n+1} is known as the Kalman gain, which can be expressed as :

$$K_{n+1} = \hat{\Sigma}^{x^a y}_{n+1|n}\left(\hat{\Sigma}^{y}_{n+1|n}\right)^{-1}.$$

Unscented transform

As one of the nonlinear extensions of classical Kalman Filter, the Unscented Kalman Filter (UKF) (Julier and Uhlmann (1997) and Quach et al. (2007)) adopts deterministic sampling aiming at using a small set of discretely sampled points, known as sigma-points (Unscented Transform, see Arulampalam et al. (2002)) to get hold of the information of higher order for both mean and covariance matrix based on normal assumptions. Before proceeding to the UKF, we first introduce the Unscented transform.

Consider X, Y two random variables and a nonlinear Borel function f:

$$Y = f(X).$$

The objective is to provide an approximation of $\mathbb{E}[Y] = \mathbb{E}[f(X)]$ in the case when X follows a normal distribution $\mathcal{N}(\bar{x}, \Sigma_x)$. We denote d as the dimension of X. First, we simulate $2d+1$ samples to represent the distribution of X, namely sigma-points :

$$\begin{cases} \chi^0 & = \overline{x} \\ \chi^i & = \overline{x} + \left(\sqrt{(d+\kappa)\Sigma_x}\right)_i & i \in \{1,\dots,d\} \\ \chi^{d+i} & = \overline{x} - \left(\sqrt{(d+\kappa)\Sigma_x}\right)_i & i \in \{1,\dots,d\} \end{cases}$$

with $\kappa > -d$. $\left(\sqrt{(d+\kappa)\Sigma_x}\right)_i$ represents the ith line or column of $\sqrt{(d+\kappa)\Sigma_x}$. Each sigma point χ^i is associated with a weight ω_i which is defined as :

$$\begin{cases} \omega_0 & = \kappa/(d+\kappa) \\ \omega_i & = 1/2(d+\kappa) \qquad i \in \{1,\dots,d\} \\ \omega_{i+d} & = 1/2(d+\kappa) \qquad i \in \{1,\dots,d\} \end{cases}$$

Thus, the following approximation can be made :

$$E[Y] = E[f(X)] \approx \sum_{i=0}^{2d} \omega_i f(\chi^{(i)}).$$

N.B.

A good choice of κ is $d-3$ (see Julier and Uhlmann (1997))

If f is linear, then the approximation of $E[Y]$ is exact which allows a third-order approximation of the expectation.

The Choleski decomposition can be used to compute the square root of $(d+\kappa)\Sigma_x$.

Unscented Kalman Filter

The Unscented Kalman Filter (UKF) (Julier and Uhlmann (1997) and Quach et al. (2007)) is known as one of the nonlinear extensions of classical Kalman Filter. When f_n^a and g_n are no longer linear, the density $p(x_n^a|y_{0:n})$ does not follow the normal distribution any more. Thanks to the Unscented Transform, the approximation of both mean and covariance matrix can be made via the sigma-points. Therefore, it does not require a tangent linear model as EKF does and claims a better precision and robustness for nonlinear models.

We denote $d_\eta = dim(\eta_{n+1})$, $d_X = dim(\hat{x}^a_{n|n})$ and $d_{\eta,X} = d_\eta + d_X$. We recall that J_n and R_n are the covariance matrices for η_n^a and ξ_n respectively. The two steps of the filtering process can be described as follows :

Algorithm : Unscented Kalman Filter

- **Initialization**
- Choose initial values for x_0^a and $\Sigma_0^{x^a}$, set accordingly $\hat{x}^a_{0|0} = x_0^a$ and $\hat{\Sigma}^{x^a}_{0|0} = \Sigma_0^{x^a}$.
- **Iteration**
- For $n = 1,\dots,t_{max}$:

  Prediction:

  - Construct the $2d_{\eta,X}+1$ sigma-points $\chi^i_{n|n}$ and their associated weights ω_i according to $\mathcal{N}\left(\hat{x}^b_{n|n}, \hat{\Sigma}^{x^b}_{n|n}\right)$, with :

$$\hat{x}^b_{n|n} = (\hat{x}^a_{n|n}, \mathbf{0}_{d_\eta}) \qquad \text{and} \qquad \hat{\Sigma}^{x^b}_{n|n} = \begin{pmatrix} \hat{\Sigma}^{x^a}_{n|n} & \mathbf{0}_{d_X,d_\eta} \\ \mathbf{0}_{d_\eta,d_X} & J_{n+1} \end{pmatrix}.$$

- Propagate the sigma-points with $\chi^i_{n+1|n} = f^a_{n+1}(\chi^i_{n|n})$ to obtain the expectations at time $n+1$:

$$\hat{x}^a_{n+1|n} = \sum_{i=1}^{2d_{\eta,X}+1} \omega_i \chi^i_{n+1|n} \quad \text{and} \quad \hat{y}_{n+1|n} = \sum_{i=1}^{2d_{\eta,X}+1} \omega_i g_{n+1}(\chi^i_{n+1|n}),$$

as well as the associated covariance matrices :

$$\hat{\Sigma}^{x^a}_{n+1|n} = \sum_{i=1}^{2d_{\eta,X}+1} \omega_i (\chi^i_{n+1|n} - \hat{x}^a_{n+1|n})^T (\chi^i_{n+1|n} - \hat{x}^a_{n+1|n}), \tag{4.24}$$

$$\hat{\Sigma}^{y}_{n+1|n} = \sum_{i=1}^{2d_{\eta,X}+1} \omega_i (g_{n+1}(\chi^i_{n+1|n}) - \hat{y}_{n+1|n})^T (g_{n+1}(\chi^i_{n+1|n}) - \hat{y}_{n+1|n}), \tag{4.25}$$

and

$$\hat{\Sigma}^{x^a y}_{n+1|n} = \sum_{i=1}^{2d_{\eta,X}+1} \omega_i (\chi^i_{n+1|n} - \hat{x}^a_{n+1|n})^T (g_{n+1}(\chi^i_{n+1|n}) - \hat{y}_{n+1|n}). \tag{4.26}$$

Correction :

- Compute the Kalman gain:

$$K_{n+1} = \hat{\Sigma}^{x^a y}_{n+1|n} \left(\hat{\Sigma}^{y}_{n+1|n} \right)^{-1}.$$

- The corrected estimator and the corresponding covariance matrix at time $n+1$ are :

$$\hat{x}^a_{n+1|n+1} = \hat{x}^a_{n+1|n} + K_{n+1}(y_{n+1} - \hat{y}_{n+1|n})^T \tag{4.27}$$

$$\hat{\Sigma}^{x^a}_{n+1|n+1} = \hat{\Sigma}^{x^a}_{n+1|n} - K_{n+1} \hat{\Sigma}^{y}_{n+1|n} K^T_{n+1}. \tag{4.28}$$

N.B.

Given the fact that f^a_{n+1} is not necessarily linear with η^a_{n+1}, we prefer to perform the transformation with the vector (X^a_n, W^a_{n+1}) instead of X^a_n described originally in the UKF algorithm.

4.3.2 Ensemble Kalman Filter

The Ensemble Kalman Filter (EnKF) (Evensen, 1994) is another extension of Kalman filter designed for nonlinear system. Established on the Monte-Carlo method coupled with the Kalman formulation, it relies on normality assumptions in order to improve the accuracy of its estimates with a more important number of samples compared to the UKF. In the EnKF, an ensemble of possible augmented hidden state vectors are randomly generated using a Monte-Carlo method to characterize the statistical properties of the vector.

Algorithm : Ensemble Kalman Filter

- **Initialization**

- Choose initial values for x_0^a and $\Sigma_0^{x^a}$, set accordingly $\hat{x}_{0|0}^a = x_0^a$ and $\hat{\Sigma}_{0|0}^{x^a} = \Sigma_0^{x^a}$.
- For $i = 1, \dots, m$, sample $x_0^{a(i)} \sim \mathcal{N}\left(\hat{x}_{0|0}^b, \hat{\Sigma}_{0|0}^{x^b}\right)$, with :

$$\hat{x}_{0|0}^b = (\hat{x}_{n|n}^a, \mathbf{0}_{d_\eta}) \qquad \text{and} \qquad \hat{\Sigma}_{0|0}^{x^b} = \begin{pmatrix} \hat{\Sigma}_{0|0}^{x^a} & \mathbf{0}_{d_X, d_\eta} \\ \mathbf{0}_{d_\eta, d_X} & J_{n+1} \end{pmatrix}.$$

- **Iteration**

- For $n = 1, \dots, t_{max}$:

  Prediction:

  - Propagate the ensemble to obtain the expectation of X_{n+1}^a given $Y_{0:n}$:

$$\hat{x}_{n+1|n}^a = \frac{1}{m} \sum_{i=1}^{m} x_{n+1|n}^{a(i)} \qquad \text{and} \qquad \hat{y}_{n+1|n} = \frac{1}{m} \sum_{i=1}^{m} g_{n+1}(x_{n+1|n}^{a(i)}),$$

  as well as the associated covariance matrices :

$$\hat{\Sigma}_{n+1|n}^{x^a} = \frac{1}{m-1} \sum_{i=1}^{m} (x_{n+1|n}^{a(i)} - \hat{x}_{n+1|n}^a)^T (x_{n+1|n}^{a(i)} - \hat{x}_{n+1|n}^a), \tag{4.29}$$

$$\hat{\Sigma}_{n+1|n}^{y} = \frac{1}{m-1} \sum_{i=1}^{m} (g_{n+1}(x_{n+1|n}^{a(i)}) - \hat{y}_{n+1|n})^T (g_{n+1}(x_{n+1|n}^{a(i)}) - \hat{y}_{n+1|n}), \tag{4.30}$$

  and

$$\hat{\Sigma}_{n+1|n}^{x^a y} = \frac{1}{m-1} \sum_{i=1}^{m} (x_{n+1|n}^{a(i)} - \hat{x}_{n+1|n}^a)^T (g_{n+1}(x_{n+1|n}^{a(i)}) - \hat{y}_{n+1|n}). \tag{4.31}$$

Correction :

- Compute the Kalman gain:

$$K_{n+1} = \hat{\Sigma}^{x^a y}_{n+1|n} \left(\hat{\Sigma}^{y}_{n+1|n} \right)^{-1}.$$

- The corrected samples of the ensemble at time $n+1$ are :

$$x^{a(i)}_{n+1|n+1} = x^{a(i)}_{n+1|n} + K_{n+1}(y_{n+1} - g_{n+1}(x^{a(i)}_{n+1|n}))^T \quad \text{for } i = 1, \ldots, m,$$

- The corrected estimator and the corresponding covariance matrix are :

$$\hat{x}^a_{n+1|n+1} = \frac{1}{m} \sum_{i=1}^{m} x^{a(i)}_{n+1|n+1}, \tag{4.32}$$

$$\hat{\Sigma}^{x^a}_{n+1|n+1} = \frac{1}{m-1} \sum_{i=1}^{m} (x^{a(i)}_{n+1|n+1} - \hat{x}^a_{n+1|n+1})^T (x^{a(i)}_{n+1|n+1} - \hat{x}^a_{n+1|n+1}). \tag{4.33}$$

4.3.3 Particle Filter

Despite the Kalman filter-based approaches, another important category of filtering methods is Particle Filter based approach. It is a recursive Bayesian filter based on Monte Carlo simulations (Arulampalam et al., 2002). Unlike Kalman filter-based approaches, Particle Filter (PF) intends to provide better approximation of the exact posterior distributions by creating a set of randomly drawn samples with each an associated time evolving weight. The main idea is that since the marginal density of $p(x^a_{0:n}|y_{0:n})$, $p(x^a_n|y_{0:n})$ is either unknown or impossible to simulate in most of the applications, as mentioned in Section 3.3, the importance sampling can be introduced to address the issue.

The importance function associated to $p(x^a_{0:n}|y_{0:n})$ is denoted $\pi_n(x^a_{0:n}|y_{0:n})$. The importance weight w_n can be defined as:

$$\forall x^a_{0:n} \in (\mathcal{X}^a)^{n+1}, \quad w_n(x^a_{0:n}) = \frac{p(x^a_{0:n}|y_{0:n})}{\pi_n(x^a_{0:n}|y_{0:n})}.$$

The MMSE estimator of the augmented hidden state vector conditioned on a series of observations is thus:

$$\hat{X}^a_n = \sum_{i=1}^{M} \tilde{w}(\tilde{x}^a_{0:n}{}^{(i)}) \tilde{x}^{a(i)}_n, \tag{4.34}$$

with $\tilde{w}_n$ the normalized importance weights and $\{\tilde{x}^a_{0:n}{}^{(i)}\}_{i=1,\ldots,M}$ the M samples (also known as trajectories) of $X^a_{0:n}$ conditional to $Y_{0:n}$, simulated from $\pi_n(x^a_{0:n}|y_{0:n})$.

One may interpret a particles as a moving point in $\mathcal{X}^a$, with $\tilde{x}^a_{0:n}{}^{(i)}$ describing the trajectory of its movement from time 0 to n. The associated weight $w_n(\tilde{x}^a_{0:n}{}^{(i)})$ is an

indicator of the likelihood of the trajectory produced compared to the real trajectory of the hidden states of the system.

For this reason, the PF algorithm is also named *Sequential Importance Sampling*. The PF based filtering methods have been studied intensively in the literature (Gordon et al., 1993; Kitagawa, 1996; Doucet et al., 2001). Among them, the most adapted to our application context can be cast into the framework of the generic Sequential Importance Sampling (SIS) algorithm, or SIR, when a resampling step is introduced (see Arulampalam et al. (2002) and Doucet et al. (2001)). The first efficient implementation of Monte-Carlo particle filters of this type dates back to Gordon et al. (1993). The authors give a generic formulation of the bootstrap PF, using importance sampling ideas in the correction step and systematic use of resampling in order to avoid the sample degeneracy problem. The term Interacting Particle Filter (IPF) is also used by Del Moral (1996) (see also Deleuze (1996); Crisan et al. (1998)) for the same PF algorithm. For other names of the same algorithm see Arulampalam et al. (2002) (page 178) and the references therein. In the following, the general algorithm of Sequential Importance Sampling is presented.

Sequential Importance Sampling

Motivated by the fact that the computation of $\{w_n(\tilde{x}_{0:n}^{(i)})\}_{i=1,\dots,M}$ can be tedious, the idea is to benefit from the Markovian feature of the state space dynamic system in order to establish a recursive relation between $w_n(\tilde{x}^a_{0:n}{}^{(i)})$ and $w_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)})$). To do this, an *importance transition density* denoted $q_{n+1}(x^a_{n+1}|x^a_n)$ is introduced which allows the particle to propagate from time n to $n+1$. This density is chosen in the way that it should be easy to sample from, which also verifies the following condition :

$$\forall x^a_{n+1} \in \mathcal{X}^a, \quad p(x^a_{n+1}|x^a_n) > 0 \quad \Rightarrow \quad q_n(x^a_{n+1}|x^a_n) > 0.$$

Therefore, the importance density can be obtained by recursion:

$$\pi_{n+1}(x^a_{0:n+1}|y_{0:n+1}) \overset{\text{def}}{=} \pi_n(x^a_{0:n}|y_{0:n}) q_{n+1}(x^a_{n+1}|x^a_n) = q_0(x^a_0) \prod_{k=0}^{n} q_{k+1}(x^a_{k+1}|x^a_k) \tag{4.35}$$

with $q_0(x^a_0)$ the importance density from which X^a_0 can be simulated.
Thanks to the Bayes' rule,

$$p(x^a_{0:n+1}|y_{0:n+1}) = p(x^a_{0:n}|y_{0:n}) \frac{p(y_{n+1}|x^a_{n+1}) p(x^a_{n+1}|x^a_n)}{p(y_{n+1}|y_{0:n})}. \tag{4.36}$$

By combining the two previous equations (4.35) and (4.36), we may obtain the importance weight in a recursive way :

$$\begin{aligned} w_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)}) &= \frac{p(\tilde{x}^a_{0:n+1}{}^{(i)}|y_{0:n+1})}{\pi_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)}|y_{0:n+1})} \\ &= w_n(\tilde{x}^a_{0:n}{}^{(i)}) \frac{p(y_{n+1}|\tilde{x}^a_{n+1}{}^{(i)}) p(\tilde{x}^a_{n+1}{}^{(i)}|\tilde{x}^{a\,(i)}_n)}{p(y_{n+1}|y_{0:n}) q_{n+1}(\tilde{x}^a_{n+1}{}^{(i)}|\tilde{x}^{a\,(i)}_n)}. \end{aligned} \tag{4.37}$$

The computation of $p(y_{n+1}|y_{0:n})$ is difficult, yet not indispensable, for the weight vector needs to be normalized before proceeding to the construction of the estimator. Therefore, the normalized weight $\tilde{w}_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)})$ can be obtained by :

$$\tilde{w}_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)}) = \frac{w_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)})}{\sum_{j=1}^{M} w_{n+1}(\tilde{x}^a_{0:n+1}{}^{(j)})} \qquad \text{pour} \quad i = 1, \ldots, M. \tag{4.38}$$

Resampling

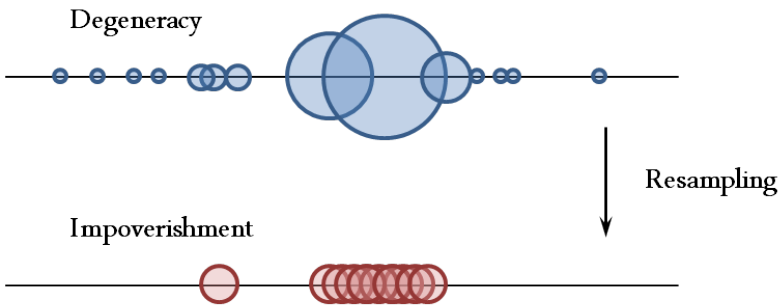


Figure 4.8: Sample degeneracy and impoverishment.

The drawback of the Particle Filter based method is that it could appear to be unstable due to the discrepancy between the particle weights. This phenomenon is known as *weight degeneracy* (Arulampalam et al., 2002)) and *sample impoverishment* (Gordon et al., 1993) (Figure 4.8). One possible solution to improve the algorithm and hopefully to avoid this problem is to introduce a resampling procedure with a sufficient frequency (Kong et al., 1994; Kitagawa, 1996). A threshold η is often chosen ($\eta = 0.8M$, see Caron (2006)) to launch the resampling procedure, which is related to the empirical variance of the particle weights.

Algorithm : Resampling

$ **Resampling** :

- Compute $N_{\text{eff}} = \left[\sum_{i=1}^{M} (\tilde{w}_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)}))^2\right]^{-1}$

- If $N_{\text{eff}} \leq \eta$:
  - For $i = 1, \ldots, M$:
    - Draw with replacement a new particle $\tilde{x}^a_{0:n+1}{}^{(i)}$ from $\{\tilde{x}^a_{0:n+1}{}^{(j)}, j \in \{1, \ldots, M\}\}$ according to a multinomial distribution

$$\mathscr{M}(1, \tilde{w}_{n+1}(\tilde{x}^a_{0:n+1}{}^{(1)}), \ldots, \tilde{w}_{n+1}(\tilde{x}^a_{0:n+1}{}^{(M)})).$$

    - Update $\tilde{w}_{n+1}(\tilde{x}^a_{0:n+1}{}^{(i)}) = 1/M$.
- Else, proceed to the next iteration.

Note that other resampling techniques also exist, for example Pham (2001); West (1993). More discussions regarding the choice of the resampling scheme for particle filtering can be found in Douc et al. (2005).

With the purpose of alleviating the undesirable side effect of resampling to improve the parametrization performance when facing restricted dataset, we opted for the Convolution Particle Filter (CPF) proposed by (Campillo and Rossi, 2009) based on the post-Regularized Particle Filter (post-RPF) (Musso and Oudjane, 1998; Oudjane and Musso, 1999; Le Gland and Oudjane, 2004).

These two filtering methods share in common a kernel smoothing method to change the discrete approximation of the filtering density.

Here, we first present briefly the kernel density estimator.

Kernel density estimator

Kernel density estimation, also known as the Parzen-Rosenblatt window method, is an approach grounded in the methodology of histogram, which aims to estimate the density function at a point x based on neighbour observations (Parzen, 1962; Silverman, 1986).

Definition 4.3.1 (Kernel) *A kernel K is defined as an application of $\mathbb{R}^d \mapsto \mathbb{R}$ which satisfies the following conditions:*

- $K(x) \geq 0, \quad \forall x \in \mathbb{R}^d$;
- *K is bounded and symmetric :* $\forall x \in \mathbb{R}^d, \quad K(x) = K(-x)$;
- *K is integrable, such that* $\int_{\mathbb{R}^d} K(x)dx = 1$.

These properties ensure that the kernel density estimation results in a probability density function. We define :

$$K_h(x) \stackrel{\text{def}}{=} \frac{1}{h^d} K(\frac{x}{h}), \qquad x \in \mathbb{R}^d,$$

with $h > 0$ a smoothing factor, which is known as the *bandwidth* parameter. Therefore, K_h is still a kernel.

Definition 4.3.2 (Parzen-Rosenblatt Kernel) *A kernel K is defined as a Parzen-Rosenblatt kernel if :*

$$|x|^d K(x) \to 0 \qquad \text{as} \quad |x| \to \infty.$$

Suppose $X_1, \ldots, X_n$ are i.i.d random variables drawn from a density distribution $f(\cdot)$ which is absolutely continuous with respect to a Lesbegue measure on $\mathbb{R}$.

Definition 4.3.3 (Kernel estimator) *(Parzen, 1962) The kernel estimator of $f_N(x)$ is defined as:*

$$\hat{f}_N(x) = \frac{1}{Nh_N^d} \sum_{i=1}^{N} K\left(\frac{x - X_i}{h_N}\right), \qquad x \in \mathbb{R}^d.$$

4.3.4 Regularized Particle Filter and Convolution Particle Filter

Regularized Particle filter

The regularization refers to the use of a kernel smoothing method to change the discrete approximation of the filtering density (induced by the weighted particles in the classical IPF) into an absolutely continuous approximation. In a way, the regularization with convolution kernels can be regarded as artificial noise. The term post-RPF is used to indicate that the regularization step takes place after the correction step. A comprehensive exposition of RPFs and improvements can be found in Musso et al. and some theoretical results in Le Gland et al. (1998) and Le Gland and Oudjane (2004). In Musso and Oudjane (1998) and Oudjane and Musso (1999), the authors compare the performance of the RPFs with the IPFs in some classical tracking problems. Let us now describe the adaptation of this method in our context.

In the initialization step of our implementation, the parameters are initialized from either informative distributions $p(x_0^a)$ or non-informative distributions for all the particles. Particle weights are assigned uniformly.

We denote by K^X the regularization kernel (usually Epanechnikov or Gaussian) associated to X_n^a and by K_h^K the rescaled kernel given by:

$$K_h^X(x^a) = h^{-d} K^X(h^{-1} x^a),$$

where $h > 0$ is the bandwidth parameter and d is the dimension of the state vector. In our application, we used the Gaussian kernel and with M particles the optimal bandwidth parameter (when the underlying density is Gaussian) is given by (see, eg., Musso et al.):

$$h_M^X = \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} M^{-\frac{1}{d+4}}. \tag{4.39}$$

Let us assume that at the end of the n-th step, the estimate of $p(x_n^a | y_{0:n})$ is given by:

$$\hat{p}(x_n^a | y_{0:n}) = \sum_{i=1}^{M} \tilde{w}_n^{(i)} K_{h_M^X}^X (x_n^a - \tilde{x}_n^{a\,(i)}),$$

where $\{(\tilde{w}_n^{(i)}, \tilde{x}_n^{a\,(i)})\}$ is the normalised weighted sample before the regularization step. The basic filtering step can be described as follows:

• **Sampling**
For $i = 1, \ldots, M$:
- Generate $I^{(i)} \in \{1, \ldots, M\}$, with $\mathbb{P}(I^{(i)} = j) = \tilde{w}_n^{(j)}$.
- Generate $\epsilon^{(i)} \sim K^X(x^a)$.
- Compute $\tilde{x}_n^{a\,(i)} = \tilde{x}_{n^-}^{a\,(I^{(i)})} + h_M^X \hat{\Sigma}_n^{1/2} \epsilon^{(i)}$, where $\hat{\Sigma}_n^{1/2}$ is the square root of the empirical covariance matrix (whitening can be used).

• **Prediction**
For $i = 1, \ldots, M$:
- Generate $\tilde{x}_{n+1}^{(i)} \sim p(x_{n+1}|\tilde{x}_n^{a\,(i)})$.
- Set $\tilde{\theta}_{n+1}^{(i)} = \tilde{\theta}_n^{(i)}$.

• **Correction and regularization**
For $i = 1, \ldots, M$:
- Set $\tilde{w}_{n+1}^{(i)} = p(y_{n+1}|\tilde{x}_{n+1}^{a\,(i)})$.
- Regularize the weighted sample $\{(\tilde{w}_{n+1}^{(i)}, \tilde{x}_{n+1}^{a\,(i)})\}$ by taking

$$\hat{p}(x_{n+1}^a|y_{0:n+1}) = \sum_{i=1}^{M} \tilde{w}_{n+1}^{(i)} K_{h_M^X}^X (x_{n+1}^a - \tilde{x}_{n+1}^{a\,(i)}).$$

As far as the regularization is concerned, it is noteworthy to mention the recent approach presented in Rossi and Vila (2006) and Campillo and Rossi (2009). The authors proposed a particle filter where a convolution kernel is used to regularize the likelihood of the observations as well. The new filter, called convolution particle filter, seems to be more appropriate to use when the likelihood cannot be computed explicitly or when signal-to-noise ratio is very low or very high (Rossi and Vila, 2006).

Convolution Particle Filter

Inspired by the post-Regularized Particle Filter (Oudjane and Musso, 1998), the objective of the Convolution Particle Filter (CPF) (Rossi, 2004; Campillo and Rossi, 2009) is also to estimate jointly the parameters and the hidden states of the dynamic system by processing the data on-line, but with both deterministic models and stochastic models. Similar to the RPF approach, the construction and the theoretical analysis of the CPF differ from the standard particle approach, as they are based on the non-parametric estimate of the conditional densities by convolution kernels.

Concerning the choice of the kernel, in our implementation, Gaussian kernels (Parzen-Rosenblatt kernel) are used, for they are more tolerant and generous with regards to the variance compared to the Epanechnikov kernel, as demonstrated by Figure 4.9.

Each filtering step is performed recurrently in two stages and occurs only at time steps when the observation is available (Campillo and Rossi, 2009):

Prediction:
The objective is to provide a kernel estimator of $p(x_{n+1}^a, y_{n+1}|y_{0:n})$ denoted by $\hat{p}(x_{n+1}^a, y_{n+1}|y_{0:n})$. M particles $\{\tilde{x}_n^{a\,(i)}, i = 1, \ldots, M\}$ are sampled from the distribution

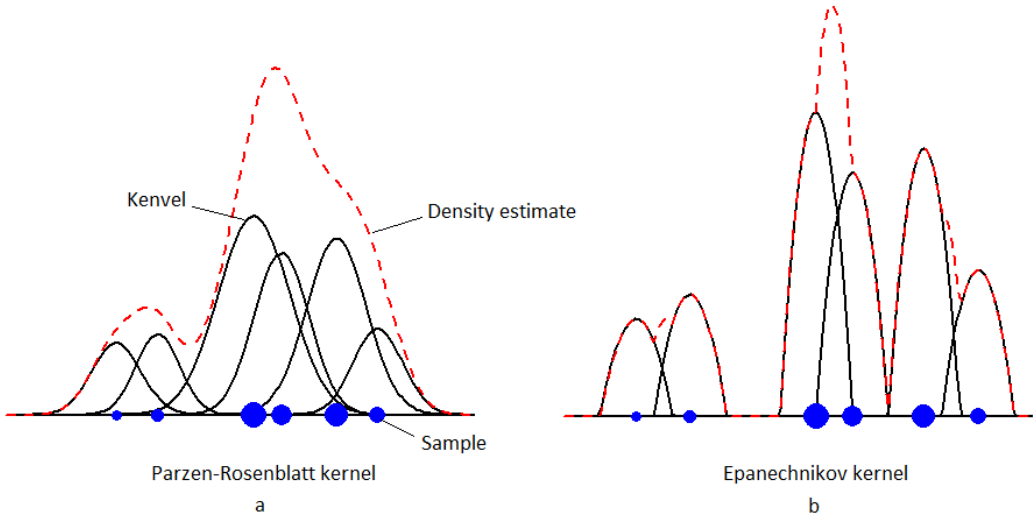


Figure 4.9: Comparison of Gaussian kernel (Parzen-Rosenblatt kernel) and Epanechnikov kernel.

with conditional density $\hat{p}(x_n^a|y_{0:n})$. The M particles are propagated through the evolution model until the next available measurement to obtain the predicted states $\{\tilde{x}^a_{n+1^-}{}^{(i)},\ i = 1,\dots,M\}$. The updating scheme relies directly on Bayes' law. The particle weights are calculated based on the experimental measurements and the predictions, and then normalized. The empirical kernel approximation of the probability density of (X^a_{n+1}, Y_{n+1}) conditional to $Y_{0:n}$ can accordingly be deduced using the Parzen-Rosenblatt kernel $K^X_{h^X_M}$, with bandwidth parameter h^X_M:

$$\hat{p}(x^a_{n+1}, y_{n+1}|y_{0:n}) = \frac{1}{M}\sum_{i=1}^{M} K^X_{h^X_M}\left(x^a_{n+1} - \tilde{x}^a_{n+1^-}{}^{(i)}\right) \cdot p\left(y_{n+1}|\tilde{x}^a_{n+1^-}{}^{(i)}\right). \tag{4.40}$$

Correction:
The *a posteriori* form of the estimation is deduced from Bayes' law and the kernel approximation for $p(x^a_{n+1}|y_{0:n+1})$ is given by:

$$\hat{p}(x^a_{n+1}|y_{0:n+1}) = \frac{\sum_{i=1}^{M} p(y_{n+1}|\tilde{x}^a_{n+1^-}{}^{(i)}) \cdot K^X_{h^X_M}(x^a_{n+1} - \tilde{x}^a_{n+1^-}{}^{(i)})}{\sum_{i=1}^{M} p(y_{n+1}|\tilde{x}^a_{n+1^-}{}^{(i)})}. \tag{4.41}$$

The part $p(y_{n+1}|\tilde{x}^a_{n+1^-}{}^{(i)})/\sum_{i=1}^{M} p(y_{n+1}|\tilde{x}^a_{n+1^-}{}^{(i)})$ can be considered as the normalized weight $\tilde{w}^{(i)}_{n+1}$ associated to the particle $\tilde{x}^a_{n+1^-}{}^{(i)}$. When the analytic form of the observation density $p(y_{n+1}|\tilde{x}^a_{n+1^-})$ is unknown, or in the case of a deterministic model, an observation kernel can similarly be introduced (Campillo and Rossi, 2009) with another Parzen-Rosenblatt kernel $K^Y_{h^Y_M}$, associated with bandwidth parameter h^Y_M:

$$\hat{p}(x^a_{n+1}|y_{0:n+1}) = \frac{\sum_{i=1}^{M} K^X_{h^X_M}(x^a_{n+1} - \tilde{x}^a_{n+1^-}{}^{(i)}) K^Y_{h^Y_M}(y_{n+1} - \tilde{y}^{(i)}_{n+1^-})}{\sum_{i=1}^{M} K^Y_{h^Y_M}(y_{n+1} - \tilde{y}^{(i)}_{n+1^-})}. \tag{4.42}$$

Sufficient conditions for the L^1-convergence of $\hat{p}(x^a_{n+1}|y_{0:n+1})$ can be found in Storvik (2002). The new set of particles $\left\{\tilde{x}^a_{n+1}{}^{(i)},\ 1 \leq i \leq M\right\}$ are then sampled from $\hat{p}(x^a_{n+1}|y_{0:n+1})$.

Algorithm : Convolution Particle Filter

- **Initialization**
- For $i = 1, \ldots, M$:
  - Choose initial values for x^a_0 from the prior distributions $p_0(\cdot)$.
- **Iteration**

  $ **Prediction step** :
  - For $i = 1, \ldots, M$:
    - Sample $\tilde{x}^a_{n+1^-}{}^{(i)} \sim p(x^a_{n+1}|\tilde{x}^{a\,(i)}_n)$
    - Sample $\tilde{y}^{(i)}_{n+1^-} \sim p(y_{n+1}|\tilde{x}^a_{n+1^-}{}^{(i)})$
    - Compute the associated weight :

$$w^{(i)}_{n+1} = K^Y_{h_M}(y_{n+1} - \tilde{y}^{(i)}_{n+1^-})$$

  $ **Normalize the weight vector** :
  - For $i = 1, \ldots, M$:
    - Set

$$\tilde{w}^{(i)}_{n+1} = w^{(i)}_{n+1} / \sum_{j=1}^{M} w^{(j)}_{n+1}$$

  $ **Correction step** :
  - Set

$$\hat{p}(x^a_{n+1}|y_{0:n+1}) = \sum_{i=1}^{M} \tilde{w}^{(i)}_{n+1} K^X_{h_M}(x^a_{n+1} - \tilde{x}^a_{n+1^-}{}^{(i)})$$

  - For $i = 1, \ldots, M$:
    - Sample $\tilde{x}^a_{n+1}{}^{(i)} \sim \hat{p}(x^a_{n+1}|y_{0:n+1})$

N.B.

- The observation Y_0 is usually not taken into account. However, it is possible to integrate it into the algorithm. To do so, for Kalman filter based methods, the initial samples (sigma-points for UKF) can be generated from time 0 according to $\mathcal{N}(x^a_0, \Sigma^{x^a}_0)$. For Particle filter based methods, we can begin with time -1 and set $p(x^a_0|\tilde{x}^a_{-1}{}^{(i)}) = p(x^a_0)$ for $i \in \{1, \ldots, M\}$.

Despite the methods presented in this chapter, other methods can also be employed to address the parameter estimation problem in a state space model, such as the Approximate Bayesian Computation (Rubin, 1984; Rau et al., 2011) and the Expectation-Maximization algorithm (Dempster et al., 1977; Andrieu and Doucet, 2003; Rau et al., 2010).

CHAPTER 5

Iterative Approaches for State-Space Model Calibration

FOR stochastic plant growth models and more generally stochastic biological models, model parameterization is generally a difficult process. Such type of models are not only characterized by a large number of interacting processes with a large number of parameters, nonlinear dynamics and scarce data for parameter estimation, resulting from costly experimental data acquisition, but also uncertainties that arise at different scales. Bayesian inference methods offer interesting perspectives to cope with such characteristics, mostly due to their robustness when confronted to scarce data, and have been used for this purpose in plant growth modeling (Wyckoff and Clark (2000), Gaucherel et al. (2008)). However, in such situations the influence of a priori distributions is prevalent (Chen and Cournède (2012)) while it is generally not easy to assess precisely.

As we all know, in the Bayesian formulation, it is assumed that the considered model is adequate to describe the phenomenon of interest. A prior distribution is then formulated over the unknown parameters, which is meant to capture our beliefs regarding the studying subject before the reception of the new information. Therefore, after updating our knowledge based on the new information, the resulting posterior distribution takes both the prior and the new data into account, from which future observations can be predicted. Thus, the selection of a prior is very important, especially when the quantity of the new information is limited. Yet many people are uncomfortable with this theoretically simple approach, simply because their view of the prior selection is arbitrary and subjective, which could possibly influence a lot the resulting distribution. Although the choice is subjective, we argue that it is however not arbitrary. It should be the one who captures the prior beliefs, which is subjective in most cases. Hence the prior should be placed modestly, with cautious. Unfortunately, many pseudo-Bayesian applications can be found in the literature, in which models, especially priors are chosen without justification and serious reflection, and

therefore, cannot represent appropriately the prior belief. For example, choosing priors for convenience or for a precise objective, varying prior according to the number of new observations, or even conducting model comparison with models that cannot be considered appropriate to describe the reality. On the other hand, undoubtedly in some situations, it can be difficult to perform properly a Bayesian method, for we may not be able to translate our prior beliefs into a adequately formulated model with sufficient prior distributions.

In the case when the prior is non-informative, and few data are available, we intend to put more weight on the data. Intuitively, one may imagine that the experiment is carried out over and over again and the same observations are obtained. Pursuing in this direction, we show in Chen et al. (2013b) that the resulting iterative approach can be regarded as an Expectation-Maximization (EM) type algorithm, at least when the prior distribution for the parameters is Gaussian. The proposed algorithm combines the robustness of Bayesian methods while preserving the prevalence of observation data and the interest of point estimation for plant growth models, in which the biophysical meaning and value of parameters are crucial. Furthermore, the term Gaussian randomization is introduced to designate that the initial model could be seen as a special case of an extended model, which includes all the Gaussian distributions with the parameters of the model as the means. In the resulting algorithm, at the beginning of each iteration, the posterior means and variances are used to initialize the updated normal distributions. By doing so, the effect of prior fades out and the estimated variances of the parameters tend to zero, we demonstrate that it converges to the Maximum Likelihood Estimator.

The chapter is organized as follows. First, we introduce the notion of Gaussian randomization and present a theoretical framework for transforming the parameter estimation problem in the initial model into a problem that can be solved with the help of an appropriate EM-type algorithm in the resulting incomplete data model. Then we detail a regularized particle filter approximation of the E-step and a MCMC approximation of the E-step of the iterative algorithm, in order to make feasible parameter estimation. Finally, we describe the averaged estimator employed and the parametric bootstrap used to provide the associated confidence interval.

5.1 Stochastic variants of an EM-type algorithm

The estimation the proposing approach is performed in the classical framework of incomplete data models via an appropriate stochastic variant of an EM-type algorithm (Dempster et al., 1977). In contrast to classical stochastic variants (Celeux and Diebolt (1985), Wei and Tanner (1990), Delyon et al. (1999), Foulley et al. (2000)) the proposed variant of EM-type algorithm is particularly adapted to the extended model resulting from the Gaussian randomization, since the last iterations of the algorithm are performed under very low variance scenarios. Some recent efforts to obtain appropriate stochastic EM-type algorithms for complex models arising in plant growth model applications can be found in Trevezas and Cournède (2013), Trevezas et al. (2013) and Baey et al. (2014). These algorithms are designed for a non-explicit M-step

and a numerical maximization algorithm is involved as well in one of the conditional maximization steps.

In this thesis, we present a simple way of turning a non-explicit M-step into an explicit one and opt for both the MCMC based methods and the SMC methods presented in Chapter 4 to perform the E-step. We mention that the use of Regularized Particle Filter was restricted until now to Bayesian type estimation.

The resulting data and parameter augmentation method enable us to estimate more easily the parameters of the initial model with the help of an iterative state estimation technique for the extended model. The term Gaussian randomization is introduced for such type of modifications. Note that the proposed method should not be confused with the Bayesian approach. The variance parameters involved in the Gaussian randomization are assumed to be unknown in contrast to the Bayesian approach in which normal priors are selected for the parameters of the model.

In the following the Gaussian Randomization and the theoretical framework of the iterative scheme is presented.

5.1.1 Gaussian Randomization

Let us consider an arbitrary parametric statistical model $m \triangleq \{(\Omega, \mathcal{A}, \mathbb{P}_\Theta); \Theta \in M\}$, where M is a Euclidean subset, and a random vector Y (representing the data vector) defined on this probability space and taking values in a measured space $\{(\mathcal{Y}, \mathcal{B}, \nu)\}$, where ν is a reference measure. Suppose that the vector Y admits a density $p(y; \Theta)$ w.r.t. ν for each $\Theta \in M$ and for a given observation $Y = y$, the likelihood function is $L(\Theta|y)$ denoted by $L(\Theta)$. The following assumption will be used in the sequel:

Assumption 1 *i) The likelihood function $L(\Theta)$ is continuous on M,*
ii) The model has a unique MLE Θ^*, i.e., for all $\Theta \in M$,

$$0 \leq L(\Theta) \leq L(\Theta^*),$$

and the second inequality is strict for $\Theta \neq \Theta^*$.

Let us suppose that maximization of $L(\Theta)$ or $\log L(\Theta)$ w.r.t. some of the parameters of Θ is very difficult to perform. In particular, let us decompose the unknown parameter $\Theta = (\Theta_1, \Theta_2)$, where Θ_1 could represent a part of the parameter vector that could not be updated explicitly in an iterative conditional maximization procedure by fixing Θ_2. The general idea consists in considering an enlarged model, by adding parameters and hidden variables to the existing model, in such a way that maximization w.r.t. to the augmented parameter vector in the enlarged model is equivalent to the initial maximization problem. In this direction, we give a first definition. Let d_x denote the dimension of a vector x. We treat the case where $\Theta_1 \in \mathbb{R}^{d_{\Theta_1}}$.

Definition 5.1.1 *Let m be a statistical model which satisfies Assumption 1 and $\Theta = (\Theta_1, \Theta_2)$, where $\Theta_1 \in \mathbb{R}^{d_{\Theta_1}}$. The statistical model $\widetilde{m}(\Theta_1)$ will be called Gaussian randomization of m w.r.t. Θ_1 if $\widetilde{m}(\Theta_1)$ is an incomplete data model, which consists in:*

i) a Gaussian hidden vector Ψ*, where*

$$\Psi \sim \mathcal{N}_{d_{\Theta_1}}(\Theta_1, \Sigma),$$

and $\Sigma = diag\{\sigma_i^2\}_{1 \leq i \leq d_{\Theta_1}}$*, where* $\sigma_i^2 > 0$*,*

ii) an observed vector Y*, where the conditional distribution of* Y *given* $\Psi = \psi$ *depends only on the parameter* Θ_2 *and satisfies:*

$$p(y|\psi; \Theta_2) = L(\psi, \Theta_2).$$

Let us take $\sigma^2 = (\sigma_i^2)_{1 \leq i \leq d_{\Theta_1}} \in (\mathbb{R}_+^*)^{d_{\Theta_1}}$ to be a minimal vector representation of the corresponding covariance matrix Σ. The parameterization corresponding to $\widetilde{m}(\Theta_1)$ is given by:

$$\phi = (\Theta, \sigma^2) = (\Theta_1, \Theta_2, \sigma^2) \in \Phi = M \times (\mathbb{R}_+^*)^{d_{\Theta_1}} \subset \mathbb{R}^{d_\phi}. \tag{5.1}$$

Let us now assume that we allow some variance parameters of Ψ to be null. This is equivalent to say that the corresponding parameters are not randomized. Denote by Θ_{11} the subvector of Θ_1 with associated strictly positive variances. According to Definition 5.1.1-i) this model is no longer a Gaussian randomization of m w.r.t. Θ_1 but rather a Gaussian randomization of m w.r.t. Θ_{11}. In order to treat in a common framework the class of all Gaussian randomizations of m w.r.t. to subvectors of Θ_1 we allow null variances in the parameterization. Most importantly, if σ^2 is assigned to the null vector $0_{d_{\Theta_1}}$, then the parameter $\phi = (\Theta_1, \Theta_2, 0_{d_{\Theta_1}})$ could be identified with the parameter $\Theta = (\Theta_1, \Theta_2)$ of the initial model m. In this way m could be understood as a submodel of $\widetilde{m}(\Theta_1)$. In the sequel we justify theoretically this intuition. Let us first state the following lemma.

Lemma 5.1.1 *The likelihood function* $\tilde{L}(\phi)$ *of the extended model* $\widetilde{m}(\Theta_1)$ *as given in Definition 5.1.1 is upper bounded by* $L(\Theta^*)$*.*

Proof: First, we compute the likelihood function $\tilde{L}(\phi)$ of the extended model $\widetilde{m}(\Theta_1)$. We have:

$$\begin{aligned} \tilde{L}(\phi) &= p(y; \phi) = \int_{\psi \in M_1} p(\psi, y; \phi) d\psi \\ &= \int_{\psi \in M_1} p(\psi; \Theta_1, \sigma^2)\, p(y|\psi; \Theta_2) d\psi \\ &= \mathbb{E}_{\Theta_1, \sigma^2}\big(p(y|\Psi; \Theta_2)\big), \end{aligned} \tag{5.2}$$

where $\Psi \sim \mathcal{N}_{d_{\Theta_1}}(\Theta_1, \Sigma)$ according to Definition 5.1.1-i). By using Definition 5.1.1-ii) and Assumption 1-ii) we get successively that for any $\phi \in \Phi$,

$$\tilde{L}(\phi) = \mathbb{E}_{\Theta_1, \sigma^2}\big(L(\Psi, \Theta_2)\big) \leq L(\Theta^*), \tag{5.3}$$

and this completes the proof. □

In order to justify the existence of the MLE in this extended model, we need to extend the likelihood function of $\widetilde{m}(\Theta_1)$ to parameter values which include null variances as explained at the beginning of this section and give the correct interpretation in this type of boundary values. We will need the following lemma.

Lemma 5.1.2 *Let $x \in \mathbb{R}^{d_x}$, $I \subset \{1, \ldots, d_x\}$ and $J = \{1, \ldots, d_x\} \setminus I$. Let also σ_I^2 and σ_J^2 be the subvectors of $\sigma^2 \in \mathbb{R}^{d_x}$ with components indexed by I and J respectively. If we consider the family of non-singular random vectors $\{X_{\sigma^2}; X_{\sigma^2} \sim \mathcal{N}_{d_x}(x, \Sigma)\}$, where $\Sigma = diag\{\sigma_i^2\}_{1 \leq i \leq d_x}$, then we have*

$$X_{\sigma^2} \xrightarrow[||\sigma_J^2|| \to 0]{weakly} X_{\sigma_I^2} \sim \mathcal{N}_{d_x}(x, \Sigma_I),$$

where Σ_I is a singular covariance matrix which results from Σ by putting $\sigma_J^2 = 0_J$. In particular, if $I = \emptyset$, then

$$X_{\sigma^2} \xrightarrow[||\sigma^2|| \to 0]{weakly} x.$$

Proof: We denote by ϕ_{σ^2}, $\phi_{\sigma_I^2}$ and ϕ_0 the characteristic functions of X_{σ^2}, $X_{\sigma_I^2}$ and the constant vector x respectively. It suffices to show that as $\sigma_J^2 \to 0_J$, ϕ_{σ^2} converges pointwise to $\phi_{\sigma_I^2}$. Indeed, let $t = (t_1, \ldots, t_{d_x})^\top \in \mathbb{R}^{d_x}$. We have

$$\phi_{\sigma^2}(t) = e^{it^\top x - \frac{1}{2} t^\top \Sigma t} \xrightarrow[||\sigma_J^2|| \to 0]{} e^{it^\top x - \frac{1}{2} t^\top \Sigma_I t} = \phi_{\sigma_I^2}(t). \tag{5.4}$$

In particular, if $I = \emptyset$, then Σ_I is the null matrix and by (5.4), ϕ_{σ^2} converges pointwise to ϕ_0. □

Proposition 5.1.3 *Let $I \subset \{1, \ldots, d_{\Theta_1}\}$ and $J = \{1, \ldots, d_{\Theta_1}\} \setminus I$. Let also Θ_I be the subvector of Θ_1 with components indexed by I, and σ_I^2, σ_J^2 and Σ_I as given in Lemma 5.1.2. We have*

$$\tilde{L}(\phi) = \tilde{L}(\Theta, \sigma_I^2, \sigma_J^2) \xrightarrow[||\sigma_J^2|| \to 0]{} \tilde{L}(\Theta, \sigma_I^2),$$

where $\tilde{L}(\Theta, \sigma_I^2)$ is the likelihood function associated with the $\widetilde{m}(\Theta_I)$ Gaussian randomization of m. In particular, if $I = \emptyset$, then

$$\tilde{L}(\phi) = \tilde{L}(\Theta, \sigma^2) \xrightarrow[||\sigma^2|| \to 0]{} L(\Theta),$$

where $L(\Theta)$ is the likelihood function associated with the initial model m.

Proof: By (5.3) the likelihood $\tilde{L}(\phi)$ is expressed as the expectation of the random variable $L(\Psi, \Theta_2)$, where $\Psi \sim \mathcal{N}_{d_{\Theta_1}}(\Theta_1, \Sigma)$. For clarity, let us denote $\{\Psi_{\sigma^2}\}$ the family of random vectors Ψ indexed by σ^2. Then, we can rewrite (5.3) as

$$\tilde{L}(\phi) = \mathbb{E}_{\Theta_1, \sigma^2}\big(L(\Psi_{\sigma^2}, \Theta_2)\big). \tag{5.5}$$

If we apply Lemma 5.1.2 for the family $\{\Psi_{\sigma^2}\}$, we have that for any (Θ_1, σ_I^2),

$$\Psi_{\sigma^2} \xrightarrow[||\sigma_J^2|| \to 0]{\text{weakly}} \Psi_{\sigma_I^2} \sim \mathcal{N}_{d_{\Theta_1}}(\Theta_1, \Sigma_I),$$

or equivalently, for any continuous and bounded function $f : \mathbb{R}^{d_{\Theta_1}} \to \mathbb{R}$,

$$\mathbb{E}_{\Theta_1, \sigma^2}\big(f(\Psi_{\sigma^2})\big) \xrightarrow[||\sigma_J^2|| \to 0]{} \mathbb{E}_{\Theta_1, \sigma_I^2, 0_J}\big(f(\Psi_{\sigma_I^2})\big), \tag{5.6}$$

where we have used the characterization of weak convergence of a family of random vectors (indexed by a continuous parameter) to a random vector in a specific limit point. By Assumption 1, the likelihood function $L(\Theta)$ is continuous on M and bounded by $L(\Theta^*)$. Consequently, for any Θ_2, the function $g : \mathbb{R}^{d_{\Theta_1}} \to \mathbb{R}$, where $\Theta_1 \mapsto L(\Theta_1, \Theta_2)$ is continuous on M_1 and bounded. Using this and (5.6), we have

$$\mathbb{E}_{\Theta_1,\sigma^2}\big(L(\Psi_{\sigma^2},\Theta_2)\big) = \mathbb{E}_{\Theta_1,\sigma^2}\big(g(\Psi_{\sigma^2})\big) \xrightarrow[||\sigma_J^2||\to 0]{} \mathbb{E}_{\Theta_1,\sigma_I^2,0_J}\big(g(\Psi_{\sigma_I^2})\big) = \mathbb{E}_{\Theta_I,\sigma_I^2}\big(L(\Psi_{I,\sigma_I^2},\Theta_J,\Theta_2)\big), \tag{5.7}$$

where Ψ_{I,σ_I^2} is the subvector of Ψ_{σ^2} indexed by I. In the last expectation $\sigma_i^2 > 0$ only for $i \in I$ and consequently the last term corresponds to the analogue of (5.5) for an equivalent representation of the likelihood function associated to the $\widetilde{m}(\Theta_I)$ Gaussian randomization of m. If $I = \emptyset$, then obviously the last term coincides with $L(\Theta)$ and the proof is complete. □

By Proposition 5.1.3 we can consider in a unified framework the class $\{\widetilde{m}(\Theta_I)\}_I$ of $2^{d_{\Theta_1}} - 1$ Gaussian randomizations of m w.r.t. to subvectors of Θ_1 together with the initial model m, which corresponds to the choice $I = \emptyset$. For this reason we extend the domain of $\tilde{L}(\phi)$, from Φ given by (5.1) to $\tilde{\Phi} = M \times (\mathbb{R}_+)^{d_{\Theta_1}}$ by allowing null variances as follows:

$$\tilde{L}(\Theta, \sigma_I^2, 0_J) = \tilde{L}(\Theta, \sigma_I^2), \tag{5.8}$$

$$\tilde{L}(\Theta, 0_{d_{\Theta_1}}) = L(\Theta). \tag{5.9}$$

With similar arguments as in the proof of Proposition 5.1.3 we can deduce from the continuity of $L(\Theta)$ on M that $\tilde{L}(\phi)$ is continuous on Φ (as $(\Theta, \sigma^2) \to (\Theta_0, \sigma_0^2) \in \Phi$ show that $(\Psi_{\Theta_1,\sigma^2}, \Theta_2)$ converges weakly to $(\Psi_{\Theta_{1,0},\sigma_0^2}, \Theta_{2,0})$). Consequently, both equations (5.8) and (5.9) determine the continuous extension of $\tilde{L}$ from Φ to $\tilde{\Phi}$ by using Proposition 5.1.3.

Theorem 5.1.4 *Let us consider the class $\widetilde{m}_{\Theta_1}$ of all Gaussian randomizations $\{\widetilde{m}(\Theta_I)\}_I$ of the initial model m, where $I \subset \{1, \ldots, d_{\Theta_1}\}$ (for $I = \emptyset$, $\widetilde{m}(\Theta_\emptyset) = m$) and Θ_I is defined in Proposition 5.1.3. If Assumption 1 holds, then the unique MLE Θ^* associated with the model m determines a unique MLE ϕ^* associated with the model $\widetilde{m}_{\Theta_1}$. In particular, we have*

$$\phi^* = (\Theta^*, 0_{d_{\Theta_1}}), \tag{5.10}$$

$$\tilde{L}(\phi^*) = L(\Theta^*). \tag{5.11}$$

Conversely, if Assumption 1-i) holds and the MLE ϕ^* associated with the model $\widetilde{m}_{\Theta_1}$ exists and is unique, then it determines a unique MLE Θ^* associated with the model m and satisfies (5.10) and (5.11).

Proof: ⇒) The restriction of $\tilde{L}(\phi)$ on $\Phi = M \times (\mathbb{R}_+^*)^{d_{\Theta_1}}$ is upper bounded by $L(\Theta^*)$ by Lemma 5.1.1. Now, if $\phi \in \tilde{\Phi} \backslash \Phi$, take the nonempty set $J = \{j \in \{1, \ldots, d_{\Theta_1}\} : \sigma_j^2 = 0\}$ and I the complement of J. Then, $\tilde{L}(\phi)$ can be expressed as the left-hand member of equations (5.8) or (5.9). By the same equations we obtain that $\tilde{L}(\phi)$ equals the

likelihood function of the extended model $\tilde{m}(\Theta_I)$ or of the initial model m respectively. Therefore, by applying Lemma 5.1.1 for Θ_I or by Assumption 1 respectively, we infer that $\tilde{L}(\phi)$ is upper bounded by $L(\Theta^*)$ also on $\tilde{\Phi} \setminus \Phi$, and consequently on $\tilde{\Phi}$. By (5.9) we have that $\tilde{L}(\Theta^*, 0_{d_{\Theta_1}}) = L(\Theta^*)$ and consequently the MLE exists and if it is unique it satisfies (5.10) and (5.11). We have that $(\Theta^*, 0_{d_{\Theta_1}}) \in M \times \{0\}^{d_{\Theta_1}}$ and by Assumption 1 and (5.9), it is unique on this set. Let us now assume that there exists $\phi = (\Theta, \sigma^2) \in M \times \left(\mathbb{R}_+^{d_{\Theta_1}} \setminus \{0\}^{d_{\Theta_1}}\right)$ such that $\tilde{L}(\Theta, \sigma^2) = L(\Theta^*)$. Take the nonempty set $I = \{i \in \{1, \ldots, d_{\Theta_1}\} : \sigma_i^2 > 0\}$ and J the complement of I. By (5.8), (5.3), the remark following (5.7) and our assumption, we have

$$\tilde{L}(\Theta, \sigma^2) = \mathbb{E}_{\Theta_I, \sigma_I^2}\big(L(\Psi_I, \Theta_2')\big) = L(\Theta^*), \tag{5.12}$$

where Θ_2' concatenates the components of Θ_1 indexed by J and Θ_2. Since by Assumption 1, the function $L(\Theta)$ is upper bounded by $L(\Theta^*)$, we obtain that the random variable $L(\Psi_I, \Theta_2')$ has the same upper bound, that is, $L(\Theta^*) - L(\Psi_I, \Theta_2')$ is a positive random variable. But, by (5.12)

$$\mathbb{E}_{\Theta_I, \sigma_I^2}\big(L(\Theta^*) - L(\Psi_I, \Theta_2')\big) = 0,$$

so we get $L(\Psi_I, \Theta_2') = L(\Theta^*)$, $\mathbb{P}_{\Theta_I, \sigma_I^2}-$a.s. , or

$$1 = \mathbb{P}_{\Theta_I, \sigma_I^2}\big(L(\Psi_I, \Theta_2') = L(\Theta^*)\big) = \mathbb{1}_{\{\Theta_2' = \Theta_2'^*\}} \mathbb{P}_{\Theta_I, \sigma_I^2}(\Psi_I = \Theta_I^*) = 0,$$

where the second equality follows by Assumption 1 and the last equality follows from the fact that Ψ_I follows a non-singular multivariate normal distribution, so it induces on $(\mathbb{R}^{d_{\Theta_I}}, \mathcal{B}(\mathbb{R}^{d_{\Theta_I}}))$ a measure which is absolutely continuous w.r.t. the Lebesgue measure which assigns measure 0 to the singleton $\{\Theta_I^*\}$. We conclude that for any $\phi \in M \times \left(\mathbb{R}_+^{d_{\Theta_1}} \setminus \{0\}^{d_{\Theta_1}}\right)$, we have $\tilde{L}(\Theta, \sigma^2) < L(\Theta^*)$ and consequently the MLE ϕ^* is unique.

$\Leftarrow$) If an MLE exists on $\tilde{\Phi}$, then $\tilde{L}$ is upper bounded on $\tilde{\Phi}$ by $\tilde{L}(\phi^*)$, and consequently is upper bounded on $M \times \{0\}^{d_{\Theta_1}}$ by the same value. By (5.9) this implies that L is upper bounded on M by $\tilde{L}(\phi^*)$. If $\phi^* \in M \times \{0\}^{d_{\Theta_1}}$, then (5.10) and (5.11) are satisfied for some $\Theta^* \in M$. Since the upper bound of L is attained on Θ^* the MLE exists. It is also unique, since if Θ_1^* and Θ_2^* are two distinct MLE, then $(\Theta_1^*, 0_{d_{\Theta_1}})$ and $(\Theta_2^*, 0_{d_{\Theta_1}})$ are two distinct MLE associated with $\widetilde{m}_{\Theta_1}$, and this contradicts the initial assumption. Now we will show that $\phi^* \in M \times \{0\}^{d_{\Theta_1}}$ is a necessary condition. Let us assume that $\phi^* \in M \times \left(\mathbb{R}_+^{d_{\Theta_1}} \setminus \{0\}^{d_{\Theta_1}}\right)$. Take the nonempty set $I = \{i \in \{1, \ldots, d_{\Theta_1}\} : (\sigma_i^2)^* > 0\}$ and J the complement of I. By applying (5.12) for ϕ^* we have

$$\tilde{L}(\phi^*) = \mathbb{E}_{\Theta_I^*, (\sigma_I^2)^*}\big(L(\Psi_I, (\Theta_2')^*)\big). \tag{5.13}$$

If $L(\Theta_I, (\Theta_2')^*) < \tilde{L}(\phi^*)$ for all $\Theta_I \in \mathbb{R}^{d_{\Theta_I}}$ then this would contradict (5.13). So, we conclude that there exists Θ_I^{**} such that $L(\Theta_I^{**}, (\Theta_2')^*) = \tilde{L}(\phi^*)$, or $\phi^{**} = (\Theta_I^{**}, (\Theta_2')^*, 0_{d_{\Theta_1}}) \neq \phi^*$ is an MLE, which contradicts the assumption of uniqueness. Therefore, $\phi^* \in M \times \{0\}^{d_{\Theta_1}}$ and the proof is complete. □

A consequence of Theorem 5.1.4 is that we can transfer the maximization problem corresponding to the initial model m to a maximization problem corresponding to the model $\widetilde{m}_{\Theta_1}$. But since the latter is formulated as an incomplete data problem, we could design an appropriate version of the EM-algorithm to solve the maximization problem or more often a stochastic variant of the EM-algorithm (Expectation-Maximization, see Dempster et al. (1977)) to handle the usually non-explicit state estimation problem. It is noteworthy that if all parameters belonging to Θ are randomized then the M-step becomes explicit. Otherwise, a GEM (G:Generalized) or stochastic GEM could be used to find the MLE. An example of this kind could be an ECM (C:Conditional), see Meng and Rubin (1993), where parameters are updated in a cyclic fashion. In the following Proposition we give the form of the Q-function, and the update formulas for the randomized parameter Θ_1.

Proposition 5.1.5 *Let $\phi' = (\Theta_1, \Theta_2, \sigma^2)' \in M \times (\mathbb{R}_+^*)^{d_{\Theta_1}}$ be the current parameter update associated with the model $\widetilde{m}_{\Theta_1}$. The update equations for the parameter (Θ_1, σ^2) are given by:*

$$\widehat{\Theta_1} = \mathbb{E}_{\phi'}\big(\Psi \mid Y = y\big) \tag{5.14}$$

$$\widehat{\sigma^2} = (\widehat{\sigma_i^2})_{1\le i\le d_{\Theta_1}} = \big(Var_{\phi'}(\Psi_i \mid Y = y)\big)_{1\le i\le d_{\Theta_1}} \tag{5.15}$$

and Θ_2 can be updated independently by maximizing

$$Q_2(\Theta_2; \phi^{'}) = \mathbb{E}_{\phi'}\big\{\log p(y|\Psi; \Theta_2) \mid Y = y\big\}. \tag{5.16}$$

Proof: Since $\phi' \in M \times (\mathbb{R}_+^*)^{d_{\Theta_1}}$, this parameter corresponds to the Gaussian randomization $\widetilde{m}(\Theta_1)$ and by the definition of the Q-function and Definition 5.1.1 we have

$$Q(\phi; \phi') = \mathbb{E}_{\phi'}\big\{\log p(\Psi, y; \Theta_1, \Theta_2, \sigma^2) \mid Y = y\big\} \tag{5.17}$$

$$= \mathbb{E}_{\phi'}\big\{\log p(\Psi; \Theta_1, \sigma^2) \mid Y = y\big\} + \mathbb{E}_{\phi'}\big\{\log p(y|\Psi; \Theta_2) \mid Y = y\big\}. \tag{5.18}$$

The last term in the right-hand member of equation (5.18) depends only on Θ_2 and coincides with (5.16), while the first term denoted by $Q_1(\Theta_1, \sigma^2; \phi^{'})$ does not depend on Θ_2. Consequently, the last statement of this theorem is true. By using the independence of the components of Ψ and the density of the normal distribution we have that Q_1 can be maximized equivalently as follows:

$$Q_1(\Theta_1, \sigma^2; \phi^{'}) \overset{\max}{\sim} -\frac{1}{2}\sum_{i=1}^{d_{\Theta_1}} \log \sigma_i^2 - \frac{1}{2}\sum_{i=1}^{d_{\Theta_1}} \frac{\mathbb{E}_{\phi'}\{(\Psi_i - \Theta_{1,i})^2 \mid Y = y\}}{\sigma_i^2}. \tag{5.19}$$

Consequently, it is elementary to show by relation (5.19) that (5.14) and (5.15) hold. □

Remark A question which arises naturally concerns the adaptation of this method in the case that some of the components of the parameter ψ have range different than $\mathbb{R}$. Then a Gaussian randomization as explained in Definition 5.1.1 cannot be applied directly and a modification is needed by suitable reparameterizations whenever possible.

At each iteration of the EM-algorithm, given the current parameter update ϕ', our objective is to approximate $\mathbb{E}_{\phi'}\big(\Theta \mid Y_{0:n} = y_{0:n}\big)$ and $\big(\mathrm{Var}_{\phi'}(\Theta_i \mid Y_{0:n} = y_{0:n})\big)_{1\leq i\leq d_{\Theta_1}}$ where n corresponds to the observation length, since the update equations for the parameters of the model given in Proposition 5.1.5 generally lead to non-explicit solutions. Several alternative algorithms can be employed to make this approximation, such as particle filters, Markov chain Monte Carlo based algorithms or Kalman-based filters.

5.2 Iterative SMC and MCMC

5.2.1 Iterative Regularized/Convolution Particle Filtering

A particular feature of the Gaussian randomization is that the optimal variance parameters equal zero (see Theorem 5.1.4). When the variances are close to zero, and this will indeed happen after some iterations of the EM-algorithm, classical particle filters cannot perform well, neither do the MCMC based methods. This is due to the fact that the particles make a discrete approximation of the filtering distribution and consequently degenerate easier under a low variance scenario due to the resampling mechanism (this is known as sample impoverishment). A successful strategy to overcome this problem was devised in Musso and Oudjane (1998) by using Regularized Particle Filters (RPF). In the light of this work, we presented in Section 4.3.4 the post-Regularized Particle Filter (Post-RPF) and the Convolution Particle Filter (CPF). First, let us explain how the filtering approach is feasible in the plant growth model calibration context.

In our application context the initial model m is a state space model and the randomized Θ can be considered as a part of the hidden state vector. The most natural way of dealing with Θ dynamically in time is to construct copies of Θ, denoted Θ_n which remain constant in time, that is, for all n, we have $\Theta_n = \Theta_{n+1}$. Despite these constant dynamics, the version of the particle filter that we describe in Section 4.3.4 enables us to modify sequentially in time the distribution of Θ given the increasingly data vector. The corresponding augmented state vector will be denoted by $X_n^a = (X_n, \Theta_n)$. In this context, all we need to approximate is $\mathbb{E}_{\phi'}\big(\Theta_n \mid Y_{0:n} = y_{0:n}\big)$ and $\big(\mathrm{Var}_{\phi'}(\Theta_{n,i} \mid Y_{0:n} = y_{0:n})\big)_{1\leq i\leq d_{\Theta_1}}$. If we have a way to approximate the intractable filtering density $p(x_n^a|y_{0:n})$ by a suitable $\hat{p}(x_n^a|y_{0:n})$, then since $x_n^a = (x_n, \theta_n)$, we can obtain an approximation of the first two conditional moments of Θ_n given the complete data vector with the help of the marginal $\hat{p}(\theta_n|y_{0:n})$.

