Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 217 (2006) 143–158

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Uncertainty quantification for porous media flows

Mike Christie *, Vasily Demyanov, Demet Erbas

Institute of Petroleum Engineering, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, Scotland, UK

Received 12 August 2005; received in revised form 22 December 2005; accepted 19 January 2006
Available online 28 February 2006

Abstract

Uncertainty quantification is an increasingly important aspect of many areas of computational science, where the challenge is to make reliable predictions about the performance of complex physical systems in the absence of complete or reliable data. Predicting flows of oil and water through oil reservoirs is an example of a complex system where accuracy in prediction is needed primarily for financial reasons. Simulation of fluid flow in oil reservoirs is usually carried out using large commercially written finite difference simulators solving conservation equations describing the multi-phase flow through the porous reservoir rocks. This paper examines a Bayesian Framework for uncertainty quantification in porous media flows that uses a stochastic sampling algorithm to generate models that match observed data. Machine learning algorithms are used to speed up the identification of regions in parameter space where good matches to observed data can be found.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Uncertainty; Stochastic sampling; Genetic algorithm; Artificial neural networks; Petroleum

1. Introduction

Uncertainty quantification is an increasingly important aspect of many areas of computational science. Weather forecasting [18], global climate modelling, complex engineering designs such as aircraft systems, all have needs to make reliable prediction and these predictions frequently depend on features that are hard to model at the required level of detail [6].

Porous media flows, and specifically prediction of uncertainty in production of oil from oil reservoirs, is another area where accurate quantification of uncertainties in predictions is important because of the large financial investments made. In the oil industry, predictions of fluid flow through oil reservoirs are difficult to make with confidence because, although the fluid properties can be determined with reasonable accuracy, the fluid flow is controlled by the unknown rock permeability and porosity. The rock properties can be measured by taking samples at wells, but this represents only a tiny fraction of the total reservoir volume, leading to significant uncertainties in fluid flow predictions.

[*] Corresponding author.
E-mail address: mike.christie@pet.hw.ac.uk (M. Christie).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.01.026

Predicting fluid flows through oil reservoirs is a challenging problem. The oil tends to be a complex mixture of pure hydrocarbon components, with experimentally determined PVT properties. In general, the oil is displaced towards producing wells by a cheaper fluid such as water or gas, although there are other more specialized ways of producing oil [1]. If gas is injected to displace the oil, there can be complex transfer of hydrocarbon components between the oil and gas phases which significantly affects the displacement, and the displacement is likely to be hydrodynamically unstable.

The major source of uncertainty for all displacements in porous media is lack of knowledge of the material properties. The fluids flow through a porous matrix whose porosity (ability to store fluid) and permeability (resistance to flow) are unknown. Both porosity and permeability vary across the reservoir, with variations determined by the geological processes that deposited the reservoir and subsequent processes such as deposition and cementation of the rock grains. The most direct way to measure porosity and permeability is to take a core sample while drilling a well – usually around a 3 in. length of rock whose properties are measured in the laboratory. This means that the sampling for these fundamental properties that can vary dramatically across a reservoir is very limited. It is also possible to use indirect methods such as logging which sample a greater, although still extremely limited volume.

Fig. 1 shows an outcrop of reservoir rock illustrating some of the complexities involved in predicting fluid flows. The horizontal distance across the outcrop is around 20 m – significantly smaller than typical interwell distances of several hundred meters to kilometers. If we had drilled two wells, one at either end of the outcrop, we would have to infer the details of the permeable layers visible in the centre of the picture indirectly, yet they would clearly influence the fluid flow.

Predictions of reservoir scale fluid flows are frequently made with commercial finite difference codes. The input to these codes is the fluid properties as measured in the laboratory, and the rock properties which have to be inferred. The code can then be run with a given set of input data and the predictions are compared with observed pressures and flow rates. Clearly, this is an inverse problem and the solution is non-unique.

2. Bayesian framework for UQ

A Bayesian framework is a statistically consistent way of updating our beliefs about a given set of models given observed data. A schematic diagram showing the framework we use is shown in Fig. 2.

We start with a set of beliefs about the details of the reservoir description. This is likely to come from a geological description of the way the reservoir was deposited. For example, a fluvial reservoir laid down by

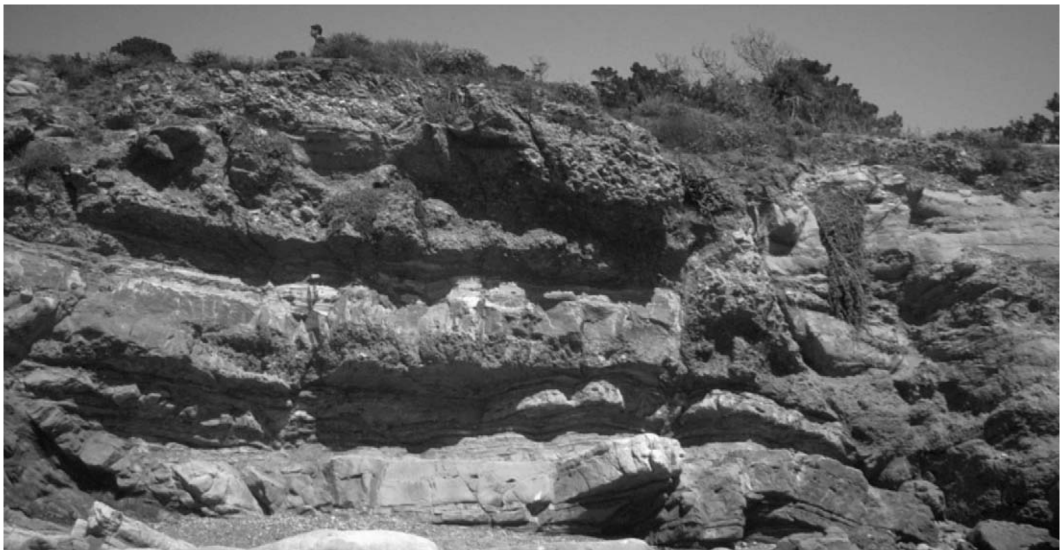

Fig. 1. Outcrop of reservoir rock, Point Lobos State Park, CA (photo courtesy P Corbett, Heriot-Watt University).

