Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Non parametric Bayesian belief nets (NPBBNs) versus ensemble Kalman filter (EnKF) in reservoir simulation

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

MARIA GHEORGHE

Delft, the Netherlands
August 2010

Delft University of Technology

MSc THESIS APPLIED MATHEMATICS

"Non parametric Bayesian belief nets (NPBBNs) versus ensemble Kalman filter (EnKF) in reservoir simulation"

MARIA GHEORGHE

Delft University of Technology

Daily supervisor | **Responsible professor**

Dr. Anca Hanea | Prof. Dr. Roger Cooke

Other thesis committee members

Dr. Remus Hanea

MSc. Mariya Krymskaya

August, 2010 Delft, the Netherlands

TABLE OF CONTENTS

Chapter 1

Introduction

In the early days of oil and gas recovery, reservoir engineering was simple: one would simply drill a hole, and at some pressure, there will be oil coming out. The depths of the wells were really modest and so the cost of drilling such holes was low. But, in this way, just a small fraction of oil can be recovered (5-15%). This is called “primary recovery” [25].

Nowadays much of the oil is offshore and the holes are drilled to some extreme depths. Of course that in these conditions the costs of drilling a well are much higher. It becomes of primary interest to produce as much as possible from the surface oil. Water or gas can be injected to increase the pressure and to push the oil out. In this way around 30-60% of the oil can be recovered [25]. Even more sophisticated techniques are used for attaining an ultimate recovery of 70-80%. For recovering very heavy oil, one can inject chemicals that make the oil less viscous so that it flows more easily. Another quite advanced recovery technique uses air injection.

Given the above, it is very important to be able to predict the flow of fluids like oil, water and gas in a reservoir. Reservoir simulation is an area of reservoir engineering that uses computer models to predict the flow of fluids (i.e. oil, water and gas) through a porous media. Reservoir simulation models are used by oil and gas companies in the development of new fields. Also, models are used in developed fields where production forecasts are needed to help make investment decisions.

When simulating a reservoir, one must account for the physical and chemical processes taking place in the subsurface. Rocks and fluids properties are very important when describing the flow in porous media. We are interested in the volume of oil present in the reservoir and its ease of flow. The emphasis will be on estimating the rock's property called *permeability*. Permeability is a measure of the ease of flow of a fluid through porous media. The problem of estimating models' parameters like permeability is often referred to as the *history matching problem* in reservoir engineering.

History matching matches the history of observable variables at wells with predictions of these variables from a numerical model by adjusting parameters like permeability. There are two main ways of correcting the parameters of the model when history matching is performed. The traditional way is to run the reservoir model for some time and then modify the parameters such that the measured and the predicted data match. This is called *manual history matching*. On the other hand, one can use computers to adjust the parameters. Therefore, *computer assisted history matching* is an alternative to the traditional history matching approach. In this thesis, the focus will be on the latter approach.

One methodology that addresses the computer assisted history matching is *data assimilation*. Data assimilation combines a mathematical physical model with available measurements in order to estimate and predict different environmental processes. There are two classes of methods in data assimilation: the variational approach and the sequentional approach.

In the variational approach an objective function defined as the distance between the measured data and the forecasts from a model, is minimized by adjusting parameters. The variational data assimilation is a constraint minimization problem, which is solved iteratively with a gradient based optimization method. The gradients are obtained using a so called adjoint method. The technique has been used when approaching a history matching problem in reservoir simulation, e.g. [26], [27].

The sequential approach includes Kalman filtering [5], a powerful tool for solving linear systems. The recently developed adaptation of the Kalman filter for handling non linear systems is called *the ensemble Kalman filter (EnKF)* and was introduced in [6].

EnKF is one of the currently most used data assimilation methods for approaching a computer assisted history matching problem in reservoir simulation, e.g. [13, 14, 15]. It is a Monte Carlo approach in which the probability distribution of the estimate is represented by an ensemble of possible realizations. The size of the ensembles used in EnKF is of great importance for the algorithm. The computational effort associated with the EnKF algorithm is proportional to the number of ensembles used. Therefore, one should use as few ensembles as possible. Moreover, the choice of the ensemble size is essential for the performance of the algorithm. When the size of the ensembles used is too small the algorithm becomes inaccurate. In this context, the main disadvantage of the EnKF appears when the number of variables to be estimated is much larger than the ensembles size. Typically for ‘real’ reservoir applications the size of the state vector can be as large as 3×10^6. In general between 50 and 100 ensembles are used in reservoir engineering applications [e.g. 26, 27]. This leads to sampling errors which make the approximated covariance matrix to be underestimated. Unfortunately this is often the case because ensemble Kalman filter is a Monte Carlo method and for being affordable a relatively small number of ensembles has to be used. In this situation we notice the presence of unrealistic (often called spurious) correlations between large distance grid points. Therefore, observations have high influence over large distances, which is not physically true. The process of eliminating these unrealistic correlations is called *localization* [8, 9, 10, 11, 12]. In this thesis we are using ensemble Kalman filter with localization. However, sometimes localization introduces inconsistencies in the system, i.e. while trying to eliminate unrealistic correlations one might end up eliminating real correlations also.

Given the above limitations we shall focus on a method that uses graphical models to approach a history matching problem in reservoir engineering.

High dimensional probabilistic modeling using graphical models has been used in several scientific fields like: statistics, biology, physics and engineering. Because of their flexibility, the use of graphical models has increased over time; hence the theory behind them has been constantly developed and extended. They combine probabilistic theory and graph theory to provide a general setting for models in which a number of variables interact. The graphical structure is represented by a collection of nodes and links. Each node in the graph represents a random variable. The links represent the qualitative dependencies between the variables.

There are two main types of graphical models: directed and undirected. The focus will be on the directed acyclic graphs called *Bayesian belief nets (BBNs)*.

The novelty from this thesis consists in applying a BBN based approach to estimate the permeability of a reservoir's field.

A BBN model describes the probability density function of a set of variables by specifying a number of conditional independence statements represented by a graph, and a set of probability density functions. BBNs are often chosen to model high dimensional probabilistic distributions because of their capacity of representing cause effect relationships between variables through the directionality of the arcs. The nodes of a BBN can be discrete and/or continuous random variables. The approach from this thesis models non-parametric continuous variables. This approach was already used in several applications. One project that uses non parametric Bayesian belief nets (NPBBNs) is CATS, which stands for *Causal Model for Air Transport Safety*. This is a large scale application that models the risks from the aviation industry.

Another project that uses NPBBNs is *Beneris,* which stands *Benefit and Risk* and it focuses on the analysis of health risks and benefits associated with food consumption.

To our knowledge this is the first attempt to apply NPBBNs to approach a history matching problem in reservoir simulation. We shall use static and dynamic NPBBNs. A static NPBBN is suitable for a static (non time dependent) data set. One can interpret a dynamic NPBBN as instances of a static NPBBN connected in discrete slices of time. The data provided by the simulator at a given time will be represented as a static NPBBN. Production data will be used to update the joint distribution of the variables of interest. Data will be assimilated at every time step by conditioning the joint distribution on the values of the measurements. The new joint distribution will be modified further in time using the reservoir model. In this way the connection between discrete slices of time will be provided by the reservoir model.

Moreover, a method for learning the structure and the parameters of a static NPBBN model from a data set will be used. This method was introduced in [6]. By using it, one can eliminate from the graph small correlations that might be representing just sampling fluctuations. In this way a dynamic NPBBN with changed structure over time will be obtained. Note the similarity between using a method for mining a database when applying a NPBBN based approach and using localization techniques when applying the EnKF. Given the above, the NPBBN based approach will be presented in a comparative way with the ensemble Kalman filter with localization.

This thesis is organized as follows. Chapter 2 presents the chemical and physical processes taking place in a reservoir. The most important rocks and fluids properties are introduced along with a two phase flow mathematical physical model. A short review of the history matching techniques and the Kalman filter methods is presented. The drawbacks of the Kalman filter methods are discussed and this provides the motivation of introducing the Bayesian belief net based approach.

Chapter 3 describes the Bayesian belief nets. Both *static* and *dynamic BBNs* are illustrated. A review of the existing Bayesian belief nets methods for discrete and continuous variables is presented. The focus is on *NPBBN*. The data mining procedure introduced in [6] will be presented. The connection between Kalman filter methods and the dynamic BBN will be made.

Chapter 4 shows the experimental setup necessary to perform the case study from chapter 5. The thesis uses a twin experiment; i.e. we assume the truth is known. Of course this kind of situation is not encountered in real life but is often used in synthetic applications for testing the performance of the method(s) used. We shall present the choice of our initial ensembles, the way we generate synthetic measurements, and the measures of performance. Moreover, one of the methods uses NPBBN. Since NPBBN are directed acyclic graphs, they represent influences between variables as arcs. Therefore, the directionality of the arcs is an important feature of the graph. In chapter 4 we shall also illustrate the choice of the arcs directionality in the non parametric Bayesian belief nets.

Chapter 5 presents the results of estimating the permeability field using the ensemble Kalman filter with localization and the continuous non parametric Bayesian belief net approach.

Finally, Chapter 6 formulates conclusions and recommendations.

Chapter 2

Reservoir simulation

Reservoir simulation describes the important physical and chemical processes taking place in a reservoir in order to make realistic predictions of a reservoir behavior. As described in [2] the construction of a reservoir model requires four major steps. First, a physical model of the flow process is developed incorporating the necessary physics to describe the phenomena. Second, a mathematical formulation of the physical model is obtained, usually involving systems of nonlinear partial differential equations. Third, a discretized numerical model of the mathematical model is produced. Finally a computer program capable of efficiently performing the necessary computations for the numerical model is used. Generally, the modeling process is not complete with one pass through these four steps. Once a computer program is developed, the output is compared with the measured observations of the physical process. If the results do not compare, one should go back and iterate through the complete modeling process, changing some intermediate steps for obtaining a better correlation between the physical observations and the computational results.

This chapter is organized as follows: section 1 gives an overview of the most important physical and chemical processes taking place in a reservoir, section 2 presents the physical equations of the two phase flow model as illustrated in [1], sections 3 and 4 introduce history matching techniques.

2.1 Reservoir properties

In order to understand the development of a physical model in the context of reservoir simulation, a brief description of the reservoir rocks and fluids flow in porous media is needed.

In general, the hydrocarbon is trapped in the microscopic pores of a rock, like sandstone, and will flow through the rock only under the influence of large pressure. The pores are formed as spaces between the sandstone. A large percentage of the pores are connected and the fluids can flow through these paths. The ratio of the volume of these pore paths available to flow to the total volume of the rock is called porosity (ϕ) and is quite small (typically 1-20%). Permeability (k) is another rock property as important as porosity because not only the actual volume of oil in reservoir is important, but also the rate at which the hydrocarbon will flow through the reservoir. Permeability is a measure of the ease of flow of a fluid through a porous medium. Viscosity (μ) is a fluid's characteristic that also contributes to the ease of fluid's movement. The less viscous the fluid, the greater the ease of flow is. The type of rock present in a reservoir influences greatly the ability of the hydrocarbon to flow through a reservoir. Since the geology of the reservoir changes, we distinguish areas of low flows and areas of high flows in a reservoir.

Normally, we have more than one fluid present in oil reservoirs. From the history of the formation of the petroleum reservoirs, it is thought that most of the reservoir is of marine origin, the pores of the rock were initially field with water. Then, the oil and/or gas moved into the reservoir displacing water. Hence a reservoir can contain water, oil and gas. The extent of occupancy of the pore spaces by a particular fluid is called saturation (S). Fluid saturation is defined as the fraction, or percent, of the total pore space occupied by a particular fluid.

The oil will flow out from the reservoir only when the pressure (p) is high. Wells are drilled into the surface of the reservoir in order to push the oil out. In order to keep the pressure high in the reservoir, a fluid (generally water) is injected into so-called injector wells. As a result the oil is pumped out through other wells, called producers.

2.2 The two phase flow model

Fluids flows in porous media are governed by the same fundamental laws that govern their flow in rivers or pipelines. These laws are based on the conservation of momentum and mass.

We shall assume that we have two phases of oil and water flowing simultaneously, that the fluids are immiscible, i.e. the fluids do not mix and that there is no mass transfer between the fluids. Water will wet the porous medium more than oil and will be called the *wetting phase fluid*. The subscript convention will be *w* for water and *o* for oil. Since both phases are flowing we have:

$$S_w + S_o = 1 \tag{2.1}$$

For each phase we have different pressures, we denote water pressure by p_w and oil pressure by p_o. The difference between these pressures is the capillary pressure p_c. We shall assume that the capillary pressure is a function of water saturation:

$$p_o - p_w = p_c\left(S_w\right) \tag{2.2}$$

A reservoir has isothermal conditions if capillary pressure is zero, i.e. $p_c\left(S_w\right)=0$, then the pressures for each fluid phase are equal: $p_o = p_w$.

Conservation of momentum in flow through porous media is usually expressed with Darcy's law. The French engineer, Henry Darcy, developed a fluid flow equation which has become one of the standard mathematical tools in petroleum engineering. The law for one fluid flow is defined as:

$$v = -\frac{k}{\mu}\frac{\partial p}{\partial x}, \tag{2.3}$$

where $v\left[m/\sec\right]$ is the apparent fluid velocity; $k\left[m^2\right]$ is the absolute permeability (generally permeability is measured in darcies but we use 1 darcy$=10^{-12}m^2$ as in [1]); μ $\left[Pa\cdot s\right]$ is the viscosity of the flowing fluid; $\frac{\partial p}{\partial x}$ $\left[atmospheres/cm\right]$ describes the pressure drop in *x*-direction. The absolute permeability represents the ability to transmit fluids through rock when a single fluid is present in the rock. When simultaneously two or three fluids flow the concept of relative permeability associated with each phase is introduced. A more detailed description of relative permeability will be given later in this chapter. The velocity, v, is not the actual velocity of the flowing fluid, but the apparent velocity also called Darcy's velocity determined by dividing the flow rate q to the cross sectional area denoted by A. Substituting v with the relationship q/A, equation (2.3) becomes:

$$q = -\frac{kA}{\mu}\frac{\partial p}{\partial x} \tag{2.4}$$

Darcy's law states that the volumetric flow rate of a fluid through a porous medium is proportional to the pressure or hydraulic gradient and to the cross-sectional area and inverse proportional to the viscosity of the fluid.

For two fluids flow we can write the same equation in vector form as[1]:

[1] We use ~ above a vector or matrix to indicate that its components are representing quantities in physical space, e.g. $\tilde{v}$ is a velocity vector with one, two or three components, depending on whether we use a one, two or three dimensional reservoir simulation.

$$\tilde{v}_i = -\frac{k_{ri}}{\mu_i}\tilde{k}\left(\nabla p_i - \rho_i g \nabla d\right), \qquad i \in \{w,o\} \tag{2.5}$$

where ∇ is the gradient operator for one two or three dimensions, k_{ri} is the relative permeability corresponding to each phase flow, $g\left[m/s^2\right]$ is the magnitude of the acceleration due to gravity, the depth $d\left[m\right]$ is a vector function of (x,y,z) pointing out in the direction of gravity. In most cases of Darcy's law in reservoir simulation, $\tilde{k}$ is assumed a special diagonal tensor. For the *1-D*, *2-D*, *3-D* cases the diagonal tensor is defined as:

$$1-D:\tilde{k}=k \qquad 2-D:\tilde{k}=\begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \qquad 3-D:\tilde{k}=\begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}, \tag{2.6}$$

where k_x , k_y , k_z are interpreted as permeabilities in the *x*, *y* and *z* directions. If $k_x = k_y = k_z$ the permeability equals in all directions at a specified point and the medium is called *isotropic*; otherwise it is called *anisotropic*.

In the two phase flow each fluid interferes with the flow of the other and the relative permeability for each phase is less than or equal to the absolute permeability of the porous medium. Relative permeability for each phase is defined as follows:

$$k_{ri} = \frac{k_i}{\tilde{k}}, \qquad i \in \{w,o\} \tag{2.7}$$

where k_{ri} denotes the relative permeability for water, oil respectively.

Relative permeability shows how much the permeability of a particular phase has been influenced by the presence of another phase. For example for the two phase flow, a relative permeability to oil of 70 percent shows that the oil permeability has been reduced 30 percent as the results of the presence of the wet phase. The lower limit of relative permeability is zero when the saturation of the phase is zero, and the upper limit of relative permeability is *100* percent when the saturation is *100* percent and $k_w = k_o = k$. We say that fluids are miscible when they flow as one phase. In this situation there is no impact between phases, hence $k_w = k_o = k$.

In experiments, relative permeability is often represented as a function of water saturation. Corey model gives the relation between relative permeability and water saturation. Let us define the following equations:

$$k_{rw} = k_{rw}^0 S^{n_w} \tag{2.8}$$

$$k_{ro} = k_{ro}^0 \left(1 - S\right)^{n_o}, \tag{2.9}$$

where k_{rw}^0 , k_{ro}^0 are the end points relative permeability for water, respectively oil, n_w, n_o are the Corey exponents for water and oil and S is scaled saturation defined as:

$$S = \frac{S_w - S_{wc}}{1 - S_{or} - S_{wc}} \qquad 0 \le S \le 1, \tag{2.10}$$

where S_{wc} is the connate water saturation and S_{or} is the residual oil saturation.

For defining the equations that govern the two phase flow of a fluid through porous media we also need to describe the law of conservation of mass. The conservation of mass per unit time says that the difference between mass in and mass out of a system $\nabla \bullet \left(\alpha\rho_i \tilde{v}_i\right)\left[kg / m^3 \cdot s\right]$ must be equal to the mass accumulated within the system $\alpha \frac{\partial\left(\rho_i \Phi S_i\right)}{\partial t}\left[kg / m^3 \cdot s\right]$ minus the mass added or extracted through an external source (well) $\alpha\rho q_i^{'''}[Kg / m^3 \cdot s]$. For the two phases fluid model the mass balance equations for each phase can be expressed as:

$$\nabla \bullet \left(\alpha\rho_i \tilde{v}_i\right) + \alpha \frac{\partial\left(\rho_i \Phi S_i\right)}{\partial t} - \alpha\rho q_i^{'''} = 0, \qquad i \in \left\{w, o\right\} \tag{2.11}$$

where $\nabla \bullet$ is the divergence operator (gradient operator), $q_i^{'''}$ is the source term expressed as flow rate per unit volume, t is time, α is a geometric factor that changes with the dimension of the flow. In 1-D flow, i.e. the movement is considered only in one direction, α is equal to the cross space sectional area of the reservoir. In 2-D flow when the movement is considered in two directions: horizontal and vertical, α is equal to the reservoir height. For the 3-D case, the movement is considered in all directions, therefore α is equal to 1.

Combining the equations from Darcy's law for each phase, i.e. equations (2.5), with the equations from the conservation of mass (2.11) yields:

$$-\nabla \bullet \left[\frac{\alpha \rho_i k_{ri}}{\mu_i} \tilde{k} \left(\nabla p_i - \rho_i g \nabla d \right) \right] + \alpha \frac{\partial \left(\rho_i S_i \Phi \right)}{\partial t} - \alpha \rho_i q_i^{'''} = 0, \qquad i \in \{w, o\} \quad (2.12)$$

Equations (2.12) contain four unknowns: p_w, p_o, S_w and S_o. We assume the absence of capillary pressures then $p_w = p_o$ and by using relation (2.1) two unknowns can be eliminated. We can express pressure as a function of density or vice versa. This is accomplished by the use of an equation that describes the relationships between fluid's pressure and density in the reservoir. We shall use fluid compressibility c:

$$c = -\frac{1}{V} \frac{\partial V}{\partial p} \bigg|_T , \qquad (2.13)$$

where V denotes the fluid's volume, p is the pressure and T is a fixed temperature. Since density is defined as mass divided by volume then compressibility can be defined also as:

$$c = -\frac{1}{\rho} \frac{\partial \rho}{\partial p} \bigg|_T \qquad (2.14)$$

Hence for each phase we have:

$$c_i = \frac{1}{\rho_i}\frac{\partial \rho_i}{\partial p_i}\bigg|_T \,, \qquad\qquad i \in \{w,o\} \tag{2.15}$$

where c_i is compressibility for water, oil respectively.

Rock compressibility describes the relationship between rock's porosity and oil pressure at the constant temperature T :

$$c_R = \frac{1}{\Phi}\frac{\partial \Phi}{\partial p_o}\bigg|_T \tag{2.16}$$

Combining equations (2.12) together with (2.15), (2.16) we obtain:

$$\begin{aligned}&-\nabla\left[\frac{\alpha\rho_w k_{rw}}{\mu_w}\tilde{k}\left(\nabla p - \rho_w g \nabla d\right)\right]+\\&+\alpha\rho_w\Phi\left[S_w\left(c_w + c_r\right)\frac{\partial p}{\partial t}+\frac{\partial S_w}{\partial t}\right]-\alpha\rho_w q_w^{'''} = 0\end{aligned} \tag{2.17}$$

$$\begin{aligned}&-\nabla\left[\frac{\alpha\rho_o k_{ro}}{\mu_o}\tilde{k}\left(\nabla p - \rho_o g \nabla d\right)\right]+\\&+\alpha\rho_o\Phi\left[\left(1-S_w\right)\left(c_0 + c_r\right)\frac{\partial p}{\partial t}+\frac{\partial S_w}{\partial t}\right]-\alpha\rho_o q_o^{'''} = 0\end{aligned} \tag{2.18}$$

Equations (2.17), (2.18) are discretized and then implemented to create the reservoir simulator used in this project. When applying these equations to build the reservoir simulator some assumptions and simplifications were made.

For the two phases flow gravity forces and capillary pressures are ignored. Reservoir permeability is assumed isotropic, i.e. it is equal in all directions at a specified point $k_x = k_y = k$. This changes the problem into a closed reservoir, where fluid can flow only through the drilled wells. The injection and extraction of fluids through the drilled wells is usually modeled with a so called *well model*. We use in this project the well model described in [1]. At the wells the flow is specified by bottom hole pressures and flow rates. Either bottom hole pressures or fluid flow rates must be prescribed. These constraints are imposed by the well model. It is important to mention that while one of these values is prescribed at injector wells, the other one is prescribed at producer wells. For instance if we prescribe bottom hole pressure at injectors then flow rate is prescribed at producers. While, for example bottom hole pressure is prescribed at some wells, the flow rates can change freely at those wells.

The implementation of the model described in this section together with the well model illustrated in [1] is provided by the reservoir simulator ‘*simsim*’ developed at TU Delft. A more complete description of the simulator is given in Chapter 4 of this project. We can use the simulator to predict future reservoir behavior and production.

Generally, no matter what technique is applied to investigate future reservoir performance, one starts by running the model for some time period for which outcomes (e.g. pressures, flow rates, saturations) are available. If the computed quantities match the available data, the procedure is correct and can be used for predictions. When quantities do not match, some of the model parameters (e.g. permeability, porosity) must be changed and the model has to be run again. This process of adapting the model parameters such that observed data matches the computed outcome quantities is called history matching. The problem of history matching is also addressed as parameter estimation because one needs to estimate some parameters in order to improve the match.

2.3 History matching techniques

As mentioned above, history matching matches the history of observable variables, e.g. flow rates, pressures, saturations from existing wells with predictions of these quantities from a numerical model by adjusting parameters like permeability, porosity. There are different ways of doing history matching. Traditional history matching consists in running the simulator for some period, then comparing the results with the known field results and manually adjusting the reservoir parameters to improve the match. Because the match is performed manually by an engineer this is also called manual history matching. The success of the manual history matching depends mostly on the engineer's experience and knowledge about the field. The method has serious drawbacks as presented in [1]:

- It is usually performed after periods of years, on a campaign basis because the approach is time consuming.
- The obtained history-matched models often violate essential geological constraints.
- The updated model may reproduce the production data almost perfectly but has no predictive capacity because it has been over-fitted by adjusting a large number of unknown parameters using a much smaller number of measurements.

To overcome these limitations, a considerable amount of research has been performed to develop different history matching methods known also as computer assisted history matching. These techniques use computer logics to give estimates of the reservoir's parameters (e.g. permeability, porosity) for which an objective function (error function) is minimized:

$$f_E = \sum_{i=1}^{n_{par}} w_i \left(d_{io} - d_{is} \right)^2, \tag{2.19}$$

where f_E denotes an error function, n_{par} is the number of reservoir parameters, w_i is the weighting coefficient and d_{io}, d_{is} correspond to the observed and the simulated data that has to be matched.

One of the highest limitations of the traditional methods is the time issue. As new data becomes available for being included into the match the whole history matching process has to be repeated using all available measurements. The computer assisted history matching solves this problem.

The accuracy of a history matching process depends on the quality of the reservoir's model and the quality and quantity of measured data, typically production data (bottom hole pressures and flow rates at well locations).

2.4 Kalman filtering and localization methods

There are several approaches to computer assisted history matching which differ in the way they obtain the parameter set that minimizes the objective function defined in the previous section. One methodology that addresses the history matching problem is data assimilation.

Data assimilation combines a mathematical physical model with available measurements in order to estimate and predict different environmental processes.

One of the most known data assimilation methods is the KF method, published by R.E.Kalman in 1960. The author is describing a recursive solution to the discrete data linear filtering problem. The Kalman Filter is a set of mathematical equations that are used to estimate the state of a process in a way that minimizes the distance between measurements and model predictions.

The KF addresses the general problem of trying to estimate the state $x \in \mathbb{R}^{n}$ of a process that is governed by the linear stochastic difference equation:

$$X(k+1) = A(k)X(k) + B(k)u(k) + G(k)W(k) \tag{2.20}$$

with measurements:

$$Z(k) = H(k)X(k) + V(k), \tag{2.21}$$

where k represents the time; $A(k)$ is a $n \times n$ matrix that relates the state at the previous time step $k-1$ to the current step k; the matrix B relates the input vector u_k to the state vector; G is the noise input matrix; the matrix H in eq. (2.21) relates the state to the measurements $Z(k)$.

The random variables $W(k)$ and $V(k)$ represent the model and measurement noises. They are assumed to be independent of each other and white Gaussian: $W(k) \sim N(0, Q(k))$, $V(k) \sim N(0, R(k))$.

To solve the filtering problem we have to determine the probability density of the state $X(k)$ conditioned on the history of available measurements $Z(k)$. This conditional density function is assumed to be Gaussian; hence, it can be fully characterized by the mean and covariance matrix. The KF will recursively calculate the state vector $X(k)$ along with its covariance matrix P, conditioned on the available measurements up to some time k, under the criterion that the estimated error covariance is minimum. A complete description of the KF algorithm is presented in Appendix (see section A.2).

The KF method becomes computationally expensive for large scale systems and is not suitable for non linear systems. To overcome these limitations of the method, a large number of new algorithms were developed. An example of such an algorithm is ensemble Kalman filter (EnKF) [30, 31]. EnKF represents the probability density of the estimate by an ensemble of possible realizations. EnKF became a popular approach for solving the history matching problem in reservoir simulation [13, 14, 15].