Strategies to increase simulation numbers

An appealing feature of an EM-type algorithm is that it guarantees an improvement of the likelihood function in every iteration of the algorithm. Suppose $\phi^{(k)} = (\Theta_1, \Theta_2, \sigma^2)^{(k)} \in M \times (\mathbb{R}_+^*)^{d_{\Theta_1}}$ be the parameter update at iteration k and $\Psi^{(k)} \sim$

$\mathcal{N}_{d_{\Theta_1}}(\Theta_1^{(k)}, \sigma^{(k)^2})$. The likelihood ascent property can be described as :

$$\Delta Q_k = Q(\phi^{(k)}; \phi^{(k-1)}) - Q(\phi^{(k-1)}; \phi^{(k-1)}) > 0, \tag{5.20}$$

where $Q(\phi^{(k)}; \phi^{(k-1)})$ is defined as in (5.18). If Θ_2 remains invariant between the two iterations, then,

$$\begin{aligned}\Delta Q_k =& \mathbb{E}\big\{ \log p(\Psi^{(k)}; \Theta_1^{(k-1)}, \sigma^{(k-1)^2}) \mid Y = y\big\} \\ &- \mathbb{E}\big\{ \log p(\Psi^{(k-1)}; \Theta_1^{(k-1)}, \sigma^{(k-1)^2}) \mid Y = y\big\}.\end{aligned} \tag{5.21}$$

However, the Q-function cannot be computed explicitly, but only approximated by a finite sample size n which entails an algorithmic error disproportional to the sample size, known as Monte Carlo error. For the sake of computational tractability, the sample size is usually limited in practice. By doing so, we sacrifice accuracy for acceptable computational complexity. Yet if the fixed sample size is too small, we risk of never converging to the MLE because of considerable sampling errors (Booth et al., 2001) and in similar ways, losing the likelihood ascent property. This motivates an increase in the sample size. Some efforts have been made in the literature to increase the sample size progressively. McCulloch (1994, 1997) proposed to increase the sample size in a deterministic fashion with a fixed number of iterations, while Polyak and Juditsky (1992) used an averaging technique to optimize convergence speed. Fort and Moulines (2003) tried to accelerate the convergence by coupling both techniques. In addition, Cappé et al. (2005) fixed the final sample size which can be found useful under some situations. There are also some attempts made aiming at more flexible solutions, such as Booth and Hobert (1999); Levine and Casella (2001), who intended to increase the sample size based on the estimated Monte Carlo error.

In this context, to assure the EM's likelihood ascent property, we compare the two strategies to increases the sample size (number of simulations) automatically. One is based on a geometric increase, the other concerns a quadratic increase. The two strategies can be described as follows :

- Geometric increase: $n \leftarrow n + \lfloor n/\kappa \rfloor, \quad \kappa = 4, 5 \text{ or } 6.$
- Quadratic increase: $n \leftarrow n + 100 \cdot i^2, \quad i \leftarrow i + 1.$

Since parameter estimates tend to make large moves in the first iterations, for the sake of computational efficiency, we propose to start with a relatively small sample size. We define a evaluation criterion $Zigzag_k$ similar to ΔQ_k in order to measure the improvements made in the two last iterations :

$$Zigzag_k = \hat{Q}(\phi^{(k-1)}; \phi^{(k-1)}) - \hat{Q}(\phi^{(k-2)}; \phi^{(k-1)}) > 0, \quad k \geq 2.$$

For iteration k, if it appears that no further improvement is possible based on the current sample size by verifying $ZigZag_k < 0$ and $\Delta\hat{Q}_k < 0$, which suggests that not only no sign of improvement is detected for the past two iterations, the sample size is consequently increased.

Another way to proceed is to define an interval for the ΔQ_k function based on a normal approximation (Booth and Hobert, 1999; Caffo et al., 2005):

$$\sqrt{n}\left(\Delta\hat{Q}_k - \Delta Q_k\right) \overset{n\to\infty}{\longrightarrow} \mathcal{N}(0, \sigma_Q^2), \tag{5.22}$$

with ΔQ_k defined as in (5.21).

An approximate asymmetric confidence interval can thus be constructed (Caffo et al., 2005) :

$$[\Delta\hat{Q}_k - z_{1-\alpha}\frac{\hat{\sigma}_Q}{\sqrt{n}}; \Delta\hat{Q}_k + z_{1-\alpha'}\frac{\hat{\sigma}_Q}{\sqrt{n}}]$$

where $z_{1-\alpha}$ and $z_{1-\alpha'}$ the α-percentile and respectively α'-percentile of the standard normal distribution. A Student's t distribution can also be chosen. The σ_Q can be estimated with $\hat{\sigma}_Q$ by the method of Batch means presented in Section 4.1.5.

When $\Delta\hat{Q}_k - z_{1-\alpha}\frac{\hat{\sigma}_Q}{\sqrt{n}} < 0$, the sample size is to be increased, while if $\Delta\hat{Q}_k + z_{1-\alpha'}\frac{\hat{\sigma}_Q}{\sqrt{n}} < \delta$, where δ is a pre-defined threshold, then the algorithm can be stopped. Note that α' is usually different from α.

5.2.2 Iterative Adapted Metropolis-within-Gibbs

In the same way, the Adapted Metropolis-within-Gibbs proposed in the Section 4.1.3 can as well be employed to perform the E-step, although unlike the particle filter based methods with resampling process engaged, the MCMC based methods may potentially suffer from the sample degeneracy and impoverishment problem due to the diminishing variance. Discussion of the related issues can be found in Bordes et al. (2007).

Note that the noise parameters could be estimated in the same fashion as the functional parameter, but the estimation performance depends crucially on the mixing properties of the chain, and is often not satisfactory. More discussions regarding the uncertainty assessment can be found in Chapter 6.

As for the convergence criterion, in the case of MCMC based methods, the Monte Carlo error computed by the Batch means can be used to claim the convergence according to the precision required. Since the sample size increases naturally with the length of the Marov chain during the computation, the sample size does not need to be pre-defined by the Q-function as in the case of RPF/CPF, which can be considered as one advantage of the MCMC based methods for such an iterative scale. However, we can still use the Q-function to provide a minimum sample size which can be used as a penalty term in the definition of the stopping criterion as presented in Section 4.1.5.

5.2.3 Implementation issues

Averaged estimator

Due to the stochastic nature of this method, the averaging technique (Cappé et al., 2005) (Chap.4) is carried out to smooth and to decrease the fluctuations of the estimates after a burn-in period of K iterations. If we denote $\hat{\Theta}^{(l)}$ and $\hat{x}_n^{(l)}$the estimates of

the parameters and the hidden state variables at the l-th iteration respectively, then for $l > K$:

$$\bar{\hat{\Theta}}^{(l)} = \frac{1}{l-K} \sum_{j=K+1}^{l} w_j \hat{\Theta}^{(j)} \quad \text{and} \quad \bar{\hat{x}}_n^{(l)} = \frac{1}{l-K} \sum_{j=K+1}^{l} w_j \hat{x}_n^{(j)} \ , \tag{5.23}$$

where $w_j \propto$ the sample size (number of the particles or length of the chain) used for each E-step.

Parametric Bootstrap

As mentioned in Section 5.1.1, since the posterior distributions of the parameters are no longer representative of the estimates' uncertainty because of the iterative scheme, parametric bootstrap (Efron and Tibshirani, 1994; Bradley and Tibshirani, 1994) is implemented to provide an approximated distribution of the estimates. Therefore, we can easily deduce approximated confidence intervals for each parameter separately. The parametric bootstrap can be carried out as follows :

1. Collect a dataset $y_{0:t_{max}}$ (simulated or experimental).
2. Estimate x_n^a with some method, including the proposed iterative approaches, Bayesian approaches or frequentist approaches.
3. If the noise parameters are unknown, then update them. (For the related estimation methods, please refer to Section 6.3).
4. Generate new datasets $\{y_{0:t_{max}}^{(i)}\}$ for $i \in \{1, \dots, M\}$ with the estimated parameters ($\hat{x}_n^a$ and noise parameters if not fixed).
5. Perform the estimation based on each of them $\{y_{0:t_{max}}^{(i)}\}$, with $i \in \{1, \dots, M\}$. The obtained estimates denote $\hat{\Theta}^{(i)}$.
6. Deduce the approximated 95% confidence interval for each parameter based on $\{\hat{\Theta}^{(i)}\}$, with $i \in \{1, \dots, M\}$.

N.B.

The number of simulated datasets affects the precision of the estimated confidence interval. Commonly, at least 500 datasets are required for a 95% confidence interval. A sample of 200 could also be acceptable, when the precision requirements are not so strict.

Regarding the type of the confidence interval, in our implementation, both the bootstrap percentile interval and the bootstrap-t interval are considered. We note that other bootstrap intervals also exist, their definitions are detailed in Section 6.4.1.

Moreover, we highlight that the algorithmic uncertainty, also known as Monte Carlo error, which is associated with the stochastic algorithm and the underlying assumption made for the approximation, can also be assessed by applying the conditional ICPF approach to the same experimental data set a large number of times, as presented by Chen and Cournède (2012).

CHAPTER 6

Uncertainty Assessment in Stochastic State-Space Models

UNCERTAINTY assessment plays a crucial role when addressing parameter estimation problems and model prediction issues. When a model contains a large number of parameters, as it is often the case for plant growth models, parameter estimation based on limited experimental data is a challenging task, for parameter identifiability problems often occur. To avoid such situations, sensitivity analysis is classically applied in advance to select the most influential parameters to be estimated.

On the other hand, as mentioned in Section 2.1, any realistic model should incorporate both modelling noises and observation noises. They allow us to avoid an excessive forcing of the model towards the observations. Consequently, a complete parametrization should involve the evaluation of these noises as well as the estimation of the functional parameters. However, these two sources of uncertainty complicate greatly the statistical analysis of the model, hence, the estimation of noise parameters is a computational and statistical challenge for state-space models. In this thesis, we elaborate different strategies for the evaluation of noise parameters. Considering that in our applications, we assume only Gaussian zero-centered noises, the parameters to estimate are those that assess the variances. Note that other forms of noises could also be used, such as Poisson, negative Binomial, log-normal or Gamma, without any difficulty.

In this chapter, we first give a general presentation of sensitivity analysis and detail the method that is used in our application. Then, some noise parameter estimation methods which can be combined with the functional parameter estimation methods presented in Chapter 4 and Chapter 5 are detailed. Finally, the Bayesian credibility interval and the frequentist's confidence interval, which can be regarded as common tools to characterize the uncertainty related to the estimations are compared.

6.1 Sensitivity analysis

Biological models usually contain a large number of interacting components. Frequently, both spatial and temporal processes across multiple scales are involved. A good comprehension of the model is fundamental to ensure a correct use and an appropriate interpretation of the obtained results. For a proper model assessment, we need to quantify the model adequacy given the observation, to identify interactions between factors, to evaluate the model relevance and eventually to simplify the model structure if possible. In all these tasks, parameter sensitivity analysis can provide useful insights. Indeed, sensitivity analysis can help characterize the uncertainty underlying a model. A possible definition of sensitivity analysis is proposed by Saltelli et al. (2004):

> "*the study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input.*"

When a model contains a large number of parameters, parameter estimation based on limited experimental data may affect strongly the quality of model prediction with important estimates uncertainty. Therefore, sensitivity analysis is classically applied in advance to select the most influential parameters to be estimated, whereas those screened as the least influential ones can be fixed to any values in their domains. In the context of sensitivity analysis, this method is called "screening" or "factor fixing" (Campolongo et al., 2007).

With this purpose, we implement the algorithm proposed by Wu et al. (2012) to compute Sobol's indices (first order and total order) of all the parameters of the studied model based on its deterministic version. The generalized least-square criterion is chosen, as the output for which sensitivity analysis is performed, in the process of parameter selection.

Here, we briefly present the general principles of sensitivity analysis based on Saltelli et al. (2000) and Saltelli et al. (2004). For the simplicity of notations, let the model of interest be presented under the following form :

$$Y = f(X_1, \ldots, X_d)$$

where $\{X_1, \ldots, X_d\}$ are the input factors considered as random variables with known probability distributions, the objective is therefore to study their influence on the model outcome Y. Note that in our application, the input factors of interest are the functional model parameters.

The basic idea of Sobol's method (Sobol, 1993) is to decompose the variance of the model output Y, denoted $V(Y)$ into terms of increasing dimensionality as follows:

$$V(Y) = \sum_{i=1}^{d} V_i + \sum_{1 \leq i,j \leq d} V_{ij} + \ldots + V_{1\ldots d}$$

where

$$V_i = \mathrm{Var}(\mathbb{E}(Y \mid X_i))$$
$$V_{ij} = \mathrm{Var}(\mathbb{E}(y \mid X_i, X_j)) - V_i - V_j$$
$$\dots$$
$$V_{1\dots d} = V - \sum_{i=1}^{d} V_i - \sum_{1\le i,j\le d} V_{ij} - \dots - \sum_{1\le i_1<\dots<i_d-1\le d} V_{i_1\dots i_{d-1}}$$

The corresponding sensitivity indices are defined : the first order index $S_i = \frac{V_i}{V}$, the second order index $S_{ij} = \frac{V_{ij}}{V}$, and in the same fashion for higher order indices. The total order indices associated to factor X_i, denoted S_{T_i}, can be defined as the sum of all the indices accounting for the factor i. The first order indices allow us to quantify the proportion of $V(Y)$ that can be explained by each factor individually, whereas the second order indices aim to quantify the interaction between two factors. In other words, the difference between the total order index and the first order index for a given factor can be seen as an indication of the importance of the interactions that concern this factor. Small difference implies that the associated interactions have no significant influence on the variance $V(Y)$.

Note that if the outcome variable Y is multidimensional and time-evolving, which is often the case for plant growth models, the index $S_i^j(t)$ should be computed which represents the sensitivity to the factor (parameter) i at time t for each component of the observation vector Y, denoted Y_j, for $j = 1, \dots, n_Y$ (Wu et al., 2012). The general sensitivity index $S_i^j(t)$ can be obtained as follows :

$$S_i^j = \frac{\sum_{t=1}^{t_{max}} S_i^j(t)\mathrm{Var}(Y_j(t))}{\sum_{t=1}^{t_{max}} \mathrm{Var}(Y_j(t))},$$

which can be understood as weighting the indices $S_i^j(t)$ with the associated variance $\mathrm{Var}(Y_j(t))$. For parameter estimation, we choose as output the GLS criterion. If we denote $\tilde{Y} = f(X_1, \dots, X_d)$, then the GLS criterion can be expressed as :

$$\left(Y^{exp} - \tilde{Y}\right)^T \Sigma^{-1} \left(Y^{exp} - \tilde{Y}\right),$$

where Σ is the covariance matrix of the observation errors and Y^{exp} is the observation vector which contains the experimental data used for parameter estimation (see Section 3.4.1 for details). The parameter ranking is deduced from their total order indices.

Note that once the influence of the parameters are determined by parameter sensitivity analysis, the parameter selection problem is transformed into a model selection problem in which candidate models have different numbers of parameters: we increase the complexity of the model by introducing parameters according to their ranking. In the simplest model, only the most influential parameter is estimated while all the others are fixed to the mean values of their distributions; in the second simplest mode, the two most influential parameters are estimated; ...; in the full model, all the parameters ranked by the sensitivity analysis are estimated. Model selection criteria can be used to select the model that is best supported by the data from all the candidate models.

In the same way, model selection criteria can also be used for noise parameter selection. In the following, some methods for the noise parameter estimation problem

are presented. First, we introduce the noise parameter estimation with model selection.

6.2 Evaluation of noise parameters by model selection

The first approach consists of using a grid of modelling noise parameters fixed at different levels for selection. SMC methods as well as MCMC based methods, or frequentist methods can be performed to estimate the functional parameters of the model with different levels of noise parameters, with the aim of obtaining the best combination with respect to a given model selection criterion.

Here, we first introduce some model selection criteria frequently employed in various applications.

Let $\mathcal{L}(\hat{\theta})$ denote the likelihood function evaluated at $\hat{\theta}$. The most famous model selection criterion is the Akaike Information Criterion (AIC). Founded on information entropy, AIC evaluates the trade-off between the goodness of fit of the model and the complexity of the model (Akaike, 1974). It is widely used as model selection criterion to avoid over-fitting problems. The AIC has two components: negative log-likelihood, which measures lack of model fit to the observed data, and a bias correction factor, which increases as a function of the number of model parameters. It is defined as follows :

$$\mathrm{AIC} = -2\ (\ln \mathcal{L}(\hat{\theta}) - d),$$

where d denotes the number of parameters that are estimated. A corrected version of the AIC, known as AICc is also frequently used to account for limited sample size (Hurvich and Tsai, 1989) :

$$\mathrm{AICc} = -2\ (\ln \mathcal{L}(\hat{\theta}) - d) + \frac{2\ d\ (p+1)}{(n-d-1)},$$

in which, n denotes the sample size of the observation vector.

When the number of observations is not many times larger than d^2, AIC increases the probability of choosing models with more parameters, which can result in over-fitting problems. Therefore, Cavanaugh (1997) and Burnham and Anderson (2002) recommend strongly the use of AICc instead of AIC.

Another similar criterion based on the likelihood evaluation is the Bayesian information criterion (BIC), also known as Schwarz criterion (Schwarz, 1978). It is defined as follows :

$$\mathrm{BIC} = -2\ \ln \mathcal{L}(\hat{\theta}) + d \ln n,$$

These two criteria are commonly used for model selection objectives with point estimations (*e.g.* obtained by the MAP estimator). According to this criterion, the best model is defined as the one that is best supported by the data and obtains the smallest value compared to the other candidate models. The BIC generally penalizes more strongly the number of parameters than AIC.

A popular model selection criterion often employed in the Bayesian model comparison framework is the Deviance Information Criterion (DIC) (Spiegelhalter et al.,

1998). It can be regarded as a natural generalisation of AIC (Claeskens and Hjort, 2008). While AIC and BIC require calculating the likelihood at its maximum over θ, DIC is easily calculated from the samples generated by a Markov chain Monte Carlo simulation in the case of Bayesian model selection.

First, we introduce the *Bayesian deviance*. The investigation of the posterior distribution of the log-likelihood under each model is conducted to derive measures of fit and hence, choose the most effective number of parameters to estimate. It is defined as follows :

$$D(\theta) = -2 \ln p(y|\theta) + 2 \ln f(y),$$

$D(\theta)$ is the posterior distribution of the log-likelihood of the data (Dempster, 1974), where $f(y)$ is a function of the data alone and does not have impact on model comparison. Therefore, the fit of the model can be assessed by the conditional expectation of the deviance given y :

$$\bar{D}(\theta) = \mathbb{E}_{\theta|y}(D(\theta)), \tag{6.1}$$

while the complexity of the model can be measured by the effective number of parameters:

$$p_D = \mathbb{E}_{\theta|y}(D(\theta)) - D(\mathbb{E}(\theta|y)) := \bar{D}(\theta) - D(\bar{\theta}), \tag{6.2}$$

where $\bar{\theta} = \mathbb{E}(\theta|y)$ is the mean estimates of θ obtained from independent runs. Finally, the Deviance Information Criterion can be defined as :

$$\text{DIC} = \bar{D}(\theta) + p_D = D(\bar{\theta}) + 2p_D. \tag{6.3}$$

Like AIC and BIC, smaller values of DIC indicate a better fitting. Although designed for model selection regarding the complexity of the model (number of parameters), these criteria are used in our context to define the level of noise which fits better the data. Based on a common dataset, the same estimation method is carried out, to fit the model. Ideally, independent runs could be made to give an idea of the Monte Carlo error. The estimation is subsequently evaluated by the criteria mentioned above. Note that for AIC, AICc and BIC, it simply amounts to evaluating the likelihood of $\hat{\theta}$. In the following comparison, DIC is used to resolve the Bayesian model selection problem, whereas AIC, AICc and BIC are used to address model selection issues of frequentist based iterative approaches. Note that if multiple noise parameters are present in the model, the computational intensity is increased drastically, since there will be more possible combinations of parameter values. Therefore, this approach should be restricted to the parameters that have an identifiability problem.

In the case when no identification problem is detected, the noise parameters can be estimated jointly with the functional parameters by SMC and MCMC based methods.

6.3 Conditional SMC and MCMC for noise parameter estimation

First, we partition the full parameter vector Θ: $\Theta = (\Theta_1, \Theta_2)$, in which Θ_1 denotes the parameters corresponding to the deterministic part of the model (state equation

and measurement equation) and Θ_2 denotes those of the noise model, including the variance parameters of the modelling noises and of the observation noises (η and ξ in (3.1)).

Ideally, functional and noise parameters should be jointly estimated. However, even if theoretically this could be feasible, in practice it could be difficult to carry out. Many factors could be responsible for this inconvenience. First, adding additional variance parameters in the estimation process could result in identifiability problems, since the number of unknown parameters could be significantly increased. Second, the true purpose of introducing noise parameters is to acknowledge that variability exists in the evolution and in the observation of the system, and thus can be used to explain some residual variability that cannot be explained by the deterministic equations involved in an assumed ideal model. However, this cannot be entirely reflected in the target function if the functional and noise parameters are estimated at the same time. Thus, to estimate Θ_1 and Θ_2 in turns seems to be a more reasonable scheme to carry out the model calibration for stochastic state-space models. Note that this scheme is very classical which is also used in the GLS estimation (see for example the 2-stage Aitken estimator proposed by Taylor (1977)). We proceed the estimation as follows :

- Update $\Theta_1 \sim \pi(\Theta_1|\Theta_2, y_{1:t_{max}})$.
- Update $\Theta_2 \sim \pi(\Theta_2|\Theta_1, y_{1:t_{max}})$.

In the first place, the estimation of the hidden states and of Θ_1 is performed by considering that Θ_2 is known. Note that the estimation of the functional parameters can be conducted either with Bayesian inference, such as SMC methods or MCMC-based methods, or with frequentist-based methods, such as the stochastic EM variant that we proposed in Chapter 5 with the E-step performed with either MCMC-based methods or particle filtering based methods. From the first estimation of Θ_1 and of the hidden states, we can estimate the parameters of the distributions of the modelling and measurement noises Θ_2 from the results. Conditionally to the new estimated Θ_2, the approach is then carried out again to estimate Θ_1 together with the hidden states. In this way, the algorithm can be iterated in turns between the model parameters Θ_1 and the noise parameters Θ_2 until the convergence of both.

In practice, however, the convergence of this method is not always assured. In our implementation, small initial variances for the noises seem to be helpful to the convergence of the algorithm towards satisfactory estimation for both the hidden states variables and the functional parameters Θ_1.

Regarding the second step for noise parameter estimation, several approaches are possible and the choice remains debatable. In this thesis, we detail and examine two of them.

6.3.1 Bayesian update of noise parameters

In the Bayesian framework, instead of updating the parameter vector at once, here we propose to divide the parameter vector into two parts, Θ_1 and Θ_2 and update them in turns. For MCMC based methods, this scheme is known as *Gibbs in blocks*. Since the functional parameters Θ_1 have the priority, and we wish that they could

explain as much variability of the model as possible, given the data, we choose to start the joint updates by Θ_1. In order to keep the stability of the estimation, the update of noise parameters Θ_2 is carried out only when the estimations of Θ_1 is claimed converged conditionally to Θ_2. Moreover, conjugate prior distributions are chosen for Θ_2, so that they can be updated by simulating directly from the target distribution instead of being updated by a MH algorithm. For instance, for variance parameters corresponding to Gaussian density distributions, prior assumption of Inverse Gamma distributions can be made in order to guarantee the positivity of the sampled noises, and most importantly, they can lead to explicit updates of the posterior distributions. The detailed implementation of this method for our test case can be found in Section 8.7.

6.3.2 Empirical noise parameter estimation

If the noise parameters are defined without a conjugate prior, or a filtering method is applied to estimate the functional parameters, then an empirical noise parameter estimation is needed at the end of the filtering update. In our implementation, under the assumption that both the modelling noises $\{\eta_n\}_{0\leq n\leq t_{max}}$ and the observation noises $\{\xi_n\}_{0\leq n\leq t_{max}}$ are mutually independent Gaussian distributions (eventually multivariate Gaussian distributions), their estimators can be derived based on the estimates of the hidden states variables and the functional parameters (denoted $\hat{x}_n^a$ as the augmented hidden state vector), as well as the observation vectors $Y_{0:n}$.

For modelling noise parameters, their estimators can be defined as the averaged difference between the estimates of the hidden state variables in $\hat{x}_n^a$ and $\tilde{x}_n^a$ obtained by propagating through the evolution equation in its deterministic form $f_n^a(\hat{x}_{n-1}^a, E_{n-1}, 0)$. Respectively, for the observation noise parameters, their estimator can be defined as the averaged difference between the estimation of the outcome variables $\hat{y}_n = g_n(\hat{x}_n^a, 0)$ (deterministic observation equation) and the experimental observation Y_n.

We note that for particle filtering approaches, the estimation of the noise parameters can be carried out for each of the M particles. Therefore, the posterior distributions of the noise parameters σ_η and σ_ξ can be obtained by the samples with their associated weights w: $\{\hat{\sigma}_\eta^{(i)}, w^{(i)}\}_{1\leq i\leq M}$ and $\{\hat{\sigma}_\xi^{(i)}, w^{(i)}\}_{1\leq i\leq M}$.

Undoubtedly, other noise estimation methods also exist in literature, which are often based on the maximum likelihood functions. More discussions regarding the noise parameter estimation can be found in de Valpine (2002) and Dennis et al. (2006).

6.4 Comparison of uncertainty intervals

The proper interpretation and distinction of the Bayesian credibility intervals, bootstrap intervals and frequentist confidence intervals appears often difficult or confusing. In this section, we attempt to present their definitions, to compare their use in practice, and to sort out the interpretation problem. The problem can be formulated as follows:

Suppose that the objective is to estimate θ conditional on the observed data y and the data generation process is known $F(y|\theta)$. The inference problem is to infer the optimal value $\hat{\theta}$ of θ with respect to a given criterion and under specific assumptions for the nature of θ. We assume for the rest of this section that the unknown parameters is unidimensional. Similar interpretations can yet be given in the multi-dimensional case as well.

6.4.1 Frequentist confidence interval

In the frequentist interpretation, a 100(1-α)% confidence interval defines a characteristic of a process to generate intervals. If we denote a 100(1-α)% confidence interval $I \equiv [lb(Y); ub(Y)]$, where $lb(Y)$ and $ub(Y)$ represent the upper bound and the lower bound respectively, and generated by the random process, then ideally I should satisfy

$$P_{\theta^*}(\theta^* \in I) = 1 - \alpha,$$

where α is known as the *confidence level* and θ^* stands for the true value of the parameter. If we generate independently data under the same conditions (and under θ^*), then with a large sample size, 100(1-α)% of the estimated confidence intervals contain the true value of the parameter.

As illustrated by Figure 6.2, among 40 simulated 90% confidence intervals, 4 do not contain the true value θ^*. This is an example of the frequentist interpretation in the case that the samples are generated from a true underlying distribution.

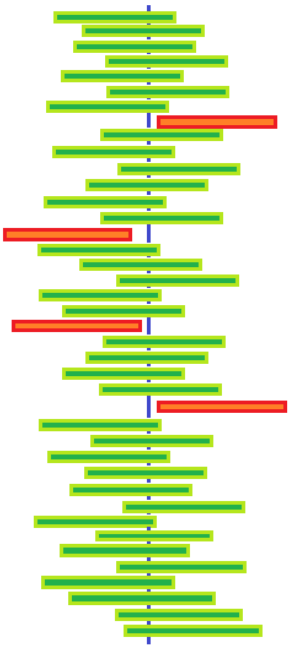


Figure 6.1: Simulated confidence intervals. The true value is represented by the vertical line.

A similar notion widely employed is the frequentist coverage.

Frequentist coverage

The frequentist coverage is calculated before the observations are gathered, therefore, it is known as pre-experimental coverage.

Definition 6.4.1
A random interval $[lb(Y); ub(Y)]$ *has 100(1-*α*)% frequentist coverage for* θ *before the new observations are made if:*

$$P_\theta(lb(Y) < \theta < ub(Y)) = 1 - \alpha.$$

Note that once the observations are gathered $Y = y$, θ^* is either contained in the interval or not, it is no longer a matter of chance.

To explain this notion, it is important to understand that for frequentist methods, a point estimate is a numerical estimate with no indication of its quality (precision, bias and variance of the estimator). If there is not a direct method to extract a 100(1-α)% confidence interval while we intend to assess the estimation quality, then we need to verify all possible random samples of the same size (n) as the experimental data $\{Y_i^{exp}\}_{i=1,\dots,n}$ that are used to obtain our point estimate in the first place. Indeed, the reasoning used in frequentist inference depends on thinking about all possible appropriate samples of the same size. Which samples are "appropriate" depend on the particular statistical procedure (model assumptions) used.

Suppose Y denotes the new random sample vector of size n, then we can define a random variable $\tilde{\theta}$ resulting from Y as an estimate of θ^*, the distribution of which is known as *sampling distribution*. The idea of frequentist coverage and confidence intervals is that we can find an appropriate k that

- The probability that $\tilde{\theta}$ lies between $\theta^* - k$ and $\theta^* + k$ is 100(1-α)%. Therefore, $\tilde{\theta}$ is random while θ^* and k are fixed.
- The previous confidence interval can be easily reformulated as the probability that θ^* lies between $\tilde{\theta} - k$ and $\tilde{\theta} + k$ is 100(1-α)%.

The resulting problem is that there are two possibilities regarding the data $\{Y_i^{exp}\}_{i=1,\dots,n}$: it can provide either a confidence interval $[\tilde{\theta} - k; \tilde{\theta} + k]$ that contains θ^* (with 100(1-α)% of the chance), or a confidence interval that does not contain θ^* (with α% of the chance). Unfortunately, it is impossible to know which possibility is true. We can only state that the procedure that we used to obtain the confidence interval works 100(1-α)% of the time.

Therefore, if one aims to achieve a post-experimental coverage, a large number of independent experiments can be carried out, and we can build one confidence interval for each of them. By the Strong Law of Large Numbers, ideally if all the intervals have 100(1-α)% coverage probability, then with a large sample size, approximately 100(1-α)% of these intervals should contain the true value of the parameter. However, the confidence intervals are many times approximate, moreover, it is not always the case that all assumptions used in deriving a confidence interval are met, therefore the actual coverage probability could be less or greater than the nominal coverage probability which corresponds to the confidence level pre-defined.

Bootstrap confidence interval

In most applications, it is extremely difficult or not possible to sample directly from the original distribution F from which $\{Y_i^{exp}\}_{i=1,\dots,n}$ were drawn from, another option is to sample from the best approximation to this distribution, based on the observed data. Let $\hat{F}$ denotes the empirical distribution function as an approximation of F. The bootstrap idea is to generate new samples from the empirical distribution function.

The construction of a bootstrap interval involves two steps with necessary assumptions for the approximations: in the Plug-In step, we consider the distribution of the sampling statistics under $\hat{F}$ instead of the distribution F to result in the bootstrap distribution, and in the Monte-Carlo step, an empirical approximation is obtained instead of the exact bootstrap distribution. There are situations that allow to compute the exact bootstrap distribution and in this case the Monte Carlo step is not needed. In the vast majority of situations the bootstrap distribution is not tractable analytically. Then simulation from the empirical distribution $\hat{F}$ yields an empirical approximation to the bootstrap distribution. This approximation can be made arbitrarily close by increasing the bootstrap sample size, given the availability of sufficient computing power. In contrast, the asymptotic equivalence of the bootstrap distribution and original distribution of the sampling statistic is far from being trivial and is based on convergence results for empirical processes.

In order to construct a confidence interval for θ we need to know how an estimator of $\hat{\theta}$ varies in repeated sampling from the population. Since all the information we have about the population is contained in the sample, bootstrap methods treat the sample as if it were the population. In the following, we present three types of bootstrap confidence intervals that are commonly used, the details of which can be found in Efron and Tibshirani (1994) and Davison and Hinkley (1997) (Chapter 5.18):

- **Bootstrap percentile interval**: $I = [q^*_{\alpha/2}; q^*_{1-\alpha/2}]$,
where $q^*_{\alpha/2}$ is the $\alpha/2$ percentile of the bootstrap distribution (respectively $1-\alpha/2$ percentile for $q^*_{1-\alpha/2}$.

- **Basic bootstrap interval**: $I = [2\hat{\theta} - q^*_{1-\alpha/2}; 2\hat{\theta} - q^*_{\alpha/2}]$,
where $\hat{\theta}$ is the estimate of θ in the original model.

- **Bootstrap-t interval**: $I = [\hat{\theta} - \hat{t}^*_{1-\alpha}\hat{SE}_{bootstrap}; \hat{\theta} - \hat{t}^*_{\alpha}\hat{SE}_{bootstrap}]$,
where $\hat{SE}$ denotes the estimate of the standard error in the original model, and $\hat{t}_\alpha$ denotes the α percentile of the bootstrapped Student's t-test : $\hat{t}^* = (\hat{\theta}^* - \hat{\theta})/\hat{SE}_{bootstrap}$, in which $\hat{SE}_{bootstrap}$ denotes the standard error of the bootstrap distribution and $\hat{\theta}^*$ denotes the bootstrap mean.

N.B.

- The difference between the bootstrap mean $\hat{\theta}^*$ and the original estimate $\hat{\theta}$ can be regarded as an estimate of the bias of $\hat{\theta}$.

Indeed, standard parametric confidence intervals can provide a measure of uncertainty related to the model parameters. Yet they require acceptance of Gaussian

assumptions for their validity. However, given the limited observation data to estimate the variability resulted from various of sources, the Gaussian assumptions may not always hold. Alternatives to the standard parametric confidence intervals are the semiparametric or nonparametric methods.

Wang et al. (1994) compared several bootstrap confidence intervals to Bayesian confidence intervals for smoothing splines. Both bootstrap and Bayesian confidence intervals are not necessarily symmetric as classical frequentist confidence intervals. They concluded that bootstrap confidence intervals work as well as Bayesian intervals concerning average coverage probability. Additionally, bootstrap confidence intervals appear to be better for small sample sizes. Based on their simulations, the "percentile-interval" bootstrap interval performs better than the other types of bootstrap intervals.

Bootstrap confidence intervals are easy to interpret and can be used with any distribution. However, because they require a large number of repetitions of the whole estimation process, their construction is computationally intensive.

6.4.2 Bayesian credibility interval

In contrast to the classical frequentist methods, a Bayesian method considers that the true θ^* is a random variable drawn from some probability distribution. The uncertainty is captured by imposing a prior knowledge and updated based on the new observations. The obtained conclusion stands only under the assumption that the prior is pertinent. The point estimate is more often obtained using the mean of the posterior distribution. Note that other estimators also exist, such as the posterior mode and the posterior median. The extent of the uncertainty is expressed by a credibility interval:

$$P(\hat{q}^{\theta}_{\alpha/2} \leq \theta \leq \hat{q}^{\theta}_{1-\alpha/2} | Y = y) = 1 - \alpha.$$

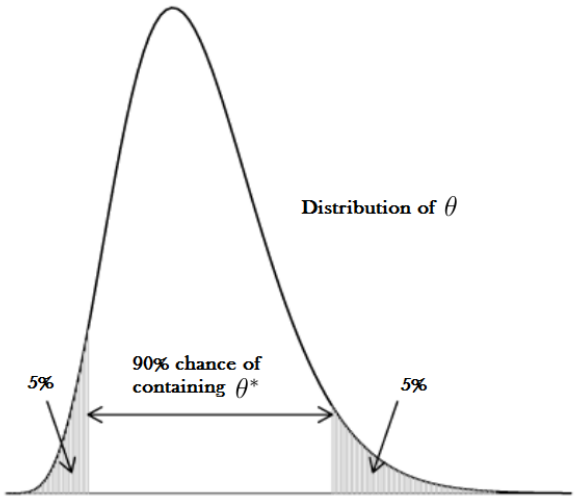


Figure 6.2: Bayesian interpretation of a 90% credibility interval.

Bayesian coverage

The Bayesian interval describes the possible location of the true parameter θ^* after the observation of y. The Bayesian coverage probability is therefore known as post-experimental.

Definition 6.4.2
An interval $[lb(y); ub(y)]$*, based on the observed data* $Y = y$ *has 100(1-*α*)% Bayesian coverage for* θ *if*

$$P(lb(y) < \theta < ub(y)|Y = y) = 1 - \alpha.$$

6.4.3 Comparisons

The comparison between the Bayesian credibility interval and frequentist confidence interval has always been a thought-provoking subject. Almost four decades ago, Jaynes (1976) examined the Bayesian and frequentist solutions to six common statistical problems to investigate the reliability of the confidence intervals provided. According to this paper, in every case without exception, the Bayesian method appeared to yield the same or better results yet easier to apply compared to the frequentist methods. Furthermore, they found that the frequentist solutions were satisfactory only when they agreed closely with those provided by the Bayesian methods.

But is it true that the Bayesian's choice is absolutely better?

It actually depends on our objective. In fact, a frequentist confidence interval is not defined with the aim of obtaining the highest probability to contain the real value of the parameter given what we can infer from the observations. Unfortunately, in practice this is often the way that the confidence interval is interpreted, which can be misleading. Therefore, if the comparison is about finding an interval where the real value of the statistics lies with probability $100(1-\alpha)\%$, then the frequentist confidence interval may not answer that question directly (which entails the problems stated in Jaynes' paper), while the Bayesian solution can. Hence the conclusion is that the Bayesian credibility interval is superior to the frequentist one in that aspect. If the former objective can be regarded as the "wrong" one for the frequentist methods, then the right objective would be to obtain an interval in which if the experiment is repeated a large number of times, then the true value of the statistic lies with $100(1-\alpha)\%$ of all the intervals obtained. In such cases, the Bayesian's solution may also be able to give an answer, although it may not be directly the credibility interval (Hartigan, 1966).

CHAPTER 7

Computational implementation with PyGMAlion

In plant growth modelling, a large variety of models coexist in the literature with generally an absence of benchmarking between the different approaches and insufficient model evaluation. In this context, the PyGMAlion (Plant Growth Model Analysis, Identification and Optimization) platform (Cournède et al., 2013) can be regarded as an attempt to promote a modelling framework and model analysis methods with the objectives of enhancing good modelling practices and increasing model design efficiency. In this chapter, this simulation platform on which all the methods presented in this thesis are implemented is presented. Some implementation issues are subsequently discussed.

Note that all the simulations and numerical tests presented in this thesis were carried out on the mesocenter of Ecole Centrale Paris, which has a total capacity of over 10 TFLOPS and comprises nearly 1000 calculation units [1]. All the parallel computations were realized with up to 10 cores.

7.1 Motivation

Undoubtedly, classical numerical and statistical software such as MATLAB®, R or WinBugs, generally provide efficient and sophisticated methods for model analysis, especially for parameter estimation. However, the platform proposed is not always adequate and convenient for the development and analysis of complex models featured by specific data structures (different from vectors or matrices), which generally need to be developed by the user, as well as specific encapsulation methods in order to make the generic analysis and estimation methods applicable. The proposed packages

1. http://www.mesocentre.ecp.fr/

provide algorithms that are operated like a black box, which often appear to be hard to be reconfigured. Moreover, various limitations are imposed by the software cited previously, especially regarding the computational resources. Indeed, computationally intensive statistical models and methods need corresponding computation facility to scale. To improve the computational performances, the use of distributed computation is an opportunity (parallel and multi-threading computation) which can be difficult to carry out with the above limitations.

Under these circumstances, the PyGMAlion C++ platform offers a framework in which the implementation of stochastic or deterministic discrete dynamic models and statistical methods can be carried out easily. All the algorithms presented in this thesis including the sensitivity analysis, uncertainty analysis, parameter estimation, model selection or data assimilation are thus implemented on this platform, so that model comparisons and method comparisons can be conducted to provide benchmarks for the different approaches, among other utilities.

7.2 State-space model formulation

To perform Monte Carlo based methods based on state-space models, first, we need to translate them into exploitable code for simulation, the following reformulations are made (Bayol et al., 2013).

- The dynamical evolution system of a meta model $\mathcal{M}$ is defined as a 6-tuple (for which the general formulation is detailed in Section 3.2):
$$\{X_n, E_n, \Theta_1, \Theta_2, INIT, NEXT\}$$
where :
  - $\{X_n, E_n, \Theta_1\}$ denote respectively the state variables, the control variables and the functional parameters involved in both the evolution step and the observation step.
  - The noise parameter vector Θ_2 is used to generate the modelling noises $(\eta_n)_n$ and the observation noises $(\xi_n)_n$ of the dynamical system, for each random sequence a pseudo-random number generator is associated.
  - $INIT$ and $NEXT$ describe respectively the initialization function for the state variables X_0 and the evolution function of the dynamical system model f_n of a meta model $\mathcal{M}$.
- An observation model is defined as a 4-tuple $OM_i = \{X_n, \Theta_1, \Theta_2, Y_n^i\}$, with Y_n^i the ith outcome variable.
- An observer is defined as a 2-tuple $O_i = \{OM_i, TML_i\}$ where TML is a timeline associated to OM_i which control the trigger of global observation of dynamical system during the evolution.

Thus the global system is denoted by a 8-tuple
$$S = \{X_n, E_n, \Theta_1, \Theta_2, INIT, NEXT, Y_n, TML\}.$$

N.B.

Instead of defining a common observer for all the outcome variable Y_n, we choose to personalize one observer for each component of Y_n in order to account for their irreg-

ularity and diversity, such as different types of observation noise, different timelines and various output forms.

7.3 Computation issues and strategy

A crucial justification of one simple Monte Carlo estimation is the assumption that we are capable of generating a set of independent random values with distribution π.

7.3.1 Random number generation

The assumption that one can create a set of values with independent π distributions has become more viable with the advent of computing power and is crucial in the justification of simple Monte Carlo estimation. The task of generating a set of independent values with distribution π is usually divided into two parts :

1. Generate a set of values $\{U(i)\}_{i=1,\dots,n}$ considered i.i.d. from $\mathcal{U}[0,1)$.
2. Transform $U(i)$ to yield $\{X(i)\}_{i=1,\dots,n}$ which follow distribution π.

Much attention has been given to this first step which is the goal of a random number generator. With the growing needs of working with increasing size of samples, sophisticated algorithms have been designed and studied to produce a sequence that approximates the properties of random numbers and is practically indistinguishable from independent uniform samples. These algorithms are known as *Pseudo-Random Number Generators* (PRNG), the advantages of which are their reproducibility and efficiency in number generation.

Generally, they operate in a recursive way such that the next number generated is determined by the last several numbers. The sequence is not truly random for it is determined by a relatively small set of initial values, in which a *random seed* needs to be required.

Another key point is that the recursive generator must be periodic, as each number is identified to finite precision. In the case when the sample size is larger than the period of the generator, the illusion of independence is impossible to maintain. Hence, the sample size should not exceed the square root of the period.

In this work, the PRNG used is called *Mersenne Twister* (Matsumoto and Nishimura, 1998). It has a period of $2^{19937} - 1$ and optimal equidistribution property in 623-dimensional output blocks to 32-bit accuracy.

For the simulations studied in this thesis, the assumption of i.i.d random variables taken from this generator appears reliable. However, more other generators can be used for comparison's sake, so as to evaluate the performance and to ensure the functionality of each generator. Further tests can therefore be carried out following this direction.

7.3.2 Implementation of Inverse-Gamma priors

Since the prior distribution types are limited by the basic distributions proposed by the library, to use an Inverse-Gamma prior, a transformation is required to compute the

corresponding expectation and standard deviation of the Gamma distribution provided by the boost C++ library. Similar problem has been encountered in other software development, more discussion of which can be found in Adjemian (2010).

Definition 7.3.1 *Let X be a Gamma distributed random variable with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$. Then $Z = X^{-1}$ can be seen as Inverse-Gamma distributed, $Z \sim IG(\alpha, \beta)$.*

Proposition 7.3.1 *The density of the continuous random variable $Z \sim IG(\alpha, \beta)$ is:*

$$f(z) = \mathcal{C}(\alpha, \beta)^{-1} \times z^{-\alpha-1} \exp^{-\frac{1}{\beta z}}$$

where $\mathcal{C}(\alpha, \beta) = \Gamma(\alpha)\beta^{\alpha}$ is the constant of integration.

Proof
Let f_x denote the density of the Gamma distribution. If we define $g(x) = \frac{1}{x}$, the density of the Inverse-Gamma distribution can be written as follows:

$$\begin{aligned} f_Z(z) &= f_X(g^{-1}(z)) \times |\frac{d}{dz} g^{-1}(z)| \\ &= \mathcal{C}(\alpha, \beta)^{-1} \times z^{-\alpha+1} \exp^{-\frac{1}{\beta z}} \times |\frac{d}{dz}\frac{1}{z}| \\ &= \mathcal{C}(\alpha, \beta)^{-1} \times z^{-\alpha-1} \exp^{-\frac{1}{\beta z}} . \end{aligned}$$

□

Proposition 7.3.2 *The expectation and variance of $Z \sim IG(\alpha, \beta)$ are:*

$$\begin{aligned} \mu &= \frac{1}{(\alpha-1)\beta} \\ \sigma^2 &= \frac{1}{(\alpha-1)^2(\alpha-2)\beta^2} \qquad \text{for any} \quad \alpha \geq 2. \end{aligned} \tag{7.1}$$

Proof
According to the pdf of the Inverse-Gamma, the first moment is

$$\begin{aligned} \mu &= \mathcal{C}(\alpha, \beta)^{-1} \int_0^{\infty} z^{-\alpha} \exp^{-\frac{1}{\beta z}} dz \\ &= \mathcal{C}(\alpha, \beta)^{-1} \int_0^{\infty} u^{\alpha-2} \exp^{-\frac{u}{\beta}} du \\ &= \frac{\mathcal{C}(\alpha-1, \beta)}{\mathcal{C}(\alpha, \beta)} \\ &= \frac{\Gamma(\alpha-1)\beta^{\alpha-1}}{\Gamma(\alpha)\beta^{\alpha}} \\ &= \frac{\Gamma(\alpha-1)\beta^{\alpha-1}}{(\alpha-1)\Gamma(\alpha-1)\beta^{\alpha}} \\ &= \frac{1}{(\alpha-1)\beta}, \end{aligned}$$

and the second moment is defined by

$$\begin{aligned}
\mathbb{E}[Z^2] &= \mathcal{C}(\alpha,\beta)^{-1} \int_0^\infty z^{-\alpha+1} \exp^{-\frac{1}{\beta z}}\, dz \\
&= \mathcal{C}(\alpha,\beta)^{-1} \int_0^\infty u^{\alpha-3} \exp^{-\frac{u}{\beta}}\, du \\
&= \frac{\mathcal{C}(\alpha-2,\beta)}{\mathcal{C}(\alpha,\beta)} \\
&= \frac{\Gamma(\alpha-2)\beta^{\alpha-2}}{\Gamma(\alpha)\beta^{\alpha}} \\
&= \frac{\Gamma(\alpha-2)\beta^{\alpha-2}}{(\alpha-1)(\alpha-2)\Gamma(\alpha-2)\beta^{\alpha}} \\
&= \frac{1}{(\alpha-1)(\alpha-2)\beta^2},
\end{aligned}$$

so that the variance

$$\begin{aligned}
\sigma^2 &= \frac{1}{(\alpha-1)(\alpha-2)\beta} - \frac{1}{(\alpha-1)^2\beta^2} \\
&= \frac{\alpha-1}{(\alpha-1)^2(\alpha-2)\beta^2} - \frac{\alpha-2}{(\alpha-1)^2(\alpha-2)\beta^2} \\
&= \frac{1}{(\alpha-1)^2(\alpha-2)\beta^2}.
\end{aligned}$$

□

Therefore, according to the desired the mean μ and the variance σ^2 of the parameter, the shape α and scale β parameters can be defined as follows:

$$\begin{aligned}
\alpha &= 2 + (\frac{\mu}{\sigma})^2, \\
\beta &= \frac{1}{\mu[1 + (\frac{\mu}{\sigma})^2]}.
\end{aligned}$$

Note that for the implementation of Gibbs sampler, this calculation is useful when an Inverse-Gamma conjugate prior is required for the noise parameter σ^2_{obs} update (for details, see Section 8.7.1).

7.3.3 Parallel simulations

To achieve the parallel computation of Monte Carlo simulations, a vector of simulations (S) is required. It can be regarded as a basic tool for statistical analysis methods, such as filtering methods, parallel MCMC, sensitivity analysis and uncertainty analysis. Figure 7.2 illustrates the flow of the Convolution Particle Filter, with each particle representing an individual simulation S.

For each simulation, a PRNG is associated to each of its noises (modelling noises or observation noises). For the filtering algorithms which contain a resampling process, some simulations are duplicated during the exploration of the state-space, therefore, a

new random seed needs to be initiated to avoid the duplication of the same sequence of random variables shared by two simulations.

A multi-threaded or distributed implementation is carried out for the filtering methods, the parallel MCMC based algorithms and sensitivity analysis algorithms.

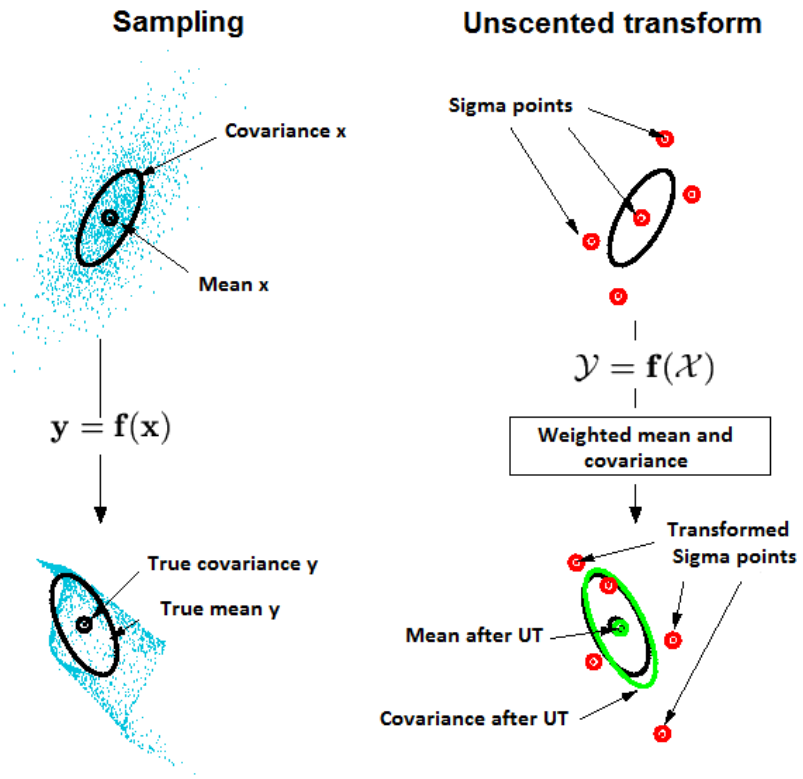


Figure 7.1: System evolution of the Unscented Kalman Filter [2].

2. http://www.cslu.ogi.edu/nsel/ukf/node6.html

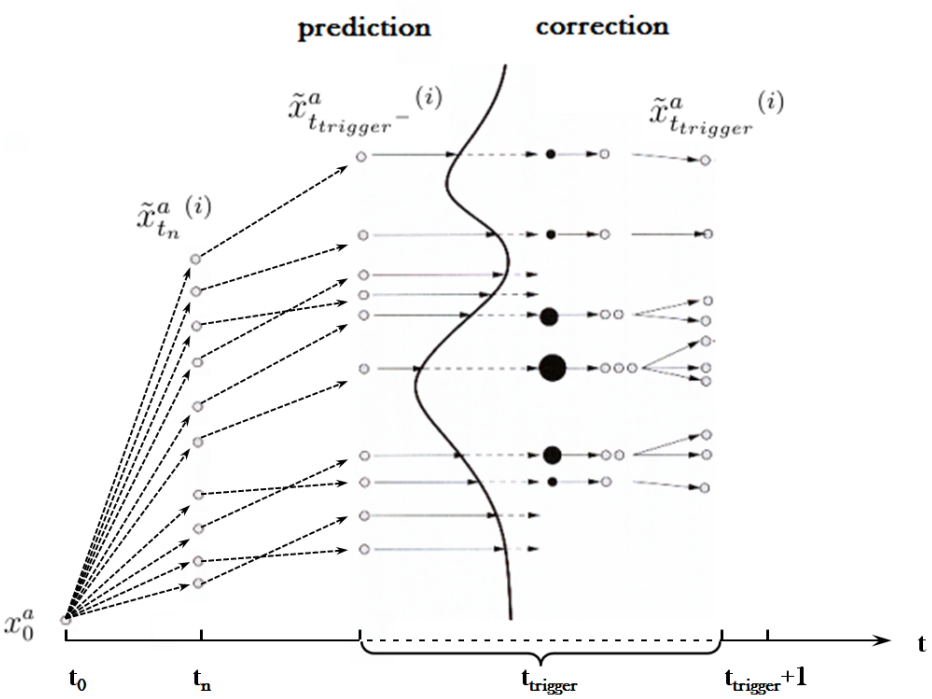


Figure 7.2: System evolution of the Convolution Particle Filter.

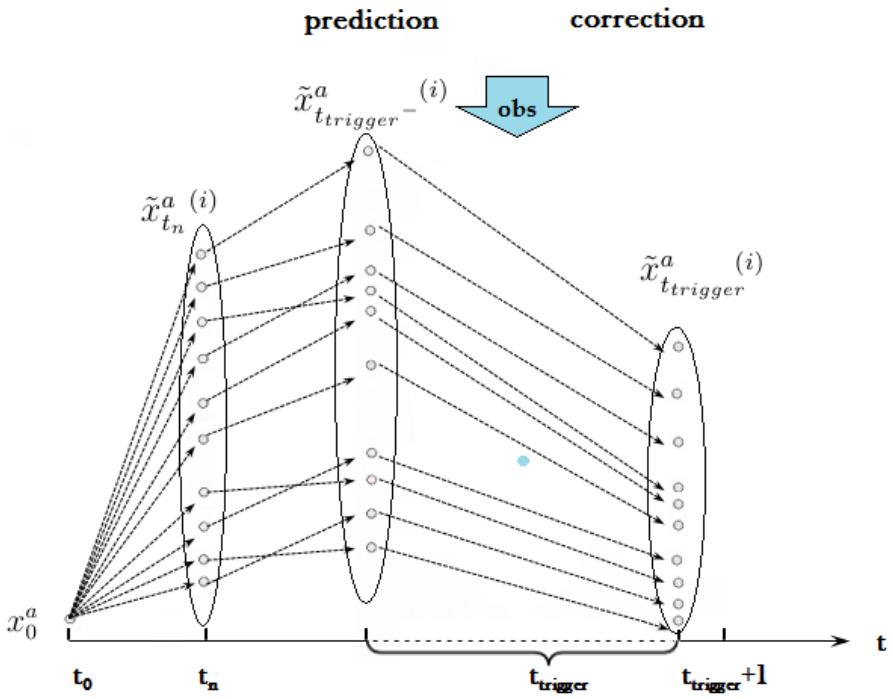


Figure 7.3: System evolution of the Ensemble Kalman Filter.

7.4 Ongoing work and perspective

From a computational point of view, the objective is to provide a stronger parallel implementation for distributed computing. So far, the parameter estimation methods like the Convolution particle filter, the Differential Evolution Adaptive Metropolis, the Interacting Metropolis-within-Gibbs as well as the algorithms to perform sensitivity analysis, such as the Sobol algorithm benefit from the distributed computing via MPI (Message Passing Interface) or OpenMP (Open Multiprocessing) approaches. The objective is to enable the possibility of distributed computation at more elementary levels (each time step for example) which should elevate the computational performance to a new level.

More generally, the platform should evolve towards an EDSL (embedded domain-specific languages), with the objective to ease for the users the development of models, the calls of analysis methods and the use of parallel architecture for computation.

CHAPTER 8

Parameter Estimation in Plant Growth Models

Plant growth models or crop models are generally deterministic and can be described in a state-space form. One can easily get their stochastic counterparts by introducing modelling and measurement noises. Such a statistical framework allows a joint estimation of parameters and hidden states. However, some difficulties arise in the implementation, not only from the uneven and irregular measurement data, but also from sudden or unusual climate changes that occasionally induce skewness, which is difficult to account for in models.

To compare the performances of different estimation methods in terms of accuracy for point estimation and appropriateness for uncertainty assessment, the LNAS model presented in Chapter 2 is used. In this chapter, the tests are organized in two parts. For the first part, only simulated datasets are tested. The performance of the Bayesian methods presented in Chapter 4 are investigated based on their estimation of the model functional parameters. Subsequently, the iterative approaches proposed in Chapter 5 are implemented and compared, followed by the implementation of a conditional version of the iterative approach which allows noise parameter estimation. In the second part, all the methods aforementioned are applied to real experimental datasets for an evaluation of their estimation quality of both the functional and the noise parameters. The results are finally discussed together with some method selection strategies. Note that all the parallel computation and simulations presented in this chapter were carried out with the mesocenter of Ecole Centrale Paris with up to 10 cores.

8.1 Study case description

With the purpose of testing the performances of the proposed algorithms, we first consider the virtual experimental datasets obtained by performing simulations based on the true experimental conditions of one experiment conducted by the French institute for sugar beet research (ITB, Paris) in 2010 (see Baey et al. (2012) for a detailed description of the experimental protocol). Here, we present the results of simulations in two different scenarios :

(i) the full dataset is observed, that is, at each day from day 1 to day 160, both the green leaf biomass Q_g and the root compartment biomass Q_r are measured, which result in a full observation vector of size 2×160,

(ii) only a restricted dataset is available, for which Q_g and Q_r are measured at 14 dates which correspond exactly to the following 14 dates of measurement $\mathcal{O}_{2010}$ in the 2010 experiment (in days after sowing)

$$\mathcal{O}_{2010} = \{54, 68, 76, 83, 90, 98, 104, 110, 118, 125, 132, 139, 145, 160\}\,.$$

The restricted observation vector is thus of size 2×14. In this way, we can assess the consequences of missing information in parameter estimation.

8.1.1 State space model representation

The LNAS model is therefore reformulated in the form of a state-space model, also known as Hidden Markov Model (Rabiner, 1989):

$$\begin{cases} X(t+1) = \Phi_t(X(t), \eta(t); \Theta) \\ \\ Y(t) = R_t(X(t), \xi(t); \Theta), \end{cases} \tag{8.1}$$

where $X(t) = (X_1(t), X_2(t), X_3(t)) = (Q_f(t), \gamma(t), Q_r(t))$, $\{\eta(t)\}_{t\geq 1}$ and $\{\xi(t)\}_{n\geq 1}$ are appropriately chosen sequences of random vectors. Note that $X(0)$ is determined by a first allocation of the seed biomass between the root and leaf compartments (corresponding to germination). The parameter vector of our model is denoted by Θ and includes six functional parameters $\{\mu_a, \lambda, \mu_\gamma, \sigma^2_\gamma, \gamma_0, \gamma_f\}$ chosen according to the sensitivity analysis (details of which can be found in Section 6.1), and eventually some noise parameters (error variances) $\{\sigma^2_{\gamma\gamma}, \sigma^2_q, \sigma^2_g, \sigma^2_r\}$. However, according to the DIC for model selection (details of which can be found in Section 6.2), when the noise parameters are fixed, the best model is obtained by estimating only three functional parameters $\{\mu_a, \mu_\gamma, \gamma_0\}$. This configuration is therefore adopted for the simulation case study.

In the following lemma we explicit the state equation and the observation equation corresponding to our model.

Lemma 8.1.1 *The model given by Equations (2.1)-(2.7) can be described as a state space model (also known as hidden Markov model). The state equation is given by the*

following system of equations :

$$\begin{cases} X_1(t+1) = X_1(t) + X_2(t) \cdot F_Q^t(X_1(t);\Theta)(1+\eta_Q(t)) \\ X_2(t+1) = \Gamma^t(\Theta)(1+\eta_\gamma(t)) \\ X_3(t+1) = X_3(t) + \dfrac{1-X_2(t)}{X_2(t)} \cdot F_Q^t(X_1(t);\Theta)(1+\eta_Q(t)), \end{cases} \tag{8.2}$$

where $\{\eta_Q(t)\}_{t\geq 1}$ *and* $\{\eta_\gamma(t)\}_{t\geq 1}$ *are mutually independent sequences of independent zero-centered Gaussian random vectors with variance* σ_Q^2 *and* $\sigma_{\gamma\gamma}^2$ *respectively.*

The observation equation is given by :

$$Y(t) = R_t(X(t),\xi(t)) = \begin{pmatrix} F_g^t(X_1(t);\Theta) \\ X_3(t) \end{pmatrix} \cdot \begin{pmatrix} (1+\xi_g(t)) \\ (1+\xi_r(t)) \end{pmatrix}, \tag{8.3}$$

where $\{\xi_g(t)\}_{t\geq 1}$ *and* $\{\xi_r(t)\}_{t\geq 1}$ *are i.i.d. sequences of centered Gaussian random variables with variances* σ_g^2 *for* $\xi_g(t)$ *and* σ_r^2 *for* $\xi_r(t)$*, and assumed to be independent from* $\{\eta_Q(t)\}_{t\geq 1}$ *and* $\{\eta_\gamma(t)\}_{t\geq 1}$*.*

Proof

$X(t)$ is a Markov process. First, we prove that the observation vector can be determined by $(X_1(t), X_2(t), X_3(t)) = (Q_f(t), \gamma(t), Q_r(t))$, Θ and the noise vector $\xi(t)$. According to the LNAS model,

$$\begin{aligned} Q_r(t+1) &= F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta) \\ &= Q_r(t) + (1-\gamma(t)) \cdot Q(t) \\ &= Q_r(t) + \frac{1-\gamma(t)}{\gamma(t)}(Q_f(t+1) - Q_f(t)), \end{aligned} \tag{8.4}$$

and

$$Q_g(t) = F_g^t(Q_f(t);\Theta) = Q_f(t) - Q_s(t),$$

where

$$Q_s(t) = G_s(\tau(t) - \tau_{sen})Q_f(t),$$

in which $G_s(\tau(t)-\tau_{sen})$ is determined by the functional parameters μ_s and σ_s, which are related to the senescence process and can be estimated only based on the senescent leaf mass. These two parameters are thus fixed in our study. Consequently, the observation vector $Y(t) = (Q_g(t), Q_r(t))$ depends only on $X(t)$, Θ and the noise vector $\xi(t)$.

Then, we prove that in the evolution equation, $X(t+1)$ depends only on $X(t)$, Θ and the noise vector $\eta(t)$. Since

$$Q_f(t+1) = Q_f(t) + \gamma(t) \cdot Q(t),$$

$$\begin{aligned} Q(t) &= F_Q^t(Q_f(t);\Theta) \cdot (1+\eta_Q(t)) \\ &= \left(\mu_a \cdot PAR(t)\left(1 - e^{-\lambda Q_g(t)}\right)\right) \cdot (1+\eta_Q(t)), \end{aligned}$$

and

$$\begin{aligned}\gamma(t) &= \Gamma(t) \cdot (1 + \eta_\gamma(t)) \\ &= (\gamma_0 + (\gamma_f - \gamma_0) \cdot G_a(\tau(t))) \cdot (1 + \eta_\gamma(t)),\end{aligned}$$

if we denote $X(t) = (\gamma(t), Q_f(t), Q_r(t))$, we can obtain the transition function :

$$\begin{aligned}p\left(X(t+1)|X(t)\right) &= p\left((\gamma(t+1), Q_f(t+1), Q_r(t+1))|(\gamma(t), Q_f(t), Q_r(t))\right) \\ &= p(\gamma(t+1))p(Q_f(t+1)|\gamma(t), Q_f(t))\delta_{Q_r(t+1),\phi(Q_f(t+1),\gamma(t),Q_f(t),Q_r(t))},\end{aligned} \tag{8.5}$$

where the first two components have a density (w.r.t. the 2-dimensional Lebesque measure) and the third component, a Dirac delta function, results from a deterministic move from the current state X(t) and Q_f(t+1) according to Equation (8.4). $p(\gamma(t+1))$ and $p(Q_f(t+1)|\gamma(t), Q_f(t))$ are densities of normal distributions which can be described as follows:

$$Q_f(t+1) \sim \mathcal{N}\left(Q_f(t) + \gamma(t)F_Q^t(Q_f(t);\Theta), \quad \gamma^2(t){F_Q^t}^2(Q_f(t);\Theta)\sigma_Q^2\right), \tag{8.6}$$

$$\gamma(t) \sim \mathcal{N}\left(\Gamma^t(\Theta), \quad {\Gamma^t(\Theta)}^2\sigma_{\gamma\gamma}^2\right). \tag{8.7}$$

Hence, $X(t+1)$ depends only on $X(t)$, Θ and the noise vector $\eta(t)$. □

N.B.

- It is important to note that since $Q_r(t+1)$ is obtained from a deterministic function of $Q_f(t+1)$, $\gamma(t)$, $Q_f(t)$ and $Q_r(t)$, the support of the transition density is in $\mathbb{R}^2$ instead of $\mathbb{R}^3$.