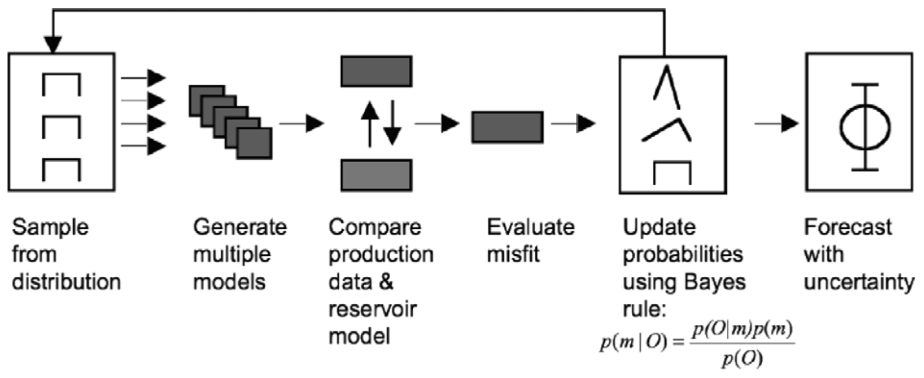
Sample from distribution
Generate multiple models
Compare production data & reservoir model
Evaluate misfit
Update probabilities using Bayes rule:
$p(m \mid O) = \frac{p(O|m)p(m)}{p(O)}$
Forecast with uncertainty

Fig. 2. Framework for generating multiple history-matched models.

an ancient river system, is likely to have a set of meandering channels with sinuosity, width, depth, etc that can be bounded by studies of equivalent outcrops. From these beliefs we form a way of parameterising the reservoir description and a set of prior probabilities for these parameters. We sample a number of possible reservoir descriptions from the prior and examine how well the predictions made using these reservoir descriptions fit the data. The models that fit the data well are assigned higher probabilities, with the numerical values of the probabilities given by Bayes rule. Bayes Theorem provides a formal way to update our beliefs about probabilities when we are provided with information. The statement of Bayes Theorem is [24]:

$$p(m|O) = \frac{p(O|m)p(m)}{\int_M p(O|m)p(m)\,\mathrm{d}m}. \tag{1}$$

The prior probability, $p(m)$, contains initial probabilities for the model parameters. These initial probabilities are chosen with limited prior knowledge about the reservoir available. At this stage the prior probabilities can be fairly vague, but should not be too narrow in its range (see [12] for a discussion on choice of non-informative priors). Using Bayes theorem, the model is updated, giving a posterior probability, $p(m|O)$, based on observations O. The likelihood function, $p(O|m)$, is a measure of to what degree the observed and modelled data differ. Computing the value of the likelihood is the key step in Bayesian analysis. The value of the likelihood comes from a probability model for the size of the difference between the observations and the simulations. The probability model contains assumptions about the statistics of the errors, and the quality of the uncertainty forecasts depends on the assumptions made in the likelihood.

The discrepancy – the difference between the observed value of some physical quantity and the simulated value – can be expressed as [8]

$$\begin{aligned} \text{discrepancy} &= \text{observed} - \text{simulated} & (2)\\ &= (\text{observed} - \text{true}) - (\text{simulated} - \text{true}) & (3)\\ &= \text{observed error} - \text{simulated error}. & (4) \end{aligned}$$

At any given time, the probability density of the discrepancy, which from Eq. (4) is given by the difference between the measurement error and simulation error, is given by a convolution integral

$$p(x) = \int p_{\text{meas}}(y)p_{\text{sim}}(x+y)\,\mathrm{d}y, \tag{5}$$

with the likelihood being given by the probability of the discrepancy being zero. This is a direct result of the additive nature of simulation and measurement errors.

If we assume Gaussian statistics for the errors, we end up with a standard least squares formulation with covariance matrix given by the sum of the measurement and solution error covariance matrices [25]. For this case, the misfit (which in this paper always refers to the negative log of the likelihood) is given by

$$M = \frac{1}{2}(\mathbf{obs} - \mathbf{sim})^{\mathrm{T}} C^{-1} (\mathbf{obs} - \mathbf{sim}). \tag{6}$$

Eq. (6) assumes that mean errors are zero for both measurement error and simulation error. This is unlikely to be the case for simulation error, in which case the mean simulation error must be included in Eq. (6). See [17] for a detailed discussion of construction of simulation error models including mean error terms.

Fig. 3 shows the conceptual framework used in our history matching approach. The top left picture represents the reservoir – the details of which are unknown except at a limited number of samples. Hydrocarbons have been produced from the reservoir and measurements have been taken of quantities such as pressures and fluid rates (top right). Based on our knowledge of the reservoir, we decide on a mathematical model – for example, black oil vs compositional, equation of state choices, and a parameterisation of the unknown aspects of the model (bottom left). We then solve the equations using a reservoir simulator (bottom right), and this introduces additional solution errors. We make inferences about the model parameters by comparing the numerical solution with the observed data.

There are several reasons why we might not get an exact match with observations. First, observations are subject to measurement error. This can take two forms: first there is the instrument accuracy – multiple observations of an identical physical value will usually yield close but not identical results; secondly, there is an error introduced in the comparison of the measurement to the simulated value. For example, in measuring bottom hole pressure, the pressure gauge may be above the perforations and will be measuring pressure at a single point somewhere on the casing. The flow up the well is likely to be three dimensional and may well be turbulent, so a simple 1D correction back to the datum depth will introduce uncertainties. A second reason why we might not get an exact match with observations is due to errors in the simulations. These errors can arise from multiple sources – for example a simulation is unlikely to be fully converged and will still have space and time truncation errors. Perhaps more importantly, on any finite grid size there will always be sub-grid phenomena that have been ignored. These are likely to be particularly important with coarsely gridded models. The final reason why we might not get an exact match is that the choice of model or parameterisation excludes predictions that agree with observations for any combination of parameters. In this case, the normalising constant in Bayes Theorem will be close to zero, indicating that a more suitable choice needs to be made.

2.1. Sampling in high dimensional parameter space

For most history matching problems we will be looking at several tens up to hundreds of unknown history matching parameters. Use of this number of problems creates a real challenge for any numerical method because of the combinatorial explosion of ways of choosing different values of parameters that could give a history match. Fig. 4 shows a plot of the resolution achieved in parameter space for a given number of unknown parameters if uniform Monte-Carlo sampling is used. The x-axis shows the resolution in terms of

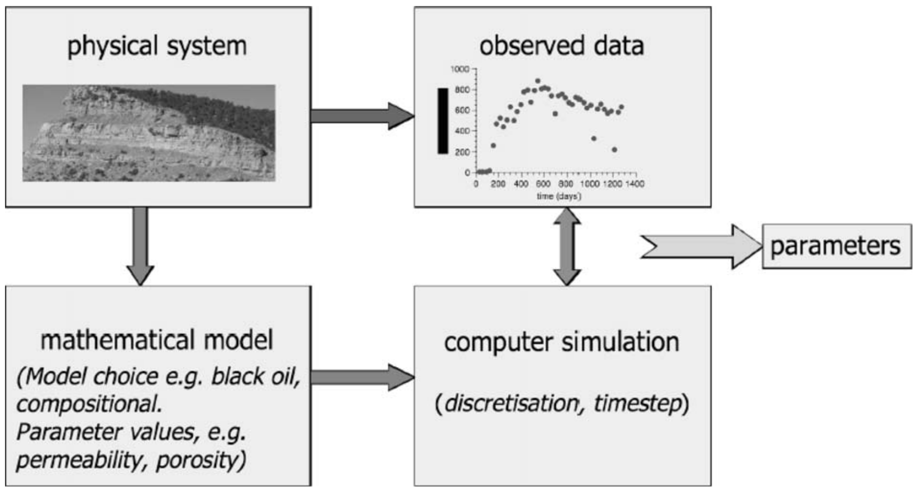

Fig. 3. Conceptual framework used in history matching.