The main advantage of EnKF is that it approximates the covariance matrix from the finite number of ensembles used, thus it becomes suitable for large non linear problems. The main disadvantage of EnKF appears when the number of variables to be estimated is much larger than the number of ensembles used. This leads to sampling errors which make the approximated covariance error to be underestimated. Unfortunately this is very often the case, since EnKF is a Monte Carlo method and for being affordable a relatively small number of ensembles are used (generally between 50 and 100 ensembles). The presence of unreal correlations, also called spurious correlations between large distance grid points is noticed in this case. Hence observations have high influence over too large distances, which is not physically true. The process of eliminating these unreal, large distance correlations is called localization [8, 9, 10, 11, 12]. The two most common localization methods are: Covariance Localization (CL) and Local Analysis (LA). A detailed description of Kalman filter, ensemble Kalman filter and both localization methods is given in the Appendix. Sometimes localization introduces inconsistencies, i.e. while trying to eliminate spurious correlations one might end up eliminating ‘good’ correlations also. In this project we shall use EnKF with local analysis to solve the history matching problem. Because of the above limitations we shall consider also a different approach to the history matching problem.

Graphical models merge graph theory and probability theory to provide a general setting in which a number of variables interact. Because of their flexibility their use has increased substantially over time. We shall use the acyclic directed graphs called Bayesian belief nets. A Bayesian belief net is one of the probabilistic graphical models, which describes the probability density function of a set of variables by specifying a number of conditional independence statements in a form of an acyclic directed graph and a set of probability functions. A description of: the Bayesian belief nets, the relationship between Kalman filter methods and Bayesian belief nets and a data mining algorithm is given in the next chapter.

Chapter 3

Bayesian belief nets (BBNs)

Human beings gather information from the environment in order to comprehend the world around them and to take proper actions. In probabilistic reasoning the events and objects in the world are represented by random variables. By assigning values to these random variables we can model the current state of the world. Many different probabilistic models have been proposed for this purpose. Graphical models are very useful tools for large scale models in which a large number of variables are connected in complex ways. Since they proved to be very flexible, their use has increased substantially, thus the theory behind them has been extended and developed over time. Graphical models are used in several scientific fields including: statistics, physics, biology and engineering. A graphical structure is a collection of nodes and links. Each node in the graph represents a random variable. The links represent the influences (dependencies) between nodes. The absence of a link between two nodes means that any dependence between these two random variables is represented via some other variables.

There are two main types of graphical models: directed and undirected. We shall refer in this Chapter to the directed acyclic graphs called Bayesian belief nets also named belief networks, Bayesian networks, probabilistic networks, casual networks, and knowledge maps.

Bayesian belief nets are often chosen to model high dimensional probabilistic distribution because of their capacity of displaying relationships between variables in an intuitive way and of representing cause effect relationships through the directionality of the arcs. One of the most important characteristics of a Bayesian belief net is that it can be used for inference. Given the values of the observed nodes, one can calculate the distribution of unobserved nodes. Considering the directionality of the arcs, if the reasoning is done bottom-up the Bayesian belief net is used for diagnosis, whereas if the reasoning is done top-down the Bayesian belief net is used for prediction.

This Chapter is organized as follows: section 1 introduces static Bayesian belief nets, section 2 presents dynamic Bayesian belief nets and the connection with the Kalman filter methods, and section 3 describes an algorithm of learning the structure and parameters of a continuous non parametric Bayesian belief net from data.

3.1 Static Bayesian belief nets

Bayesian belief nets are directed acyclic graphs. The nodes of the graph represent univariate random variables which can be continuous or discrete. The arcs of a Bayesian belief net represent direct influences. The absence of arcs can be interpreted as conditional independence statements between the nodes of the graph. A static Bayesian belief net models the dependence structure between a number of variables at a fixed time. We shall denote the static Bayesian belief net by BBN.

Let us consider a set of random variables $\left(X_1, X_2, ..., X_n\right)$, then from basic probabilities we know that the joint density/mass function can be written as follows:

$$f_{1,...,n}\left(x_1, x_2, ..., x_n\right) = f_1\left(x_1\right)\prod_{i=2}^{n} f_{i|1,...,i-1}\left(x_i \middle| x_1, ..., x_{i-1}\right), \qquad (3.1)$$

where f_i is the marginal densities of variable X_i and $f_{1,\dots,n}$ denote the joint density for $\left(X_1, X_2, \dots, X_n\right)$, $f_{i|1,\dots,i-1}$ is the conditional density of variable X_i.

We notice that for specifying the joint distribution of these variables we must specify values of an *n*-dimensional function. BBNs provide a more intuitive way of describing high dimensional distributions. The graph imposes that each variable is conditionally independent of all its predecessors in an ordering, given its direct predecessors. The direct predecessors of a node i corresponding to variable X_i are called *parents*. We shall denote the set of all parents of node i by $Pa(i)$.

Another way of checking a (conditional) independence directly from the BBN is through graphical criterion[2] which we shall explain using an example. Let us consider four nodes A, B, C, D as follows:

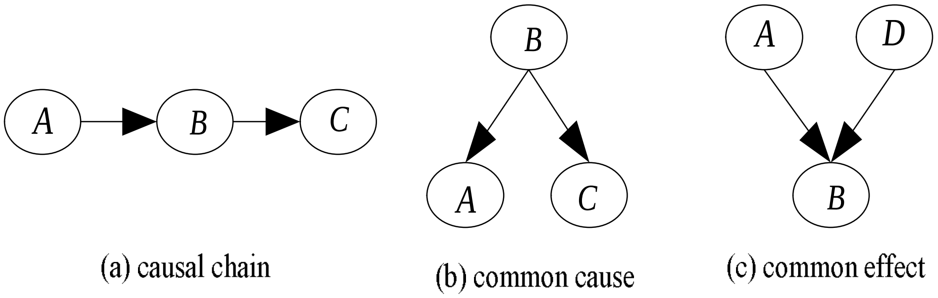
- A represents the variable “eating junk food”
- B represents the variable “overweight”
- C represents the variable “heart disease”
- D represents the variable “stress”

Figure 3.1 describes the possible conditional independence relations induced by a BBN: (a) causal chain, (b) common cause, (c) common effect.

A causal chain of three nodes is when A causes B which in turn causes C. In Figure 3.1 (a) we read the causal chain as “eating junk food causes overweight which causes heart disease.” A causal chain represents a conditional independence relation. In Figure 3.1 (a) the probability of C given B is exactly the same as the probability of C given both A and B. Knowing that A has occurred does not make any difference to our

[2] This is called the d-separation criterion.

beliefs about C if we already know that B has occurred, e.g. knowing that a person eats junk food does not tell us anything new about his heart condition if we already know that the subject is overweighed. Therefore, heart disease is conditionally independent of eating junk food given that the subject is overweight.



(a) causal chain (b) common cause (c) common effect

Figure 3.1

The variables A and B having a common cause B are represented in Figure 3.1 (b). In our example overweight is a common cause for two effects: eating junk food and heart disease[3]. Common causes represent the same conditional independence statements as chains. Hence, the graph from Figure 3.1 (b) tells us that eating junk food is conditionally independent of heart disease if we already have evidence of a person being overweight.

A common effect is represented by a BBN in a v-structure. This illustrates the situation when a node (the effect) has two causes. Thus, in Figure 3.1 (c) both eating junk food and stress are causes of getting a heart disease. Common effects produce the opposite independence structure to that of chains and common causes. In this case the parents are independent when there is no information about the common effect, but become dependent given information about the common effect, i.e. they are conditionally dependent. Therefore if we observe the effect (e.g. overweight), and then we find out that one of the causes is absent (e.g.

[3] The influence from B to A might not be so evident in the example used, but we shall consider it for the theoretical purpose.

junk food) this rises the probability of the other cause (e.g. stress). Hence, eating junk food and stress are conditionally dependent given evidence about a person being overweighed.

Next, we shall illustrate an example where both common causes and common effects relations are present. Figure 3.2 shows a simple static Bayesian belief net with four nodes corresponding to four random variables: X_1, X_2, X_3, X_4, where X_1 and X_2 form the sets of parents $Pa(3)$ and $Pa(4)$; X_3 and X_4 are called children of variables X_1 and X_2.

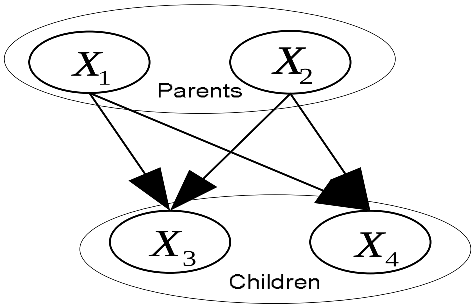


Figure 3.2. A simple BBN on 4 variables

Notice that we have the following common causes: X_1 is a common cause for X_3 and X_4, also X_2 is a common cause for X_3 and X_4. Therefore the graph tells us that X_3 and X_4 are a conditionally independent given X_1 and X_2.

Given evidence about X_3 and X_4, X_1 and X_2 become conditionally dependent because: X_3 is a common effect for X_1 and X_2, also X_4 is a common effect for X_1 and X_2. However, when no evidence about X_3 and X_4 is available, X_1 and X_2 are independent.

Each variable is associated with a conditional probability function of that variable given its parents in the graph, $f_{i|Pa(i)}\left(x_i \middle| x_{Pa(i)}\right), i = 1, ..., n$. As shown, when using a Bayesian belief net the representation, the joint distribution is significantly simplified by the (conditional) independence relations imposed by the graph. Hence instead of specifying values of an *n*-dimensional function we shall only have to specify values of a $p < n$ dimensional function (where p is the maximum number of parents of any node in the graph). The joint probability function from (3.1) becomes:

$$f_{1,\ldots,n}\left(x_1, x_2, \ldots, x_n\right) = \prod_{i=1}^{n} f_{i|Pa(i)}\left(x_i \middle| x_{Pa(i)}\right) \qquad (3.2)$$

In a BBN we distinguish a qualitative and a quantitative component. The graph itself and the conditional independence relations entailed by it form the qualitative part of a BBN. The quantitative part of a BBN consists in conditional probability functions associated with each variable in the graph. After these functions are quantified the BBN can be used for inference (prediction or diagnosis).

3.1.1 "Classical" BBNs

As mentioned in the previous section the nodes of a BBN can be discrete or continuous univariate random variables. We shall use the term "classical" for the discrete BBNs and the Gaussian BBNs.

The BBNs for which the nodes have discrete random variables are called discrete BBNs. These models specify marginal distribution for source nodes and conditional probability tables for child nodes.

Let us consider the BBN from Figure 3.2 with discrete nodes, each node taking m values, denoted $x_i^j, i = 1,\ldots,4; j = 1,\ldots,m$. The marginal distribution of X_1 and X_2 and the conditional probability table of X_3 and X_4 have to be specified. Table 3.1 shows the conditional probability table for variable X_3. A similar conditional probability table can be obtained for variable X_4.

X_1	X_2	$P\left(X_3 = x_3^1 \middle| X_1, X_2\right)$	$P\left(X_3 = x_3^2 \middle| X_1, X_2\right)$		$P\left(X_3 = x_3^m \middle| X_1, X_2\right)$
x_1^1	x_2^1	?	?	$\cdots$	?
$\cdots$ $\cdots$	$\cdots$ $\cdots$	?	?	$\cdots$ $\cdots$	?
x_1^m	x_2^m	?	?	$\cdots$	?

Table 3.1 Conditional probability table for X_3

Table 3.1 contains m^3 entries. In the case of binary variables, 8 values have to be specified. Imagine that the variables take 10 possible values each, and then just for three variables we must specify 10^3 entries, i.e. 1000 conditional probabilities. Such an assessment burden can only be reduced by a drastic discretization of the nodes, or simplification of the model. Both solutions can be viewed as disadvantages of the discrete models [16].

Another limitation of the discrete BBNs is with respect to their flexibility to changes in modeling; i.e. if one parent node is added, then conditional probability table of the child has to be re-quantified.

Some real world models can only be described in a continuous framework, therefore methods for continuous BBNs were developed. Typically there are two common ways to deal with continuous BBNs. One is to discretize the continuous variables and work with the corresponding discrete model. Another one is to assume that variables corresponding to the nodes of the graph have joint normal distribution. The first attempt to work with continuous BBNs was actually for joint normal variables. These are called Gaussian BBNs (or normal BBNs). Continuous BBNs developed for joint normal variables interpret the influences from parents to children as partial regression coefficients when the children are regressed on to the set of parents. The partial regression coefficient gives the amount by which a dependent variable increases when one independent variable is increased by one unit and all other independent variables are held constant. Each arc corresponds to a partial regression coefficient. In absence of data, these coefficients should be assessed by experts.

Let $X = \left(X_1, X_2, ...X_n\right)$ have a multivariate normal distribution. For Gaussian BBNs the conditional probability functions associated with the variables are of the form:

$$X_{i|Pa(i)} \sim \mathrm{N}\left(\mu_i + \sum_{j \in Pa(i)} b_{ij}\left(X_j - \mu_i\right); v_i\right), \tag{3.3}$$

where $\mu = \left(\mu_1, ..., \mu_n\right)$ is the mean vector, $v = \left(v_1, ..., v_n\right)$ is a vector of conditional variances and b_{ij} 's are linear coefficients that can be thought as partial regression coefficients.

Gaussian BBNs are easy to build and quantify if the assumption of joint normal distribution holds. If the normality assumption does not hold then the individual variables must be transformed to normals. The conditional variance in normal units must be constant. Notice that the partial regression coefficients that have to be assessed to the arcs apply to the normal units of the transformed variables, not to the original units [16]. Hence regressors coefficients depend on the set of regressors. These conditions induce a heavy burden on experts, when expert judgment is used to quantify these models. Moreover if a parent node is added or removed, after quantification, the previously assessed partial regression coefficients must be re-assessed.

Often the normality assumption does not hold, hence all of the above requirements make the Gaussian BBNs unappealing for modeling high dimensional probabilistic models.

3.1.2 Continuous non parametric Bayesian belief nets

The previous section illustrated the two most common methods that deal with static BBNs: the discrete BBNs and the Gaussian BBNs. Some of the limitations of both these approaches were discussed in the previous section. To overcome these limitations when dealing with arbitrary continuous variables a new method is proposed in [22]. In this method no joint distribution is assumed for the variables, this makes the BBN non parametric. We shall use the notation NPBBNs for static continuous non-parametric Bayesian belief nets.

We shall first present the basic theoretical concepts, definitions and notations used in this section, [e.g. 1,2].

Definition 1*: The product moment correlation* ρ of two random variables X and Y with finite expectations $E(X)$ and $E(Y)$ and finite variances σ_X^2 and σ_Y^2, is:

$$\rho(X,Y)=\frac{E(XY)-E(X)E(Y)}{\sigma_X\sigma_Y} \tag{3.4}$$

Definition 2: *The rank correlation* r of two random variables X and Y with cumulative distribution functions F_X and F_Y is:

$$r(X,Y)=\rho(F_X(X),F_Y(Y)) \tag{3.5}$$

The rank correlation is the product moment correlation of the ranks of variables X and Y.

Definition 3: *The conditional correlation* of X and Y given Z is the product moment correlation of X and Y given Z:

$$\rho_{XY|Z} = \rho\left(X\left|Z, Y\right|Z\right) = \frac{E\left(XY|Z\right) - E\left(X|Z\right)E\left(Y|Z\right)}{\sigma_{X|Z}\sigma_{Y|Z}} \tag{3.6}$$

Definition 4: *The conditional rank correlation* of X and Y given Z is:

$$r_{X,Y|Z} = r_{X^*,Y^*}, \tag{3.7}$$

where $\left(X^*, Y^*\right)$ has the distribution of $\left(X, Y\right)$ given $Z = z$.

The (conditional) rank correlation is a property of copulas. Hence they can be realized by copulas.

Definition 4: *The copula* of two continuous random variables X and Y is the joint distribution of $F_X\left(X\right)$ and $F_Y\left(Y\right)$, where F_X and F_Y are the cumulative distribution functions of X, Y respectively. The copula of $\left(X, Y\right)$ is a distribution on $\left[0,1\right]^2 = I^2$ with uniform marginal distributions.

Definition 5: If ϕ_ρ is the bivariate normal cumulative distribution function with product moment correlation ρ and ϕ^{-1} the inverse of the standard univariate normal distribution function then:

$$C_\rho\left(u,v\right) = \phi_\rho\left(\phi^{-1}\left(u\right), \phi^{-1}\left(v\right)\right), \qquad \left(u,v\right) \in I^2 \tag{3.8}$$

is called the *normal copula*. Note that ρ is a parameter of the normal copula.

Some copulas have the property of representing independence as zero correlation. Such copula is said to have the *zero independence property*. Normal copula has this property.

Another useful concept is partial correlation. *The partial correlations* $\rho_{12;3,\ldots,n}$ can be interpreted as the correlation between the orthogonal projection of X_1 and X_2 on the plane orthogonal to the space spanned by $X_3,\ldots,X_n$. Partial correlations can be computed from correlations with the following recursive formula:

$$\rho_{12;3,\ldots,\mathrm{n}} = \frac{\rho_{12;3,\ldots,\mathrm{n-1}} - \rho_{1\mathrm{n};3,\ldots,\mathrm{n-1}}\rho_{2\mathrm{n};3,\ldots,\mathrm{n-1}}}{\sqrt{1\text{-}\rho^2_{1\mathrm{n};3,\ldots,\mathrm{n-1}}}\sqrt{1-\rho^2_{2\mathrm{n};3,\ldots,\mathrm{n-1}}}} \tag{3.9}$$

In general the partial correlation is not equal to the conditional correlation. However, for the joint normal distribution the partial and conditional correlations are equal.

Another property specific for normal variables that gives a relationship between the rank correlation and the product moment correlation was developed by Pearson (1907).

Proposition 1: Let (X,Y) be a random vector with the joint normal distribution, then:

$$\rho(X,Y) = 2\sin\left(\frac{\pi}{6}r(X,Y)\right) \tag{3.10}$$

We shall now move on, to present in more detail the continuous non-parametric BBNs [1, 2].

In the procedure proposed in [17] nodes are associated with continuous invertible distribution and arcs with (conditional) rank correlations. Figure 3.3 presents the BBN from the previous sections as a NPBBN. The nodes X_1, X_2, X_3 and X_4 reprezent continuous variables and arcs are associated with the (conditional) rank correlations $\{r_{13}, r_{14}, r_{23|1}, r_{24|1}\}$. We specify parent-child rank correlations. If a variable has only one parent then we shall assess an unconditional rank correlation between them. For a variable with more than one parent we first deicide upon an ordering of the parents. The rank correlation between the child and its first parent is an unconditional rank correlation. The rank correlation between the child and its next parent is conditional on the values of the previous parent, etc. These (conditional) rank correlations are realized by a copula with the zero independence property. The property is necessary because the absence of an arc in the NPBBN translates into (conditional) independence between the corresponding variables in the graph and is specified by a zero (conditional) rank correlation.

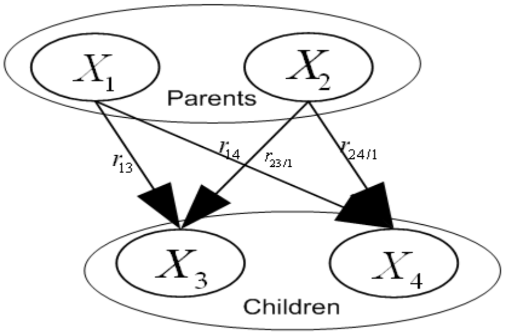


Figure 3.3 A NPBBN with four nodes

As mentioned, no joint distribution is assumed, which makes the BBN non-parametric. In order to quantify the BBNs using this approach, one needs to specify all one dimensional marginal distribution and a number of (conditional) rank correlations equal to the number of arcs in the NPBBN. One advantage of the NPBBNs over the “classical” BBNs is the flexibility of such a model. If a parent node is added or removed, then the previously assessed (conditional) rank correlations need not be re-assessed. We consider constant (conditional) rank correlations; however this needs not be always the case [16].

It has been shown in [16] that the marginal distribution together with the (conditional) rank correlations which are realized by the chosen copula uniquely determine the joint distribution. The (conditional) rank correlations and the marginal distribution needed to specify the joint distribution represented by the NPBBN can be obtained from data, if available, or elicited from experts [18]. For instance in order to elicit the rank correlations r_{13} and $r_{23|1}$ for the NPBBN from Figure 3.3 we ask the experts the following questions:

- *Suppose that variable X_1 was observed above its q^{th} quantile, what is the probability that variable X_3 will also be observed above its q^{th} quantile?*
- *Suppose that both X_3 and X_1 were observed to be above their medians. What is the probability that also X_2 will be observed above its median?*

To stipulate the joint distribution it is sufficient to sample it. Further in this thesis we shall use a sampling protocol based on the normal copula [16]. However, other sampling methods using different copulas are available in [1,2]. These methods are based on the sampling methods for other graphical models called vines. For further details regarding the sampling procedure of a NPBBN using vines we refer the reader to [1,2]. Suffice to say here that sometimes these sampling procedure have serious drawback mostly in terms of computational time [16]. To overcome this limitation we are using the joint normal copula assumption. To our knowledge normal copula is the only copula that allows rapid updating while preserving the zero independence property.

Let us illustrate the sampling procedure using normal copula assumption introduced in [16], in a particular case. We consider the NPBBN from Figure 3.4.

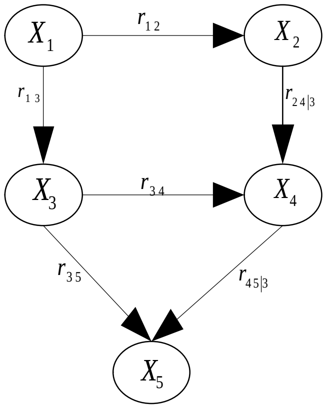


Figure 3.4

1. Let $(X_1,...,X_5)$ be a random vector and let F_i denote the cumulative distribution function of X_i, $i=1,...,5$. We transform the margins X_i to standard normal random variables via the transformation

   $$Y_i = \phi^{-1}\left(F_i\left(X_i\right)\right), \qquad i=1,...,5 \qquad (3.11)$$

   where ϕ is the cumulative distribution function of the standard normal distribution.
2. We build the NPBBN from Figure 3.4 for the transformed variables. Given the monotonicity of the transformation 3.11 we can asssign the same (conditional) rank correlations to the new NPBBN. We shall realize this specification using the normal copula. Hence we can use the properties of the joint normal distribution.
3. To each arc of the NPBBN we assign a partial correlation calculated as follows. We start with a (conditional) rank

correlation and use Pearson's transformation (3.10) to calculate the corresponding (conditional) product moment correlation. We know that conditional and partial correlations are equal for the normal variables. Finally we compute the correlation matrix C using the recursive formula 3.9.

4. We sample the joint normal distribution $Y_1,...,Y_5$ with the correlation matrix C
5. For each sample we transform the normal margins back to the original marginal distribution by calculating $\left(F_1^{-1}\left(\phi\left(y_1^j\right)\right),...,F_5^{-1}\left(\phi\left(y_5^j\right)\right)\right)$, where $\left(\left(y_1^j\right),...,\left(y_5^j\right)\right)$ is the j^{th} sample from the previous step.

In this way we obtain a sample of the joint distribution of the initial variables $\left(X_1,...,X_5\right)$ together with the dependence structure realized by the normal copula.

One of the most important characteristics of probabilistic graphical models is that they can be used for inference. One can calculate the distribution of unobserved nodes, given the values of the observed ones. When the normal copula is being used by transforming the marginals to standard normals we are basically transforming the distribution to a joint normal distribution. In this setting any conditional distribution will also be normal. Hence the conditioning can be performed analytically [23].

If we have the conditional distribution of the transformed variables we can find the conditional distribution of the original variables by transforming back.

Proposition 2: Let $\left(Y_1,Y_2\right)$ have a bivariate normal distribution, with standard normal marginals. Let F_1 and F_2 be two continuous, invertible distribution functions and $X_i = \Phi^{-1}\left(F_i\left(Y_i\right)\right)$, $i=1,2$. Then the conditional distribution of $X_1|X_2$ can be calculated as $F_1^{-1}(\phi(Y_1|Y_2))$. A proof of this proposition is given in [16].

This section presents the NPBBNs method as introduced in [16]. Our goal is to model a reservoir simulation problem, i.e. to solve the history matching problem as illustrated in Chapter 2 of this thesis. Kalman filter methods are popular techniques that deal with this issue (see Appendix). We shall consider, as an alternative a NPBBNs based approach to solve the history matching problem in reservoir simulation. The system is observed at different discrete times, therefore a static NPBBNs can be built at each time. In order to model the process we have to connect these static NPBBNs over time. This is possible with the help of a dynamic Bayesian belief net. In the next section we present the dynamic Bayesian belief nets and their relationship with the Kalman filter methods.

3.2 Dynamic Bayesian belief nets and Kalman Filter methods

Most of the events we meet in everyday life are not represented just on a particular point in time, but they can be described as evolving in time through multiple states. The field that deals with methods for reasoning about temporal relationships is generally known as time-series analysis. According to [19] a time series is a sample realization of a stochastic process, consisting of a set of observations made sequentially in time. A dynamic Bayesian belief net is one of the tools to model time series and is a special case of the static BBN.

We can say that the dynamic BBN consists of a sequence of sub--models (static BBNs) each representing the system at a particular point or interval in time (time slice). The relationships between variables in a time slice are represented by intra-slice arcs. Generally the structure of a time slice does not change over time. Note that this is not a theoretical constraint [21] but merely one made for the simplification of the calculations. Therefore we can model dynamic BBNs with the same or with different structures over time. The connections between different time slices are realized by inter-slice arcs. The arcs between slices are drawn from left to right reflecting a casual flow of time.

Figure 3.5 shows a dynamic BBN which contains slices of a static BBN at different discrete times. Let the variables corresponding to the nodes of the static BBN X_1, X_2, X_3, X_4 be considered at different times $t = 1,...,T$. We define the state vector of a system as the vector that contains all the nodes in a static BBN at some time t. Therefore the state vector of the dynamic BBN from Figure 3.5 is:

$$X(t) = \begin{pmatrix} X_1(t) \\ X_2(t) \\ X_3(t) \\ X_4(t) \end{pmatrix}, \qquad t = 1,...,T \tag{3.11}$$

The states of any system described as a dynamic BBN satisfy the first order Markov property [19] which says that the state of a system at time *t* depends only on its immediate past, i.e. the state of the system at time *t-1*. A Markov process is a process that has the first order Markov property.