- We point out that for the state variables representing biomasses, we are under the positivity assumptions which could theoretically be violated by the normal distributions. Although in practice, the probability of obtaining $\gamma(t) > 1$ or a negative value for the biomasses is very small, Equation (8.5) remains an approximation. Note that the adaptation of the formulation of the state space model with truncated normal distributions or log-normal distributions would be straightforward, and so would be the adaptation in the implementation section.

8.1.2 Selection of priors

In this thesis, from an application point of view, priors of the considered plant growth models are often chosen according to recommended values from literature. Even in the tests with simulated datasets, we opt for truncated Gaussian priors centered on the recommended values with relatively large variances. However, in the case that no recommended values are available, either a more dispersed prior or a uniform one is used. In the case when an important number of data are available, the prior effect is supposed to fade out. To accelerate the convergence, the starting point(s) of the chain(s) for MCMC-based methods can be determined with the help of a first calibration of the model performed with a frequentist method, such as the Generalized

Least Squares Estimator (Cournède et al., 2011), for this method is computationally low cost and provides an estimate close to the MLE.

8.2 Implementation of MCMC-based methods

In this section, we apply three MCMC-based methods: Adapted Metropolis-within-Gibbs, Differential Evolution Adaptive Metropolis and Interacting parallel MCMC to the LNAS model. They represent respectively the three strategies of MCMC-based methods: "one long run", "some median runs" and "many short runs".

In the first place, we detail the implementation process of these methods, including the formulation of the full conditionals, the initialization and the convergence criterion. Then, their estimation results based on simulated datasets are given and discussed according to their configuration features. Accordingly, some implementation issues are discussed.

8.2.1 Implementation descriptions

Formulation

Let $\pi(\Theta) = \pi(\mu_a, \mu_\gamma, \gamma_0)$ be the joint prior distribution of the LNAS model parameters. First, we consider the case that the noise parameters of the model are known.

Given the fact that $\pi(\Theta|y_{1:t_{max}})$ is analytically intractable, the hidden state variables are introduced to provide an analytical expression, and consequently make the numerical implementation easier. A joint estimation of both unknown parameters and hidden states is thus required. Given the target $p(\Theta, x_{1:t_{max}}|y_{1:t_{max}})$, the joint updates' scheme derived from the Gibbs sampling can be described as follows (Cappé et al., 2005):

- Update $\Theta \sim \pi(\theta|x_{1:t_{max}}, y_{1:t_{max}})$.
- Update $X_{1:t_{max}} \sim \pi(x_{1:t_{max}}|\Theta, y_{1:t_{max}})$.

However, because of the poor mixing between Θ and $X_{1:t_{max}}$ (as described in Section 4.1.4), we opt for a joint update for Θ and $x_{1:t_{max}}$:

- Update $\{\Theta, X_{1:t_{max}}\} \sim \pi(\theta, x_{1:t_{max}}|y_{1:t_{max}})$.

In our application, since the observation vector $Y(t) = (Q_g(t), Q_r(t))$ can be expressed as a function of $X(t) = (Q_f(t), \gamma(t), Q_r(t))$, the latter are chosen as the hidden state variables in the implementation of the three MCMC-based methods. We recall that among the three hidden state variables, the value of $Q_r(t)$ is obtained in a deterministic fashion based on $Q_r(t-1), Q_f(t-1), Q_f(t)$ and $\gamma(t-1)$ (see Lemma 8.1.1 for details).

Therefore, based on Equations (8.7) and (8.6), the full likelihood of the parameter vector Θ and the hidden state variable $x_{1:n}$ given the observation $y_{1:n}$ can be described as :

$$
\begin{aligned}
\mathcal{L}(\Theta, x_{1:n}|y_{1:n}) &\propto p(x_{1:n}|\Theta)p(y_{1:n}|\Theta, x_{1:n}) \\
&\propto \prod_{t=0}^{n-1} p(X(t+1)|X(t),\Theta) \cdot \prod_{t=0}^{n-1} p(Y(t+1)|X(t+1),\Theta) \\
&\propto \prod_{t=0}^{n-1} \frac{1}{\sigma_Q \gamma(t) F_Q^t(Q_f(t);\Theta)} \cdot \exp\{-\frac{\left(Q_f(t+1) - Q_f(t) - \gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}{2\left(\sigma_Q\gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}\} \\
&\quad \prod_{t=1}^{n} \frac{1}{\sigma_{\gamma\gamma}\Gamma^t(\Theta)} \cdot \exp\{-\frac{(\gamma(t)-\Gamma^t(\Theta))^2}{2(\sigma_{\gamma\gamma}\Gamma^t(\Theta))^2}\} \cdot \prod_{t=0}^{n-1} \delta_{Q_r(t+1),\phi(Q_f(t+1),\gamma(t),Q_f(t),Q_r(t))} \\
&\quad \prod_{t=0}^{n-1} \frac{1}{\sigma_g F_g^{t+1}(Q_f(t+1);\Theta)} \cdot \exp\{-\frac{(F_g^{t+1}(Q_f(t+1);\Theta) - Y_g^{exp}(t+1))^2}{2(\sigma_g F_Q^{t+1}(Q_f(t+1);\Theta))^2}\} \\
&\quad \prod_{t=0}^{n-1} \frac{1}{\sigma_r F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta)} \\
&\qquad \cdot \exp\{-\frac{(F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta) - Y_r^{exp}(t+1))^2}{2(\sigma_g F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta))^2}\}.
\end{aligned}
$$

Note that the term $\prod_{t=0}^{n-1} \delta_{Q_r(t+1),\phi(Q_f(t+1),\gamma(t),Q_f(t),Q_r(t))}$ represents the fact that $Q_r(t)$ is deduced in a deterministic fashion.

The posterior density of the parameter vector Θ can thus be expressed as :

$$
\begin{aligned}
\pi(\Theta, x_{1:n}|y_{1:n}) &\propto \mathcal{L}(\Theta, x_{1:n}|y_{1:n}) \cdot \pi_\Theta(\Theta) \\
&\propto \mathcal{L}(\Theta, x_{1:n}|y_{1:n}) \cdot \pi_{\mu_a}(\mu_a)\pi_{\mu_\gamma}(\mu_\gamma)\pi_{\gamma_0}(\gamma_0),
\end{aligned}
$$

where $\pi_\Theta(\Theta)$ denotes the density of the prior distribution for Θ.

Truncated normal prior distributions are assumed for the parameters $(\mu_a, \mu_\gamma, \gamma_0)$:

$$
\begin{aligned}
\mu_a &\sim \mathcal{N}(m_{\mu_a}, \sigma^2_{\mu_a}) I_{\mathbb{R}_+}, \\
\mu_\gamma &\sim \mathcal{N}(m_{\mu_\gamma}, \sigma^2_{\mu_\gamma}) I_{\mathbb{R}_+}, \\
\gamma_0 &\sim \mathcal{N}(m_{\gamma_0}, \sigma^2_{\gamma_0}) I_{(0,1)}.
\end{aligned}
$$

If we denote $\Phi(\cdot)$ the cumulative distribution function of the standard normal distribution :

$$
\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{t^2}{2}\} dt,
$$

then $1 - \Phi(a)$ represents the normalizing factor in the probability density function of a truncated normal distribution, with support in $[a, +\infty)$.

As a result, the posterior distribution in the general implementation of a MCMC-based

method should have the following form :

$$
\begin{aligned}
\pi(\Theta, x_{1:n}|y_{1:n}) \propto & \prod_{t=0}^{n-1} \frac{1}{\sigma_Q \gamma(t) F_Q^t(Q_f(t);\Theta)} \\
& \cdot \exp\{-\frac{\left(Q_f(t+1) - Q_f(t) - \gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}{2\left(\sigma_Q\gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}\} \\
& \prod_{t=1}^{n} \frac{1}{\sigma_{\gamma\gamma}\Gamma^t(\Theta)} \cdot \exp\{-\frac{(\gamma(t)-\Gamma^t(\Theta))^2}{2(\sigma_{\gamma\gamma}\Gamma^t(\Theta))^2}\} \cdot \prod_{t=0}^{n-1} \delta_{Q_r(t+1),\phi(Q_f(t+1),\gamma(t),Q_f(t),Q_r(t))} \\
& \prod_{t=0}^{n-1} \frac{1}{\sigma_g F_g^{t+1}(Q_f(t+1);\Theta)} \cdot \exp\{-\frac{(F_g^{t+1}(Q_f(t+1);\Theta) - Y_g^{exp}(t+1))^2}{2(\sigma_g F_Q^{t+1}(Q_f(t+1);\Theta))^2}\} \\
& \prod_{t=0}^{n-1} \frac{1}{\sigma_r F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta)} \\
& \cdot \exp\{-\frac{(F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta) - Y_r^{exp}(t+1))^2}{2(\sigma_g F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta))^2}\} \\
& \cdot \frac{1}{\sigma_{\mu_a}\sigma_{\mu_\gamma}\sigma_{\gamma_0}} \cdot \exp\{-\frac{(\mu_a - m_{\mu_a})^2}{2\sigma_{\mu_a}^2} - \frac{(\mu_\gamma - m_{\mu_\gamma})^2}{2\sigma_{\mu_\gamma}^2} - \frac{(\gamma_0 - m_{\gamma_0})^2}{2\sigma_{\gamma_0}^2}\} \\
& \cdot \left(1 - \Phi(\frac{-m_{\mu_a}}{\sigma_{\mu_a}})\right)^{-1} \left(1 - \Phi(\frac{-m_{\mu_\gamma}}{\sigma_{\mu_\gamma}})\right)^{-1} \\
& \cdot \left(\Phi(\frac{1-m_{\gamma_0}}{\sigma_{\gamma_0}}) - \Phi(\frac{-m_{\gamma_0}}{\sigma_{\gamma_0}})\right)^{-1} \\
& \cdot I(\mu_a > 0) \cdot I(\mu_\gamma > 0) \cdot I(\gamma_0 \in (0,1)).
\end{aligned} \tag{8.8}
$$

One important issue is that due to the complicated structure, it is almost impossible to sample directly from the posterior distribution. Thus, the Metropolis-Hastings algorithm is implemented. In the computation of the acceptance ratio, the three normalizing factors $1-\Phi(\cdot)$ remain constant and as a result can be cancelled. According to the algorithm given in Section 4.1.4, we regard $\{\Theta', x'_{1:n}\}$ as a joint proposal, thus the acceptance ratio can be expressed as (details can be found in Equation (4.10)):

$$
\begin{aligned}
& \alpha(\{\Theta, x_{1:n}\}, \{\Theta', x'_{1:n}\}) \\
= & \min\left(1, \frac{p(y_{1:n}|\Theta', x'_{1:n})p(x'_{1:n}|\Theta')\pi(\Theta')}{p(y_{1:n}|\Theta, x_{1:n})p(x_{1:n}|\Theta)\pi(\Theta)} \frac{q(\Theta|\Theta')q(x_{1:n}|\Theta)}{q(\Theta'|\Theta)q(x'_{1:n}|\Theta')}\right).
\end{aligned} \tag{8.9}
$$

Considering the strong correlation between Θ and $x_{1:n}$, if we choose $q(x'_{1:n}|\Theta') = p(x'_{1:n}|\Theta')$, then Equation (8.9) can be simplified :

$$
\begin{aligned}
& \alpha(\{\Theta, x_{1:n}\}, \{\Theta', x'_{1:n}\}) \\
= & \min\left(1, \frac{p(y_{1:n}|\Theta', x'_{1:n})\pi(\Theta')}{p(y_{1:n}|\Theta, x_{1:n})\pi(\Theta)} \frac{q(\Theta|\Theta')}{q(\Theta'|\Theta)}\right).
\end{aligned} \tag{8.10}
$$

Ideally, the random walk proposal distribution for Θ should be truncated, therefore to obtain the acceptance ratio it requires to calculate:

$$\begin{aligned}\frac{q(\Theta|\Theta')}{q(\Theta'|\Theta)} &= \frac{\frac{1}{\sigma_{RW}\sqrt{2\pi}}\exp\left(\frac{-(\Theta-\Theta')^2}{2\sigma_{RW}^2}\right)}{1-\Phi\left(\frac{-\Theta'}{\sigma_{RW}}\right)} \cdot \left(\frac{\frac{1}{\sigma_{RW}\sqrt{2\pi}}\exp\left(\frac{-(\Theta'-\Theta)^2}{2\sigma_{RW}^2}\right)}{1-\Phi\left(\frac{-\Theta}{\sigma_{RW}}\right)}\right)^{-1} \\ &= \frac{1-\Phi\left(\frac{-\Theta}{\sigma_{RW}}\right)}{1-\Phi\left(\frac{-\Theta'}{\sigma_{RW}}\right)}, \end{aligned} \tag{8.11}$$

where σ_{RW}^2 denotes the variance for the proposal distribution of the random walk. However, for the sake of computational efficiency, we may also consider the non-truncated symmetric normal random walk distribution, thus $\frac{q(\Theta|\Theta')}{q(\Theta'|\Theta)}$ =1. In this case, the positivity of Θ is still ensured thanks to the prior distribution, for negative candidate will obtain 0 as acceptance probability.

Let $\tilde{Q}_g(t)$ denote the estimate of $Q_g(t)$ which can be obtained by a deterministic function of the estimates of $Q_f(t)$, and let $\tilde{Q}_r(t)$ denote the estimate of $Q_r(t)$, to compute the acceptance ratio, we need to calculate :

$$\begin{aligned} & p(y_{1:n}|\Theta, x_{1:n}) \cdot \pi(\Theta) \\ \propto & \prod_{t=0}^{n-1} p(Y(t+1)|X(t+1),\Theta) \cdot \pi_{\mu_a}(\mu_a)\pi_{\mu_\gamma}(\mu_\gamma)\pi_{\gamma_0}(\gamma_0) \\ \propto & \prod_{t=1}^{n} \frac{1}{\sigma_g \tilde{Q}_g(t)} \cdot \exp\left\{-\frac{(\tilde{Q}_g(t)-Y_g^{exp}(t))^2}{2(\sigma_g\tilde{Q}_g(t))^2}\right\} \\ & \prod_{t=1}^{n} \frac{1}{\sigma_r \tilde{Q}_r(t)} \cdot \exp\left\{-\frac{(\tilde{Q}_r(t)-Y_r^{exp}(t))^2}{2(\sigma_r\tilde{Q}_g(t))^2}\right\} \\ & \cdot \frac{1}{\sigma_{\mu_a}\sigma_{\mu_\gamma}\sigma_{\gamma_0}} \cdot \exp\{-\frac{(\mu_a-m_{\mu_a})^2}{2\sigma_{\mu_a}^2} - \frac{(\mu_\gamma-m_{\mu_\gamma})^2}{2\sigma_{\mu_\gamma}^2} - \frac{(\gamma_0-m_{\gamma_0})^2}{2\sigma_{\gamma_0}^2}\} \\ & \cdot I(\mu_a>0)\cdot I(\mu_\gamma>0)\cdot I(\gamma_0\in(0,1)), \end{aligned} \tag{8.12}$$

with $Y_r^{exp}(t)$ and $Y_g^{exp}(t)$ the observation data for Q_g and Q_r respectively at time t.

In the following, we detail in the first place the implementation of the one chain Adapted Metropolis-within-Gibbs algorithm.

Initialization of the algorithms

The initialization of the MCMC based algorithms can be carried out as follows :

- Define prior distributions for each parameter of Θ: $p_0(\cdot)$ with the expectation μ_{p_0} (normal distributions).
- Set $\Theta_0^{(1:d)}$. Simulate one value for each parameter from the prior distribution

while verifying that they are all positive: $\Theta \sim p_0(\cdot)$. The noise parameters are fixed, and the hidden state variables $x_0^{(0)}$ are initialized to 0.
– Choose initial values for μ_0, Σ_0 and λ_0 for the adaptive scheme. In our application, we set $\mu_0 = \Theta_0^{(1:d)}$, $\Sigma_0 = (2.38^2/d)(\mu_0 - \mu_{p_0})(\mu_0 - \mu_{p_0})^T$ and $\lambda_0 = 1$ if the global scaling is used.

Convergence criteria

The convergence criteria for the **one chain Adapted Metropolis-within-Gibbs** algorithm are detailed in Section 4.1.5, putting thresholds on both the relative difference between the overlapping batch means and the precision (i.e. the width of the confidence intervals) of the mean estimates determined by the estimated Monte Carlo errors. In our implementation, 4 batches of length 3000 with overlapped length 1000 are used for the convergence diagnostic and unfixed (increasing) batch length is used to compute the precision of the estimates.

As for the parallel chains algorithm **DREAM**, the Gelman-Rubin criterion is used. A threshold is defined for the scale parameter $\sqrt{\hat{R}}$, in which $\hat{R}$ is the estimated ratio of the current estimate of variance to the within-chain variance corrected by a factor accounting for the extra variance considered from the Student's t distribution. In our implementation, 1.0001 is chosen. More details about the Gelman-Rubin criterion can be found in Section 4.2.3.

Finally, for the **Interacting parallel MCMC**, a similar stopping criterion as the convergence diagnostic based on mean estimates for one chain Metropolis-within-Gibbs algorithm is employed, except the mean estimates are computed from the posterior distribution which consists of the samples from the last iteration of all the chains.

8.2.2 Adapted Metropolis-within-Gibbs

The one chain Adapted Metropolis-within-Gibbs (AMwG) algorithm represents the "one long run" strategy in our implementation. In this section, some tests are conducted to explore various configurations and to investigate the robustness of this method.

List of the tests for the AMwG algorithm :

I	Decomposition of variance
II	Latent variable estimation
III	Multivariate Vs. componentwise proposals
IV	Number of iterations and estimation precision
V	Proposal scaling effect on the convergence
VI	Impact of priors and initial values
VII	Normal proposal Vs. Student proposals

General settings :

Here we provide the standard configuration for all the tests if no contradictory details are given in the setting descriptions. The modelling noise parameters and the observation noise parameters are fixed to 0.02 and 0.1 respectively. Multivariate proposals are used with global scaling. Chains with the length of 350 000 are considered (if the length is fixed), for this length usually ensures the convergence in our context.

I Decomposition of variance

The aim of this test is to evaluate the stability of the proposed AMwG algorithm, both in terms of the mean estimates and the variance estimates. If the algorithm is well implemented, then it should respect the law of total variance :

$$\mathbb{V}(\Theta) = \mathbb{V}(\mathbb{E}(\Theta|Y)) + \mathbb{E}(\mathbb{V}(\Theta|Y)).$$

Settings :

To achieve this purpose, 600 parameter sets are generated from a given distribution $p_0(\cdot)$, the corresponding observation datasets are as well simulated. The prior distribution used for these tests is the same as the one used to generate the datasets :

$$\forall i \in 1, \ldots, 600, \Theta_i^* \sim p_0(\cdot) \Rightarrow \{Y_i\}_{i=1,\ldots,600} \xrightarrow{p_0(\cdot)} \{\tilde{\Theta}_i\}_{i=1,\ldots,600}$$

where $\tilde{\Theta}_i = \hat{\mathbb{E}}(\Theta|Y_i)$ and the approximation $\hat{\mathbb{E}}$ of the expectation $\mathbb{E}(\Theta|Y_i)$ is performed with the ergodic mean of the samples generated by the AMwG algorithm for each dataset Y_i.

Observations and remarks :

	Noise 0.02		Prior		Estimation	Variance decomposition		
	μ_0^*	σ_0^{2*}	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))+\mathbb{E}(\mathbb{V}(\Theta\|Y))$
μ_a	3.595	**0.0104**	3.600	0.0100	3.597	0.0069	0.0035	**0.0104**
γ_0	0.7519	**0.0060**	0.7500	0.0064	0.7513	0.0058	0.0003	**0.0061**
μ_γ	599.92	**401.78**	600.00	400.00	599.92	99.42	303.87	**403.29**

	Noise 0.05		Prior		Estimation	Variance decomposition		
	μ_0^*	σ_0^{2*}	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))+\mathbb{E}(\mathbb{V}(\Theta\|Y))$
μ_a	3.596	**0.0102**	3.600	0.0100	3.599	0.0067	0.0038	**0.0105**
γ_0	0.7520	**0.0062**	0.7500	0.0064	0.7518	0.0058	0.0004	**0.0062**
μ_γ	599.81	**400.32**	600.00	400.00	599.96	69.31	333.92	**403.23**

Table 8.1: Estimation and uncertainty assessment provided by one chain MCMC based on 600 simulated datasets (14 dates) generated with the parameter vectors simulated from the same prior distribution. Each test contains 350 000 simulations. Two modelling noise levels are tested. μ_0^* and σ_0^{2*} denote the sample mean and variance.

– The conditional expectation formula and the variance decomposition are verified which is a good indicator of the proper assessment of the uncertainty related to the estimations (bias and variance of the target distribution).

– With different modelling noise levels, we may infer from the results given by Table 8.1 that the noise influences differently the within chain (intra-chain) variation $\mathbb{E}(\mathbb{V}(\Theta|Y))$ and between chain variation $\mathbb{V}(\mathbb{E}(\Theta|Y))$ as estimated by the algorithm. The more the modelling noise is important, the more the intra-chain variance increases, and the more the between chain variance decreases.

– Based on the sensitivity analysis (see Section 9.2.2), the estimates of the most sensitive parameter γ_0 appear to be less influenced by the increase of the modelling noises than μ_a and μ_γ.

II Latent variable estimation

The objective is to study the impact of the lack of available data on the estimation of the hidden state variables (also known as *latent variables*).

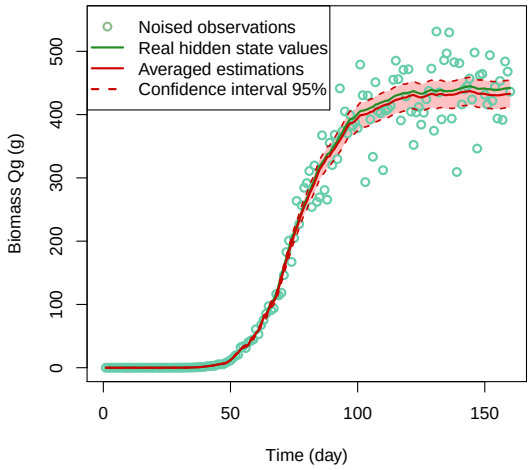
Settings :

The same simulated dataset is used for the estimation of the hidden state variables Q_r and Q_g (which can be inferred from the estimation of Q_f with a deterministic function) with the setting of 14 dates (restricted dataset) and 160 dates (full dataset).

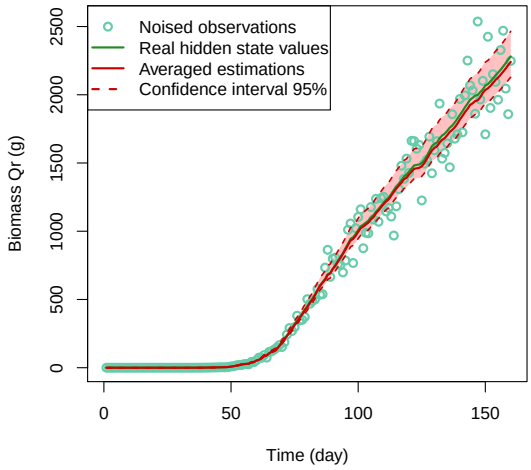
Observations and remarks :

– According to Figure 8.1, the mean estimates remain quite accurate in both cases, which proves the robustness of the proposed algorithm.

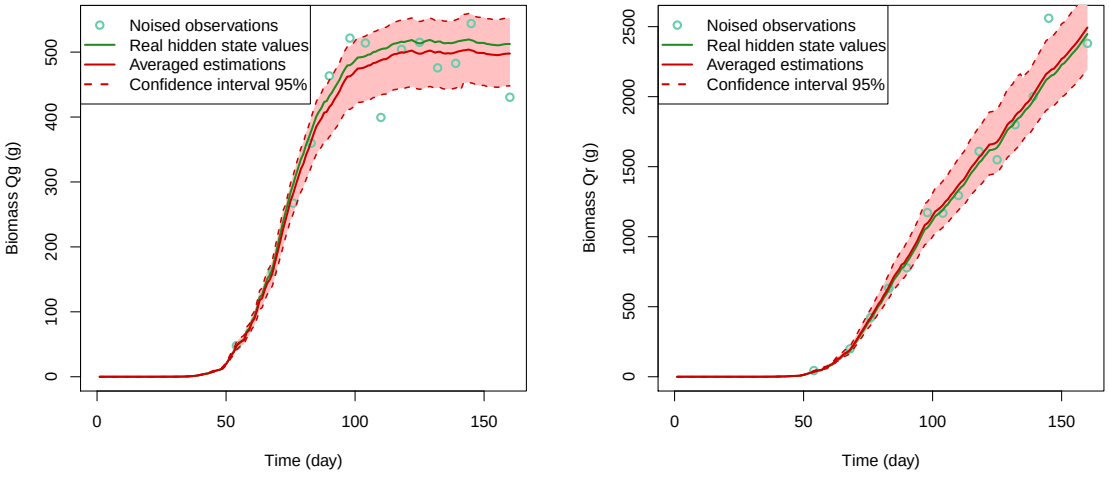
– The lack of data brings not only larger estimation errors, but most importantly more uncertainty associated with the estimation of the hidden states. The estimated 95% confidence intervals are significantly larger in the case of the scarce dataset compared to the full dataset.



(a) Q_g (full dataset)



(b) Q_r (full dataset)



(c) Q_g (restricted dataset) (d) Q_r (restricted dataset)

Figure 8.1: Hidden states estimation given by one chain MCMC Adapted Metropolis-within-Gibbs algorithm based on a full dataset (160 dates) and on its restricted form (14 dates) with 350 000 iterations. Both averaged estimations and the associated 95% confidence intervals are provided. Left panel: green leaf biomass Q_g evolution. Right panel: root biomass Q_r evolution.

III Multivariate Vs. componentwise proposals

The goal of this test is to compare the Monte Carlo error associated with each one of the proposed updating strategies: to update all the parameters together as a block (Multivariate, denoted $\mathcal{M}$) or to update them separately (Univariate, denoted $\mathcal{N}$) introduces bias in the estimation under both restricted dataset and full dataset scenarios.

Settings :

The tests are carried out with the same dataset, in both its complete form (160 dates) and its restricted form (14 dates). 200 repetitions of the same test are performed in each case to evaluate the Monte Carlo error. Two priors are considered.

$$\Theta^* \Rightarrow Y \begin{cases} \xrightarrow{p_1(\cdot)+\mathcal{N}} \{\tilde{\Theta}_i^{\mathcal{N}^1}\}_{i=1,\ldots,200} \\ \xrightarrow{p_1(\cdot)+\mathcal{M}} \{\tilde{\Theta}_j^{\mathcal{M}^1}\}_{j=1,\ldots,200} \\ \xrightarrow{p_2(\cdot)+\mathcal{N}} \{\tilde{\Theta}_k^{\mathcal{N}^2}\}_{k=1,\ldots,200} \\ \xrightarrow{p_2(\cdot)+\mathcal{M}} \{\tilde{\Theta}_l^{\mathcal{M}^2}\}_{l=1,\ldots,200} \end{cases}$$

Observations and remarks :

Full dataset	Prior 1		Componentwise proposals				Multivariate proposals			
Real value	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a 3.560	3.40	0.20	**3.606**	6.45e-5	0.0114	1.40e-6	**3.602**	2.69e-5	0.0199	7.64e-7
γ_0 0.625	0.90	0.10	**0.630**	8.07e-6	0.00076	1.18e-8	**0.626**	1.02e-6	0.00094	1.70e-9
μ_γ 550.00	650.00	40.00	**531.51**	22.234	2551.90	7.92e4	**542.37**	3.313	2708.11	5.67e4
	Prior 2									
	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a 3.560	3.80	0.40	**3.619**	6.42e-5	0.0118	1.60e-6	**3.611**	1.78e-5	0.0207	1.35e-6
γ_0 0.625	0.40	0.20	**0.633**	9.18e-6	0.00084	1.14e-8	**0.628**	5.80e-7	0.00096	1.84e-9
μ_γ 550.00	350.00	60.00	**521.72**	10.58	2674.84	6.51e4	**534.04**	2.58	2870.53	1.19e4

Restricted dataset	Prior 1		Componentwise proposals				Multivariate proposals			
Real value	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a 3.560	3.40	0.20	**3.449**	1.39e-4	0.0150	4.45e-6	**3.456**	8.23e-6	0.0199	2.47e-7
γ_0 0.625	0.90	0.10	**0.659**	1.30e-5	0.00117	2.60e-8	**0.657**	3.89e-7	0.00153	1.42e-9
μ_γ 550.00	650.0	40.00	**555.43**	12.68	1909.57	2.22e4	**558.83**	0.99	2486.97	3.74e3
	Prior 2									
	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a 3.560	3.80	0.40	**3.474**	7.10e-5	0.0190	2.42e-6	**3.480**	8.23e-6	0.0257	5.29e-7
γ_0 0.625	0.40	0.20	**0.698**	1.44e-5	0.00208	6.35e-8	**0.691**	7.78e-7	0.00257	8.85e-9
μ_γ 550.00	350.0	60.00	**468.73**	11.50	2454.50	4.19e4	**473.58**	1.13	3016.41	6.60e3

Table 8.2: Estimation and uncertainty assessment provided by one chain MCMC based on one simulated dataset with two settings (14 dates and 160 dates). 200 repetitions are performed, with fixed noise parameters. The given estimations are based on a total of 200 tests, with a maximum number of 240 000 iterations for each, taking into account a burn-in period of 40 000 iterations for the restricted dataset and 10 000 for the full dataset. Two prior distributions are used.

- In Table 8.2, notice that the componentwise proposal works almost as well as the multivariate proposal in the case of three parameters for both the full dataset and the scarce dataset. However, the use of the multivariate proposal results in generally flatter distributions (estimations of variance $\mathbb{E}(\mathbb{V}(\Theta|Y))$ tend to be larger compared to the componentwise approach) which suggests the existence of an effect when the covariance among the three parameters are taken into account.

- Both proposals provided similar averaged mean estimates $\mathbb{E}(\mathbb{E}(\Theta|Y))$, except perhaps for the parameter μ_γ which has slightly different estimates from time to time given by the two approaches. Note that μ_γ is the least influential parameter among the three that we estimate.

- Larger prior variances generally lead to larger variance estimations for the target distribution, since the number of data is limited.

- Regarding the Monte Carlo error, componentwise proposals result in larger variability between the individual tests, for both the mean estimates and the

variance estimates ($\mathbb{V}(\mathbb{E}(\Theta|Y))$ and $\mathbb{V}(\mathbb{V}(\Theta|Y))$). This is observed for the different priors that we tested.

– Since the multivariate proposal is much less time consuming (the computational time was reduced by $\approx 62\%$), and it allows us to take into account the covariance among parameters, it is preferred for the following tests.

IV Number of iterations and estimation precision

The objective of this test is to identify the influence of the number of iterations on the performance of the algorithm, especially on the precision of the estimates.

Settings :

Batch means and the variance of expected values of 100 independent runs are used to estimate the Monte Carlo Error (MCE) based on the same restricted dataset. Six configurations for the number of iterations are investigated. The evolution of the estimated MCE is studied with a long chain of 5 000 000 iterations.

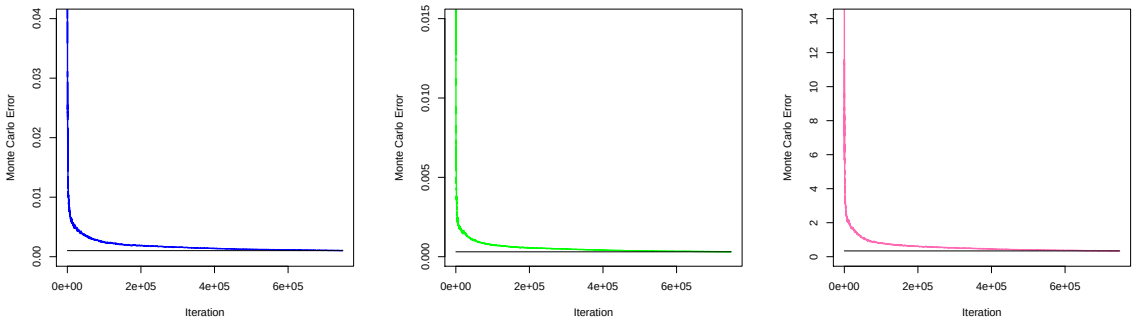
$$\Theta^* \Rightarrow Y \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\dots,100}$$

Observations and remarks :

– Figure 8.2 illustrates the decrease of the estimated Monte Carlo errors when the number of iterations increases for a long run, which is in agreement with the CLT.

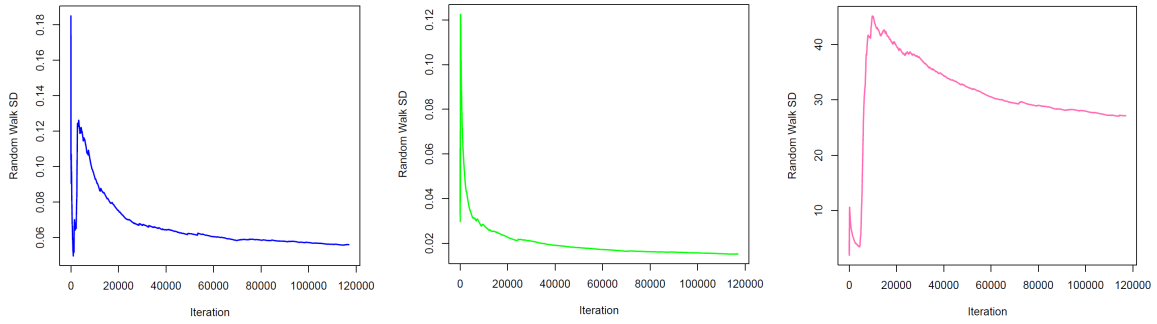
– Figure 8.3 manifests the evolution of the random walk standard deviation tuned by the adaptive scheme. The covariance matrix of the normal instrumental distribution tends to stabilize with the iterations.

– The posterior distributions of the three parameters are given by Figure 8.4.



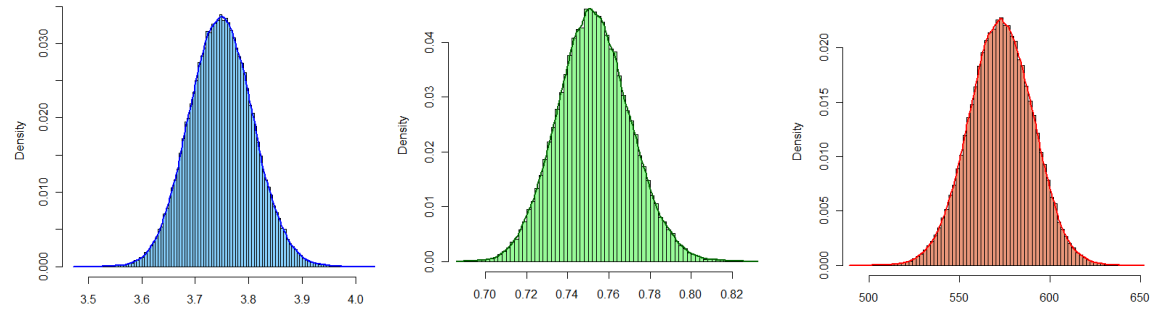
(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.2: Evolutions of the Monte Carlo errors computed by batch means based on a simulated dataset (14 dates) with a long run of 5 000 000 iterations.



(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.3: Evolutions of the standard deviations of random walk steps for one test based on a restricted dataset (14 dates).



(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.4: Posterior distribution estimated by a long run of 10 000 000 iterations based on the same simulated dataset (14 dates) and the same prior distribution.

	Real value		350 000 (iter.)			200 000			100 000		
			$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$
μ_a	3.746		3.7485	3.54e-3	2.48e-7	3.7487	3.47e-3	5.65e-7	3.7486	3.45e-3	9.90e-7
γ_0	0.7525		0.7527	3.1e-4	2.44e-8	0.7527	3.1e-4	4.72e-8	0.7527	3.0e-4	8.19e-8
μ_γ	579.00		573.241	307.71	2.65e-2	573.25	305.58	4.24e-2	573.21	305.31	7.90e-2
	Prior		50 000 (iter.)			30 000			10 000		
	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$
μ_a	3.6	0.01	3.7485	3.45e-3	1.91e-6	3.7485	3.44e-3	3.41e-6	3.7483	3.39e-3	9.79e-6
γ_0	0.75	0.0064	0.7527	3.0e-4	1.68e-7	0.7538	3.0e-4	3.18e-7	0.7527	3.0e-4	9.13e-7
μ_γ	600.0	400.00	573.23	307.89	1.67e-1	573.27	305.03	2.98e-1	573.33	302.49	9.46e-1

Table 8.3: Evolution of estimations and uncertainty assessments provided by one chain AMwG, with the increase of the number of iterations, based on the same simulated dataset (14 dates). For each configuration, the mean estimates are based on 100 independent runs.

– According to Table 8.3, the averaged estimates ($\mathbb{E}(\mathbb{E}(\Theta|Y))$) are always accurate when comparing the six configurations. However, the estimated

$\mathbb{V}(\mathbb{E}(\Theta|Y))$ for the three parameters, which indicate the precision of the mean estimates and can also be regarded as a way to evaluate the MCE, increase quickly when the number of iterations decreases. Therefore, a lower bound of the number of iterations should be defined if a certain precision level is desired.

– The variance estimates $\mathbb{E}(\mathbb{V}(\Theta|Y))$ also begin to be less precise with the configuration of 10 000 iterations. This test can be regarded as a preliminary test when implementing a MCMC-based method, as it gives a vague idea of the behaviour of the chains, especially to decide the threshold for the stopping criteria.

V Proposal scaling effect on the convergence

As mentioned in Section 4.1.1, it is possible to apply the global multivariate adaptive scaling to improve the convergence of the algorithm. Here, we attempt to verify the scaling effect compared to the simple Adaptive Metropolis.

Settings:

This test is conducted with the same simulated restricted dataset by the two algorithms, with global scaling and without global scaling.

Observations and remarks :

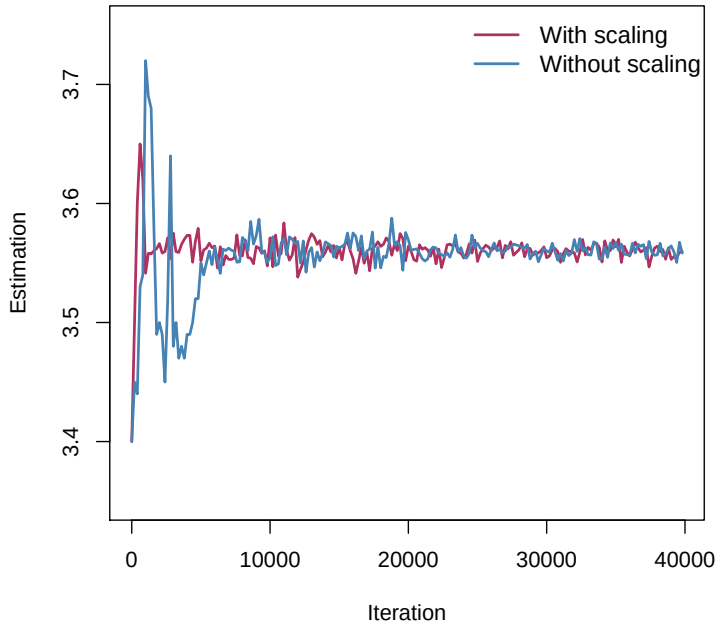


Figure 8.5: Estimation with and without global scaling of the adaptive scale based on the same simulated restricted dataset.

Figure 8.5 demonstrates that the scaled proposal has an impact at the beginning of the chain, but for a long run, there is no significant difference between the chains run with and without scaling.

VI Impact of priors and initial values

As we know, the prior distribution plays an important role in Bayesian inference. However, the prior effect fades out with an increasing number of observations. In this test, we try to identify the importance of the prior in the scenario with restricted dataset.

Settings :

Three priors are used for the estimation based on a restricted dataset. No global scaling is used. The estimates are based on 100 repetitive long runs with a fixed number of iterations (350 000).

$$\Theta^* \Rightarrow Y \quad \begin{array}{l} \xrightarrow{p_1(\cdot)} \{\tilde{\Theta}_i^1\}_{i=1,\ldots,100} \\ \xrightarrow{p_2(\cdot)} \{\tilde{\Theta}_j^2\}_{j=1,\ldots,100} \\ \xrightarrow{p_3(\cdot)} \{\tilde{\Theta}_k^3\}_{k=1,\ldots,100} \end{array}$$

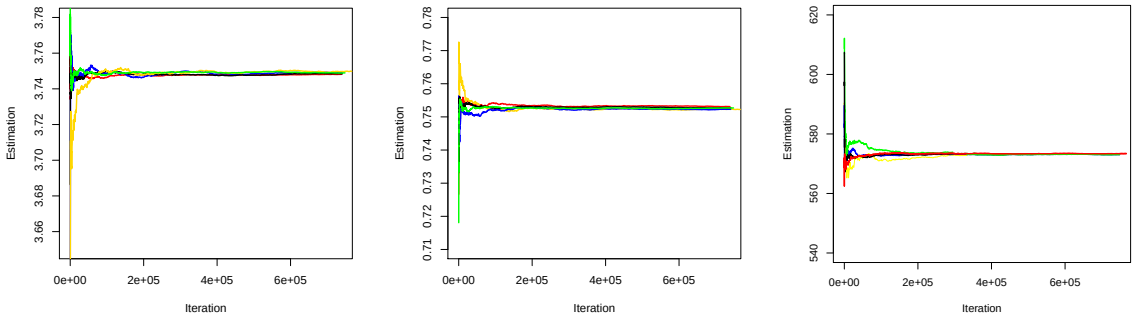
Observations and remarks :

– Table 8.4 illustrates that the estimates resulting from different priors are quite different, which indicates the lack of data to erase the prior information. This result suggests that in the scenario with restrict dataset, the prior should be chosen cautiously, for it has important influence on the posterior distribution.

	Real value	**Prior 1**		Estimation		Monte Carlo error	
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	σ_{BM}
μ_a	3.746	3.60	0.01	3.748	0.0035	0.000498	0.000888
γ_0	0.7525	0.750	0.0064	0.7527	0.00031	0.000156	0.000265
μ_γ	579.00	600	400	573.24	307.71	0.1627	0.2648
		Prior 2					
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	σ_{BM}
μ_a	3.746	3.70	0.01	3.818	0.0036	0.000572	0.000954
γ_0	0.7525	0.720	0.0064	0.7231	0.00026	0.000155	0.000260
μ_γ	579.00	650	400	614.91	315.91	0.1817	0.2890
		Prior 3					
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	σ_{BM}
μ_a	3.746	3.752	0.0225	3.804	0.0050	0.000646	0.001013
γ_0	0.7525	0.744	0.0100	0.7453	0.00042	0.000191	0.000296
μ_γ	579.00	590.695	625	562.94	430.93	0.1925	0.3023

Table 8.4: Estimation and uncertainty assessment provided by one chain MCMC based on one simulated dataset (14 dates) with 100 repetitions. For each test, the length of the chain is 350 000. Three prior distributions are used.

– It is likely that the batch means method over-estimates the Monte Carlo errors, since they are more important than the estimations obtained with the 100 repetitions of the same test.



(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.6: Evolutions of the averaged estimations computed by batch means based on the same simulated dataset (14 dates) and the same prior distribution, performed for 5 independent chains departing from different initial points.

– The estimated Monte Carlo errors are relatively small, which imply small algorithmic errors when long chains are employed. Since the initial point of each chain is drawn from the prior distribution, this result also indicates that the posterior distribution obtained is independent of the departure point (Figure 8.6). A long run could ensure a good performance of convergence to the invariant distribution independently of the departure point as long as they are drawn from the same prior distribution.

VII Normal proposal Vs. Student proposal

This test aims to investigate the impact of normal and Student proposals, since they are both symmetric proposals but with different types of distribution tails.

Settings :

Based on the same simulated full dataset, the AMwG algorithm is carried out with both normal proposals for random walk and Student proposals for random walk, each with 200 repetitions.

$$\Theta^* \Rightarrow Y \begin{array}{l} \xrightarrow{\mathcal{N}(\cdot)} \{\tilde{\Theta}_i^{\mathcal{N}}\}_{i=1,\ldots,200} \\ \xrightarrow{\mathcal{T}(\cdot)} \{\tilde{\Theta}_j^{\mathcal{T}}\}_{j=1,\ldots,200} \end{array}$$

Observations and remarks :

– As suggested by Table 8.5, both methods result in similar estimations. Nonetheless, Student distribution has heavier tails than the normal one, which implies more tolerance. The posterior distribution given by the Student proposal results in slightly larger variances, however the estimation of the target distribution should not be influenced by the proposal distribution. Thus, it is possible that some of the tests performed with the Student proposal have not converged, some outlier samples may be consequently included.

	Real value	Prior		Student proposals			
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a	3.560	3.80	0.40	**3.613**	1.01e-5	0.0216	1.64e-6
γ_0	0.625	0.40	0.20	**0.629**	4.61e-7	0.00098	7.24e-9
μ_γ	550.00	350.00	60.00	**534.39**	2.41	3004.42	2.47e4
	Real value	Prior		Normal proposals			
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a	3.560	3.80	0.40	**3.611**	1.78e-5	0.0207	1.35e-6
γ_0	0.625	0.40	0.20	**0.628**	5.80e-7	0.00096	1.84e-9
μ_γ	550.00	350.00	60.00	**534.04**	2.58	2870.53	1.19e4

Table 8.5: Estimation and uncertainty assessment provided by normal and Student proposals based on one simulated dataset (160 dates) with 200 repetitions, with fixed noise parameters. The given estimations are based on a total of 200 tests, with a maximum of 120 000 iterations for each, taking into account a burn-in period of 10 000 iterations.

– Note that the Student distribution is widely used to correct the estimation of the variance for MCMC based methods. Although the risk of over-estimation of the variance is not desirable with the objective of reducing uncertainties related to the parameter estimates for future prediction use, the effect of correcting the variance estimator could be attractive for multiple chains algorithms, such as the Interacting parallel MCMC, when the shrinkage (sample degeneracy) is present. It is also used in the Gelman-Rubin criterion (Gelman, 1992).

8.2.3 Differential Evolution Adaptive Metropolis

The Differential Evolution Adaptive Metropolis (DREAM) approach uses the strategy of "some median runs". Therefore, the assumption has been made for this approach that the target distribution of the M parallel chains is the product of the original target distributions $\pi^M(\cdot)$ (Vrugt et al., 2009a). Thanks to the MCMC's scheme, a population of estimates resulting from M chains move from one iteration to another conditionally on the estimates of the previous iteration. As a result, the empirical moments can be derived. In the case that parallel chains are engaged without communication, the overall speed of convergence is limited by the convergence of each chain and the number of iterations to be discarded for each of them. The DREAM algorithm proposes to hasten the convergence of individual chains by creating communications between the chains. In the following, the performance and the features of this algorithm are studied.

List of the tests for the DREAM algorithm :

I	Comparison of convergence property with one chain MCMC
II	Impact of the number of parallel chains
III	Comparison of the computational costs of different configurations
IV	Discussion: Comparison of computational strategies

General settings :

As for the previous tests with one chain AMwG, the modelling noise parameters and the observation noise parameters are fixed at 0.02 and 0.1 respectively. Multivariate proposals are used without global scaling.

I Comparison of convergence property with one chain MCMC

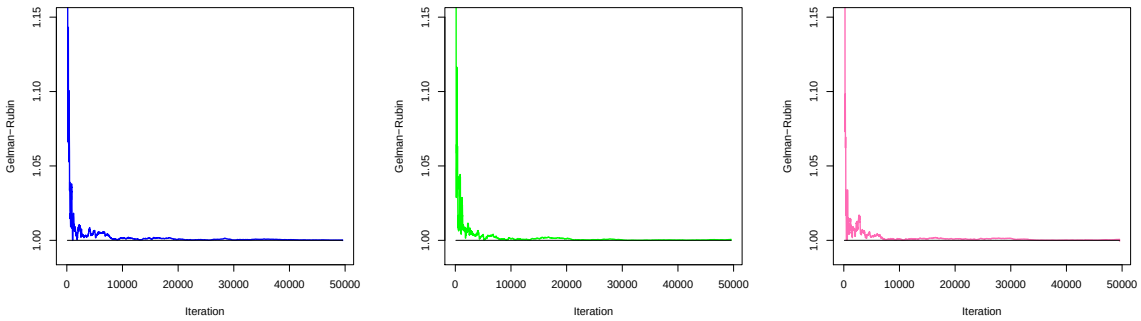
The objective is to investigate the convergence behaviour of this parallel chain algorithm compared to the one chain AMwG algorithm.

Settings :

Given a restricted dataset, we monitor the evolution of the Gelman-Rubin scale parameters as well as the corresponding evolution of the estimates. Then, 100 runs are conducted based on the same dataset with the same total number of iterations to compare the estimation quality of DREAM and one chain AMwG.

$$\Theta^* \Rightarrow Y \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\ldots,100}$$

Observations and remarks :

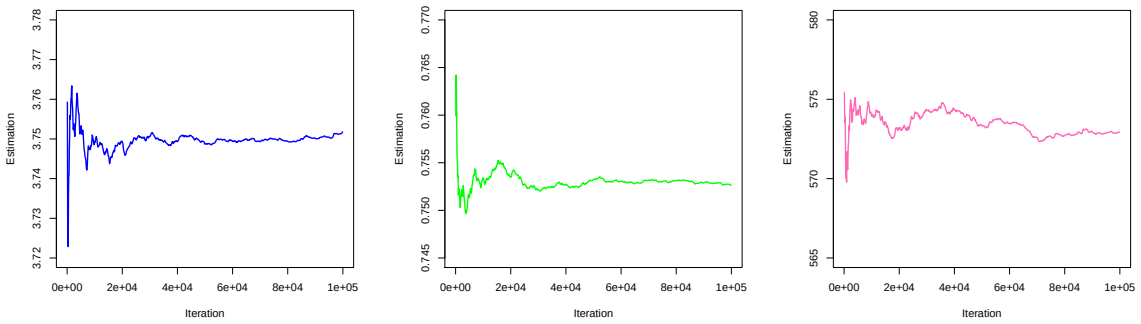


(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.7: Evolutions of the Gelman-Rubin criterion of the DREAM algorithm based on a restricted dataset (14 dates).

	Real value	Prior		AMwG		DREAM		Theoretical value*	
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\tilde{\mu}^*$	$\tilde{\sigma^2}$
μ_a	3.746	3.6	0.01	3.748	0.0035	3.749	0.0024	3.7487	0.0035
γ_0	0.7525	0.75	0.0064	0.7527	0.00031	0.7531	0.00021	0.7526	0.00031
μ_γ	579.00	600.0	400.00	573.24	307.71	572.79	211.07	573.21	307.89

Table 8.6: Estimation and uncertainty assessment provided by one chain MCMC and DREAM based on the same simulated dataset (14 dates) with 100 repetitions. Each test contains 360 000 simulations for both DREAM and one chain MCMC. *: The theoretical values are provided by taking the average of three long runs of AMwG with 10 000 000 iterations.



(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.8: Evolutions of the estimates provided by the DREAM algorithm based on a restricted dataset (14 dates).

Based on the evolution of the Gelman-Rubin diagnostic illustrated by Figure 8.7, we notice that the scaling parameters converge very quickly, while it is not the case for the evolution of the estimates as demonstrated by Figure 8.8. This result suggests that the Gelman and Rubin criterion is probably too tolerant for small numbers of chains.

Table 8.6 implies that the mean posterior variances are very likely to be underestimated when the recommended configuration is used (Ter Braak and Vrugt, 2008), in which the number of chains is $2d$, with d the dimension of the target distribution. Ter Braak (2006) showed that $M = 2d$ or $3d$ worked well for unimodal posteriors, but $M = 10d$ to $20d$ were needed when multimodality or high correlation was present in the target distribution. Therefore, more chains might improve the inference of the posterior distribution.

II Impact of the number of parallel chains

Following this idea, tests with more parallel chains are conducted to see their impacts on the variance estimation. The performance of the Gelman-Rubin criterion is also evaluated.

Settings :

With the same restricted dataset, 12 configurations with different numbers of parallel chains are tested with each 50 independent runs.

$$\Theta^* \Rightarrow Y \quad \begin{array}{l} \xrightarrow{1d} \{\tilde{\Theta}^{1d}\}_{i=1,\ldots,50} \\ \ldots \\ \xrightarrow{12d} \{\tilde{\Theta}^{12d}\}_{i=1,\ldots,50} \end{array}$$

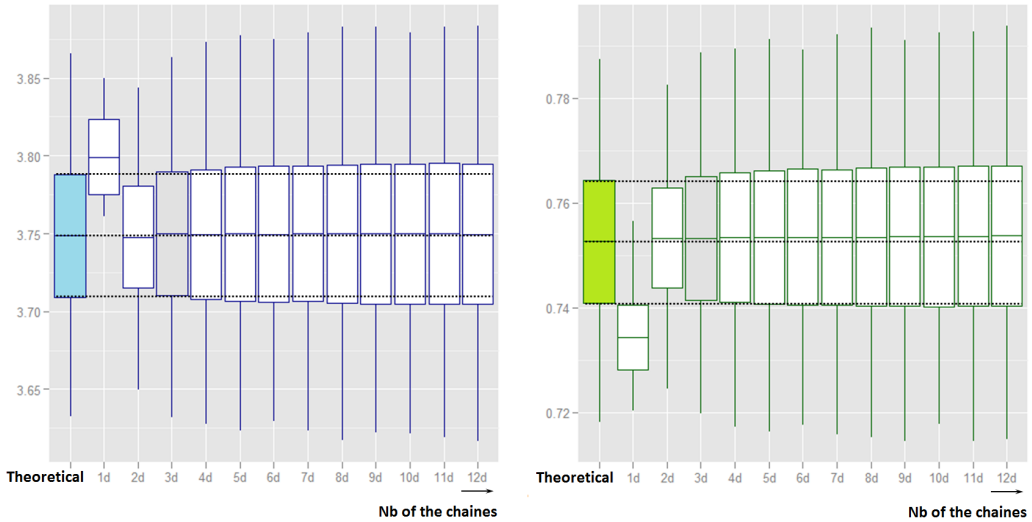
Observations and remarks :

In this test, configurations with more pooled parallel chains are adopted to estimate the target distribution. According to Figure 8.9, the mean estimates appear to be quite accurate when $M \geq 2d$. However, the variance estimations are not stable for different

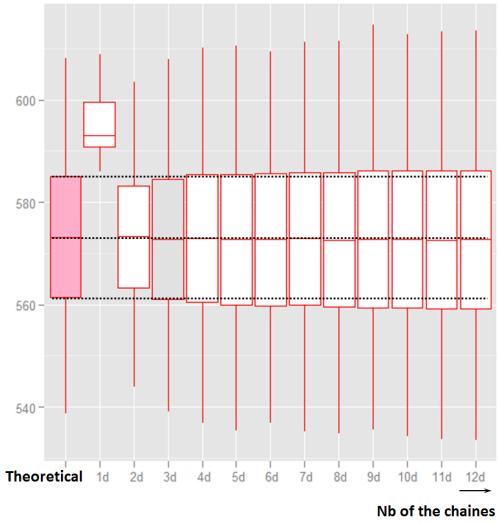
configurations. The variance estimations increase when more chains are involved, as demonstrated by Table 8.8. This is probably due to the fact that the burn-in period is difficult to handle for all the chains in an independent way (Geyer, 1991). Therefore, when the number of chains increases, more and more outliers could be included in the final pooled estimates which are used to construct the posterior distribution. In other words, some individual chains may not have converged.

	μ_a		γ_0		μ_γ	
	$\tilde{\mu}^*$	$\tilde{\sigma^2}^*$	$\tilde{\mu}^*$	$\tilde{\sigma^2}^*$	$\tilde{\mu}^*$	$\tilde{\sigma^2}^*$
Theoretical*	3.7487	3.47e-3	0.7526	3.06e-4	573.21	307.89
Nbr of chains	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$
3 (1d)	3.757	8.02e-4	0.7401	7.77e-5	605.98	130.14
6 (2d)	3.748	2.47e-3	0.7543	2.24e-4	572.02	230.75
9 (3d)	3.750	3.50e-3	0.7533	3.07e-4	572.62	314.95
12 (4d)	3.750	3.84e-3	0.7536	3.38e-4	572.59	347.49
15 (5d)	3.750	4.10e-3	0.7534	3.57e-4	572.78	364.60
18 (6d)	3.749	4.23e-3	0.7536	3.64e-4	572.87	377.89
21 (7d)	3.750	4.42e-3	0.7535	3.83e-4	572.64	392.98
24 (8d)	3.750	4.47e-3	0.7534	3.88e-4	572.74	396.70
27 (9d)	3.750	4.54e-3	0.7536	3.95e-4	572.71	396.59
30 (10d)	3.750	4.54e-3	0.7536	3.97e-4	572.43	398.97
33 (11d)	3.749	4.53e-3	0.7536	3.96e-4	572.80	397.60
36 (12d)	3.750	4.55e-3	0.7535	3.98e-4	572.75	399.41

Table 8.7: Comparison of the estimations and uncertainty assessments resulting from different configurations of the DREAM algorithm based on the same simulated dataset (14 dates) with 50 independent runs. *: The theoretical values are provided by taking the average of three long runs of MCMC with 5 000 000 iterations.



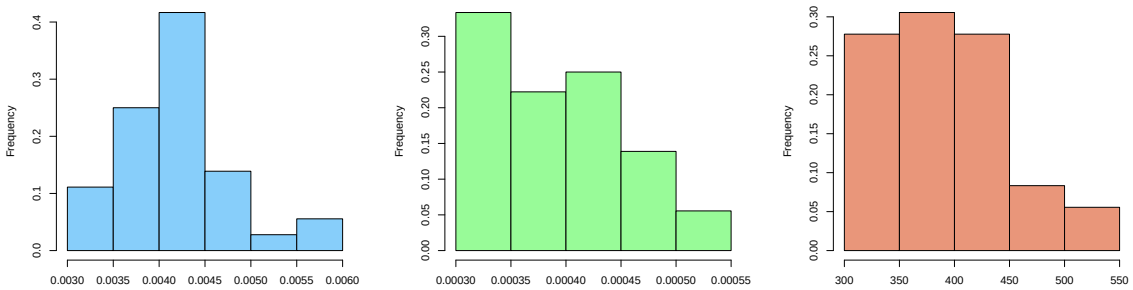
(a) μ_a (b) γ_0



(c) μ_γ

Figure 8.9: Evolution of the estimations with different configurations of the DREAM algorithm based on the same simulated dataset (14 dates) with 50 repetitions.

It is confirmed by checking the intra-chain variances of each chain (Figure 8.10) which are quite variable. It is possible that the jumps introduced between chains to improve the convergence speed, could be too frequent and result in bias. The components of the chains exhibit important dependencies which may damage the adaptive scheme and dissimulate the fact of non-convergence. If a few chains have converged, the overall behaviour of the within chain variance and between chain variance could easily appear to be stable and thus claim the convergence too fast, especially when a large number of chains are involved. Many parallel chains could artificially exhibit a robust behaviour, for instance the mean estimates may appear to be stable while the target distribution is not properly characterized.

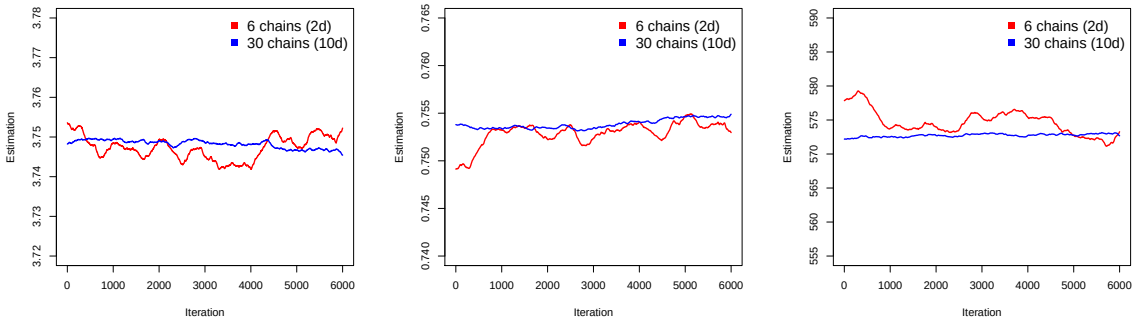


(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.10: Histograms of the intra-chain variance provided by the DREAM algorithm configured with 24 parallel chains ($8d$) based on a restricted dataset (14 dates).

Indeed, facing multimodality or high dimensional problems, one strategy that is widely used concerns building a pool of samples resulting from multiple chains performed in parallel. However, the quasi-ergodicity problem may frustrate this simple idea. Our result suggests that when the lengths of the chains are limited, multiple chains may not necessarily provide a better estimation of the target distribution than the single chain, especially for nonlinear models and limited data. Most importantly, it seems more difficult to define a proper way to diagnose the convergence for multiple parallel chains. As stated previously, our results imply that there is a strong possibility that the Gelman and Rubin criterion is too tolerant for small numbers of chains, for example, in the case when $M = 1d$, both the mean estimates and the variance estimates are inaccurate (Figure 8.9).

The Gelman and Rubin criterion puts emphasis on reducing bias in estimation by using a shrink factor, which can be interpreted as follows : when the between-chain variance is dominated by the pooled within-chain variance, it approaches to one, and thus all the chains should have escaped the influence of their departure points (initial values of the chains) and have visited the entire target distribution. However, an important feature of its implementation is that it strongly relies on the user ability to find an appropriate starting distribution which is over-dispersed. If a small number of chains are involved, not only the dispersion of the departure points might not be important enough, but it is also possible that the communications between chains which are meant to accelerate the convergence speed, may result in an underestimation of the variance between chains. As a result, the convergence is claimed much earlier than it should be. In Ter Braak (2006), similar results were obtained, they showed that the Gelman and Rubin criterion is too optimistic about the convergence. This result suggests a reassessment of this diagnostic criterion.



(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.11: Evolution of the mean estimates given by the configurations of 6 chains and 30 chains of the DREAM algorithm based on the same simulated dataset (14 dates) after the burn-in period of 4000 iterations already discarded.

The mean estimates are more variable with smaller numbers of chains than larger numbers of chains, as illustrated by Figure 8.11 with a burn-in period of 4000 iterations already discarded. Although the final mean estimates given by the two configurations

are quite close, it seems that the configuration of 30 chains provide much more stable estimates. Thus, the computational cost is crucial to determine the appropriate number of chains.

III Comparison of the computational costs of different configurations

The objective is to identify the computational cost when different number of chains are used in the DREAM algorithm.

Settings :

Twelve configurations regarding the number of chains are considered. A minimum number of 4000 iterations are required and regarded as a penalty term to avoid an over-early convergence claim. Once the convergence criterion is satisfied ($\sqrt{\hat{R}} < 1.001$), another 10 000 iterations are conducted to build the posterior distributions.

	Computational cost			
	Number of iterations	Memory (Gb)	CPU time	Real computational time
3 ($1d$)	12000	1.06	26m	4m
6 ($2d$)	69500	8.00	1h44m	17m
9 ($3d$)	45800	10.02	2h17m	24m
12 ($4d$)	62300	14.47	5h35m	59m
15 ($5d$)	78500	20.60	5h59m	1h6m
18 ($6d$)	51200	20.01	5h35m	1h5m
21 ($7d$)	48900	19.23	6h34m	1h15m
24 ($8d$)	53000	22.01	7h42m	1h23m
27 ($9d$)	51900	25.15	7h38m	1h25m
30 ($10d$)	46000	25.93	9h16m	1h32m
33 ($11d$)	41100	25.59	8h30m	1h28m
36 ($12d$)	40300	25.41	8h35m	1h29m

Table 8.8: Comparison of the computational cost for the different configurations of the DREAM algorithm based on the same simulated dataset (14 dates).

Observations and remarks :

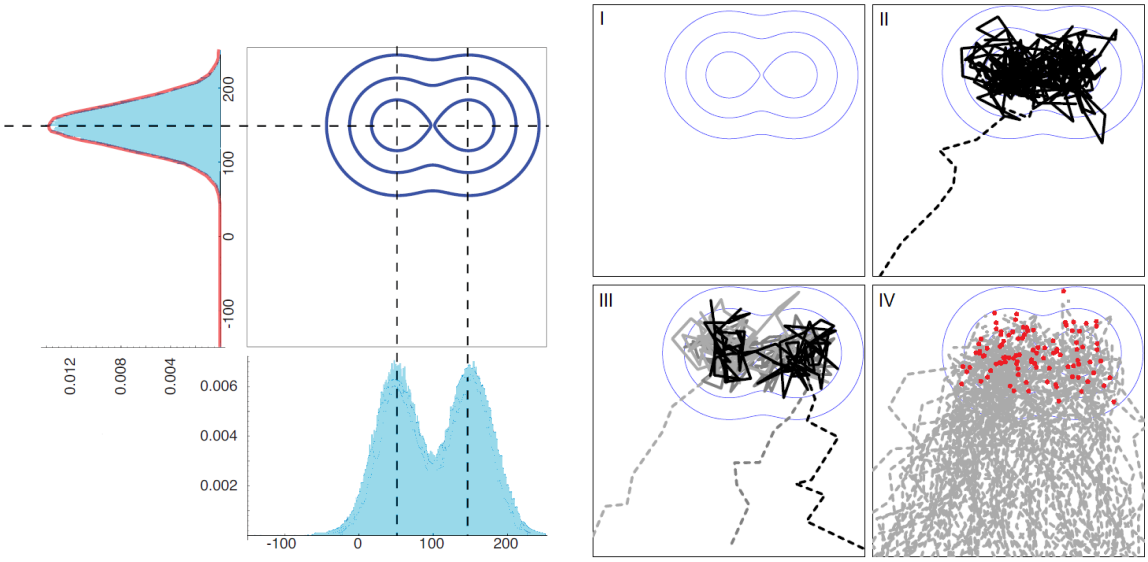
Table 8.8 compares the computational time and memory requirement of the 12 configurations. Despite the adaptive scheme, when the target distribution increases in dimension, it may result in an undesirable slow convergence compared to a single chain. For all the M chains to converge, there is an extra computational cost which is very important as illustrated by Table 8.8. Note that it could be worse if slow mixing occurs.