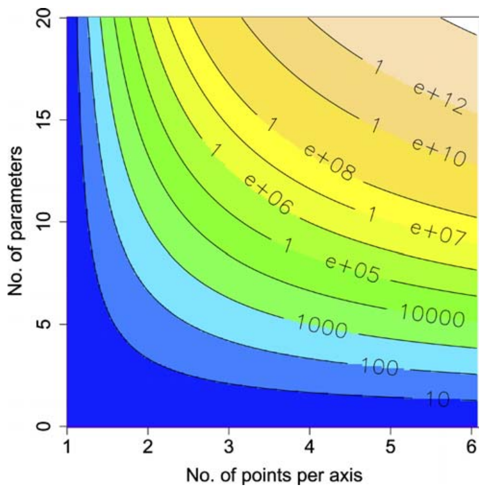

Fig. 4. Resolution in parameter space with uniform sampling.

the number of sampled points per parameter, and the y-axis is the number of unknown parameters. The shading represent the number of simulations to achieve that resolution. Since the CPU time of most reservoir simulation models ranges from 10 s of minutes for very simple small models to 24 h or more for large models, we can see that, even with large linux clusters, it is impossible to get significantly greater resolution than two points per axis for any reasonable number of parameters if we restrict ourselves to uniform sampling.

In general, misfit surfaces can have very complex structure. Fig. 5 shows a cross section of a misfit surface from a simple three-parameter problem (the IC Fault Model [5] described later), and we can clearly see that there are many local minima.

Since the misfit surface is used for posterior inference, the quality of sampling of the misfit surface determines the quality of our inference.

The complexity of the surface forces us to use adaptive sampling, so that the algorithm can concentrate its sampling in regions of good fit, and the multiple minima force us to use stochastic algorithms rather than gradient methods. This is primarily because we are making inference from all the point sampled – if we were seeking only a single maximum likelihood solution, it would make sense to combine a gradient algorithm with a stochastic search.

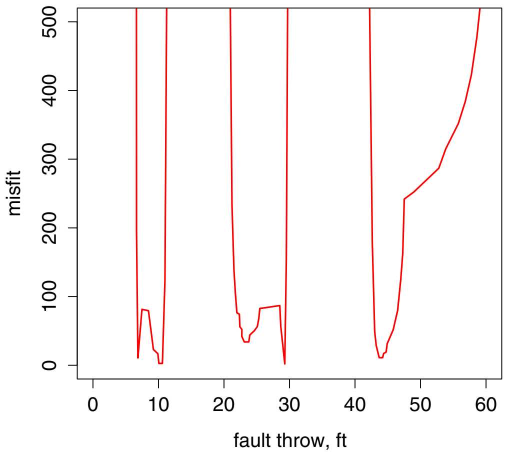

Fig. 5. 1D Section of misfit surface showing multiple minima.

2.2. Bayesian inference and the neighbourhood algorithm

In general, for posterior inference we are interested in integrals of the form

$$J = \int_M g(\mathbf{m})P(\mathbf{m})\,\mathrm{d}\mathbf{m}, \tag{7}$$

where $g(\mathbf{m})$ is some function of the reservoir model output. Because we have to run an expensive forward model, we use Monte-Carlo integration to estimate the value of the integral. The Monte-Carlo estimate is given by

$$\widehat{J} = \frac{1}{N}\sum_{k=1}^{N}\frac{g(\mathbf{m_k})}{h(\mathbf{m_k})}P(\mathbf{m_k}), \tag{8}$$

with $h(\mathbf{m}_k)$ being the sampling density.

In the applications discussed in this paper, we will be sampling the parameter space irregularly using a genetic algorithm [3]. Genetic Algorithms (GAs) are stochastic search and optimization techniques inspired by Darwin's theory of natural evolution. These algorithms are designed to search the parameter space by mimicking crossover, mutation and natural selection phenomena and employing a "survival of the fittest" strategy. Since the 1970s, they have been used for a wide range of optimization problems, including those which have complex objective functions with non-linearities or discontinuities [11]. In petroleum engineering research, the primary studies involving GAs date back to the mid-1980 on pipeline optimization. Then, over the last two decades the use of GAs has become more common in petroleum geology and engineering applications including stochastic reservoir modeling [23], well placement optimization [9], reservoir characterisation and history matching [19,22,7]. GAs are expected to be competitive on finding several minima in cases where the search space is large and not perfectly smooth [16]. Therefore, they are suitable for generating multiple good-fitting reservoir models in high dimensional parameter space to assess forecast uncertainties.

The use of a stochastic sampler means that we have to reconstruct the posterior probability density function from a limited number of irregularly spaced samples. We chose to use the NAB (NA-Bayes) algorithm described by Sambridge [21] which constructs a Voronoi mesh in parameter space, with the centre of each Voronoi cell being one of the sample points, and assumes that the posterior probability density is constant on each Voronoi cell. The NAB algorithm uses a Gibbs sampler to importance sample the posterior probability density, and so Eq. (8) becomes

$$\widehat{J} = \frac{1}{N}\sum_{k=1}^{N} g(\mathbf{m_k}), \tag{9}$$

with the Gibbs sampler ensuring that $h(\mathbf{m}_k) \approx P(\mathbf{m}_k)$.

3. Neural networks for response surfaces

It is important to be able to sample the misfit surface with adequate resolution, yet the number of simulations required increases exponentially with the dimension of the parameter space. There are a number of factors that mean that the task of quantifying uncertainty is less intractable than might first appear from the demands of fully resolving the misfit surface. Key amongst them is that it is rare for commercial or technical decisions to depend sensitively on all the unknown parameters [6]. There are also a number of sampling techniques, including Genetic Algorithms [3,5], and the Neighbourhood Algorithm [20] which adapt their sampling to the structure of the misfit surface.

However, generation of multiple history-matched models is still an extremely computationally expensive task using finite difference reservoir simulators. There are various ways to improve computational efficiency for uncertainty assessment. Streamline simulators [2] can be used to improve the speed of the forward runs, although it can be hard to quantify the errors introduced. Another approach is to use coarse simulation grids from upscaled models [26], where again an appropriate simulation error model must be used if reasonable results are expected. Alternative methods like experimental design can reduce the number of forward simulation run but have difficulties in exploring complex non-stationary misfit surface patterns.

In this paper, we examine an alternative approach to increase computing efficiency (reduce numerical cost) without losing accuracy in uncertainty assessment. The key problem is to improve exploration of the parameter space and interpolation of the misfit surface without using numerous time consuming forward simulations. Given a limited initial number of sampled models we aim to estimate the misfit surface with high resolution using a data-driven interpolation algorithm. The rationale of this approach is to identify regions of poor fit cheaply and save the expensive model runs for the good fitting regions of parameter space. The NAB algorithm for posterior inference uses Voronoi interpolation of the misfit surface in the multi-dimensional parameter space. However, there are a number of potential problems associated with this approach, especially if the cost of the forward simulations means that the resolution is limited. In our case, we are using the NAB algorithm with commercial reservoir simulators, and the CPU time for each forward model is several orders of magnitude greater than the receiver function inversion problem for which NAB was developed and tested. This means that our interpolation surface will in general be coarsely resolved.