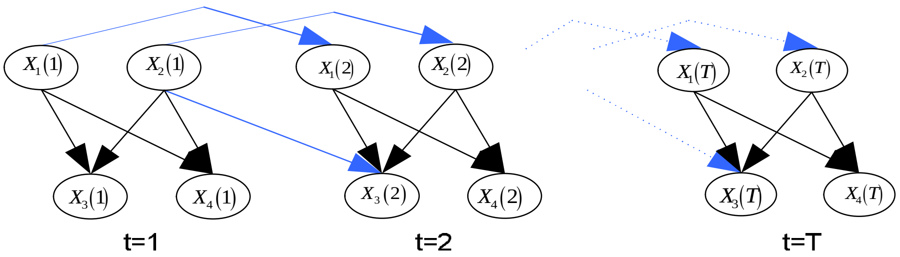


Figure 3.5: A dynamic Bayesian belief net

We shall recall the Kalman filter methods presented in Appendix. Kalman filter models give a recursive procedure for estimating the state space of a system governed by the equations:

$$X_t = A_t X_{t-1} + w_t, \qquad w_t \sim N\left(0, Q_t\right) \tag{3.12}$$

$$Y_t = H_t X_t + v_t, \qquad v_t \sim N\left(0, R_t\right), \tag{3.13}$$

Where t is the time, X_t is a state vector, Y_t[4] is an observation vector, A and H are matrices and w_t , v_t are model and measurements errors normally distributed with mean zero and standard deviation Q_t , R_t respectively. The Kalman filter method is a special case of a dynamic BBN with conditional linear Gaussian distribution [20].

Figure 3.6 presents the Kalman filter methods as dynamic BBNs. Note that the first order Markov property holds i.e. $X_{t+1} \perp X_{t-1} \middle| X_t$. Also $Y_t \perp Y_{t'} \middle| X_t, \forall t < t'$, which shows that the measurements are conditionally independent given the model state. As illustrated in Appendix, the Kalman filter methods assume that the dynamic system is jointly Gaussian. The main advantage of using dynamic BBNs over Kalman filter is that the dynamic BNNs can use arbitrary marginal probability distributions.

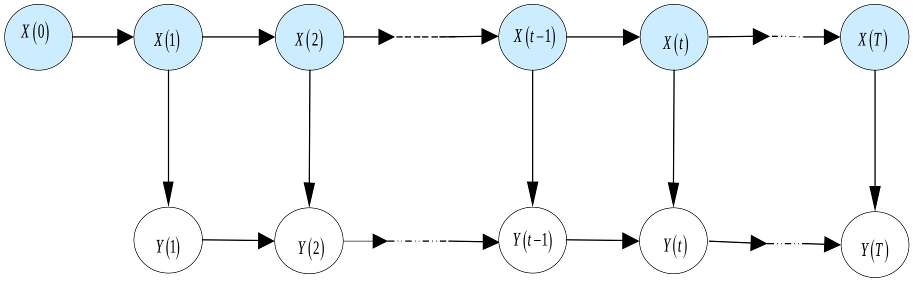


Figure 3.6: Kalman filter methods as dynamic BBNs

[4] There is no connection between the notation Y from this section and the notation Y used in section 3.1.2 of this Chapter.

As illustrated in this section a dynamic BBN contains slices of static BBNs at discrete times. The slices of static BBNs may or may not have the same structure over time [21]. We shall recall that our goal is to solve the history matching problem at different discrete times. For complex applications dependencies and relations between variables tend to change over time. Next section will present a data mining procedure for learning the structure of a static BBN together with its corresponding parameters from a given data set at each time. The connections between different time slices will be functional relationships. They will be given by the simulator 'simsim'. In this way a dynamic BBN with changed structure over time can be obtained. Figure 3.7 shows an example of the dynamic BBN presented in Figure 3.5 with changed structure over time.

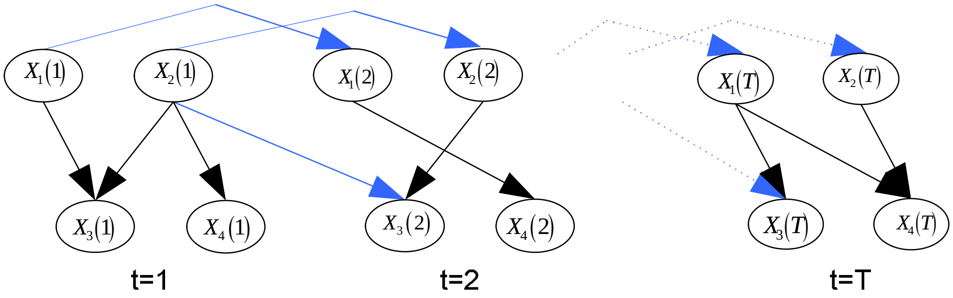


Figure 3.7: A dynamic BBN with changed structure over time

3.3 Data mining with NPBBN

Generally, there are two possible situations regarding the availability of the data for which we want to define a graphical structure and assess the required parameters. The data is given, or the data does not exist or is very limited. In the latter case expert judgment must be used. However, if we have data, we would like to extract a model that fits these data. In the process of learning a model from data two aspects can be of interest: learning the parameters of the model given the structure, and learning the structure itself together with the parameters. Our goal is to learn the structure of a NPBBN together with the corresponding parameters.

We shall consider the static NPBBN as a tool for data mining. Data mining is the process of extracting and analyzing information from large data sets. The method is designed for ordinal data. An ordinal multivariate data set is one in which the numerical ordering of values for each variable is meaningful. A database of student names is not ordinal, but a database of permeability values for a reservoir is ordinal; higher permeability allows an easier flow in the reservoir. A NPBBN induced from data can be used to investigate relationships between variables, as well as to make predictions, by computing conditional probability distribution of a variable given the values of some others variables (see section 3.1.2).

When learning a NPBBN from a database the one dimensional marginal distribution are taken directly from data, and the model assumes only that the joint distribution has a normal copula. We can say that the rank dependence structure of the variables is that of a joint normal distribution. The assumption of joint normal copula allows for rapid conditionalization (see section 3.1.2).

The purpose is to build and validate a NPBBN model that captures most of the dependencies present in a database. While learning a NPBBN from data we must also define ways to validate it. Validation of the model requires two steps as showed in Figure 3.8:

(a) Firstly, we assume that our data comes from the joint normal copula. In most cases if this condition is not satisfied we shall stop because the method is not suitable for the data set. However, given some special conditions or specifications in the modeled system one could consider modeling a data set that does not satisfy this condition.

(b) A saturated graph represents all possible dependencies present in a data set. We are interested in learning the structure of a NPBBN that approximates the saturated graph by introducing conditional independence relations. We validate a learned NPBBN which is an adequate model for the saturated graph.

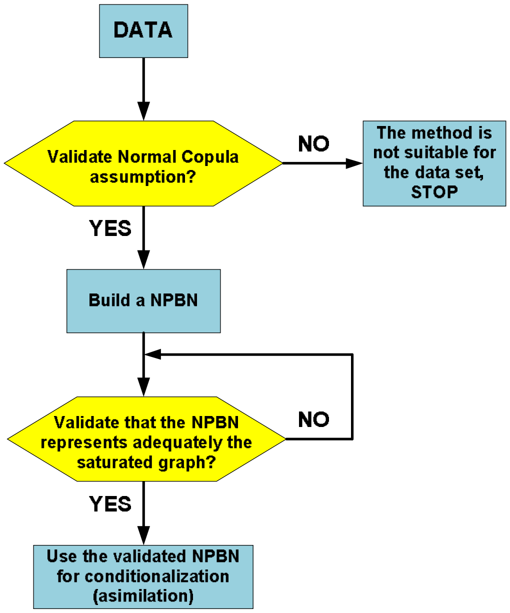

DATA
Validate Normal Copula assumption?
NO
The method is not suitable for the data set, STOP
YES
Build a NPBN
Validate that the NPBN represents adequately the saturated graph?
NO
YES
Use the validated NPBN for conditionalization (asimilation)

Figure 3.8: The model's validation steps

Validation of the two steps described above requires an overall measure of dependence on which statistical tests can be based. The method uses the determinant of the correlation matrix. For understanding the choice of the determinant as a measure of dependence for this method we shall present the following theorem [16].

Theorem 1: Let D be the determinant of an n-dimensional correlation matrix $(D > 0)$. For any BBN with partial correlation specified in the arcs:

$$D = \prod \left(1 - \rho_{ij;D_{ij}}^{2}\right), \tag{3.15}$$

where, $\rho_{ij;D_{ij}}$ is the partial correlation associated with the arc between node i and node j, D_{ij} is the conditioning set for the arc between node i and node j, and the product is taken over all arcs in the NPBBN. A proof of this theorem is given in [16]. Hence the determinant factorizes on the arcs of the NPBBN.

The determinant attains the maximum value of 1 if all variables are uncorrelated and a minimum value of 0 if there is linear dependence between variables. Since we are working with copulas models it is more natural to work with the rank correlations instead of product moment correlations. Therefore we use the fact that partial correlation is approximately equal to the (conditional) rank correlation. The approximation is reasonable given the following arguments: we use the normal copula to realize the (conditional) rank correlations associated with the arcs of the NPBBN; for the normal variables the (conditional) product moment correlations are equal to the partial correlations. Because we have normal variables we can use Pearson’s transformation (equation 3.10) to compute (conditional) rank correlations from (conditional) product moment correlations.

In this context we define the following determinants of the rank correlation matrices: DER is the determinant of the empirical rank correlation matrix; DNR is the determinant of the empirical normal rank correlation matrix; DBBN is the determinant of the empirical Bayesian belief net rank correlation matrix. Figure 3.9 describes the ordinal data mining procedure presented in this section.

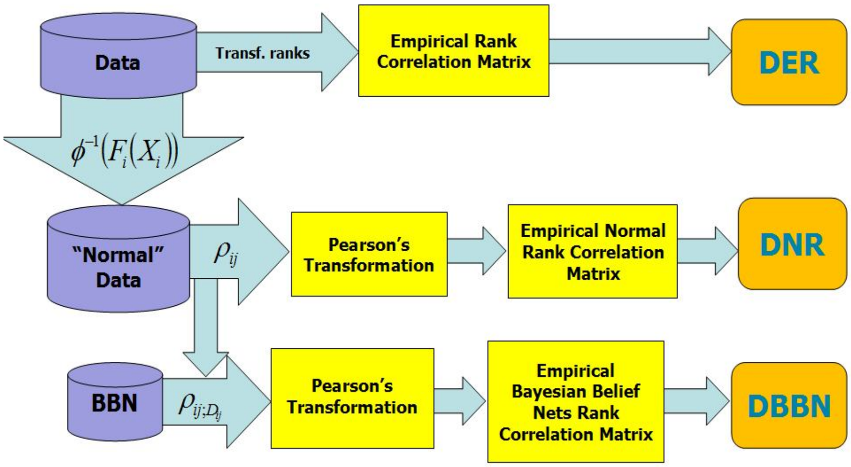


Figure 3.9: Ordinal data mining

If we transform the original data to ranks we can calculate DER for the empirical rank correlation matrix.

DNR is obtained by transforming the marginals to standard normals, and then transforming the product moment correlations to rank correlations using Pearson's transformation (equation 3.10). DNR is not in general equal to DER because DNR assumes the normal copula, which generally differs from the empirical copula. A statistical test for the suitability of DNR for representing DER is to obtain the sampling distribution of DNR and check whether DER is within 90% confidence band of DNR. If DNR is not rejected on the basis of this test, we shall build a NPBBN which represents the data. Since the method assumes the normal copula we can only recover the dependence structure represented by the *DNR*. Therefore, we add arcs between variables only if the rank correlations from the empirical normal rank correlation matrix are among the largest.

DNR corresponds to the determinant of the saturated NPBBN, therefore this determinant has values close to 0. Moreover for the saturated graph we have $DNR = DBBN$. Generally, many of the connections represented by the saturated graph are small and do not reflect the truth. Our purpose is to build a model that eliminates these noisy influences while maintaining most of the important dependencies in the database. As we eliminate arcs from the NPBBN, we introduce (conditional) independencies; hence DBBN gets further from *0*. Thus for the non-saturated graph the relationship $DBBN > DNR$ holds. The second validation step is based on a similar statistical test as the first one. The sampling distribution of DBBN is computed and we check whether DNR is within 90% confidence band of DBBN. If DBBN is not rejected on basis of this test we validate the NPBBN model. The procedure of building a NPBBN from a dataset is not automated. Therefore, there is more than one model that can be validated to represent the same database. Once the NPBBN is learned from data, we can further use it for prediction (see section 3.2.1).

The method presented in this section is implemented in Uninet, a software developed at TU Delft.

Chapter 4

Experimental setup

The purpose of this Chapter is to introduce the experimental setup necessary to perform the case study from Chapter 5 of this project. We shall recall that our goal is to solve the history matching problem, i.e. to estimate the permeability field. The case study will compare results of estimating the permeability field by using the EnKF method (with local analysis) with the results obtained by using the continuous NPBBN approach. This Chapter is organized as follows: section 1 gives a detailed description of the problem, section 2 presents the twin experiment, and section 3 illustrates the measures of performance used for comparing the two methods.

4.1 Problem description

We shall consider a synthetic two dimensional squared petroleum reservoir with a size of $700(m)\times 700(m)$ equipped with a uniform Cartesian grid consisting of 21 grid cells in each direction as showed in Figure 4.1[5]. The reservoir has an injector well in the middle of the field and four producers, one in each corner. One fluid (typically water) is injected through the injector well and oil is pumped out through the producers. This is the case of a so called *two phase flow* in which we

[5] For the ease of illustration we represent the 21X21 grid block in a smaller size.

distinguish two fluids: oil and water. The next section will give more details of the two phase flow for the reservoir illustrated in Figure 4.1.

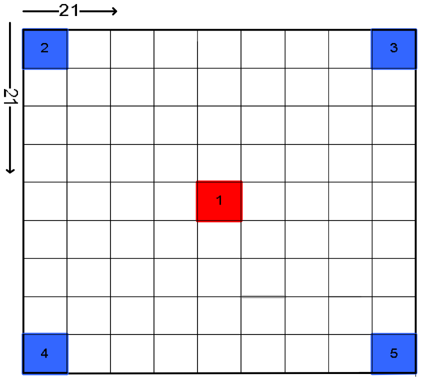


Figure 4.1: Reservoir field description

4.1.1 Two phase flow

We shall consider two fluids, water and oil, that flow into the reservoir described in the previous section. Water is injected through the injector well and oil is pumped out through the four producer wells located at each corner of the reservoir. A mathematical physical model is applied to the two dimensional square grid from Figure 4.1. We shall use the two phase flow mathematical physical model presented in Chapter 2 of this thesis.

The reservoir is considered a closed space where liquid gets in and out only through the drilled wells. Therefore, the drilled wells become reservoir's boundaries. As mentioned in Chapter 2, the injection and extraction of fluids through the drilled wells is modeled with the well model described in [24]. The flow is specified at the wells by bottom hole pressure and fluid flow rates. The well model imposes that either bottom hole pressure or fluid flow rates must be prescribed. We shall consider the case in which the injection well is constrained by the prescribed flow rates and production wells are constrained by the bottom hole pressure.

In reservoir simulation the two phase flow mathematical model is combined with the well model. The two phase flow mathematical model's implementation, together with the well model is provided by the simulator 'simsim'. Note that we shall consider the model as being perfect (without uncertainties).

As mentioned, the reservoir from Figure 4.1 is a 21 grid cells in each direction. Therefore we distinguish a total of 441 grid blocks. Each grid cell is defined to have its own grid block pressure p and saturation[6] S. Let the grid block pressures and saturations be considered at different times $t=1,...,T$. We define the state vector of the system as the vector that contains grid block pressures and saturations for all the grid blocks at some time t:

$$X(t)=\begin{pmatrix} p(t) \\ S(t) \end{pmatrix}, \tag{4.1}$$

In this thesis we assume that the only available observations are the ones from the drilled wells. Therefore, given the well constraints, we measure bottom hole pressure, denoted by bhp, at the injector, and total flow rates, denoted by q, at the producers. Note that here by total flow rates we mean the summation of the water flow rate and oil flow rate. Separate oil and water flow rates could be considered. Nevertheless we chose to work with the sum of the two. The bhp and q measurements are often referred as the *production data* in reservoir simulation. We define the vector of measurements at each time t as:

$$Z(t)=\begin{pmatrix} bhp(t) \\ q(t) \end{pmatrix} \tag{4.2}$$

[6] Here by S we refer to water saturation which was denoted S_w in Chapter 2 of this thesis.

Measurements are functions of field pressures and saturations. Observations are added to the state vector:

$$X(t) = \begin{pmatrix} p(t) \\ S(t) \\ bhp(t) \\ q(t) \end{pmatrix} \tag{4.3}$$

The purpose of this project is to estimate the permeability field. We denote permeability with k . Generally the values of the permeabilities are very small, e.g. values of $10^{-13}\left[m^2\right]$ are usual. This is why we use the natural logarithm of permeability instead of permeability. We augment the state vector with the log-permeability. The final form of the state vector is:

$$X(t) = \begin{pmatrix} \log k(t) \\ p(t) \\ S(t) \\ bhp(t) \\ q(t) \end{pmatrix}, \tag{4.4}$$

where $\log k(t)$ is a vector which consists of log-permeability values corresponding to each grid cell, i.e. 441 values. Hence the size of the state vector is $\mathbb{R}^{1328}$.

In the state vector presented above we distinguish three types of variables:

- the dynamic variables: pressure and saturations. These are called dynamic because they change in time.
- the static parameter log-permeability which is time invariant. This parameter is updated (corrected) at each assimilation (conditionalization) step.
- the observations from the wells.

4.2 Twin Experiment

In this project we shall perform a twin experiment, i.e. we assume that the ‘truth’ is known. Of course this kind of situation is not encountered in real life but is often used in synthetic applications to test the performance to the method(s) used. We have available 1000 realizations of the permeability field. We chose the ‘true’ permeability field as one realization from the 1000 members. Note that any realization out of the 1000 can serve as the ‘true’ permeability. The ‘true’ permeability field in both normal and logarithmic scale considered in this experiment is presented in Figure 4.2.

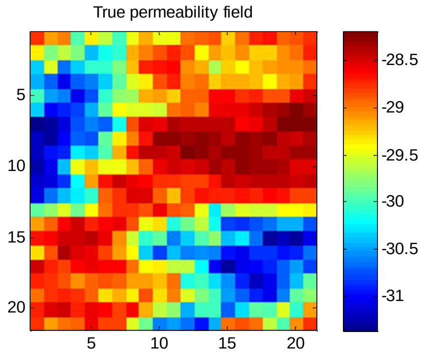


Figure 4.2: The realization chosen as the true permeability presented in logarithmic scale

Both methods used in the estimation problem require measurements and an initial guesses. In the next subsections we shall present the measurements and the initial permeability ensembles used to compare the two methods. Section 4.2.1 shows the measurements generation, section 4.2.2 describes the initial ensembles chosen from the 1000 realizations at our disposal. Moreover, one of the methods uses directed acyclic graphs to represent influences between variables as arcs. The directionality of the arcs is an important feature of the graph. Section 4.2.3 illustrates the choice of the arcs directionality in the NPBN.

4.2.1 Measurements generation

Typically, the reservoir is in a state of equilibrium at the starting time. Therefore, the initial dynamic variables (i.e. initial pressure and water saturation corresponding to each grid block) are assumed to be known and equal in every grid cell. The grid block pressure is equal to $p = 3*10^7 \left[Pa\right]$ and the grid block saturation is equal to $S = 0.2$. Moreover, at the starting time no production data is available. Hence in this setting the only choice that one has to make is the initial permeability. This will completely determine the behavior of the reservoir in time. By running the simulator with the chosen 'true' permeability field we obtain the 'true' production data. We generate synthetic measurements by adding error to the 'true' data. The error for each observable variable is taken to be normally distributed with mean zero and variance as 5%[7] of the actual value of the variable of interest. We consider measurements for every 60 days for a period of 420 days. Figure 4.3 shows the noisy data derived from the 'true' data for the two phase flow.

[7] Typically the production data can have an error between 5%-20%.

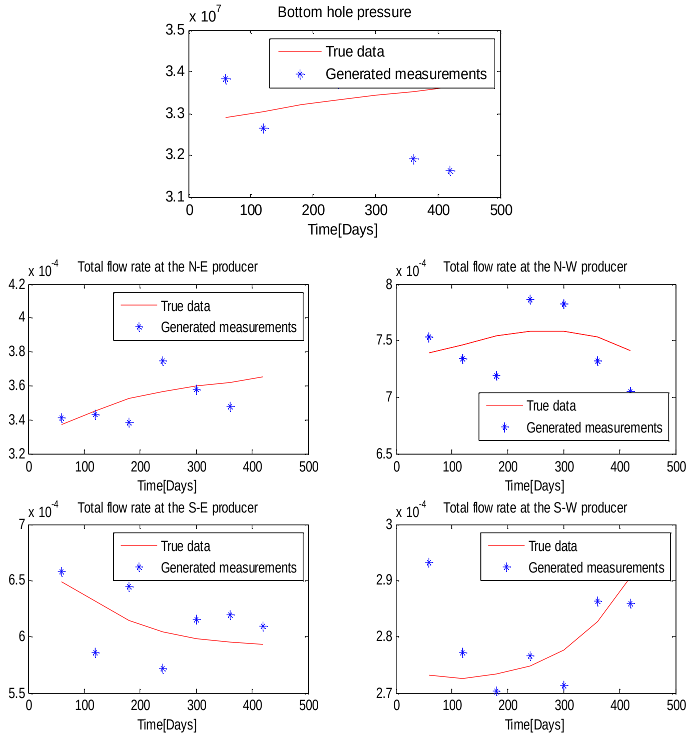


Figure 4.3: The generated data with 5% error from the true observations used for experiment

4.2.2 Initial ensemble

We established that the reservoir is in a state of equilibrium at the starting time. Hence the grid block pressure and saturations are perfectly known and equal for each grid cell, and, the only choice that one has to make is the initial permeability. We shall perform experiments using 100, 300 and 900 ensembles chosen from the 1000 realizations that we have at our disposal. Figures 4.5 , 4.6 and 4.7 show the mean permeability value for the initial guess for 100, 300 and 900 ensembles. Figure 4.8 shows plots of 8 ensembles from the initial choice.

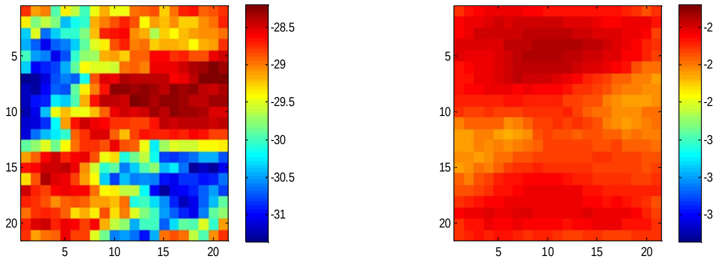


Figure 4.5: Left: the true log-permeability. Right: the mean of the 100 log-permeabilities chosen as initial guess

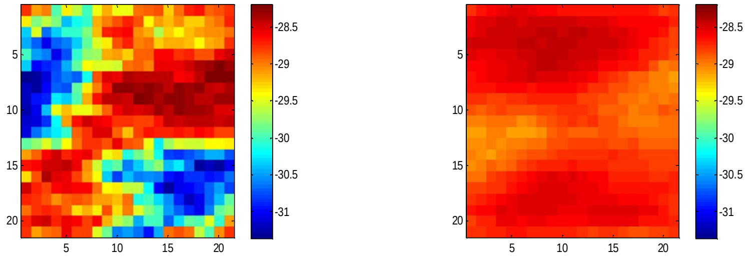


Figure 4.6: Left: the true log-permeability. Right: the mean of the 300 log-permeabilities chosen as initial guess

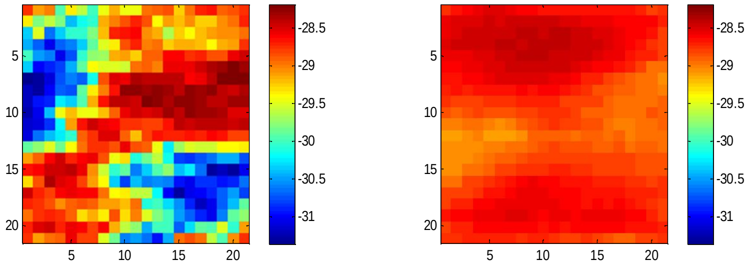


Figure 4.7: Left: the true log-permeability. Right: the mean of the 900 log-permeabilities chosen as initial guess

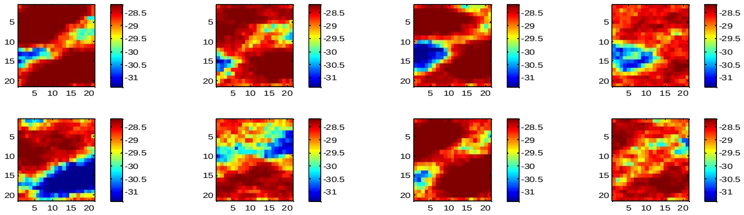


Figure 4.8: An example of 8 ensembles from the initial guess

4.2.3 Arcs directionality in the NPBBN

When applying the NPPBN based method we build a directed acyclic graph that contains the variables defined in the state vector described in section 4.1 of this Chapter. Each node in the directed acyclic graph is associated with a variable from the state vector at a certain location of the grid block. The dimension of the state vector is $\mathbb{R}^{1328}$. Therefore for estimating the permeability field for the entire grid one may consider 1328 nodes in the directed acyclic graph of the NPBBN. However certain simplifications that will eventually reduce the size of the model will be applied. For the simplicity of the exposition we shall use the term permeability when we are actually referring to the log-permeability.

For building the directed acyclic graph of the NPBBN we must determine the flow of influence between the variables of the state vector, at different locations (see Figure 4.9). This section tries to answer questions like: what is the variable that influences all other variables in the graph? Is permeability directly influencing the grid block pressure or the grid block pressure directly influences the permeability? Or for instance, what can one say about the direction of an arc between pressure and bottom hole pressure?, etc.

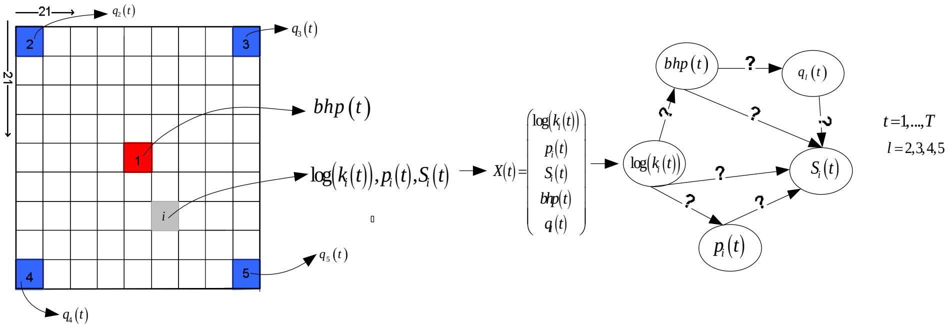


Figure 4.9: Questions about the arcs directionality

The variable that influences all others is the permeability. Firstly, the directionality between permeability and grid block pressure is given by Darcy's law (see formulas 2.4 and 2.5). According to Darcy's law a modification of permeability has an immediate direct effect on the grid block pressure. Therefore arcs will be directed from permeability to grid block pressure.