IV Discussion: Comparison of computational strategies

Indeed, it could be considered as inefficient to run multiple chains while a substantial number of the first estimates are discarded, in the sense that if one runs a single chain of 20 000 iterations in comparison of 20 independent chains with each 1 000 iterations, then the last 19 000 iterations of the long chain are more likely to be drawn from

the real target distribution than those reached by any of the other chains. Figures of 8.12 give an efficient example to compare the performance of the two strategies: "one long run" and "many short runs" in the case of a bimodal target distribution when two parameters are involved. When the number of the chains is limited, as well as their lengths, effectively, there is no guarantee that the weight of each mode is assigned proportionally to its relative likelihood evaluation, therefore, the variance estimations are less accurate. In Geyer (1991), a debate is carried out regarding these two strategies, their advantages and disadvantages are accordingly discussed.

In order to determine the number of samples of the chains that should be discarded pertinently, an efficient way is to take only the last sample of each chain, as showed by 8.12(b)(IV). This leads to the idea of using a large number of even "shorter" runs to estimate the target distribution. This is exactly the technique employed by the Interacting parallel MCMC algorithm.



(a) True density. (b) Estimation trajectories.

Figure 8.12: One long run compared to many short runs for a bimodal target distribution. The evolution of the estimation given by one long run (II), three short runs (III) and 100 short runs (IV) compared to the target distribution (I). The red points in (IV) correspond to the estimations resulting from the last iteration of each chain.

8.2.4 Interacting parallel MCMC

The Interacting parallel MCMC algorithm adopts the strategy of "many short runs". Introduced by Campillo et al. (2009), the idea of Interacting MCMC is to concentrate the computational effort on the zone of interest. To achieve this purpose, each individual chain proposes one candidate for all the chains. Only the last estimate of each chain is taken into account to build the posterior distribution.

List of the tests for the Interacting MCMC algorithm :

I	Convergence monitoring
II	Impact of the prior distribution
III	Impact of the number of chains
IV	Improvement tests

General settings :

As for the previous tests, the modelling noise parameters and the observation noise parameters are fixed at 0.02 and 0.1 respectively. Multivariate proposals are used without global scaling. No adaptive scale is used to tune the proposals. The first convergence criterion for one chain AMwG described in Section 4.1.5 is used for the Interacting parallel MCMC algorithm.

I Convergence monitoring

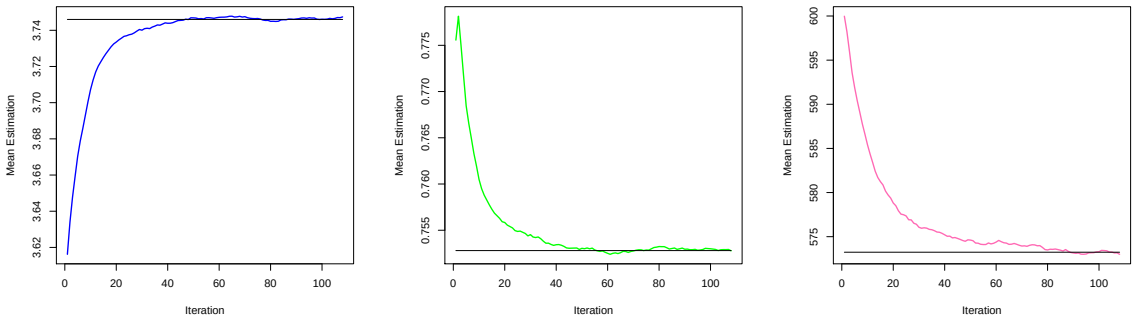
The objective is to monitor the evolution of the estimates given by the Interacting MCMC in order to observe their convergence behaviour.

Settings :

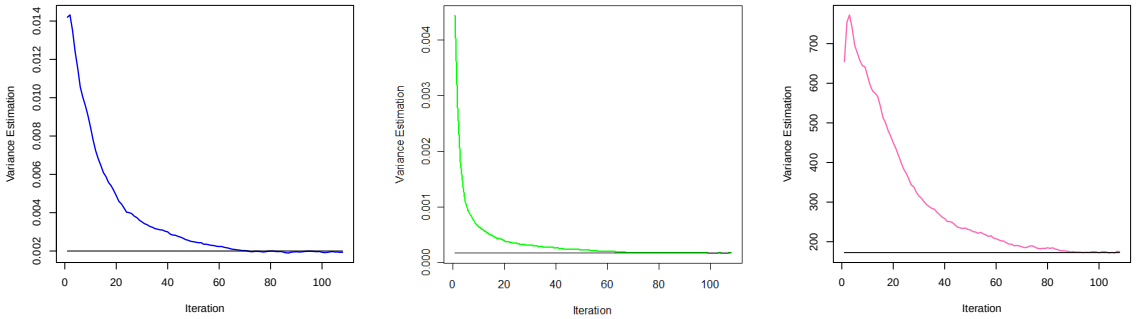
For a given restricted dataset, 1000 parallel chains are used, the overall estimates are built from only the last samples of all the chains.

Observations and remarks :

The monitoring of both the mean and the variance estimates of the posterior distributions (Figure 8.13) shows that they converge after about 120 iterations. Further investigations are presented in the following tests.



(a) $\mathbb{E}(\tilde{\mu}_a|Y)$ (b) $\mathbb{E}(\tilde{\gamma}_0|Y)$ (c) $\mathbb{E}(\tilde{\mu}_\gamma|Y)$



(d) $\mathbb{V}(\tilde{\mu}_a|Y)$ (e) $\mathbb{V}(\tilde{\gamma}_0|Y)$ (f) $\mathbb{V}(\tilde{\mu}_\gamma|Y)$

Figure 8.13: Evolutions of the mean estimates and the variance estimates based on the same simulated dataset (14 dates) and the same prior distribution, performed by Interacting MCMC with 1000 chains. The horizontal lines represent the converged values.

II Impact of the prior distribution

The objective is to observe the behaviour of the Interacting MCMC when performed with different priors compared to the one chain AMwG.

Settings :

100 independent runs are conducted with the same restricted dataset and two different priors.

$$\Theta^* \Rightarrow Y \begin{array}{l} \xrightarrow{p_1(\cdot)} \{\tilde{\Theta}_i^1\}_{i=1,\ldots,100} \\ \xrightarrow{p_2(\cdot)} \{\tilde{\Theta}_j^2\}_{j=1,\ldots,100} \end{array} \tag{8.13}$$

	Theoretical value*		**Prior 1**		Estimation		Monte Carlo error	
	$\tilde{\mu}$	$\tilde{\sigma}^2$	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}$(MSE)
μ_a	3.749	0.0035	3.60	0.01	3.748	0.0019	1.23e-5	1.24e-5
γ_0	0.7526	0.00031	0.750	0.0064	0.7524	0.00016	1.26e-6	1.33e-6
μ_γ	573.21	307.89	600	400	573.26	165.27	1.26	1.25
	Theoretical value*		**Prior 2**					
	$\tilde{\mu}$	$\tilde{\sigma}^2$	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}$(MSE)
μ_a	3.818	0.0036	3.70	0.01	3.817	0.0020	1.26e-5	1.27e-5
γ_0	0.7230	0.00026	0.720	0.0064	0.7229	0.00014	9.06e-7	9.17e-7
μ_γ	614.97	315.71	650	400	615.02	170.32	1.05	1.06

Table 8.9: Estimation and uncertainty assessment provided by Interacting MCMC based on one simulated dataset (14 dates) with 100 independent runs. Each test contains a maximum of 350 000 simulations (1000 chains), and two prior distributions are used. The theoretical values are obtained by three long runs of AMwG with 5000000 iterations.

Observations and remarks :

Based on the results given in Table 8.9, despite the mean estimates which appear to be in agreement with the results of one long run, it seems that the Interacting MCMC underestimates the variance of the posterior compared to the result of one long run. It is possibly due to the limitation regarding the number of samples. In the following test, configurations with different numbers of chains are compared.

III Impact of the number of chains

We seek to optimize the performance of this algorithm by adjusting the number of chains. Given the fact that the posterior distribution is solely built based on the last samples, the number of chains is crucial to assure a proper interpretation of the results.

Settings :

Based on the same restricted dataset, we maintain the same total numbers of performed simulations (350 000) and compare the estimation results given by 5000 chains of 70 iterations, 3500 chains of 100 iterations and 1000 chains of 350 iterations. 100 independent runs are processed to compare the estimation of Monte Carlo errors.

$$\Theta^* \Rightarrow Y \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\dots,100}$$

Observations and remarks :

Table 8.10 illustrates a comparison test between three different configurations of the Interacting MCMC algorithm with the same number of simulations in total. The same configurations with independent chains are carried out as well. Obviously, the Interacting MCMC's scheme allows to accelerate significantly the estimations, since none of the multiple independent chains appear to have converged.

An interesting remark is about the Mean Squared Error (MSE) estimations. Since

$$MSE(\Theta) = bias^2(\Theta) + Var(\Theta),$$

the estimated MSEs are obtained based on the assumption that the theoretical values are given by one very long run of AMwG. In all the three cases, they are more important than $\mathbb{V}(\mathbb{E}(\Theta|Y))$. Among the three configurations tested with the same total number of simulations, $\mathbb{V}(\mathbb{E}(\Theta|Y))$, which can be regarded as an estimation of Monte Carlo error, increases when the number of chains decreases, and so is MSE.

However, since one iteration of Interacting MCMC generally takes more time than one iteration of parallel independent MCMC, for the independent chains are updated only with their own proposed candidate, the computational cost for one iteration increases exponentially with the number of chains. Hence, the CPU time is taken into account and the DIC is used to judge the goodness of fit, as presented by Figure 8.14.

The interacting scheme clearly improves the convergence behaviour. The considered selection technique can be regarded as a variant of importance sampling, with each candidate assigned an importance weight in order to compute their acceptance probability. In this way, the computational efforts are mostly concentrated in the zone of interest.

	Real value	Prior		Theoretical value*			
		μ_0	σ_0^2	$\tilde{\mu}$	$\tilde{\sigma}^2$		
μ_a	3.746	3.60	0.01	**3.7487**	3.47e-3		
γ_0	0.7525	0.750	0.0064	**0.7526**	3.10e-4		
μ_γ	579.00	600	400	**573.21**	307.89		
	5000(chains) * 70(iter.)					Indep.	
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	MSE**	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$
μ_a	3.7479	1.17e-6	2.15e-3	5.88e-9	1.70e-6	3.7108	7.16e-3
γ_0	0.7527	9.49e-9	1.80e-4	5.45e-11	1.04e-8	0.7591	5.94e-4
μ_γ	573.54	1.21e-1	189.63	34.75	2.20e-1	581.45	514.09
	3500(chains) * 100(iter.)					Indep.	
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	MSE**	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$
μ_a	3.7479	2.52e-6	1.99e-3	6.99e-9	3.49e-6	3.7222	6.36e-3
γ_0	0.7526	2.33e-8	1.70e-4	4.30e-11	4.49e-8	0.7563	5.14e-4
μ_γ	573.35	1.91e-1	172.54	23.67	2.05e-1	580.51	461.05
	1000(chains) * 350(iter.)					Indep.	
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	MSE**	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$
μ_a	3.7479	4.82e-6	1.87e-3	1.87e-8	5.04e-6	3.7384	5.21e-3
γ_0	0.7524	4.65e-7	1.62e-4	1.41e-10	5.34e-7	0.7534	4.06e-4
μ_γ	573.46	5.11e-1	166.05	144.79	5.67e-1	578.03	335.43

Table 8.10: Evolution of estimations and uncertainty assessments provided by one chain AMwG, with 350 000 simulations of different configurations, based on the same simulated dataset (14 dates). For each configuration, the mean estimates are based on 100 independent runs. *: The theoretical values are obtained by taking the average of three long runs of MCMC with 5000 000 iterations. **: Mean Square Error computed based on the theoretical mean values. Indep.: the same configuration but with independent chains.

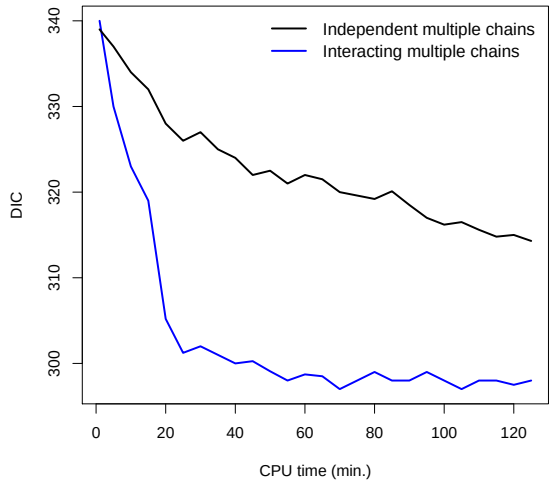


Figure 8.14: Comparison of the evolution of the DIC for the parallel independent MCMC and the interacting MCMC with both 500 chains and the same simulated dataset (14 dates).

Nevertheless, this algorithm does suffer from variability, for the estimated variances are much smaller than the results given by one chain AMwG. Indeed, as a variant of importance sampling, the considered selection process could result in sample impoverishment problem, for the regions with small acceptance probability are rarely visited. This can be seen as a counterpart of concentrating the computational efforts on the zone of interest. Moreover, compared to DREAM, here we pursue a much more important number of parallel chains, considering that each chain provides a candidate for all the chains, the computational cost is much more important, while a sampling bias is introduced. Thus, if only the mean estimation is desired, this method is very efficient, however it cannot provide an appropriate estimation of the entire target distribution.

One possible improvement of this method is to loosen the convergence criterion and to run the Interacting parallel MCMC algorithm until its convergence is claimed, and then continue to run the parallel chains in an independent way in order to characterize correctly the target distribution.

IV Improvement tests

Motivated by the attractive efficiency of the original algorithm and its drawback of underestimation of the variance to characterize the target distribution, we aim to improve this algorithm by adding a step of independent parallel runs. The objective of this test is to study the performance of this new algorithm.

Settings :

Based on the same configuration as test II conducted to examine the impact of the prior distribution, we perform the improved Interacting parallel MCMC with 1000 chains. The total number of 350 000 iterations is maintained. For the first 100 iterations, the interactions are introduced while for the last 250 iterations, the chains remain independent without interactions. The Adaptive Metropolis with scaling is used to tune the proposals of the 1000 chains for the last 250 iterations. 100 independent runs are conducted.

$$\Theta^* \Rightarrow Y \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\ldots,100}$$

Observations and remarks :

	Theoretical value*		**Prior 1**		Interacting		Improv. Int.	
	$\tilde{\mu}$	$\tilde{\sigma}^2$	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$
μ_a	3.749	0.0035	3.60	0.01	3.748	0.0019	3.740	0.0032
γ_0	0.7526	0.00031	0.750	0.0064	0.7524	0.00016	0.7521	0.00029
μ_γ	573.21	307.89	600	400	573.26	165.27	575.62	330.74

Table 8.11: Estimation and uncertainty assessment provided by Interacting MCMC and Improved Interacting algorithms based on one simulated dataset (14 dates) with 100 independent runs. The theoretical values are obtained by three long runs of AMwG with 5000 000 iterations. Improv. Int.: Improved Interacting parallel MCMC.

Table 8.11 suggests that the new algorithm improves the variance estimation in a considerable way compared to the original Interacting MCMC. The counterpart is that

the mean estimates become less accurate, especially for μ_γ. The fact of introducing the adaptive scheme with global scaling helps to accelerate the process of finding the appropriate random walk step, which is very useful to this implementation context with important number of parallel chains. However, further studies are required to optimize the performance of this new algorithm, such as to reduce the number of parallel chains in the first phase of the estimation when interactions are introduced, and to increase the number of parallel chains in the second phase of the estimation.

8.3 Implementation of SMC methods

In this section, we present the implementation of three SMC filtering methods: Regularized/Convolution Particle Filter, Ensemble Kalman Filter and Unscented Kalman Filter to the LNAS model based on the simulated datasets.

In the first place, we detail the implementation process of these methods with an emphasis on the initialization. Then, some tests based on the simulated datasets are conducted according to the features of the RPF/CPF and the EnKF algorithms. The test results and some implementation issues are discussed. Note that the UKF algorithm is only considered as a reference method, thus its estimation performance is only evaluated in the comparison tests presented in Section 8.5.

8.3.1 Implementation descriptions

As stated previously, a possible choice of hidden state vector for the LNAS model is $X(t) = (Q_f(t), \gamma(t), Q_r(t))$. The functional parameter vector of our model is $\Theta = \{\mu_a, \mu_\gamma, \gamma_0\}$ and the error variances $\{\sigma^2_{\gamma\gamma}, \sigma^2_q, \sigma^2_g, \sigma^2_r\}$ are considered known.

Initialization

The initialization of the SMC based algorithms consists of the following steps :

- Define truncated normal prior distributions for each parameter of Θ: $p_0(\cdot)$ with the expectation μ_{p_0} and variance $\sigma^2_{p_0}$. Note that uniform or other form of priors can also be considered.

- For EnKF and RPF/CPF, the number of samples (particles) M is required to be defined in advance, while for UKF, the number of the sigma points is defined by $2(d_{x^a} + d_\eta) + 1$, with d_{x^a} the dimension of the augmented hidden state vector $dim(x^a_{n|n})$ and d_η the dimension of the modelling noise parameter vector $dim(\eta_n)$.

- Initialize $x^a_0 = \{\Theta_0, x_0\}$. For EnKF and RPF/CPF, set $\{{\Theta_0}^{(i)}\}_{1\leq i\leq M}$ by simulating one value for each parameter from the prior distribution $\Theta \sim p_0(\cdot)$ for each i. The hidden state variables $x_0^{(0)}$ are determined by a first allocation of the seed biomass between the root and leaf compartments (corresponding to germination). The weight associated to each sample (particle) is assigned uniformly

($\frac{1}{M}$ for each). For UKF, initialize the sigma points and their associated weights according to the unscented transform (details can be found in Section 4.3.1).

8.3.2 Regularized Particle Filter/Convolution Particle Filter

List of the tests for the RPF/CPF algorithm :

I	Choice of the kernel
II	Decomposition of variance
III	Influence of the number of particles
IV	Impact of priors

General settings :

The modelling noise parameters and the observation noise parameters are fixed at 0.02 and 0.1 respectively. Without specific test settings, the configuration of 250 000 particles is used for the restricted dataset and 100 000 for the full dataset.

I Choice of the kernel

For the RPF/CPF approach, one or two kernel functions K^X (or K^X and K^Y for CPF) associated respectively to X^a and Y are defined. Campillo and Rossi (2009) mentioned that in practice, the choice of the kernel did not affect appreciably the results. In our implementation, a Gaussian kernel is chosen for K^{X^a}. If d_{X^a} denotes the dimension of the augmented hidden state vector X^a, $K^X(x)$ can thus be regarded as a Parzen-Rozenblatt kernel for $|x|^{d_{X^a}} K^X(x) \to \infty$ as $|x| \to \infty$. Hence :

$$K^X_{h^X_M}(x) = \frac{1}{{h^X_M}^{d_{X^a}}} K^X\left(\frac{x}{h^X_M}\right) = \left(\frac{1}{\sqrt{2\pi} h^X_M}\right)^{d_{X^a}} e^{-|x|^2/{h^X_M}^2}.$$

where the positive bandwidth parameter h^X_M is defined as:

$$h^X_M = C_X M^{-1/(4+d_{X^a})},$$

which is optimal for the Mean Squared Error criteria. Likewise, when the density of the modelling noise is unknown, a similar kernel $K^Y_{h^Y_M}(y)$ can be defined for Y, replacing h^X_M by $h^Y_M = C_Y M^{-1/(4+d_Y)}$. It is proved that this choice of h^X_M and h^Y_M allows the L^1-convergence of $\hat{p}(x^a_n|y_{0:n})$ towards $p(x^a_n|y_{0:n})$ when $M \to \infty$ (see Campillo and Rossi (2009) and Rossi (2004)).

Another noteworthy point is that the choice of C_X is crucial for density estimation, since it is related to the dispersion of the particle population. Some sophisticated methods have been proposed, like in Devroye and Lugosi (2001), yet cannot be applied in the on-line estimation context. Generally, C_X is defined as:

$$C_X = c_x[\mathrm{Cov}(\tilde{x}^a_{n+1}{}^{(1)}, \ldots, \tilde{x}^a_{n+1}{}^{(M)})]^{1/2}.$$

In Campillo and Rossi (2009), authors mention that a $c_x \approx 1$ gives good results. Compared to the Gaussian kernel of the RPF (4.39), the value of c_x corresponds to $((4/(d+2))^{(1/(d+4))}$, with d the dimension of x^a_n. In our application to LNAS model, if $d = 5$ (3 parameters and 2 hidden states), then $c_x \approx 0.940$, if $d = 8$ (6 parameters and 2 hidden states), then $c_x \approx 0.926$.

Indeed, small values of c_x (for example < 0.001) imply that the particle $\tilde{x}^a_{n+1}{}^{(i)}$ will remain close to $\tilde{x}^{a\,(i)}_n$, which may constrain the movement of the particles in a limited zone, while a large value of c_x (≈ 0.8) allows the particles to explore more thoroughly the state space. Therefore, the value of c_x should be chosen with care to optimize the perturbation rate of the particles after the selection step.

In this study, we notice that $\mathrm{Cov}(\tilde{x}^a_{n+1}{}^{(1)}, \ldots, \tilde{x}^a_{n+1}{}^{(M)})$ is very important at the beginning of the iteration, especially when a dispersed prior is given. Therefore, when the number of data is limited as in our study case, an important c_x may result in a large population of the particles with small importance weights, which in turn causes sample degeneracy and instability of the algorithm. The obtained estimates may appear to be very unstable (large $\mathbb{V}(\mathbb{E}(\Theta|Y))$ when the same estimation process is repeated). In the worst scenario, either all the importance weights are close to 0 (≈ 0 numerically) and cause numerical problems, or the least worst particles are chosen, which leads to unsatisfactory estimations.

By keeping these constraints in mind, the adopted technique is to assign a large value to C_X from the start and decrease the value when moving forward with the filtering process in order to avoid the over-estimation of the variance as well as to improve the estimation accuracy.

Settings :

For the same simulated dataset, in restricted or complete form, five configurations of c_x are tested, including a time-evolving $\tilde{c}^*_x$ chosen as follows :

- For the full dataset,

$$\tilde{c}^*_x = 0.8 * 0.95^i,$$

where i denotes the number of the dates at which the filtering has already been performed;

- For the restricted dataset,

$$\tilde{c}^*_x = 0.5 * 0.7^i.$$

In this test, 100 independent runs are processed to evaluate the MCE of the algorithm. The estimation provided by the one chain AMwG is also listed for comparison purpose.

$$\Theta^* \Rightarrow Y \begin{array}{l} \xrightarrow{c_x=0.9} \{\tilde{\Theta}_i^1\}_{i=1,\ldots,100} \\ \xrightarrow{c_x=0.5} \{\tilde{\Theta}_i^2\}_{i=1,\ldots,100} \\ \xrightarrow{c_x=0.3} \{\tilde{\Theta}_i^3\}_{i=1,\ldots,100} \\ \xrightarrow{c_x=10^{-4}} \{\tilde{\Theta}_i^4\}_{i=1,\ldots,100} \\ \xrightarrow{c_x=\tilde{c}_x^*} \{\tilde{\Theta}_i^5\}_{i=1,\ldots,100} \end{array} \tag{8.14}$$

Observations and remarks :

Multiple configurations of c_x are tested for both a full dataset and a restricted dataset, the results of which are given in Table 8.12. With an important value of c_x, the variability among the estimates is very important. Although it allows the particles to explore the state space with more liberty, the mean estimates remain quite inaccurate compared to the results given by smaller value of c_x, so is the estimated Monte Carlo errors $\mathbb{V}(\mathbb{E}(\Theta|Y))$ (which can also be interpreted as the algorithmic errors).

Restricted data	μ_a			γ_0			μ_γ		
Real value	**3.592**			**0.7411**			**612.27**		
Prior	$\mathcal{N}(3.60,\ 0.10^2)$			$\mathcal{N}(0.75,\ 0.08^2)$			$\mathcal{N}(600.00,\ 20.00^2)$		
c_x	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$
0.9	3.536	9.00e-06	7.35e-03	0.7609	1.00e-06	7.98e-04	596.25	1.000	586.221
0.5	3.538	4.00e-06	4.81e-03	0.7614	4.90e-07	4.57e-04	595.96	0.706	390.734
0.3	3.538	1.00e-06	3.86e-03	0.7618	2.50e-07	3.59e-04	595.74	0.410	335.769
0.0001	3.533	1.02e-06	3.27e-03	0.7632	1.96e-07	3.05e-04	595.07	0.403	304.769
$\tilde{c}_x^*$	3.533	1.00e-06	3.29e-03	0.7624	1.60e-07	3.07e-04	594.48	0.303	306.610
AMwG$^\diamondsuit$	3.533	1.00e-06	3.37e-03	0.7622	2.50e-07	3.05e-04	594.77	0.372	306.190

Complete data	μ_a			γ_0			μ_γ		
Real value	**3.746**			**0.7525**			**579.00**		
Prior	$\mathcal{N}(3.60,\ 0.10^2)$			$\mathcal{N}(0.75,\ 0.08^2)$			$\mathcal{N}(600.00,\ 20.00^2)$		
c_x	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$
0.9	3.890	1.07e-03	9.80e-03	0.5381	0.2314	8.29e-03	702.34	257.250	182.352
0.0001	3.702	1.10e-03	4.67e-04	0.7623	1.20e-04	3.56e-05	579.77	140.353	40.870
$\tilde{c}_x^{**}$	3.730	3.09e-04	3.26e-04	0.7552	1.39e-05	4.38e-05	582.82	35.353	47.068
AMwG$^\diamondsuit$	3.730	1.61e-05	3.60e-04	0.7561	4.68e-06	4.26e-05	582.29	4.542	45.019

Table 8.12: Parameter estimation and uncertainty assessment provided by RPF based on the same target dataset (14 dates or 160 dates), with different values of c_x. The given estimations are based on 100 independent runs. $\tilde{c}_x^*$: c_x variable, start with 0.5 and decreases gradually $(0.5 * 0.7^i)$ with i the number of dates at which the filtering has already been performed. $\tilde{c}_x^{**}$: c_x variable, start with 0.8 and decreases gradually $(0.8 * 0.95^i)$. AMwG$^\diamondsuit$: Estimates obtained based on 100 independent runs of AMwG, configured with the same number of iterations as of particles for the RPF/CPF approach.

One thought-provoking point is that in the case of a restricted dataset, a very small c_x provides satisfactory estimations while it is not the case with a full dataset.

To understand this behaviour, it is important to understand the role of the correction step (by c_x) of CPF/RPF. The objective is to provide more dynamics to the particle population and to avoid degeneracy problems. However, when few observations are available with an important number of missing observations, the values of the covariance matrix usually keep inflating, therefore, there is no need to introduce more variability among the particles with the C_X, especially when a large number of particles are simulated. On the other hand, if abundance observations are available, since the resampling is performed at each time step when an observation is available, it is highly possible that the degeneracy problem could occur. Thus, the correction with C_X appears to be necessary to maintain the variability within the particle population.

Based on these arguments, a configuration strategy is proposed in this thesis to handle the two situations differently. In a general way, the c_x can be defined as :

$$c_x^n = c_x^0 * \alpha^{n-1}, \quad \text{with } \alpha \in [0; 1].$$

with n the number of time steps at which the filtering has been performed.

When there is a sufficient number of dates at which the observations data can be used for filtering, then the initial value assigned to c_x, denoted c_x^0, should be smaller than 1, but remain as close to 1 as possible, to assure a controlled exploration of the state-space without numerical issues occurred (too large covariances can result in a population of particles with associated weights close to 0). For the last filtering steps, the perturbations provided by C_X should be insignificant in order to assure that the algorithm converges to the real target distribution and no artificial variance is introduced. Thus, the value of c_x should be decreasing gently with the filtering steps and getting close to 0 at the end. An α being slightly smaller than 1 seems to be a good choice to maintain the dynamics of the population so as to avoid the degeneracy problems at the same time.

On the contrary, if there is an important lack of data, and an important number of particles are involved, then the objective is to prevent introducing too much variability to the particles, and to assure the convergence of them. Therefore, a smaller α is preferred.

In our implementation, we opt for $c_x^0 = 0.5$, $\alpha = 0.7$ for the restricted data, and $c_x^0 = 0.8$, $\alpha = 0.95$ for the full dataset. According to Table 8.12, the proposed strategy regarding the configuration of c_x appears to give satisfactory results for both the complete dataset and the restricted dataset.

II Decomposition of variance

In the previous test, the proposed strategy appears to work. In order to identify the stability of the method in terms of both mean and variance estimations, the following tests are carried out to verify the law of total variance $\mathbb{V}(\Theta) = \mathbb{V}(\mathbb{E}(\Theta|Y)) + \mathbb{E}(\mathbb{V}(\Theta|Y))$.

Settings :

300 parameter sets are generated from a given distribution $p_0(\cdot)$, and the corresponding observation datasets are simulated. The prior distribution used for this test is the same as the one used to generate the datasets :

$$\forall i \in 1,\ldots,300, \Theta_i^* \sim p_0(\cdot) \Rightarrow \{Y_i\}_{i=1,\ldots,300} \quad \begin{array}{l} \overset{c_x=0.9}{\longrightarrow} \{\tilde{\Theta}_i^1\}_{i=1,\ldots,300} \\ \overset{c_x=0.5}{\longrightarrow} \{\tilde{\Theta}_i^2\}_{i=1,\ldots,300} \\ \overset{c_x=0.3}{\longrightarrow} \{\tilde{\Theta}_i^3\}_{i=1,\ldots,300} \\ \overset{c_x=10^{-4}}{\longrightarrow} \{\tilde{\Theta}_i^4\}_{i=1,\ldots,300} \\ \overset{c_x=\tilde{c}_x^*}{\longrightarrow} \{\tilde{\Theta}_i^5\}_{i=1,\ldots,300} \end{array} \tag{8.15}$$

			Prior	
	μ_0^*	$\sigma_0^{2^*}$	μ_0	σ_0^2
μ_a	3.598	**0.010**	3.60	0.01
γ_0	0.7544	**0.0070**	0.750	0.0064
μ_γ	600.28	**398.98**	600	400.00

	Estimation	Variance decomposition		
c_x=0.9	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))+\mathbb{E}(\mathbb{V}(\Theta\|Y))$
μ_a	3.597	0.007	0.007	**0.014**
γ_0	0.7537	0.0070	0.0008	**0.0078**
μ_γ	598.41	54.64	573.36	**628.00**
c_x=0.5				
μ_a	3.599	0.007	0.005	**0.012**
γ_0	0.7545	0.0068	0.0005	**0.0073**
μ_γ	598.68	66.11	406.11	**472.22**
c_x=0.3				
μ_a	3.600	0.007	0.004	**0.011**
γ_0	0.7547	0.0068	0.0004	**0.0071**
μ_γ	598.47	76.17	340.58	**416.75**
c_x=0.0001				
μ_a	3.591	0.009	0.003	**0.012**
γ_0	0.7573	0.0098	0.0003	**0.0102**
μ_γ	599.10	102.71	306.18	**408.89**
$c_x=\tilde{c}_x^*$				
μ_a	3.594	0.007	0.003	**0.010**
γ_0	0.7568	0.0074	0.0003	**0.0077**
μ_γ	599.08	92.71	307.91	**400.62**
AMwG$^\diamondsuit$				
μ_a	3.599	0.007	0.003	**0.010**
γ_0	0.7543	0.0069	0.0003	**0.0072**
μ_γ	598.36	88.93	308.92	**397.85**

Table 8.13: Estimation and uncertainty assessment provided by RPF based on 300 simulated datasets (14 dates) generated with the parameter vectors simulated from the same prior distribution. Each test contains 250 000 simulations. AMwG$^\diamondsuit$: Estimates obtained based on 100 independent runs of AMwG, configured with the same number of iterations as of particles for the RPF/CPF approach. μ_0^* and $\sigma_0^{2^*}$ denote the sample mean and variance.

Observations and remarks :

Table 8.13 illustrates the decomposition of the variance based on a population of datasets. The adaptive strategy proposed for the choice of c_x showed the best performance compared to the other configurations. Both the mean estimates and the variance estimates are quite accurate, which suggests a satisfactory performance of this particle filtering method. The resulting variances are the closest to the target ones, and remain comparable with the MCMC estimates.

N.B.

- In the same way, if K^Y is required, the associated constant C_Y can be defined as:

$$C_Y = c_y[\text{Cov}(\tilde{y}_{n+1}{}^{(1)}, \ldots, \tilde{y}_{n+1}{}^{(M)})]^{1/2},$$

where the constant c_y can be assigned in a way that the particle weights remain in a reasonable range so as to avoid the numerical instability.

- In our implementation, the likelihood of the observations is considered known. Hence, the RPF algorithm is chosen to perform the parameter estimation rather than the CPF algorithm.

- The CPF algorithm allows to introduce more dynamics to the model than the RPF algorithm for it approximates the likelihood function by defining another kernel for Y. Therefore, it can also perform with deterministic models, which is a very desirable feature of this method, as will be illustrated in Section 9.3.

III Influence of the number of particles

One important configuration required in the initialization is the number of particles. In this test, we seek to find a good compromise between the computational cost and the precision of the estimations.

Settings :

Six configurations of M (which denotes the number of particles) are tested based on the same restricted dataset. Algorithmic error is evaluated by 100 repetitions of independent runs.

$$\Theta^* \Rightarrow Y \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\ldots,100}$$

Observations and remarks :

The estimation results illustrated by Table 8.14 suggest that the increase of the number of particles to a certain level can improve the precision, but then it is less helpful in terms of the averaged estimation accuracy. However, it does help to reduce the estimated Monte Carlo errors $\mathbb{V}(\mathbb{E}(\Theta|Y))$, which is understandable since MCE decreases when the number of samples increases.

	Theoretical value*		M =350 000			200 000			100 000		
	$\tilde{\mu}^*$	$\tilde{\sigma^2}^*$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$
μ_a	3.749	3.47e-3	3.7439	3.42e-3	1.95e-6	3.7440	3.42e-3	4.11e-6	3.7448	3.44e-3	5.27e-6
γ_0	0.7526	3.1e-4	0.7538	3.0e-4	2.20e-7	0.7538	3.0e-4	4.10e-7	0.7536	3.0e-4	5.69e-7
μ_γ	573.21	307.89	571.06	298.79	0.416	571.07	298.05	0.553	571.32	298.29	0.987
	Prior		M =50 000			30 000			10 000		
	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$
μ_a	3.6	0.01	3.7452	3.46e-3	1.35e-5	3.7451	3.40e-3	2.15e-5	3.7462	3.47e-3	6.98e-5
γ_0	0.75	0.0064	0.7535	3.0e-4	1.21e-6	0.7535	3.0e-4	2.28e-6	0.7536	3.0e-4	7.13e-6
μ_γ	600.0	400.00	571.60	294.50	2.062	571.73	292.66	4.294	571.79	293.29	10.320

Table 8.14: Evolution of estimations and uncertainty assessments provided by RPF, with the increase of the number of particles (M), based on the same simulated dataset (14 dates). For each configuration, the mean and variance estimates are computed based on 100 repetitions. *: The theoretical values are provided by taking the average of three long runs of AMwG with 5000 000 iterations.

IV Impact of priors

The prior distribution has established its importance, especially when a restricted dataset is presented. In this test, we look to identify the impact of priors on the RPF estimates.

Settings :

Like in the study case of the one chain AMwG algorithm, 350 000 particles are simulated based on the same restricted dataset to estimate the augmented hidden state vector. Three priors are used, the averaged variance and expectation are obtained based on 100 independent runs.

$$\Theta^* \Rightarrow Y \quad \begin{array}{l} \xrightarrow{p_1(\cdot)} \{\tilde{\Theta}_i^1\}_{i=1,\ldots,100} \\ \xrightarrow{p_2(\cdot)} \{\tilde{\Theta}_j^2\}_{j=1,\ldots,100} \\ \xrightarrow{p_3(\cdot)} \{\tilde{\Theta}_k^3\}_{k=1,\ldots,100} \end{array}$$

Observations and remarks :

- The estimation results obtained based on different priors are very different according to Table 8.15, which confirms the important impact of the prior for the restricted dataset. The variance estimations are obviously correlated to the prior distribution, which confirms that an appropriate uncertainty assessment relies on an appropriate prior.

- Little difference between the MCMC estimates and the RPF estimates is noticed. Further comparison is needed in this regard to compare the estimation quality of these two types of methods.

- The estimated MCE is more important with the Prior 2 probably because of the important distance between the expectation of the prior distribution and the real

value used to generate the data.

	Real	Theoretical	AMwG		**Prior 1**		Estimation		MCE	MSE
	value	value*	μ^*	σ^{*2}	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	
μ_a	3.746	3.749	3.748	0.0035	3.60	0.01	3.744	0.0034	0.0014	2.53e-5
γ_0	0.7525	0.7526	0.7527	0.00031	0.750	0.0064	0.7538	0.00047	4.69e-4	1.52e-6
μ_γ	579.00	573.21	573.24	307.71	600	400	571.06	298.79	0.6448	5.0650
	Real	Theoretical	AMwG		**Prior 2**		Estimation		MCE	
	value	value*	μ^*	σ^{*2}	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	MSE
μ_a	3.746	3.818	3.818	0.0036	3.70	0.01	3.813	0.0035	0.0026	3.10e-5
γ_0	0.7525	0.7230	0.7231	0.00026	0.720	0.0064	0.7244	0.00025	7.13e-4	2.26e-6
μ_γ	579.00	614.97	614.91	315.91	650	400	612.10	299.93	1.08	9.35
	Real	Theoretical	AMwG		**Prior 3**		Estimation		MCE	
	value	value*	μ^*	σ^{*2}	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	MSE
μ_a	3.746	3.803	3.804	0.0050	3.752	0.0225	3.794	0.0048	0.0011	8.58e-5
γ_0	0.7525	0.7453	0.7453	0.00042	0.744	0.0100	0.7477	0.00041	3.82e-4	6.01e-6
μ_γ	579.00	563.06	562.94	430.93	590.695	625	559.92	418.70	0.42	10.05

Table 8.15: Estimation and uncertainty assessment provided by RPF based on one simulated dataset (14 dates) with 100 repetitions. Each test contains 350 000 simulations, and three prior distributions are used. MCE: the estimated Monte Carlo error. The AMwG estimates are obtained based on 100 independent runs with each 350 000 iterations. *: The theoretical values are provided by taking the average of three long runs of AMwG with 5000 000 iterations. The Mean Squared Errors (MSE) are evaluated based on the theoretical values.

8.3.3 EnKF

The difference between the EnKF and the UKF is that EnKF is less dependent of the normal assumption and has larger sample size, so that in applications with important scales, EnKF usually performs better. Since EnKF is regarded as a reference method in our application, here, only the optimal configuration of the sample size is studied.

Settings :

Six configurations of ensemble sample size are considered. Based on the same restricted dataset, 100 independent runs are processed to evaluate the algorithmic errors of this method.

$$\Theta^* \Rightarrow Y \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\dots,100}$$

Observations and remarks :

According to Table 8.16, we notice very slight changes among the estimates when the ensemble size increases. As expected, the estimated MCE (provided by $\mathbb{V}(\mathbb{E}(\Theta|Y))$) decreases with the increase of the ensemble size. Yet, the estimations remain quite different from those given by RPF and AMwG.

Further investigations are needed for a proper comparison of this Kalman filter based method with the MCMC based methods and particle filtering based methods.

	Real value	Theoretical value	350 000			200 000			100 000		
			$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$
μ_a	3.746	3.749	3.7333	3.73e-3	4.76e-8	3.7332	3.73e-3	9.26e-8	3.7329	3.73e-3	1.60e-7
γ_0	0.7525	0.7526	0.7131	1.08e-3	1.46e-8	0.7131	1.08e-3	2.23e-8	0.7132	1.08e-3	6.83e-8
μ_γ	579.00	573.21	590.25	366.70	6.86e-3	590.28	366.48	1.11e-2	590.30	366.39	2.37e-2
	Prior		50 000			30 000			10 000		
	μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$
μ_a	3.6	0.01	3.7321	3.73e-3	3.84e-7	3.7313	3.70e-3	6.81e-7	3.7272	3.80e-3	2.89e-6
γ_0	0.75	0.0064	0.7134	1.08e-3	9.99e-8	0.7136	1.08e-3	1.97e-7	0.7146	1.08e-3	4.97e-7
μ_γ	600.0	400.00	590.28	366.45	4.06e-2	590.30	366.84	1.10e-1	590.50	366.94	2.19e-1

Table 8.16: Evolution of estimations and uncertainty assessments provided by EnKF, with the increasing ensemble sizes, based on the same simulated dataset (14 dates). For each configuration, the mean estimates are based on 100 independent runs. *: The theoretical values are provided by taking the average of three long runs of AMwG with 5000 000 iterations.

8.4 Implementation of the Generalized Least Squares method

As a classical frequentist approach widely used in plant growth modelling (Cournède et al., 2011; Guo et al., 2006), the Generalized Least Squares Estimator is applied to the LNAS model with the objective of providing a comparison between the estimation results of this frequentist approach and the Bayesian approach in the following sections.

The version of the GLS estimator applied is described in Section 3.4.1, with a diagonal error covariance matrix initially chosen, consisting of equal error variances for all Q_g and all Q_r respectively. The multiplicative error assumption is made. Note that the estimator does not support modelling noises, thus the modelling noise parameters σ_q and $\sigma_{\gamma\gamma}$ are set to 0 during the estimation. Moreover, to apply the GLS estimator, an initialization of the departure point of each parameter is required. The algorithm is stopped when the difference between the estimates given by two successive iterations is small enough. A threshold is accordingly defined.

As a reference method like EnKF and UKF, here we only investigate the influence of initialization on the GLS estimates.

Influence of initialization

In the following tests, the mean values of the prior distribution defined for the Bayesian methods are used as the initial points. The objective is to compare the results provided by two different sets of departure points.

Settings :

Based on the restrict dataset generated with small modelling noises ($\sigma_{\gamma\gamma}$ and σ_Q set to 0.02) and observation noises (σ_g and σ_r set to 0.1), two different initializations

regarding the departure parameter sets are considered. The estimates of the last iteration and the estimated variance are provided.

Observations and remarks :

We test the influence of the initial departure points in the following test with a simulated restricted dataset (14 dates).

Table 8.17 suggests that the estimation of GLS does not depend on the initial points. Both the mean estimates and the variances estimates remain the same when the algorithm is initialized with two sets of very different points. The log-likelihood evaluation of the estimates is better than the evaluation of the real parameter set used to generate the data, probably due to the fact that in the case of restricted dataset, there are more possible trajectories than in the scenario of full dataset, therefore the MLE is not necessarily obtained with the real parameters set (and often not the case in practice). In the meantime, it is noteworthy that since GLS cannot cope with modelling noises, for those stochastic models built with modelling noise(s), GLS estimates are not necessarily the MLE.

	Real value	Initialization 1	Estimates	
		μ_0	Est.	Var
μ_a	3.746	3.40	**3.732**	0.0044
γ_0	0.7525	0.90	**0.7940**	0.00083
μ_γ	579.00	650.00	**494.47**	935.55
	Real value	Initialization 2	Estimates	
		μ_0	Est.	Var
μ_a	3.746	3.80	**3.732**	0.0044
γ_0	0.7525	0.40	**0.7940**	0.00083
μ_γ	579.00	350.00	**494.47**	935.55
$-2\mathcal{L}$	311.06	-	300.53	-

Table 8.17: Estimation and uncertainty assessment provided by one chain GLS based on one simulated dataset (14 dates), without noises. Two initial sets are used.

8.5 Method comparison

With the purpose of identifying the estimation performance of the previous algorithms, some tests of comparisons are conducted in this thesis to evaluate the parameter estimation performance of the methods investigated, including AMwG, DREAM, Interacting parallel MCMC, RPF/CPF, EnKF, UKF and GLS.

List of the comparison tests :

General settings :

As for the previous tests, if without specific notice, the modelling noise parameters and the observation noise parameters are fixed at 0.02 and 0.1 respectively. Multivariate proposals are used without global scaling for MCMC-based methods. For the Interacting MCMC algorithm, the original version is used, which means that no adaptive scale is used to tune the proposals. A variable tuning parameter c_x^* is defined for the RPF algorithm using the strategy proposed in Section 8.3.2. The DREAM algorithm is performed with 9 chains, while the Interacting parallel MCMC uses 1000 chains. In order to provide a reference value for the sake of comparison, three independent long chains of AMwG (5000 000) are carried out to define the theoretical values of the estimation, on which the Mean Squared errors are evaluated for all the Bayesian methods.

8.5.1 Precision on one dataset

In this test, we aim to investigate the precision of the estimation given by the seven methods that we implemented in this thesis, both in terms of the averaged mean estimation and the variance estimation based on a single restricted dataset. Since from a Bayesian point of view, the objective is to update the knowledge based on the new observations, the prior distribution plays an important role.

Settings :

With a fixed total number of simulations, the MCMC-based methods (one chain AMwG, DREAM, Interacting MCMC) and the filtering based methods (UKF, EnKF, RPF) are subsequently performed with each 100 independent runs based on the same simulated restricted dataset. The DIC is used to evaluate their estimates. The GLS estimates are also given as a reference result which is obtained based on the deterministic form of the LNAS model.

$$\Theta^* \Rightarrow Y \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\dots,100}$$

Observations and remarks :

According to Table 8.18, all the seven methods appear to be able to perform the estimation with satisfactory estimation quality. Judging from the averaged estimates, all the MCMC-based methods provide similar estimations and small MSE, while the filtering methods disagree with each other.

Furthermore, if we are under the assumption that the estimates of one very long chain MCMC are the theoretical values, then we consider the difference between the MSE and $\mathbb{V}(\mathbb{E}(\Theta|Y))$ as the bias. MCMC-based methods appear to have smaller bias than the filtering methods.

	Real value		Prior		AMwG		DREAM		Interacting	
			μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$
μ_a	3.746		3.6	0.01	3.748	0.0035	3.749	0.0035	3.748	0.0020
γ_0	0.7525		0.75	0.0064	0.7527	0.00031	0.7531	0.00031	0.7526	0.00017
μ_γ	579.00		600.0	400.00	573.24	307.71	572.79	314.95	573.35	172.54
$-2\mathcal{L}(\bar{\Theta})$	311.06		346.40		308.11		308.22		308.19	
DIC	-		340.06		296.21		297.67		298.74	
	Theoretical value*		GLS		RPF		EnKF		UKF	
	$\tilde{\mu}^*$	$\tilde{\sigma^2}^*$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$
μ_a	3.7487	0.0035	3.732	0.0044	3.744	0.0034	3.733	0.0037	3.634	0.0065
γ_0	0.7526	0.00031	0.7940	0.00083	0.7538	0.00030	0.7131	0.00108	0.7373	0.00148
μ_γ	573.21	307.89	494.47	935.56	571.06	298.80	590.25	366.70	598.62	348.60
$-2\mathcal{L}(\bar{\Theta})$	308.10		300.53		307.73		325.30		326.96	
DIC	296.19		-		296.32		296.52		312.94	

Table 8.18: Estimation and uncertainty assessment provided by one chain AMwG, DREAM, Interacting AMWG, RPF, UKF, EnKF and GLS based on the same simulated dataset (14 dates) with 100 independent runs. All the test contains 350 000 simulations, except for GLS and UKF (number of sigma points predefined). The same prior is shared for all the methods. *:

	Real value		Prior		AMwG		DREAM		Interacting	
			μ_0	σ_0^2	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	MSE	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	MSE	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	MSE
μ_a	3.746		3.6	0.01	2.48e-7	2.77e-7	4.95e-6	5.39e-6	1.23e-5	1.24e-5
γ_0	0.7525		0.75	0.0064	2.44e-8	2.58e-8	4.29e-7	6.77e-7	1.26e-6	1.26e-6
μ_γ	579.00		600.0	400.00	2.65e-2	2.70e-2	4.72e-1	6.50e-1	1.91e-1	2.05e-1
	Theoretical value♠		GLS		RPF		EnKF		UKF	
	$\tilde{\mu}^*$	$\tilde{\sigma^2}^*$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	MSE*	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	MSE	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	MSE	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	MSE
μ_a	3.7487	0.0035	-	-	1.95e-6	2.53e-5	4.76e-8	2.39e-4	8.98e-5	1.32e-2
γ_0	0.7526	0.00031	-	-	2.20e-7	1.52e-6	1.46e-8	1.56e-3	3.45e-5	2.69e-4
μ_γ	573.21	307.89	-	-	4.16e-1	5.06	6.86e-3	290.23	3.91	649.04

Table 8.19: Comparison of the estimated Monte Carlo errors of one chain AMwG, DREAM, Interacting AMWG, RPF, UKF and EnKF. The Mean Squared Error (MSE) of the proposing estimates is evaluated based on the same simulated dataset (14 dates) with 100 independent runs. The same prior is shared for all the Bayesian methods. Each test contains 350 000 simulations except for DREAM and UKF, for which the recommended number of simulations are used. MSE*: the MSE of the frequentist GLS method cannot be evaluated with the theoretical values provided by the Bayesian based AMwG.

Based on the evaluation of the posterior distribution given by the DIC, the one chain AMwG, RPF and EnKF show similar performances. The one long run AMwG scores the best DIC among all the methods, which confirms that its result could be considered as the theoretical target distribution.

DREAM is performed with $3d = 9$ chains based on the tests of Section 8.2.3. Therefore, the variance estimation of one chain AMwG, RPF and DREAM are relatively close. RPF has smaller $\mathbb{E}(\mathbb{V}(\Theta|Y))$ than one chain AMwG, but more important

variation of its estimates $\mathbb{V}(\mathbb{E}(\Theta|Y))$ which implies larger MCE (algorithmic error), hence it is less stable than one chain AMwG in this test case.

Table 8.19 illustrates the MSE for each of the methods regarding the mean estimates. The variability of the estimates $\mathbb{V}(\mathbb{E}(\Theta|Y))$ provided by RPF and Interacting MCMC is similar, which could be understood by the fact that the Interacting MCMC is based on the importance sampling technique as the RPF. We also notice that both multiple chain MCMC methods, DREAM and Interacting MCMC fail to estimate the variance of the posterior distribution in an accurate way. For DREAM, it is highly possible that outlier samples are included in the pooled samples used to build the posterior distributions, while for Interacting MCMC, either a sample size of 1000 is not sufficient to capture and to characterize the variability of the target distribution, or the posterior distribution is biased, for the states with small likelihood evaluations are seldomly reached.

Finally, the log-likelihood evaluation of the mean estimates suggests that the GLS estimator provides the best point estimation. However, when we have reliable prior information and we desire an estimate with interpretable biological explanations, the Bayesian approaches are preferred.

8.5.2 General behaviour and coverage comparison

After studying the performance of the methods performed on one single dataset, we are also interested in their more global performance, including the assessment of their post-experimental coverage probability.

Settings :

For this test, we study the estimation precision of those methods when dealing with the 1000 simulated datasets generated with the same parameter set. The confidence intervals are bootstrap percentile intervals, constructed based on the 2.5% percentile (lower bound, denoted lb) and the 97.5% percentile (upper bound, denoted ub) of the 1000 bootstrap mean estimates ($\mathbb{E}(\Theta|Y)$). The post-experimental coverage is computed based on the Bayesian credibility intervals provided by each test.

$$\Theta^* \Rightarrow \{Y_i\}_{i=1,\dots,1000} \Rightarrow \{\tilde{\Theta}_i\}_{i=1,\dots,1000}$$

Observations and remarks :

Regarding the averaged estimated values based on a large number of simulated datasets (1000) generated with one parameter set (Table 8.20), the three MCMC-based methods and RPF provide similar results. The fact of performing 1000 repetitions of the tests with the same parameter set allows us to average the observation noises effect and the modelling noise effect.

	Real value	Prior		GLS				
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	lb_B	ub_B	Coverage*
μ_a	3.7458	3.6	0.01	3.7394	0.0072	3.5373	3.9216	100%
γ_0	0.7525	0.75	0.0064	0.7587	0.00104	0.6923	0.8393	98.2%
μ_γ	579.00	600.0	400.00	576.71	1426.34	469.63	657.79	94.7%
	AMwG					Interacting MCMC		
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	lb_B	ub_B	Coverage	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	lb_B	ub_B	Coverage
μ_a	3.7070	0.0034	3.6193	3.7916	95.9%	3.7060	0.0019	3.6222	3.7888	83.2%
γ_0	0.7564	0.00030	0.7317	0.7804	98.8%	0.7563	0.00021	0.7314	0.7789	94.4%
μ_γ	591.16	306.06	575.018	607.14	99.8%	591.29	189.77	574.25	606.36	94.6%
	DREAM					RPF				
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	lb_B	ub_B	Coverage	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	lb_B	ub_B	Coverage
μ_a	3.7071	0.0035	3.6218	3.8020	92.3%	3.7086	0.0033	3.6235	3.8039	94.9%
γ_0	0.7563	0.00031	0.7295	0.7821	94.0%	0.7560	0.00030	0.7333	0.7843	98.6%
μ_γ	591.65	315.73	573.91	608.76	95.1%	591.05	303.70	575.65	608.50	99.7%
	EnKF					UKF				
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	lb_B	up_B	Coverage	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	lb_B	up_B	Coverage
μ_a	3.7003	0.0122	3.6148	3.7748	99.9%	3.7224	0.0068	3.6125	3.8534	98.4%
γ_0	0.7680	0.0038	0.7042	0.8418	100%	0.7531	0.0016	0.7019	0.7923	99.3%
μ_γ	594.77	366.21	587.33	601.66	99.8%	596.94	349.47	582.13	610.43	91.2%

Table 8.20: Estimation and uncertainty assessment provided by one chain MCMC, DREAM, Interacting AMWG, RPF, UKF, EnKF and GLS based on 1000 simulated datasets (14 dates) generated with the same parameter set. Each test contains 350000 simulations except for DREAM and UKF, with which the recommended number of simulations are used, and the same prior is shared for all the methods.

Among the Bayesian based methods, and the rest of the results also show great coherence, since they all consider the same prior knowledge. The EnKF's estimation of variance is much more important than those given by the other methods and UKF appears to suffer from the insufficient sigma point sample size.

The one chain AMwG and RPF propose globally very similar results for both the averaged estimates and the variance estimates. Their coverage probabilities also indicate that in terms of risk management, they are more reliable and better optimize the posterior credibility interval than the other methods do. It is important to note that although some methods have better coverage probability than these two methods, they also provide much larger variance estimates, which does not correspond to the objective of optimizing the coverage probability while keeping the CI as narrow as possible.

Remarks have also been made that the variance estimates are still greatly biased in the case of Interacting MCMC, illustrating again that the algorithm suffers from the underestimation of variability.

Comparison of the computational cost

The computational cost plays an important role in numerical applications. Based on the test 8.5.1 conducted based on one dataset (which result is given in Table 8.18), the mean computational time and memory consummation are listed as follows :

	GLS	UKF	EnKF	AMwG	Interacting MCMC	DREAM	RPF/CPF
Number of simulations	20	15	350 000	350 000	350 000[a]	349 998[b]	350 000
CPU time	2s	2s	22m36s	22m20s	8h31m	1h34m	30m34s
Real computational time	2s	2s	12m17s	22m20s	1h49m	17m24s	14m30s
Memory (Gb)	5e-3	4e-3	5.02	4.80	5.74	8.75	22.76

Table 8.21: The averaged computational cost of one test with simulated dataset (14 dates) performed by one chain AMwG, DREAM, Interacting AMWG, RPF, UKF, EnKF and GLS based on the 1000 tests conducted with different datasets. For GLS, the estimation is performed based on the deterministic form of the LNAS model. [a]: Interacting MCMC is performed with 1000 chains and 350 iterations. [b]: DREAM is performed with 9 chains and 38889 iterations.

Observations and remarks :

Among the seven methods, only the one chain AMwG algorithm cannot be parallelized. However, the parallel computing does not appear to optimize the efficiency of all the algorithms, especially for the multiple-chain algorithms.

Considering both the real computational time and the estimation precision, EnKF, one chain AMwG and RPF provide the best compromise and thus are considered to have outperformed the other methods in this study case.

8.5.3 Influence of the modelling noise level on scarce dataset

In this test, we intend to identify the impact of the modelling noise level on the seven studied methods, both in terms of stability and accuracy.

Settings :

Three levels are defined for modelling noise parameters η : 0, 0.02 and 0.05. 600 parameter sets generated from the same distribution are used to generate observation datasets with different levels of modelling noises. The observation noise level is fixed to 0.1. According to the previous tests, the one chain AMwG, the Interacting MCMC, EnKF and RPF, representing each an estimation technique, are applied to the 1800 datasets. Considering that in practice, no exact priors are known, flatter prior distribution is defined compared to the real distribution used to generate $\{\Theta_i\}_{i=1,\dots,600}$.

$$\forall i \in 1,\dots,600, \Theta_i^* \sim p_0^*(\cdot) \quad ,\forall \Theta_i \begin{array}{ll} \xrightarrow{\eta_1} Y_i^1 & \Rightarrow \{\tilde{\Theta}_i^{\eta_1}\}_{i=1,\dots,600} \\ \xrightarrow{\eta_2} Y_i^2 & \Rightarrow \{\tilde{\Theta}_i^{\eta_2}\}_{i=1,\dots,600} \\ \xrightarrow{\eta_3} Y_i^3 & \Rightarrow \{\tilde{\Theta}_i^{\eta_3}\}_{i=1,\dots,600} \end{array}$$

	Distribution of Θ^*		Prior	
	$\mathbb{E}(\Theta_i^*)$	$\mathbb{V}(\Theta_i^*)$	μ_0	σ_0^2
μ_a	3.5954	**0.0104**	3.60	0.04
γ_0	0.7519	**0.0058**	0.750	0.01
μ_γ	599.92	**400.784**	650	900.00

Observations and remarks :

Generally speaking, according to Table 8.22, the four selected methods provide comparable results. Yet the mean estimates are still far from the distribution of Θ^*, which implies a general lack of information. Considering the scarce dataset that we have provided for this test, it is completely understandable.

Regarding the averaged mean estimation and the averaged variance estimation, the results given by the one chain AMwG and RPF share much similarity. The estimations of RPF have more important variability than those of the one chain AMwG ($\mathbb{V}(\mathbb{E}(\Theta|Y))$), however, the variance estimations $\mathbb{E}(\mathbb{V}(\Theta|Y))$ of RPF are slightly less important. It refers to the between-particle variation for RPF and to the intra-chain variation for AMwG. This result indicates that there is a small bias regarding the variance estimation of RPF.

	Modelling noise 0				**Modelling noise 0.02**				**Modelling noise 0.05**			
	Adapted Metropolis-within-Gibbs											
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a	3.6218	0.0117	0.0048	4.61e-07	3.6265	0.0117	0.0049	4.57e-07	3.6260	0.0119	0.0053	4.30e-07
γ_0	0.7402	0.00541	4.19e-4	6.41e-09	0.7339	0.00538	4.23e-4	6.49e-09	0.7269	0.00541	4.76e-4	6.66e-09
μ_γ	629.80	223.53	558.82	645.37	630.48	226.99	560.32	649.21	632.36	234.86	599.79	897.25
	Interacting MCMC											
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a	3.6215	0.0138	0.0030	3.36e-7	3.6254	0.0136	0.0029	2.68e-7	3.6255	0.0132	0.0034	3.34e-7
γ_0	0.7399	0.00580	2.19e-4	2.33e-9	0.7380	0.00579	2.33e-4	2.33e-9	0.7215	0.00526	2.73e-4	2.07e-9
μ_γ	630.32	286.34	306.99	1119.75	631.84	288.62	309.97	1031.06	631.30	257.98	335.27	1272.48
	RPF											
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a	3.6214	0.0119	0.0047	7.61e-07	3.6263	0.0130	0.0048	7.18e-07	3.6252	0.0122	0.0053	7.28e-07
γ_0	0.7410	0.00546	4.10e-4	6.07e-09	0.7357	0.00545	4.17e-4	6.04e-09	0.7273	0.00583	4.68e-4	6.17e-09
μ_γ	629.69	229.53	553.77	786.46	630.98	229.40	561.76	766.90	632.23	230.94	598.36	777.42
	EnKF											
	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{V}(\Theta\|Y))$	$\mathbb{V}(\mathbb{V}(\Theta\|Y))$
μ_a	3.6029	0.0137	0.0014	4.42e-8	3.5978	0.0117	0.0014	5.50e-8	3.6071	0.0117	0.0014	4.85e-8
γ_0	0.7107	0.00670	0.00632	2.63e-6	0.7173	0.00606	0.00622	2.46e-6	0.7131	0.00642	0.00663	2.60e-6
μ_γ	646.55	34.87	825.89	841.76	647.14	33.36	823.23	970.77	646.43	36.87	826.82	992.02

Table 8.22: Estimation and uncertainty assessment provided by one chain AMwG, Interacting MCMC, RPF and EnKF based on 600 simulated datasets (14 dates) generated with the same parameter set, but with different level of modelling noises. The given estimations are tests with a maximum of 350 000 iterations for each, taking into account a burn-in period of 10 000 iterations.

Furthermore, by considering at the amplitude of the fluctuation of the variance estimation $\mathbb{V}(\mathbb{V}(\Theta|Y))$, we highlight the fact that RPF appears to be the most stable method, for the variance estimation of the parameter μ_γ. In fact, the drastic increase of the variance among the variance estimations given by the one chain AMwG ($\mathbb{V}(\mathbb{V}(\Theta|Y))$) in the case of μ_γ suggests that the method does not cope well with the increase of the noise level. Yet, in a general way, the estimations of noise level 0 and 0.02 are closer compared to those of the noise level 0.05, which implies a common lack

of precision when the model is too noisy.

The modelling noise level does not appear to influence the estimation in a remarkable way, still, we notice that the mean estimates get closer to Θ^* when the noise level decreases for the parameter γ_0, from the results of the three first methods (except EnKF). The variability between the estimates (*cf* $\mathbb{E}(\mathbb{V}(\Theta|Y))$) tends to be slightly more important in the case of heavier modelling noises for all the parameters, particularly the less sensitive ones, μ_a and μ_γ. The EnKF estimations again make an exception. In fact, EnKF seems to be the method least influenced by the noise level. Nevertheless, its estimations are very close to the prior distributions and the associated variance between the estimates are very small as well, which suggest that it has not fully assimilated the information in the observations.

Based on the result, the Interacting MCMC still suffers from the variability estimation. Not only its mean estimations are smaller than those of other methods, but the huge variability of the variance estimations confirms this assumption. Therefore, the two most robust methods regarding this test are the one chain MCMC and CPF. In the following, we will concentrate on investigating these two methods. The prior influence is studied in the first place.

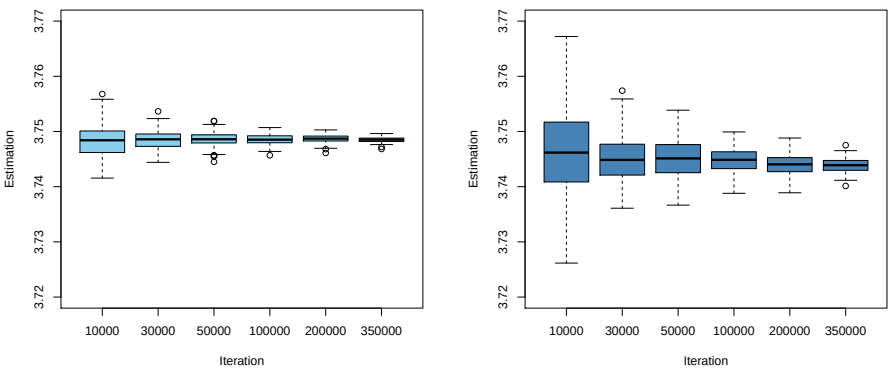
8.5.4 Efficiency comparison

Based on the former results, the RPF with time-evolving tuning parameter c_x and the one chain AMwG illustrate the most convincing estimation performance. In this test, the impact of the number of simulations are reviewed for both methods. The aim is to identify the more efficient method.

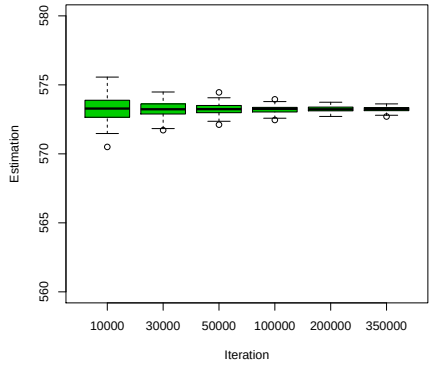
Settings :

The estimation is performed based on the same restricted dataset with 100 independent runs. The boxplot of estimations provided by the one chain AMwG and the RPF under various configurations are displayed in Table 8.15.