The above problems motivate us to improve the present algorithm by combining it with an efficient interpolation model to obtain an accurate misfit surface at relatively low computer cost. Such problems are often effectively solved with machine learning (ML) algorithms. The Artificial Neural Network (ANN) is a machine learning algorithm which is well known and widely used in many fields. ANNs are data-driven – they depend only on the data and learn from them, thus do not assume any predefined analytical or statistical dependence. ANNs are very efficient at solving multi-dimensional interpolation problems, handling non-linearities and complex noisy data patterns. More details on ANNs can be found in [10].

3.1. A brief recap of artificial neural networks

Artificial Neural Networks were introduced in the 1960s as alternative data-driven learning models. Since then a broad field of Machine Learning (ML) algorithms has been developed and now unites data-driven and related algorithms: neural networks, genetic algorithms, support vector machines, fuzzy logic, pattern recognition, wavelets and many others. Among the advantages of neural networks are their ability to fit highly complex and non-linear data, from which the dependencies and patterns cannot be distinguished. They are also robust to outliers and noise in data. While neural networks are often described algorithmically, they can usefully be viewed as statistical models with parameters estimated from data [14]. Multi-layer perceptrons are well known types of neural networks. A basic multi-layer perceptron (MLP) structure is shown in Fig. 6. The functional form of the neural network is given by

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j \psi(\gamma_{\mathbf{j}}^{\mathbf{t}} \mathbf{x_i}) + \epsilon_i. \tag{10}$$

The model inputs ($\mathbf{x}_i$) are passed to the hidden neurons which operate on linear combinations of the model inputs ($\gamma_{\mathbf{j}}^{\mathbf{t}}\mathbf{x_i}$). ψ is the sigmoidal shaped basis function, and β_j are the nodal weights. The number of inputs depends on the number of parameters in the model. Hidden neurons can be arranged in one or two hidden

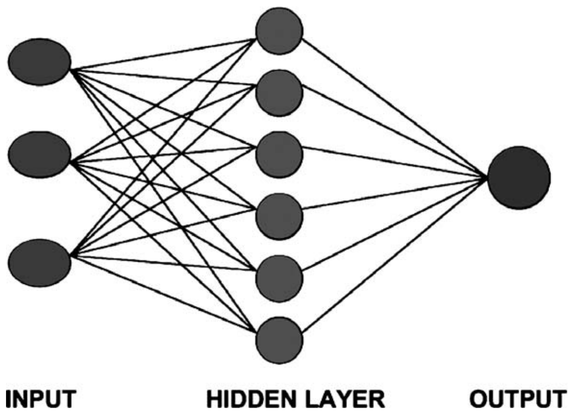

Fig. 6. Basic MLP structure: 3 inputs, 6 neurons in a single hidden layer, and 1 output.

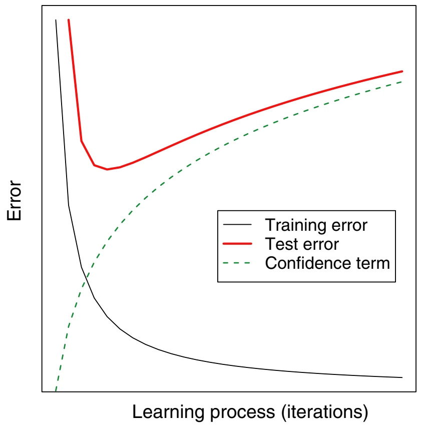

Fig. 7. Risk minimization theory: training and test error dynamics whilst learning progress.

layers and the hidden layer in general is responsible for modeling non-linearity. The output neurons correspond to the target functions that are being modelled. The output layer sums the signals from the hidden layer and delivers the final prediction.

Multi-layer perceptrons are trained using a supervised learning rule – the backpropagation algorithm. The backpropagation algorithm is an iterative algorithm designed to minimise the difference between the actual output of the neural network and the desired output. It calculates the error measure and updates the weights on all neurons. The recursive algorithm starts at the output neurons and works back to the first hidden layer propagating the output errors backward (additional details can be found in [13,4]). One of the key tasks in ANN modelling is to select representative training and test data sets for the training procedure. Usually, the training set is used in MLP training (via back propagation) when the records from the data set are directly exposed to the MLP inputs and outputs. The test data set is used to control the quality of the training procedure. According to the general theory of risk minimization, both training and test error decrease initially once the training starts; the training error decreases gradually towards zero given an efficient optimization algorithm, while the test error reaches its minimum and then goes up again (see Fig. 7 [13]). At this point MLP starts losing its ability to generalise – to predict accurately data different from the training set, and such overlearning (or overfitting) should be avoided.

4. Applications

4.1. The IC Fault Model

The IC Fault Model is an extremely simple three-parameter model set up by Carter [5] as a test example for automated history matching. It has proved extremely difficult to history match, and one of the conclusions published in [5] has been that the best history-matched model (from 159,000 models) is of limited use as a predictor. The geological model consists of six layers of alternating high and low permeability sands (Fig. 8). The three high permeability layers have identical properties, and the three low permeability layers have a different set of identical properties. The porosities of the high and low permeability layers are 0.30 and 0.15 respectively. The thickness of the layers has an arithmetic progression, with the top layer having a thickness of 12.5 ft., the bottom layer a thickness of 7.5 ft., and a total thickness of 60 ft. The width of the model is 1000 ft., with a simple fault in the middle. There is a water injector well at the left-hand edge, and a producer well on the right-hand edge. Both wells are completed on all layers, and operated at fixed bottom hole pressures. The simulation model is 100×12 grid blocks, with each geological layer divided into two simulation layers with equal thickness, each grid block is 10 ft. wide. The well depth is 8325–8385 ft. The model has three unknown parameters for history matching: high and low permeability (k_{high} and k_{low}) and the fault throw (h). Our prior model has uniform distribution with ranges: $k_{\text{high}} \in [100, 200]$, $k_{\text{low}} \in [0, 50]$ and $h \in [0, 60]$. The IC Fault Model study specified a point in the parameter space as the true parameter values: $k_{\text{high}} = 131.6$, $k_{\text{low}} = 1.3$ and $h = 10.4$.