We shall recall that we inject water at the injector and given the pressure formed in the reservoir oil is pushed out at the producers. In time, water will spread from the injector in the directions of the producers. After some time, typically more than 500-600 days water will get to the producers and not only oil but also water will be pumped out from these wells. This is referred to as the moment when *water breaks through*. Before water breaks through the values of the saturations are very close to 0.2. After water breaks through the saturations values have a sudden increase to values near 0.8. Hence, theoretically saturation has an almost constant value at a given time (0.2 or 0.8). Any constant is independent of any other variable. That is why we have decided to exclude saturations from the NPBBN model.

The relation between the grid block pressure and the bottom hole pressure, and the relations between the grid block pressure and the flow rates are given by the well model [24]. According to this, a modification of the grid block pressure has a direct impact to both bottom hole pressure and total flow rates. In the injector well, the well model gives a relation between the grid block pressure and the observable bottom hole pressure. In the producers, the well model gives the relation between the grid block pressure and the total flow rates. Therefore the arcs in the NPBBN are oriented from the grid block pressures to the bottom hole pressure and total flow rates. Moreover the well model gives a one to one relation between the grid block pressure and the observable bottom hole pressure at the injector well. Also, at the producers the well model gives a one to one relation between the grid block pressure and the observable flow rates. Therefore we do not consider grid block pressures for the locations corresponding to the drilled wells.

What about the directionality of arcs between bottom hole pressure and total flow rates? Since there was no evident answer for this question we have performed experiments for both situations/directions. Our results suggest that there is no major difference between the estimated permeabilities in both situations. Sometimes, differences of order 10^{-1} were observed for the estimated permeabilities. We have decided to arbitrarily consider arcs directed from bottom hole pressure to total flow rates. The NPBBN reprezentening our choices is pictured in Figure 4.10.

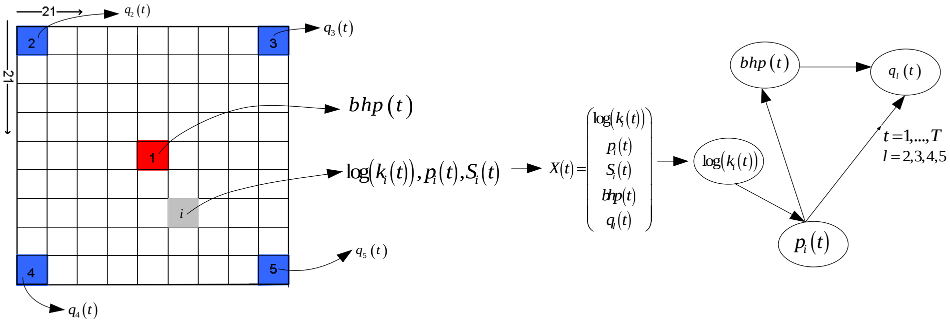


Figure 4.10: Choices of some arcs directionality

But, what about the causality between permeabilities for different locations (see Figure 4.11)? Because the water flows from injector to producers pushing out the oil from the surface, we shall consider the arcs to be directed from the injector to the producers. We have applied the same reasoning for variables like: grid block pressure and permeability.

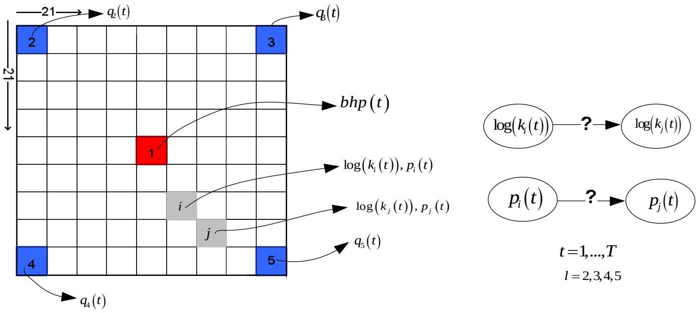


Figure 4.11: Arcs between permeabilities and pressures

Regarding arcs directionality between the flow rates from the four producers there is no obvious answer. Again, one should perform experiments with different changes in the order of the flow rates and see whether any significant changes are noticed in the estimated permeabilities. Rarely differences of 10^{-1} were noticed for the estimates. We have randomly chosen the directionality between the flow rates. A final NPBBN representation of the state vector is given in Figure 4.12.

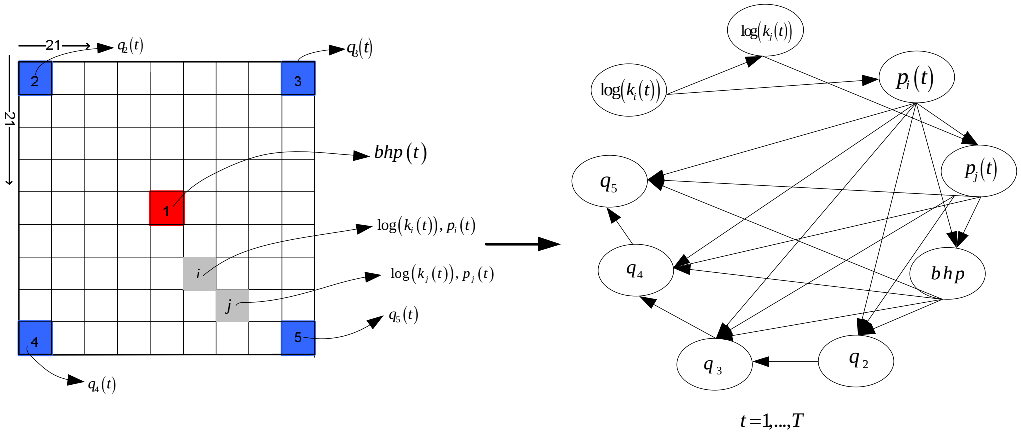


Figure 4.12: The DAG of the NPBBN

4.3 Measures of performance

While performing experiments using different methods one must determine a way of evaluating them. Typically, we can measure the quality of history matching and the quality of predicting the production data.

To measure the quality of the history matching we compare the ‘true’ model parameter with the estimated model parameter (i.e. permeability). A measure of discrepancy is the root mean square error (RMSE). RMSE can be computed for every time step as:

$$RMSE(t)=\sqrt{E\left(k^{true}(t)-k\ \ (t)\right)^{2}=\frac{\sum_{i=1}^{n}\left(k_i^{true}(t)-k_i\ (t)\right)}{n}},t=1,\ldots,T \qquad (4.5)$$

where k^{true} is the true permeability, k is the estimated permeability, t is the time step, n is the number of grid block cells considered (e.g. for *n=441* we compute the RMSE for the entire grid block at every time, for *n=1* we can compute the RMSE for one location at every time step). When RMSE is computed for all locations at every time step, it gives just a general idea of the resemblance to the truth.

On the other hand we may refer to the measures of performance on the basis of the quality of the predicted production data. There are two approaches here. The first one is called *forecast*[8]. After obtaining our estimate for the parameter, forecast can be performed in two ways. One way is to run the simulator further in time from the last time step of estimation. Another way is to run the simulator from the starting time (time zero) with the estimated permeability field and check how much it resembles the true forecast.

[8] Forecast is also called prediction in reservoir engineering.

We can also check the quality of the predicted production data by performing a so called *data mismatch*. Data mismatch shows how well the history matched model is able to reproduce the actual data used for history matching. If the model is able to reproduce the past behavior of the reservoir then, hopefully it will also be able to reproduce future reservoir behavior (forecast). Data mismatch is a scalar defined as:

$$\left(Z-H\left(X\right)\right)^{T}R^{-1}\left(Z-H\left(X\right)\right), \tag{4.6}$$

where Z is the vector of measurements used for the history matching, X is the estimated state, R is the measurements error, $H\left(X\right)$ is the vector of measurements obtained with the estimated permeability field (obtained after the model is matched).

In this thesis we shall only use the RMSE. Nevertheless, when performing a twin experiment it is always helpful to also visually observe the resemblance of the estimated permeabilities with the truth. Therefore, pictures of the estimated permeabilities will be compared with the true permeabilities. One can argue that this is a more intuitive measure of performance; but sometimes, although the RMSE shows a certain general/avergae behavior for an estimated filed, the pictures of the permeabilities may reveal some extra information not captured by the RMSE.

The next Chapter will compare the performance of the two methods: EnKF (with localization) and NPBBN, given this experimental setup.

Chapter 5

Case study

This chapter presents the most important results for approaching a history matching problem using the ensemble Kalman filter method with localization and the non parametric Bayesian belief net based approach. The goal is to compare the performance of the two methods in estimating the permeability field. This chapter is organized as follows: we start by presenting results of estimating permeabilities from four locations (section 5.1). Next, we estimate permeabilities for larger grids: we chose 7x7, 13x13 from the 21x21 grid block (section 5.2). In section 5.3 we perform experiments to check the sensitivity of the method to the initial ensemble chosen from the 1000 realizations at our disposal.

5.1 Permeabilities estimated for 4 locations

We use the synthetic two dimensional squared petroleum reservoir presented in Chapter 4 of this thesis. We arbitrarily chose four different locations for estimating the permeabilities, as shown in Figure 5.1. The next subsection shows the results of estimating the permeabilities using the ensemble Kalman filter with localization with 100 and 300 ensembles. The same setup will be used in subsection 5.1.2 for estimating the permeabilities using the saturated NPBBN graph. In Chapter 3 of this thesis we presented a method for learning the structure of a NPBBN from data. In section 5.1.3 we shall compare results of estimating the desired parameter by using EnKF, the saturated NPBBN and the NPBBN learned from data.

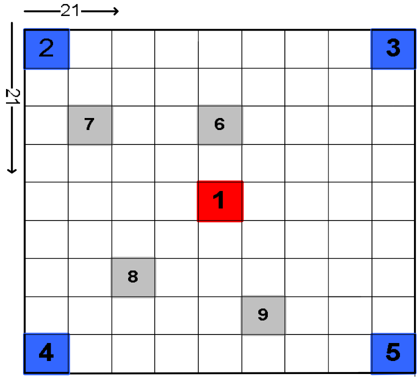


Figure 5.1: Four chosen locations

5.1.1 EnKF (with local analysis)

Our intention is to apply ensemble Kalman filter with localization for estimating the permeability for the filed presented in Figure 5.1. The appendix shows results of experiments using EnKF with local analysis for estimating the permeability field. It has been shown that for our field localization is effective when a small number of ensembles are used (e.g. less than 80 ensembles). When the size of the ensemble is larger than 80, the positive effects of localization are considerably diminished, i.e. the estimated permeability field using EnKF with localization is the same as the estimated permeability using EnKF only. Therefore, when using 100 and 300 ensembles no localization is needed. The estimate of the permeabilities for the field illustrated in Figure 5.1 is presented as the mean of the estimated permeabilities for the ensembles used. Although EnKF is applied to the entire field in this section we shall illustrate results of estimating the permeabilities only in the four different locations of interest . Figure 5.2 shows the true prmeabilities, the initial choice for the permeabilitis and the estimated permeabilities afterv420 days for the four locations. We notice that the permeabilities for all four locations are underestimated. The best estimate is achieved for location 7, whereas the worst is for location 9. As one would expect, the initial choice from location 7 is the best one; i.e. the closest to the true values. Note that the initial from location 9 is quite far from the truth. We may say that the most significant improvement is observed in location 6 since we start with a clearly bad initial choice. Using a similar argument we notice that the smallest improvement to the estimate when using EnKF with 100 ensembles is obtained in location 8.

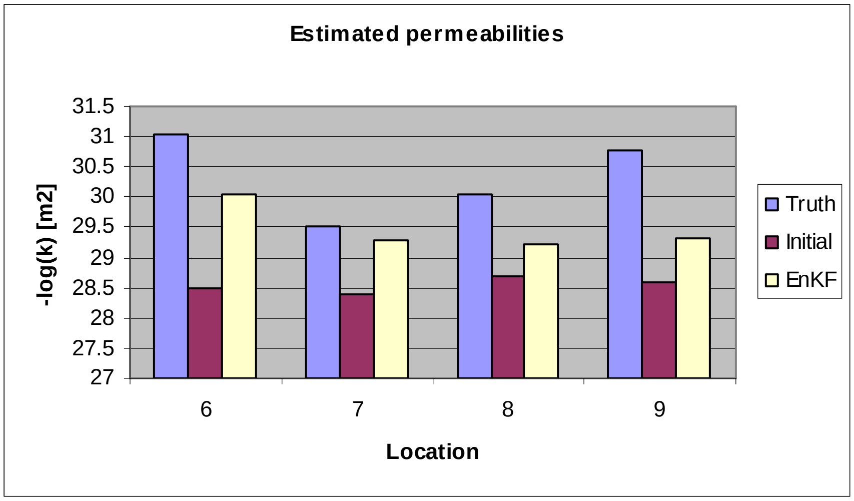


Figure 5.2: Permeabilities estimated using EnKF with 100 ensembles

We increase the number of ensembles to 300. Figure 5.3 presents the truth and the EnKF with 100 and 300 ensembles. One can observe that there are some small differences in the estimated permeabilities between EnKF with 100 ensembles and EnKF with 300 ensembles. Figure 5.4 illustrates the RMSE for each location using both 100 and 300 ensembles. One should expect to obtain better or slightly better results when using EnKF with 300 ensembles than when using EnKF with 100 ensembles. This is the case for locations 7 and 9. However, for locations 6 and 8 the results with 100 ensembles are slightly better than the results with 300 ensembles. The differences are visible in Figures 5.3 and 5.4. In search of an explanation for this behavior we looked at the initial ensemble. A better initial might give a better estimate. However, Figure 5.5 shows the plot of the truth together with the mean of the initial for 100 and 300 ensembles. The absolute differences between the initials are all of order 10^{-2} . This is actually suggesting that the initial ensemble is not influencing the final result.

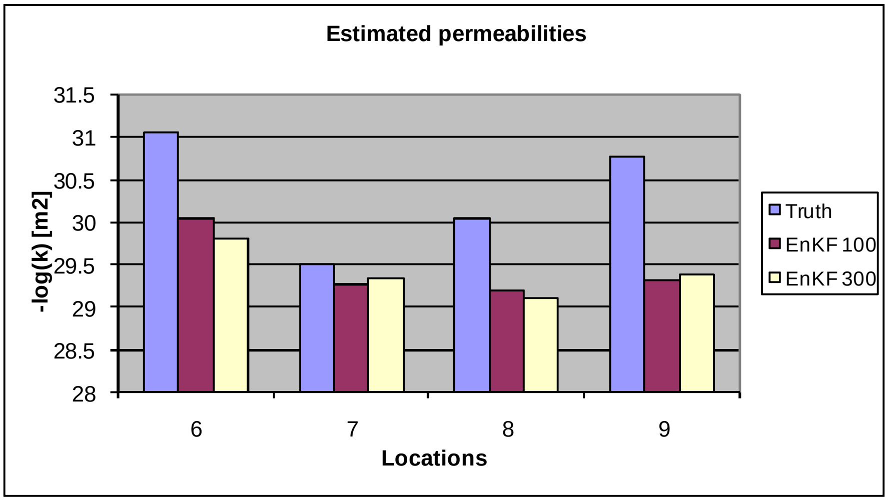


Fig ure 5.3: Permeability estimated using EnKF with 100 versus 300 ensembles

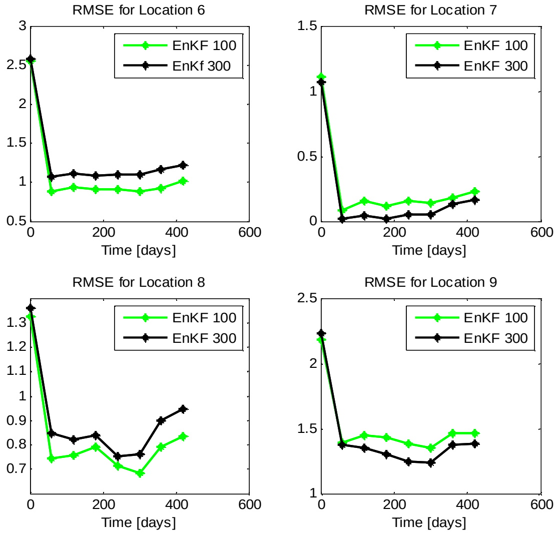


Figure 5.4: RMSE for each location using EnKF with 100 ensembles versus EnKF with 300 ensembles

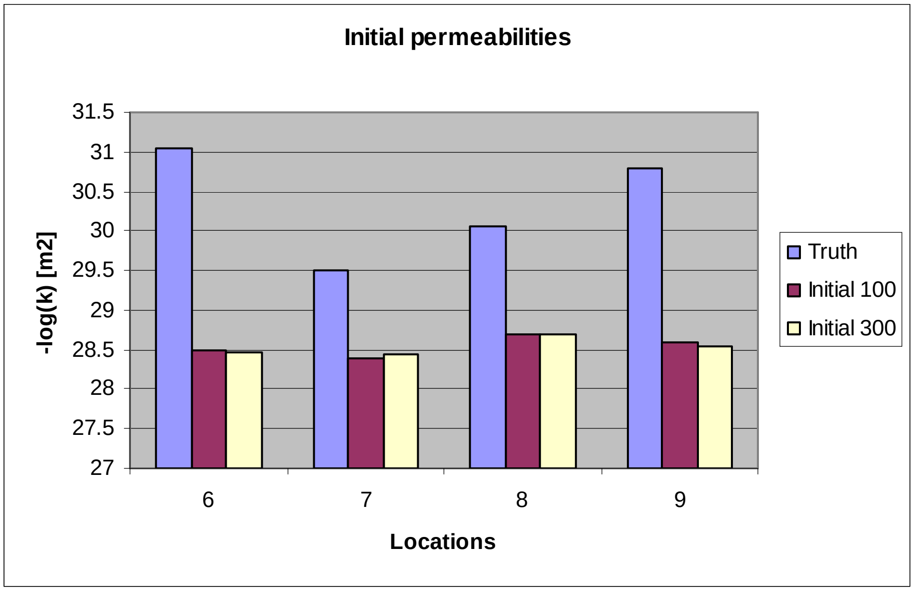


Figure 5.5: Initial permeabilities for 100 ensembles versus 300 ensembles

5.1.2 The Saturated NPBBN

We shall use the NPBBN based approach to estimate the permeabilities from the four locations specified in the previous section. Experiments with 100 and 300 ensembles will be performed.

First we consider 100 ensembles. We start by running the simulator 'simsim' for 60 days. We obtain the joint distribution of the variables after 60 days. This joint distribution is given in the form of a data set that can be mined and represented using a static NPBBN. Our intention is to learn a model that contains the variables defined in the state vector presented in section 4.1. Each node in the graph will be associated with a variable from the state vector at a certain location of the grid. In section 4.2.3 we illustrated the choice of arcs directionality in the NPBBN. Given the above, we can build a static saturated NPBBN model corresponding to the four locations specified in Figure 5.1. We first consider the saturated graph because we want to represent all possible dependencies. Afterwards, we shall simplify the structure of the NPBBN model by using the learning algorithm presented in section 3.3. In order to represent the data using a NPBBN we assume the normal copula (see section 3.1.2). The assumption must be validated. We use the determinant of the correlation matrix as a measure of dependence. Given a data set, we may distinguish:

- *DER*, the determinant of the empirical rank correlation matrix;
- *DNR*, the determinant of the rank correlation matrix obtained by transforming the univariate distribution to standard normals and then transforming the product moment correlations to rank correlations using Pearson's transformation;
- *DBBN*, the determinant of the rank correlation matrix of a NPBBN using the normal copula.

A statistical test for validating the joint normal copula is to obtain the sampling distribution of *DNR* and check whether *DER* falls within the 90% confidence band of *DNR*. If *DNR* is not rejected on the basis of this test, we shall build a NPBBN which represents the data set.

Any location that is not a drilled well location has its corresponding grid block pressure and permeability as shown in Figure 5.6. We denote the grid block pressures by *p_6, p_7, p_8, p_9*; and we denote the permeabilities by *k_6, k_7, k_8, k_9*. Moreover we measure bottom hole pressure (*bhp)* at the injector well and the total flow rates denoted by *total_rate_1, total_rate_2, total_rate_3, total_rate_4,* at each producer. Therefore, we are interest in the joint distribution of *13* variables.

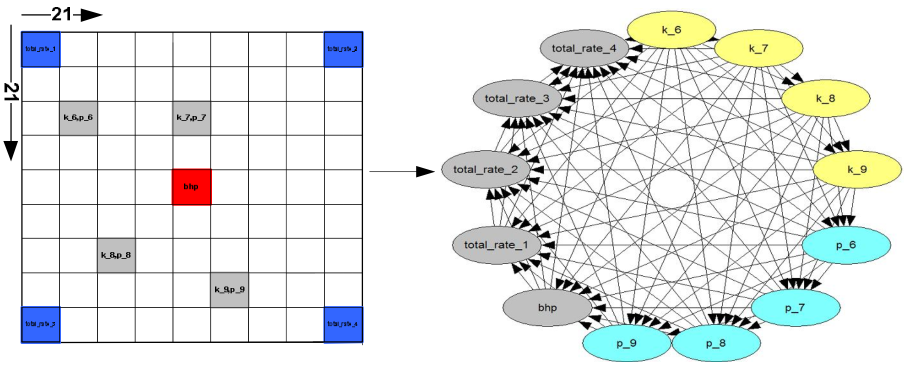


Figure 5.6: The Saturated NPBBN for four locations

The NPBBN model is build in Uninet. By selecting the option *view nodes as histograms* in Uninet we can see the distribution of each variable. Figure 5.7 illustrates the NPBBN model from Figure 5.6 with all nodes viewed as histograms. The mean and standard deviation of each variable is shown in the respective histogram.

The normal copula assumption after 60 days is validated, i.e. the *DER* falls within the 90% confidence band of the *DNR*. We obtained that $DER = 6.21 \times 10^{-9} \in [3.66;85.26] \times 10^{-9}$.

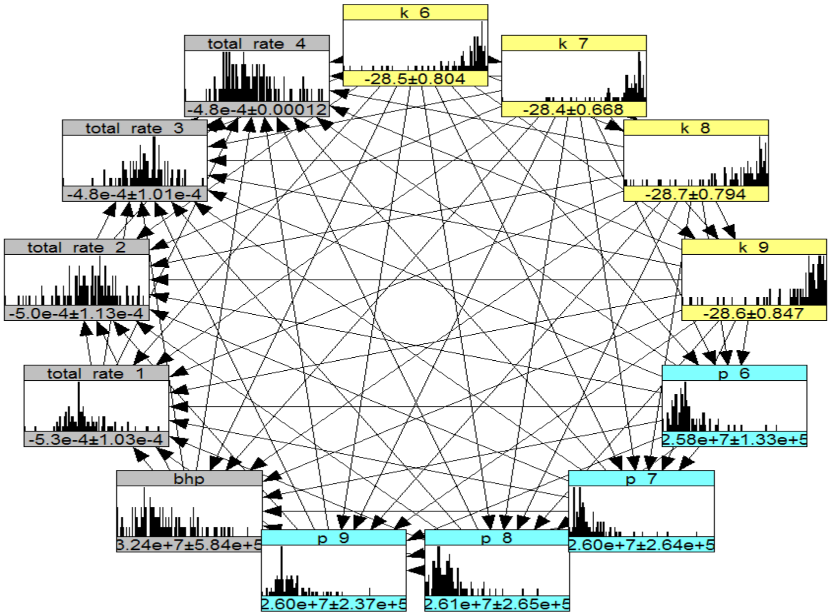


Figure 5.7: Saturated NPBBN with histograms. The marginal distributions of the variables after 60 days

One of the most important features of the NPBBN is that they can be used for inference. Hence, the static NPBBN will be used to perform the conditionalization/ assimilation step.

Given the observed values of measurements at the wells, we can calculate the conditional distributions of the other variables. This is done in Uninet by switching to *analytical conditioning view.* One can notice that the observable variables are not normally distributed. Nevertheless, Gaussian noise is added when generating measurements.

The conditioning is performed analytically as explained in Chapter 3. Figure 5.8 shows the same NPBBN from previous pictures conditioned on the observations from the wells. We notice the change in the distributions of the unobserved nodes. The grey distributions are the original distributions of the variables and the black ones are the conditional distributions.

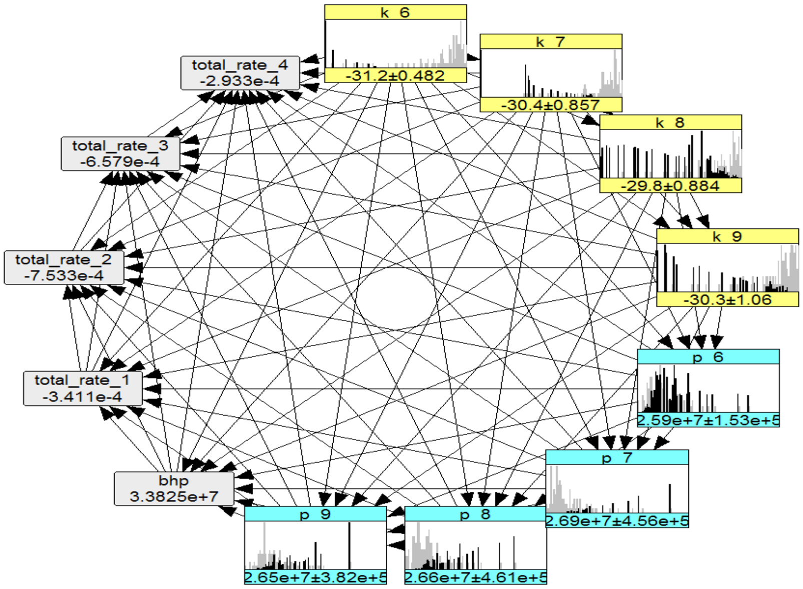
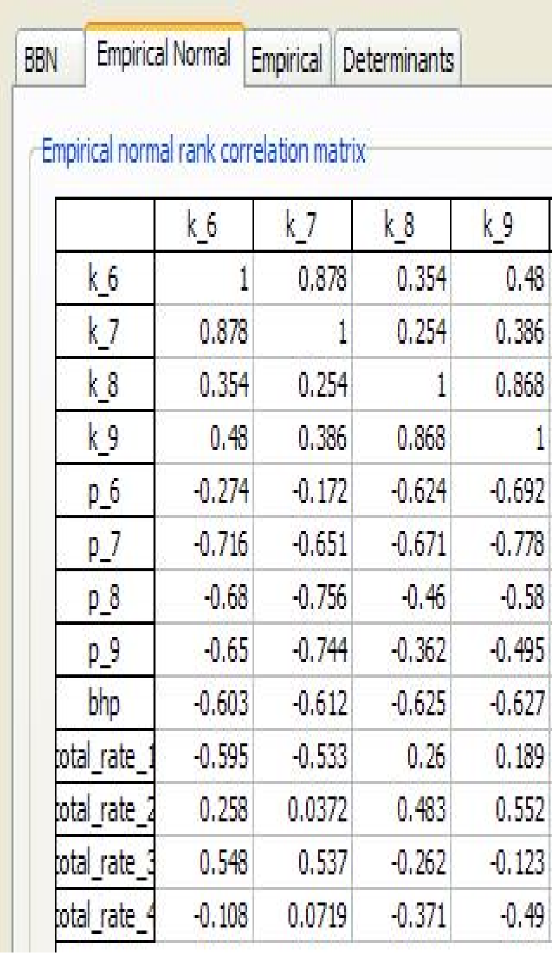


Figure 5.8: Conditional distribution of 8 out of 13 variables.