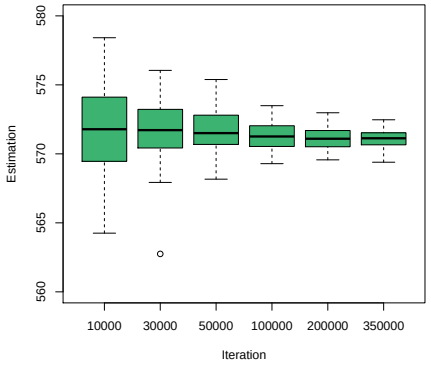
Observations and remarks :



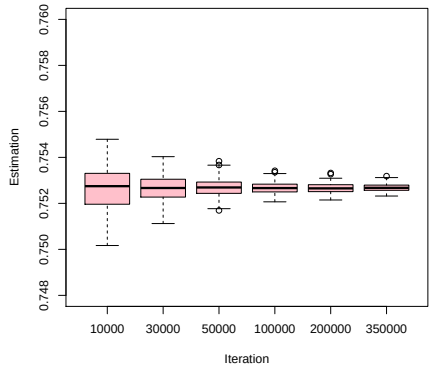
(a) μ_a (AMwG) (b) μ_a (RPF)



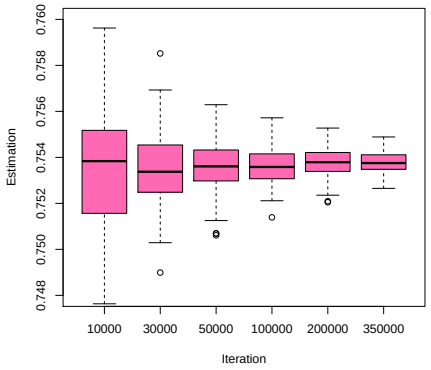
(c) μ_γ (AMwG)



(d) μ_γ (RPF)



(e) γ_0 (AMwG)



(f) γ_0 (RPF)

Figure 8.15: Boxplot of 100 mean estimates based on 100 independent runs of the same test performed with different numbers of iterations for the three parameters μ_a, μ_γ, γ_0 respectively. Left panel: estimations given by Adapted Metropolis-within-Gibbs. Right panel: estimations given by RPF.

There is obviously less variation among the mean estimates of the one chain AMwG compared to those of the RPF, which indicates that the one chain AMwG is much more stable while the RPF estimates could be biased.

8.6 Implementation of Iterative SMC and MCMC algorithms

To put more emphasis on the observation data, in the following, the frequentist based iterative algorithms are applied to the LNAS model with the simulated data. As for Bayesian based methods and for GLS, three functional parameters are estimated.

In this section, we first detail the implementation description of the iterative approach to the LNAS model. Then, two tests are conducted to study the performance

of this EM based approaches in the case when only the three functional parameters are estimated. Subsequently, the results are presented with the objective of giving some directions when confronting real experimental data.

List of the tests for the iterative approaches :

8.6.1 Implementation descriptions

Gaussian randomization

In this part we apply the ideas developed in Section 5.1.1 for the state space model described in Lemma 8.1.1. In the first place, we assume for simplicity of presentation that all error variances $\{\sigma^2_{\gamma\gamma}, \sigma^2_q, \sigma^2_g, \sigma^2_r\}$ are known or a priori fixed to some values. The interest will be focused on the estimation of the structural parameters and let $\Theta = (\mu_a, \lambda, \mu_\gamma, \sigma_\gamma, \gamma_0, \gamma_f)$. Note also that this state space model can equivalently be described as a non-homogeneous hidden Markov model Cappé et al. (2005), and usual estimation techniques are based on some variant of the EM algorithm, by combining an approximation of the E-step and a usually explicit M-step. Nevertheless, we can easily see that there is no explicit solution for Θ even in the complete log-likelihood of this model, therefore leading to a non-explicit M-step. Otherwise, conditional maximization approaches should be invoked and this often leads to complicated generalized EM-type algorithms (see, eg., Trevezas and Cournède (2013)), where a numerical maximization procedure should be implemented as well.

It is also well known (see, eg., McLachlan and Krishnan (2008)) that the EM-type sequences do not necessarily converge to the true MLE (if it exists). In typical plant growth models with a large number of parameters, it is very difficult to ensure the existence of the MLE and the subsequent convergence of the EM iterates to the true MLE. The best that we can hope is to have a robust algorithm and the solutions to the initial maximization problem to be biologically relevant. For this reason, we take Assumption 1 presented in Section 5.1.1 for granted.

When the noise parameters are fixed, let $\Theta = (\mu_a, \lambda, \mu_\gamma, \sigma_\gamma, \gamma_0, \gamma_f)$ when six functional parameters are estimated, or $\Theta = (\mu_a, \mu_\gamma, \gamma_0)$ when only three of them are estimated.

The results of Section 5.1.1 are directly applicable to Θ. In particular, let $\widetilde{m}(\Theta)$ the Gaussian randomization of m w.r.t. Θ. By Definition 5.1.1 we have that $\widetilde{m}(\Theta)$ is an incomplete data model which consists in:

i) a Gaussian hidden vector Ψ, where

$$\Psi \sim \mathcal{N}_{d_\Theta}(\eta, \Sigma),$$

and $\Sigma = \mathrm{diag}\{\sigma^2_i\}_{1 \leq i \leq d_\Theta}$, where $\sigma^2_i > 0$,

ii) an observed vector Y, where conditioned on $\Psi = \psi$ is a state space model detailed in lemma 8.1.1 with the parameter Θ replaced by ψ.

The parameter of the extended model is given by $\phi = (\Theta, \sigma^2) \in \mathbb{R}^{d_\Theta} \times (\mathbb{R}_+^*)^{d_\Theta}$. By Proposition 5.1.5, the problem of parameter estimation of ϕ is transformed in an iterative state estimation problem. Starting with a fixed value $\phi^{(0)}$, the EM-update equations given by (5.14) and (5.15) produce a sequence $\phi^{(n)}$ which converges to a stationary point of the initial likelihood. Unfortunately, the conditional moments involved in these equations cannot be computed explicitly for nonlinear models of this type and for this reason we need to implement a stochastic variant of the EM-algorithm to approximate them. In order to tackle this problem, we describe next a MCMC-based method, such as a one chain Adapted Metropolis-within-Gibbs algorithm or a SMC-based method, such as CPF to perform the E-step. Parametric bootstrap is used to provide the uncertainty assessment (confidence interval for the parameter vector Θ).

Convergence criterion

As a stopping rule we use either the one proposed in Booth and Hobert (1999), which claims convergence when the relative change in the estimates from three successive iterations is reasonably small, as for the Interacting MCMC, or the one using Monte Carlo error evaluation (Q-function) to build a tolerant interval detailed in Section 5.2.1.

8.6.2 Iterative AMwG Vs. Iterative RPF

At each iteration of the EM-algorithm, given the current parameter update ϕ', our objective is to approximate

$$\mathbb{E}_{\phi'}\big(\Theta \mid Y_{0:n} = y_{0:n}\big)$$

and

$$\big(\mathrm{Var}_{\phi'}(\Theta_i \mid Y_{0:n} = y_{0:n})\big)_{1 \le i \le d_{\Theta_1}},$$

where n corresponds to the observation length, since the update equations for the parameters of the model given in Proposition 5.1.5 generally lead to non-explicit solutions. Several alternative algorithms can be employed to make this approximation for this E-step, according to our former tests, the one chain AMwG and the CPF algorithms have demonstrated the best performance and as a result have been selected.

Settings :

In the following, 20 updates of the prior distribution are carried out (20 iterations of the proposed variant of EM-type algorithm) with the E-step performed by the one chain AMwG and RPF based on the same simulated restrict dataset.

Observations and remarks :

Figure 8.16 displays the evolutions of the mean estimates and the associated credibility intervals given by the two iterative algorithms. The variance estimates shrink drastically with the EM iterations in all cases. We recall that they are no longer variances of the Bayesian posterior distribution $p(\theta|y)$ inferred from the limited data y, but are supposed to converge to zero with the iteration of the EM algorithm. For a proper uncertainty assessment, it is thus necessary to perform parametric bootstrap. In a general way, the estimations resulting from the iterative RPF and the iterative

one chain AMwG are quite close. Further evaluation is conducted for a more precise comparison.

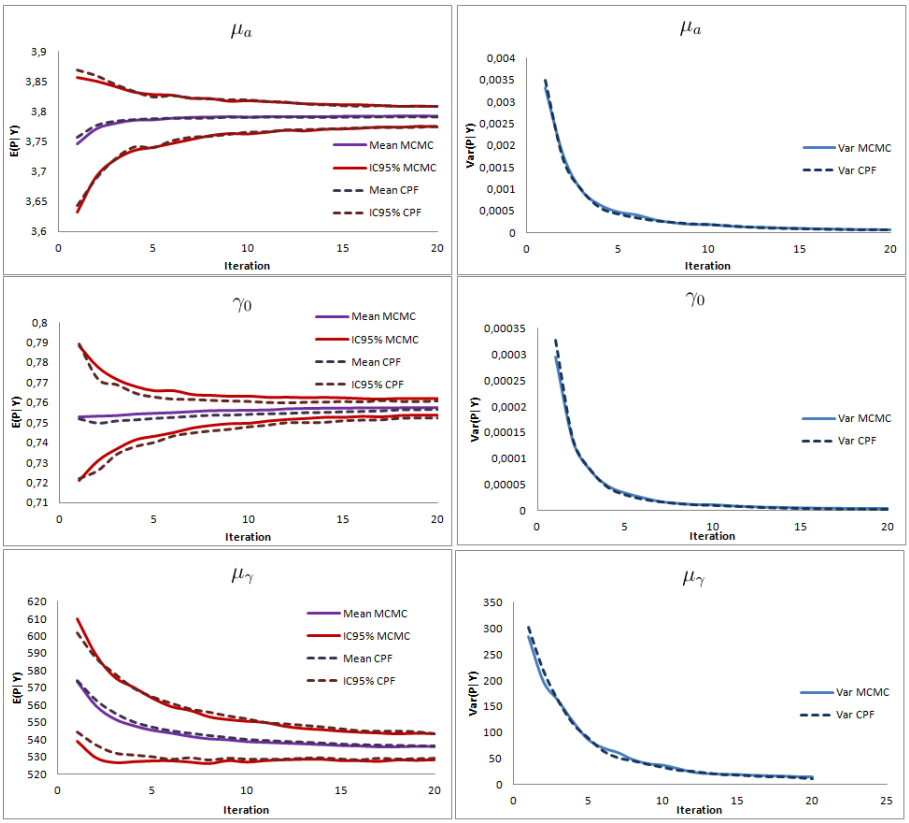


Figure 8.16: Comparison of estimations (of 3 functional parameters) based on the proposed variant of EM-type algorithm with the E-step performed by CPF and Adapted Metropolis-within-Gibbs.

8.6.3 Strategies to increase the number of particles in the Iterative-RPF algorithm

To improve the precision of our estimation, considering the fact that for the particle filtering based methods, the number of simulations (particles) should be defined before one complete iteration, different strategies are compared for the augmentation of the Monte Carlo sample size (geometric and quadratic), after each iteration of the EM algorithm.

Settings :

Based on the same restricted dataset, two strategies, the geometric and the quadratic increases of the number of particles are used with the iterative RPF. 100 independent runs are conducted. A large constant sample size is also used to have a more ideal (but more expensive) EM algorithm (100 000 particles for the restricted dataset).

Observations and remarks :

	Real value	Quadratic Increase		Geometric Increase		Constant (Max)	
		$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\mathbb{V}(\mathbb{E}(\Theta\|Y))$
μ_a	3.56	3.562	2.40e-5	3.560	9.18e-6	3.560	1.63e-6
γ_0	0.625	0.626	3.3e-5	0.625	7.21e-6	0.625	2.69e-6
μ_γ	553.9	550.85	5.33	551.64	2.18	552.04	0.34
Stop iteration* (Time)		51	(61min)	41	(38min)	24	(102min)

Table 8.23: Comparison of strategies to increase the number of particles. Both quadratic and geometric increases have a maximum of 100 000 particles, while the constant configuration has 100 000 particles for each iteration. The results are based on 100 independent runs, based on a full dataset (160 dates). The same stopping rule is used for all the tests. *: EM iterations.

Figure 8.17 illustrates the two strategies based on the same observation dataset. In Table 8.23, we compare the results of the two strategies with those that we obtained with the constant number of particles. The geometric increase provided better results compared to the quadratic increase. The increase of the number of particles takes place every time that the Monte Carlo error in the evaluation of the Q-function is considered to be significant, either by detecting a zig-zagging in parameter estimates, or by a tolerant interval (See Section 5.2.1). Our preliminary tests have showed that both criteria work with similar performance.

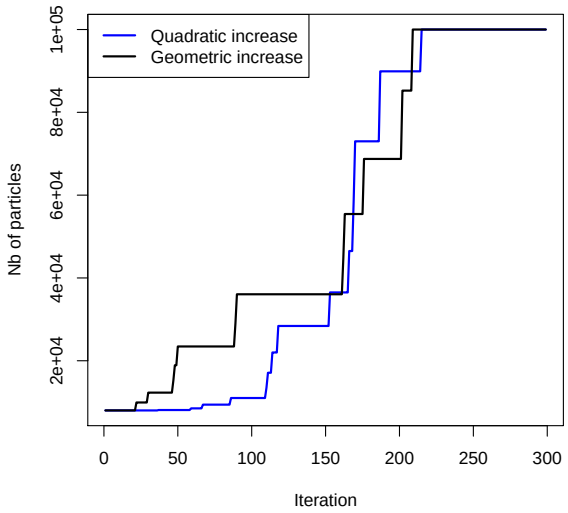


Figure 8.17: Comparison of geometric and quadratic increases of the number of particles for the IRPF algorithm.

Without a strong effort to optimize the increasing strategy, the computational time is reduced significantly. The results show that both increase techniques induce slightly higher variability in the estimates among different runs of the algorithm than in the

case with a constant number of particles. Consequently, the gain in computational time from the decrease of the total number of simulations is counterbalanced by this effect.

8.7 Conditional iterative approach for full parameter estimation

Until now, we are always under the assumption that the noise parameters of the model are known, although in practice, this is generally not the case. To meet the requirement of providing reasonable estimates for the noise parameters, some possible conditional approaches are presented in Section 6.3 with SMC approaches or with MCMC based methods. In the study case with simulated dataset, we illustrate the two approaches when the noise estimation is combined with the Iterative RPF and the Iterative AMwG algorithms. We highlight the fact that the noise estimation can naturally carried out with all the Bayesian methods presented in this thesis as well.

This section is organized as follows, we first detail the implementation of the noise parameter estimation combined with a MCMC base-method for functional parameter estimation, then another noise parameter estimation method is presented when it is combined with a SMC method which is used to estimate the functional parameters and the hidden state variables. Finally, the corresponding results are compared with a simulated dataset in both its complete and restricted form in order to evaluate the impact of the lost of information.

8.7.1 Iterative AMwG with noise updates

Implementation description
We recall that the noise parameters $\{\sigma^2_{\gamma\gamma}, \sigma^2_q, \sigma^2_g, \sigma^2_r\}$ are unknown and to be estimated, as the six functional parameters of the LNAS model: $\{\mu_a, \lambda, \mu_\gamma, \sigma_\gamma, \gamma_0, \gamma_f\}$. Regarding the priors, Inverse Gamma prior distributions are assumed for the noise parameters, and normal priors are assumed for the functional parameters:

$$
\begin{aligned}
\sigma^2_{\gamma\gamma} &\sim \mathcal{IG}(\alpha_{\sigma_{\gamma\gamma}}, \beta_{\sigma_{\gamma\gamma}}), \\
\sigma^2_q &\sim \mathcal{IG}(\alpha_{\sigma_q}, \beta_{\sigma_q}), \\
\sigma^2_g &\sim \mathcal{IG}(\alpha_{\sigma_g}, \beta_{\sigma_g}), \\
\sigma^2_r &\sim \mathcal{IG}(\alpha_{\sigma_r}, \beta_{\sigma_r}),
\end{aligned}
\qquad
\begin{aligned}
\mu_a &\sim \mathcal{N}(m_{\mu_a}, \sigma^2_{\mu_a}) I_{\mathbb{R}_+}, \\
\lambda &\sim \mathcal{N}(m_\lambda, \sigma^2_\lambda) I_{\mathbb{R}_+}, \\
\mu_\gamma &\sim \mathcal{N}(m_{\mu_\gamma}, \sigma^2_{\mu_\gamma}) I_{\mathbb{R}_+}, \\
\sigma_\gamma &\sim \mathcal{N}(m_{\sigma_\gamma}, \sigma^2_{\sigma_\gamma}) I_{\mathbb{R}_+}, \\
\gamma_0 &\sim \mathcal{N}(m_{\gamma_0}, \sigma^2_{\gamma_0}) I_{(0,1)}, \\
\gamma_f &\sim \mathcal{N}(m_{\gamma_f}, \sigma^2_{\gamma_f}) I_{(0,1)}.
\end{aligned}
$$

The target distribution (8.8) is accordingly updated:

$$
\begin{aligned}
\pi(\Theta|x_{1:n}, y_{1:n}) \propto & \prod_{t=0}^{n-1} \frac{1}{\sigma_Q \gamma(t) F_Q^t(Q_f(t);\Theta)} \\
& \cdot \exp\{-\frac{\left(Q_f(t+1) - Q_f(t) - \gamma(t) F_Q^t(Q_f(t);\Theta)\right)^2}{2\left(\sigma_Q \gamma(t) F_Q^t(Q_f(t);\Theta)\right)^2}\} \\
& \prod_{t=1}^{n} \frac{1}{\sigma_{\gamma\gamma}\Gamma^t(\Theta)} \cdot \exp\{-\frac{(\gamma(t) - \Gamma^t(\Theta))^2}{2(\sigma_{\gamma\gamma}\Gamma^t(\Theta))^2}\} \\
& \prod_{t=0}^{n-1} \delta_{Q_r(t+1),\phi(Q_f(t+1),\gamma(t),Q_f(t),Q_r(t))} \\
& \prod_{t=0}^{n-1} \frac{1}{\sigma_g F_g^{t+1}(Q_f(t+1);\Theta)} \\
& \cdot \exp\{-\frac{(F_g^{t+1}(Q_f(t+1);\Theta) - Y_g^{exp}(t+1))^2}{2(\sigma_g F_Q^{t+1}(Q_f(t+1);\Theta))^2}\} \\
& \prod_{t=0}^{n-1} \frac{1}{\sigma_r F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta)} \\
& \cdot \exp\{-\frac{(F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta) - Y_r^{exp}(t+1))^2}{2(\sigma_g F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta))^2}\} \\
& \cdot \frac{1}{\sigma_{\mu_a}\sigma_\lambda\sigma_{\sigma_\gamma}\sigma_{\gamma_f}\sigma_{\mu_\gamma}\sigma_{\gamma_0}} \cdot \left(1 - \Phi(\frac{-m_{\mu_a}}{\sigma_{\mu_a}})\right)^{-1} \left(1 - \Phi(\frac{-m_\lambda}{\sigma_\lambda})\right)^{-1} \\
& \cdot \left(1 - \Phi(\frac{-m_{\sigma_\gamma}}{\sigma_{\sigma_\gamma}})\right)^{-1} \left(1 - \Phi(\frac{-m_{\mu_\gamma}}{\sigma_{\mu_\gamma}})\right)^{-1} \\
& \cdot \left(\Phi(\frac{1 - m_{\gamma_f}}{\sigma_{\gamma_f}}) - \Phi(\frac{-m_{\gamma_f}}{\sigma_{\gamma_f}})\right)^{-1} \left(\Phi(\frac{1 - m_{\gamma_0}}{\sigma_{\gamma_0}}) - \Phi(\frac{0 - m_{\gamma_0}}{\sigma_{\gamma_0}})\right)^{-1} \\
& \cdot \exp\{-\frac{(\mu_a - m_{\mu_a})^2}{2\sigma_{\mu_a}^2} - \frac{(\lambda - m_\lambda)^2}{2\sigma_\lambda^2} - \frac{(\sigma_\gamma - m_{\sigma_\gamma})^2}{2\sigma_{\sigma_\gamma}^2} \\
& \quad - \frac{(\gamma_f - m_{\gamma_f})^2}{2\sigma_{\gamma_f}^2} - \frac{(\mu_\gamma - m_{\mu_\gamma})^2}{2\sigma_{\mu_\gamma}^2} - \frac{(\gamma_0 - m_{\gamma_0})^2}{2\sigma_{\gamma_0}^2}\} \\
& \cdot \frac{1}{(\sigma_{\gamma\gamma}^2)^{\alpha_{\sigma_{\gamma\gamma}}+1}(\sigma_q^2)^{\alpha_{\sigma_q}+1}(\sigma_g^2)^{\alpha_{\sigma_g}+1}(\sigma_r^2)^{\alpha_{\sigma_r}+1}} \\
& \cdot \exp\{-\frac{\beta_{\sigma_{\gamma\gamma}}}{\sigma_{\gamma\gamma}^2} - \frac{\beta_{\sigma_q}}{\sigma_q^2} - \frac{\beta_{\sigma_g}}{\sigma_g^2} - \frac{\beta_{\sigma_r}}{\sigma_r^2}\} \\
& \cdot I(\mu_a > 0) \cdot I(\lambda > 0) \cdot I(\sigma_\gamma > 0) \\
& \cdot I(\gamma_f \in (0,1)) \cdot I(\mu_\gamma > 0) \cdot I(\gamma_0 \in (0,1)). \qquad (8.16)
\end{aligned}
$$

Since in practice, direct simulation is difficult to achieve, here we propose an iterative approach to update the functional parameters Θ_1, the hidden state variables $x_{1:n}$ and the noise parameters Θ_2 in turns with three steps :

- Initialization

  – Choose initial values for Θ_1, Θ_2 and $x_{1:n}$.

- Iteration

  For $i = 1, \ldots, m$:
  -Step 1 Estimate Θ_1 by MH with a joint proposal $\{\Theta_1, x_{1:n}\}$ conditional to Θ_2 until convergence is claimed.

  -Step 2 Fix Θ_1 to the estimated expectation and estimate $x_{1:n}$ conditional to Θ_1 and Θ_2 by MH until convergence is claimed,

  or, estimate $x_{1:n}$ by MH with a joint proposal $\{\Theta_1, x_{1:n}\}$ (the proposal distribution for Θ_1 is fixed to the posterior distribution obtained in Step 1) until convergence is claimed.

  -Step 3 Update Θ_2 directly from the target distribution.

In the following, we detail each of the three steps.

• **Step 1**: sample $\{\Theta_1, X_{1:n}\} \sim \pi(\theta_1, x_{1:n}|y_{1:n}, \Theta_2)$ (carried out by MH).

The objective of this step is to estimate the functional parameter vector Θ_1 given $y_{1:n}$. The hidden state variables $x_{1:n}$ are introduced only in order to make the computation of the acceptance probability possible.

According to (8.12), the target distribution for the parameters can be simplified in the case of six functional parameters, conditional to the four noise parameters. To compute the acceptance ratio, we need to calculate :

$$
\begin{aligned}
p(y_{1:n}|\Theta, x_{1:n}) \cdot \pi(\Theta) \propto & \prod_{t=1}^{n} \frac{1}{\sigma_g \tilde{Q}_g(t)} \cdot \exp\left\{ -\frac{(\tilde{Q}_g(t) - Y_g^{exp}(t))^2}{2(\sigma_g \tilde{Q}_g(t))^2} \right\} \\
& \prod_{t=1}^{n} \frac{1}{\sigma_r \tilde{Q}_r(t)} \cdot \exp\left\{ -\frac{(\tilde{Q}_r(t) - Y_r^{exp}(t))^2}{2(\sigma_r \tilde{Q}_g(t))^2} \right\} \\
& \cdot \frac{1}{\sigma_{\mu_a}\sigma_\lambda\sigma_{\sigma_\gamma}\sigma_{\gamma_f}\sigma_{\mu_\gamma}\sigma_{\gamma_0}} \cdot \exp\{ -\frac{(\mu_a - m_{\mu_a})^2}{2\sigma_{\mu_a}^2} - \frac{(\lambda - m_{\mu_\lambda})^2}{2\sigma\lambda^2} \\
& - \frac{(\sigma_\gamma - m_{\sigma_\gamma})^2}{2\sigma_{\sigma_\gamma}^2} - \frac{(\gamma_f - m_{\gamma_f})^2}{2\sigma_{\gamma_f}^2} - \frac{(\mu_\gamma - m_{\mu_\gamma})^2}{2\sigma{\mu_\gamma}^2} - \frac{(\gamma_0 - m_{\gamma_0})^2}{2\sigma_{\gamma_0}^2} \} \\
& \cdot I(\mu_a > 0) \cdot I(\lambda > 0) \cdot I(\sigma_\gamma > 0) \\
& \cdot I(\gamma_f \in (0,1)) \cdot I(\mu_\gamma > 0) \cdot I(\gamma_0 \in (0,1)).
\end{aligned}
$$

• **Step 2**: sample $X_{1:n} \sim \pi(x_{1:n}|y_{1:n}, \Theta_1, \Theta_2)$ (carried out by MH).

In the previous step, the candidate of the hidden state variables $x_{1:n}$ is simulated based on the candidate of Θ_1 only in the objective to compute the acceptance ratio. For the sake of computational efficiency, it is simulated only once which clearly does not optimize the trajectory given $y_{1:n}$. Given the fact that the hidden state estimates are required to update the noise parameters, in this step, we aim to improve the hidden states estimates given the posterior distribution of Θ_1 provided by step 1 when it has converged. Thus a nested new Markov chain is built for this purpose.

- If Θ_1 is fixed to the mean estimate, then according to Equation (4.12), the computation of acceptance ratio requires :
For $t = 1, \ldots, n$:

$$
\begin{aligned}
& p(X(t+1)|X(t),\Theta)p(Y(t)|X(t),\Theta) \\
\propto\ & \frac{1}{\sigma_Q\gamma(t)F_Q^t(Q_f(t);\Theta)} \cdot \exp\{-\frac{\left(Q_f(t+1)-Q_f(t)-\gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}{2\left(\sigma_Q\gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}\} \\
& \cdot \frac{1}{\sigma_{\gamma\gamma}\Gamma^{t+1}(\Theta)} \cdot \exp\{-\frac{(\gamma(t+1)-\Gamma^{t+1}(\Theta))^2}{2(\sigma_{\gamma\gamma}\Gamma^{t+1}(\Theta))^2}\} \\
& \cdot \delta_{Q_r(t+1),\phi(Q_f(t+1),\gamma(t),Q_f(t),Q_r(t))} \\
& \cdot \frac{1}{\sigma_g F_g^t(Q_f(t);\Theta)} \cdot \exp\{-\frac{(F_g^t(Q_f(t);\Theta)-Y_g^{exp}(t))^2}{2(\sigma_g F_Q^t(Q_f(t);\Theta))^2}\} \\
& \cdot \frac{1}{\sigma_r F_r^{t-1}(Q_r(t-1),Q_f(t),\gamma(t-1);\Theta)} \\
& \cdot \exp\{-\frac{(F_r^{t-1}(Q_r(t-1),Q_f(t),\gamma(t-1);\Theta)-Y_r^{exp}(t))^2}{2(\sigma_g F_r^{t-1}(Q_r(t-1),Q_f(t),\gamma(t-1);\Theta))^2}\}. \qquad (8.17)
\end{aligned}
$$

- If Θ_1 is drawn from the posterior distribution obtained in step 1, then according to (4.13), the same result as (8.17) can be obtained to compute the acceptance ratio. Note that in this case, an independence sampler is used for Θ_1 instead of a random walk proposal as the case in Step 1.

N.B.

This step can be regarded as building another Markov chain dedicated only to the estimation of $x_{1:n}$, in which the noise parameters are considered fixed, as a result, the samples are not used to construct the posterior distribution of Θ_1.

• **Step 3**: update $\Theta_2 \sim \pi(\theta_2|x_{1:n}, y_{1:n}, \Theta_1)$.

Once the estimations of the hidden states have converged, the noise parameters are updated. Conjugate priors are chosen for the noise parameters Θ_2 with the aim of realizing an explicit update by simulating from the corresponding full conditional distributions (8.16):

$$\sigma_q^2 \sim \mathcal{IG}\left(\alpha_{\sigma_q} + \tfrac{n}{2}, \quad \beta_{\sigma_q} + \frac{1}{2}\sum_{t=0}^{n-1} \frac{\left(Q_f(t+1) - Q_f(t) - \gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}{\left(\gamma(t)F_Q^t(Q_f(t);\Theta)\right)^2}\right),$$

$$\sigma_{\gamma\gamma}^2 \sim \mathcal{IG}\left(\alpha_{\sigma_{\gamma\gamma}} + \tfrac{n}{2}, \quad \beta_{\sigma_{\gamma\gamma}} + \frac{1}{2}\sum_{t=0}^{n-1} \frac{\left(\gamma(t) - \Gamma^t(\Theta)\right)^2}{\Gamma^t(\Theta)^2}\right),$$

$$\sigma_g^2 \sim \mathcal{IG}\left(\alpha_{\sigma_g} + \tfrac{n}{2}, \quad \beta_{\sigma_g} + \frac{1}{2}\sum_{t=0}^{n-1} \frac{\left(F_g^{t+1}(Q_f(t+1);\Theta) - Y_g^{exp}(t+1)\right)^2}{\left(F_g^{t+1}(Q_f(t+1);\Theta)\right)^2}\right),$$

$$\sigma_r^2 \sim \mathcal{IG}\left(\alpha_{\sigma_r} + \tfrac{n}{2}, \quad \beta_{\sigma_r} + \frac{1}{2}\sum_{t=0}^{n-1} \frac{\left(F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta) - Y_r^{exp}(t+1)\right)^2}{\left(F_r^t(Q_r(t), Q_f(t+1), \gamma(t);\Theta)\right)^2}\right).$$

Note that this three steps should be iterated, until the convergence of both Θ_1 and Θ_2. Similar stopping rules used in the implementation of MCMC without noise parameter update can be used, the details of which can be found in Section 4.1.5.

Synthetic study case

The estimation framework presented is applied to a simulated dataset in both its complete and restricted forms. The objective is to evaluate the robustness of this approach when applied to the LNAS model.

Settings :

In this tests, two different initializations are carried out in each case and the averaging technique (Cappé et al. (2005)) is used to smooth parameter estimates after a small burn-in period. One simulated dataset is used in both its complete form and restricted form. A maximum of 350000 iterations are defined for each iteration (E-step) with the restrict dataset and 150 000 iterations respectively for complete dataset. The standard errors of the variability of parameter estimates from independent runs of the algorithm (20 repetitions based on the same dataset) are also given, which corresponds to the algorithmic uncertainty. The log-likelihood values of the estimates are estimated by the means of 10 independent evaluations (the standard errors were small, approximately 0.005).

Observations and remarks :

The parameters' real values and the estimation results are presented in Table 8.24. As expected, the one chain AMwG based EM variant does not cope well with the complete dataset due to the variance degeneracy, as illustrated by Table 8.24 for both initializations. The log-likelihood evaluations indicate that the estimates are quite poor. The important variance between the estimates implies that the method may suffer from the updated prior distribution during the initialization of each EM iteration, which becomes narrower and narrower and can exhibit important influence in the computation for the acceptance ratio and constrain the evolution of the chain. However, when confronting the restricted dataset, the method does provide generally better results. Regarding the difference between the estimates resulted from the two

different priors, in the case of restricted dataset it is more remarkable.

This result suggests that the one chain AMwG may not be suitable to perform the E-step of the EM variant algorithm with complete datasets. In the following, the iterative RPF/CPF algorithm is performed with the same configuration of the test.

Initialization 1							
Parameter	Real value	Initial distribution		Restricted dataset ♠		Full dataset ♣	
		μ_0	σ_0	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$
μ_a	3.56	3.40	0.15	3.650	0.052	3.567	0.072
λ	56.6	60.0	5.0	59.71	0.86	55.83	2.34
γ_0	0.625	0.58	0.10	0.676	0.042	0.623	0.011
γ_f	0.1035	0.12	0.025	0.081	0.028	0.050	0.027
μ_γ	550.0	500.0	50.00	495.41	8.36	628.24	23.73
σ_γ	950.0	880.0	50.00	1373.29	35.71	1682.52	47.53
σ_g	0.1	$\sigma_g^2 \sim \mathcal{IG}(25,2)$		0.109	0.007	0.090	0.002
σ_r	0.1	$\sigma_r^2 \sim \mathcal{IG}(25,2)$		0.112	0.008	0.104	0.003
σ_q	0.05	$\sigma_q^2 \sim \mathcal{IG}(55,2)$		0.039	0.003	0.048	0.002
$\sigma_{\gamma\gamma}$	0.05	$\sigma_{\gamma\gamma}^2 \sim \mathcal{IG}(55,2)$		0.048	0.003	0.052	0.002
$-2\mathcal{L}(\bar{\Theta})$	273.75 ♠ 1870.70 ♣	853.74 ♠ 34624.20 ♣		272.76	0.70	1907.50	7.20
Initialization 2							
Parameter	Real value	Initial distribution		Restricted dataset ♠		Full dataset ♣	
		μ_0	σ_0	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$
μ_a	3.56	3.60	0.15	3.670	0.067	3.568	0.065
λ	56.6	52.0	5.0	59.87	0.98	55.70	1.78
γ_0	0.625	0.7	0.10	0.684	0.045	0.624	0.009
γ_f	0.1035	0.09	0.025	0.072	0.031	0.053	0.028
μ_γ	550.0	580.0	50.00	511.12	12.53	627.47	19.66
σ_γ	950.0	1000.0	50.00	1589.75	46.91	1790.25	46.91
σ_g	0.1	$\sigma_g^2 \sim \mathcal{IG}(25,2)$		0.097	0.007	0.092	0.002
σ_r	0.1	$\sigma_r^2 \sim \mathcal{IG}(25,2)$		0.103	0.008	0.102	0.003
σ_q	0.05	$\sigma_q^2 \sim \mathcal{IG}(55,2)$		0.047	0.003	0.048	0.002
$\sigma_{\gamma\gamma}$	0.05	$\sigma_{\gamma\gamma}^2 \sim \mathcal{IG}(55,2)$		0.050	0.002	0.051	0.003
$-2\mathcal{L}(\bar{\Theta})$	273.75 ♠ 1870.70 ♣	776.93 ♠ 8212.03 ♣		272.826	0.79	1916.42	8.59

Table 8.24: Comparison of the estimations resulting from two different initializations with one chain AMwG based EM variant, both for the restricted (maximum of 350 000 iterations for each E-step) and the complete dataset (maximum of 150 000 iterations for each E-step). ♠: estimated log-likelihood based on the restricted dataset (14 dates); ♣: estimated log-likelihood based on the complete dataset (160 dates).

8.7.2 Iterative RPF with noise updates

For the filtering approaches, the noise estimation can be carried out only at the end of the algorithm. Hence, we have to update the noise parameters Θ_2 and the functional

parameters Θ_1 in turns with the proposed estimators. Nevertheless, we highlight that this noise estimation approach can also be combined with other Bayesian methods, such as the MCMC-based methods.

In the following, we propose four estimators for both modelling noise parameters and observation noise parameters which are built based on the estimation of the augmented hidden states vector.

Implementation descriptions

In the application of LNAS model, we recall that the modelling noises η_Q appear in the biomass production process (2.1):

$$Q(t) = F_Q^t(Q_f(t); \Theta) \cdot (1 + \eta_Q(t)),$$

and η_γ in the biomass allocation process (2.4):

$$\begin{aligned}\gamma(t) &= \Gamma^t(\Theta) \cdot (1 + \eta_\gamma(t)) \\ &= (\gamma_0 + (\gamma_f - \gamma_0) \cdot G_a(\tau(t))) \cdot (1 + \eta_\gamma(t)).\end{aligned}$$

Since

$$\begin{aligned}Q_r(t+1) &= F_r^t(Q_r(t), Q_f(t+1), \gamma(t); \Theta), \\ &= Q_r(t) + (1 - \gamma(t)) \cdot Q(t) \\ &= Q_r(t) + \frac{1 - \gamma(t)}{\gamma(t)}(Q_f(t+1) - Q_f(t)), \\ Q_f(t+1) &= Q_f(t) + \gamma(t) \cdot Q(t).\end{aligned}$$

and

$$Q_g(t) = F_g^t(Q_f(t); \Theta) = Q_f(t) - Q_s(t),$$

the observation vector depends only on $Q_f(t)$ and $Q_r(t)$, which is why they are chosen as hidden state variables to estimate. We emphasize again that the hidden state variables and the functional parameters Θ_1 can be estimated by either the Bayesian approaches presented in Chapter 4 or the iterative approaches presented in Chapter 5, based on the observations $y_{0:n}$. In our first application, this approach is combined with the iterative filtering approach IRPF.

The obtained estimates $\hat{Q}_r^{0:n}$ and $\hat{Q}_f^{0:n}$ can be used to deduce $\hat{\gamma}^{0:n}$ and $\hat{Q}^{0:n}$:

$$\hat{\gamma}(t) = \frac{\hat{Q}_f(t+1) - \hat{Q}_f(t)}{\hat{Q}_r(t+1) - \hat{Q}_r(t) + \hat{Q}_f(t+1) - \hat{Q}_f(t)},$$

$$\hat{Q}(t) = \hat{Q}_r(t+1) - \hat{Q}_r(t) + \hat{Q}_f(t+1) - \hat{Q}_f(t).$$

Thus, by using the estimated functional parameter vector $\hat{\Theta}_1$, the estimators of the standard error of the modelling noise η_Q denoted σ_q, and the standard error of the modelling noise η_γ denoted $\sigma_{\gamma\gamma}$, can be defined as follows :

Definition 8.7.1 (Estimator of modelling noise parameter)

$$\hat{\sigma}_q = \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}\left(\frac{\hat{Q}(t) - F_Q^t(\hat{Q}_f(t);\hat{\Theta}_1)}{F_Q^t(\hat{Q}_f(t);\hat{\Theta}_1)}\right)^2},$$

$$\hat{\sigma}_{\gamma\gamma} = \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}\left(\frac{\hat{\gamma}(t) - \Gamma^t(\hat{\Theta}_1)}{\Gamma^t(\hat{\Theta}_1)}\right)^2}.$$

In the same way, the estimator of the standard errors for the observation noises ϵ_g and ϵ_r can be derived:

Definition 8.7.2 (Estimator of observation noise parameter)

$$\hat{\sigma}_g = \sqrt{\frac{1}{N_{obs}-1}\sum_{n=1}^{N_{obs}}\left(\frac{Y_{t_n}^g - F_g^{t_n}(\hat{Q}_f(t_n);\Theta_1)}{F_g^{t_n}(\hat{Q}_f(t_n);\Theta_1)}\right)^2},$$

$$\hat{\sigma}_r = \sqrt{\frac{1}{N_{obs}-1}\sum_{n=1}^{N_{obs}}\left(\frac{Y_{t_n}^r - \hat{Q}_r(t_n)}{\hat{Q}_r(t_n)}\right)^2}.$$

with N_{obs} the number of available observation data.

The conditional estimation process begins with the estimation of Θ_1 given Θ_2, then Θ_2 is estimated empirically based on the estimates of the hidden states. The estimation then proceeds with the new value of Θ_2 and iterates. The convergence of both Θ_1 and Θ_2 are claimed by a standard stopping rule based on the relative changes in the estimations during three successive estimations (See Section 8.6.1).

Synthetic study case

For the conditional IRPF approach, the following tests are carried out to evaluate and to improve its efficiency when applied to the LNAS model with simulated dataset.

List of the tests for the conditional IRPF approaches :

I	Influence of priors and of missing data
II	Uncertainty assessment with parametric bootstrap
III	Strategy to increase the number of particles

I Influence of priors and of missing data

Settings :

In the following tests, two different initializations are tested with both the full dataset and the restricted dataset derived from the full dataset. The averaging technique (Cappé et al. (2005)) is used to smooth parameter estimates after a small burn-in period. One simulated dataset is used in both its complete form and restricted form. A maximum of 350000 iterations are defined for each iteration (E-step) with the restrict dataset and 150 000 iterations respectively for complete dataset. The standard errors of the variability of parameter estimates from independent runs of the algorithm (20 repetitions based on the same dataset) are also given, which corresponds to the algorithmic uncertainty. The log-likelihood values of the estimates are estimated by the means of 10 independent evaluations (the standard errors were small, approximately 0.005).

Note that usually, five estimations in turns of the functional and noise parameters are performed.

Observations and remarks :

The iterative RPF/CPF provides better estimates in terms of Log-likelihood evaluation compared to the former results given by the MCMC based approach, as demonstrated by Table 8.25, for both the restricted dataset and the complete dataset. It is thus more suitable to perform the E-step for this EM variant algorithm.

As in the previous test case, estimations are carried out based on two initial distributions. We remark again that for the complete dataset the estimates based on these two different initializations are very close, which suggests a negligible effect of the initial distributions. However, for the restricted dataset, there is a difference in the estimates of the last three parameters (γ_f, μ_γ, σ_γ) and the effect of the initialization seems to be non negligible, especially in the case of σ_γ. This can be explained by the sensitivity of model outputs Q^g and Q^r to parameters. A sensitivity analysis of the LNAS model based on Sobol's method is presented in chapter 9 and the results show that the last three parameters are the least influential ones among the six which are estimated. The increased variability of these parameters (see the corresponding standard deviations) from independent runs of the algorithm (20 repetitions) based on the same dataset is also an argument in this direction, since the great loss of information for the restricted dataset seems to affect their estimation quality.

The results of log-likelihood evaluations also seem to suggest that despite that the estimated MLE could slightly differ from two different initializations, the most sensitive functional parameters can be estimated without difficulty even in the case of the restricted dataset, together with the observation and model noise parameters. It may as well be confirmed by the results given by the AMwG based algorithm 8.24.

By comparing the estimation results (with the same initialization) between the restricted and the complete dataset, we also conclude that the estimates of the most influential parameters are closer to the real values with the complete dataset than the restricted one, as expected, due to the great amount of missing information.

Initialization 1

Parameter	Real value	Initial distribution		Restricted dataset ♠		Full dataset ♣	
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$
μ_a	3.56	3.40	0.15	3.664	0.049	3.564	0.005
λ	56.6	60.0	5.0	59.73	0.22	55.68	0.12
γ_0	0.625	0.58	0.10	0.679	0.026	0.628	0.002
γ_f	0.1035	0.12	0.025	0.088	0.013	0.047	0.006
μ_γ	550.0	500.0	50.00	492.91	2.44	624.55	9.07
σ_γ	950.0	880.0	50.00	1338.75	20.25	1633.29	98.66
σ_g	0.1	0.08	-	0.083	0.001	0.091	0.001
σ_r	0.1	0.08	-	0.106	0.001	0.108	0.001
σ_q	0.05	0.035	-	0.047	0.006	0.037	0.005
$\sigma_{\gamma\gamma}$	0.05	0.035	-	0.059	0.010	0.061	0.005
$-2\mathcal{L}(\bar{\Theta})$	273.75 ♠ 1870.70 ♣	853.74 ♠ 34624.20 ♣	-	271.12	0.36	1858.81	3.25

Initialization 2

Parameter	Real value	Initial distribution		Restricted dataset ♠		Full dataset ♣	
		μ_0	σ_0^2	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$
μ_a	3.56	3.60	0.15	3.668	0.058	3.566	0.003
λ	56.6	52.0	5.0	59.94	0.50	55.73	0.07
γ_0	0.625	0.7	0.10	0.686	0.036	0.628	0.002
γ_f	0.1035	0.09	0.025	0.070	0.024	0.048	0.005
μ_γ	550.0	580.0	50.00	510.24	4.60	623.14	6.69
σ_γ	950.0	1000.0	50.00	1610.93	46.88	1609.01	82.70
σ_g	0.1	0.08	-	0.106	0.002	0.091	0.001
σ_r	0.1	0.08	-	0.083	0.001	0.108	0.001
σ_q	0.05	0.035	-	0.048	0.008	0.040	0.004
$\sigma_{\gamma\gamma}$	0.05	0.035	-	0.055	0.012	0.060	0.004
$-2\mathcal{L}(\bar{\Theta})$	273.75 ♠ 1870.70 ♣	776.93 ♠ 82812.03 ♣	-	271.07	0.654	1858.52	2.54

Table 8.25: Comparison of the estimations from two different initializations with iterative RPF/CPF (EM variant), both for the restricted (maximum of 350 000 iterations for each E-step) and the complete dataset (maximum of 150 000 iterations for each E-step). ♠: estimated log-likelihood based on the restricted dataset (14 dates); ♣: estimated log-likelihood based on the complete dataset (160 dates).

II Uncertainty assessment with parametric bootstrap

The uncertainty related to parameter estimation is evaluated by parametric bootstrap. As a better approximation of the MLE, we considered the means from the estimates of both initializations. In Table 8.26 we present the results that we obtained with a bootstrap sample of size 100. Except for the least influential parameters, the bootstrap means are very close to the MLE in the case of the complete dataset, but when a great amount of data is missing a small bias is introduced. By comparing the

standard deviations of the estimates for the complete and the restricted dataset, it is clear that the additional amount of information reduces considerably the standard deviations of all the estimated parameters, except for the 3 least influential functional parameters. Moreover, the standard deviations of γ_f and σ_γ are prohibitively high to allow for reliable estimations. However, for all the other parameters, despite the increased standard deviations in the case of the restricted dataset, we finally get reliable estimates of the true values within the range of one standard deviation.

Parameter	Real value	Restricted dataset ♠			Full dataset ♣		
		MLE	Mean	Std.	MLE	Mean	Std.
μ_a	3.560	3.666	3.615	0.137	3.565	3.567	0.058
λ	56.6	59.8	57.7	4.6	55.7	55.7	1.2
γ_0	0.625	0.680	0.660	0.057	0.628	0.626	0.017
γ_f	0.1035	0.079	0.097	0.033	0.048	0.040	0.030
μ_γ	550.00	501.58	503.60	77.95	623.85	633.29	53.62
σ_γ	950.00	1474.84	1292.37	314.81	1621.15	1852.70	308.58
σ_g	0.1	0.106	0.101	0.016	0.108	0.106	0.005
σ_r	0.1	0.083	0.079	0.021	0.091	0.092	0.006
σ_q	0.05	0.047	0.049	0.012	0.038	0.038	0.004
$\sigma_{\gamma\gamma}$	0.05	0.057	0.054	0.016	0.061	0.052	0.006

Table 8.26: Uncertainty assessment in the case of both restricted and complete simulated datasets by parametric bootstrap. The means and the standard deviations are given on the basis of a 100-bootstrap sample for the restricted (14 dates) and the complete (160 dates) datasets.

III Strategy to increase the number of particles

Settings :

Like in the previous test case without the noise parameter (Θ_2) estimation, we compare the results of the geometric increase (from 60 000 to 150 000 particles) with those that we obtained with the constant number of particles (150 000 particles) under the same settings as in the previous test case.

Observations and remarks :

In Table 8.27, the results confirm that the geometric increase induces higher variability in the estimates among different runs than in the case with a constant particle number, we also remark that the computational time was reduced by 23%, thus the gain in computational time from the decrease of the total number of simulations is counterbalanced by this effect. Further investigations can be conducted in this direction to seek for the most efficient use of Monte Carlo resources.

In Figure 8.18 and Figure 8.19, we illustrate the convergence of the estimations of the six functional parameters and four noise parameters respectively for the restricted dataset based on a single run of the algorithm. The modelling noise parameters appear to converge to a flatter distribution than the observation noise parameters, especially for $\sigma_{\gamma\gamma}$.

Parameter	Real value	Initial distribution		Constant →		Geometric increase ↑	
		Mean	Std.	Mean estimate	Std.	Mean estimate	Std.
μ_a	3.56	3.40	0.15	3.664	0.049	3.662	0.082
λ	56.6	60.0	5.0	59.73	0.22	59.67	0.54
γ_0	0.625	0.58	0.10	0.679	0.026	0.677	0.063
γ_f	0.1035	0.12	0.025	0.088	0.001	0.089	0.001
μ_γ	550.0	500.0	50.00	492.91	2.44	495.27	7.71
σ_γ	950.0	880.0	50.00	1338.75	20.25	1372.40	32.92
σ_g	0.1	0.08	-	0.106	0.001	0.107	0.002
σ_r	0.1	0.08	-	0.083	0.001	0.085	0.001
σ_q	0.05	0.035	-	0.047	0.006	0.051	0.008
$\sigma_{\gamma\gamma}$	0.05	0.035	-	0.059	0.010	0.061	0.013
$-2\mathcal{L}(\bar{\Theta})$	273.75	853.74	-	271.02	0.36	271.29	0.34

Table 8.27: Comparison of parameter estimates with two different configurations of particle numbers based on 200 independent runs. →: a constant number of 150 000 particles. ↑: starting from 60 000, the number of particles increases geometrically until a maximum number of 150 000.

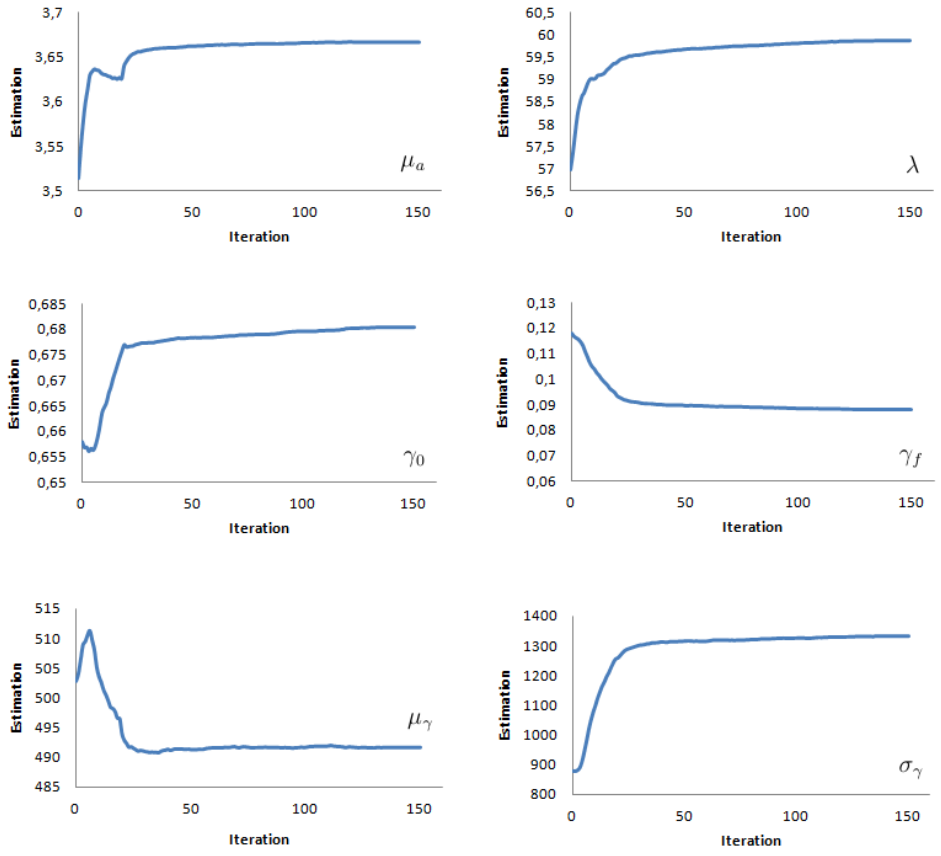
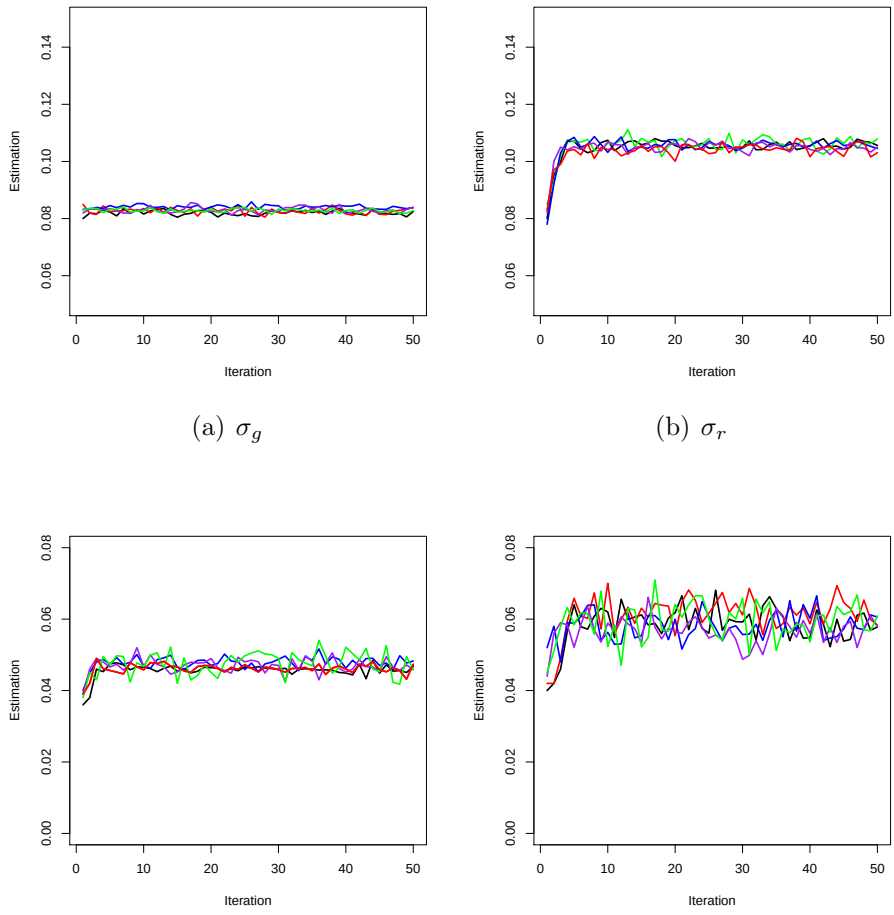


Figure 8.18: Evolution of the averaged estimates of the six functional parameters: μ_a, λ, γ_0, γ_f, μ_γ and σ_γ with the iterative RPF algorithm (35 0000 particles) for the restricted dataset.



(a) σ_g (b) σ_r

(c) σ_q (d) $\sigma_{\gamma\gamma}$

Figure 8.19: Evolution of the estimated values of the four noise parameters: μ_a, λ, γ_0, γ_f, μ_γ and σ_γ with the RPF-EM algorithm (35 0000 particles) for the restricted dataset.

8.8 Estimation with real experimental data

In this section, we present the identification results from a real dataset with the LNAS model, where parameter estimation is performed either in the Bayesian framework with MCMC and SMC based methods, or in the frequentist framework with the iterative version of the filtering methods and parametric bootstrap for uncertainty assessment.

8.8.1 Data description

This dataset concerns limited experimental observations realized by the French institute for Sugar Beet research in 2010 (details of the experimental protocols are

presented in Baey and Cournède (2011)). Dry matter of root (denoted by Q_r) and leaves (denoted by Q_g) were collected on 50 plants at 14 different dates. The observation vector, which records at each available date the average mass per square meter, is given by Table 8.28.

n	54	68	76	83	90	98	104	110	118	125	132	139	145	160
$Y_{n,1}$	85.2	372.9	447.6	440.8	620.4	523.8	541.4	620.2	627.5	757.6	760.5	598.3	670.7	628.4
$Y_{n,2}$	23.1	199.8	302.4	409.2	709.2	768.1	863.9	1232.5	1498.8	1770.2	1878.2	1913.7	2118.4	2274.7

Table 8.28: Experimental dataset provided by the French institute for Sugar beet research (ITB) based on an experiment in 2010. 14 dates of measurement are available (denoted by n). The observation vector contains the averaged mass (from 50 plants) per square meter of green leaf compartment (denoted by $Y_{n,1}$) and root compartment (denoted by $Y_{n,2}$) in g.

8.8.2 Bayesian approaches

Based on the 2010 experimental dataset, we first aim to carry out the Bayesian based methods for functional parameter estimation. All the Bayesian methods studied previously are applied. The objective is to evaluate and to compare their performance with real experimental data.

List of the tests for the conditional IRPF approaches :

I	Preliminary test (with six parameters)
II	Model selection
III	Method comparison

I Preliminary test

Settings :

According to sensitivity analysis (results of which can be found in Section 9.2.2), those functional parameters $\{\mu_a, \lambda, \gamma_0, \gamma_f, \mu_\gamma, \sigma_\gamma\}$ which has a total order index over 0.02 are chosen to be estimated. Since the noise parameters are not included in the sensitivity analysis, all the four of them are estimated. Three Bayesian based approaches, AMwG, RPF and EnKF are first applied.

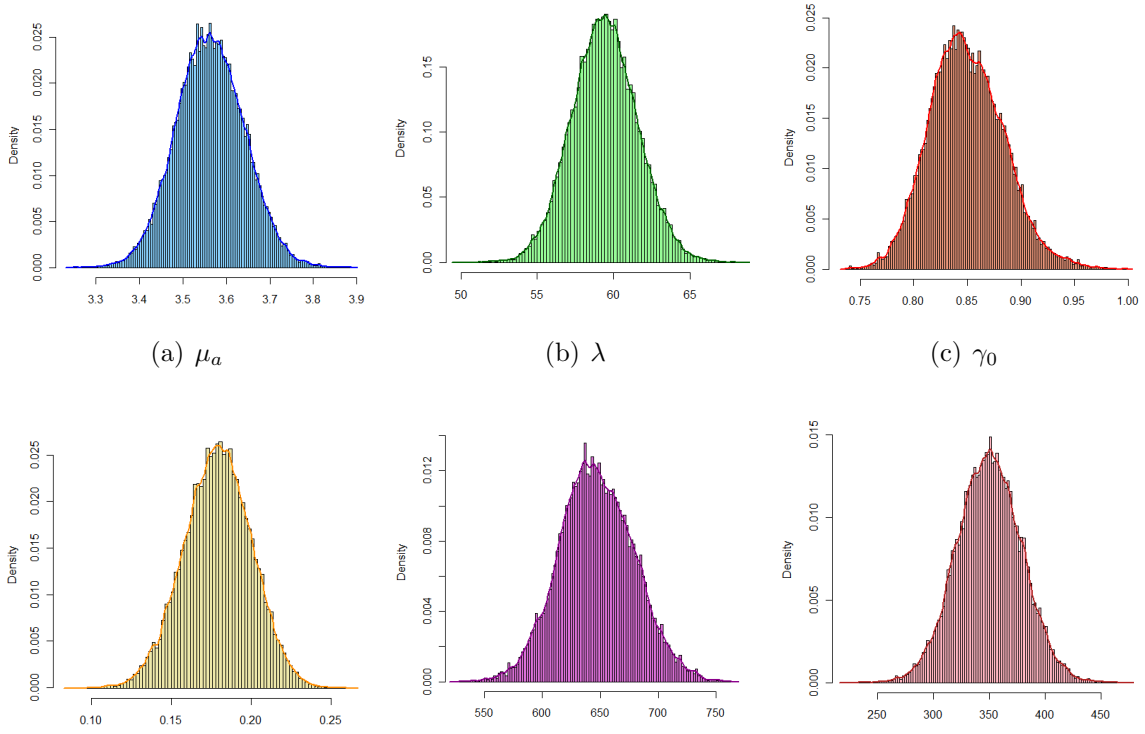
Observations and remarks :

According to the results given by Table 8.29, for the functional parameters, RPF and AMwG provided similar estimates while for noise parameters, especially $\sigma_{\gamma\gamma}$, the three methods gave different opinions. Moreover, we observe the slow mixing behaviour for AMwG, since the acceptance rate is very low (≈ 0.08). This is probably due to a strong correlation among the parameters, thus, they evolve very slowly and showed an important dependence on the prior distribution. Similar problem occurred for the filtering methods as well, the noise parameter update of $\sigma_{\gamma\gamma}$ also appears to be

unstable with important estimated variance, which implies identifiability problem. The estimations given by the three methods are quite different, especially regarding the variance estimates.

	Prior		AMwG		RPF		EnKF	
	μ_0	σ_0^2	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$
μ_a	3.40	0.04	3.556	0.006	3.551	0.005	3.507	0.013
λ	58.0	9.0	59.29	4.51	58.95	4.90	56.99	6.94
γ_0	0.90	0.01	0.8496	0.00116	0.8452	0.00108	0.8106	0.00161
γ_f	0.18	0.0025	0.178	0.00042	0.177	0.00038	0.211	0.00054
μ_γ	650.0	1600.0	647.42	921.23	650.1	946.91	644.99	1184.28
σ_γ	350.0	900.0	350.30	843.22	351.68	898.55	317.07	889.90
σ_g	$\sigma_g^2 \sim \mathcal{IG}(25,2)$		0.139	9.2e-6	0.140	1.3e-5	0.157	1.4e-5
σ_r	$\sigma_r^2 \sim \mathcal{IG}(25,2)$		0.163	9.5e-6	0.165	2.4e-5	0.166	3.1e-5
σ_q	$\sigma_q^2 \sim \mathcal{IG}(55,2)$		0.040	2.2e-7	0.046	2.5e-5	0.053	2.4e-5
$\sigma_{\gamma\gamma}$	$\sigma_{\gamma\gamma}^2 \sim \mathcal{IG}(55,2)$		0.038	2.4e-7	0.061	1.6e-4	0.071	2.5e-4
DIC	336.21		331.32		331.39		332.04	

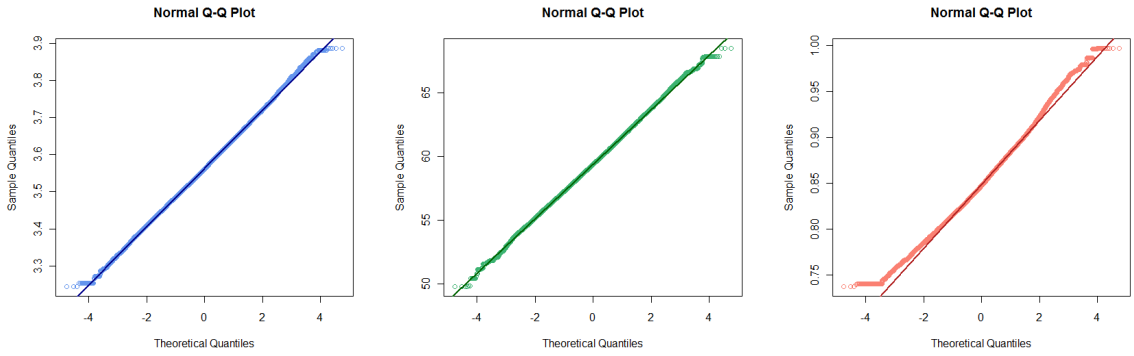
Table 8.29: Comparison of the estimations resulted from the one chain AMwG and RPF in the case of the 2010 experimental dataset.



(a) μ_a (b) λ (c) γ_0

(d) γ_f (e) μ_γ (f) σ_γ

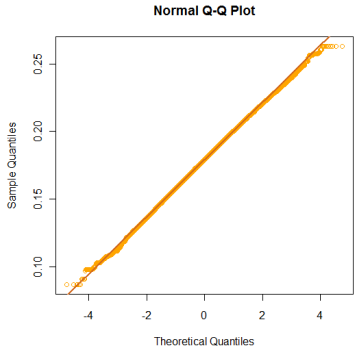
Figure 8.20: Histogram of the posterior distributions of the six functional parameters: μ_a, λ, γ_0, γ_f, μ_γ and σ_γ in the LNAS model estimated by one chain AMwG (5000 000 iterations) with noise estimation based on the 2010 experimental dataset.



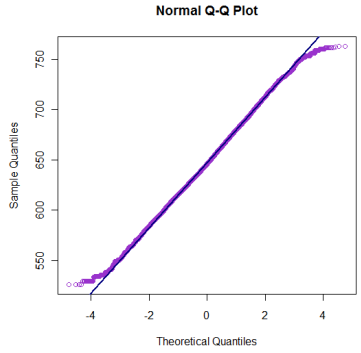
(a) μ_a

(b) λ

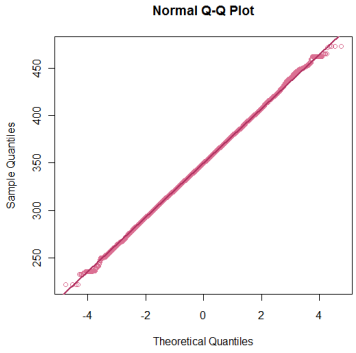
(c) γ_0



(d) γ_f



(e) μ_γ



(f) σ_γ

Figure 8.21: Normality test of the posterior distributions of the six functional parameters: μ_a, λ, γ_0, γ_f, μ_γ and σ_γ with QQplot in the LNAS model estimated by one chain AMwG (5000 000 iterations) with noise estimation based on the 2010 experimental dataset.

Figure 8.21 provides the normality test result of all the six posterior distributions, which are illustrated in Figure 8.20. All the posteriors appear to be Gaussian, except γ_0. Considering that a large number of samples (5000 000 iterations) are included, this result is not surprising. However, it also suggests that no multimodality was detected and affirms that standard error is representative and capable of characterizing the variability within the normal-like posterior distribution.

Given the fact that these methods do not cope well with this parameter, a Bayesian model selection is carried out aiming at the most appropriate configuration with the noise level which provides the best fitting.

II Model selection

In a general way, the estimation of the modelling noise parameters seem to be less stable than those of the observation noise parameters, and therefore may be evaluated via model selection methods and fixed in the following analysis.