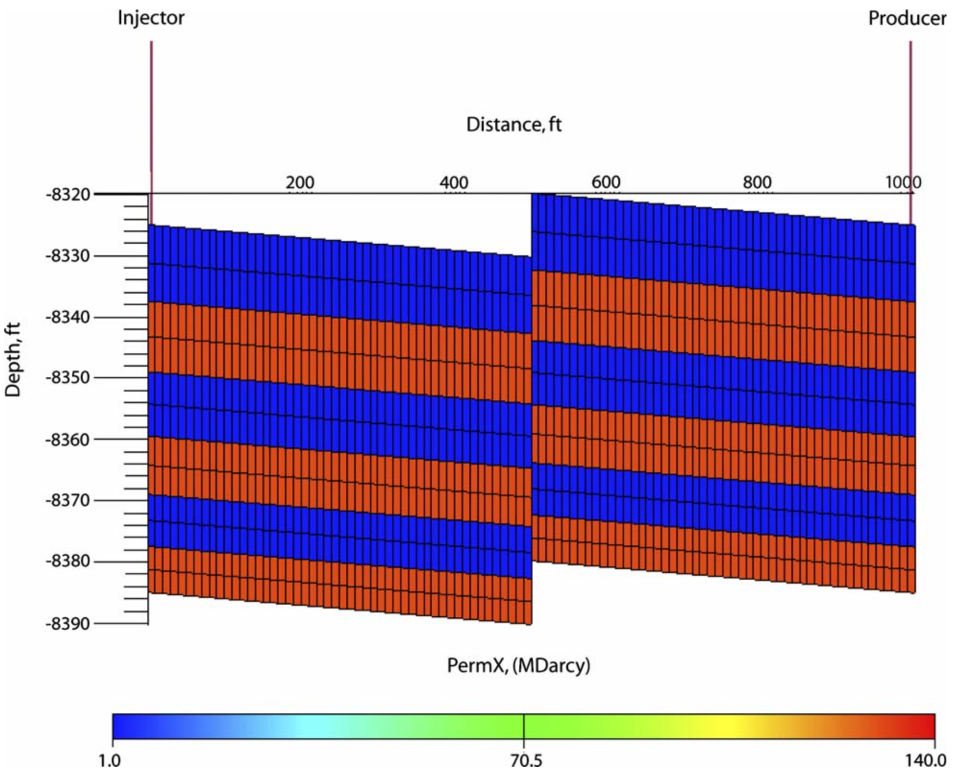

Fig. 8. IC Fault Model (Z. Tavassoli, Jonathan N. Carter, and Peter R. King, Errors in history matching, SPE 86883, 2004).

The misfit was defined using a standard least squares model using the discrepancy between simulated and observed oil and water production rates for the 3-year history-matching period. The standard deviation is proportional to the observed data and is given by $\sigma_p = 0.03 \times \text{obs}_p$ where p is the measured parameter (oil and water production rates in this case). The likelihood used was the same as the original IC Fault Model study, namely the exponential of the negative average misfit over times where the reservoir was producing, i.e. $\exp(-M)$ where

$$M = \frac{1}{n_\text{o}} \sum_{i=1}^{n_\text{o}} \frac{(\text{obs}_\text{o} - \text{sim}_\text{o})_i^2}{2\sigma_\text{o}^2} + \frac{1}{n_\text{w}} \sum_{i=1}^{n_\text{w}} \frac{(\text{obs}_\text{w} - \text{sim}_\text{w})_i^2}{2\sigma_\text{w}^2}. \tag{11}$$

Oil rate squared discrepancies are summed over the history matching period, while the water rate squared discrepancies are summed over the interval from the start of water production to the end of the history matching period.

4.2. Bayesian inference using direct simulation

Since one of the conclusions of the original study [5] was that the maximum likelihood model had no predictive value, we examined the use of stochastic sampling and Bayesian inference to put uncertainty bounds around the maximum likelihood model. We used two stochastic algorithms to search parameter space for models that fit the observed data well.

The first stochastic algorithm we used was a steady state Genetic Algorithm (GA) [3], which was the most efficient GA for this particular problem. In a steady state GA a number of the best models in each generation are retained and passed on to the next generation step. This provides more stable optimization and saving in computational cost.

In the IC Fault Model, ‘‘convergence’’ of our steady state GA was achieved after generating an ensemble of 6000 models. For this application, ‘‘convergence’’ means that the GA started sampling in a relatively narrow

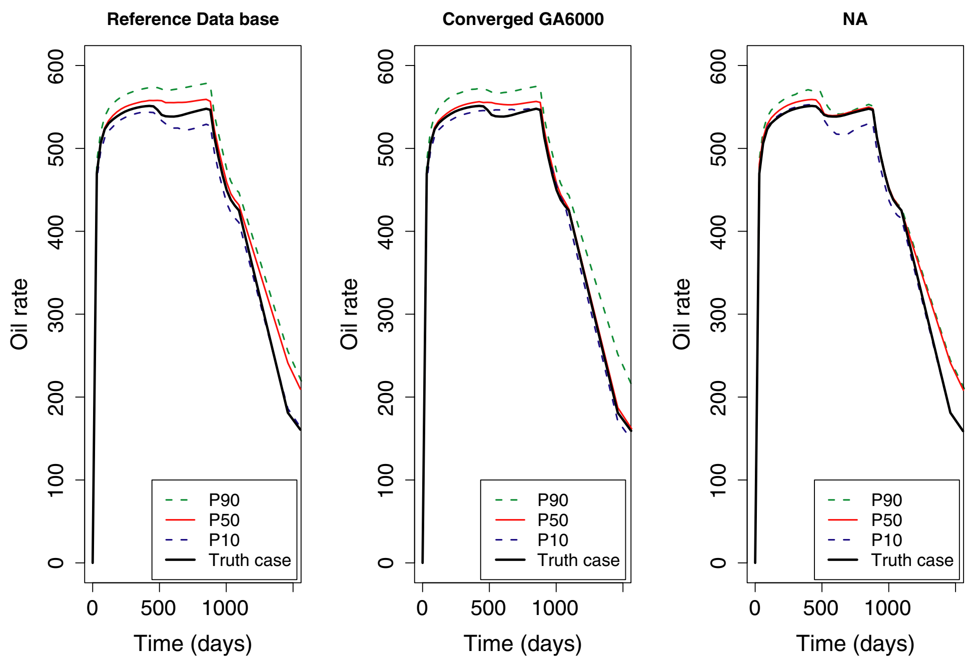

Fig. 9. Uncertain inference from GA and NA sampling algorithms for IC Fault Model.

range of parameter space. This ensemble retained only 328 relatively likely models (due to the nature of steady state GA) from the 6000 samples. The high misfit models are discarded as the algorithm evolves the population.

We also used the Neighbourhood Algorithm [20], which works by approximating the misfit surface in high dimensions by constant values in a set of Voronoi cells around each sample location and then selectively refining the low misfit locations.

We then used the NAB code [21] described in Section 2.2 to determine the $p10$, $p50$, $p90$ values for the oil and water rate time series. At each point in time, there is an 80% probability that the true value of oil or water production rate lies between the $p10$, $p90$ bands shown. We compared these with inference from the full 159,000 models in the IC Fault Model database provided by Carter, and the results are shown in Fig. 9. We are able to accurately capture the uncertainty bounds with a relatively low number of samples using a choice of stochastic sampling algorithms.

4.3. Use of MLP to predict misfit surface

We examined three ways to use an MLP to estimate the misfit for given model parameters. The first way is to use the MLP directly for misfit prediction. In this case the MLP structure is simple, with only a single output neuron, and requires less training. The second and third ways obtain MLP predictions of production variables (oil and water production rates) and evaluate the misfit based on their estimated values rather than the exact rates from the reservoir simulator.

In general, the compute time required to generate an ensemble of models with adequate resolution of areas of high probability is high. This observation motivates our use of MLP interpolation based on models generated early in the sampling process. We used the 246 low misfit models from the first 2000 model generation steps in the GA run (one third of the total run time) to predict the misfit surface with increased resolution.

4.3.1. Direct prediction of misfit

In order to make direct predictions of the misfit, we created an MLP with 3 inputs (corresponding to the 3 model parameters) and 1 output (corresponding to the misfit). The data was initially split into training and test sets. This separation was performed by imposing a $3 \times 3 \times 3 \times 3$ grid in the four-dimensional space formed by the three parameter values and the misfit. A number of points (184) were then sampled at random in each grid cell to select the training set, with the remainder (62) forming the test set.