One of the assumptions of the EnKF method is that the conditional distribution is Gaussian. Figure 5.8 contradicts this statement by showing that not even the margins are normally distributed. After conditioning, we stipulate the conditional distribution by sampling it. We shall use the sampling procedure illustrated in Chapter 3 of this project. Further, we introduce in ‘simsim’ the updated values (the grid block pressures and permeabilities) and we run the simulator for another 60 days. In this way we shall obtain the distribution of the variables after 120 days (with assimilation step after 60 days). The new joint distribution will be modeled with another static NPBBN. The two NPBBNs connected through the ‘simsim’ simulator are basically a dynamic BBN, where the temporal relations are functional. We shall repeat the above steps for a period of 420 days. Every time step we validate the normal copula assumption. Hence, we build a dynamic BBN for seven discrete times.

In Uninet we also have the possibility of looking at the rank correlation matrix for each static NPBBN model. Figure 5.9 illustrates the correlations between permeability and other variables after 60 days (before the first conditionalization/assimilation step), 120 days and 420 days. Generally, the absolute values of the rank correlations between permeabilities decrease after the first time step (first 60 days) and tend not to be recovered by the last time step. For instance, the absolute values of the rank correlation between *k_6* and *k_7* is *0.878* after 60 days, it becomes *0.727* after 120 days and by the last time step it decreases to *0.444*. When referring to the absolute value of the rank correlations between permeabilities and grid block pressures one can say that some rank correlations recover while others do not. For example, the absolute values of the rank correlation between *p_6* and *k_6* is *0.274* before the first conditionalization/assimilation time step, it decreases to *0.0421* after the second time step and by the 420 days it increases to *0.107*. However, the absolute value of the rank correlation between *p_9* and *k_9* decreases from *0.495* at time 1 to *0.0368* at time 2. At time 7 the absolute value of the rank correlation continues to decrease to *0.00868*. Most of the absolute value of the correlations between the permeabilities and the observable variables (bottom hole pressure and total flow rates) tend to decrease after the first time step and then some of them show a little increase by the end of the simulation.

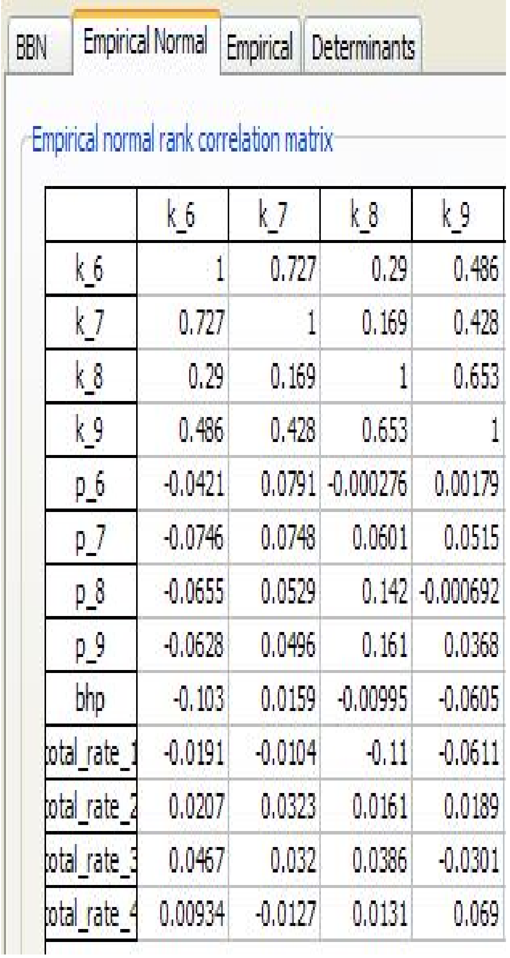
Rank correlations - Time 1



Empirical normal rank correlation matrix

	k_6	k_7	k_8	k_9
k_6	1	0.878	0.354	0.48
k_7	0.878	1	0.254	0.386
k_8	0.354	0.254	1	0.868
k_9	0.48	0.386	0.868	1
p_6	-0.274	-0.172	-0.624	-0.692
p_7	-0.716	-0.651	-0.671	-0.778
p_8	-0.68	-0.756	-0.46	-0.58
p_9	-0.65	-0.744	-0.362	-0.495
bhp	-0.603	-0.612	-0.625	-0.627
total_rate_1	-0.595	-0.533	0.26	0.189
total_rate_2	0.258	0.0372	0.483	0.552
total_rate_3	0.548	0.537	-0.262	-0.123
total_rate_4	-0.108	0.0719	-0.371	-0.49

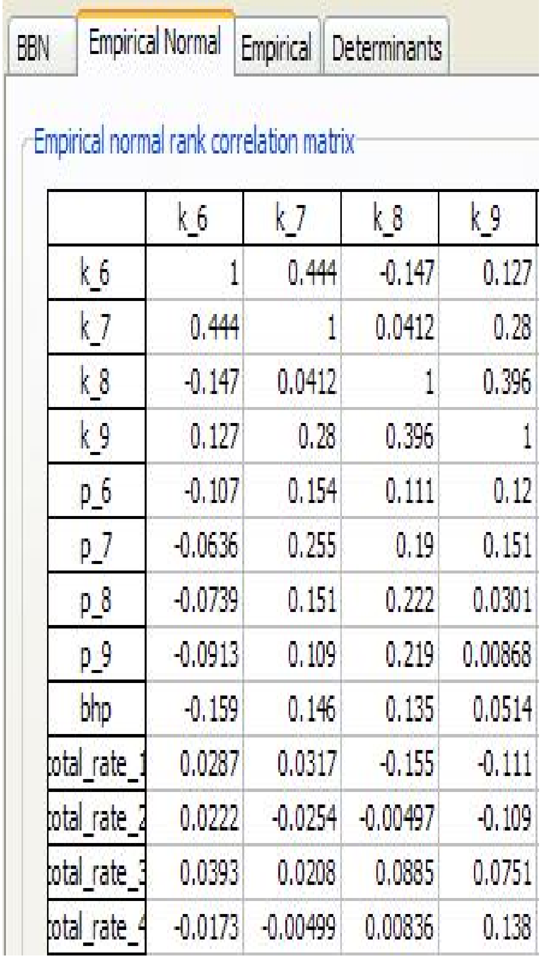
Rank correlations – Time2



Empirical normal rank correlation matrix

	k_6	k_7	k_8	k_9
k_6	1	0.727	0.29	0.486
k_7	0.727	1	0.169	0.428
k_8	0.29	0.169	1	0.653
k_9	0.486	0.428	0.653	1
p_6	-0.0421	0.0791	-0.000276	0.00179
p_7	-0.0746	0.0748	0.0601	0.0515
p_8	-0.0655	0.0529	0.142	-0.000692
p_9	-0.0628	0.0496	0.161	0.0368
bhp	-0.103	0.0159	-0.00995	-0.0605
total_rate_1	-0.0191	-0.0104	-0.11	-0.0611
total_rate_2	0.0207	0.0323	0.0161	0.0189
total_rate_3	0.0467	0.032	0.0386	-0.0301
total_rate_4	0.00934	-0.0127	0.0131	0.069

Rank correlations - Time 7



Empirical normal rank correlation matrix

	k_6	k_7	k_8	k_9
k_6	1	0.444	-0.147	0.127
k_7	0.444	1	0.0412	0.28
k_8	-0.147	0.0412	1	0.396
k_9	0.127	0.28	0.396	1
p_6	-0.107	0.154	0.111	0.12
p_7	-0.0636	0.255	0.19	0.151
p_8	-0.0739	0.151	0.222	0.0301
p_9	-0.0913	0.109	0.219	0.00868
bhp	-0.159	0.146	0.135	0.0514
total_rate_1	0.0287	0.0317	-0.155	-0.111
total_rate_2	0.0222	-0.0254	-0.00497	-0.109
total_rate_3	0.0393	0.0208	0.0885	0.0751
total_rate_4	-0.0173	-0.00499	0.00836	0.138

Figure 5.9: The rank correlations between the permeability and other variables in three discrete times.

The estimated permeabilities from the four locations using the saturated NPBBN based method are showed in Figure 5.10. When 100 ensembles are used we observe that the permeabilities using the saturated NPBBN are overestimated. The best estimates are obtained for locations 6 and 9, while the worst estimate is obtained for location 7. In Figure 5.12 the RMSE for each time step for each location is illustrated. We notice that the RMSE has an increasing behavior for most of the locations of interest when using this number of ensembles. One reason could be that the joint distribution of 13 variables is not described well enough by only 100 ensembles. Hence, we increase the size of the ensembles used to 300. We chose to use 300 ensembles because the correlation matrix is stable when using 300 ensembles, i.e. the differences between the correlation matrices calculated from 300 and 400 ensembles are of order 10^{-2}. Therefore, we expect to improve our estimate by using 300 ensembles.

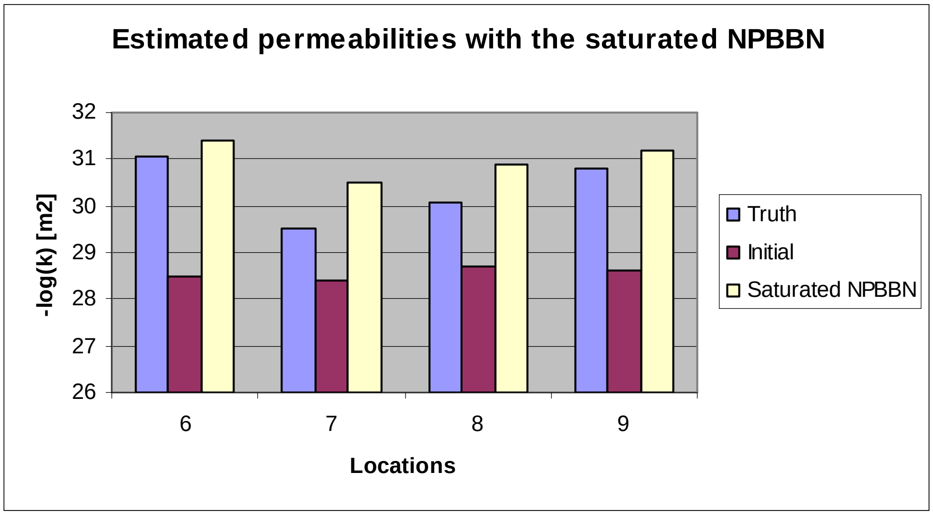


Figure 5.10: Estimated pemeabilities with the saturated NPBBN using 100 ensembles.

We build a static saturated NPBBN model using 300 ensembles for every time step. Every time step we validate the normal copula assumption. For instance after 60 days, *DER* is within the *90%*

confidence band of DNR’s distribution, i.e. we obtain that $DER = 1.29 \times 10^{-8} \in [5.84; 97.6] \times 10^{-8}$.

Figure 5.11 compares the results of estimating the permeabilities from the four locations using 100 and 300 ensembles. As expected, the saturated NPBBN with 300 ensembles performs better than the saturated NPBBN with 100 ensembles. The largest improvement is noticed for locations 8 and 9. Small improvements are observed for locations 6 and 7. Locations 6, 8 and 9 have a RMSE smaller than 0.5 (see Figure 5.12). Note that although improved, the RMSE for location 7, shows an oscillating (unstable) behavior.

Looking at the correlations between the permeabilities from the four locations and other variables after 60 days, 120 days and 420 days we observe the same general behavior as when using 100 ensembles. Therefore, the correlations decrease considerably after the first 60 days. Some of the correlations tend to slightly increase by the 7th, but most of them remain very small. From a correlation of order 10^{-1} at the fist time step, one can see a correlation of order 10^{-3} at the 7th time step. If we compare the correlation matrix obtained using 100 ensembles with the one obtained when using 300 ensembles, the correlations are smaller in the latter situation for all time steps. Differences of order 10^{-1} can be noticed between the correlation matrix obtained when using 100 ensembles and the correlations matrix obtained when using 300 ensembles. This suggests that the ‘real’ correlations are indeed very small after the first time step.

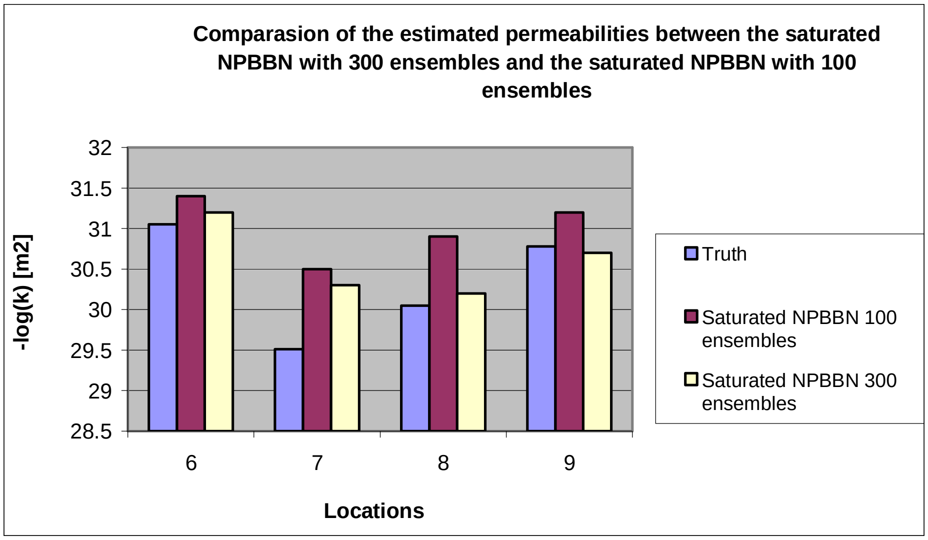


Figure 5.11: Mean of the estimated permeabilities using 100 and 300 ensembles

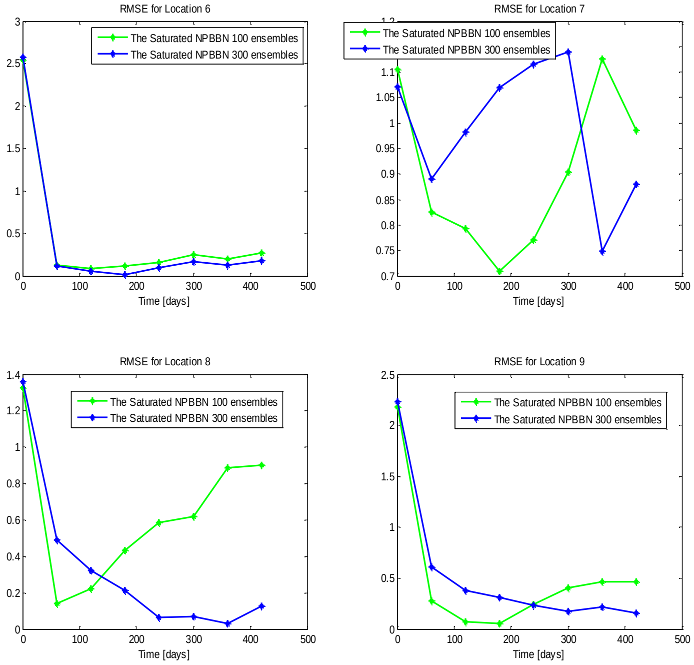


Figure 5.12: RMSE for each location when using the saturated NPBBN with 100 ensembles versus the saturated NPBBN with 300 ensembles

5.1.3 The Learned NPBBN

The previous section illustrates results of estimating the permeabilities using the saturated graph. In this situation one considers all possible dependencies from a data base. However, some of the correlations might be unreal because they are due to sampling fluctuations. Our goal is to build a NPBBN model that eliminates these unreal correlations while approximating the saturated graph. In this section we shall learn the structure of the static NPBBN models used in the analysis. We are using the learning algorithm illustrated in section 3.3. When learning the structure of a NPBBN that approximates the saturated graph we introduce conditional independence relations. Hence one should validate that the learned model adequately represents the saturated graph. Again we use the determinant of the correlation matrix as an overall measure of dependence. For validating that the learned model adequately represents the saturated graph we use a validation test similar to the one presented in the previous section for validating the joint normal copula assumption. We compute the sampling distribution of *DBBN* and we check whether *DNR* is within the *90%* confidence band of *DBBN*. If *DBBN* is not rejected on the basis of this test we validate the learned NPBBN model.

Firstly we consider the NPBBN using 100 ensembles. Some arcs are deleted from the saturated NPBBN. We validate a simplified model (see Figure 5.13) that approximates the saturated graph. We obtain that *DNR* is within the *90%* confidence band of the distribution of DBBN, i.e. $DNR = 2.01 \times 10^{-8} \in [1.71;44.18] \times 10^{-8}$. Every time step we learn a static NPBBN. Both the normal copula assumption and the model were validated for every static NPBBN. In this way a dynamic NPBBN with changes structure over time is built. Figure 5.14 shows the estimated permeabilities with the saturated NPBBN versus the learned NPBBN after 420 days. We notice that for some locations (e.g. location 6 and location 7) the estimate improves when using the learned NPBBN. For location 8 almost the same estimate is obtained when using both the saturated and the learned graph, although the saturated graph seems to perform slightly better. For location 9 the saturated NPBBN gives a better estimate than the learned one. These results can be observed in Figure 5.14 where the mean permeabilities are plotted, and also in Figure 5.15 where we compare the RMSE for each location. Sometimes the saturated RMSE gives better estimates than the learned one. This can be the case if we eliminate small, yet real correlations. Nevertheless, if using the same NPBBN we encounter the opposite situation as well (the learned NPBBN better than the saturated NPBBN) for some locations. Drawing an unifying conclusion becomes difficult. The fact that both situations are encountered for the same NPBBN, for different locations might also be a sign of numerical instabilities.

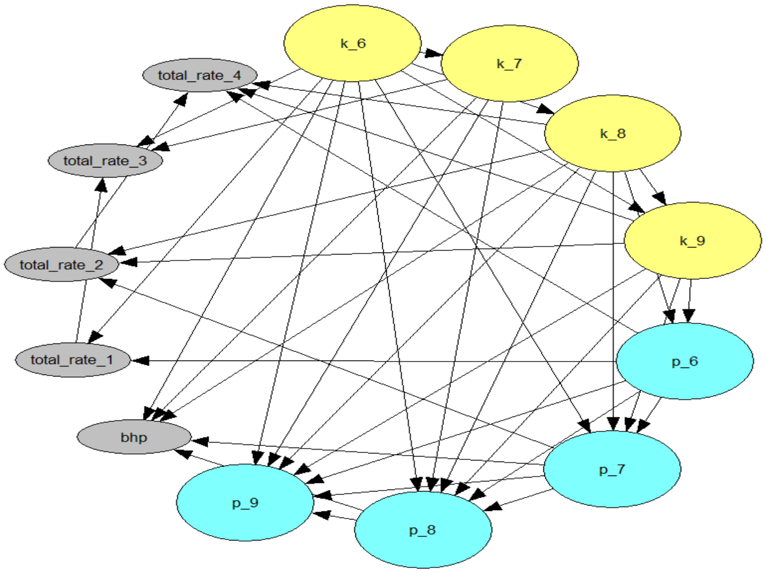


Figure 5.13: The learned NPBBN when using 100 ensembles

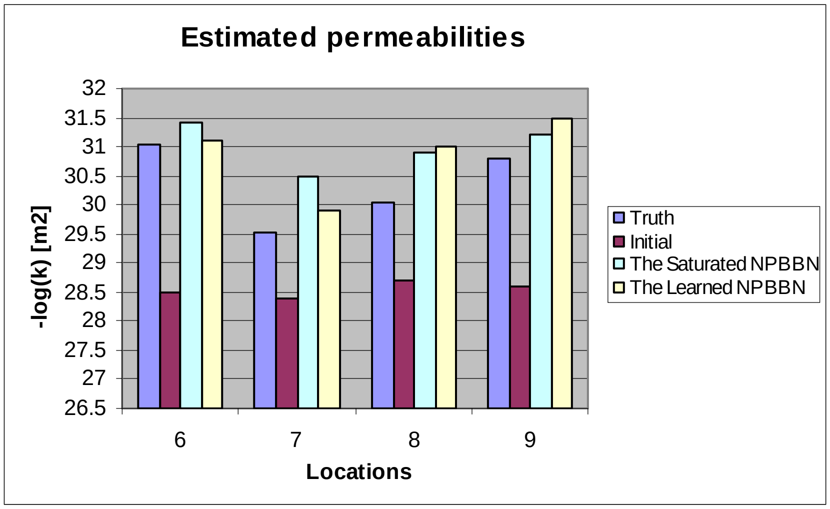


Figure 5.14: The estimated permeabilities using the saturated NPBBN versus the learned NPBBN with 100 ensembles

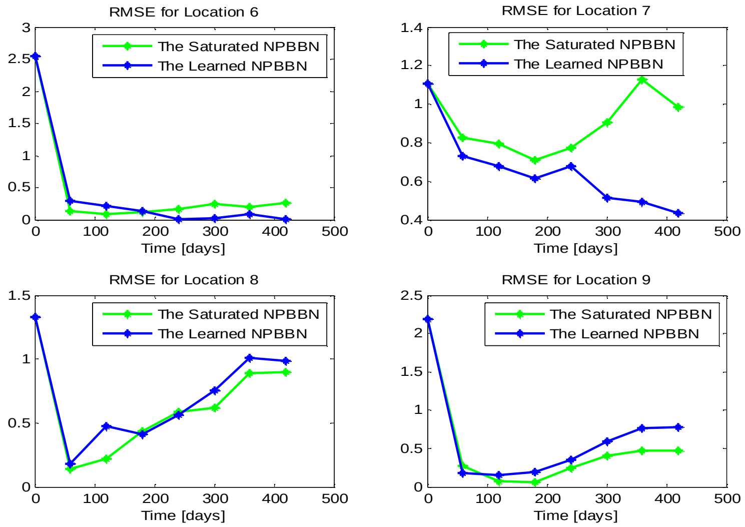


Figure 5.15: RMSE for each location using the saturated NPBBN versus the learned NPBBN with 100 ensembles

Next, we shall learn the structure of the NPBBN model when using 300 ensembles. A learned model that adequately represents the saturated graph is validated every time step. For example, after 60 days we validate a learned model for which *DNR* falls within the 90%

confidence band of *DBBN's* distribution, i.e. $DNR = 2.63 \times 10^{-8} \in [2.38;13.17] \times 10^{-8}$.

Figure 5.16 illustrates results of estimating the permeabilities after 420 days. We observe that the saturated graph performs better for locations 7, 8, 9. For location 6 the saturated and the learned NPBBN perform equally well. The plot from Figure 5.17 shows the RMSE for each location, for the saturated NPBBN and the learned NPBBN. Therefore for all time steps, for most of the locations, the saturated graph gives a better estimate than the learned one. If comparing results of applying the learning procedure for the graph when using 100 and 300 ensembles, one notices that as the number of ensembles increases there is less need of deleting arcs from the NPBBN. That is because in this case, the dependence structure is much better represented with 300 ensembles that just with 100. A similar situation was noticed when applying localization to the EnKF method. As the number of ensembles used for the EnKF increased, there was less need to apply localization.

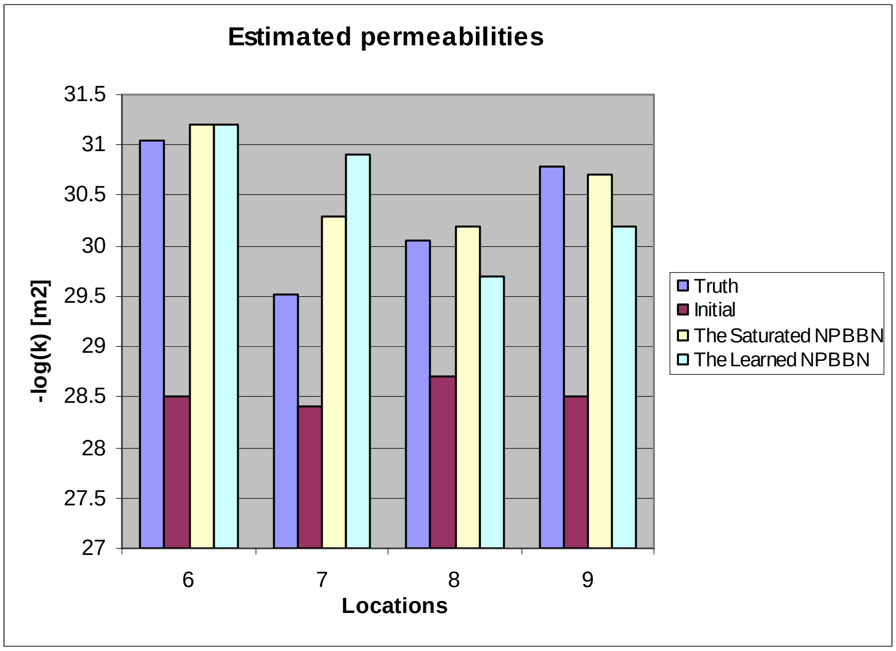


Figure 5.16: Estimated permeabilities using the saturated versus the learned NPBBN with 300 ensembles

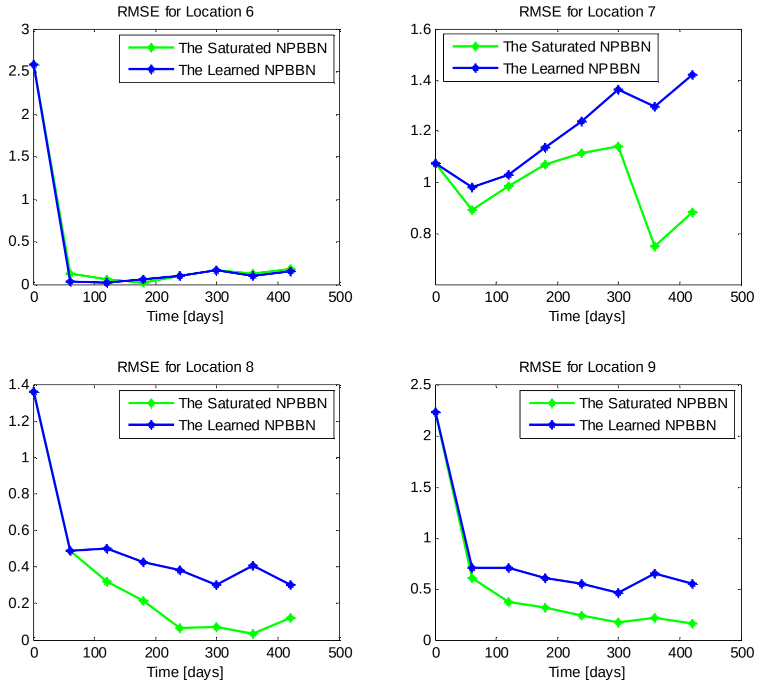


Figure 5.17: RMSE for each location using the saturated versus the learned NPBBN with 300 ensembles

5.1.4 EnKF versus NPBBN

In this section we shall compare the results from the previous sections when using the EnKF method and the NPBBN based approach.