Settings :

As presented in Chapter 6, a grid of 10 modelling noise levels is formed for the parameter $\sigma_{\gamma\gamma}$ to determine its optimum choice according to the Deviance Information Criterion.

Different configurations of the number of functional parameters are also evaluated. The order of the reduction of the parameter vector is give by the sensitivity analysis, detailed in Section 6.1. The values of those parameters that are not estimated are fixed according to their recommended values. Considering the Bayesian model selection framework, the most suitable selection criterion is DIC. The RPF approach is used to calibrate the model under different configuration, since it is more efficient than one chain AMwG (as shown in the computational comparison in Section 8.5.4), and provides similar result according to the previous tests.

d	1	2	3	4	5	6
$\sigma_{\gamma\gamma}$			DIC			
0.00	329.92	326.01	322.10	324.03	326.01	332.40
0.01	329.10	325.88	322.15	323.89	325.92	331.98
0.02	329.05	325.49	321.98	323.92	325.93	332.09
0.03	328.85	325.53	322.02	324.11	325.71	331.94
0.04	327.97	325.48	321.95	323.86	325.66	331.81
0.05	327.95	325.77	321.80	324.08	325.41	331.42
0.06	328.14	325.61	321.93	323.79	325.44	331.39
0.07	328.08	326.01	321.89	323.82	325.60	331.54
0.08	328.17	325.82	322.27	323.94	325.69	332.01
0.09	329.52	326.21	323.41	323.93	325.82	332.12
0.10	329.29	327.58	324.69	324.06	326.07	332.04

Table 8.30: Evaluation of DIC with increasing number of functional parameters to estimate in the LNAS model based on the 2010 experimental dataset.

d	1		2		**3**		4		5		6	
	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$
γ_0	0.9699	0.0130	0.9716	0.0146	0.8607	0.0340	0.8420	0.0316	0.8346	0.0304	0.8486	0.0329
μ_a	3.500	-	3.393	0.035	3.554	0.078	3.565	0.076	3.559	0.070	3.558	0.073
μ_γ	550.00	-	550.00	-	659.41	28.17	655.56	29.27	643.62	30.23	649.20	31.77
λ	60.00	-	60.00	-	60.00	-	59.13	2.12	59.03	2.02	59.54	2.08
γ_f	0.150	-	0.150	-	0.150	-	0.150	-	0.196	0.018	0.177	0.020
σ_γ	300.00	-	300.00	-	300.00	-	300.00	-	300.00	-	350.52	29.69
σ_g	0.161	-	0.158	-	0.152	-	0.148	-	0.150	-	0.145	-
σ_r	0.202	-	0.197	-	0.171	-	0.166	-	0.165	-	0.160	-
σ_q	0.081	-	0.076	-	0.060	-	0.062	-	0.058	-	0.048	-
$\sigma_{\gamma\gamma}$	0.050	-	0.050	-	0.050	-	0.050	-	0.05	-	0.050	-
$\bar{D}$	343.90		335.51		332.82		333.82		333.54		332.46	
$D(\bar{\Theta})$	359.84		345.24		343.82		343.56		341.66		330.50	
DIC	327.96		325.76		**321.80**		324.08		325.41		331.42	

Table 8.31: Evaluation of DIC with increasing number of functional parameters to estimate in the LNAS model based on the 2010 experimental dataset. The noise parameter $\sigma_{\gamma\gamma}$ is fixed at 0.05. d: Number of functional parameters.

Observations and remarks :

The results showed in Table 8.30 suggest that the modelling noise parameter $\sigma_{\gamma\gamma}$ is better fixed at 0.05 based on the DIC. As for the number of functional parameters to estimate, the best fitting is obtained in the case of three functional parameters which are the most influential ones to the model based on the parameter screening method of sensitivity analysis.

Therefore, the model calibration is carried out again with this obtained configuration.

III Method comparison

In the following test, all the Bayesian estimation methods and GLS are applied to the 2010 experimental dataset. The objective is to evaluate their performance facing real experimental scenario.

Settings :

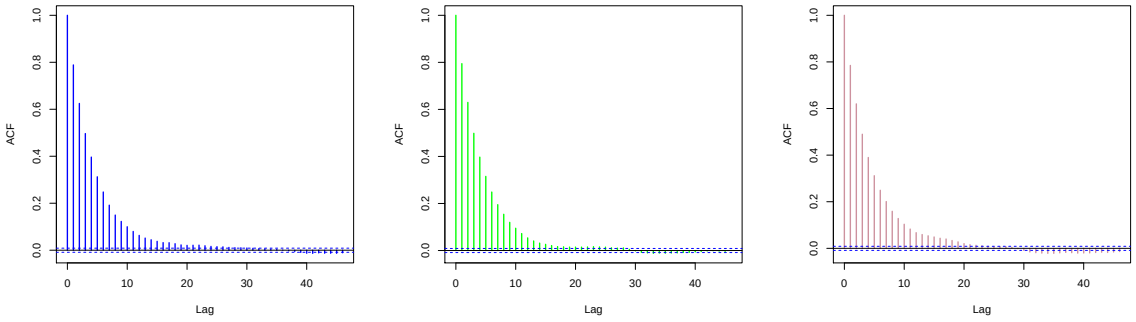
The configuration of three functional parameters and three noise parameters obtained in the previous test is adopted. The noise parameter $\sigma_{\gamma\gamma}$ is fixed to 0.05. The stopping criteria are used for each method to evaluate their estimation efficiency and computational cost. Noted that for GLS, although the estimation is performed with the deterministic version of the model, the observation noise parameters σ_g and σ_r can still be estimated with the estimators given in Section 6.3. However, the modelling noise parameters cannot be estimated, since no hidden state estimations are provided by the method.

	Prior		MCMC		DREAM		Interacting	
	μ_0	σ_0	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$
μ_a	3.40	0.20	3.553	0.079	3.554	0.071	3.554	0.059
γ_0	0.90	0.10	0.8612	0.0341	0.8599	0.0302	0.8594	0.0236
μ_γ	650.0	40.0	658.08	28.18	658.98	25.22	659.31	19.90
σ_g	$\sigma_g^2 \sim \mathcal{IG}(21,2)$		0.148	0.002	0.151	0.002	0.149	0.002
σ_r	$\sigma_r^2 \sim \mathcal{IG}(21,2)$		0.165	0.002	0.168	0.003	0.164	0.002
σ_q	$\sigma_q^2 \sim \mathcal{IG}(56,2)$		0.039	0.001	0.040	0.002	0.038	0.001
$-2\mathcal{L}(\bar{\Theta})$	348.52		344.63		344.89		344.72	
DIC	343.72		321.77		322.25		321.85	
	GLS*		RPF		EnKF		UKF	
	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$	$\mathbb{E}(\Theta\|Y)$	$\sqrt{\mathbb{V}(\Theta\|Y)}$
μ_a	3.653	0.102	3.554	0.078	3.557	0.062	3.410	0.090
γ_0	0.8120	0.0354	0.8607	0.0340	0.8995	0.0876	0.8717	0.0299
μ_γ	707.47	36.44	659.41	28.17	652.00	33.48	666.20	28.33
σ_g	0.138	-	0.152	0.002	0.148	0.003	0.167	0.005
σ_r	0.155	-	0.171	0.002	0.165	0.003	0.191	0.008
σ_q	-	-	0.060	0.003	0.045	0.004	0.065	0.007
$-2\mathcal{L}(\bar{\Theta})$	338.186		343.82		357.34		360.20	
DIC	-		321.80		324.51		332.17	

Table 8.32: Comparison of the estimations resulting from the one chain AMwG, DREAM, Interacting MCMC, GLS, RPF, EnKF and UKF in the case of the 2010 experimental dataset with noise parameter $\sigma_{\gamma\gamma}$ fixed to 0.05. GLS*: the estimation is performed based on the deterministic version of the LNAS model.

Observations and remarks :

Like in the case of simulated data presented previously, the three MCMC-based methods provided very close estimations. According to Table 8.32, similar results are also given by RPF. The DIC confirms their comparable estimation qualities, with the estimates of AMwG and RPF slightly better than those of the others. EnKF and UKF depend on normal assumptions, and UKF suffers from limited samples, their estimates are thus less accurate. The part of variability explained by the noises is thus more important.



(a) μ_a (b) γ_0 (c) μ_γ

Figure 8.22: Autocorrelation of estimations given by the one chain AMwG based on the 2010 experimental dataset with noise parameter $\sigma_{\gamma\gamma}$ fixed to 0.05.

The corresponding computational cost is given for all the above methods.

	UKF	EnKF	AMwG	Interacting MCMC	DREAM	RPF/CPF
Number of simulations	15	200 000	186 500	62 000[a]	153 700[b]	200 000
CPU time	2s	12m46s	11m18s	1h27m	58m	16m13s
Real computational time	2s	6m11s	11m18s	15m22s	9m17s	7m58s
Memory (Gb)	4e-3	2.85	2.48	3.34	6.41	14.27

Table 8.33: The averaged computational cost of three functional parameter estimation based on the 2010 experimental dataset, performed by one chain AMwG, DREAM, Interacting AMWG, RPF, UKF, EnKF and GLS. [a]: Interacting MCMC is performed with 1000 chains. [b]: DREAM is performed with 9 chains.

GLS provided the best point estimation based on the likelihood evaluation, however, since for the Bayesian methods, only the MMSE estimator is used instead of the MAP estimator, the likelihood evaluation based on the mean estimates is not really comparable.

For AMwG, the autocorrelation function is evaluated (Figure 8.22). In the case that only three functional parameters are estimated, the acceptance rate obtained is close to 22%, which also confirms an appropriate behaviour of convergence.

In Table 8.33, the computational cost of all the methods considered are given. This result illustrates the advantage of the parallel computation. Although RPF requires

more memory than the other methods, considering its estimation results, the real computational time is very reasonable. In the meantime, among the MCMC-based methods, the parallel chains did not help to reduce the computational time. For DREAM, the convergence of all the chains are difficult to obtain, and for Interacting MCMC, the 1000 parallel chains appear to be computational expensive.

8.8.3 Frequentist based iterative approaches

For the following tests, attempts are made to put more emphasis on the data despite the lack of information. The iterative scheme is considered. Since the iterative approaches aim to an point estimation that maximize the likelihood evaluation, to perform the model comparison, MAP based criterion such as AIC and BIC are more suitable, and are thus chosen.

We note that the model selection can also be carried out under this context to identify the best combination of noise parameters to estimate and the complexity of the model, as what we have performed for the Bayesian based approaches. However, this test is extremely computationally expensive with iterative based filtering methods with noise parameter assessment. Therefore, in this study, all the noise parameters are estimated by the conditional approaches.

Model selection

Settings :

Based on the sensitivity analysis (results detailed in Section 9.2.2), the model selection for the functional parameters is carried out with IRPF performing the fitting.

d	1	2	3	4	5	6
$2\mathcal{L}$	345.65	341.97	338.52	335.77	332.76	329.47
AIC	347.65	345.97	344.52	343.77	342.76	341.47
AICc	347.80	346.45	345.52	345.50	345.49	345.49
BIC	348.98	348.63	348.52	349.10	349.42	349.46

Table 8.34: Evaluation of AIC, AICc and BIC with increasing number of functional parameters to estimate in the LNAS model based on the 2010 experimental dataset. All the four noise parameters are estimated.

Observations and remarks :

Unfortunately, according to Table 8.34, the opinion is not shared by different criteria which put us in a dilemma. The AIC favours the model with six functional parameters estimated, while the BIC prefers the model with three functional parameters. AICc gives similar evaluation among the model with 3-6 parameters. To illustrate the performance of the iterative approaches, we decided to estimate the six functional parameters and the four noise parameters with the iterative approaches.

Parameter estimation

Settings :

The uncertainty related to parameter estimation was assessed by parametric bootstrap, by averaging the estimates obtained from both initializations for IRPF, and the estimation yields the best log-likelihood evaluation for IAMwG. The estimation results are presented in Table 8.36 together with estimated standard errors from 100 independent runs. The Monte Carlo sample size was increased geometrically and other implementation details are similar to those that have been used for the simulated data, which can be found in Section 8.6.

We finally compare the different filtering methods in the iterative framework. For the conditional IRPF approach, 60 000 particles were initialized with the same prior distributions as for the conditional IEnKF and IUKF approach. The maximum number of particles is 350 000. For the conditional IEnKF approach, an ensemble size of 350 000 is adopted for the comparison purpose. A parametric bootstrap is also implemented to evaluate the estimates' uncertainty. Standard deviations and confidence intervals are hence obtained from 300 bootstrap samples. The corresponding results are given in Table 8.39.

Observations and remarks :

In Table 8.37, we present the means and the standard deviations of a 300-bootstrap sample. Notice that the estimates of the three most important parameters (high sensitivity indices) are very satisfactory, with low standard deviations relatively to their mean values. Nevertheless, the estimations from the least influential parameters are less satisfactory with higher standard deviations as expected from the simulated case. A small bias in the mean estimated values from the bootstrap sample also confirms the difficulty in estimating these parameters. Note also that the estimates of the noise parameters are satisfactory.

Parameter	Initialization I				Initialization II			
	Initial distribution		Estimation		Initial distribution		Estimation	
	μ_0	σ_0	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	μ_0	σ_0	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$
μ_a	3.40	0.2	3.563	0.078	3.45	0.4	3.599	0.092
λ	58.0	3.0	59.40	2.14	57.0	5.0	59.36	3.36
γ_0	0.90	0.10	0.849	0.034	0.78	0.15	0.834	0.040
γ_f	0.18	0.05	0.188	0.021	0.20	0.05	0.198	0.023
μ_γ	650.0	40.0	642.01	32.12	630.00	50.0	640.81	36.21
σ_γ	350.0	30.00	319.96	28.84	300.0	40.0	306.13	36.58
σ_g	0.10	-	0.138	0.003	0.12	-	0.139	0.001
σ_r	0.10	-	0.152	0.003	0.12	-	0.155	0.001
σ_q	0.02	-	0.040	0.009	0.03	-	0.041	0.010
$\sigma_{\gamma\gamma}$	0.05	-	0.058	0.011	0.03	-	0.062	0.012
$-2\mathcal{L}$	347.82	-	333.44	0.36	338.71	-	335.96	0.43

Table 8.35: Comparison of the functional and noise parameter estimations resulting from two different initializations based on the one chain AMwG performed E-step of the iterative approach, in the case of the 2010 experimental dataset.

Note that in Table 8.36, the influence of the initialization is negligible for all the parameters but σ_γ, which is the least influential parameter. This is also suggested

by the small difference of the log-likelihood evaluations at the corresponding solutions. The variability from independent runs of the algorithm was relatively high for the modelling noise parameters (σ_q, $\sigma_{\gamma\gamma}$), but by averaging a moderate number of independent runs a better precision can be obtained.

Parameter	Initialization I				Initialization II			
	Initial distribution		Estimation		Initial distribution		Estimation	
	μ_0	σ_0	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$	μ_0	σ_0	$\mathbb{E}(\mathbb{E}(\Theta\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(\Theta\|Y))}$
μ_a	3.40	0.2	3.565	0.062	3.45	0.4	3.563	0.059
λ	58.0	3.0	59.78	0.37	57.0	5.0	59.33	0.46
γ_0	0.90	0.10	0.844	0.017	0.78	0.15	0.836	0.021
γ_f	0.18	0.05	0.190	0.002	0.20	0.05	0.198	0.001
μ_γ	650.0	40.0	642.64	5.96	630.00	50.0	641.02	6.46
σ_γ	350.0	30.00	319.59	13.62	300.0	40.0	297.78	14.80
σ_g	0.10	-	0.142	0.004	0.12	-	0.143	0.001
σ_r	0.10	-	0.164	0.004	0.12	-	0.167	0.001
σ_q	0.02	-	0.042	0.012	0.03	-	0.043	0.012
$\sigma_{\gamma\gamma}$	0.05	-	0.064	0.012	0.03	-	0.064	0.012
$-2\mathcal{L}$	347.82	-	331.17	0.23	338.71	-	331.18	0.26

Table 8.36: Comparison of the estimations resulting from two different initializations based on the iterative RPF/CPF approach (EM variant), in the case of the 2010 experimental dataset.

RPF	MLE	Mean	Std.
μ_a	3.564	3.562	0.123
λ	59.55	59.70	3.13
γ_0	0.840	0.843	0.044
γ_f	0.194	0.182	0.053
μ_γ	642.33	654.89	62.40
σ_γ	308.69	336.62	109.95
σ_g	0.142	0.143	0.032
σ_r	0.165	0.158	0.024
σ_q	0.042	0.039	0.008
$\sigma_{\gamma\gamma}$	0.064	0.073	0.015

MCMC	MLE	Mean	Std.
μ_a	3.563	3.565	0.169
λ	59.40	59.15	3.57
γ_0	0.849	0.850	0.062
γ_f	0.188	0.182	0.077
μ_γ	642.01	651.05	73.64
σ_γ	319.96	346.56	181.97
σ_g	0.138	0.139	0.006
σ_r	0.152	0.151	0.005
σ_q	0.040	0.039	0.002
$\sigma_{\gamma\gamma}$	0.058	0.060	0.003

Table 8.37: Uncertainty assessment of estimates based on the 2010 experimental dataset performed by parametric bootstrap and Iterative RPF and Iterative MCMC (EM variant with the E-step performed by RPF or MCMC).

	IUKF	IEnKF	IAMwG	IRPF
CPU time	1m11s	4h13s	5h37s	6h13s
Real computational time	1m11s	1h42m	5h37s	2h21s
Memory (Gb)	0.15	6.51	6.19	25.47

Table 8.38: The averaged computational cost for the point estimation of 100 tests, performed by the four iterative methods IEnKF, IRPF, IUKF and IAMwG, based on the 2010 experimental dataset.

	IEnKF		IRPF		IUKF		IAMwG		GLS*	
	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$
μ_a	3.600	0.154	3.564	0.123	3.900	0.270	3.563	0.169	3.591	0.142
λ	60.16	6.51	59.55	3.13	60.18	6.49	59.40	3.57	59.49	7.96
γ_0	0.830	0.087	0.840	0.044	0.800	0.064	0.849	0.062	0.824	0.112
γ_f	0.206	0.058	0.194	0.053	0.216	0.048	0.188	0.077	0.202	0.078
μ_γ	639.39	83.28	642.33	62.40	579.98	73.75	642.01	73.64	651.87	77.20
s_a	276.18	149.83	308.69	109.95	338.31	44.89	319.96	181.97	277.72	228.59
σ_g	0.137	0.028	0.142	0.032	0.156	0.070	0.138	0.006	0.132	-
σ_r	0.166	0.027	0.165	0.024	0.193	0.092	0.152	0.005	0.149	-
σ_q	0.040	0.004	0.042	0.008	0.052	0.048	0.040	0.002	-	-
$\sigma_{\gamma\gamma}$	0.061	0.002	0.064	0.015	0.072	0.031	0.058	0.003	-	-
$-2\mathcal{L}$	334.75	-	329.46	-	353.37	-	333.44	-	329.97	-

Table 8.39: Estimated values and standard deviations for all the parameters of the LNAS model provided by the iterative approaches as an EM variant with IEnKF, IRPF, IUKF and IAMwG to perform the E-step. GLS*: GLS is performed with the deterministic version of the model.

	Prior		AMwG		RPF		EnKF	
	μ_0	σ_0^2	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$	$\mathbb{E}(\Theta\|Y)$	$\mathbb{V}(\Theta\|Y)$
μ_a	3.40	0.04	3.556	0.006	3.551	0.005	3.507	0.013
λ	58.0	9.0	59.29	4.51	58.95	4.90	56.99	6.94
γ_0	0.90	0.01	0.8496	0.00116	0.8452	0.00108	0.8106	0.00161
γ_f	0.18	0.0025	0.178	0.00042	0.177	0.00038	0.211	0.00054
μ_γ	650.0	1600.0	647.42	921.23	650.1	946.91	644.99	1184.28
σ_γ	350.0	900.0	350.30	843.22	351.68	898.55	317.07	889.90
σ_g	$\sigma_g^2 \sim \mathcal{IG}(25,2)$		0.139	9.2e-6	0.140	1.3e-5	0.157	1.4e-5
σ_r	$\sigma_r^2 \sim \mathcal{IG}(25,2)$		0.163	9.5e-6	0.165	2.4e-5	0.166	3.1e-5
σ_q	$\sigma_q^2 \sim \mathcal{IG}(55,2)$		0.040	2.2e-7	0.046	2.5e-5	0.053	2.4e-5
$\sigma_{\gamma\gamma}$	$\sigma_{\gamma\gamma}^2 \sim \mathcal{IG}(55,2)$		0.038	2.4e-7	0.061	1.6e-4	0.071	2.5e-4
DIC	336.21		331.32		331.39		332.04	

Table 8.40: Comparison of the estimations resulted from the one chain AMwG, RPF and EnKF in the case of the 2010 experimental dataset.

Compared to the iterative RPF, Table 8.35 suggests that the iterative AMwG based algorithm provides estimates that are more dependent of the initial distribution. Unlike the RPF approach, the shrinkage of the variance results in a more important variability between the estimates given by independent runs for the iterative AMwG.

Since the IAMwG cannot be performed in a parallelized way as for the other SMC based iterative approaches, according to Table 8.39, it is the most time consuming. Judging from the estimation performance and the computational cost, despite the important memory requirement, IRPF appears to be the most efficient method.

We recall the estimation results (Table 8.29) given by the same Bayesian approaches (without the iterative scheme). In comparison, the iterative approaches provide results with better likelihood evaluations. In counterpart, the iterative approaches are much more computational expensive.

8.9 Discussion

Despite built on different inference bases, globally, all the three types of methods, the generalized nonlinear least squares method, the MCMC-based methods and the filtering methods illustrate their robustness in model calibration facing the dynamic state-space plant growth model LNAS. Only UKF based methods showed less good fits compared to the other methods (Table 8.39 and Table 8.32) due to its small sample size and the dependence of the normal assumption. GLS often gives unrealistic confidence intervals (CI for s_a with the 2010 experimental data), while other methods perform better in this respect yet with a much longer computational time. If only judging on the mean estimates, the estimations of iterative approaches yield better log-likelihood evaluation than the Bayesian approaches, which can be understood since they put more emphasis on the data and the prior distributions weight less in the final estimations. Although GLS mean estimates seem to score the best the log-likelihood evaluation in various occasions, the fact that it considers only the best fit estimate instead of probability distribution makes the error term less meaningful. In the meantime, GLS aims to find the MLE by avoiding the local maximums, while in the plant growth context, sometimes the local maximums are preferred for they are biologically more interpretable. On the other hand, the GLS estimator retains only the parameter set that illustrates the best fit, while the mean estimates of the Bayesian based methods provide the most frequently visited value, therefore, their estimates are not comparable.

If we compare the MCMC-based methods and the filtering methods, both of them are Monte Carlo methods. However, in the Bayesian framework, the MCMC-based methods are iterative while the filtering methods are not, for the former aim at $p(X_{0:n}|Y_{0:n})$ while the latter compute only $p(X_n|Y_{0:n})$. The filtering methods have the advantage of being sequential, and therefore being able to process on-line, while the MCMC-based methods cannot. Yet to achieve the off-line parameter estimation objective, the MCMC-based methods appear to be more stable with less important algorithmic error (Monte Carlo error). The inconvenient part is the need of full conditional distributions or the tuning of the instrumental proposal distribution in the MCMC-based algorithms, although the latter can be solved by applying an adaptive scheme. Conventionally, a successful implementation of the MCMC-based methods (achieve interpretable convergence) requires a pertinent prior knowledge of the studied subject and a good practice of the MCMC algorithms, whereas the filtering methods have usually less tuning parameters, there is no need of burn-in period, the concept is easier to understand and thus the implementation is simpler. The small drawback is that for an efficient exploration of the state-space, an important number of simulations are necessary.

Furthermore, regarding the computational cost, both the parallel-chain methods

and the SMC methods can be parallelized when a large sample size is required, while the one chain AMwG cannot. The counterpart for the parallel MCMC is that the convergence is difficult to achieve when an important number of chains are implemented. The burn-in period is difficult to define for each of the chains and therefore the posterior composed of pooled samples from all the chains are less accurate than one chain AMwG. In the case of Interacting MCMC, by taking only the last sample of each chain, although the mean estimation is very accurate for the computational power is concentrated on the zone of interest by the importance sampling scheme, unfortunately, a sampling bias is accordingly introduced, for the samples with less important acceptance probability are rarely visited. Therefore, despite its efficiency of finding the mean estimates, the resulting posterior is often less flat than it ought to be. This could be potentially an issue for an appropriate uncertainty assessment, and influence the future prediction results.

As a result, the following selection technique is suggested. If a rapid estimation is required to get a rough idea of the parameters, then the GLS is recommended when no prior information is available, and the Interacting MCMC is recommended when the user has a prior knowledge on the studied subject with a good understanding of the MCMC-based methods. On the other hand, if there is no time constraint, with a good knowledge of the MCMC based methods, a one chain AMwG is advised, so that the target distribution can be inferred appropriately. Finally, a compromise between the estimation quality and the computational time are CPF/RPF and EnKF, which are simple to develop and to implement and can be parallelized to be even more efficient.

N.B.

The model selection step is performed with RPF while other estimation methods can also be considered, which could potentially results in different conclusions.

In this study, only normal priors are used while non-informative uniform priors can also be used.

CHAPTER 9

Prediction and Uncertainty Assessment with Data Assimilation

Sequential data assimilation techniques are considered as invaluable tools due to their capability to improve prediction performances. The conventional strategy is to use reference models like SUCROS (Guérif and Duke, 1998, 2000; Launay and Guérif, 2005) or CERES (Dente et al., 2008) and to integrate the information obtained by remote sensing to enhance the prediction quality. A few methods were developed under this perspective (see Dorigo et al. (2007) for a review). The forcing method consists in replacing a state variable of the model by the observed data (Delécolle et al. (1992); Dente et al. (2008)). However, a considerable part of the model state variables cannot be or are not observed and thus cannot be updated simultaneously at each time step. Moreover, the method does not take into account the observation error, which should not be neglected considering the general lack of accuracy of remote sensing data.

Another approach is to use the available observation data to recalibrate some model parameters and / or initial states that may presumably vary with local conditions (Bouman, 1992; Guérif and Duke, 2000; Launay and Guérif, 2005). The drawback of this method is that it requires a sufficient number of data to perform the calibration, which is often costly and difficult to obtain. Besides, the global approach of this calibration step usually fails to capture and to maintain the system dynamics and thus is unable to improve the prediction accuracy.

In other research domains, the data assimilation problem have been commonly reformulated and studied with a Bayesian probabilistic perspective, which allows the sequential estimation of model states and parameters simultaneously (Jazwinski, 1970; Van Leeuwen and Evensen, 1996) in the framework of generalized state-space models. The most important applications concern weather forecasting (*e.g.* Anderson (2001))

and hydrology (*e.g.* Reichle et al. (2002)).

In plant growth modelling, the first attempt to adapt a relatively simple crop model into this perspective was made by Makowski et al. (2004). The method implementation relies on a probabilistic framework of crop model which is used to derive prior distributions of the model state variables and parameters at time steps with available observations while taking into account uncertainty in model prediction. Conditionally to the experimental observations and the observation error, posterior distributions are deduced according to Bayes' law. An updated prediction of the model state variables can thus be inferred. The procedure is repeated at all measurement dates. Classical filtering methods used for this purpose are Ensemble Kalman Filter (see Jones and Graham (2006) for an application in the context of crop models). However, among many competing data assimilation approaches, the Particle Filters, have gained their popularity because they are adaptive to nonlinearity and non-Gaussianity (see Naud et al. (2007) for an application in the context of crop models).

However, one important constraint of this approach lies on the fact that it requires the plant growth model described in a probabilistic framework, as a hidden Markov model (Cappé et al., 2005). The classical and complex crop models (like STICS (Brisson et al., 1998), APSIM (Keating et al., 2003), CERES (Jones and Kiniry, 1986), etc.) are not built in this perspective and their stochastic reformulation is therefore far from straightforward: the large number of involved processes may potentially lead to a drastic increase in the number of parameters to model process errors. One simple solution to circumvent this problem is to only consider observation errors (Guérif et al., 2006), but it may hinder a proper update of hidden state variables.

In this context, aiming at integrating limited information from different levels in order to comprehend and to predict reasonable outcome, we propose in this chapter a generic approach for data assimilation and prediction purpose in plant growth modelling, which consists of 3 steps: sensitivity analysis, model calibration and data assimilation. The performance of the data assimilation techniques with both Kalman filter based methods and particle filter based methods are tested in the prediction of biomass production and allocation in a dynamically evolving plant-growth model that can be formalized as a nonlinear stochastic state-space model. We demonstrate that this approach can also cope with classical complex crop models like STICS, originally built in a deterministic framework.

The chapter is organized as follows. We first outline the full methodology proposed for plant growth model prediction which consists of three principal steps. For each step, the concerning methods and algorithms are presented, followed by the application to the LNAS and STICS models with real experimental scenarios. Experimental data collected in a different year, location or with a new cultivar are used for prediction. The performances of the different algorithms are compared and the prediction related issues are discussed.

9.1 Three-step approach for prediction

With the purpose of achieving proper predictions with plant growth models given some experimental datasets, which can be considered as a way to provide prior knowledge, in this thesis, we propose a three step approach which consists of parameter selection, parameter estimation and data assimilation. In the first place, the least influential model parameters are screened using sensitivity analysis methods (Campolongo *et al.*Campolongo et al. (2007)) and are thus fixed to either their recommended values, or the center value of their prior confidence intervals. Here, we consider cases in which no satisfying problem specific prior distributions are available for the considered parameters, so that a first calibration step is performed on a previously obtained dataset in a similar context to provide appropriate prior distributions. Afterwards, the obtained prior distributions are used for data assimilation applications during which experimental data of early growth stages are assimilated to improve model prediction at later stages. Of course, the preliminary calibration step can be skipped if satisfying prior distributions are available.

9.1.1 Parameter selection

The estimation of the unknown parameters from the available experimental data is an important step. The prediction performance of the model depends on the accuracy of the model calibration (Makowski et al., 2006). However, it is unrealistic to estimate all the model parameters based on a limited dataset, for identifiability problem may occur, especially when a complex model with an important number of parameters is considered. Moreover, it is suggested that by fixing some parameters, the uncertainty related to the estimated parameters could be reduced (Guo et al., 2006; Lamboni et al., 2011). Therefore, a common strategy can be described as follows :

1. Select a subset of the model parameters and rank them according to sensitivity analysis. Fix the least influential parameters to some nominal values (Wallach et al., 2002; Monod et al., 2006).
2. Based on the ranking provided by sensitivity analysis, decide the number of parameters to estimate based on the selection criteria (AIC / AICc / BIC / DIC) according to the estimation method employed.

Sensitivity analysis

When a model contains a large number of parameters, as it is often the case for plant growth models, parameter estimation based on limited experimental data is usually considered to be a key issue which may affect strongly the quality of model prediction with important estimates uncertainty. Therefore, sensitivity analysis (Campolongo et al., 2007; Monod et al., 2006) can be applied in advance to select the most influential parameters to be estimated, whereas those screened as the least influential ones can be fixed to any values in their domains, conventionally the mean values are chosen. In

the context of sensitivity analysis, this method is called "screening" or "factor fixing" (Campolongo et al., 2007).

With this purpose, we implement the algorithm proposed by Wu et al. (2012) to compute Sobol's indices (first order and total order) of all the parameters of the studied model based on its deterministic version. The generalized least-square criteria is chosen as the output for the variance decomposition and the parameters are ranked according to their total-order indices. More details of this method can be found in Section 6.1.

We remark that a potentially important issue in sensitivity analysis can be the determination of the distributions representing the uncertainty in the inputs, particularly when the parameters are empirical with no explicit biological meaning, and specific methods have to be devised for this purpose (Wernsdörfer et al., 2008). Generally speaking, in crop models, all the functions represent well-known processes, as for sugar beet with the LNAS model or for winter wheat with the STICS model, with an important literature, so that it is relatively easy to assess proper variation intervals for the parameters. The question for us is the following: for each parameter, what is the reasonable range of possible values, or when calibrating our system what would be acceptable values for the parameters? In regards to such criteria, it also seems appropriate to select the uniform distribution after assessing the appropriate variation intervals. Moreover some tests were performed by increasing the range of the least important parameters while decreasing the range of the most important ones to check the possible bias induced by the choice of the ranges, but it did not affect the importance ranking of the parameters, which is the result of interest for parameter screening.

Model selection criteria

Once the influence and the ranks of the parameters are determined, the parameter selection problem is equivalent to a model selection problem in which candidate models have different numbers of parameters. From a model with only one parameter (the most influential one) to the model with all the parameters, the number of parameters is increased one by one with respect to the ranking of the parameters. In other words, the most influential parameters have the priority to be included in the model. The objective is to find a balance between the complexity of the model and the quantity of the data available based on an appropriate selection criterion. Indeed, by adding more parameters to estimate, the likelihood can be increased. Unfortunately, this may also entails overfitting. To avoid this problem, model selection criteria conventionally take into account a penalty term for the number of parameters to estimate, such as the Akaike Information Criterion (AIC), the corrected AIC (known as AICc which contains an additional correction term for finite sample size), the Bayesian Information Criterion (BIC) and the Deviance Information Criterion (DIC) usually considered for Bayesian inference.

For the detailed description of these criteria, please refer to Section 6.2.

9.1.2 Parameter estimation

In this step, numerous parameter estimation methods can be carried out based on the calibration dataset. In this thesis, both Bayesian and frequentist perspectives are considered. First, we distinguish the selected functional model parameters Θ_1 from the noise model parameters Θ_2. A joint update scheme is adopted to estimate Θ_1 and Θ_2 as follows (detailed in Section 6.3) :

the estimation of the selected functional parameters Θ_1 are conducted with different methods, mainly with both MCMC based methods (Adaptive MCMC, Interacting MCMC, DREAM) and filtering-based SMC methods (UKF, EnKF, RPF). In addition, we also employ one frequentist approach, the Generalized Least Squares estimator, which is commonly used in plant growth model calibration.

Moreover, in order to put more emphasis on the calibration dataset and to reduce the influence of the prior distribution, an iterative scheme is considered and formulated as a variant of an EM type of algorithm. More precisely, the posterior distribution of iteration k is regarded as the prior distribution for iteration $k+1$ as if the same dataset is repeatedly obtained. In this way, the lack of available data can be mended. However, since this iterative framework results in diminishing variances that approach 0, the uncertainty related to the unknown parameters is assessed by parametric bootstrap. (For a theoretical justification of this approach when normal priors are used for the selected parameters, please refer to Section 5.1).

Note that except with GLS, all the other methods allow an estimation of hidden state variables combined with the estimation of Θ_1. Once the estimation of Θ_1 and the hidden states variables of interest are converged, the modelling and measurement noises Θ_2 can thereafter be evaluated. Three methods are proposed in this thesis to evaluate the noise parameters : noise parameter selection, noise parameter update with conjugate priors for MCMC based methods and empirical estimation (see Chapter 6 for detailed descriptions).

Based on our tests presented in Chapter 8, we propose in Table 10.1 a strategy with the objective to select the most appropriate method according to the type of situation, the specific need / emphasis in this step.

9.1.3 Data assimilation

In this prediction step, a new experimental dataset, comparable to the calibration dataset (for instance same type of crop but observed in a different year, or at a different location, or of different genotype...) with few early measurements is introduced. With the purpose of performing predictions of yield or other state variables of interest, an on-line filtering approach (such as EnKF, CPF/RPF and UKF) can again be carried out by regarding the results of the calibration step as prior information (Chen and Cournède, 2014). In this step, the probability density is represented by a large number of samples (particles) which evolve with time. Hence, after a short readjustment period while model parameters and state variables are updated based on the available measurements, the particles continue to propagate so as to forecast the system evolution and to evaluate the uncertainty related to these state variables of interest.

	Conditions		Recommended method(s)
•	Time constraint, no reliable prior, only point estimation is needed	⇒	GLS
•	Time constraint, slightly non linear model	⇒	UKF
•	Unknown parameter vector with important dimension	⇒	DREAM
•	Precision of the mean value	⇒	Interacting MCMC
•	Reliable prior, need precise CI%, no time constraint, implementation expertise in MCMC	⇒	One long run AMwG
•	Reliable prior, time constraint	⇒	CPF, EnKF
•	Less reliable priors	⇒	Iterative SMC, especially ICPF/IRPF

Table 9.1: Selection strategy for estimation method.

Note that as a stochastic approach, the prediction and the data assimilation results entail Monte Carlo errors. To reduce the uncertainty related to both the mean estimates and the variance estimates, in our implementation, a large number of particles are chosen (350 000) which generally ensure an appropriate precision.

Comparison criteria
For method comparison purpose, the Root Mean Squared Error for Prediction (RMSEP) and Relative Error (RE) are chosen as the criteria to evaluate the quality of prediction in our application, for they seem to be the most suitable considering the structure of our data.

If n denotes the total number of prediction dates, $\hat{Y}_i$ and Y_i^{exp} denote the mean predicted value and the experimental observation for the i^{th} prediction date respectively, then for each type of observation (Q_g and Q_r in our application), the RMSEP is defined as :

$$\text{RMSEP} = \sqrt{\mathbb{E}[(Y^{exp} - \hat{Y})^2]}.$$

The unbiased estimator of RMSEP can be defined as follows (Wallach et al., 2006) :

$$\hat{\text{RMSEP}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i^{exp})^2}.$$

The RE can be obtained by :

$$\text{RE} = \frac{1}{Y_i^{exp}}|\hat{Y}_i - Y_i^{exp}|.$$

9.2 Application to the LNAS model

9.2.1 Experimental data

The data used for this study were obtained by the French institute for sugar beet research (ITB, Paris, France) in 2006, 2008 and 2010 with slightly different cultivars

and in different locations with different observed densities (details of the experimental protocols can be found respectively in Lemaire et al. (2008), Lemaire et al. (2009) and Baey et al. (2013a)). For the test case, the 2010 dataset is chosen for estimation since more observation points are available compared to the other five datasets (of course, in real applications, the older datasets are supposed to be used for parameter estimation). Dry matter of root and leaves were collected on 50 plants at 14 dates (in days after sowing):

$$\mathcal{O}_{2010} = \{54, 68, 76, 83, 90, 98, 104, 110, 118, 125, 132, 139, 145, 160\}\,.$$

whereas for assimilation and prediction, other datasets (2006 and 2008) are used. The same type of observations were made at 7 different dates given by

$$\mathcal{O}_{2006} = \{54, 59, 66, 88, 114, 142, 198\}\,,$$

and

$$\mathcal{O}_{2008} = \{39, 60, 67, 75, 88, 122, 158\}\,,$$

respectively. For each plant, the green foliage mass denoted by Q_g and the root compartment mass denoted by Q_r were measured. The observation vector Y_n is obtained by averaging each data on all the samples and extrapolated at m^2 level by multiplying by the observed density.

We note that among the four 2008 datasets used for assimilation in different density conditions, three of them are of the same genotype as in the 2010 dataset (11.9 plants/m^2) *Radar*, but with different density conditions (5.4 plants/m^2, 10.9 plants/m^2, 16.4 plants/m^2). The other 2008 dataset is of a different genotype *Harmonia* with a density of 10.7 plants/m^2. As for the 2006 dataset, the density condition is 9.6 plants/m^2.

9.2.2 Results and method comparison

Step 1.1: Parameter ranking by Sensitivity Analysis

Sobol's indices (first order and total order) of all the functional parameters of the LNAS model (without counting the noises parameters) were computed using a generalized least-squares criterion as output (see Cournède et al. (2013)). According to the sensitivity analysis result, we screened the parameters μ_{sen} and s_{sen}, as their total order indices are all below 0.02. We note that for these two senescence related parameters, some complementary experimental data can be used to determine appropriate values. For the six other functional parameters, we choose to use the model selection approach to determine which of them will be estimated from the experimental data.

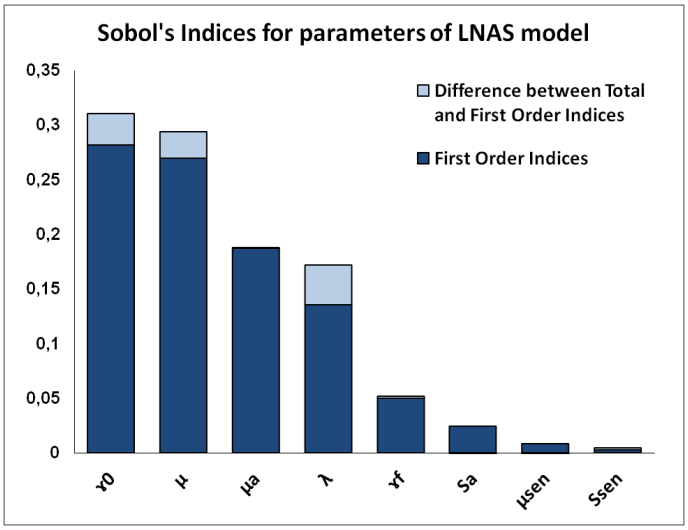
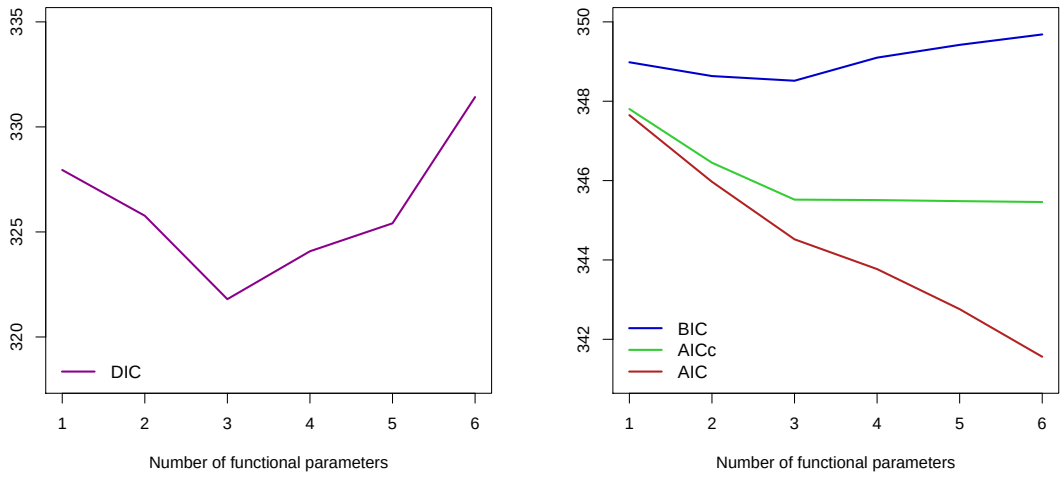


Figure 9.1: Comparison of the first and total order indices for parameters of the LNAS model: μ_a, λ, γ_0, γ_f, μ_γ, s_a, μ_{sen}, s_{sen}.

Step 1.2: Model selection



(a) Bayesian approaches

(b) Iterative approaches

Figure 9.2: (a): Model selection based on the DIC with $\sigma_{\gamma\gamma}$ fixed at 0.05 for the Bayesian approaches. (b): Model selection based on the AIC, the AICc and the BIC for the iterative approaches with all the noise parameters estimated.

Settings :

Based on the sensitivity analysis results, the full unknown parameter vector for the deterministic part of the model is assumed to be $\Theta_1 = (\mu_\gamma, \lambda, \mu_a, \gamma_0, \gamma_f, s_a)$ and the unknown noise parameter vector is $\Theta_2 = (\sigma_Q, \sigma_\gamma, \sigma_g, \sigma_r)$. To determine the number of parameters to estimate, a model selection is conducted. RPF is performed with different configurations of the LNAS model to determine the best fitting. Due to the

identifiability issue of the modelling noise parameter $\sigma_{\gamma\gamma}$, its value is fixed for the Bayesian approaches, while for the frequentist iterative approaches, all the four noise parameters are estimated. The implementation details and detailed test results can be found in Section 8.8.2 for Bayesian inference and Section 8.8.3 respectively for the iterative approaches.

Observations and remarks :

Figure 9.2 sums up the model selection results. For the Bayesian approaches, the Deviance Information Criterion suggests that the model of three functional parameters outperforms the others. On the other hand, for the iterative approaches, the three model selection criteria give different conclusions. For the purpose of illustrating the robustness of the proposed iterative approach, the model with six functional parameters is selected and considered as the most robust one. Moreover, the AIC are generally preferred in a prediction context.

Step 2: Parameter Estimation

For this second step, we can perform the parameter estimation with any of the methods presented in Chapters 4 and 5 based on the 2010 experimental dataset. Below, we recall the results obtained in Chapter 8 for the two types of approaches :

- the Bayesian approaches with three functional parameters estimated $\{\mu_a, \mu_\gamma, \gamma_0\}$ as suggested by the DIC,
- the iterative approaches with six functional parameters estimated $\{\mu_a, \lambda, \mu_\gamma, \sigma^2_\gamma, \gamma_0, \gamma_f\}$ as suggested by the AIC and AICc.

Note that the estimates given by the frequentist method GLS are also listed as a reference.

Step 2a: Parameter Estimation with Bayesian approaches

The Bayesian inference is carried out based on the 2010 experimental dataset. The Bayesian approaches presented in Chapter 4 are implemented, the configuration and the implementation details of which are detailed in Section 8.8.2. Here we simply recall the obtained results (Table 8.32).

Step 2b: Parameter Estimation with iterative approaches

In the same fashion, the iterative approaches presented in Chapter 5 are applied to the 2010 dataset. The obtained results are summarized in Table 8.39. For more details regarding the implementation, please refer to Section 8.8.3.

	Prior		AMwG		DREAM		Interacting	
	μ_0	σ_0	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$
μ_a	3.40	0.20	3.553	0.079	3.554	0.071	3.554	0.059
γ_0	0.90	0.10	0.8612	0.0341	0.8599	0.0302	0.8594	0.0236
μ_γ	650.0	40.0	658.08	28.18	658.98	25.22	659.31	19.90
σ_g	$\sigma_g^2 \sim \mathcal{IG}(21,2)$		0.148	0.002	0.151	0.002	0.149	0.002
σ_r	$\sigma_r^2 \sim \mathcal{IG}(21,2)$		0.165	0.002	0.168	0.003	0.164	0.002
σ_q	$\sigma_q^2 \sim \mathcal{IG}(56,2)$		0.039	0.001	0.040	0.002	0.038	0.001
$-2\mathcal{L}(\bar{\theta})$	348.52		344.63		344.89		344.72	
DIC	343.72		321.77		322.25		322.94	
	GLS*		RPF		EnKF		UKF	
	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$
μ_a	3.653	0.102	3.554	0.078	3.557	0.062	3.410	0.090
γ_0	0.8120	0.0354	0.8607	0.0340	0.8995	0.0876	0.8717	0.0299
μ_γ	707.47	36.44	659.41	28.17	652.00	33.48	666.20	28.33
σ_g	0.138	-	0.152	0.002	0.148	0.003	0.167	0.005
σ_r	0.155	-	0.171	0.002	0.165	0.003	0.191	0.008
σ_q	-	-	0.060	0.003	0.045	0.004	0.065	0.007
$-2\mathcal{L}(\bar{\theta})$	338.186		343.82		357.34		360.20	
DIC	-		321.80		322.51		332.17	

Table 9.2: Comparison of the estimations resulting from the one chain AMwG, DREAM, Interacting MCMC, GLS, RPF, EnKF and UKF in the case of the 2010 experimental dataset with noise parameter $\sigma_{\gamma\gamma}$ fixed to 0.05. GLS*: reference results.

	IEnKF		IRPF		IUKF		IAMwG		GLS*	
	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(\mathbb{E}(P\|Y))$	$\sqrt{\mathbb{V}(\mathbb{E}(P\|Y))}$	$\mathbb{E}(P\|Y)$	$\sqrt{\mathbb{V}(P\|Y)}$
μ_a	3.600	0.154	3.564	0.123	3.900	0.270	3.563	0.169	3.591	0.142
λ	60.16	6.51	59.55	3.13	60.18	6.49	59.40	3.57	59.49	7.96
γ_0	0.830	0.087	0.840	0.044	0.800	0.064	0.849	0.062	0.824	0.112
γ_f	0.206	0.058	0.194	0.053	0.216	0.048	0.188	0.077	0.202	0.078
μ_γ	639.39	83.28	642.33	62.40	579.98	73.75	642.01	73.64	651.87	77.20
s_a	276.18	149.83	308.69	109.95	338.31	44.89	319.96	181.97	277.72	228.59
σ_g	0.137	0.028	0.142	0.032	0.156	0.070	0.138	0.006	0.132	-
σ_r	0.166	0.027	0.165	0.024	0.193	0.092	0.152	0.005	0.149	-
σ_q	0.040	0.004	0.042	0.008	0.052	0.048	0.040	0.002	-	-
$\sigma_{\gamma\gamma}$	0.061	0.002	0.064	0.015	0.072	0.031	0.058	0.003	-	-
$-2\mathcal{L}$	334.75	-	329.46	-	353.37	-	333.44	-	329.97	-

Table 9.3: Estimated values and standard deviations for all the parameters of the LNAS model provided by the iterative approaches as an EM variant with IEnKF, IRPF, IUKF and IAMwG to perform the E-step. GLS*: reference results.

Step 3: Data Assimilation and model prediction

In this step, a SMC filtering method is used to perform data assimilation and model prediction based on a new dataset. The model calibrations carried out by both the Bayesian inference and the iterative approaches are considered.

Step 3a: Data Assimilation with the calibrations performed by Bayesian inference

We first investigate the predictive capacity of the Bayesian estimates. The posterior distribution resulting from the Bayesian inference was used as prior information in the assimilation step, in which the (classical) CPF algorithm is applied.

Data assimilation and model prediction with three functional parameters
In this first attempt, the optimized configuration obtained in the previous model selection step (estimation with three functional parameters) is adopted. The objective is to compare the prediction results with and without data assimilation of data from the early growth stages.

Settings :

For both the 2006 and 2008 datasets, 500000 particles are simulated, all but the last two measurements (corresponding to data until day 114 for the 2006 dataset, and until day 88 for the 2008 dataset) are used to update and correct the parameter and the state estimates. The propagation of particles through the stochastic dynamic model is carried on without any further correction until day 198 for the 2006 dataset and day 158

for the 2008 dataset. At last, the simulated values of the state variables Q_g and Q_r on day 142 and day 198 (resp. day 122 and day 158 for the 2008 dataset) given by all the particles as well as their associated weights are used to build the posterior predictive distributions, from which both the mean prediction and the credibility interval are obtained.

In order to provide reference values of the prediction without assimilation, an Uncertainty Analysis (UA) is performed. 500000 particles are initialized in the same way as in the CPF approach, from the prior distribution given by the parameter estimation step, and the distribution of the model outputs of interest is approximated by propagating independently through the stochastic dynamic system all the particles. No parameter or state update is performed from the experimental data of the early stages.

Observations and remarks :

The prediction results based on the Bayesian estimations seem to be unsatisfactory after being performed with both the 2006 and 2008 datasets (Tables 9.4 and 9.5 respectively). Almost all the predictions relative errors are above 10% for the 2006 dataset, which is quite disappointing. The relative errors for the Q_g prediction of the 2008 dataset appear also to be quite significant. The estimation of EnKF which is associated with larger variance in the calibration step seems to show slightly better performance in the prediction step. Considering that the estimations given by AMwG, DREAM and RPF in the previous step are very close, the similarity of their prediction results is understandable.

	Real value	AMwG		DREAM		RPF	
		RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
$Q_g\,(t_{142})$	355.2	21.9%	85.2	22.2%	84.4	21.8%	85.4
$Q_g\,(t_{198})$	320.6	25.1%	76.3	25.0%	74.8	25.2%	76.0
$Q_r\,(t_{142})$	1459.2	24.8%	302.2	24.7%	300.0	24.9%	301.9
$Q_r\,(t_{198})$	2400.0	14.6%	454.1	14.6%	453.2	14.7%	453.8

GLS*		Interacting		EnKF		UKF	
RE (%)	Std.	RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
20.0%	79.1	30.3%	92.2	11.8%	78.4	19.9%	95.4
24.1%	71.5	32.2%	86.3	17.9%	72.8	22.3%	88.0
24.3%	284.9	26.9%	301.2	24.9%	302.0	17.8%	329.0
14.5%	430.6	16.2%	454.4	15.4%	454.2	9.8%	494.1

Table 9.4: Comparison of the prediction capacity of the estimations resulting from the one chain AMwG, DREAM, Interacting MCMC, GLS, RPF, EnKF and UKF in the case of the 2006 experimental dataset with data assimilation. RE: relative error in %. GLS*: the GLS estimates are given as a reference.

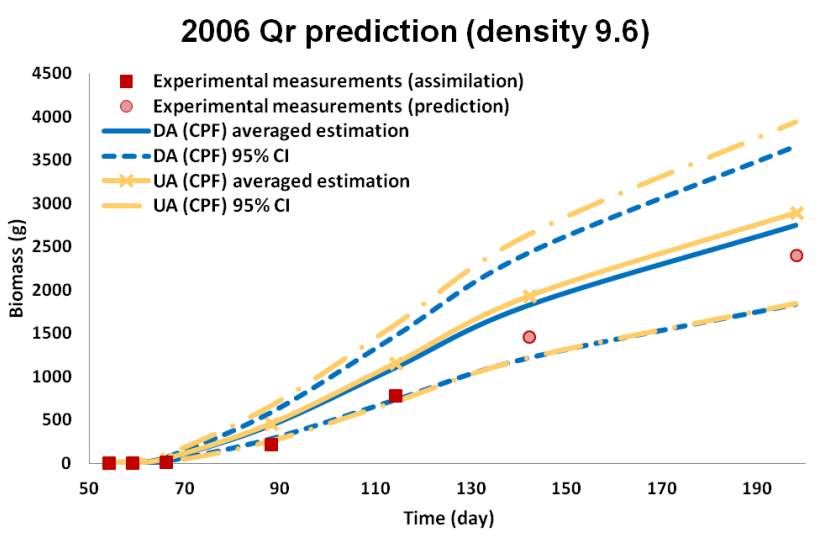


Figure 9.3: Comparison of the predictions for Q_r of the LNAS model given by the CPF approach with the RPF's estimates and predictions given by the uncertainty analysis (UA) based on the 2006 dataset. The red squares correspond to the assimilated experimental data while the pink squares represent the data used for validation.

	Real value	AMwG		DREAM		RPF	
		RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
$Q_g\,(t_{122})$	373.5	22.1%	68.2	22.3%	68.0	22.2%	68.5
$Q_g\,(t_{158})$	380.6	24.0%	71.3	24.1%	70.2	24.2%	71.1
$Q_r\,(t_{122})$	1559.1	6.9%	242.5	6.6%	241.8	6.8%	242.6
$Q_r\,(t_{158})$	2327.7	8.3%	357.7	8.4%	357.3	8.4%	357.6

GLS*		Interacting		EnKF		UKF	
RE (%)	Std.	RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
20.5%	61.7	31.1%	69.4	19.9%	63.3	24.8%	79.3
23.8%	69.3	32.8%	73.2	25.3%	23.7	25.3%	89.5
7.3%	233.2	8.7%	243.5	6.8%	244.6	11.7%	329.0
9.5%	349.1	10.3%	357.7	10.4%	357.9	5.8%	494.1

Table 9.5: Comparison of the prediction capacity of the estimations resulting from the one chain AMwG, DREAM, Interacting MCMC, GLS, RPF, EnKF and UKF in the case of the 2008 experimental dataset with data assimilation. RE: relative error in %. GLS*: the GLS estimates are given as a reference.

Figure 9.3 displays the prediction results with and without data assimilation when the calibration was performed with the RPF method. We remark the small reduction of the uncertainties associated to the prediction with DA compared to the results of UA. In this concrete example, DA did not apparently improve much the prediction performance. Indeed, most methods showed the same poor prediction performance. The prediction results performed based on different calibrations are comparable for both Q_r and Q_g in both the 2006 and 2008 datasets. It could be resulted from either

the lack of data, or the misspecification of the model, or the lack of liberty to re-adjust the model with only three functional parameters. The first two possibilities are difficult to verify, while the last possibility can be verified by taking into account all the possible candidate models selected in the first step.

Data assimilation and model prediction with model comparison
The following test is conducted to identify the best candidate model (with different numbers of parameters calibrated in step 2 and updated in step 3) based on the perspective of data assimilation in the prediction step.

Settings :

Six configurations of the LNAS model with different numbers of functional parameters being calibrated in step 2 and updated in step 3 are evaluated with regards to their prediction performance. RPF is selected to perform the calibration based on the 2010 dataset, and CPF is used to perform the data assimilation and prediction for all the six candidate models. For both situations, 350 000 particles are used. In the following, we present the prediction results performed based on the 2006 experimental dataset (density 9.6) and on the 2008 experimental dataset (density 10.9). The average estimates are given with their corresponding 95% credibility intervals obtained with the posterior distributions.

The Root Mean Squared Error for Prediction (RMSEP) and Relative Error (RE) are used as the criterion to evaluate the quality of prediction for model comparison purpose.

Nb of	1		2		3	
parameters d	RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
$Q_g\,(t_{142})$	67.9%	96.6	44.6%	88.4	21.8%	85.4
$Q_g\,(t_{198})$	64.5%	84.6	44.1%	79.4	25.2%	76.0
$Q_r\,(t_{142})$	31.2%	377.8	28.3%	349.9	24.9%	301.9
$Q_r\,(t_{198})$	19.1%	564.8	17.5%	516.6	14.7%	453.8
	4		5		6	
	RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
$Q_g\,(t_{122})$	7.2%	64.5	11.1%	68.6	15.6%	73.3
$Q_g\,(t_{158})$	9.5%	56.1	14.2%	70.1	11.4%	67.1
$Q_r\,(t_{122})$	8.7%	261.0	6.2%	312.3	12.7%	302.3
$Q_r\,(t_{158})$	7.6%	426.9	4.5%	468.9	8.7%	456.4

Table 9.6: Comparison of the prediction capacity with different numbers of parameters estimated by RPF in the calibration step and updated in the data assimilation step by CPF based on the 2006 experimental dataset. RE: relative error in %.

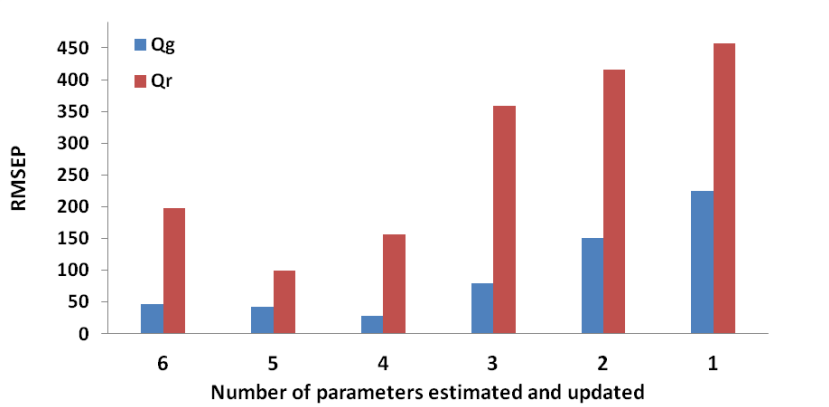


Figure 9.4: Comparison of model prediction capacity of different numbers of parameters estimated and updated by RPF, evaluated by Root Mean Squared Error of Prediction (RMSEP) based on the 2006 datasets with assimilation of data at the early growth stages.

Observations and remarks :

According to Table 9.6 and Table 9.7, we notice that the selected configuration in the first step (model selection for the calibration) did not result in the best performance in the data assimilation and prediction step. For the 2006 dataset, the best prediction result is obtained with the models with 4 and 5 parameters. For the 2008 dataset, again the model with 4 parameters appears to provide the most accurate prediction.

Nb of	1		2		3	
parameters d	RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
$Q_g\,(t_{122})$	60.4%	69.1	26.4%	68.8	22.2%	68.5
$Q_g\,(t_{158})$	54.0%	72.0	42.0%	71.0	24.0%	71.1
$Q_r\,(t_{122})$	11.0%	288.3	8.2%	270.5	6.8%	242.6
$Q_r\,(t_{158})$	12.6%	422.0	9.3%	395.3	8.4%	357.6
	4		5		6	
	RE (%)	Std.	RE (%)	Std.	RE (%)	Std.
$Q_g\,(t_{122})$	6.3%	58.1	15.9%	66.2	13.9%	65.5
$Q_g\,(t_{158})$	4.8%	58.9	18.3%	70.9	14.2%	69.2
$Q_r\,(t_{122})$	2.6%	250.5	5.4%	241.1	4.8%	236.7
$Q_r\,(t_{158})$	3.4%	370.2	7.0%	355.7	6.0%	350.2

Table 9.7: Comparison of the prediction capacity with different numbers of parameters estimated by RPF in the calibration step and updated in the data assimilation step by CPF based on the 2008 experimental dataset (density 10.9). RE: relative error in %.

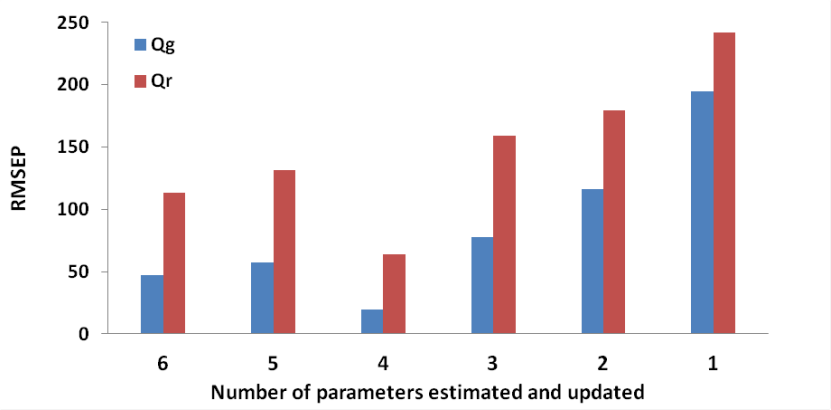


Figure 9.5: Comparison of model prediction capacity of different numbers of parameters estimated and updated by RPF, evaluated by Root Mean Squared Error of Prediction (RMSEP) based on the 2008 datasets with assimilation of data at the early growth stages.

Figure 9.4 and Figure 9.5 indicate that the model selection results with the perspectives of calibration and of prediction may not share the same conclusion, especially when models with fewer parameters are preferred in the calibration step since the penalty term for the number of parameters to be estimated is important in the selection criteria. While in the prediction step, it seems that models with more functional parameters calibrated tend to outperform those with less parameters. One potential issue is that with limited data available to perform the calibration, models with more parameters could entail identifiability and over-fitting problems. Undoubtedly, it is also important to note that this model comparison conducted based on the evaluation of prediction capacity is only a posterior comparison, in real applications, the choice regarding the model configuration should be made before the assimilation step. This issue is further discussed in Section 9.2.8.

Step 3b: Data Assimilation with the calibration performed by iterative approaches

Regarding the estimates of the iterative approaches, the model with six functional parameters was selected in the calibration step. Given the fact that generally, the IRPF estimates obtain often better likelihood evaluation than those of the other iterative approaches, the prediction performance based solely on the IRPF estimates is thoroughly investigated in the first place.

Settings :

Similar to the settings used with the Bayesian estimates for prediction purpose, based on the parameter estimation results of the 2010 dataset, we compare the predictive capacity of the model for the 2006 and the 2008 experiments, with and without data assimilation (denoted DA and UA respectively), regarding the last two observations (t_{142} and t_{198} for 2006 and t_{122} and t_{158} for 2008). The RPF approach is used

with 350 000 particles to perform DA. The average estimates are given with their corresponding 95% credibility intervals obtained with the posterior distributions.

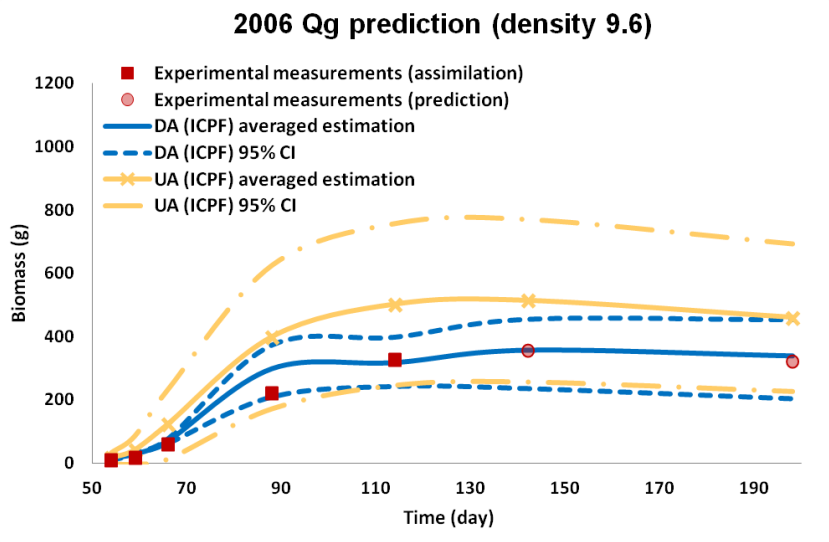


Figure 9.6: Comparison of the predictions for Q_g with and without data assimilation regarding the 2006 dataset. Based on the calibration performed by IRPF, the data assimilation is performed by the RPF approach and the predictions without data assimilation is performed by the uncertainty analysis. The red squares correspond to the assimilated experimental data while the pink squares represent the data used for validation.