The number of hidden neurons was chosen to give good predictions of misfit without overfitting the available data. We chose an MLP structure with 50 neurons in a single hidden layer, providing 200 connections in the MLP (between 3 input, 50 hidden and 1 output neurons). Given 184 training samples with 3 input data points for each sample (552 values) this provides approximately 3 misfit values per MLP connection. This ratio is justified due to the expected complexity of misfit surface. Slight variations in the number of hidden neurons make no significant differences to the estimated misfit values. The size of the training dataset should be larger than the number of the MLP connections(degrees of freedom), otherwise the network would not be able to train but just reproduce the given values. If there are too few connections, the MLP will be unable to represent complex structures in data. The training of the MLP was performed in two steps: first, simulated annealing was used to find stochastically optimal starting weights for the second step: Levenberg–Marquardt gradient optimization [15]. The data were scaled to $[0.01, 0.99]$ interval and the misfit values were additionally log-transformed to mitigate the influence of extremely high values. According to the usual practice, the root mean square error (RMSE) was taken as the training evaluation criterion. This choice prevents overlearning when the MLP becomes able to reproduce only the training data and loses its ability to generalise (predict independent test data). The trained MLP was used to find additional locations with low misfits. After this procedure the model ensemble was enlarged by 923 models with MLP estimated misfits, which were added to the initial 246 survived models to make more representative ensemble for the inference stage.

4.3.2. Estimation of misfit using MLP prediction of production time series

The second way we used an MLP to compute the misfit estimates was to predict a time series of the production variables and then use these to evaluate the misfit value. MLP can be used successfully for temporal predictions. However, the MLP structure becomes more complicated as we need to take into account the additional time variable. Time can be incorporated into the MLP structure as an additional input parameter along with the model parameters (perm levels and the throw). Two production history variables (oil and water rates) become MLP outputs. In this case such MLP is trained based on the simulated production history (at 38 time steps) for all GA generated models (246). Thus the training data ensemble is quite large over 9000 records (which were split into 7703 training and 1401 test sets). An MLP with 50 hidden neurons in a single layer was trained for a significantly longer time than the MLP for direct misfit predictions.

There are several problems with MLP modelling of the time series of oil and water rates. A set of rates for different times are combined and presented to the MLP model which treats time as simply another parameter similar to the geomodel parameters (permeability, throw). Hence, the MLP does not implicitly recognise the effects of causality in the data. Another important draw back of this approach is a lack of physical meaning of the predicted variables. Thus, for instance, water production should be constrained to be positive, and linked to oil rate after breakthrough.

4.3.3. Generalised linear model for misfit prediction

The final approach we considered for modelling the production history time series to calculate the misfit was to use a generalised linear model (GLM). In the IC Fault case study the production profiles for oil and water rates are relatively simple and can be approximated by a combination of simple analytical functions to produce physically more realistic production curves.

The GLM for oil rate was

$$\begin{aligned} q_o(t) &= a_1 \log(t) + b_1 t + c_1 \quad t < t_1, \\ q_o(t) &= a_2 \quad t_1 < t < t_2, \\ q_o(t) &= a_3 t + b_3 \quad t > t_2, \end{aligned} \tag{12}$$

with the parameters estimated by the MLP being $(a_1, a_2, a_3, b_1, b_3, c_1, t_1)$. The history period starts 1 month after the start of production, so there is no problem with the $\log(t)$ term in Eq. (12).

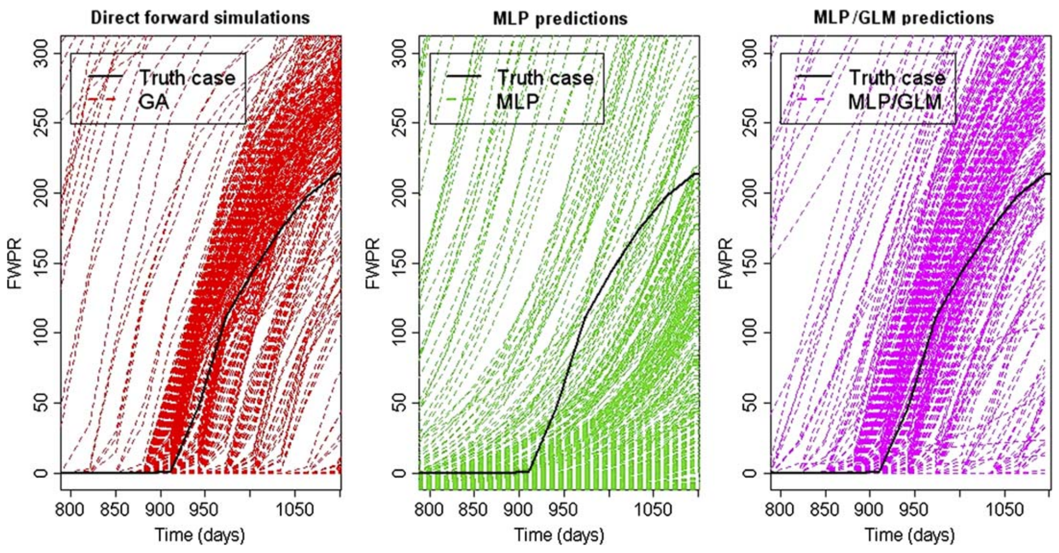

Fig. 10. Time series and GLM predictions of water rate compared with direct simulation.

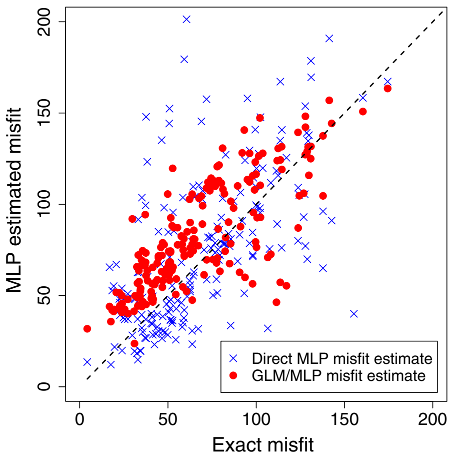

Fig. 11. Scatterplot of direct misfit estimation and GLM estimation against exact misfit.