5.1.4.1 EnKF versus NPBBN for 100 ensembles

Figure 5.18 shows the estimated permeabilities after 420 days using the EnKF, the saturated and the learned NPBBN for 100 ensembles. In Figure 5.19 the RMSE for each location is illustrated. We notice that the permeabilities from all four locations are underestimated when using the EnKF and overestimated when using both the saturated and the learned NPBBN. The NPBBN gives a better estimate than EnKF for locations 6 and 9. For location 6 the learned graph gives a better estimate, while for location 9 the saturated NPBBN is closer to the truth. EnKF performs better than both the saturated and the learned NPBBN for location 7. With respect to location 8, we notice that all methods give similar estimates. However if we look at the RMSE, the EnKF method seems to give a slightly better estimate than NPBBN based approach for this location.

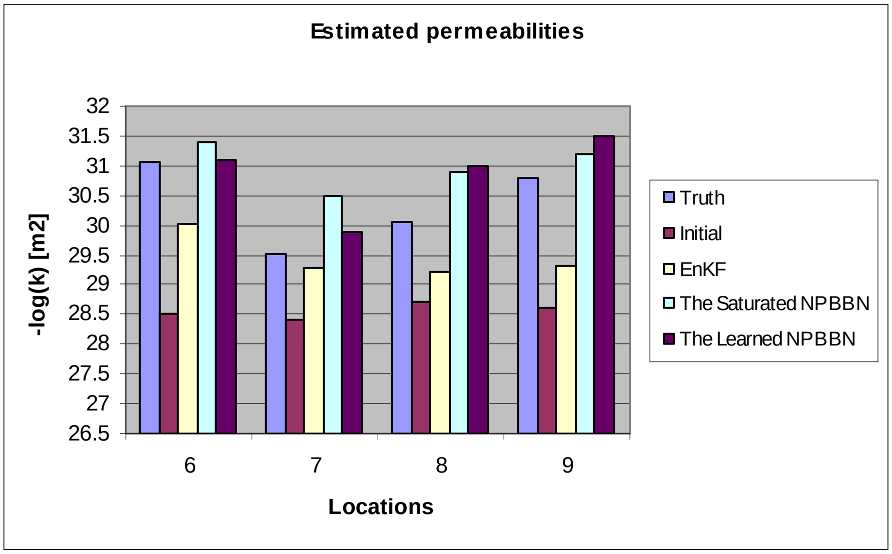


Figure 5.18: Comparative results of the estimated permeabilities using the EnKF versus the NPBBN based approach (with 100 ensembles)

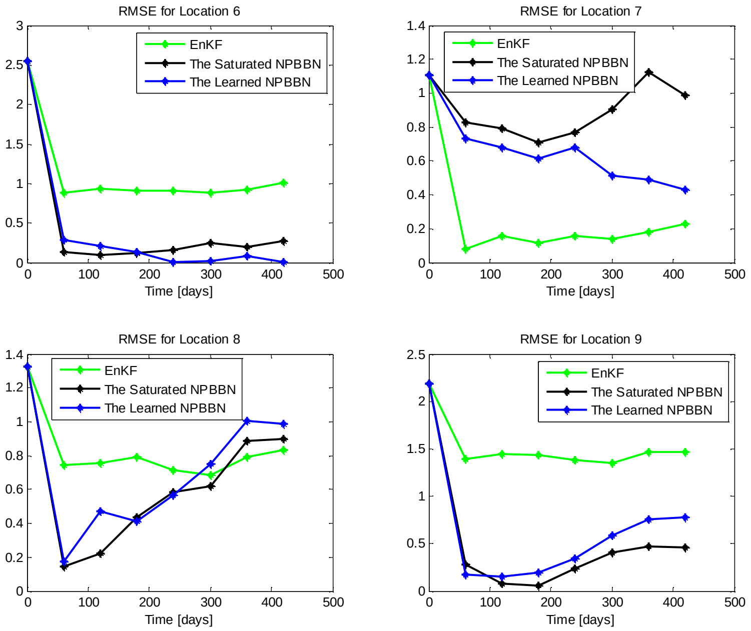


Figure 5.19: RMSE for every location when using EnKF versus NPBBN based approach (with 100 ensembles)

5.1.4.2 EnKF versus NPBBN for 300 ensembles

When increasing the size of the ensembles to 300, the situation changes for most of the locations in favor of the NPBBN method. As shown in section 5.1.1 of this chapter the EnKF method does not show any major improvements when the size of the ensemble is increased from 100 to 300. But, the NPBBN method, especially when using the saturated graph, gives much better results when we used 300 ensembles compared with 100 ensembles. Figure 5.20 presents the results of the estimated permeabilities after 420 days. In Figure 5.21 the RMSE is showed for each location. For locations 6, 8 and 9 both the saturated and the learned NPBBN perform better than the EnKF method. However, EnKF still gives a better estimate than NPBBN (both the learned and the saturated graph) for location 7.

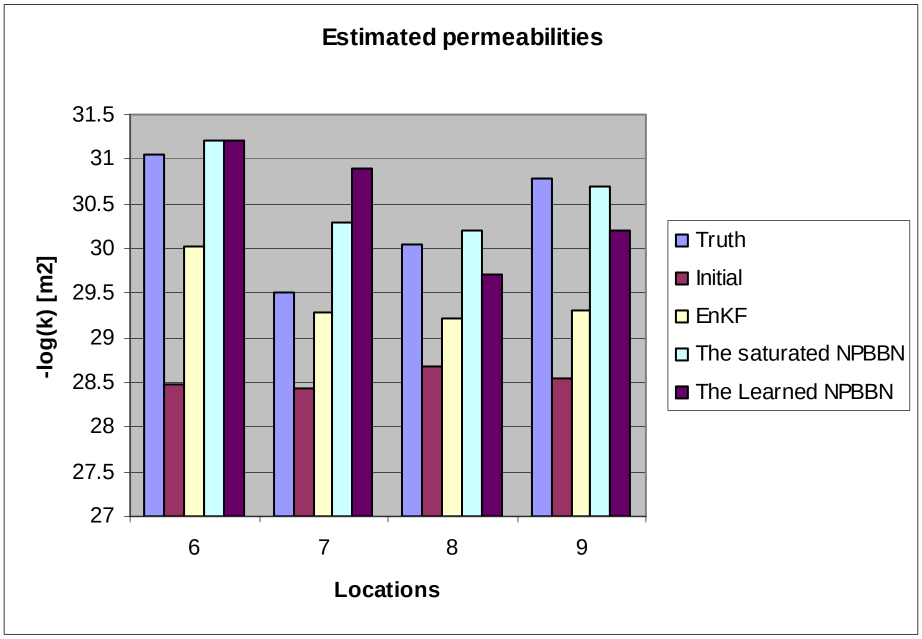


Figure 5.20: Comparative results of the estimated permeabilities using the EnKF versus the NPBBN based approach (with 300 ensembles)

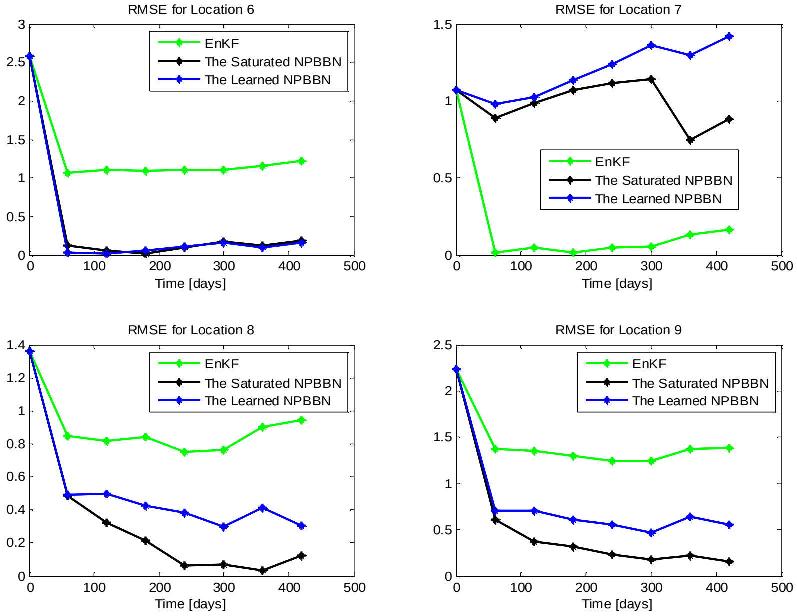


Figure 5.21: RMSE for every location when using EnKF versus NPBBN based approach (with 300 ensembles)

5.2 Permeabilities estimated for a medium size grid block

This section illustrates results of estimating the permeabilities from a 7x7 and a 13x13 square grids (see Figure 5.22). For estimating the 49 permeabilities from a 7x7 grid we have 103 variables involved in the NPBBN based approach. When estimating the permeabilities for the 13x13 grid we have 423 variables included in the NPBBN model. For these number of variables used, the determinants of the correlation matrices have very small values, e.g. values of order 10^{-200}. Therefore the values of the determinants become meaningless and the validation steps inconclusive. Hence, we were not able to validate the normal copula assumption and a model that approximates the saturated graph. In these conditions we performed experiments using the saturated graph and not the learned NPBBN. We shall present comparative results of estimating the permeabilities using the EnKF and the saturated NPBBN. It is worth mentioning that we use 900 initial ensembles chosen from the 1000 realizations.

Firstly we consider the 7x7 grid block chosen as in Figure 5.22 left. We build a static saturated NPBBN for every time step for a period of 480 days. An example of such a saturated NPBBN is showed in Figure 5.23. A saturated graph on 103 variables contains 5253 arcs, so is practically impossible to visualize all nodes and arcs in Figure 5.23.

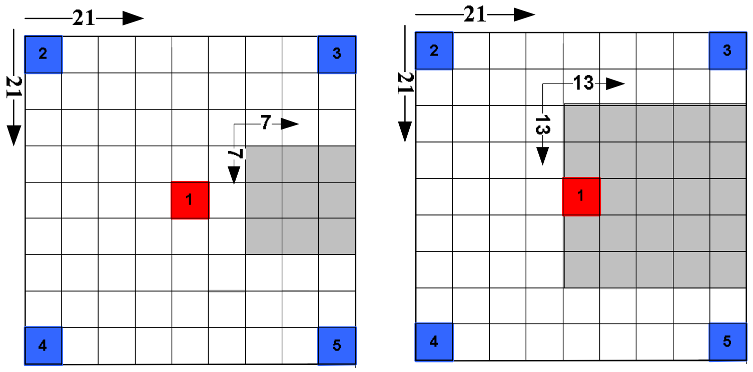


Figure 5.22: Medium size grid blocks chosen from the 21x21 grid block

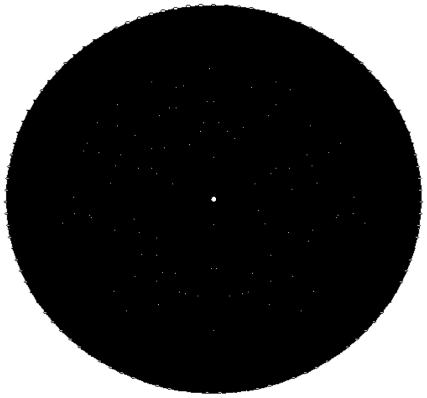


Figure 5.23: An example of the saturated NPBBN for the 7x7 grid block

Figure 5.24 illustrates comparative results using both methods for the 7x7 grid. It presents the true permeabilities, the 900 initial ensembles, the estimated permeabilities after 480 days using the EnKF and the saturated NPBBN. We notice that the saturated NPBBN gives a better estimate than the EnKF for this grid. Figure 5.25 shows the RMSE at each time step. The behavior of the RMSE is quite different for the two methods. For the EnKF, the RMSE decreases at the first time step and then has an oscillating behavior with a tendency to stabilize around the 4th time step to a value of *0.7*. The RMSE for the NPBBN has an increase after the first time step and afterwards is decreasing for every time step reaching a value of 0.5 by the 8th time step. Theoretically one would expect that the RMSE would decrease for every time step because as more data is assimilated, as closer the estimate should get to the truth. However, in practice , this seems not always to be the case. The NPBBN method uses rank correlations in the analysis. After the first time step the rank correlations between variables seemed to be quite large, as observed in section 5.1. the estimation of the correlation matrix is very sensitive to noise, so a wrong input can be amplified and and generate very wrong estimates. Maybe this could be an explanation of the RMSE behaviour after the first time step when using the NPBBN method. When more information is added to the model this seems to improve the performance as expected. On the other hand it seems that after the 4th time step EnKF can not assimilate more information.

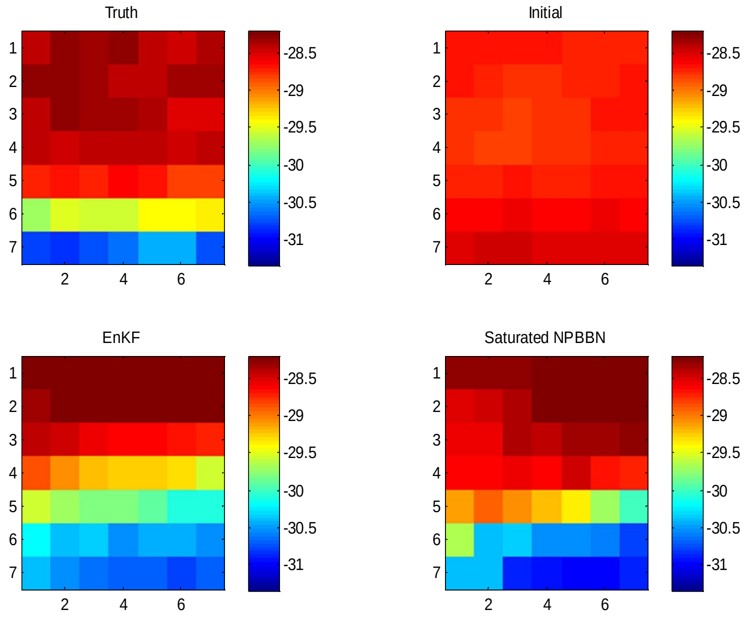


Figure 5.24: Top-left: the true permeability field. Top-right: the initial permeability field. Bottom-left: the permeability field estimated with the EnKF. Bottom-right: the permeability field estimated with the saturated NPBBN.

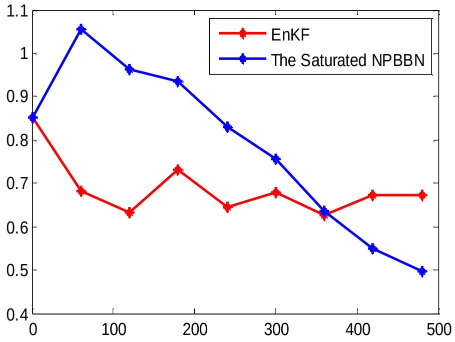


Figure 5.25: RMSE for the estimated permeabilities using EnKF versus the saturated NPBBN

We shall move on to present the results of estimating the permeabilities for the 13x13 grid as presented in Figure 5.22 right. Again, we illustrate comparative results using the EnKF and the saturated NPBBN with 900 initial ensembles. In Figure 5.26 we show the estimated permeabilities after 480 days when using both methods. The results indicate that the two methods are comparable. One could say that the two fields look equally well or even that the estimate using the saturated NPBBN looks slightly better than the field estimated using EnKF. However, the RMSE from Figure 5.27 contradicts the visual analysis. The RMSE for the entire field is clearly smaller and more stable for the EnKF than for the saturated NPBBN. The RMSE for enKF has a considerable decrease after the first time step and then a constant behavior, it stabilizes around the value of *0.5*. Note that the RMSE for EnKF tells as that after the first time step the EnKF does not really assimilate anything. We observe that the RMSE for the saturated NPBBN decreases after the first 60 days, it generally increases for the next five time steps and eventually decreases again for the last two assimilation step. One could ask whether the RMSE for the saturated NPBBN will continue to decrease if we assimilate further. When assimilating for one more time step, we observe that the RMSE increases again (see Figure 5.27). One possible explanation of this unstable behavior can be the fact that the number of ensembles used for this number of variables is too small. Note that we used 900 ensembles for 423 variables included in the saturated NPBBN. However, differences between the correlation matrices calculated with 800 and 900 ensembles are of order 10^{-2}. This suggests that 900 ensembles should suffice for 342 variables accounted in the saturated NPBBN. Still, this might not be the only way of checking whether the number of ensembles used is large enough.

With respect to the ensembles used in next section we shall present a set of experiments that point out the fact that there is dependency between the 1000 realizations from which we have chosen the initials. The sensitivity of both methods to the choice of the initial ensemble will be investigated.

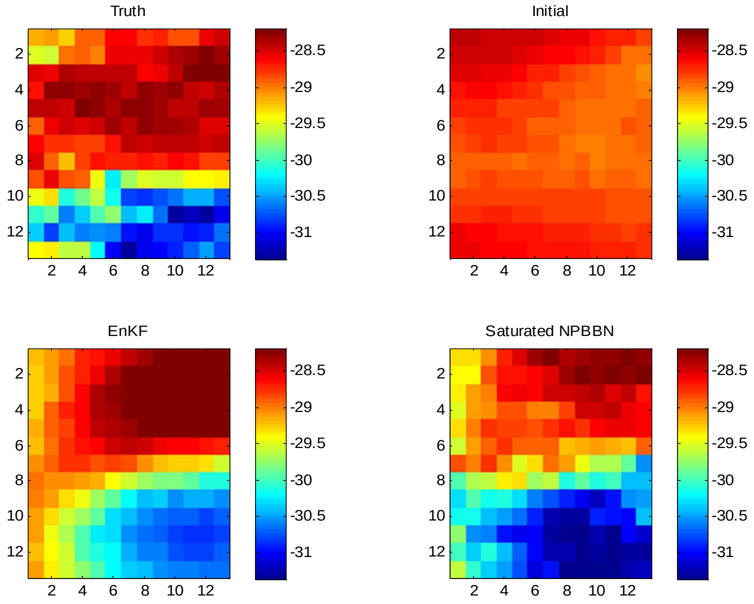


Figure 5.26: Top-left: the true permeability field. Top-right: the initial permeability field. Bottom-left: the permeability field estimated with the EnKF. Bottom-right: the permeability field estimated with the saturated NPBBN.

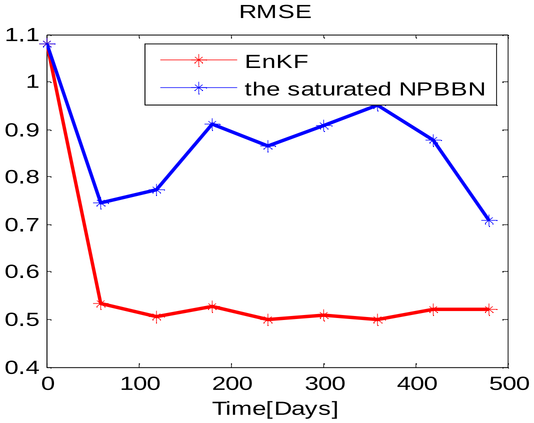


Figure 5.27: RMSE for the estimated permeabilities using EnKF versus the saturated NPBBN

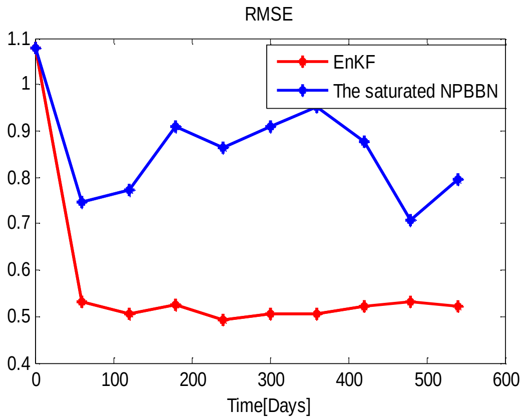


Figure 5.28: RMSE for the estimated permeabilities using EnKF versus the saturated NPBBN for a period of 540 days

5.3 Sensitivity of the two methods to the initial ensemble

In this section we perform experiments using different choices of the initial ensembles from the 1000 realizations at our disposal. The estimated permeabilities after 420 days using the saturated NPBBN with different initial ensembles are compared. With the same setup we run experiments using the EnKF. For our experiments, we shall consider the four locations from section 5.1.

We chose 100 ensembles from the 1000 realizations. Figure 5.29 illustrates the mean of the different 100 realizations chosen. Note that the initial mean of the ensembles is very similar for different choices of 100 realizations. The maximum absolute value difference between the mean of the initial choices noticed was 0.1.

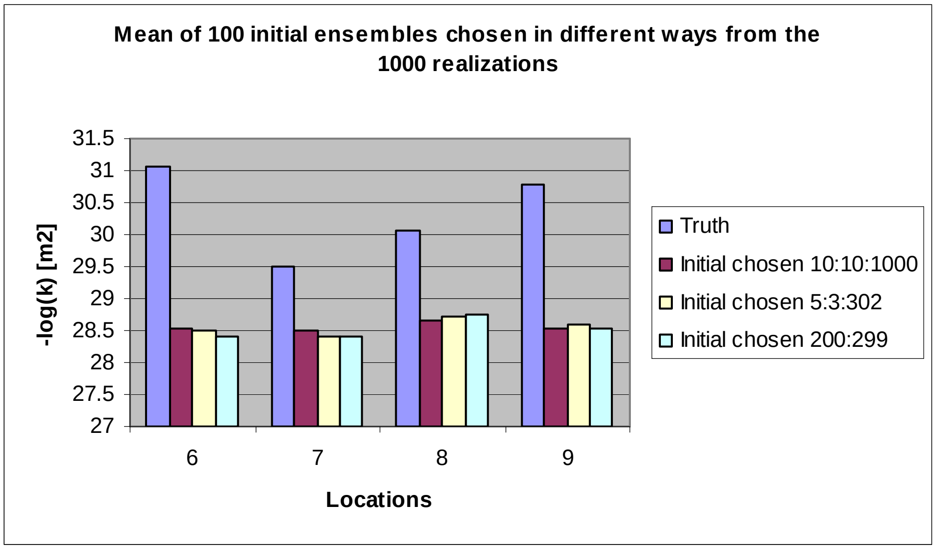


Figure 5.29: Mean of the 100 initial ensembles chosen in different ways from the 1000 realizations

Figure 5.30 illustrates the estimated permeabilities after 420 days using the EnKF. In Figure 5.31 we present the RMSE for each location using the EnKF method. For location 6 and 7 the choice of the initial does not make much difference. The best estimate for location 6 is obtained when using the initial 100 ensembles with step 3. For location 7 the best estimate is noticed when using the initial 100 chosen with step 10. Locations 8 and 9 seem to be more sensitive to the choice of the initials than locations 6 and 7. Both locations suggest that the 100 ensembles chosen with step 3 give a better estimate than the other two choices. Moreover, the worst estimate for both locations is obtained when using the 100 ensembles chosen in row.

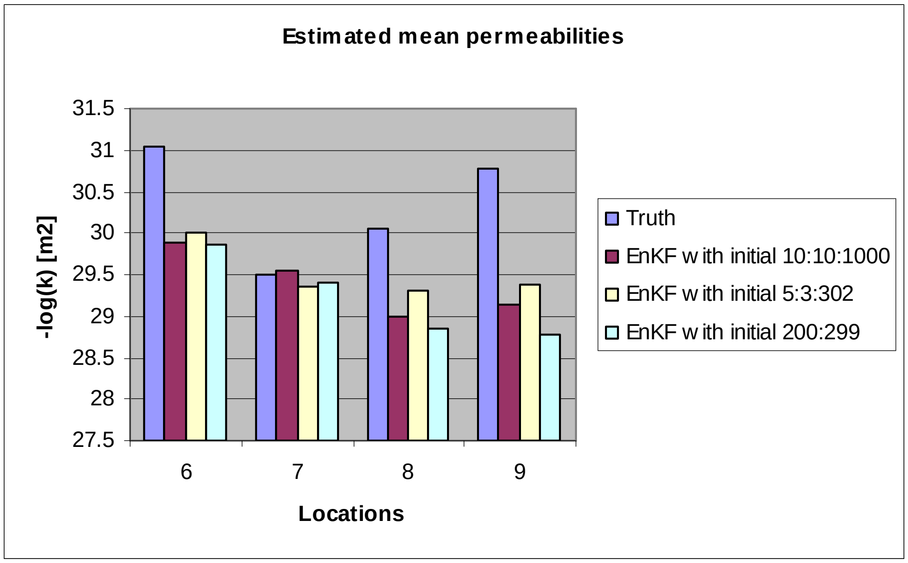


Figure 5.30: The estimated permeabilities using the EnKF with different choices of the initial ensembles from the 1000 realizations

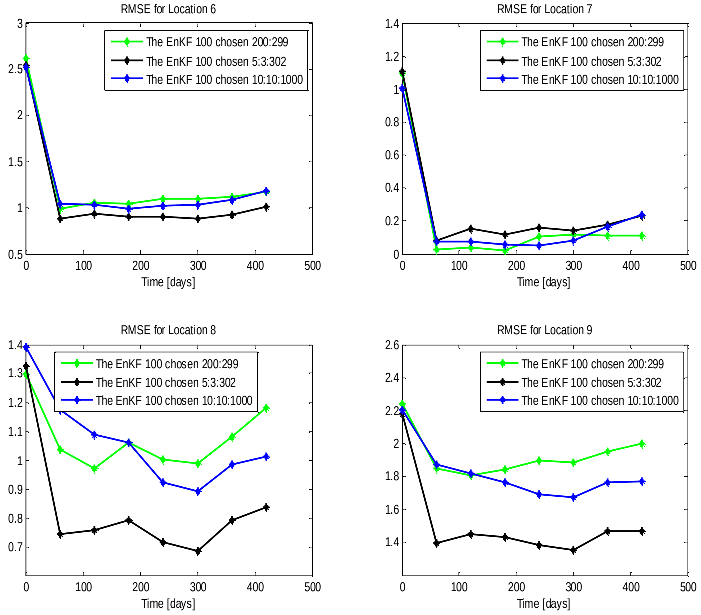


Figure 5.31: RMSE for each location. Comparative results of the estimated permeabilities using the EnKF with different choices of the 100 initial ensembles from the 1000 realizations.