Observations and remarks :

Although the six experiments are quite different (different locations, in different years, and different cultivars), the proposed RPF-based data assimilation approach was proved to provide fair predictions in most cases and managed to reduce the prediction errors compared to the results obtained with UA (without data assimilation), as demonstrated by Figure 9.6. These results confirm the assumption that by estimating only the three most influential functional parameters while fixing the others to recommended values like in the previous test case, the data assimilation cannot be performed efficiently. It is particularly spectacular for the leaf biomass (Q_g) prediction (Table 9.8, Table 9.15).

	Real Data 2006	DA estimates (relative error in %)	95% CI	UA estimates (relative error in %)	95% CI
$Q_g\,(t_{142})$	355.2	357.5 (0.7%)	[202.5; 518.6]	513.3 (44.5%)	[256.0; 770.3]
$Q_g\,(t_{198})$	320.6	339.4 (5.9%)	[174.5; 516.2]	459.9 (43.4%)	[226.3; 693.7]
$Q_r\,(t_{142})$	1459.2	1583.8 (8.5%)	[1238.8; 1979.3]	1928.7 (32.2%)	[1222.0; 2639.4]
$Q_r\,(t_{198})$	2400.0	2469.5 (2.9%)	[1933.5; 3080.4]	2893.8 (20.6%)	[1849.0; 3938.3]

Table 9.8: Comparison of the prediction results of the LNAS model with and without data assimilation based on the 2006 dataset (density 9.6).

	Real Data 2008	DA estimates (relative error in %)	95% CI	UA estimates (relative error in %)	95% CI
$Q_g\,(t_{122})$	373.5	403.6 (8.1%)	[294.4; 526.0]	521.7 (39.7%)	[253.7; 789.7]
$Q_g\,(t_{158})$	380.6	408.7 (7.4%)	[288.0; 570.3]	510.6 (34.2%)	[247.1; 774.1]
$Q_r\,(t_{122})$	1559.1	1466.7 (5.9%)	[1094.4; 1920.6]	1673.0 (7.3%)	[1053.6; 2292.3]
$Q_r\,(t_{158})$	2327.7	2187.6 (6.0%)	[1610.1; 2889.2]	2349.5 (0.9%)	[1494.1; 3205.0]

Table 9.9: Comparison of the prediction results of the LNAS model with and without data assimilation based on the 2008 dataset (density 10.9).

	Real Data 2008	ICPF estimates (relative error in %)	95% CI	UA estimates (relative error in %)	95% CI
$Q_g\,(t_{122})$	385.8	407.2 (5.6%)	[324.4; 510.1]	525.9 (36.3%)	[379.6; 672.3]
$Q_g\,(t_{158})$	345.1	416.9 (20.8%)	[315.9; 552.7]	502.0 (45.5%)	[358.9; 645.0]*
$Q_r\,(t_{122})$	1326.8	1410.6 (6.3%)	[1200.4; 1641.2]	1654.5 (24.7%)	[1372.5; 1936.6]*
$Q_r\,(t_{158})$	1778.9	2051.1 (15.3%)	[1725.5; 2417.3]	2350.6 (32.1%)	[1924.6; 2776.7]*

Table 9.10: Comparison of model prediction with and without data assimilation based on the 2008 dataset variety *Harmonia* (density 10.7). *: the predicted credibility interval does not contain the real observed data.

	Real Data 2008	DA estimates (relative error in %)	95% CI	UA estimates (relative error in %)	95% CI
$Q_g\,(t_{122})$	318.6	407.5 (27.9%)	[279.7; 555.1]	552.8 (73.5%)	[403.1; 702.5]*
$Q_g\,(t_{158})$	385.5	399.5 (3.6%)	[269.9; 548.2]	523.6 (35.2%)	[378.2; 669.0]
$Q_r\,(t_{122})$	1319.5	1482.5 (12.35%)	[1158.0; 1659.2]	1689.2 (28.0%)	[1401.5; 1977.0]*
$Q_r\,(t_{158})$	2368.5	2260.9 (4.5%)	[1906.5; 2560.7]	2416.0 (2.0%)	[1983.4; 2848.7]

Table 9.11: Comparison of the prediction results of the LNAS model with and without data assimilation based on the 2008 dataset (density 16.4). *: the predicted credibility interval does not contain the real observed data.

	Real Data 2008	DA estimates (relative error in %)	95% CI	UA estimates (relative error in %)	95% CI
$Q_g\,(t_{122})$	297.6	351.1 (18.0%)	[276.0; 486.6]	483.0 (62.3%)	[342.1; 624.0]*
$Q_g\,(t_{158})$	408.1	387.4 (5.1%)	[268.9; 493.8]	467.5 (14.6%)	[328.5; 606.6]
$Q_r\,(t_{122})$	1551.4	1462.7 (5.7%)	[1288.8; 1632.0]	1590.7 (2.5%)	[1316.9; 1864.5]
$Q_r\,(t_{158})$	2535.0	2266.5 (10.6%)	[1946.6; 2585.3]	2285.6 (9.8%)	[1886.3; 2684.8]

Table 9.12: Comparison of the prediction results of the LNAS model with and without data assimilation based on the 2008 dataset (density 5.4). *: the predicted credibility interval does not contain the real observed data.

Tables 9.8 to 9.12 illustrate the prediction results of the two methods (with or without data assimilation) for different experiments. According to the results based on the 2006 dataset (Table 9.8) and on the 2008 dataset with observed density 10.9 (Table 9.9), the prediction results are clearly improved both in terms of the mean prediction

and the associated uncertainty assessment for nearly all the predictions given by the CPF-based approach compared to those provided by UA. All the predictions have a relative error close or inferior to 10%, which indicates that the ICPF calibration results combined with CPF-based data assimilation method result in an excellent predictive performance. Although in Table 9.9, the mean predictions of UA are already quite accurate, the standard deviations of the DA predictions are much smaller than those given by the UA. Considering the important level of observation error (≈15%), the uncertainty associated to the mean estimates is indeed crucial in yield prediction. The proposed credibility intervals all include the real value of the last two measurements and are generally narrower than those given by the UA.

On the other hand, as suggested by Figure 9.6, the 95% credibility interval provided by the UA can be considered non-reliable since it does not always contain the real measurement values in several cases (*cf.* Table 9.11, Table 9.12 and Table 9.10). These cases correspond to very different crop densities compared to the 2010 dataset used for the calibration (16.4 and 5.4 plants/m^2 respectively compared to 11.9 plants/m^2) or a different cultivar (*Harmonia* compared to *Rada*). The limitation of the simple prediction approach (performed by UA) can thus be recognized, since the prediction conditions are out of the strict range of validity of the domain. Therefore, it is important to learn that the data assimilation approach broadens the range of applicable cases and enhances the model predictive capacity.

9.2.3 Influence of the noise level

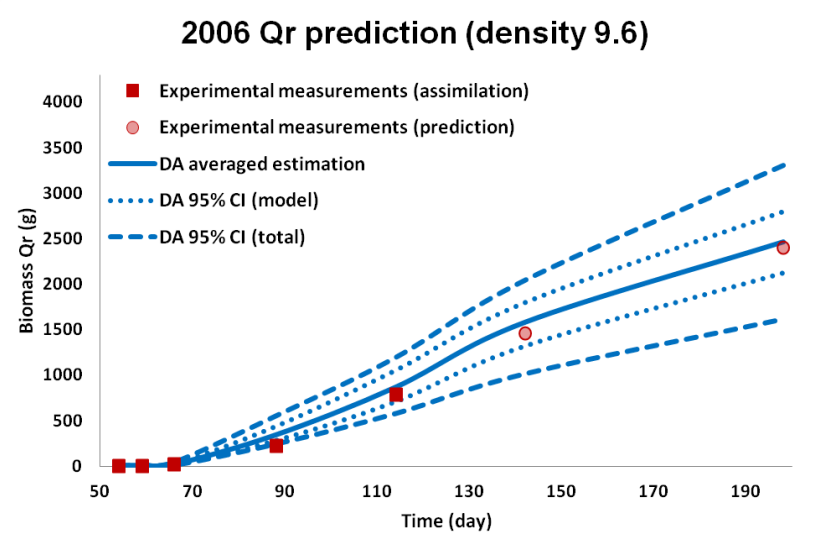


Figure 9.7: Decomposition of the uncertainties related to the prediction of Q_r in 2006 (density 9.6) performed with data assimilation based on the ICPF estimates.

We highlight that when the step 3 is performed with the RPF approach, the likelihood of the observations is considered known explicitly, thus we are able to decompose the uncertainty associated to the prediction into two parts, one related to the parame-

ter and modelling errors, and the other due to the observation errors, as demonstrated by Figure 9.7.

Undeniably, the noise related to parameters play a crucial role in the uncertainty assessment of the predictions, however, we are not certain neither if the Gaussian assumption holds for the observation likelihood function, nor if the noise level remains the same for observations made in different locations or in different years. Bearing this thought in mind, the RPF and CPF approaches for data assimilation and uncertainty propagation are compared, three observation noise levels are taken into account.

Settings :

Three assumptions for observation noise level, 0.01, 0.08 and 0.15, are tested for the data assimilation step, performed with CPF (kernel estimator, considering the likelihood function unknown) and RPF (considering the likelihood function known) with 350 000 particles.

	Small noise (1%)		Medium noise (8%)		Large noise (15%)	
DA(RPF)	relative error in %	Std.	relative error in %	Std.	relative error in %	Std.
$Q_g\,(t_{142})$	0.9%	15.8	1.2%	36.1	0.8%	80.5
$Q_g\,(t_{198})$	8.1%	15.1	6.9%	34.3	5.9%	72.1
$Q_r\,(t_{142})$	4.0%	51.4	6.6%	142.2	8.9%	205.3
$Q_r\,(t_{198})$	3.8%	58.3	4.1%	189.4	5.2%	317.9
DA(CPF)	relative error in %	Std.	relative error in %	Std.	relative error in %	Std.
$Q_g\,(t_{142})$	0.8%	15.9	2.2%	38.4	2.1%	82.4
$Q_g\,(t_{198})$	10.2%	15.3	7.0%	36.1	7.2%	75.5
$Q_r\,(t_{142})$	4.8%	52.9	12.9%	146.7	7.9%	238.2
$Q_r\,(t_{198})$	4.2%	59.8	7.3%	187.3	4.9%	344.4
UA	relative error in %	Std.	relative error in %	Std.	relative error in %	Std.
$Q_g\,(t_{142})$	44.7%	53.6	44.7%	89.8	44.3%	138.3
$Q_g\,(t_{198})$	43.5%	48.1	43.6%	74.3	43.1%	116.8
$Q_r\,(t_{142})$	32.3%	102.5	32.3%	193.8	32.0%	354.2
$Q_r\,(t_{198})$	20.1%	136.2	20.2%	282.1	20.2%	522.3

Table 9.13: Comparison of model prediction errors and predicted standard deviation (Std.) with and without data assimilation for the LNAS model based on the 2006 dataset (density 9.6) according to different assumptions of noise levels.

Observations and remarks :

According to our tests (Table 9.13), the uncertainty associated to the prediction results is quite sensitive to the level of observation noises compared to the mean predicted values. The prediction results obtained with Gaussian assumptions appear to be closer to the real measurements and more stable when the noise level assumption changes, which suggests a small advantage of the RPF approach.

Another meaningful thought is that since the level of noise parameter evaluated on one year may not represent properly the corresponding level in another year or in

a different location, applying the same level of noises in the assimilation step with a different dataset is debatable. Further studies are hence needed to address the influence of the observation noise level to the prediction quality.

We remark that in the LNAS model, we opt for a multiplicative noise model while additive models and log normal models are also widely used and could also be tested. Preliminary tests have suggested better performances of the multiplicative noise model compared to the additive one in the calibration step. However, further investigation is still required.

9.2.4 Impact of the calibration precision

Prediction performance of calibrations performed by IEnKF, IRPF, IUKF and IAMwG

Regarding the prediction based on the iterative approaches, as illustrated by Table 9.14, the best performance in the evaluation of the log-likelihood was achieved by the IRPF estimates. The same conclusion holds for the AICc and the BIC evaluations as well. In the following test, the goal is to identify the difference of the predictive performance only due to the estimation precision in the calibration step (prior information of the assimilation step).

In this first study case, we choose the RPF as the only algorithm to perform the assimilation step. The prediction performed by EnKF and UKF are further compared in Section 9.2.5. Another thought-provoking comparison between the prior information given by the RPF and the IRPF algorithm with the same configuration in the context of data assimilation problems can be found in Chen et al. (2013a).

Settings :

Based on the 2006 experimental dataset (density 9.6) and the 2008 experimental dataset (density 10.9), the calibration carried out by the iterative approaches are compared with regards to their prediction capacity. RPF with 350 000 particles is used to perform the data assimilation and the prediction. The uncertainty analysis is also conducted to illustrate the predictive performance of the raw estimates without the assimilation of data obtained at early growth stages. The Root Mean Squared Error for Prediction (RMSEP) is used as the criterion to evaluate the quality of prediction.

Observations and remarks :

Generally speaking the assimilation step has well established its value as demonstrated by Table 9.14 compared to the prediction without assimilation (UA). The estimation relative error was greatly reduced when data assimilation was performed. In the meantime, the uncertainties (standard errors) associated to the prediction were also significantly decreased in all cases with the assimilation of data from the early growth stages. IAMwG and IRPF estimates resulted in similar estimations, with IAMwG a slightly better prediction for Q_r and IRPF a small advantage in Q_g prediction.

Estimation methods		IRPF		IEnKF		IUKF		IAMwG	
		DA	UA	DA	UA	DA	UA	DA	UA
$Q_g(t_{142})$	Relative error	0.7%	44.5%	3.2%	45.8%	5.7%	55.6%	0.8%	45.1%
	Std.	80.5	128.7	80.8	163.7	89.7	165.0	82.7	162.8
$Q_g(t_{198})$	Relative error	5.9%	43.4%	9.8%	46.5%	12.1%	58.5%	6.1%	43.2%
	Std.	72.1	116.8	75.1	141.2	94.3	144.2	77.4	149.7
$Q_r(t_{142})$	Relative error	8.5%	32.2%	7.8%	29.8%	9.3%	43.1%	8.0%	31.7%
	Std.	205.3	354.3	208.0	384.3	259.4	479.2	192.8	367.1
$Q_r(t_{198})$	Relative error	2.9%	20.6%	2.0%	18.6%	4.0%	29.8%	2.6%	20.3%
	Std.	317.9	522.3	321.4	553.4	402.7	693.6	296.9	539.1

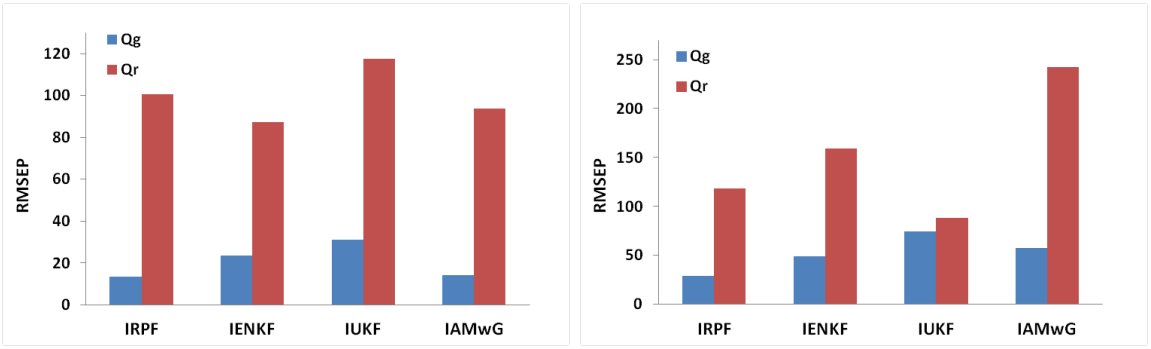
Table 9.14: Comparison of model prediction capacity based on the estimates provided by four iterative methods (IEnKF, IRPF, IUKF and IAMwG) performed with the 2006 dataset (density 9.6). DA: with assimilation of data at the early growth stages, UA: uncertainty analysis without data assimilation.

	Real Data 2008	IEnKF estimates (relative error in %)	Std.	IRPF estimates (relative error in %)	Std.
$Q_g(t_{122})$	373.5	419.5 (12.3%)	68.9	403.6 (8.1%)	59.2
$Q_g(t_{158})$	380.6	433.1 (13.8%)	73.4	408.7 (7.4%)	71.8
$Q_r(t_{122})$	1559.1	1460.5 (6.3%)	197.5	1466.7 (5.9%)	210.7
$Q_r(t_{158})$	2327.7	2125.6 (8.7%)	324.1	2187.6 (6.0%)	326.2
	Real Data 2008	IUKF estimates (relative error in %)	Std.	IAMwG estimates (relative error in %)	Std.
$Q_g(t_{122})$	373.5	440.9 (18.1%)	80.4	423.8 (13.5%)	79.5
$Q_g(t_{158})$	380.6	461.1 (21.2%)	91.2	444.1 (16.7%)	90.0
$Q_r(t_{122})$	1559.1	1508.8 (3.2%)	235.2	1391.2 (10.8%)	220.5
$Q_r(t_{158})$	2327.7	2213.7 (4.9%)	361.4	2028.4 (12.9%)	344.2

Table 9.15: Comparison of model prediction capacity of estimates provided by four iterative methods (IEnKF, IRPF, IUKF and IAMwG) based on the 2008 dataset with assimilation of data at the early growth stages.

For the green leaf biomass allocation Q_g, it has always been the IRPF estimates that led to the best predictions for both years, as indicated by Table 9.15 and Table 9.14. This may be related to the important nonlinearity of its evolution.

IUKF provided the best 2008 root prediction as indicated by Figure 9.8. A possible explanation lies on the fact that the root biomass allocation is more linear than the green leaf biomass allocation according to Figure 9.9, Figure 9.10 and Figure 9.11. This may also explain the fact that the prediction based on the IUKF estimates for the green leaf biomass are less accurate compared to the results given by the other three methods for the two years. The limited sample size of sigma-points could still have some difficulty when facing relatively important nonlinearity with multidimensional estimations required.



(a) 2006 dataset (density 9.6) (b) 2008 dataset (density 10.9)

Figure 9.8: Comparison of model prediction capacity of estimates provided by four iterative methods (IEnKF, IRPF, IUKF and IAMwG) evaluated by Root Mean Squared Error of Prediction (RMSEP) based on the 2006 and 2008 datasets with assimilation of data at the early growth stages.

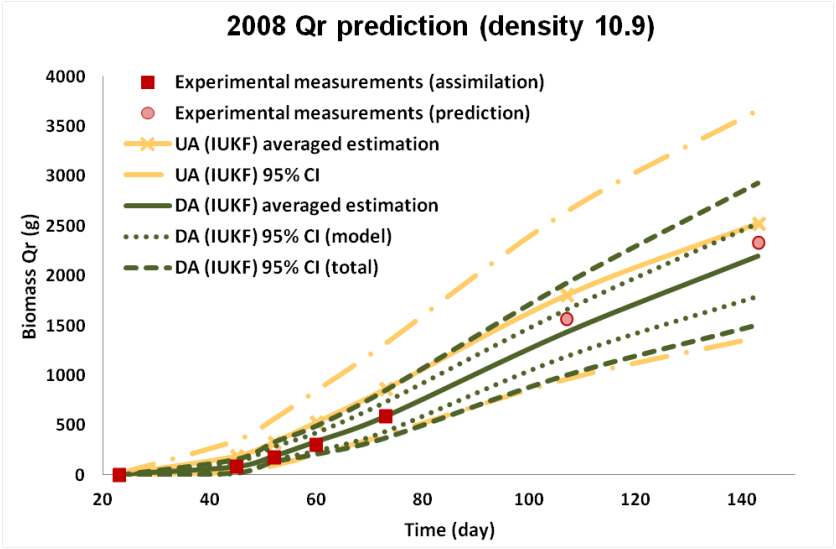


Figure 9.9: Comparison of the predictions of Q_r in 2008 (density 10.9) performed with or without data assimilation (Uncertainty Analysis) based on IUKF estimates.

However, it is not always the case for the 2006 root biomass prediction. Although the IEnKF estimates displayed the best performance, we notice that the performance of the four sets of predictions were relatively close, as suggested by Table 9.14.

Therefore in general, the most accurate predictions are those obtained based on the IRPF estimates (in the calibration step), if we disregard the computation time and the memory requirements.

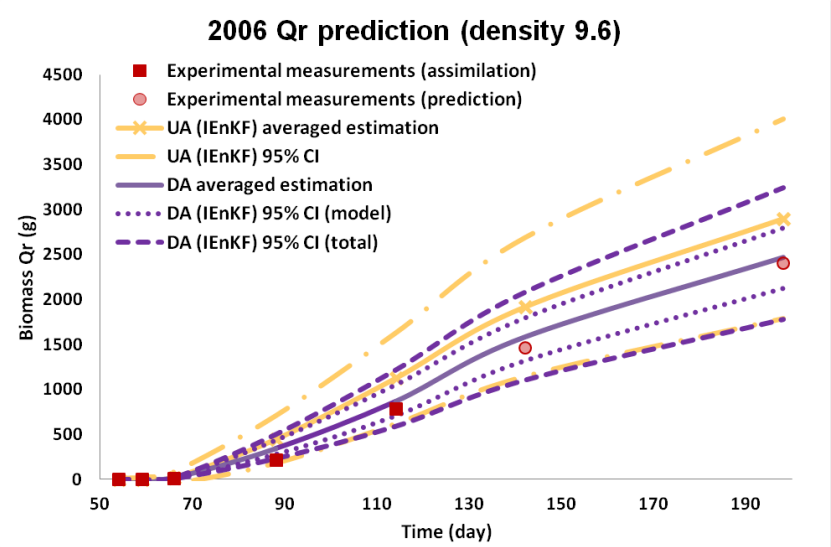


Figure 9.10: Comparison of the predictions of Q_r in 2006 (density 9.6) performed with or without data assimilation (Uncertainty Analysis) based on IEnKF estimates.

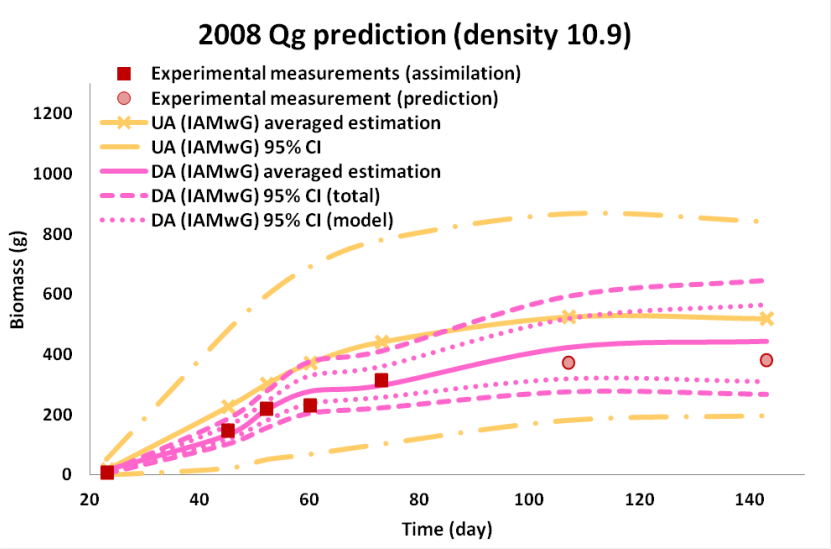


Figure 9.11: Comparison of the predictions of Q_g in 2008 (density 10.9) performed with or without data assimilation (Uncertainty Analysis) based on IAMwG estimates.

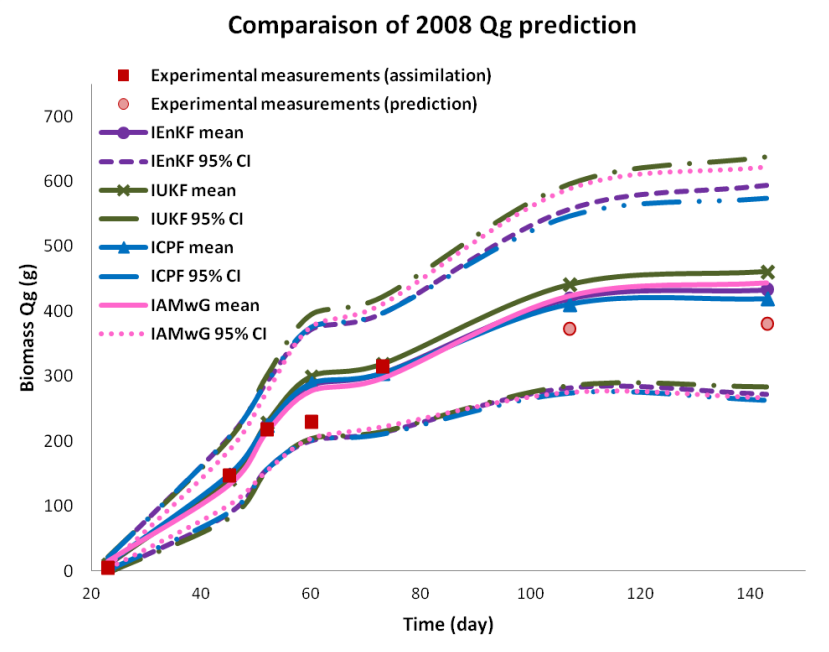


Figure 9.12: Comparison of the predictions of Q_g in 2008 performed with data assimilation based on IEnKF, IUKF, ICPF and IAMwG estimates.

Note that the UKF and the EnKF algorithms can also be adapted to perform the data assimilation step. Therefore, the data assimilation step is also carried out with these Kalman filter based methods in the following study case.

9.2.5 Data assimilation with UKF and EnKF

As mentioned previously, other SMC filtering methods than CPF/RPF can also be used to perform the model prediction by propagating the uncertainty and re-adjusting the system on-line. In this test, we aim to compare the predictive capacity of UKF, EnKF and RPF with the calibration performed based on their iterative version with the 2010 dataset.

Settings :

EnKF is performed in the data assimilation step with an ensemble size of 350 000 based on the IEnKF calibration. The prediction with UKF is conducted with the unscented transform (21 sigma-points) based on the IUKF calibration. Their predictive performances are evaluated by both the relative error and the associated prediction uncertainty, and are compared to the RPF approach using 350 000 particles with the IRPF calibration. The experimental dataset of 2006 is used for the prediction test.

	Real Data 2006	EnKF (RE in %)	Std.	RPF (RE in %)	Std.	UKF (RE in %)	Std.
$Q_g\,(t_{142})$	355.2	396.4 (11.6%)	83.4	357.2 (0.7%)	80.5	363.5 (2.3%)	99.7
$Q_g\,(t_{198})$	320.6	370.6 (16.6%)	78.4	339.4 (5.9%)	72.1	328.4 (2.4%)	98.3
$Q_r\,(t_{142})$	1459.2	1459.2 (2.9%)	203.2	1583.8 (8.5%)	205.3	2002.1 (37.2%)	270.0
$Q_r\,(t_{198})$	2400.0	2312.5 (3.6%)	321.4	2469.5 (2.9%)	317.9	2499.7 (4.2%)	438.9

Table 9.16: Comparison of model prediction capacity of three filtering methods (IEnKF, IRPF and IUKF) using the estimates obtained with their respective iterative approaches based on the 2006 dataset with data assimilation. Relative error (RE) and standard deviations (Std.) for the prediction are given.

Observations and remarks :

According to Table 9.16, all the three SMC methods, EnKF, RPF and UKF were used to perform the prediction (data assimilation) step based on the estimation results of their respective iterative approaches. We remark that the confidence intervals given by UKF are the widest, so are the prediction error for the root compartment Q_r. The EnKF and the RPF estimates are comparable, especially for Q_r prediction. However, the IUKF estimates seem to be improved for Q_g prediction compared to Table 9.14 while the IEnKF estimates appear to be worse. In all cases, their uncertainty associated to the predictions are increased compared to the DA step performed by RPF. The result thus suggests that RPF is more appropriate to perform the assimilation step, since it is not constrained by the Gaussian assumption as the Kalman-based the filtering methods are.

9.2.6 Influence of the number of assimilated data

As argued previously, the experimental data are obtained with heavy protocols which are extremely costly. Therefore, in applications, we should seek for the best compromise between the cost of the data and the prediction quality. The tests presented in this section can be regarded as a first step towards the optimization of experimental design to improve the predictive capacities of a model.

Settings :

In the following test, based on the 2006 experimental data, we attempt to eliminate some of the observation dates, and evaluate the impact on the prediction accuracy. The last two dates t_{122} and t_{158} are used for prediction. The impact of the remaining five dates on the prediction results is investigated. In Table 9.17, the observation dates that are available for data assimilation among the total five are marked as "o", while those that are eliminated are marked as "$-$".

For all prediction tests, the DA step is carried out with RPF (350 000 particles) based on the 2010 IRPF estimates obtained in the calibration step.

DA	5		4		4		4		4		4	
dates	(t_{54-114})		(t_{54-88})		(t_{59-114})		$(t_{54}, t_{59}, t_{88}, t_{114})$		(t_{54-66}, t_{114})		$(t_{54}, t_{66}, t_{88}, t_{114})$	
	ooooo		oooo-		-oooo		oo-oo		ooo-o		o-ooo	
	RE	Std	RE	Std	RE	Std	RE	Std	RE	Std	RE	Std
$Q_g\,(t_{122})$	0.7%	80.5	2.6%	119.6	4.4%	82.4	2.5%	82.3	0.3%	85.3	2.9%	82.4
$Q_g\,(t_{158})$	5.9%	72.1	2.9%	108.3	12.2%	75.2	8.6%	75.3	4.0%	79.7	9.1%	75.6
$Q_r\,(t_{122})$	8.5%	205.3	11.0%	258.8	7.8%	206.5	8.7%	207.3	8.9%	208.3	8.7%	211.2
$Q_r\,(t_{158})$	2.9%	317.9	4.2%	376.1	3.6%	320.1	2.0%	322.2	2.6%	322.2	2.5%	330.6
	3		3		3		2		2		2	
	(t_{54-66})		(t_{66-114})		$(t_{54}, t_{66}, t_{114})$		(t_{54}, t_{59})		(t_{88}, t_{114})		(t_{59}, t_{88})	
	ooo- -		- -ooo		o-o-o		oo- - -		- - -oo		-o-o-	
	RE	Std	RE	Std	RE	Std	RE	Std	RE	Std	RE	Std
$Q_g\,(t_{122})$	24.9%	142.4	0.8%	83.3	0.2%	85.5	29.4%	149.2	4.0%	83.0	1.6%	107.8
$Q_g\,(t_{158})$	27.6%	118.8	5.8%	76.7	4.1%	80.1	32.0%	124.9	10.1%	76.5	3.3%	94.9
$Q_r\,(t_{122})$	19.2%	294.2	8.7%	211.2	8.8%	211.8	20.9%	288.6	8.8%	214.9	9.7%	254.1
$Q_r\,(t_{158})$	10.1%	414.2	3.2%	331.2	2.5%	330.9	11.1%	406.0	1.9%	340.5	2.4%	375.8
	1		1									
	(t_{114})		(t_{88})									
	- - - -o		- - -o-									
	Error	Std	Error	Std								
$Q_g\,(t_{122})$	3.6%	84.0	17.6%	110.6								
$Q_g\,(t_{158})$	9.3%	77.9	24.9%	99.6								
$Q_r\,(t_{122})$	9.2%	216.9	9.1%	254.3								
$Q_r\,(t_{158})$	2.9%	343.7	3.0%	378.4								

Table 9.17: Comparison of the relative errors (RE) and standard deviations corresponding to the prediction uncertainty resulting from the data assimilation step performed with different numbers of data used for assimilation by RPF, based on the IRPF estimates with the 2006 experimental dataset. On each observation date, two measurements Q_g and Q_r are obtained.

Observations and remarks :

Table 9.17 illustrates the point prediction quality resulting from different combinations of observation dates. As expected, the result suggests that the most important data to enhance the prediction accuracy is the one closest to the prediction. When observation dates are close, the data appear to be less useful: we can see that eliminating one observation date between t_{54} or t_{59} did not influence much the prediction results. The root compartment Q_r prediction does not vary much among the various configurations, while the green leaf compartment Q_g prediction is more sensitive to the lack of certain data. This is probably due to the nonlinearity of the Q_g evolution. If only the observations of the first two dates are available (t_{54} and t_{59}), which are very close, we are at risk of over-correcting the evolution curve. Taking the last two observations dates (t_{88} and t_{114}) seems to be the optimal choice.

Likewise, the impact of different observation dates on the uncertainty assessment varies. When the observation date is close to the prediction date, the uncertainty associated to the prediction is reduced in a more significant way. However, this result may be contextual. Further tests should be conducted.

Another key issue is the type of data to observe. Here we only considered the green leaf biomass Q_g and root compartment biomass Q_r, however, one can imagine using data obtained by drone or satellite images (Baret et al., 2007). The observations such as the LAI can be derived with image processing methods. Indeed, remote sensing

images tend to be less costly than the experimental measurements, which could be considered as a very attractive source of information despite the important observation error. Therefore, this research subject is worth further investigations.

9.2.7 Influence of the number of updated parameters

In this test, we try to decrease the number of parameters to update in the prediction step to see its influence on the prediction performance. Note that the prediction tests are conducted based on the six functional parameters estimated in step 2 by IRPF.

Settings :

The functional parameters are eliminated according to the order given by the parameter ranking of the sensitivity analysis. Those parameters that are calibrated but not selected for the re-calibration are propagated as in the case of UA (distributions characterized by the mean value and the standard deviation). The data assimilation is performed by RPF with the IRPF estimations given in step 2. 350 000 particles are used by RPF for prediction purpose.

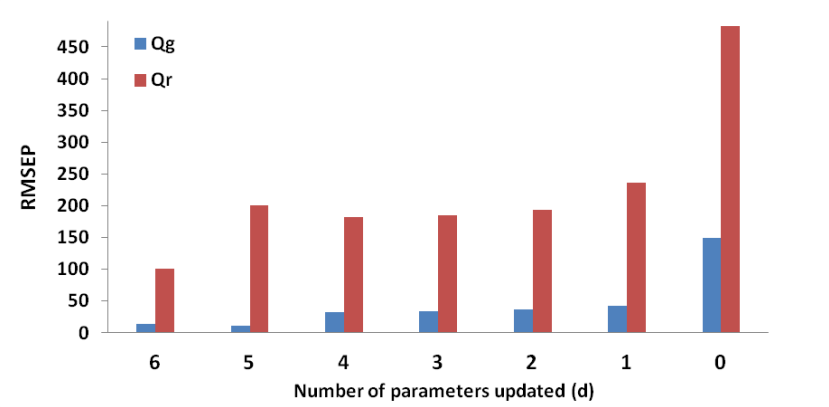


Figure 9.13: Comparison of model prediction capacity of IRPF estimates with different numbers of parameters updated by RPF, evaluated by Root Mean Squared Error of Prediction (RMSEP) based on the 2006 datasets with and without ($d = 0$) assimilation of data at the early growth stages.

d	6		5		4		3		2		1		0 (UA)	
	RE	Std	RE	Std	RE	Std	RE	Std	RE	Std	RE	Std	RE	Std
$Q_g\,(t_{122})$	0.7%	80.5	0.4 %	96.9	5.7%	100.2	6.8%	101.5	7.7%	102.8	9.3%	103.1	44.5%	128.7
$Q_g\,(t_{158})$	5.9%	72.1	4.8 %	105.2	12.8%	98.4	12.9%	97.6	13.5%	98.9	15.4%	99.6	43.4%	116.8
$Q_r\,(t_{122})$	8.5%	205.3	12.2 %	241.5	12.5%	251.3	12.7%	252.5	12.9%	252.1	14.6%	253.4	32.2%	354.3
$Q_r\,(t_{158})$	2.9%	317.9	9.2 %	350.2	7.6%	367.8	7.7%	368.0	8.3%	370.4	10.7%	371.1	20.6%	522.3

Table 9.18: 2006 experimental dataset: comparison of the relative errors (RE) and standard deviations resulting from the data assimilation step performed with different numbers of parameters updated by RPF, based on the IRPF estimates of the calibration step.

Observations and remarks :

Table 9.18 and Figure 9.13 indicate that it is important to recalibrate all the six functional parameters, for right from the configuration with five parameters (with only σ_γ eliminated), the prediction of Q_r became significantly less accurate (even though the prediction of some variables may be improved). Between the configurations with 3 and 4 parameters recalibrated, the prediction relative errors are very close, while in the following when more parameters are eliminated, the prediction quality is largely deteriorated. This result somehow coincides with the result of the sensitivity analysis (Figure 9.1) for the parameter 4 and 3 (λ and μ_γ resp.) have similar impacts. The same remark is made for the parameter 6 and 5 (σ_γ and γ_f resp.) as well.

Note that unlike the results given by the Bayesian approach (*cf* Table 9.6 and Table 9.7), the uncertainty related to the prediction was not greatly influenced by the number of updated parameters, for the uncertainty associated with those parameters are still taken into account and propagated in the prediction while in the Bayesian case, the non-selected parameters are fixed at their recommended values.

Likewise, this result could be contextual.

9.2.8 Discussion

The three-step data assimilation approach proposed in this thesis generally appears to be robust, which allows us to improve the prediction accuracy and to reduce the associated uncertainty.

However, when the Bayesian methods were used in the parameter estimation step (step 2), the model configuration (in terms of the number of parameters to estimate) selected by the DIC based on the learning dataset proved not to be the best one regarding the predictive capacities in the assimilation step (step 3). Therefore, the model selection result can be contextual or the DIC may not be the most adapted selection criterion for prediction perspective. It is a risk to take when applying such an approach, since in real application, the prediction results are unknown. It seems that by increasing slightly the number of parameters suggested by the DIC in the estimation step, or rather by using a more tolerant criteria, the quality of the prediction results obtained with data assimilation could be improved.

As for the prediction results based on the iterative calibration, the predicted credibility intervals provided by the RPF-based approach are nearly in all cases narrower than those of the UA. It is of course an expected result for data assimilation: to reduce the prediction uncertainty based on the available information. However, regarding the point estimations, the results of the CPF-based method were not always more accurate than those provided by UA, especially for the root mass, as shown in Table 9.15 for $Q_r(t_{158})$ or in Table 9.12 for $Q_r(t_{122})$ or $Q_r(t_{158})$. However, when it happened, the corresponding predictions of leaf biomass given by the UA were nonetheless far from the real observed values and can be considered as unreliable. This may reveal some particular plasticity in root biomass production that was not well captured by the model, which can be regarded as a drawback of our simplified model LNAS. However, even in the few cases when the proposed method failed to improve the point prediction

of some variables, it always managed to provide reasonable credibility intervals which contained the real values of the observations.

Concerning the prediction results based on the 2008 dataset with the cultivar *Harmonia* (cf Table 9.10), there is an important improvement of the predictive performance with data assimilation compared to the case without assimilation (UA), for which the predicted credibility intervals for 3 variables out of 4 do not even contain the real values of the measurements. However, the relative errors with data assimilation are all around 15% to 20% for the last measurement date, which is not satisfactory for practical applications. This may be linked to the fact that the prior information resulting from the calibration step is not appropriate, it represents the level of uncertainty of the parameters of a specific cultivar, in a given environment. If the adaptation to other environmental conditions via data assimilation seems to work rather well, the adaptation to a different cultivar appears more difficult, at least in the case when the environmental conditions are also different. However, it is still reassuring to note that the real values of the last two observations are still in the computed credibility intervals.

Regarding the uncertainty assessment, when the estimated modelling noises are quite small (around 2%), it usually implies a good adaptation of the model, whereas when it is above 5%, the prediction gain with data assimilation could be limited. When the observation noises are important, it indicates that the observations may not be reliable enough to update parameters and state variables. Thus in the assimilation step they may also prevent the algorithm to retrieve the most useful information out of the few available data, especially when we consider the likelihood function cannot be computed explicitly (CPF).

By keeping in mind the importance of the uncertainty assessment for the predictive capacity of the data assimilation method, some numerical tests were performed using artificially different levels of observation noise in place of the ones estimated from the 2010 dataset. A comparison of prediction capacity linked to three different levels of observation noises (1%, 8% and 15%) was carried out. UA provided close point estimations in all the three cases while the width of the 95% credibility interval increased with the observation noises. However, the level of the noises influenced significantly the point estimations given by the CPF-based method. For the last two measurements regarding the biomass of root compartment, the prediction quality was greatly improved with the decrease of the observation noises, while for the prediction of leaf biomass, it is the reverse, the prediction may even be slightly deteriorated on account of the small noises.

Hence, given the sensitivity of the filtering-based data assimilation methods with regards to the observation noises, their proper evaluation is crucial to improve the accuracy and reliability of the method, especially regarding the uncertainty associated to the prediction. It is clearly a bottleneck for real applications since the observation noises in practice tend to vary a lot according to the experimental configurations (for example in two different years or in two different fields), so that their proper evaluation may remain quite difficult.

In this study case, the three SMC methods were used to carry out the prediction and data assimilation step. RPF showed a small advantage compared to EnKF. The

estimates achieved by IRPF, IEnKf, IUKF and IAMwG in the calibration step were also compared. Overall, the IRPF estimation provided the most accurate predictions with narrowest confidence intervals. Nonetheless, the IEnKF approach despite the normality assumption which led to slightly worse prediction results, sometimes provides similar or even more accurate predictions. By taking into account the computational cost, the use of IEnKF in the calibration step could be preferable, unless the nonlinearity is remarkable, the Gaussian assumption does not hold or more precise calibration results are required. In these cases, the ICPF approach is more suitable.

Regarding the time-consuming problem related to an iterative use of the particle filtering methods, it might also be interesting to be less strict on the model parameter estimation and loosen the convergence criterion so as to reduce the iteration numbers. Given the fact that the parameters will be further adjusted in the assimilation step, the importance of the first calibration step might be over-evaluated.

Finally, towards an optimized design of experiments, a few tests have been conducted to study the effect of the number of observation data and of the selection of the observation dates on the prediction accuracy. The results in Table 9.17 suggest that it is not evident that more observations lead to more precise predictions. Irregular and non-dispersed observation dates usually result in poor predictions except when the observation dates are close enough to the prediction dates. Consequently, the collection of new data should be guided by targeting the optimal conditions. Similar results have been obtained by Varella et al. (2010). The strategy proposed is that ideally, the observation dates should be evenly placed, with the preference of being close to the prediction dates. It is obvious that the latest observation weighs more and is thus more valuable, however the objective remains to provide accurate yield predictions with less new data and as early as possible. Satellite images (remote sensing) although often imprecise may help to follow closely the evolution of LAI, and this could be a useful and efficient way to update the prediction so as to improve its accuracy and reduce the associated uncertainties (Bouman, 1992; Delécolle et al., 1992; Guérif and Duke, 1998, 2000; Dorigo et al., 2007; Baret et al., 2007; Dente et al., 2008).

9.3 Application to the STICS model

9.3.1 Experimental data

The data used for this study were obtained by INRA (Institut National de Recherche Agronomique) in the context of the Aquateam project (http://www.projet-aquateam.org/) whose objective is to develop decision aid tools for crop irrigation. The experiments were carried out at Villamblain (France). The growth of two commercial varieties of winter wheat were monitored: *Raffy* in two experimental campaigns, 2011-2012 (sowing date: 25 October 2011; harvesting date: 25 July 2012) and 2012-2013 (sowing date: 29 October 2012; harvesting date: 30 July 2013) and *Numeric* in one experimental campaign, 2012-2013 (sowing date: 29 October 2012; harvesting date: 30 July 2013). In our study, the 2012 dataset of variety *Raffy* was used for parameter estimation. For the sake of clarity for the readers already

familiar with the STICS model, we use the classical notations for state variables and parameters recalled extensively in Brisson et al. (2008). Dry matter of green leaves (denoted *mafv*, g·m^{-2}), above-ground dry matter (denoted *masec*, g·m^{-2}), soil averaged water content (denoted *hur*, mm/unit area) and dry matter of grain yield (denoted *magrain*, g·m^{-2}) were measured and collected at different dates in 2012 (in days after sowing):

$$\begin{aligned}\mathcal{O}_{2012}^{mafv} &= \{155, 185\}\,,\\ \mathcal{O}_{2012}^{magrain} &= \{269\}\,,\\ \mathcal{O}_{2012}^{masec} &= \{155, 185, 213, 239, 269\}\,,\\ \mathcal{O}_{2012}^{hur} &= \{155, 171, 178, 192, 203, 219, 234, 247, 260\}\,.\end{aligned}$$

For data assimilation and prediction, the two 2013 datasets corresponding to both varieties were used. Additional measurements of LAI (denoted *LAI*, m^2 leaf ·m^{-2} soil) were available for the 2013 dataset (obtained with the SunScan Canopy Analysis System of Delta-T Devices), while the dry matter of green leaves was not measured. The observation dates are given as follows:

$$\begin{aligned}\mathcal{O}_{2013}^{hur} &= \{21, 79, 114, 155, 170, 182, 198, 210, 226, 240, 255, 275\}\,,\\ \mathcal{O}_{2013}^{magrain} &= \{266\}\,,\\ \mathcal{O}_{2013}^{LAI} &= \{162, 175, 191, 203, 212, 219, 233\}\,.\end{aligned}$$

Daily mean values of air temperature, solar radiation, potential evapotranspiration, and total daily rainfall were obtained from French meteorological advisory services (Météo France) 3 km away from the experimental site.

9.3.2 Results and method comparison

Parameter Screening by Sensitivity Analysis

Since STICS has a large number of parameters, and regarding the reduced experimental data sets that are available for its parameterization, a preliminary parameter selection was conducted in the first place. 16 variety-dependent parameters that are relevant to the concerned model state equations were selected to perform the sensitivity analysis, for the output chosen as a generalized least-square criterion based on the 2012 dataset.

According to Sobol total order indices, the four most influential parameters, VLAIMAXP, EFFICIENCE, STAMFLAXV and UDLAIMAXP were selected for the next calibration step. Their definitions are given by Table 9.19. The other parameters (with total order indices below 0.02) were hence fixed to the mean values of their variation intervals (deduced from literature (Brisson et al., 2008)).

As illustrated by Figure 9.14, the difference between the first order and the total order indices is little, which indicates a high linearity of the system (Campolongo et al., 2007).

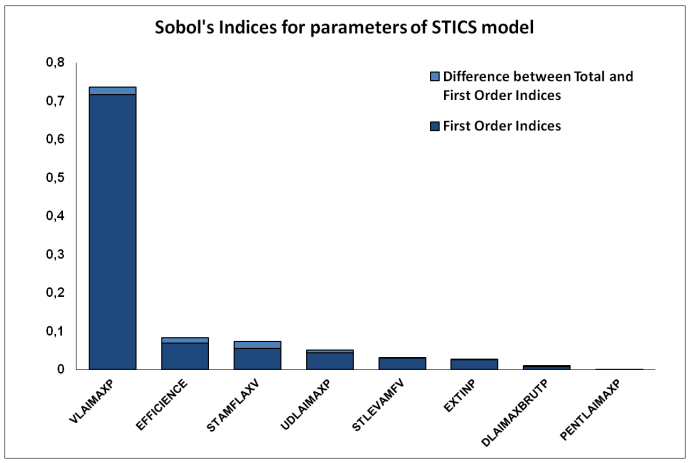


Figure 9.14: First and total order indices for parameters of STICS wheat crop model.

EFFICIENCE: maximum radiation use efficiency (g·MJ^{-1})
STAMFLAXV: duration between the end of the juvenile phase and the maximal of leaf area index (deg C.days)
UDLAIMAXP: maximal daily relative development of LAI (no unit)
VLAIMAXP: daily relative development of LAI at the inflexion point (no unit)

Table 9.19: Definition of the selected parameters for the calibration of STICS.

Parameter Estimation

In the deterministic case, the conditional ICPF approach coupled with the parametric bootstrap used for the LNAS model is not possible. We thus estimated the unknown parameter vector

$$\Theta = (EFFICIENCE, STAMFLAXV, UDLAIMAXP, VLAIMAXP)$$

with the classical CPF. 100000 particles were drawn from the distributions used for sensitivity analysis and obtained from the literature (Brisson et al., 2008). Means and standard deviations of all the four parameters are thus obtained based on the posterior distribution provided by the population of the particles and their associated weights. The results are presented in Table 9.20.

Data Assimilation with CPF

Data assimilation was subsequently performed using as prior distributions the posterior distributions of the parameters provided by the estimation step. 100000 particles were simulated for the two 2013 datasets. The recalibration was carried out based on the first seven observations of soil water content *hur* and the first three observations of *LAI*, corresponding for both cases to data obtained before Day 199 after sowing. The values of these state variables along with the grain yield were then simulated for

Parameter	Prior Distribution	CPF Estimates	Std.
EFFICIENCE	$\mathcal{N}(4.00, 0.20^2)$	3.62	0.19
STAMFLAXV	$\mathcal{U}(250, 350)$	277.05	35.88
UDLAIMAXP	$\mathcal{N}(3.00, 0.5^2)$	2.89	0.67
VLAIMAXP	$\mathcal{N}(2.20, 0.2^2)$	1.91	0.14

Table 9.20: Estimated values and approximated standard deviations obtained by CPF for the 4 selected parameters of the STICS winter wheat model, based on the 2012 experimental data.

all particles until the end of the growing season (Day 275 after sowing), which allows us to build the posterior distribution of the prediction.

Uncertainty Analysis (UA) was also performed in this study to provide reference values for the prediction with 100000 particles as well initialized in the same way as for the prediction with assimilation.

Figure 9.15 and Figure 9.16 illustrate the model prediction for the LAI variable, with and without data assimilation. The assimilation step has clearly enhanced the predictive capacity of the model. The results are detailed in Table 9.21 for the variety *Raffy* and Table 9.22 for the variety *Numeric*: the predictions and relative prediction errors as well as the prediction uncertainty (given by the credibility intervals) are given for the soil water content *hur*, *LAI*, and grain yield *magrain*, with and without data assimilation. The relative error of prediction was greatly reduced when taking advantage of the early data by assimilation, and the prediction uncertainty was also significantly decreased in almost all the cases.

	Real Data 2013	DA estimates (RE in %)	95% CI	UA estimates (RE in %)	95% CI
$hur\,(t_{210})$	0.296	0.311 (5.2%)	[0.298; 0.324]	0.308 (4.0%)	[0.304; 0.312]
$hur\,(t_{226})$	0.296	0.305 (3.2%)	[0.292; 0.318]	0.303 (2.3%)	[0.296; 0.310]
$hur\,(t_{240})$	0.293	0.304 (3.7%)	[0.291; 0.317]	0.301 (2.8%)	[0.294; 0.308]
$hur\,(t_{255})$	0.247	0.266 (7.8%)	[0.253; 0.279]	0.273 (10.6%)	[0.255; 0.292]
$hur\,(t_{275})$	0.252	0.253 (0.4%)	[0.238; 0.268]	0.268 (6.5%)	[0.233; 0.304]
$LAI\,(t_{203})$	3.51	2.91 (17.2%)	[2.16; 3.65]	2.04 (41.9%)	[0.27; 3.81]
$LAI\,(t_{212})$	4.39	3.87 (11.9%)	[3.10; 4.69]	2.25 (49.8%)	[0.00; 4.51]
$LAI\,(t_{219})$	4.37	4.06 (7.2%)	[3.24; 4.87]	2.33 (46.7%)	[0.00; 5.06]
$LAI\,(t_{233})$	5.21	5.00 (4.0%)	[3.65; 6.35]	2.48 (52.4%)	[0.00; 6.56]
$magrain\,(t_{266})$	829.93	769.70 (7.3%)	[613.18; 926.22]	449.11 (45.9%)	[41.08; 857.14]

Table 9.21: Comparison of the prediction results of the STICS model with and without data assimilation performed with CPF calibration for the 2013 dataset variety *Raffy*. RE: relative error.

9.4 Discussion

Similar to the application of the LNAS model for sugar beet, the mean prediction and prediction credibility intervals are generally greatly improved with data assimila-

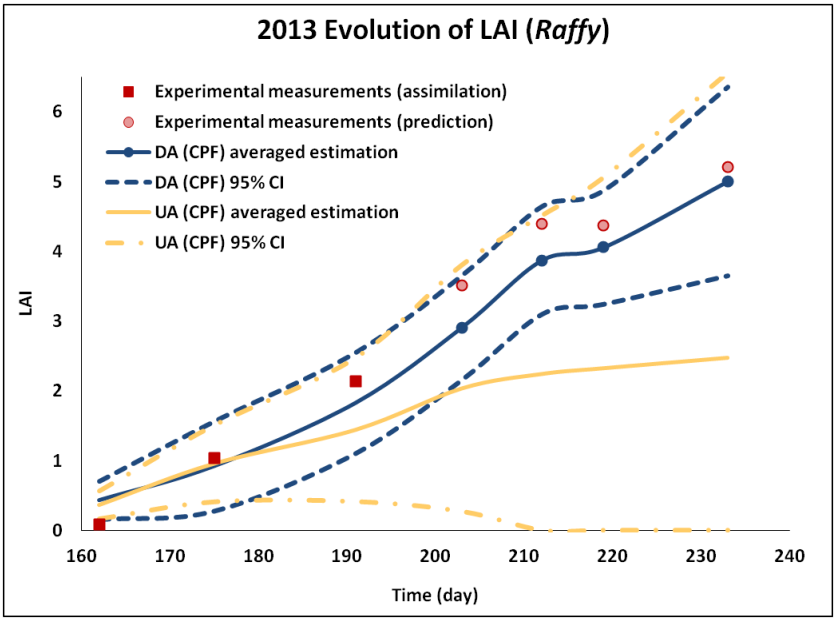


Figure 9.15: Comparison of the predictions of LAI obtained with CPF-based assimilation and by the uncertainty analysis (without assimilation) based on the 2013 dataset for variety *Raffy*. The red squares correspond to the assimilated experimental data while the pink squares represent the data used for validation.

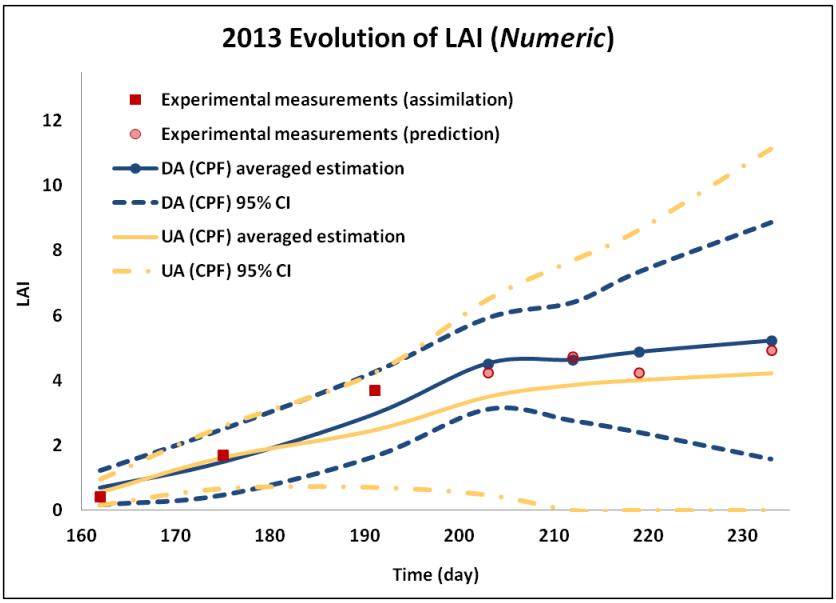


Figure 9.16: Comparison of the predictions of LAI obtained with CPF-based assimilation and by the uncertainty analysis (without assimilation) based on the 2013 dataset for variety *Numeric*. The red squares correspond to the assimilated experimental data while the pink squares represent the data used for validation.

tion compared to simple prediction by uncertainty analysis.

A noteworthy point concerns the special climatic conditions of year 2013 for the

	Real Data 2013	DA estimates (RE in %)	95% CI	UA estimates (RE in %)	95% CI
$hur\,(t_{198})$	0.276	0.272 (1.7%)	[0.260; 0.283]	0.290 (5.0%)	[0.286; 0.295]
$hur\,(t_{210})$	0.274	0.303 (10.3%)	[0.284; 0.322]	0.308 (12.2%)	[0.303; 0.313]
$hur\,(t_{226})$	0.289	0.297 (2.6%)	[0.278; 0.316]	0.303 (4.7%)	[0.295; 0.310]
$hur\,(t_{255})$	0.239	0.262 (9.7%)	[0.240; 0.283]	0.272 (14.0%)	[0.253; 0.291]
$hur\,(t_{275})$	0.257	0.260 (1.2%)	[0.229; 0.291]	0.267 (4.0%)	[0.233; 0.302]
$LAI\,(t_{203})$	4.24	4.51 (6.6%)	[3.10; 5.92]	3.5 (17.8%)	[0.47; 6.49]
$LAI\,(t_{212})$	4.74	4.63 (2.3%)	[2.75; 6.40]	3.84 (18.9%)	[0.00; 7.69]
$LAI\,(t_{219})$	**4.23**	4.87 (15.2%)	[2.39; 7.34]	3.99 (5.5%)	[0.00; 8.64]
$LAI\,(t_{233})$	4.91	5.22 (6.2%)	[1.57; 8.86]	4.27 (13.0%)	[0.00; 11.21]
$magrain\,(t_{266})$	819.38	804.33 (1.8%)	[611.68; 996.98]	550.58 (32.8%)	[128.97; 972.20]

Table 9.22: Comparison of the prediction results of the STICS model with and without data assimilation performed with CPF calibration for the 2013 dataset for variety *Numeric*.

winter wheat data, since severe stresses due to heavy rain and frost in winter were observed which resulted in a far lower plant density compared to 2012. However, since there was more space available for each plant, a compensation phenomenon was observed. The LAI curve got delayed but finally nearly caught up with what would have been observed with the regular climate conditions, and therefore grain yield was not seriously influenced. However, the prediction performed without assimilation was not able to capture this compensation phenomena and thus failed to provide a reasonable prediction for the LAI and for the final yield (*cf.* variable *magrain* in Table 9.22 and Table 9.21), while the proposed approach has remarkably tackled this issue by updating properly the parameters and hidden variables based on the available data. As a result, the satisfactory predictions demonstrated its robustness in case of extreme weather scenarios.

On the other hand, the prediction improvement is less impressive with the soil water content variable, probably due to the fact that the prediction results obtained without data assimilation were already satisfactory, and there was no soil parameter selected by sensitivity analysis, and only light water stress conditions: the influence of the crop parameters updated in the data assimilation is limited, so that the difference with the case without assimilation is not so apparent (while still to the advantage of the assimilation method). In the light of this example, we may also consider a more subtle selection of parameters in the first step of our approach to improve the prediction of a specific output variable of interest when needed.

Finally, it is important to underline that when the proposed approach is applied to a deterministic model, the resulting uncertainty assessment is less rigorous in both the estimation and assimilation steps. As presented in Section 8.3.2, CPF can bring artificial dynamics to the deterministic STICS model by introducing two kernel functions associated to X_n^a and Y_n, so that more variations are tolerated and the estimation can be considered as a distribution. In practice, a value of $c_x \approx 1$ assured a good exploration of the state-space. Nevertheless, in this case, the posterior distributions obtained with the CPF approach are extensively influenced by the tuning parameters of the algorithm, especially the choice of the kernel functions and their bandwidth parameters, so that the credibility intervals computed should be regarded as contextual

approximations. This directly results from the fact that the model does not have a probabilistic framework, which of course restricts the validity of the statistical analysis.

CHAPTER 10

Discussion and Perspective

To improve the predictive performance of plant growth models, various efforts are made from different perspectives in literature. However, based on the existing approaches and strategies, the plant growth prediction application appears to be very case-specific, the methods and the data analysis settings employed are often either too superficial regarding the prediction objective or simply chosen based on their availability, without a profound analysis. A full methodology dedicated to guide a good practice of the data assimilation technique could benefit the development of applications.

In this context, this thesis encompasses the essential core steps of modelling, including model design, sensitivity analysis, model calibration (also known as parameterization), model selection, uncertainty analysis, model prediction and eventually design of experiments. Aiming to enhance the predictive performance in terms of both point estimation and the associated uncertainty assessment, we attempted to provide such an approach described in a probabilistic Bayesian framework to perform data assimilation. In this last chapter, we overview and discuss the principal results and contributions of this thesis, some possible future research directions are accordingly proposed.

10.1 Contributions and results

The objective of this thesis is to develop a full methodology to address the data assimilation and prediction issues in plant growth models from different perspectives. Our main contributions can be summarized in four categories, described below.

10.1.1 Model design

Firstly, from a model design perspective, a simple crop model, the Log-Normal Allocation and Senescence (LNAS) model was elaborated and adapted to the imple-

mentation of a Bayesian approach to perform model prediction with a special focus on the characterization of the inevitable uncertainty present in plant growth models. To facilitate the application of Bayesian based methods, the plant growth model used for prediction purpose should be described in a general state-space model form. The current trend in plant growth modelling is to develop sophisticated models with detailed description of plant-environment interactions Yin and Struik (2010), which leads to an important number of parameters in the model. With insufficient number of available observation data, estimation difficulties can frequently occur, while the generally deterministic model form does not allow a proper assessment of uncertainty.

In this context, we aim to find a balance between the complexity of the model and the quantity of available data in order to avoid identifiability problems. In this regard, being built from a probabilistic perspective, the LNAS model describes only the essential ecophysiological processes involved in biomass budget based on the analysis of Delécolle et al. (1992), for instance, biomass production, biomass allocation, senescence and leaf surface development. Although the parameters of these functions are no longer supposed to be from genetic origin only, our assumption is that the empirical functions describing the ecophysiological processes are flexible enough to adapt to a large range of situations, and robust enough so that the mean prediction remains pertinent. Moreover, such simplification allows an easier representation of the model errors without increasing significantly the number of parameters. Indeed, an extra attention is paid in the construction of the LNAS model to describe the uncertainty in crop growth. The introduction of both modelling noises and observation noises helps to avoid an excessive forcing of the model towards the observations and allows to take into account multiple stochastic factors, including environmental noises and various stress and compensation factors not included in the model. As a counterpart, the combination of these two types of noises alters the dependence structure and produces significant computational difficulties. Therefore, extensive numerical simulation integration methods are required for conducting statistical inference. The parameter estimation of such a state space model becomes a major challenging research subject. In the following, we present the contributions of this thesis in these regards.

10.1.2 Parameter estimation and uncertainty assessment

Dynamic crop models are generally characterized by complex interacting processes and a large number of model parameters. As a specificity of agricultural systems, experimental data acquisition tends to be costly (when direct field data collections are involved) or inaccurate (for indirect measurements such as satellite images), and generally irregular. Therefore, due to the nonlinear dynamics of the system equations, the restricted experimental data and the considerable uncertainty of the inputs, the parameterization of these models is generally regarded as a key issue which may strongly affect the quality of model prediction. Efficient and precise parameter estimation with proper assessment of the parameter uncertainty is thus crucial to ensure reliable and satisfactory prediction performance.

In this thesis, a thorough research is conducted regarding model parameterization methods, with a special focus on Bayesian estimation, involving both MCMC-based

methods and SMC methods.

Bayesian inference

Regarding MCMC based methods, we investigated the three strategies that are often opposed in applications, namely "one long run", "some median runs" and "many short runs". For this purpose, we respectively implemented and studied the Adapted Metropolis-within-Gibbs algorithm (Haario et al., 2001; Andrieu and Thoms, 2008), the DREAM algorithm (Vrugt et al., 2009a) and interacting parallel MCMC (Campillo et al., 2009). For the three strategies, a new scheme was proposed to deal with the issue of poor mixing between hidden states and parameters in state-space plant growth models. The convergence properties and behaviours of each method were studied and compared based on both the simulated datasets and a real experimental dataset.

Globally, the three algorithms provided accurate mean estimates. Although the one long run has the advantage to provide accurate mean and variance estimates, the algorithm cannot be parallelized and suffers from the long computational time. For the sake of computational efficiency, parallel chains are also widely studied.

When parallel chains are engaged without communication, the overall speed of convergence is limited by the convergence of each chain. The DREAM algorithm proposes to hasten the convergence of individual chains by creating communications between the chains. However, according to our results, it still appears to have slow convergence issues and the burn-in period for each of its chains is difficult to define. Moreover, it requires an appropriate starting distribution which is over-dispersed. The performance of the classical Gelman-Rubin criterion for multiple chains convergence diagnostic is also proved to be inefficient. Therefore, when the number of chains increases, more and more outliers could be included in the final pooled estimates which are used to construct the posterior distribution, while a small number of chains are very likely to underestimate the mean posterior variance. In order to determine the number of samples of the chains that should be discarded pertinently, an efficient way is to take only the last sample of each chain, as in the case of the interacting parallel MCMC algorithm.

By gathering the candidates of all the individual chains together and being selected by all the chains, the idea is to concentrate the computational effort on the zone of interest. Although one iteration of Interacting MCMC generally takes more time than one iteration of parallel independent MCMC, our tests demonstrated that the interacting scheme, which can be regarded as a sort of importance sampling, clearly improved the convergence efficiency. Nevertheless, manifestly the algorithm suffered from the sample impoverishment issue and failed to provide appropriate variance estimates to characterize precisely the target distributions, for the regions with small acceptance probability are rarely visited. Although some efforts of improvement were made to extend the interacting parallel runs without the interacting scheme, the variance estimates still remained less accurate than those of a long run.