The water rate GLM was

$$\begin{aligned} q_w(t) &= 0 \quad t < t_2, \\ q_w(t) &= at^2 + bt + c \quad t > t_2. \end{aligned} \tag{13}$$

This provides us with 10 parameters for explicit reproduction of the oil and water rate curves. These parameter values are obtained from fitting GLMs to the production data for available from the forward simulations carried out for a limited number of models. Hence, for 246 sets of k_{high}, k_{low}, h there are 246 corresponding sets of 10 parameters, which represent the simulated production for the models. Then, we can build and train an MLP neural network to predict the GLM coefficients for any combination of the geomodel parameters. In this study, we used separate MLPs for oil and water rates with the same 3 inputs. The MLP for oil rate has 7 outputs ($a_1, b_1, c_1, t_1, a_2, a_3, b_3$) and 40 hidden neurons. The MLP for water rate has 3 (a, b, c) outputs and 20 hidden neurons. The water break through time t_2 is common to both models and is derived from the water rate GLM prediction when $q_w(t_2) = 0$. Both MLPs were trained using 246 data records as described above. The MLP for water rate is smaller and took less training time than the oil rate MLP. A comparison of

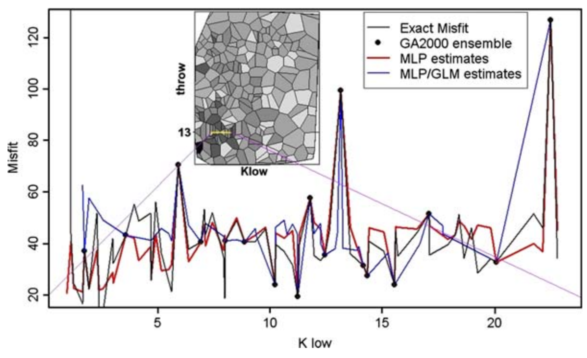

Fig. 12. Comparison of misfit predictions with exact misfit for 3 MLP methods.

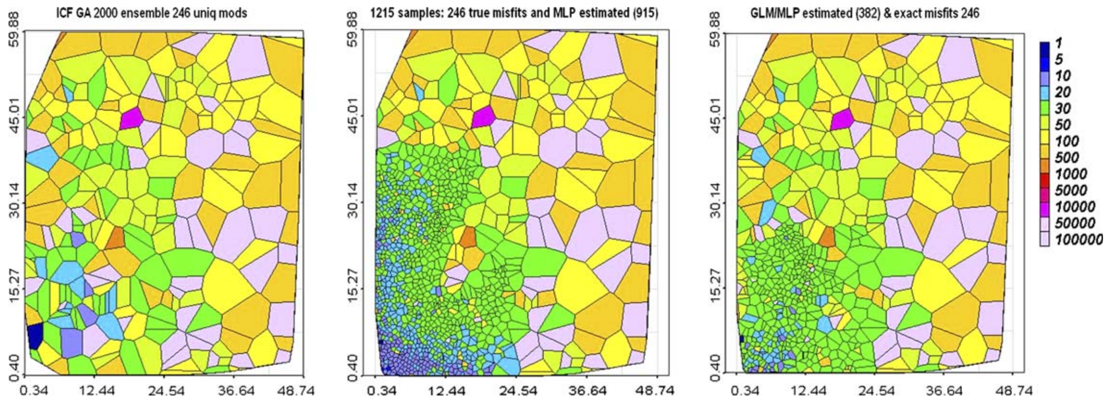

Fig. 13. Misfit surface for coarsely sampled surface plus MLP extended sampling.

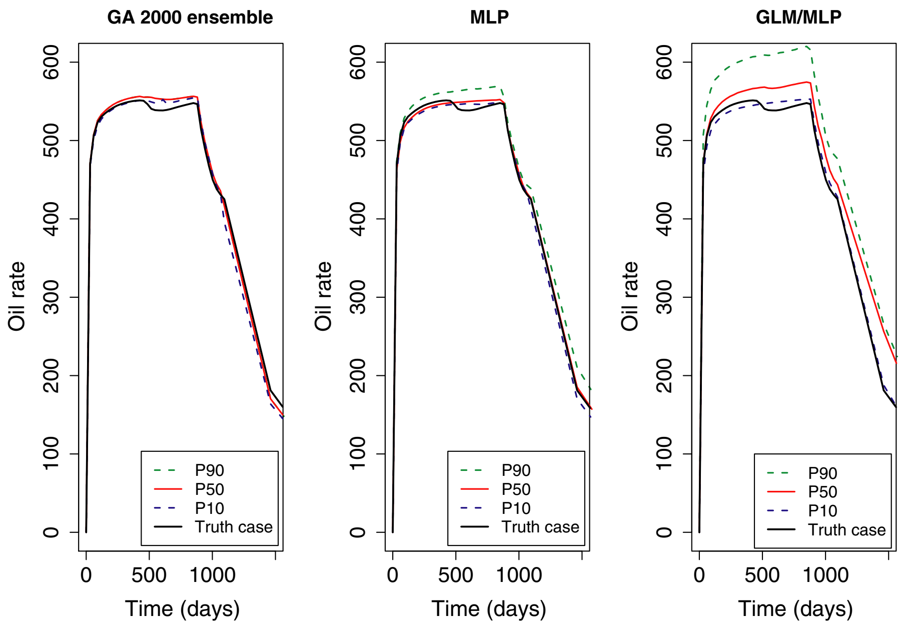

Fig. 14. MLP predictions of oil production for ensemble of GA models.

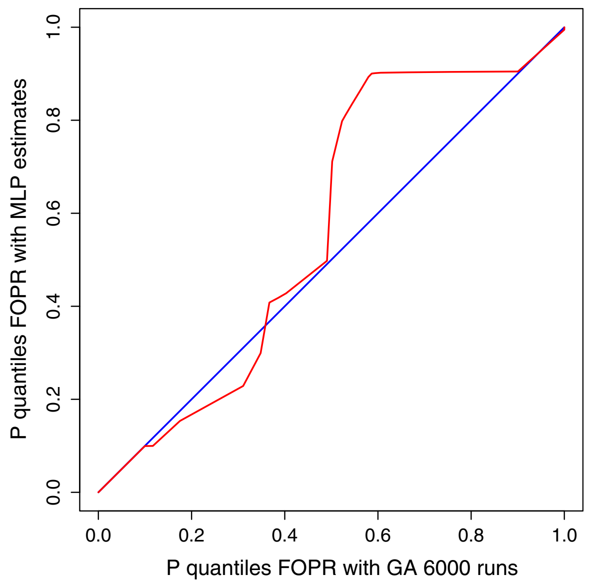

Fig. 15. Calibration curve of oil rate probability at the end of the history match period.

the water rate predictions for direct simulation, the time series approach, and the GLM approach is shown in Fig. 10, which shows that the shape of the curves generated by the GLM follow those from direct simulation much more closely than the time-series approach.

Once the GLM coefficients can be determined for any model in the parameter space, the relevant misfit is computed for the GLM estimated production. The estimated misfits are compared with the exact ones for the direct MLP estimation method and estimation using MLP predicted coefficients for GLM production model (see Fig. 11) The direct MLP misfit estimates feature better correlation, while the ones using GLM suffer from a positive bias (although the least squares criterion used to fit the model is unbiased, we are looking at a small part of the fitting region, namely the low misfit models, in this figure). This causes overestimation of the misfit surface in general, though some relative features of the local misfit surface minima are in common for both methods. A comparison of the methods along a cross section through the misfit surface near the true solution is shown in Fig. 12.

4.4. *Inference with MLP predictions*

We compared our predictions of uncertainty in oil and water rates using the MLP with those obtained from exact simulation of the flow equations shown in Fig. 9. To predict the confidence intervals, we used NAB resampling to generate the posterior probabilities. We compared the results using the initial GA ensemble with 2000 (246 low misfit) models; the extended GA ensemble of 6000 (328 low misfit) models, and the ensemble with an additional 969 models estimated using the MLP trained to estimate misfit directly. The additional sampling is shown in Fig. 13, which plots the sampled points projected onto a 2D Voronoi diagram. The increased resolution provided by both the direct estimation of misfit and calculation using a GLM can clearly be seen.