The estimated permeabilities after 420 days using the saturated NPBBN are presented in Figure 5.31. Figure 5.32 illustrates the RMSE for each location with different choices of the initial. We notice that the NPBBN based approach is very sensitive to the choice of 100 the initial ensembles. The best estimates for all four locations are obtained when using 100 initial ensembles chosen with step 10. The results suggest that the more spread is in the initials, the better the estimates obtained.

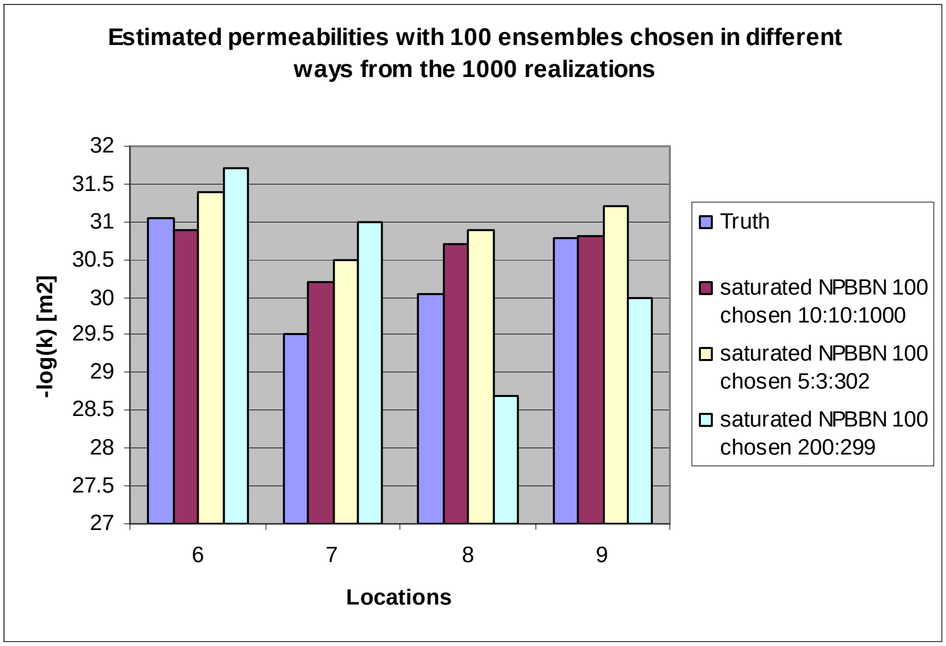


Figure 5.32: The estimated permeabilities using the saturated NPBBN with different choices of the initial ensembles from the 1000 realizations

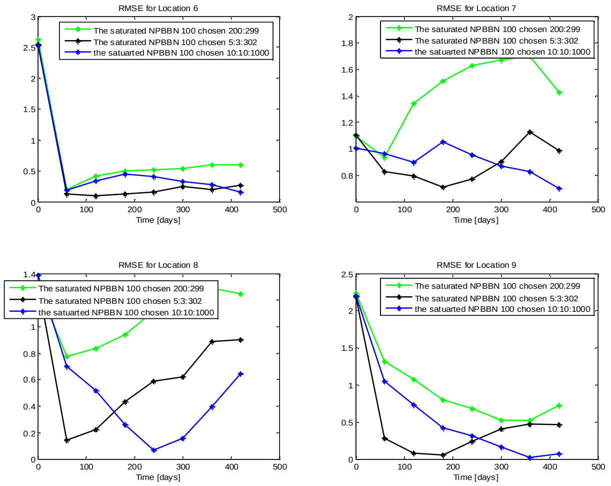


Figure 5.33: RMSE for each location. Comparative results of the estimated permeabilities using the saturated NPBBN with different choices of the 100 initial ensembles from the 1000 realizations.

The results of this section show that the choice of the initial ensemble, from the 1000 realizations, influences the estimated permeabilities when both the EnKF and the NPBBN method are used. However, the NPBBN based approach seems to be more sensitive to the choice of the initial ensemble than the EnKF. This might be because estimating the correlation matrix is more demanding than estimating the first two moments, mean and covariance matrix.

If the 1000 realizations are completely independent of each other, one would expect that the choice of the initial should not influence so much the final estimate. The obtained results might give an indication that the 1000 realizations are dependent. As a matter of fact the 1000 ensembles are generated as linear combinations of 200 snapshots of a training image. This introduces dependencies in the ensembles. We believe that the performance of both techniques, especially the NPBBN method should improve if completely independent initial realizations are to be available.

Chapter 6

Conclusions and recommendations

Reservoir simulation is an area of reservoir engineering that uses computer models to predict the flow of fluids in a reservoir. These models are used by oil companies for two purposes. The first one concerns the development of new fields, i.e. the drill of new production wells is of great interest. Models are also used in developed fields where the production forecast is used to make investment decisions. Typically, one is interested in the flow of fluids like oil, water and gas. The ease of flow of such a fluid through a porous media is characterized by the permeability of the rock. The goal of this thesis was to estimate the permeability of the subsurface. The problem of estimating parameters like permeability is often referred to as the history matching problem in reservoir simulation.

This was the first attempt to approach a history matching problem in reservoir simulation using the NPBBN method. The new approach was presented in a comparative way with the EnKF method. The idea was to use the ensemble Kalman filter with localization. However, given the setup for our experiments the localization proved not to be necessary.

The results clearly indicate that the two methods are comparable when used to solve a history matching problem in reservoir engineering. However we would like to point out to the differences between them.

Firstly, 100 ensembles are enough for estimating the permeabilities for the entire field using the EnKF. In section 5.3.1 we noticed that the difference between the estimates using 100 and 300 ensembles (for the EnKF) was very small. The situation changes when using the NPBBN method. If one looks at the stability of the correlation matrix, then 300 ensembles were necessary for estimating the permeabilities from four locations only. Note that the increase in the number of ensembles used for the NPBBN method does not interfere with the computational speed in contrast with the EnKF.

Localization proved to be necessary for the EnKF when the size of the ensembles chosen for the entire filed (21x21 grid block) was very small (e.g. less than 80 ensembles). Similar results were noticed when applying the data mining procedure for learning the structure of the NPBBN from data. The saturated NPBBN showed better results than the learned one when 300 ensembles were used for estimating the permeabilities from four locations. However, when a NPBBN was learned and 100 ensembles were used, the results were inconclusive. For some locations, the estimate using the learned NPBBN improves, for others the saturated NPBBN gave better results.

When presenting the twin experiment setup in Chapter 4 of this thesis we mentioned that the measurements are generated by adding Gaussian noise to the truth. Looking at the histograms of the variables we noticed that neither bottom hole pressure at the injector well, nor total flow rates at the producers are normally distributed. The assumption of normality is used for the EnKF method. For a fair comparation we decided to use the same setup. It could be of future interest to add errors to the observable variables with closer distributions to their true ones.

One of the most important assumptions made by the EnKF method is that the conditional joint distribution is joint normal. Therefore, it can be characterized just by the first two moments (mean and variance). When the NPBBN model was built for four locations we observed that the margins of the assumed Gaussian distribution were far from being normally distributed. Note that no validation of the assumption of joint normality distribution is used for the EnKF method. On the other hand, the NPBBN approach uses the assumption of joint normal copula. Hence, no assumption about the marginal distributions is made. In contrast with the EnKF, for the NPBBN method one can validate the assumption of joint normal copula by using a statistical test.

As mentioned the NPBBN based approach uses statistical tests for two validation steps. Firstly, the joint normal copula is validated. Secondly, a learned NPBBN model that approximates the saturated graph is validated. The overall measure of dependence on which the statistical tests are based is the determinants of the correlation matrices. When a static NPBBN model was built for 13 variables both steps were validated using the statistical tests. However, as the number of variables in the graph increased, e.g. for 103 or 342 variables involved in the NPBBN model, the values of the determinants of the correlation matrices became meaningless. Therefore, the validation was not significant anymore. Different statistical tests, which are maybe based on a different measure of dependence would be of great interest.

We presented results of estimating the permeabilities for: four different locations randomly chosen from the field at our disposal, a 7x7 grid block and a 13x13 grid. For three out of four randomly chosen locations, the obtained estimates when using the NPBBN approach were closer to the truth than the ones obtained when using the EnKF. Better results were obtained with the NPBBN than with the EnKF when estimating the permeabilities for a 7x7 grid. However, when the permeabilities were estimated for a 13x13, the two methods performed more or less the same. One could argue that the RMSE for the NPBBN method shows a worse estimate. It is worth mentioning that RMSE gives just a general idea of the behavior of the estimate. The visual representation of the fields can be a powerful tool to evaluate the performance of the applied methods in a twin experiment. In this particular case, the images of the fields show that the NPBBN approach gives similar results to ones obtained with the EnKF method.

Our goal was to estimate the permeabilities for the entire field using both methods. However, we only illustrated results for a maximum of 13x13 grid block. Building a saturated NPBBN for a larger grid becomes computationally infeasible. The problem lies in the number of arcs and the number of computations needed in order to quantify them. The maximum number of permeabilities that we could estimate was 169 out of 441. This constitutes a considerable limitation of the NPBBN based approach. Nevertheless, interpolation methods could be employed for estimating the permeabilities for larger fields.

When comparing the estimated permeabilities using both methods, the EnKF was applied to the entire field (21x21 grid block). However, if EnKF would be applied to a smaller size grid block, e.g. a 7x7 or a 13x13, one would expect to obtain a worse estimate because information is lost. The NPBBN was limited to be applied for estimating the permeabilities for a maximum of 13x13 grid block. We did not consider imposing the same limitation to the EnKF.

In section 5.3 we performed some experiments to see how the choice of the initial ensemble influences the estimated permeabilities. The obtained results gave an indication that the 1000 ensembles might be dependent. As a matter of fact the 1000 realizations were generated as linear combination of 200 snapshots of a training image. This introduces dependencies in the ensembles. One should expect that the performance of both methods will improve if, in the future, completely independent initial ensembles are to be used. Our results also suggested that the NPBBN method is more sensitive to the initial choice than the EnKF. One reason for that could be the fact that the NPBBN method is based on the relationships among variables. Therefore the method works with the correlations between variables. Estimating the correlations matrix is more demanding than estimating the mean and covariance used by the EnKF.

We shall further present a summary of the main conclusions and recommendations discussed in this chapter:

- The NPBBN based approach uses more ensembles than the EnKF method. However, this does not influence the computational speed in contrast to the EnKF.
- For being affordable, the EnKF is limited to use a small number of ensembles. Typically between 50 and 100 ensembles are used in reservoir applications. Given that the dimension of the state vector for real reservoir engineering applications can be of order 10^6, the ensemble size is clearly too small. One way of dealing with these issues is to use localization methods. However, sometimes localization introduces inconsistencies in the system.
- The NPBBN technique is limited to estimate the permeabilities for a maximum of a 13x13 grid block. We suggest employing interpolation methods for estimating permeabilities for larger grid blocks.
- The NPBBN technique is more sensitive than EnKF to the choice of the initial ensemble. Better results are to be expected for both methods, especially for the NPBBN based approach, if completely independent initial ensembles are to be used.
- The NPBBN method works with the assumption of joint normal copula. No assumption of the marginal distribution is made. Moreover, the method validates the assumption by using statistical tests. When the number of the variables involved in the NPBBN were greater than 100, the values of the statistical tests become meaningless. Hence, new

statistical tests for the NPBBN based approach are of great interest.

- The EnKF uses the assumption that the joint distribution is normal. No validation of this assumption is involved when using the EnKF. Moreover the histograms of the variables point out the fact that the marginal distributions of the assumed Gaussian are far from being normal.
- Measurements are generated by adding Gaussian noise to the true observations. Histograms of the observable variables indicate that the measurements are not normally distributed. Measurements generated by adding noise to the observable variables with closer distributions to their true one are of future interest.

If one would have to make a choice it might be too early to say which approach performs better. We believe that further research considering all of the above mentioned recommendations should give more conclusive results when comparing the two approaches.

Appendix

A.1 Introduction

The purpose of this appendix is to present the two most common localization methods in EnKF systems: Covariance Localization (CL) and Local Analysis (LA). Both of these methods are popular for large scale applications with the EnKF [e.g.5,6,12,8,7,9].

The appendix is organized as follows: in Section A2 and Section A3 Kalman Filter and Ensemble Kalman Filter are are placed in the context of data assimilation methods. Section A4 presents the distance based localization methods. A small case study of a petroleum engineering application is presented in Section A5.

A.2 Kalman Filter

Data assimilation is the methodology that combines a mathematical-physical model with available measurements for estimating and predicting different environmental processes.

One of the most known data assimilation methods is the Kalman Filter, published by R.E.Kalman in 1960. The author is describing a recursive solution to the discrete data linear filtering problem. Since that time, mostly due to the computer advances, the Kalman Filter has been the subject of extensive study and research. A large number of new algorithms were developed for solving different type of applications. The Kalman Filter is a set of mathematical equations that are used to estimate the state of a process in a way that minimizes the distance between measurements and model predictions.

The Kalman Filter addresses the general problem of trying to estimate the state $x \in \mathbb{R}^n$ of a process that is governed by the liner stochastic difference equation

$$X(k+1) = A(k)X(k) + B(k)u(k) + G(k)W(k) \tag{1}$$

With measurements

$$Z(k) = H(k)X(k) + V(k) \tag{2}$$

k represents the time; $A(k)$ is a $n \times n$ matrix that relates the state at the previous time step $k-1$ to the current step k; the matrix B relates the input vector u_k to the state vector; the matrix H in eq. (2) relates the state to the measurements $Z(k)$.

The random variables $W(k)$ and $V(k)$ represent the model and measurement noises. They are assumed to be independent of each other and white Gaussian.

$$W(k) \sim N(0, Q(k))$$

$$V(k) \sim N(0, R(k))$$

To solve the filtering problem we have to determine the probability density of the state $X(k)$ conditioned on the history of available measurements $Z(k)$. This conditional density function is assumed to be Gaussian; hence, it can be fully characterized by the mean and covariance matrix. The Kalman Filter will recursively calculate the state vector $X(k)$ along with its covariance matrix P, conditioned on the available measurements up to some time k, under the criterion that the estimated error covariance is minimum. The main steps of the algorithm are described in the following Figure:

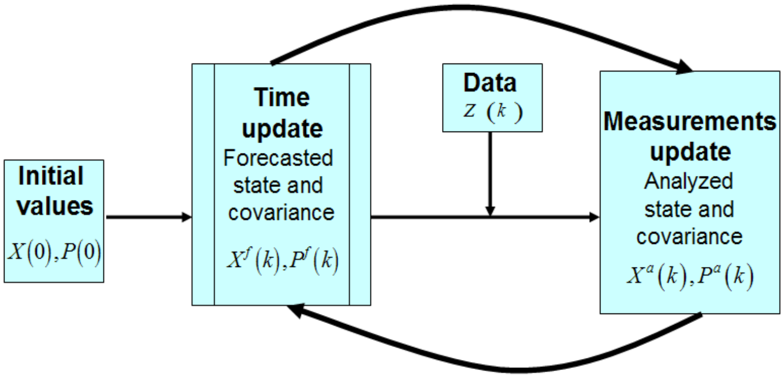


Figure 1: Kalman Filter algorithm

The main steps of the algorithm can be summarized as follows:

Initialization

$$X^a(0) = X(0) \tag{5}$$

$$P^a(0) = P(0) \tag{6}$$

Time update

$$X^f(k+1) = A(k)X^a(k) + B(k)u(k) + G(k)W(k) \tag{7}$$

$$P^f(k+1) = A(k)P^a(k)A(k)^T + G(k)Q(k)G(k)^T, \tag{8}$$

where the subscript *f* stands for forecasted value and the subscript *a* stands for analyzed.

Measurement updates

$$X^a(k+1) = X^f(k+1) + K(k+1)\big(Z(k+1) + H(k+1)X^f(k+1)\big) \tag{9}$$

$$
\begin{aligned}
P^{a}\left(k+1\right) &= P^{f}\left(k+1\right)-K\left(k+1\right)H\left(k+1\right)P^{f}\left(k+1\right) \\
&= \left(I-K\left(k+1\right)H\left(k+1\right)\right)P^{f}\left(k+1\right)
\end{aligned}
\qquad (10)
$$

$P^{f}\left(k+1\right)$ is also called the background error covariance matrix. $K\left(k+1\right)$ is the Kalman Gain or weighting matrix, describing the difference between the observed and the forecasted variables. It is computed using the following equation:

$$
K(k+1)=P^{f}\left(k+1\right)H\left(k+1\right)^{T}[H\left(k+1\right)P^{f}\left(k+1\right)H\left(k+1\right)^{T}+R\left(k+1\right)]^{-1} \qquad (11)
$$

A.3 Ensemble Kalman Filter (EnKF)

The Kalman Filer is computationally expensive for large scale systems and is not suitable for non linear systems. The most time consuming operation of the filter algorithm for large scale systems is calculating the first term in the time update of the covariance matrix:

$$
A\left(k+1\right)P^{a}(k+1)A\left(k+1\right)^{T} \qquad (12)
$$

For large scale systems it is not attractive to store the matrix $A\left(k\right)$ and to determine its transpose. In order to obtain a computationally feasible filter, a square root algorithm can be used. This algorithm is based on the fact that a covariance matrix P can be factorized as follows:

$$
P=LL^{T} \qquad (13)
$$

Where L is a square root of P. The factorization is not unique and there exist many square roots of P. Most of the algorithms that have been developed for large scale systems are based on a square root representation of the covariance matrix P [e.g. 6, 12]. An example of such an algorithm is Ensemble Kalman Filter (EnKF) for which the probability density of the state estimate is represented by a finite number M of randomly generated system states:

$$\xi_i(k+1), i=1,\ldots,M \tag{14}$$

The optimal estimate and the square root of the covariance matrix of the estimation error are now given by:

$$X(k+1)=\frac{1}{M}\sum_{i=1}^{M}\xi_i(k+1) \tag{15}$$

$$L(k+1)=\left[\left(\xi_1(k+1)-X(k+1)\right) \quad \ldots \quad \left(\xi_M(k+1)-X(k+1)\right)\right]^T \tag{16}$$

The square root $L(k+1)$ defines an approximation of the covariance matrix $P(k)$ from the finite number of ensembles:

$$P_{aprox}(k+1)=\frac{1}{M-1}L(k+1)L(k+1)^T \tag{17}$$

Matrix $L(k)$ is also called the ensemble anomalies matrix. The main steps of the Ensemble Kalman Filter are described in Figure 2.

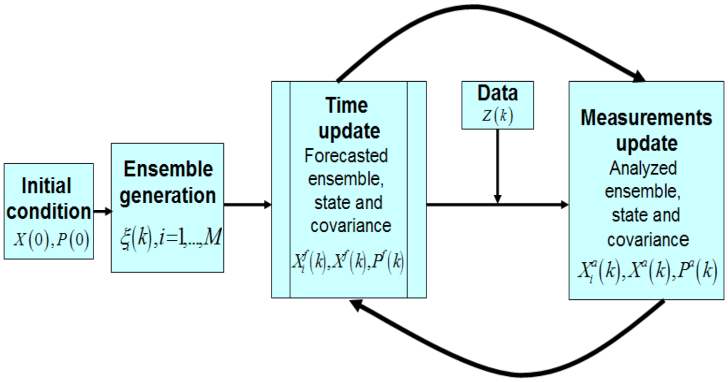


Figure 2: Ensemble Kalman Filter algorithm

The Ensemble Kalman Filter algorithm's steps from Figure 2 can be summarized as follows:

Time update

$$\xi_i^f\left(k+1\right)= f\left(\xi_i^f\left(k\right),k\right)+G\left(k\right)W_k^i \tag{18}$$

$$X^f\left(k+1\right)=\frac{1}{M}\sum_{i=1}^{M}\xi_i^f\left(k+1\right) \tag{19}$$

$$L^f\left(k+1\right)=\left[\left(\xi_1^f\left(k+1\right)-X^f\left(k+1\right)\right) \quad \ldots \quad \left(\xi_M^f\left(k+1\right)-X^f\left(k+1\right)\right)\right]^T \tag{20}$$

$$P_{aprox}^f\left(k+1\right)=\frac{1}{M-1}L^f\left(k+1\right)L^f\left(k+1\right)^T \tag{21}$$

Measurements update

$$K\left(k+1\right)=P_{aprox}^f\left(t+1\right)H\left(k+1\right)^T\left[H\left(k+1\right)P_{aprox}^f\left(t+1\right)H\left(k+1\right)^T+R\left(k+1\right)\right]^{-1} \tag{22}$$

$$\xi_i^a\left(k+1\right)=\xi_i^f\left(k+1\right)+K\left(k+1\right)\left(Z\left(k+1\right)+V_i\left(k+1\right)-H\left(k+1\right)\xi_i^f\left(k+1\right)\right) \tag{23}$$

$$X^a\left(k+1\right)=\frac{1}{M}\sum_{i=1}^{M}\xi_i^a\left(k+1\right) \tag{24}$$

$$L^a\left(k+1\right)=\left[\left(\xi_1^a\left(k+1\right)-X^a\left(k+1\right)\right) \quad \ldots \quad \left(\xi_M^a\left(k+1\right)-X^a\left(k+1\right)\right)\right]^T \tag{25}$$

$$P_{aprox}^a\left(k+1\right)=\frac{1}{M-1}L^a\left(k+1\right)L^a\left(k+1\right)^T \tag{26}$$

A.4 Localization

The main advantage of the EnKF is that it approximates the state error covariance matrix, thus it becomes suitable for large non linear problems. Generally, the EnKF is used for applications that have state vectors of very large dimensions. The main disadvantage of EnKf appears when the dimension of state vector is much larger than the number of ensembles used. This leads to sampling errors which makes the approximated covariance error to be underestimated. As shown, in EnKF the covariance error equation (8) is approximated by Eq. (21). When the ensemble size is much smaller compared to the dimension of state vector, then this approximated covariance error is inaccurate. Unfortunately this is very often the case, since EnKF is a Monte Carlo Method and for being affordable for large systems a relatively small number of ensembles are used (generally 100 ensembles, but sometimes even less). The presence of unreal correlations, also called spurious correlations between large distance grid points is noticed in this case. In this way the model and state variables are updated in regions where they should not be updated, thus observations have high influence over too large distances, which is not physically true.

One way of reducing these effects is to use localization methods. Errors can be correlated locally, but as mentioned, sometimes covariance shows correlations between large distance grid points. The process of eliminating these unreal, large distance correlations is called localization. The two classical localization methods are: Covariance Localization (CL) and Local Analysis (LA). We shall briefly describe both of them in the following sections.

A.4.1 Covariance Localization (CL)

Covariance localization uses a Schur product of the covariance of the background error calculated from the ensembles and a correlation function with local support, denoted here by *Corr*. This method is described in [10], [28].

$$P_{aprox}^{f} \rightarrow Corr \circ P_{aprox}^{f} \tag{27}$$

By the Schur product theorem ([29]) the product function is also a covariance function. The Schur, often called the Hadamard product of two matrices *A* and *B* is the matrix C having the same dimensions as *A* and *B*, $C_{i,j} = A_{i,j} B_{i,j}$. The Schur Product is applied to the Kalman Gain as follows:

$$K^{loc} = \left(Corr \circ P^f_{aprox} \right) H^T \left[H \left(Corr \circ P^f_{aprox} \right) H^T + R \right]^{-1} \tag{28}$$

We denote by K^{loc} the Kalman Gain for which we have applied the Schur product and $Corr \circ P^f_{aprox}$ denotes the Schur product of the correlation matrix $Corr$ with the covariance matrix P^f_{aprox} . The order of the forward interpolation and the Schur product can be changed without altering the results, so that (28) can be written as follows:

$$K^{loc} = Corr \circ \left(P^f_{aprox} H^T \right) \left[HCorr \circ \left(P^f_{aprox} H^T \right) + R \right]^{-1} \tag{29}$$

$P^f_{aprox} H^T$ can be calculated directly from the ensembles:

$$P^f_{aprox} H^T = \frac{1}{M-1} \sum_{i=1}^{M} \left(X_i^f - X^f \right) \left(HX_i^f - HX^f \right)^T \tag{30}$$

Now, the question is “How to choose the correlation function also called sometimes a tapper function?” In [29] a number of examples of correlation functions with local support are given. Widely used is the fifth order polynomial with similar shape of a Gaussian function, proposed by Gaspari and Cohn, 1999, eq 4.10. The correlation functions are generally smooth and monotonically decreasing with distance. These functions reduce and smooth the effect of observations at large distances. The effect varies from 1 at the observation point to 0 at some specified distance from the grid point being analyzed. . The two safest choices are Gaussian and Gaspari and Cohn functions. To be applicable to a particular scheme, CL as defined in eq. (27) requires that the update equations are formulated in terms of *P*. So, covariance localization can be applied only to schemes that use *P*.

A.4.2 Local Analysis (LA)

Local analysis is defined within a local region which means that only observations within a certain distance from the grid point being analyzed will impact the analysis in that grid point. This localization method is applied to the estimate update. Localization is done grid point by grid point so every time a small model state is solved. Observations within the local region can have equal or decreased influence of the observation on the analysis point regarding the local functions used. In [9] was proposed to use a cut off radius for which only observations within this defined radius are used for the grid point analyzed. This was one of the first attempts of doing localization in EnKF. It was shown that the use of a cut off radius produces discontinuity analysis increments, so that it becomes a source of noise itself. Another example presented in [9] suggested using a smoother function, a cylinder, such that only observations located within a given radius would be used for each analyzed grid point. Generally, we can use a function like normal or exponential 3 [1] that decreases locally from 1 to 0 at some defined radius or distance. The same kind of functions applied globally for covariance localization can be used locally for local analysis. The main difference between the two approaches is that for covariance localization we apply the function to the background error covariance matrix, while for local analysis we apply the function to the estimate update.[9]

[9] $\exp(-1/2*(dis\tan ce/radius)^3)$

The choice of observations to be used, in other words the choice of the radius for each grid point is up to the user of the method, a good choice must consider the particular system being modeled and the size of the ensembles (more ensembles should allow more distant observations to be used). It is also important that most of the measurements used in an analysis of a certain grid point to be used for the neighboring grid points in order not to have a drastic change of the values of very close grid points. Generally, the best choice of the radius is determined experimentally for the particular system modeled.

A.5. Case Study

We shall apply localization to the problem described in chapter 5 of this project. All experiments setup are according to Chapter 5.

A. 5.1 The choice of the localization method

The first choice of what localization method to implement for our field was covariance localization (CL), the choice was mainly based on the fact that covariance localization is less computationally demanding than local analysis. In order to apply covariance localization we need to multiply the correlation function with the background error covariance matrix as described in Section AA4.1. Due to different units and scales for measuring the variables of the state vector pressure and saturation (pressure is of order 10^7 and saturation is of order 10^{-3}) the matrix-matrix and matrix-vector multiplication from Kalman Gain became an issue. In [28], pages 48, 49 a trick is presented, which consists in the use of some scaling matrices such that the multiplications matrix-matrix and matrix-vector result in matrices which elements do not sufficiently differ in order. Unfortunately, the multiplication with the scaling matrices separates matrices *L* and *L'* such that it is impossible to apply the correlation function to the background covariance matrix P^f . Then covariance localization becomes impossible to be applied to the available EnKF for the described field.