Our results suggest that multiple chains perform less efficiently than a single chain (at least for the type of situations considered in this thesis). They may save some computational time, but their estimation of posterior distribution is less accurate than

one long run methods, especially for nonlinear models and limited data. Most importantly, it seems more difficult to define a proper way to diagnose the convergence for multiple parallel chains.

Another important category of Bayesian estimation methods are the SMC filters. The Unscented Kalman Filter (Julier and Uhlmann, 1997; Quach et al., 2007), the Ensemble Kalman Filter (Evensen, 1994) and the Regularized/Convolution Particle Filter (Musso and Oudjane, 1998; Oudjane and Musso, 1998; Campillo and Rossi, 2009) were also implemented and studied. The three methods are designed to confront nonlinear systems. Contrary to the MCMC-based approaches, the filtering methods are performed in a sequential way. They are able to take into account the variation of parameters over time and carry out on-line updating. Based on the Kalman prediction-correction filter equations, UKF uses a small set of sigma-points to approximate both the mean and the covariance matrix in the prediction step. EnKF relies on normality assumptions in order to improve the accuracy of its estimates with a more important number of samples. Finally, CPF and RPF are particle filter-based methods, they intend to provide better approximation of the exact posterior distributions by creating a set of randomly drawn samples that are propagated in the dynamic system. Each trajectory has an associated weight, evolving at each filtering step. A kernel smoothing method is used to change the discrete approximation of the filtering density into an absolutely continuous approximation. The difference between CPF and RPF is that for CPF, a convolution kernel is used to regularize the likelihood of the observations as well. Since in our application to the LNAS model, the likelihood of the observation is considered known, RPF was used to perform the model calibration.

According to our tests, unlike the MCMC-based methods that generally provided similar estimations and remain stable, UKF, EnKF and RPF usually disagreed on the estimation results. The estimations given by RPF were often similar to those of the MCMC based methods, which suggests a reliable performance, even though the Monte Carlo error was generally larger. On the other hand, UKF is limited by its sample size which often led to poor estimations, especially when the system is characterized by a strong nonlinearity. In comparison, the other Kalman based filter EnKF behaved better. Its larger sample size makes it more adapted to nonlinear systems than UKF, but it is still affected by the underlying Gaussian approximation, and finally yielded less accurate estimations than those of RPF.

Iterative approaches

We highlight that when the number of data is limited, one crucial point to define the performance of Bayesian inference is the choice of prior. In the context of plant growth modelling, when a new model is developed (as for LNAS), there might be a lack of accuracy in the prior knowledge that could be gathered, since few specific studies on the parameterization of the model exist. The desirable strategy is hence to put more emphasis on the data, while still preserving an interpretable context and to benefit from the robustness of Bayesian based methods. Motivated by this thought, in this thesis, an iterative scheme was proposed based on the SMC or MCMC Bayesian estimation methods to improve the estimation accuracy. It can be regarded as a variant of the Expectation-Maximization algorithm under the normal assumption of the prior

distributions. A theoretical framework was elaborated in this regard to describe this variant of EM type algorithm. The term Gaussian randomization was introduced to designate that the initial model could be seen as the special case of an extended model which includes all the Gaussian distributions with the parameters of the original model as the means.

The estimation results showed that the iterative RPF performed better than the MCMC-based iterative approach. It can be explained by the fact that since the variance of the posterior tends to 0 with the EM iterations, the resampling scheme allows RPF to introduce and to maintain the dynamics among the population or particles, which cannot be achieved by the MCMC-based iterative approach. Accordingly, parametric bootstrap was used to evaluate the uncertainties associated with the mean estimates of the parameters.

For both Bayesian and iterative methods, when the parameters of the noise model are unknown, a conditional approach was described for their estimation. The noise parameters were jointly estimated with the functional parameters and the hidden state variables to characterize the uncertainty present in plant growth models. The noise parameters were either updated with conjugate priors, or estimated with empirical estimators, or fixed by model selection. The simulation results indicate that such a framework usually allows for the quantification of uncertainty both at parameter level and error model level.

Finally, a decision aid table for the choice of estimation strategies is proposed as follows with regards to different situations and needs:

Conditions		Recommended method(s)
• Time constraint, no reliable prior, only point estimation is needed	⇒	GLS
• Time constraint, slightly non linear model	⇒	UKF
• High-dimensional parameter vector	⇒	DREAM
• Precision of the mean estimate	⇒	Interacting MCMC
• Reliable prior, need precise CI%, no time constraint, implementation expertise in MCMC	⇒	One long run AMwG
• Reliable prior, time constraint	⇒	CPF, EnKF
• Less reliable priors	⇒	Iterative SMC, especially ICPF/IRPF

Table 10.1: Selection strategy for estimation methods.

10.1.3 Data assimilation approach

Thirdly, a three step data assimilation methodology was proposed as a guide to tackle model prediction problems. The most influential parameters are first selected by global sensitivity analysis and accordingly estimated in a Bayesian framework by either Sequential Monte Carlo (SMC) methods or Markov chain Monte Carlo (MCMC) methods in the calibration step. The posterior distribution obtained from this step

can be subsequently considered as prior information for the prediction performed with data assimilation. In this last step, a SMC-based on-line estimation method is applied to update state and parameter estimates based on the data obtained at early growth stages, with the purpose of improving model predictions and assessing or even reducing the associated prediction uncertainty.

Sensitivity analysis and model selection
For the first step, a sensitivity analysis method is carried out first to rank the functional model parameters according to their influences on model outcome variables of interest. Then, by selecting an increasing number of parameters to estimate (following the importance ranking and starting with the most influential one), a family of models is built. The parameters that are not selected are fixed to their recommended values. To reduce the number of candidate models, a threshold can also be defined to even exclude *a priori* of the selection process the least influential parameters.

A selection method is then applied to determine the best model, and subsequently, the number of parameters to estimate. Generally the selection process is based on model calibration and standard criteria, like AIC, AICc or DIC.

In the applications to the LNAS and STICS models, this step allowed us to reduce the number of parameters from 6 to 3 and from 16 to 4 respectively. The other parameters are thus fixed to their recommended values given by literature and considered as constants.

However, it is important to note that the best model selected from the calibration step is not proved to be the best model in a predictive context, with data assimilation. If the DIC criterion is used with bayesian estimation methods in the calibration step, it seems that it penalizes too much the number of parameters, finally allowing too little flexibility in the data assimilation step, to the detriment of model adaptation and predictive capacity. This results suggest that a model selection criterion with less penalty on the number of the model parameters could be preferred for the prediction objective.

Parameter estimation for stochastic state space models
In this second step, when a calibration data set is available, the estimation methods investigated in Chapter 4 and Chapter 5 can be applied. For the type of models considered in this thesis and the type of classically available data, Bayesian methods (principally one long run AMwG and RPF/CPF) were able to provide fair posterior distributions, that could be used in the data assimilation step.

The iterative approaches were also able to provide estimates that approach MLE, which is a desirable feature when the prior information is not pertinent. The confidence intervals obtained by parametric bootstrap were however far larger than the credibility intervals given by the Bayesian approaches. Although these intervals cannot be compared for they bear different meanings, the predictive performances of the two approaches (Bayesian versus Iterative) were evaluated in our data assimilation application for the LNAS model.

Prediction with assimilation
In the last step, the model prediction with assimilation of data from early growth stages was carried out with the four SMC filtering methods (CPF, RPF, UKF, EnKF). Unlike RPF, CPF applies when the likelihood cannot be computed explicitly, another convolution kernel being introduced for the likelihood computation of the observation function. It is useful when the form of the observation error is not known. In our tests, data assimilation with CPF showed similar prediction performances as RPF.

Regarding the comparison with the prediction by UKF and EnKF, generally, CPF/RPF yielded more accurate mean estimates and reduced more the prediction uncertainty than the other two Kalman based filtering methods, and therefore CPF and RPF appears to be more robust and suitable in our application context.

The influence of the calibration precision was also studied. Generally, the precision has a direct impact on the prediction accuracy. However, some exceptions were detected, specifically when the environmental conditions were very different in the prediction phase compared to the calibration phase, and when the model was not able to handle this difference. In that case, the accuracy of the prior induced some bias in the prediction step, by restraining the model adaptability.

Likewise, multiple observation noise levels were considered. The impact on the mean predictions was small, but it was important on the evaluation of prediction uncertainty. This result suggests that the evaluation of the observation noise level is crucial and should be determined with care, since it could vary from one experiment to another and can be influenced by many factors that are difficult to assess.

Moreover, we also demonstrated that in our study case, the number of parameters recalibrated in the data assimilation step was also of great importance. Generally speaking, the issue is to determine the proper number of parameters to be re-calibrated allowing a sufficient liberty for the calibrated system to re-adjust itself to the new context, while preserving the model robustness and avoiding the identifiability issues. As mentioned above, the optimal choice in terms of prediction is difficult to determine *a priori*. It seems that by increasing slightly the number of parameters suggested by the DIC in the estimation step, or rather by using a more tolerant criteria, the quality of the prediction results obtained with data assimilation could be improved.

Finally, the impact of the number of data based on which data assimilation can be performed and the impact of their acquisition dates were evaluated. As expected, the latest observed data (that is to say those that are close to the prediction dates) tend to be the most informative, and consequently, they improve greatly prediction results compared to cases when the same data is not available for assimilation. More dispersed observation dates also appeared to yield better prediction precision. Furthermore, the number of repetitions of observations did not appear to be as important as their acquisition dates for the prediction quality. This part opens some perspectives towards the optimal design of experiments, regarding the choice of the observation dates and of the type of data to collect, with the objective to improve the predictive capacity.

10.1.4 Application with real experimental data

Finally, from an application point of view, the proposed methods are implemented and evaluated with two crop models, the STICS model for winter wheat and the LNAS model for sugar beet, to illustrate the robustness of the proposed data assimilation approach. Five datasets obtained in various experimental conditions were used for the sugar beet LNAS model, and three datasets for the winter wheat STICS model. In both studies, one dataset was used for *a priori* parameter estimation and the others were used to test the model predictive capacity, both with and without data assimilation. The RPF-based data assimilation approach showed promising predictive capacity and provided robust and reduced credibility intervals in various test configurations (different years for calibration and prediction by assimilation, different experimental sites, different cultivars, different crop densities, different levels of water stresses), which suggests that the combination of such an approach with both types of crop models (simple probabilistic model or complex deterministic model) is quite reliable and can therefore be regarded as a potential tool for yield prediction applications in agriculture. The fact that the method can be applied straightforwardly to deterministic model is very interesting. There is however a limitation since the confidence intervals produced by this method are not statistically relevant (since there is no statistical model for this purpose).

10.2 Perspectives

While answering a few of the initial questions addressed in this thesis, a lot of issues remain unsolved or were raised during this work. In the following, some perspective and research directions are discussed.

10.2.1 Modelling Uncertainty

Assumption for noise models
The proper assessment of uncertainty appears as a crucial point for the usefulness of plant models (Ford and Kennedy, 2011). In the construction of the LNAS model introduced in Section 2.1, a multiplicative Gaussian noise structure is adopted for both the modelling noises and the observation noises.

However, other distributions could have been tested in a model selection perspective. The log-normal distribution is a classical choice, specifically to prevent positivity loss. Both log additive or log multiplicative noise can be considered. In this case, our evolution model can be transformed to:

$$\begin{aligned} Q(t) &= F_Q^t(Q_f(t);\Theta)\, e^{\eta_Q(t)}, \\ \gamma(t) &= \Gamma^t(\Theta)\, e^{\eta_\gamma(t)}, \qquad t \geq 1, \end{aligned}$$

where $\{\eta_Q(t)\}_{t\geq 1}$ and $\{\eta_\gamma(t)\}_{t\geq 1}$ are independent from X_0 sequence of mutually independent centered Gaussian random variables with variances σ_Q and $\sigma_{\gamma\gamma}$ respectively.

So as the observation equations:

$$Y(t) = \begin{pmatrix} \log Q_g(t) + \xi_g(t) \\ \log Q_r(t) + \xi_r(t) \end{pmatrix}, \tag{10.1}$$

where $\{\xi_g(t)\}_{t\geq 1}$ and $\{\xi_r(t)\}_{t\geq 1}$ are i.i.d. mutually independent sequences of centered Gaussian random variables with variances σ_g^2 for $\xi_g(t)$ and σ_r^2 for $\xi_r(t)$, and assumed to be independent from $\{\eta_Q(t)\}_{t\geq 1}$ and $\{\eta_\gamma(t)\}_{t\geq 1}$.

Regarding the deterministic STICS model presented in this thesis, it could also be interesting to consider its adaptation in a state-space form described in a probabilistic framework in order to characterize and identify different sources of uncertainty. Similar log-normal noise models can be introduced for biomass production, allocation and the stress functions. Of course, some more experimental data would be available to avoid identifiability problems, since potentially many different modelling noises could be introduced.

Finally, the evaluation of the appropriateness of a noise model could be contextual, the assessment of uncertainty still remains a critical and crucial point to determine the performance of the prediction. An excessive forcing of the model towards the observations is not desired, yet too much noise could conceal the useful information. How to extract most information from the available data while still addressing the uncertainty in an appropriate way in order to provide reliable credibility intervals? Further investigations are still required.

Climatic uncertainty

A very strong assumption was used in all our prediction tests: we assumed that the climatic data were known, even after the last data acquisition that was used for assimilation. Of course, this is not the case in real situations. For real prediction applications, it would be useful to also consider the uncertainty in the climatic variables. The most classical way in this objective is to use historical climate data, for example in the last 50 years, and use these data to run simulations for each of these 50 climatic scenarios. Ideally, these 50 simulations should be run for each particle describing the posterior distribution obtained in the filtering process. However, since after the last filtering step, we are in a pure uncertainty analysis situation (propagation of uncertainty in the plant dynamic system), the number of particles can be reduced.

A problem of this classical approach based on historical scenarios is that the simulations run after the last filtering step at date t_f are based on climatic scenarios that are independent of the observed climatic variables until t_f while these are available, and there exist long dependence chains in the climatic time series. For this purpose, it would be interesting to couple our approach with stochastic weather generators that simulate weather scenarios conditionally on the observed climatic variables (Yiou, 2014).

Towards population-based models

In this thesis, both the LNAS and the STICS models consider only mean population values to assess the impact of covariates and to perform the model prediction. However, if there is a strong variability among individuals, the conclusion can be severely

biased. Indeed, describing plant populations only in terms of mean growth is often not adequate. Although the modelling noises allow to take into account some variations resulting from the population effect, it is still different from studying specifically the inter-individual variability, which could help us to understand population behaviour and thus enhance the predictive capacity of plant growth models at field scale.

For this purpose, individual centered models like the LNAS or GreenLab models can be extended to a population level using mixed effect models, as proposed by Baey et al. (2013b); Baey (2014); Baey et al. (2014). They also proposed an estimation method combining multiple chains Monte Carlo with Expectation-Maximization algorithm (Stochastic Approximation of EM algorithm, see Delyon et al. (1999)) to mimic the population dynamics. In order to study the covariates' effect, the maximum likelihood function used in the MCMC algorithm should therefore be based on the joint density of the observation given the covariates, and the Wald test can be used for model selection. Different types of covariance structures as well as different noise models can be considered and compared, so as the significance of the random effect introduced. Given the fact that the variance estimates of the random effects provided by the Maximum Likelihood method can be biased (Meza et al., 2007), it can be assessed with the Restricted Maximum Likelihood (REML) (see Foulley et al. (2000) and Foulley and Van Dyk (2012)).

Finding population data adapted to the GreenLab population model iss difficult since data at organ scales are heavy to collect, and it is of course multiplied when we are interested in a population of individuals ... Moreover, the population formulation was based on a deterministic version of the GreenLab model, without modelling noise. Therefore, further study can be conducted by combining this population approach with other simplified plant growth models described in a probabilistic framework to avoid identifiability problems and to take into account growth process variations via modelling noises. The LNAS model is a good candidate for this purpose since it requires far less data than the GreenLab model for its parameterization and would thus open the way to wider applications, with more individual plant samples.

Moreover, if an important number of sample plants are available, we could assume that some modelling noises (random effects) are generated from a non-parametric distribution (*e.g.* a mixture of mass points with each corresponding to a cluster of plants) in order to admit more variability among the plant population while avoiding over-fitting problem by introducing more parameters (see for example Debashis et al. (2011)).

It would also be interesting to adapt the Bayesian methods investigated in our thesis, to take advantage of the prior knowledge on model parameters, and also to avoid the iterative process of the EM algorithm which repeats a large number of MCMC computations and is thus very expensive in the context of mixed models when the population size becomes important. One chain should be enough in the Bayesian frame, even if of course the convergence could be very slow according to the number of parameters and of mixed effects to consider.

Finally, methods like cross validation can be used to identify the bias of model selection. Population based models and average individual based models can therefore be compared with regards to their predictive capacity.

10.2.2 Estimation Methods

In terms of estimation, the results obtained by AMwG with our proposed scheme to handle the parameter - hidden states mixing issue, the RPF / CPF approach, and the iterative versions of RPF / CPF are very satisfactory regarding our parameterization and uncertainty assessment objectives. However a few interesting perspectives could still be considered. The first one concerns the interacting paralell MCMC (Campillo et al., 2009), for which a few things could be tested to improve the under-evaluation of the posterior variances. Another promising approach is the hybrid Particle MCMC method introduced by Andrieu et al. (2010).

A hybrid Particle MCMC method

The objective of MCMC based methods for parameter estimation in state-spaces models can be reformulated as follows: given a batch of observation $y_{1:t_{max}}$, we seek the posterior distribution $\pi(\Theta|y_{1:t_{max}}) \propto p(y_{1:t_{max}}|\Theta)\pi(\Theta)$ with $\pi(\Theta)$ the prior distribution. This posterior distribution is typically not available in closed form. A possible remedy is to include the hidden state variables $x_{1:t_{max}}$ as auxiliary variables. Therefore, a joint update scheme is proposed.

- Update $\Theta' \sim \pi(\Theta|x_{1:t_{max}}, y_{1:t_{max}})$.
- Update $x'_{1:t_{max}} \sim \pi(x_{1:t_{max}}|\Theta, y_{1:t_{max}})$.

By sampling Θ conditionally on both the observed data and the hidden state variables, it generally simplifies the update. For non-conjugate models as in our case, a MH step is used to replace the exact update. However, the second step is much less straightforward.

In practice, slow mixing occurs frequently when strong temporal dependencies are present in the model. Many efforts are made in the literature to improve the sampling process of the hidden states estimation, including dividing the trajectory in blocks and updating them subsequently (see Carter and Kohn (1994) for block independence sampler), or updating the whole trajectory from its full conditional (see Rabiner and Juang (1986) for the Kalman filter algorithm and Scott (2002); Fearnhead (2006) for the forward-backward algorithms). However, the block updating may result in slow mixing at the boundaries of the blocks, Shephard and Pitt (1997) suggest that the blocks to be updated can be chosen randomly.

Another alternative consists in simulating directly from the full conditional of the hidden state variables, but there is the important constraint that the full conditional can be expressed analytically, which cannot be applied to many models. Likewise, independent proposals are built under the Gaussian assumption and thus can only be used for limited models. It is possible to obtain good proposals for state-space models in a more general way, but this can be a challenging task, especially for models with high-dimensional states and important nonlinearity.

Note that this can be regarded as another argument for simplified model design, since the more sophisticated a model is, the more difficult its parameterization will be, and very limited methods can be applied considering the computational constraint.

A promising approach for block updates of hidden state variables has been recently proposed by Andrieu et al. (2010), which is based on employing SMC methods within a Markov chain. The idea is to benefit from the efficiency of the SMC methods by generating a proposal distribution for the trajectory of the hidden state variables inside an MCMC algorithm. This framework is described as Particle Markov chain Monte Carlo (PMCMC).

As we stated in Section 4.1.4, ideally the proposal $q(x'_{1:t_{max}}|\Theta')$ should be taken as $p(x'_{1:t_{max}}|\Theta', y_{1:t_{max}})$, but since it is difficult to achieve, anther way to approximating the intractable $\pi(\Theta|y_{1:t_{max}})$ required for the computation of the acceptance probability of the MH update is to use importance sampling estimates :

$$\tilde{\pi}(\Theta|y_{1:t_{max}}) = \frac{1}{N}\sum_{k=1}^{N}\frac{\pi(\Theta, x_{1:t_{max}}(k)|y_{1:t_{max}})}{q(x_{1:t_{max}}(k)|\Theta)} \quad \text{with} \quad x_{1:t_{max}}(k) \overset{i.i.d.}{\sim} q(x_{1:t_{max}}|\Theta).$$

So that the acceptance probability can be simplified as :

$$\min\left(1, \frac{\tilde{\pi}(\Theta'|y_{1:t_{max}})q(\Theta|\Theta')}{\tilde{\pi}(\Theta|y_{1:t_{max}})q(\Theta'|\Theta)}\right).$$

However, in a general way, the success of this approach relies greatly on using a large number of particles M in the underlying particle filtering method, to obtain accurate kernel approximations. Although it seems to be a natural requirement, it also implies that a lot of computational resources are wasted. This can be explained by the fact that at each iteration of the Gibbs sampler, a large number of particles are used to extract a single state trajectory.

Based on the actual version of AMwG described in Section 4.1.4 and its demonstrated performance, a more simple way to improve the PMCMC performance could be to introduce a particle filter inside the parameter update every k iterations for the updates of hidden states, or to consider a smaller sample size for a more often hidden state update. The performance of different proposal distributions can also be explored. We believe this idea is worth further investigation for our applications of interest in plant growth model calibration.

10.2.3 Data assimilation applications

Of course, a deeper investigation on real test cases is necessary to show the robustness of the proposed approach. For example, a potentially interesting performance comparison for future study can be using both a sophisticated model like STICS and a probabilistic model like LNAS to perform model prediction with the same crop and same observation datasets.

With real applications in mind, a crucial aspect is the assimilation of data obtained via remote sensing techniques, either from satellite images or aerial (drone for example) images. It has been a long-standing challenge and there were already several pioneer applications carried out Guérif and Duke (2000); Launay and Guérif (2005). It will be interesting to see how this kind of information can be integrated into our probabilistic

model and how it helps for its predictive capacity. The information provided by these techniques are potentially cheap and quite frequently available during the crop cycle. However, few biophysical variables can be estimated, and the level of uncertainty remains pretty high. Observations of some agronomic variables in the field (phenological stages for example, or direct LAI measurements with LAI analyzers, some biomass sampling ...) could greatly complement the remote sensing data.

More generally, since the acquisition of field measurements is expensive, the study of experimental design can be useful to reduce the number of measurements and to find a compromise between the prediction accuracy and the measurement cost. Not only the number of data, their precision and the date of observations can be studied, but of course also the type of measured variables, and the combination of different acquisition sources (*e.g* LAI analyzer versus satellite / drone image analysis). The answer would differ with the type of models considered.

The 'Information Theory' is the scientific field that can help tackle the issue of Design of Experiments (DOE). A design space has to be chosen, and the question is mainly about how to identify the aggregation of the data points among all the candidate points in order to best reflect the information in the considered design space. In the meantime, it is also important to identify some validation points in order to test the accuracy of the model. Various methods are proposed in the literature, among which two main statistical approaches (Koehler and Owen (1996), chapter 9), one based on frequentist sampling technique, like *Latin Hypercube Sampling*, *Scrambled net* and *Grids*. The other concerns the Bayesian experimental design techniques which takes into account prior knowledge, such as *Minimas Designs*, *Mean Squared-Error designs* or *Maximum Entropy Sampling*. For both approaches an optimal design is one that maximizes a given optimality criterion. The two most suitable DOE techniques for sequential design problems as for plant growth modelling are D-optimal design and Bayesian Entropy Sampling.

To sum up, plant growth modelling offers promising perspectives, for example for virtual experimentation in the context of breeding, or to provide yield potential prediction for decision support. However, there still remain a considerable amount of issues to be addressed to ensure the quality of model predictions. How to find a balance between the complexity of the model (a good indicator of this complexity being the number of functional model parameters) and the necessary data for its parameterization in order to avoid over-fitting and under-fitting? How do parameters vary across years, different locations or genotypes? How to distinguish and to identify the different sources of variations and uncertainty? How to take into account in the yield prediction the uncertainty related to the climate (and climate change!)? How to optimize the design of experiments? These issues should be considered crucial and properly addressed by the new generation of plant models.

Publications

• Publications in peer-reviewed journals :

Y.-T. Chen and P.-H. Cournède. Data assimilation to reduce uncertainty of crop yield prediction based on the Log-normal allocation and senescence crop model and convolution particle filtering. *Ecological Modelling*. Accepted, 2014.

Y.-T. Chen, S. Trevezas, P.-H. Cournède. A regularized particle filter EM algorithm based on Gaussian randomization with an application to plant growth modeling. Submitted, 2013.

P.-H. Cournède, Y.-T. Chen, Q.-L. Wu, C. Baey, B. Bayol. Development and evaluation of plant growth models: Methodology and implementation in the Pygmalion platform. *Mathematical Modelling of Natural Phenomena*, 2013, 8(4), pages 112-130.

• Publications in peer-reviewed conferences/proceedings :

Y.-T. Chen, S. Trevezas, A. Gupta, P.-H. Cournède. Some sequential Monte Carlo techniques for Data Assimilation in a plant growth model. *15th Conference of Applied Stochastic Models and Data Analysis (ASMDA, Barcelona, Spain), Proceedings of ASMDA*, 2013.

Y.-T. Chen, S. Trevezas, P.-H. Cournède. Iterative convolution particle filtering for nonlinear parameter estimation and data assimilation with application to crop yield prediction. *SIAM Conference on Control and its Applications (CT13, San Diego, USA)*, pages 67-74, 2013.

B. Bayol, Y.-T. Chen and P.-H. Cournède. Towards an EDSL to Enhance Good Modeling Practice for Non-linear Stochastic Discrete Dynamical Models - Application to Plant Growth Models. *Third International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2013, Reykjavik, Iceland)*, SciTePress, pages 132-138, 2013.

Y.-T. Chen and P.-H. Cournède. Assessment of parameter uncertainty in plant growth model identification. *Fourth International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications (PMA'12, Shanghai, China)*, IEEE press, pages 85-92, 2012.

Y.-T. Chen, B. Bayol, C. Loi, S. Trevezas, P.-H. Cournède. Filtrage par noyaux de convolution itératif. *44e Journées de Statistique (JdS'2012, Brussels, Belgium), Proceedings of JdS'2012*, 2012.

Annexes

Notations

- Plant Growth State-Space Model
  - $\mathcal{M}$: Metamodel.
  - Q_n : Biomass produced at time n.
  - t_{max} : Last date of observations.
  - Θ_1 : Functional parameter vector.
  - Θ_2 : Noise parameter vector.
  - Θ : Parameter vector, $\Theta = (\Theta_1, \Theta_2)$.
  - X_n : Hidden state vector.
  - X_n^a : Augmented hidden state vector, $X_n^a = (X_n, \Theta_n)$.
  - Y_n : Observation vector.
  - E : Environmental control vector.

- Probability
  - P : Probability.
  - $\mathbb{E}$: Expectation.
  - $\sim$: Sampled from or distributed according to.
  - $\mathbb{V}$: Variance.
  - Var, Cov : Variance, covariance.
  - diag(V) : Diagonal matrix with elements of V on the diagonal.
  - $p(\cdot)$: Probability distribution function.

- Abbreviations
  - ACF : Autocorrelation function.
  - AIC : Akaike information criterion.
  - AICc : Akaike information criterion with correction.
  - AM : Adaptive Metropolis.
  - AMwG : Adaptive Metropolis-within-Gibbs.
  - *a.s.* : almost surely.
  - BIC : Bayesian information criterion.
  - CLT : Central limit theorem.
  - CPF : Convolution particle filter.
  - DA : Data assimilation.
  - DIC : Deviance information criterion.
  - DREAM : Differential evolution adaptive Metropolis.
  - EM : Expectation-Maximization.
  - EnKF : Ensemble Kalman filter.
  - GLS : Generalized least squares.
  - i.i.d. : independent and identically distributed.
  - KF : Kalman filter.
  - MCMC : Markov chain Monte Carlo.
  - MH : Metropolis-Hastings.
  - ML : Maximum likelihood.
  - MLE : Maximum likelihood estimator.
  - RPF : Regularized particle filter.
  - SA : Sensitivity analysis.
  - SMC : Sequential Monte Carlo.
  - UA : Uncertainty analysis.
  - UKF : Unscented Kalman filter.

Bibliography

S. Adjemian. Prior distributions in Dynare. 2010. URL http://www.dynare.org/stepan/dynare/text/DynareDistributions.pdf.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, (6):716–723, 1974.

B. Anderson and J. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

J. L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Mon. Wea. Rev.*, 129:2884–2903, 2001.

C. Andrieu and A. Doucet. Online expectation-maximization type algorithms for parameter estimation in general state space models. In *Proc. IEEE Conf. ICASSP*, pages 69–72, 2003.

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient computations. *Annals of Statistics*, 37(2):697–725, 2009.

C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18:343–373, 2008.

C. Andrieu, N. de Freitas, and A. Doucet. Robust full Bayesian learning for radial basis networks, 2001.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society*, (72):269–342, 2010.

M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. In *Connection between forest resources and wood quality: modelling approaches and simulation software*, number 50(2), pages 174–188. IEEE Trans. Signal Proces., 2002.

K. B. Athreya, H. Doss, and J. Sethuraman. On the convergence of the Markov chain simulation method. *The Annals of Statistics*, 24(1):1–448, 1996.

C. Baey. *Modélisation de la variabilité inter-individuelle dans les modèles de croissance de plantes et sélection de modèles pour la prévision*. PhD thesis, Ecole Centrale Paris, 2014.

C. Baey and P.-H. Cournède. Using a hierarchical segmented model to assess the dynamics of leaf appearance in plant populations. In *Proceedings of ASMDA 2011*, 2011.

C. Baey, A. Didier, S. Li, S. Lemaire, F. Maupas, and P.-H. Cournède. Evaluation of the predictive capacity of five plant growth models for sugar beet. In *4th international symposium on Plant Growth and Applications (PMA12), Shanghai, China*. IEEE, 2012.

C. Baey, A. Didier, S. Lemaire, F. Maupas, and P.-H. Cournède. Parametrization of five classical plant growth models applied to sugar beet and comparison of their predictive capacity on root yield and total biomass. *Ecological Modelling*, 2013a.

C. Baey, A. Didier, S. Lemaire, F. Maupas, and P.-H. Cournède. Modelling the interindividual variability of organogenesis in sugar beet populations using a hierarchical segmented model. *Ecological Modelling*, 263:56–63, 2013b.

C. Baey, S. Trevezas, and P.-H. Cournède. A nonlinear mixed effects model of plant growth and estimation via stochastic variants of the EM algorithm. Submitted, 2014.

F. Baret, V. Houles, and M. Guérif. Quantification of plant stress using remote sensing observations and crop models: the case of nitrogen management. *Journal of Experimental Botany*, 58(4):869–880, 2007.

T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53(370), 1763.

B. Bayol, Y. Chen, and P.-H. Cournède. Towards an EDSL to enhance good modeling practice for non-linear stochastic discrete dynamical models - application to plant growth models. In *International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), 2013, Reykjavik, Iceland*, 2013.

M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164:1139–1160, 2003.

M. Bédard. Optimal acceptance rates for Metropolis algorithms: Moving beyond 0.234. *Stochastic Processes and their Applications*, 118(12):2198–2222, 2008.

J. Berger and R. Wolpert. *The Likelihood Principle*. Hayward, CA: Institute of Mathematical Statistics, 1984.

J. Bertheloot, P.-H. Cournède, and B. Andrieu. NEMA, a functional-structural model of nitrogen economy within wheat culms after flowering: I. Model description. *Annals of Botany*, 108(6), 2011.

J. G. Booth, J. P. Hobert, and W. Jank. A survey of Monte Carlo algorithms for maximizing the likelihood of a two-stage hierarchical model. *Statistical Modelling*, 1(4):333–349, 2001.

J. G. J. Booth and J. P. J. Hobert. Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. 61(1):265–285, 1999.

L. Bordes, D. Chauveau, and P. Vandekerkhove. A stochastic EM algorithm for a semiparametric mixture model. *Comp. Stat. and Data Analysis*, 51:5429–5443, 2007.

B. Bouman. Linking physical remote sensing models with crop growth simulation models, applied for sugar beet. *International Journal of Remote Sensing*, 13(14): 2565–2581, 1992. doi: 10.1080/01431169208904064.

E. Bradley and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.

N. Brisson, B. Mary, M. H. Ripoche, D.and Jeuffroy, F. Ruget, B. Nicoullaud, P. Gate, F. Devienne-Barret, R. Antonioletti, C. Durr, G. Richard, N. Beaudoin, S. Recous, X. Tayot, D. Plenet, P. Cellier, J. Machet, J. M. Meynard, and R. Delecolle. Stics : a generic model for the simulation of crops and their water and nitrogen balances. i. theory, and parameterization applied to wheat and corn. *Agronomie*, 18:311–346, 1998.

N. Brisson, C. Gary, E. Justes, R. Roche, B. Mary, D. Ripoche, D. Zimmer, J. Sierra, P. Bertuzzi, P. Burger, F. Bussière, Y. Cabidoche, P. Cellier, P. Debaeke, J. Gaudillère, C. Hénault, F. Maraux, B. Seguin, and H. Sinoquet. An overview of the crop model STICS. *European Journal of Agronomy*, 18:309–332, 2003.

N. Brisson, M. Launay, B. Mary, and N. Beaudoin, editors. *Conceptual Basis, Formalisations and Parameterization of the Stics Crop Model*. Éditions QUAE, Versailles, France, 2008.

S. Brooks and G. Roberts. Assessing convergence of Markov chain Monte Carlo algorithms. *Statistics and Computing*, 8:319–335, 1998.

G. Buck-Sorlin and K. Bachmann. Simulating the morphology of barley spike phenotypes using genotype information. *Agronomie*, 20:691–702, 2000.

K. Burnham and D. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Verlag, 2nd editio edition, 2002.

B. S. Caffo, W. Jank, and G. L. Jones. Ascent-based Monte Carlo expectation- maximization. 67(2):235–251, 2005.

F. Campillo and V. Rossi. Convolution particle filter for parameter estimation in general state-space models. *IEEE Transactions in Aerospace and Electronics.*, 45 (3):1063–1072, 2009.

F. Campillo, R. Rakotozafy, and V. Rossi. Parallel and interacting Markov chain Monte Carlo algorithm. *Mathematics and Computers in Simulation*, 79:3424–3433, 2009.

F. Campolongo, J. Cariboni, and A. Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental Modelling and Software*, 22:1509–1518, 2007.

O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13:907–929, 2004.

O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, New York, 2005.

F. Caron. *Inférence Bayésienne pour la Détermination et la Sélection de Modèles Stochastiques.* PhD thesis, Ecole Centrale de Lille, 2006.

C. Carter and R. Kohn. On Gibbs sampling for state space models. *Biometrika*, 83 (3):541–553, 1994.

G. Casella and E. Lehmann. *Theory of point estimation*. Springer, Berlin, 1998.

J. E. Cavanaugh. Unifying the derivations for the Akaike and Corrected Akaike Information Criteria, 1997.

G. Celeux and J. Diebolt. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82, 1985.

D. Chauveau and J. Diebolt. An automated stopping rule for MCMC convergence assessment. *Computational Statistics*, 14(3):419–442, 1999.

D. Chauveau and J. Diebolt. Estimation of the asymptotic variance in the CLT for Markov chains. *Stochastic Models*, 19(4):449–465, 2003.

D. Chauveau and P. Vandekerkhove. Improving convergence of the Hastings-Metropolis algorithm with a learning proposal. *Scan J Stat*, 29:13–29, 2002.

Y. Chen and P.-H. Cournède. Assessment of parameter uncertainty in plant growth model identification. In M. Kang, Y. Dumont, and Y. Guo, editors, *Plant growth Modeling, simulation, visualization and their Applications (PMA12)*, 2012.

Y. Chen and P.-H. Cournède. Data assimilation to reduce uncertainty of crop yield prediction based on the log-normal allocation and senescence crop model and convolution particle filtering. *Ecological Modelling*, Accepted, 2014.

Y. Chen, S. Trevezas, and P.-H. Cournède. Iterative convolution particle filtering for nonlinear parameter estimation and data assimilation with application to crop yield prediction. In *Society for Industrial and Applied Mathematics (SIAM): Control & its Applications, San Diego, USA*, 2013a.

Y. Chen, S. Trevezas, and P.-H. Cournède. A regularized particle filter EM algorithm based on Gaussian randomization with an application to plant growth modeling. *Submitted*, 2013b.

G. Claeskens and N. L. Hjort. *Model Selection And Model Averaging*. Cambridge Series In Statistical And Probabilistic Mathematics. Cambridge University Press, New York, 2008.

P.-H. Cournède, V. Letort, A. Mathieu, M.-Z. Kang, S. Lemaire, S. Trevezas, F. Houllier, and P. de Reffye. Some parameter estimation issues in functional-structural plant modelling. *Mathematical Modelling of Natural Phenomena*, 6(2):133–159, 2011.

P.-H. Cournède, Y. Chen, Q. Wu, C. Baey, and B. Bayol. Development and evaluation of plant growth models: Methodology and implementation in the PYGMALION platform. *Mathematical Modelling of Natural Phenomena*, 8:112–130, 2013.

D. Crisan, P. Del Moral, and T. Lyons. Discrete Filtering Using Branching and Interacting Particle Systems. *Markov Processes and Related Fields*, 5(3):293–318, 1998.

F. d'Alché Buc and N. Brunel. *Estimation of Parametric Nonlinear ODEs for Biological Networks Identification*, pages 61–96. 2010. MIT Press.

A. Davison and D. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, Cambridge, UK, 1997.

P. de Reffye, F. Blaise, S. Chemouny, T. Fourcaud, and F. Houllier. Calibration of hydraulic growth model on the architecture of cotton plants. *Agronomie*, 19:265–280, 1999.

P. de Reffye, M. Goursat, J. Quadrat, and B. Hu. The Dynamic Equations of the Tree Morphogenesis Greenlab Model. Technical Report 4877, INRIA, 2003.

P. de Valpine. Review of methods for fitting time-series models with process and observation error and likelihood calculations for nonlinear, non-Gaussian state-space models. *Bulletin of Marine Science*, 70(2):455–471, 2002.

P. de Valpine and A. Hastings. Fitting population models incorporating process noise and observation error. *Ecological Monographs*, 72(1):57–76, 2002.

P. Debashis, P. Jie, and B. Prabir. Semiparametric modeling of autonomous nonlinear dynamical systems with application to plant growth. *The Annals of Applied Statistics*, 5(3):2078–2108, 2011.

P. Del Moral. Nonlinear Filtering: Interacting Particle Resolution. *Markov Processes and Related Fields*, 2(4):555–580, 1996.

R. Delécolle, S. Maas, M. Guérif, and F. Baret. Remote sensing and crop production models: present trends. *ISPRS Journal of Photogrammetry and Remote Sensing*, 47 (23):145 – 161, 1992.

C. Deleuze. *Pour une dendrométrie fonctionnelle: essai sur l'intégration de connaissances écophysiologiques dans les modèles de production ligneuse*. PhD thesis, Univeristé Claude Bernard - Lyon I, 1996.

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, 27:94–128, 1999.

A. Dempster. The direct use of likelihood for significance testing. In *In Proceedings of Conference on Foundational Questions in Statistical Inference*, pages 335–352. Department of Theoretical Statistics, University of Aarhus, 1974.

A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 39:1–38, 1977.

B. Dennis, J. M. Ponciano, S. R. Lele, M. L. Taper, and D. F. Staples. Estimating density dependence, process noise, and observation error. *Ecological Monographs*, 76 (3):323–341, 2006.

L. Dente, G. Satalino, F. Mattia, and M. Rinaldi. Assimilation of leaf area index derived from ASAR and MERIS data into CERES-wheat model to map wheat yield. *Remote Sensing of Environment*, 112(4):1395 – 1407, 2008.

L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer Verlag New York, 2001.

S. Donnet, J.-L. Foulley, and A. Samson. Bayesian analysis of growth curves using mixed models defined by stochastic differential equations. *Biometrics*, 66(3):733–741, 2010.

W. Dorigo, R. Zurita-Milla, A. de Wit, J. Brazile, R. Singh, and M. Schaepman. A review on reflective remote sensing and data assimilation techniques for enhanced agroecosystem modeling. *International Journal of Applied Earth Observation and Geoinformation*, 9(2):165 – 193, 2007.

R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. *CoRR*, 2005.

A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer-Verlag, New-York, 2001.

M. M. Drugan and D. Thierens. Recombinative emcmc algorithms. In *Congress on Evolutionary Computation*, pages 2024–2031. IEEE, 2005.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC Monographs on Statistics and Applied Probability, 1994.

G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, (99(C5)): 10143–10162, 1994.

G. Evensen. *Data assimilation: The ensemble Kalman Filter*. Springer, 2006.

P. Fearnhead. Exact and efficient inference for multiple changepoint problems. *Statistics and Computing*, 16:203–213, 2006.

P. Fearnhead. MCMC for State-Space Models. In Brooks, Steve (ed.) et al., editor, *Handbook of Markov chain Monte Carlo.* Boca Raton, FL: CRC Press. Chapman, 2011.

J. M. Flegal and G. L. Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*, 38(2):1034–1070, Apr. 2010.

E. D. Ford and M. C. Kennedy. Assessment of uncertainty in functional-structural plant models. *Annals of Botany*, 108(6):1043–1053, 2011. doi: 10.1093/aob/mcr110.

G. Fort and E. Moulines. Convergence of the monte carlo expectation maximization for curved exponential families. *The Annals of Statistics*, 31(4):1220–1259, 2003.

J.-L. Foulley and D. A. Van Dyk. The PX-EM algorithm for fast stable fitting of Henderson's mixed model. *Genetics Selection Evolution*, 32(2):1–21, 2012.

J.-L. Foulley, F. Jaffrézic, and C. Robert-Granié. EM-REML estimation of covariance parameters in Gaussian mixed models for longitudinal data analysis. *Genetics Selection Evolution*, 32(2):129–141, 2000.

T. Fourcaud, X. Zhang, A. Stokes, H. Lambers, and C. Körner. Plant growth modelling and applications: The increasing importance of plant architecture in growth models. *Annals of Botany*, 101(8), 2008.

D. B. Fowler, A. E. Limin, and J. T. Ritchie. Low-temperature tolerance in cereals: Model and genetic interpretation. *Crop Science*, 39(3):626–633, 2003.

B. Gabrielle, R. Roche, P. Angas, C. Cantero-Martinez, L. Cosentino, M. Mantineo, M. Langen-Siepen, C. Hénault, P. Laville, B. Nicoullaud, and G. Gosse. A priori parameterisation of the ceres soil-crop models and tests against several European data sets. *Agronomie*, 22-2:119–132, 2002.

C. Gaucherel, F. Campillo, L. Misson, J. Guiot, and J. Boreux. Parameterization of a process-based tree growth model: Comparison of optimization, mcmc and particle filtering algorithms. *Environmental Modelling and Software*, 23(10-11):1280–1288, 2008.

A. E. Gelfand, S. E. Hills, A. Racine-Poon, and A. F. M. Smith. Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association*, 85(412):972–985, 1990.

A. Gelman, C. Robert, N. Chopin, and J. Rousseau. Bayesian data analysis, 1995.

A. Gelman, G. Roberts, and W. Gilks. Efficient Metropolis jumping rules. *Bayesian Statistics*, V:599–608, 1996.

A. Gelman, J. B. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis, 2nd edition.* London, Chapman & Hall, 2004.

D. Gelman, A.and Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511, 1992.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

C. Geyer. Markov chain Monte Carlo maximum likelihood. *Computing science and Statistics: proceedings of the 23rd Symposium on the interface*, pages 156–163, 1991.

C. Geyer. Practical Markov chain Monte Carlo (with discussion). *Statistical Science*, 7(4):473–482, 1992.

W. Gilks, G. Roberts, and E. George. Adaptive direction sampling. *Statistician*, (43): 179–189, 1994.

W. Gilks, G. Roberts, and S. Sahu. Adaptive Markov chain Monte Carlo through regeneration. *J Amer Stat Assoc*, (93):1045–1054, 1998.

N. Gordon, D. Salmond, and A. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Proc. Inst. Electr. Eng.*, Part F 140:107–113, 1993.

M. Guérif and C. Duke. Calibration of the SUCROS emergence and early growth module for sugar beet using optical remote sensing data assimilation. *European Journal of Agronomy*, 9:127–136, 1998.

M. Guérif and C. Duke. Adjustment procedures of a crop model to the site specific characteristics of soil and crop using remote sensing data assimilation. *Agriculture, ecosystems & environment*, 81(1):57–69, 2000.

M. Guérif, V. Houlès, D. Makowski, and C. Lauvernet. Data assimilation and parameter estimation for precision agriculture using the crop model stics. In D. Wallach, D. Makowski, and J. Jones, editors, *Working with Dynamic Crop Models*, pages 391–398. Elsevier, 2006.

Y. Guo, Y. Ma, Z. Zhan, B. Li, M. Dingkuhn, D. Luquet, and P. de Reffye. Parameter optimization and field validation of the functional-structural model Greenlab for maize. *Annals of Botany*, 97:217–230, 2006.

H. Haario, S. J., and J. Tamminen. Adaptive proposal distribution for random walk Metropolis algorithm. *Comput. Statist.*, 14:375–395, 1999.

H. Haario, S. J., and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7: 223–242, 2001.

H. Haario, M. Laine, A. Mira, and E. Saksman. DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16(4):339–354, 2006.

J. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

G. Hammer, M. Cooper, F. Tardieu, S. Welch, B. Walsh, F. Van Eeuwijk, S. Chapman, and D. Podlich. Models for navigating biological complexity in breeding improved crop plants. *Trends in Plant Science*, 11(12):587–593, 2006.

J. Harrison and M. West. *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, 1989.

J. Hartigan. Note on the confidence prior of Welch and Peers. *J. Roy. Statist. Soc. B*, 28:55–56, 1966.

W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

C. Heinrich. The mode functional is not elicitable. *Biometrika*, 101(1):245–251, 2014.

R. Hilborn and M. Ledbetter. Analysis of the British Columbia salmon purse-seine fleet: Dynamics of movement. *J. Fish. Res*, 36(4):384–391, 1979.

J. Hillier, D. Makowski, and B. Andrieu. Maximum likelihood inference and bootstrap methods for plant organ growth via multi-phase kinetic models and their application to maize. *Annals of Botany*, (96):137–148, 2005.

M. Y. Hirai, M. Yano, D. B. Goodenowe, S. Kanaya, T. Kimura, M. Awazuhara, M. Arita, T. Fujiwara, and K. Saito. Integration of transcriptomics and metabolomics for understanding of global responses to nutritional stresses in arabidopsis thaliana. *Proceedings of the National Academy of Sciences of the United States of America*, 101(27):10205–10210, 2004. doi: 10.1073/pnas.0403218101.

V. Houlès, B. Mary, M. Guérif, D. Makowski, and E. Justes. Evaluation of the ability of the crop model STICS to recommend nitrogen fertilisation rates according to agro-environmental criteria. *Agronomie*, 24:339–349, 2004.

P. J. Huber. The behavior of maximum likelihood estimation under nonstandard conditions. In L. M. L. Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, 1967. University of California Press.

C. M. Hurvich and C. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76:297–307, 1989.

I. A. Ibragimov and Y. V. Linnik. *Independent and Stationary Sequences of Random Variables*. Wolters-Noordhoff, Groningen, 1971.

A. Jasra, D. A. Stephens, and C. C. Holmes. On population-based simulation for static inference. *Statistics and Computing*, (3):263–279, 2007.

E. T. Jaynes. Confidence intervals vs Bayesian intervals. *Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science*, pages 175–258, 1976.

A. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.

C. Jones and J. Kiniry. *CERES-Maize : A simulation model of maize growth and development*. Texas A&M University Press, 1986.

G. Jones and J. Hobert. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*, 16:312–334, 2001.

G. Jones and J. Hobert. Sufficient burn-in for Gibbs samplers for a hierarchical random effects model. *The Annals of Statistics*, 32:784–817, 2004.

G. L. Jones, M. Haran, B. S. Caffo, and R. Neath. Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101: 1537–1547, 2006.

J. Jones and W. Graham. Application of extended and ensemble Kalman filters to soil carbon estimation. In D. Wallach, D. Makowski, and J. Jones, editors, *Working with Dynamic Crop Models*, pages 55–100. Elsevier, 2006.

S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *International Symposium of Aerospace/Defense Sensing, Simulation and Controls*, 1997. Orlando. Fl.

S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transaction on Automatic Control*, 45(3):477–482, 2000.

A. Jullien, J.-M. Allirand, A. Mathieu, B. Andrieu, and B. Ney. Variations in leaf mass per area according to nitrogen nutrition, plant age, and leaf position reflect ontogenetic plasticity in winter oilseed rape (brassica napus l.). *Field Crops Research*, 114(2):188 – 197, 2009.

T. Kailath, A. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey, 2000.

J. Kaipio and E. Somersalo. *Statistical and Computational Inverse problems*. Applied Mathematical Science. Springer Science, 2005.

R. Kalman. A new approach to linear filtering and prediction problem. *Journal of the basic engineering*, 82:35–45, 1960.

B. Keating, P. Carberry, G. Hammer, M. Probert, M. Robertson, D. Holzworth, N. Huth, J. Hargreaves, H. Meinke, Z. Hochman, G. McLean, K. Verburg, V. Snow, J. Dimes, M. Silburn, E. Wang, S. Brown, K. Bristow, S. Asseng, S. Chapman, R. McCown, D. Freebairn, and C. Smith. An overview of APSIM, a model designed for farming systems simulation. *European Journal of Agronomy*, 18(3-4):267–288, 2003.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

J. R. Koehler and A. Owen. Computer experiments. In *Handbook of statistics 13*. 1996.

A. Kong, J. Liu, and W. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

M. Lamboni, H. Monod, and D. Makowski. Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliability Engineering & System Safety*, 96(4):450–459, 2011.

K. B. Laskey and J. Myers. Population Markov chain Monte Carlo. In *Machine Learning*, pages 175–196. University Press, 2003.

M. Launay and M. Guérif. Assimilating remote sensing data into a crop model to improve predictive performance for spatial applications. *Agriculture, Ecosystems & Environment*, 111:321–339, 2005.

F. Le Gland and N. Oudjane. Stability and uniform approximation of nonlinear filters using the Hilbert metric and application to particle filters. *Ann. Appl. Probab.*, 14 (1):144–187, 2004. doi: 10.1214/aoap/1075828050.

F. Le Gland, C. Musso, and N. Oudjane. An Analysis of Regularized Interacting Particle Methods for Nonlinear Filtering. In *Proceedings of the 3rd IEEE European Workshop on Computer-Intensive Methods in Control and Signal Processing*, pages 167–174, 1998.

J. Lecoeur, R. Poiré-Lassus, A. Christophe, B. Pallas, P. Casadebaig, P. Debaeke, F. Vear, and L. Guilioni. Quantifying physiological determinants of genetic variation for yield potential in sunflower. sunflo: a model-based analysis. *Functional Plant Biology*, 38:246–259, 2011.

S. Lemaire, F. Maupas, P.-H. Cournède, and P. de Reffye. A morphogenetic crop model for sugar-beet (*beta vulgaris l.*). In *International Symposium on Crop Modeling and Decision Support: ISCMDS 2008, April 19-22, 2008, Nanjing, China*, 2008.

S. Lemaire, F. Maupas, P.-H. Cournède, J.-M. Allirand, P. de Reffye, and B. Ney. Analysis of the density effects on the source-sink dynamics in sugar-beet growth. In B.-G. Li, M. Jaeger, and Y. Guo, editors, *3rd international symposium on Plant Growth and Applications(PMA09), Beijing, China*. IEEE Computer Society (Los Alamitos, California), November 9-12 2009.

R. A. Levine and G. Casella. Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

J. Liu. *Fraction of missing information and convergence rate of data augmentation*. Interface Foundation of North America, Fairfax Station, VA, 1994.

J. Liu, W. Wong, and A. Kong. Covariance structure and convergence rate of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81:27–40, 1994.

C. Loi. *Analyse probabiliste, étude combinatoire et estimation paramétrique pour une classe de modèles de croissance de plantes avec organogenèse stochastique*. PhD thesis, Ecole Centrale Paris, 2011.

C. Loi, P.-H. Cournède, and S. Trevezas. Bayesian estimation in Functional-Structural Plant Models with stochastic organogenesis. In *Proceedings of ASMDA*, 2011.

D. Ludwig and C. Walters. Measurement errors and uncertainty in parameter estimates for stock and recruitment. *Canadian Journal of Fisheries and Aquatic Sciences*, 38 (6):711–720, 1981.

S. J. Maas. Using satellite data to improve model estimates of crop yield. *Agronomy Journal*, 80(4):655–662, 1988.

J. Mailhol, A. Olufayo, and P. Ruelle. Sorghum and sunflower evapotranspiration and yield from simulated leaf area index. *Agri. Water Manag.*, 35:167–182, 1997.

D. Makowski, D. Wallach, and M. Tremblay. Using a Bayesian approach to parameter estimation; comparison of the GLUE and MCMC methods. *Agronomie*, 22(2):191–203, 2002.

D. Makowski, J. Jeuffroy, and M. Guérif. *Bayesian methods for updating crop-model predictions, applications for predicting biomass and grain protein content.* Wageningen UR Frontis Series, 2004.

D. Makowski, J. Hillier, D. Wallach, B. Andrieu, and M.-H. Jeuffroy. Parameter estimation for crop models. In D. Wallach, D. Makowski, and J. Jones, editors, *Working with Dynamic Crop Models*, pages 55–100. Elsevier, 2006.

M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.

J. P. Mattern, M. Dowd, and K. Fennel. Particle filter-based data assimilation for a three-dimensional biological ocean model and satellite observations. *J. Geophys. Res. Oceans*, 118(5):2746–2760, 2013.

C. E. McCulloch. Maximum likelihood variance components estimation for binary data. *Journal of the American Statistical Association*, 89(425):330–335, 1994.

C. E. McCulloch. Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American statistical Association*, 92(437):162–170, 1997.

G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* John Wiley & Sons Inc., 2008.

X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika,*, 80:267–278, 1993.

K. L. Mengersen and C. P. Robert. *Population Markov chain Monte Carlo: the pinball sampler.* in Bayesian Statistics, Oxford University Press, 2003.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21(1087-1091), 1953.

C. Meza, F. Jaffrézic, and J.-L. Foulley. REML estimation of variance parameters in nonlinear mixed effects models using the SAEM algorithm. *Biometrical journal. Biometrische Zeitschrift*, 49(6):876–88, 2007.

R. Mittler. Abiotic stress, the field environment and stress combination. *Trends in Plant Science*, 11(1):15 – 19, 2006.

H. Monod, C. Naud, and D. Makowski. Uncertainty and sensitivity analysis for crop models. In D. Wallach, D. Makowski, and J. Jones, editors, *Working with Dynamic Crop Models*, pages 55–100. Elsevier, 2006.

J. Monteith. Climate and the efficiency of crop production in Britain. *Proceedings of the Royal Society of London B*, 281:277–294, 1977.

M. Moran, Y. Inoue, and E. Barnes. Opportunities and limitations for image-based remote sensing in precision crop management. *Remote Sensing of Environment*, 61 (3):319 – 346, 1997.

S. Moulin, A. Bondeau, and R. Delecolle. Combining agricultural crop models and satellite observations: From field to regional scales. *International Journal of Remote Sensing*, 19(6):1021–1036, 1998. doi: 10.1080/014311698215586.

L. M. Murray. Distributed Markov chain Monte Carlo. In *NIPS Workshop: Learning on Cores, Clusters and Clouds*, 2010.

C. Musso and N. Oudjane. Regularization schemes for branching particle systems as a numerical solving method of the nonlinear filtering problem. In *Proceedings of the Irish Signals and Systems Conference*, 1998.

C. Musso, N. Oudjane, and F. Le Gland. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice. New York*.

P. Mykland, L. Tierney, and B. Yu. Regeneration in Markov chain samplers. *Journal of the American Statistical Association*, 90:233–241, 1995.

C. Naud, D. Makowski, and M.-H. Jeuffroy. Application of an interacting particle filter to improve nitrogen nutrition index predictions for winter wheat. *Ecological Modelling*, 207(24):251 – 263, 2007.

N. Oudjane and C. Musso. Regularized particle schemes applied to the tracking problem. In *International Radar Symposium, Munich, Proceedings*, 1998.

N. Oudjane and C. Musso. Multiple model particle filter. In *17ème Colloque GRETSI, Vannes 1999*, pages 681–684, 1999.

E. Parzen. On Estimation of a Probability Density Function and Mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

D. Pham. Stochastic Methods for Sequential Data Assimilation in Strongly Nonlinear Systems. *Monthly Weather Review*, 129(5):217–244, 2001.

T. Polacheck, R. Hilborn, and A. Punt. Fitting surplus production models: Comparing methods and measuring uncertainty. *Can. J. Fish. Aquat. Sci.*, 50:2597–2607, 1993.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

M. Quach, N. Brunel, and F. d'Alché Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23(23):3209–3216, 2007.

L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–284, 1989.

L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, 1986.

C. Rao. *Linear Statistical Inference and Its Applications*. Wiley, New York, 1973.

A. Rau, F. Jaffrézic, J.-L. Foulley, and R. Doerge. An empirical Bayesian method for estimating biological networks from temporal microarray data. *Statistical Applications in Genetics and Molecular Biology*, 9(1):1–28, 2010.

A. Rau, F. Jaffrézic, J.-L. Foulley, and R. Doerge. Reverse Engineering Gene Regulatory Networks Using Approximate Bayesian Computation. *Statistics and Computing*, 22:1257–1271, 2011.

R. H. Reichle, P. W. Jeffrey, D. K. Randal, and R. H. Paul. Extended versus ensemble Kalman filtering for land data assimilation. *J. Hydrometeorol.*, 3(6):728–740, 2002.

B. Ripley. *Stochastic Simulation*. John Wiley, 1988.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

C. Robert. *The Bayesian Choice*. Springer Texts in Statistics Series. Springer-Verlag, 2001.

C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics Series. Springer-Verlag GmbH, 1999.

C. Robert, T. Rydén, and D. Titterington. Convergence controls for MCMC algorithms, with applications to hidden Markov chains. *Journal of Statistical Computation and Simulation*, 64:327–355, 1999.

C. P. Robert and G. Casella. *Introducing Monte Carlo methods with R*. Springer, New York, 1 edition, 2010.

G. Roberts. Markov chain concepts related to sampling algorithms. *Markov Chain Monte Carlo in Practice*, 7:45–57, 1996.

G. Roberts and J. Rosenthal. *Statistical Science*.

G. Roberts and J. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.

G. Roberts and S. K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *Journal of the Royal Statistical Society, Series B*, 59:291–317, 1997.

G. Roberts and R. Tweedie. *Understanding MCMC*. Springer, 2008.

G. Roberts, A. Gelman, and W. Gilks. *The annals of applied probability*.

J. S. Rosenthal. Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 90:558–566, 1995.

V. Rossi. *Filtrage non linéaire par noyaux de convolution: Application à un procédé de dépollution biologique*. PhD thesis, Ecole National Supérieure Agronomique de Montpellier, 2004.

V. Rossi and J.-P. Vila. Nonlinear filtering in discrete time: A particle convolution approach. *Ann. Inst. Stat. Univ. Paris*, 3:71–102, 2006.

D. Rubin. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics*, 12(4):1151–1172, 1984.

S. Sahu and A. A. Zhigljavsky. Self regenerative Markov chain Monte Carlo with adaptation. *Bernoulli*, 9:395–422, 2003.

A. Saltelli, K. Chan, and E. M. Scott. *Sensitivity Analysis*. Wiley, 2000.

A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. John Wiley & Sons, 2004.

R. L. J. Schnute, Jon T. The influence of error on population estimates from catch-age models. *Canadian Journal of Fisheries and Aquatic Sciences*, 52(10):2063–2077, 1995.

G. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, 1978.

S. L. Scott. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97:337–351, 2002.

S. R. Searle, G. Casella, and C. E. McCulloch. *Variance components*. Wiley & Sons, N. Y., 1992.

N. Shephard and M. K. Pitt. Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84:653–667, 1997.

B. Silverman. *Density Estimation*. Chapman and Hall, London, 1986.

A. Smith and G. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *J. Roy. Statist. Soc. Ser.*, B(55):3–24, 1993.

I. Sobol. Sensitivity analysis for non-linear mathematical models. *Mathematical Modeling and Computational Experiment*, 1:407–414, 1993.

H. Sorenson. *Kalman Filtering: Theory and Applications*. IEEE Press, 1985.

D. J. Spiegelhalter, N. G. Best, and B. P. Carlin. Bayesian deviance, the effective number of parameters, and the comparison of arbitrarily complex models. Technical report, 1998.

C. Spitters, H. Van Keulen, and D. Van Kraalingen. *A simple and universal crop growth simulator: SUCROS87*. PUDOC, Wageningen, 1989.

G. Storvik. Particle filters in state space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289, 2002.

M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. with discussion and with a reply by the authors. *Journal of the American Statistical Association*, 82(398):528–550, 1987.

F. Tardieu. Virtual plants: modelling as a tool for the genomics of tolerance to water deficit. *Trends in Plant Science*, 8(1):9–14, 2003.

W. Taylor. Small sample properties of a class of two-stage Aitken estimator. *Econometrica*, 45(2):497–508, 1977.

C. J. Ter Braak. A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. *Stat Comp*, 16:239–249, 2006.

C. J. Ter Braak and J. A. Vrugt. Differential evolution Markov chain with snooker updater and fewer chains. *Stat Comp*, 18:435–446, 2008.

S. Trevezas and P.-H. Cournède. A sequential Monte Carlo approach for MLE in a plant growth model. *Journal of Agricultural, Biological, and Environmental Statistics*, 18 (2):250–270, 2013.

S. Trevezas, S. Malefaki, and P.-H. Cournède. Simulation techniques for parameter estimation via a stochastic ECM algorithm with application to plant growth modeling. Preprint, http://hal.archives-ouvertes.fr/hal-00798695, 2013.

P. Van Leeuwen and G. Evensen. Data assimilation and inverse methods in terms of a probabilistic formulation. *Monthly Weather Review*, 124:2898–2913, 1996.

H.-V. Varella, M. Guerif, S. Buis, and N. Beaudoin. Soil properties estimation by inversion of a crop model and observations on crops improves the prediction of agro-environmental variables. *European Journal of Agronomy*, 33(2):139–147, 2010.

J. A. Vrugt, C. J. ter Braak, C. G. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon. Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):273–290, 2009a.

J. A. Vrugt, C. J. ter Braak, H. V. Gupta, and B. A. Robinson. Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling? *Stochastic Environmental Research and Risk Assessment*, 23(7):1011–1026, 2009b.

D. Wallach, B. Goffinet, J. Bergez, P. Debaeke, D. Leenhardt, and J. Aubertot. The effect of parameter uncertainty on a model with adjusted parameters. *Agronomie*, 22:159–170, 2002.

D. Wallach, D. Makowski, and J. Jones. *Working with Dynamic Crop Models: Evaluation, Analysis, Parameterization, and Applications*, chapter Evaluating crop models, pages 11–53. Elsevier Science Ltd, 2006.

E. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. *IEEE Symposium 2000, Lake Louise, Alberta, Canada*, 2000.

Y. Wang, Y. Wang, G. Wahba, and G. Wahba. Bootstrap confidence intervals for smoothing splines and their comparison to Bayesian 'confidence intervals'. *J. Statist. Comput. Simulation*, 51:263–279, 1994.

G. Wei and M. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704, 1990.

H. Wernsdörfer, V. Rossi, G. Cornu, S. Oddou-Muratorio, and S. Gourlet-Fleury. Impact of uncertainty in tree mortality on the predictions of a tropical forest dynamics model. *Ecological Modelling*, 218(3):290–306, 2008.

M. West. Approximating posterior distribution by mixtures. *Journal of Royal Statistical Society*, B(55):409–442, 1993.

Q. Wu, J. Bertheloot, A. Mathieu, B. Andrieu, and P.-H. Cournède. Assessment of non-linearity in functional-structural plant models. In T. De Jong, J. Vos, and A. Escobar, editors, *6th international workshop on Functional-Structural Plant Models (FSPM10), Davis, USA*, November 9-12 2010.

Q. Wu, P.-H. Cournède, and A. Mathieu. An efficient computational method for global sensitivity analysis and its application to tree growth modelling. *Reliability Engineering & System Safety*, 107:35–43, 2012.

P. Wyckoff and J. Clark. Predicting tree mortality from diameter growth: a comparison of maximum likelihood and Bayesian approaches. *Canadian Journal of Forest Research*, 30(1):156–167, 2000.

H. Yan, M. Kang, P. De Reffye, and M. Dingkuhn. A dynamic, architectural plant model simulating resource-dependent growth. *Annals of Botany*, 93:591–602, 2004.

X. Yin and P. C. Struik. Modelling the crop: from system dynamics to systems biology. *Journal of Experimental Botany*, 61(8):2171–2183, 2010. doi: 10.1093/jxb/erp375.

P. Yiou. Anawege: a weather generator based on analogues of atmospheric circulation. *Geoscientific Model Development*, 7(2):531–543, 2014.

Z. Zhan, P. de Reffye, F. Houllier, and B. Hu. Fitting a structural-functional model with plant architectural data. In B. Hu and M. Jaeger, editors, *Plant Growth Models and Applications*, pages 236–249. Tsinghua University Press and Springer (Beijing, China), 2003.