The results of the Bayesian inference using these extended misfit surfaces are presented as $p50$ production prediction curves bounded by $p10$, $p90$ confidence intervals. The truth case results are also shown. Fig. 14 shows that the confidence interval based on a limited ensemble of models generated by the initial 2000 GA model runs is unrealistically narrow compared with the one based on the extended GA ensemble of 6000 or the full ensemble shown in Fig. 9. The confidence intervals from the GLM are wider than those predicted from direct estimation of misfit, and overestimate the uncertainty.

The best results are those obtained with direct estimation of misfit using the MLP. These results demonstrate good agreement with the extended GA sampling, matching the $p50$ prediction and providing a realistic range of uncertainty given by the $p10$, $p90$ interval. The good agreement is also illustrated by the calibration curve shown in Fig. 15. In Fig. 15 the cumulative probability of the oil production at time step day 1095 (end of history match period) is plotted the cumulative probability obtained from the MLP predictions. The result shows excellent agreement (almost straight line) up to the 50% percentile, and reasonable agreement for higher probabilities.

5. Conclusions

In many aspects of engineering design, we are faced with the task of quantifying the uncertainty in a prediction for cases in which the simulations needed are very time consuming to run. Predictions of fluid flows in oil reservoirs is an example of such a system.

In this paper, we have examined the use of a surrogate system, namely a neural network, to improve the efficiency of our sampling of parameter space. Our goal is to seek model parameter values that match observed system responses and to make inferences about system performance using all parameter values with an acceptable degree of match.

Our approach is to use a stochastic sampling algorithm to make an initial exploration of parameter space. We use the misfit values from this initial exploration to train a neural network. We then use this neural network to guide our sampling, biasing our sampling to regions of good fit. Note that we must still continue some sampling in regions where the neural network says that the fit will be bad if we wish to ensure that we can find regions of good fit that have been missed in the initial exploration of parameter space.

The approach was demonstrated on a simple three parameter sampling problem that had proved a difficult problem in previous studies. By using the neural network to guide sampling within the context of a stochastic search algorithm, and running the expensive forward model principally in regions of good fit, we are able to easily generated a significant number of models that match history well. If these models are then used in a Bayesian inference step, a good forecast of uncertainty is obtained, comparable with the uncertainty envelope generated with two orders of magnitude additional simulations in the original study.

The best results were obtained using direct prediction of misfit with a trained multi-layer perceptron. This method provided an estimate of uncertainty that was very close to the estimate provide using a steady state genetic algorithm which involved a significantly higher number of forward model evaluations.

Acknowledgements

Funding for this work was provided by UK Engineering and Physical Sciences Research Council (GR/T24838/01) and by the industrial sponsors of the Heriot-Watt Uncertainty Project. We also thank Jonathan Carter of Imperial College, UK, for making available the IC Fault Model input data and results database, Malcolm Sambridge of ANU for many helpful discussions, and Schlumberger-Geoquest for providing access to the ECLIPSE reservoir simulator.

References

[1] J.S. Archer, C.G. Wall, Petroleum Engineering: Principles and Practice, Kluwer Academic Publishers, 1986.
[2] R.P. Batycky, M.J. Blunt, M.R. Thiele, A 3D field-scale streamline based reservoir simulator, SPE Reserv. Eval. Eng. 4 (1997) 246–254.
[3] P.J. Bentley, Evolutionary Design by Computers, Morgan Kaufmann, 1999.
[4] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[5] J.N. Carter, P.J. Ballester, Z. Tavassoli, P.R. King, Our calibrated model has no predictive value: An example from the petroleum industry, in: Proceedings of the Fourth International Conference on Sensitivity Analysis, 2004.
[6] M.A. Christie, J. Glimm, J.W. Grove, D.M. Higdon, D.H. Sharp, M.M. Wood-Schultz, Error analysis and simulations of complex phenomena, Los Alamos Sci. 29 (2005) 6–25.
[7] F.J.T. Floris, M.D. Bush, M. Cuypers, F. Roggero, A.-R. Syversveen, Methods for quantifying the uncertainty of production forecasts: a comparative study, Petrol. Geosci. 7 (2001) S87–S96.
[8] J. Glimm, D.H. Sharp, Prediction and the quantification of uncertainty, Physica D (1999).
[9] B. Guyaguler, R.N. Horne, L. Rogers, J.J. Rosenzweig, Optimization of well placement in a Gulf of Mexico Waterflooding Project, Society of Petroleum Engineers Preprint, 78266, 2002.

[10] S. Haykin, Neural networks. A comprehensive foundation, Macmillan College Publishing Company, New York, 1999.
[11] E.R. Jaffery, Design applications of genetic algorithms, Society of Petroleum Engineers Preprint, 26367, 1993.
[12] E.T. Jaynes, Probability Theory: The Logic of Science, Cambridge University Press, 2003.
[13] M. Kanevski, M. Maignan, Analysis and Modelling of Spatial Environmental Data, EPFL Press, 2004.
[14] H.K.H. Lee, Bayesian Nonparametrics via Neural Networks, ASA-SIAM, Philadelphia, 2004.
[15] T. Masters, Practical Neural Network Recipes in C++, Academic Press, 1993.
[16] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, 1996.
[17] A. O'Sullivan, M.A. Christie, Error models for reducing history match bias, Comput. Geosci. 9 (2) (2005).
[18] T.N. Palmer, Predicting uncertainty in forecasts of weather and climate, Rep. Prog. Phys. 63 (2000) 71–116.
[19] C.E. Romero, J.N. Carter, A.C. Gringarten, R.W. Zimmerman, A modified genetic algorithm for reservoir characterisation, Society of Petroleum Engineers Preprint, 64765, 2000.
[20] M. Sambridge, Geophysical inversion with a neighbourhood algorithm – 1. Searching a parameter space, Geophys. J. Int. (138) (1999) 479–494.
[21] M. Sambridge, Geophysical inversion with a neighbourhood algorithm – II. Appraising the ensemble, Geophys. J. Int. (138) (1999) 727–745.
[22] R.W. Schulze-Riegert, J.K. Axmann, O. Haase, D.T. Rian, Y.-L. You, Evolutionary algorithms applied to history matching of complex reservoirs, Society of Petroleum Engineers Preprint, 77301, 2002.
[23] M.K. Sen, D.G. Akhil, P.L. Stoffa, L.W. Lake, G.A. Pope, Stochastic reservoir modeling using simulated annealing and genetic algorithm, Society of Petroleum Engineers Preprint, 24754, 1995.
[24] D.S. Sivia, Data Analysis – A Bayesian Tutorial, Clarendon Press, Oxford, 1996.
[25] A. Tarantola, Inverse Problem Theory, Methods for Data Fitting and Model Parameter Estimation, Elsevier Science Publishers, Amsterdam, 1987.
[26] O.I. Tureyen, J. Caers, A parallel multiscale approach to reservoir modelling, Comput. Geosci. 9 (2) (2005).