We decided to apply local analysis. In this way localization is applied locally to the estimate update (measurements update) and not to the Kalman Gain matrices. Hence we avoid complications related with re-scaling the state vector. Next Section AAwill present experiments using local analysis for different number of ensemble members.

A.5.2 Experiments

Because localization is related to the number of ensembles used, firstly we chose to perform experiments with a very small number of ensembles, thus we run EnKF with 15 ensembles with different localization functions. The normal function is chosen as our local function, this function decreases from 1 at the observation point to 0 at some specified radius. Figure 3 shows the RMSE for EnKF different radius for simulations run with 15 ensembles. The same exact ensembles are used for all simulations.

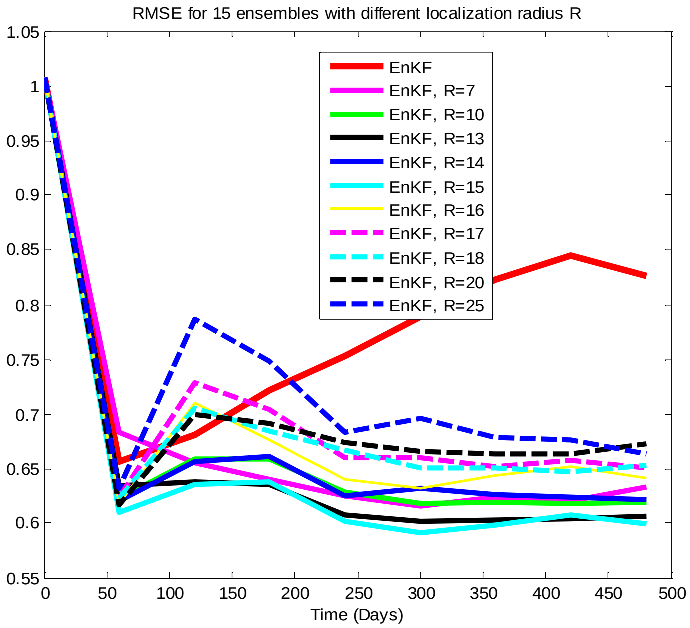
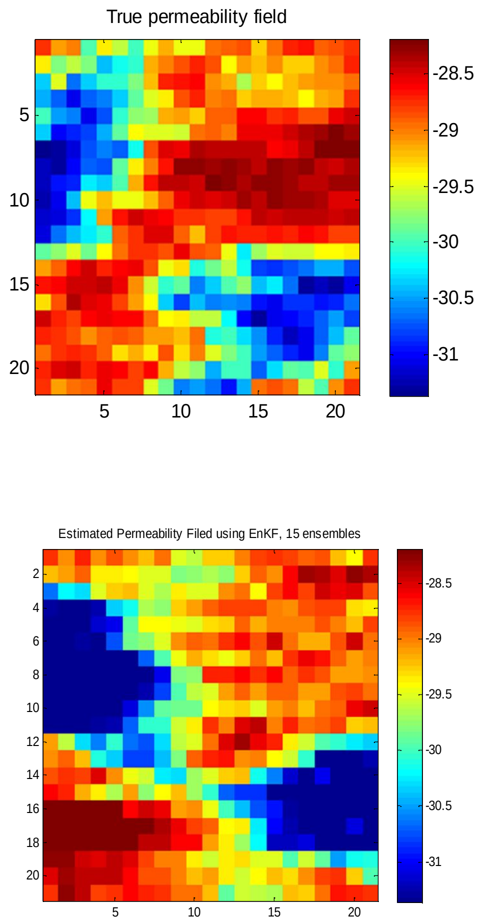


Figure 3: RMSE for 15 ensembles with different localization radius R

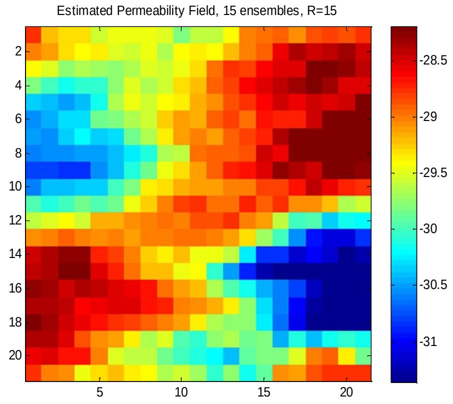
From Figure 3 we notice that indeed by applying local analysis RMSE decreases, the smallest RMSE is observed for the local radius at the value 15. Also, because we use such a small number of ensembles, we observe that at the beginning of time RMSE tends to increase. After approximately 200-250 days RMSE decreases when we use localization but the increasing trajectory is kept for the use of EnKF only. Figure 4 shows the estimated permeability field using EnKF. Figures 7 describe the estimated permeability field for EnKF using the local radius 15 as suggested in Figure 3. We notice that by using localization we obtain a closer estimate to the true value of the Permeability filed. Thus, for a small number of ensembles, as 15, the local analysis improves significantly the estimate.



True permeability field

Estimated Permeability Filed using EnKF, 15 ensembles

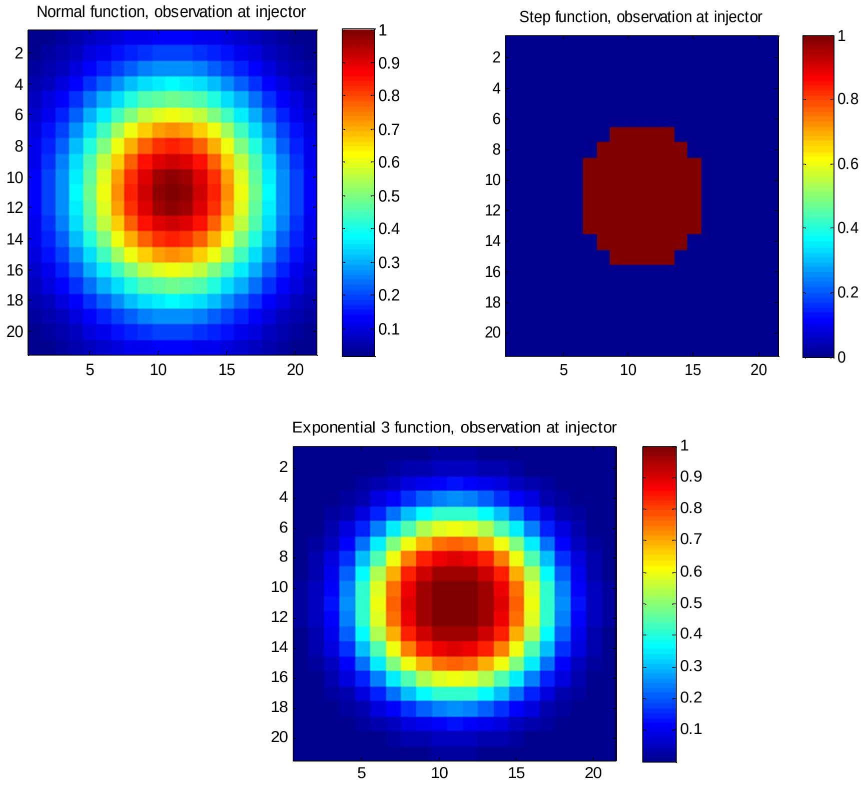
Figure 4: True Permeability field (top); Estimated permeability Field using EnKF (bottom)



Estimated Permeability Field, 15 ensembles, R=15

Figure 5: Estimated Permeability Field using EnKF with the local radius, R=15

For the previous experiments we used a normal function. The question is whether another function could perform better. In order to answer this question we perform simulations for the Step function and Exponential 3 functions also. Figure 6 presents the three comparative functions with radius 5 for the observation at the injector well.


Normal function, observation at injector

Step function, observation at injector

Exponential 3 function, observation at injector

Figure 6. Local functions, top left-Normal function; Top-right-Step function; Bottom-Exponential 3 function

Figure 7 describes the RMSE for the local radius 15 for each of the local functions presented in Figure 6. We notice that the best performance for this radius is for the normal function. The worse performance, as expected is registered for the step function, since this is the simplest case where only values 1 and 0 are used. Exponential 3 has a better behavior than the Step function, but still the Normal function performs better. Hence we conclude that is better to use a decreasing function from 1 at the observation point to 0 at some specified distance from the observation. We decide to continue our experiments using the Normal function.

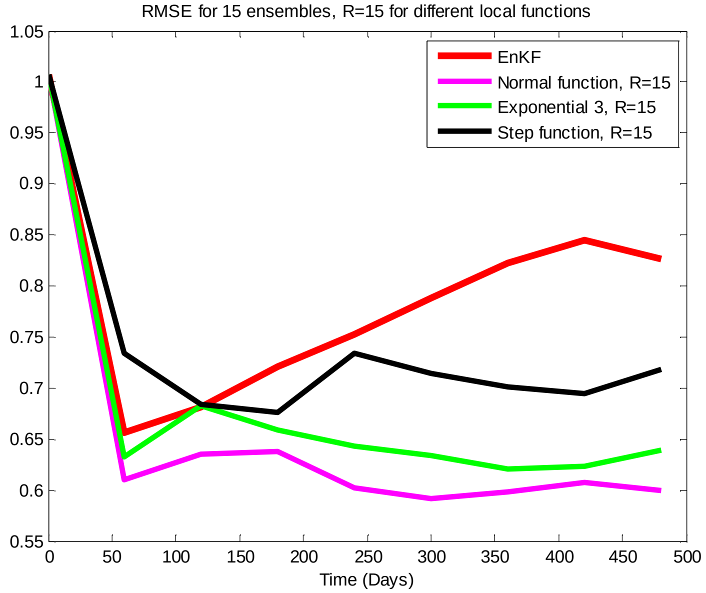


Figure 7: RMSE for EnKF, R=15, with different local functions

As the number of ensembles is increased there is less need of localization and the local radius increases as well, thus experiments with 60 and 80 ensembles should reveal a larger local radius. Figure 8 describes the RMSE for 60 ensembles with different local radiuses. As expected, the radius increases and also we notice that there aren't big differences within different radiuses. The performance of EnKF without localization is very similar with the behavior of EnKF with localization, no matter the radius used. Some improvements could be considered for R=18. Figures 11, 12 and 13 show the estimated permeability fields for EnKF with 60 ensembles, EnKF with 60 ensembles and radius 18 and EnKF with 60 ensembles with the largest possible radius 30, so that every element of the grid point is updated for each observation. We notice small differences between the three pictures, but still localization improves slightly the estimate. We observe that even for the maximum radius we still obtain a better estimate in comparison with using EnKF without localization, this is because although we update all grid points for each observation we give decreased weights from the observation point to the rest of the grid points, thus the estimate is more accurate.

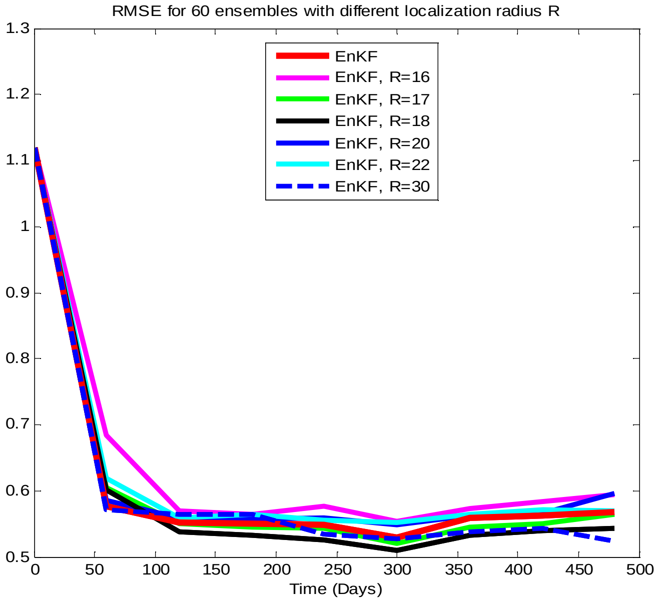


Figure 8: RMSE for different local radiuses using 60 ensembles

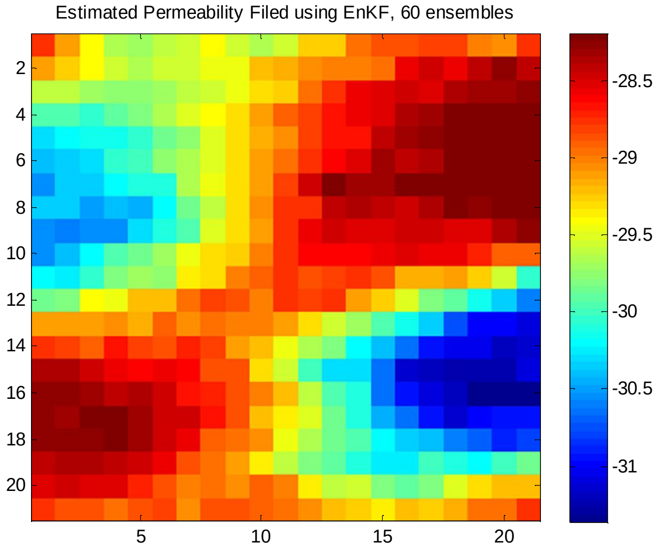


Figure 9: Estimated permeability using EnKF with 60 ensembles

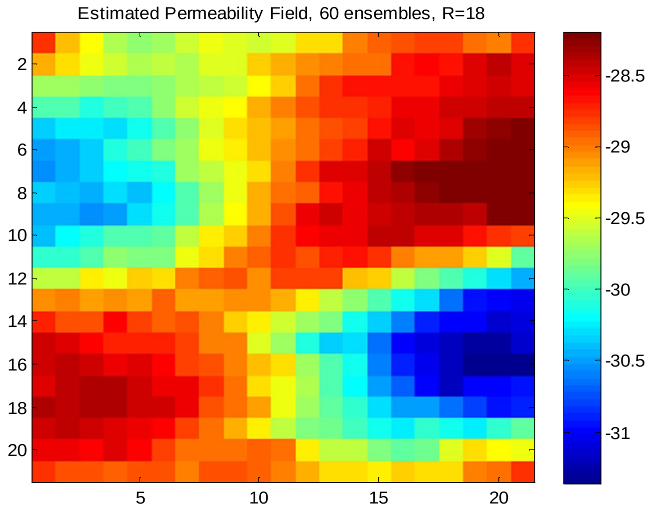


Figure 10: Estimated permeability using EnKF with local analysis, R=18, with 60 ensembles

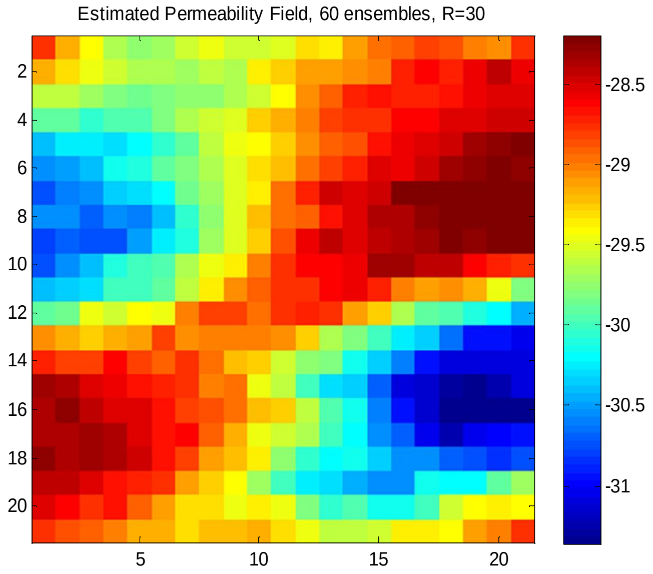


Figure 11: Estimated permeability using EnKF with local analysis, R=30, with 60 ensembles

The last round of experiments performed for this study is by considering 80 ensembles. As mentioned, we expect that there is less need of localization as the size of the ensembles increases. Figure 12 shows that the values of RMSE are vey close for very different local radiuses. There is almost no difference between EnKF and EnKF with localization. Still, EnKF with R=25 seems to produce a slightly smaller RMSE. Figures 15, 16 and 17 present EnKF, EnKF with R=15 and EnKF with R=30. Although localization does not change in a very drastic way the estimate for this number of ensembles, still there are some improvements. Considering the dimension of our field and the fact that the maximum distance between observation and a grid point is approximately 900m, local analysis shows some improvements for the estimate.

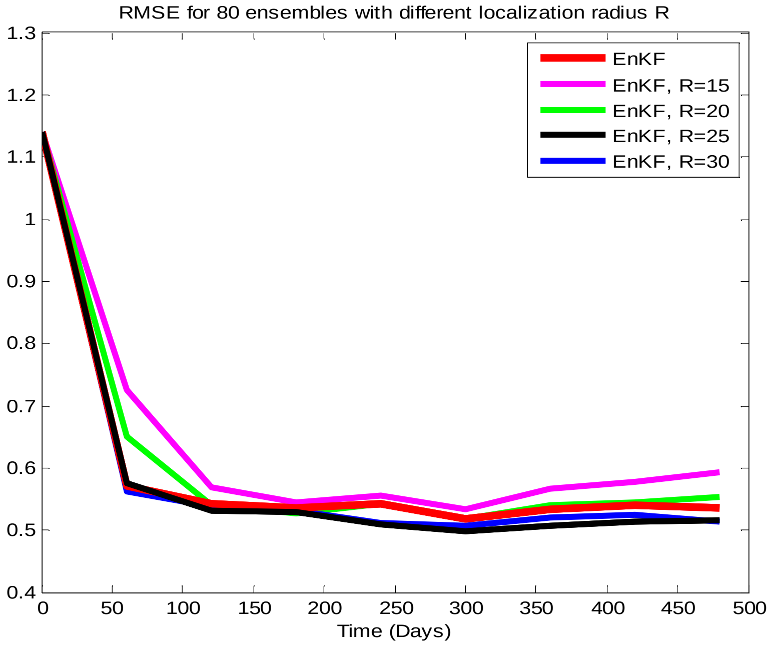


Figure 12: RMSE for EnKF with different localization radius using 80 ensembles

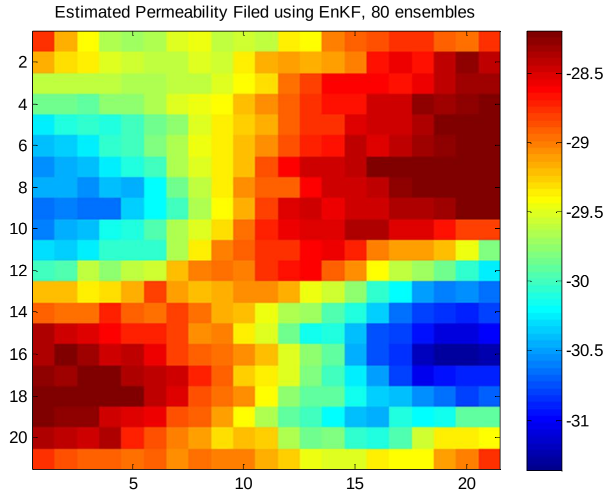


Figure 13: Estimated permeability using EnKF with 80 ensembles

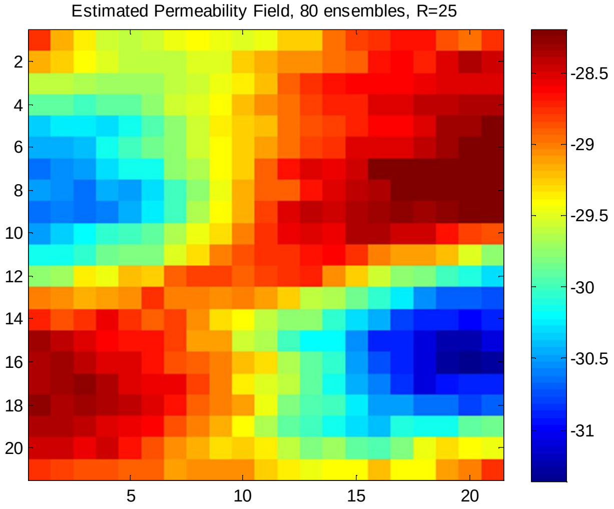


Figure 14: Estimated permeability using EnKF with local analysis, R=25, with 80 ensembles

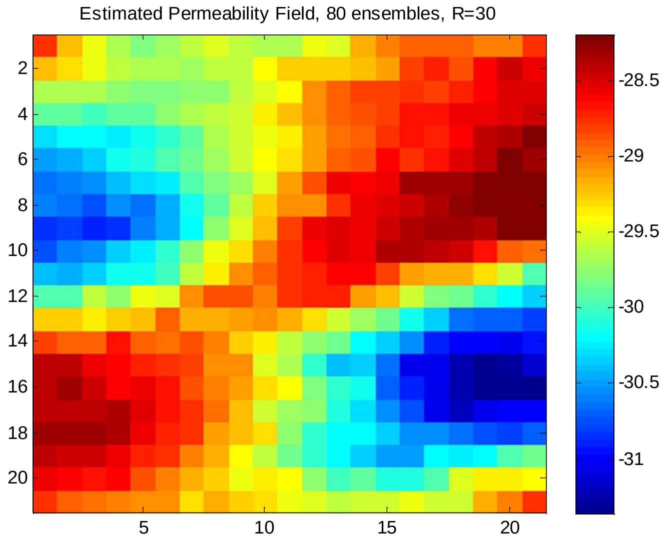


Figure 17: Estimated permeability using EnKF with local analysis, R=30, with 80 ensembles

A.6. Conclusions

Our goal was to describe and apply the two most common localization methods within the Ensemble Kalman filter techniques. Sections A2 and A3 present briefly the Kalman Filter and Ensemble Kalman Filter algorithms. The main advantage of the EnKF is that is suitable for large nonlinear systems. Still, because in general is used for large applications, the dimension of the state vector is much bigger than the size of the ensemble used. This introduces large correlations between distant grid points, also called spurious correlations, which lead to inaccuracy in our covariance error. Localization techniques deal with diminishing these spurious correlations and improving the accuracy of the estimate. The two most common localization methods are introduced in this study. Covariance localization (CL) is applied globally, to the Kalman Gain by multiplying the background error with some correlation function. The second method, local analysis is applied locally to the estimate update, every time only measurements situated within a certain radius from the grid point analyzed contribute to the estimate. Section A4 presents the two localization methods.

Section A5 presents a small application for which we use local analysis. The reasons for which covariance localization could not be applied for our field are also described. Different functions with local support are used and the choice between them is based on their performance on our example. Although the dimension of the state vector is rather small, the largest distance between the observation and a grid point is around 900 m, we could still observe improvements by applying local analysis, especially for a very small number of ensembles, like 15. As the number of ensembles increases the need of localization decreases and there are less and less improvements noticed, although even for 80 ensembles, using the maximum localization radius we can still notice some positive effects.

As shown in our study, if the ensemble size is small we must use localization methods. We presented the classical distance based methods, but many interesting alternatives exist in the literature, e.g. the so called "Adaptive localization Methods" [10], [11]. Alternatives have their own set of strengths and weaknesses, thus there is not one obvious superior method proffered by the community. The choice of the localization method should be based on criteria likes system's characteristics, numerical effectiveness and computational properties.

Bibliography

[1] Jansen, J.D.: System theory for reservoir management. Version 5e, 2009, lecture notes.

[2] Z. Chen, G. Huan, and Y. Ma. Computational Methods for Multiphase Flows in Porous Media. Society for Industrial and Applied Mathematics, Philadelphia, 2006.

[3] T. Ertekin, J. H. Abou-Kassen, and G. R. King. Basic Applied Reservoir Simulation. Society of Petroleum Engineers, Richardson, 2001.

[4] A.T. Watson, J.H.Seinfield, G.R. Gavalas, P.T.Woo. History matching in two-phase petroleum reservoirs, 1980.

[5] R. E Kalman, A new Approach to Linear Filtering and Prediction Problems, 1960.

[6] Geir Evenson. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, 1994.

[7] P.L. Houtekamer and Herschel L.Metchell. Data assimilation using Ensemble Kalman Filter Technique, 1997.

[8] Thomas M. Hamil, Jeffery S Whitaker, Chris Snyder. Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter, 2001.

[9] Brian R Hunt, Eric J. Kostelich, Istvan Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter, 2006.

[10] Jeffrey L. Anderson. Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter, 2006.

[11] Craig H. Bishop and Daniel Hodyss. Flow adaptive moderation of spurious ensemble correlations and its use in ensemble based data assimilation, 2007.

[12] Houtekamer Pl, Mitchell HL. A sequential ensemble Kalman filter for atmospheric data assimilation, 2001.

[13] Naevdal G., Johnsen L.M., Aanonsen S.I. and Vefring E.H. Reservoir monitoring and continuous model updating using ensemble Kalman filter, 2003.

[14] Aanonsen S.I., Naedval G., Oliver D.S., Reynolds A.C. and Valles B. The Ensemble kalman Filter in Reservoir Engineering, 2009.

[15] Evenson G. Data assimilation. The Ensemble Kalman Filter, 2007.

[16] Anca Hanea. Algorithms for Non Parametric Bayesian Belief Nets; 2008.

[17] Dorota Kurowicka and Roger Cooke. Uncertainty Analysis with High Dimensional Dependence Modelling.

[18] O.Morales, D.Kurowicka, A.Roelen. Eliciting conditional and unconditional rank correlations from conditional probabilities, 2007.

[19] V.Mihajlovic, M.Petkovic. Dynamic Bayesian Networks: A state of the Art.

[20] Kevin Patrick Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning, 2002.

[21] Kevin E. Korb, Ann E. Nicholson. Bayesian artificial intelligence, Volume 1, 2004.

[22] Kurowicka, D. and R. Cooke. Distribution - Free Continuous Bayesian belief nets. Preceedings Mathematical Methods in reliability Conference, 2004.

[23] Whittaker, J. Graphical Models in applied multivariate statistics, 1990.

[24] Jan Dirk Jansen. The use of principal analysis (PCA) to create constrained geological realizations from a training image. TU Delft.

[25] Jeroen C. Vink. Computing Challenges in Oil and Gass Field Simulation, 2007.

[26] A.C. Reynolds, N.Chu.L. He and D.S. Olivier. Reparametrization techniques for Generating Reservoir Descriptions Conditioned to Variograms and Well-Test Pressure Data, 2006.

[27] R Li, A.C. Reynolds and D.S. Olivier. History matching of three phase flow production data, 2003.

[28] Mariya Victorova Krymskaya, MS Thesis. Parameter Estimation in reservoir Engineering Models via Data Assimilation Techniques, 2007.

[29] Gregory Gaspari, Stephen E. Cohn. Construction of correlation functions in the two and three dimensions, 1999.

[30] Heemink, A.W. Chapter 5, Data Assimilation Methods, lecture notes.

[31] Greg Welch and gary Bishop. An Introduction to the Kalman Filter, 2006.