HISTORY MATCHING WITH UNCERTAINTY QUANTIFICATION

Lars Holden, Harald H. Soleng, and Anne Randi Syversveen
Norwegian Computing Center
PO-Box 114, 0314 Oslo,
Norway

SUMMARY

We present three methods for history matching and uncertainty quantification tested on a synthetic test case. The test case contains 1761 active grid blocks. There are six production wells in the reservoir.

A Bayesian approach is used. Based on production data and well observations of permeability and porosity, samples from the posterior distribution for permeability and porosity are generated. By running a reservoir simulator on each of these samples, a production forecast with estimated uncertainty is generated.

The first method is the Oliver approach. This is an approximative method that gives samples from a distribution which is believed to be close to the posterior distribution.

The second method is based on a genetic algorithm. This is an optimisation technique, and we do not generate samples from the posterior distribution. But we get a very good match to the production data.

The third method is an adaptive Metropolis–Hastings (MH) algorithm. Ordinary MH methods are too slow to use for history matching because the reservoir simulator has to be run for each iteration. However, this method will sample correctly from the posterior distribution. The adaptive method makes use of earlier proposals to create new proposals. In this way, fewer iterations are needed to get convergence, compared to ordinary Metropolis–Hastings.

1. INTRODUCTION

The work reported here is a part of the EC sponsored PUNQ (Production forecasting with Uncertainty Quantification) project. The PUNQ project considers the history matching problem in a Bayesian setting.

A synthetic case study was set up, based on a real field case. Production data was generated for the synthetic reservoir model. The model contains $19 \times 28 \times 5$ grid blocks, of which 1761 blocks are active. The reservoir is characterised by porosity, vertical and horizontal permeability.

The partners were given the production data added some Gaussian noise, and well data, consisting of permeability and porosity values in six production wells, also added Gaussian noise. The production data are collected for 8 years. The total simulation period, including the forecasting period is 16.5 years. The production data to be matched are bottom hole pressure, gas-oil ratio and water cut. Five infill wells were added after the end of the history matching period. We have no well data or production data from these wells. The true reservoir was not known for the partners before the end of the project.

A more throughly description of the test case is given in Floris *et al.* [1].

2. DESCRIPTION OF MODELS

2.1 Stochastic model

We use a transformed Gaussian random field as prior model for the porosity, vertical and horizontal permeability conditioned on well observations. The transformation is such that values of the Gaussian random field correspond to porosity values between 0 and 0.3. For the horizontal and vertical permeability, we used the log-transform in order to get only positive permeability values.

The Gaussian random field is constructed as follows. Let R be a vector containing all reservoir characteristics. Given a vector of model parameters, θ, we write

$$R = F\,\theta + \epsilon_r$$

where F is a matrix and ϵ_r is Gaussian random noise with zero mean and covariance matrix Σ_r. The stochastic counterpart to θ, Θ, is also Gaussian with mean μ_θ and covariance matrix Σ_θ. The vector θ is of length 21, and represents a constant term in each layer for each of the three characteristics, plus a linear term in x- and y-direction for each of the characteristics. The parameters in the priors were chosen uncorrelated and almost non-informative. Well observations of reservoir characteristics, o_w are assumed to be Gaussian distributed after the transformation mentioned above. Since both prior and likelihood are Gaussian, the posterior distribution for R given only well observations is Gaussian, and can be written as

$$f(r|o_w) \propto \int f(r|\theta)\, f(\theta)\, f(o_w|r)\, d\theta. \tag{1}$$

The production data are assumed to be equal to the output of the reservoir simulator plus some Gaussian distributed noise,

$$O_p = \omega(r) + U_p$$

and the likelihood function for production data is given by

$$L(o_p|r) = N(\omega(r), \mathbf{\Sigma}_p)$$

where ω is the reservoir simulator and o_p is observed production data. In our case, the production data are independent, so the covariance matrix $\mathbf{\Sigma}_p$ is a diagonal matrix. Therefore the likelihood function can be written as a product of one-dimensional Gaussian densities. The variances are reported in Floris *et al.* [1]. The posterior model, from which we want to sample is now given by:

$$f(r|o_p, o_w) \propto f(r|o_w)\, L(o_p|r) \tag{2}$$

2.2 The Oliver approach

A generalisation of the Oliver approach is used, see Oliver *et al.* [2] and Omre *et al.* [3]. This simulation algorithm will only sample exactly from (2), if the reservoir simulator had been a linear function. Tjelmeland [4] and Omre *et al.* [5] have done some tests on the goodness of this approximation. The algorithm tend to have too large variability in the pilot points.

First, we generate a start realization of the reservoir characteristics conditioned to well data from the distribution (1). An object function consisting of two terms is defined. One term measures deviation from the start realization, and one term measures deviation from the observed production data added some Gaussian noise with covariance Σ_p. The object function is given by

$$(r - r_s)\Sigma_c^{-1}(r - r_s)^T + \left(\omega(r) - o_p^0\right)\Sigma_p^{-1}\left(\omega(r) - o_p^0\right)^T \tag{3}$$

where Σ_c is the covariance matrix for the Gaussian distribution (1) and (r_s, o_p^0) are the start realization and the production data added noise, also referred to as the pivot sample. The object function (3) is minimised with respect to a set of control variables. We use 90 control variables which are the porosity and horizontal and vertical permeability values in six grid nodes in all five layers.

The stochasticity of the Oliver algorithm lies in generating the start realization from the prior and the noise added to the production data. If the object function instead measures deviation from the prior mean and the production data without noise added, minimisation of the object function gives the MAP estimator.

2.3 The genetic algorithm

Genetic algorithms are efficient optimisation methods which can be used in optimisation problems with multi-modal objective functions or where traditional analytical methods fail.

The steady state genetic algorithm starts by initialising a population of N individuals. In each generation a fixed fraction of the population is selected for mating using *fitness* as a selection criterion. This produces a number of offspring which are added to the population. The population is then cut back to its original size by removing the least fit individuals. Through Darwinian ‘survival of the fittest’, the fitness of the population increases.

The fitness function φ was chosen to be

$$\varphi = \frac{1}{1 + \sqrt{Q}}$$

where Q is a weighted sum of squared deviations between observed and simulated historic data. Thus $\varphi \in \langle 0, 1]$ with 1 for a perfect fit.

father	fffff
mother	mmmmm
1st child	fffmm
2nd child	mmmff

Table 1: One point crossover scheme.

In order to apply a genetic algorithm to history matching one must define a *genetic* representation of the reservoir. The reservoir genome was coded as a single chromosome. The number of genes in the chromosome equals the number of active grid blocks. Each gene carries a set of three petrophysical block values, namely the horizontal permeability, the vertical permeability, and the porosity. Thus in this model, each gene codes the complete set of petrophysical data for a reservoir block. Recombination was coded as simple one point crossover of the chromosome; a pair of parents gets two children each with opposite parts of the genome from each of the parents (Table 1). We added a 1% probability of mutation with the mutation operator equal to grid block swapping (observed blocks in well had zero probability of participating in this operator).

We used Gaussian fields conditioned on well observations in the initialisation function. However, crossover and mutation lifted the restriction of Gaussian fields. Therefore, the history matched reservoir is not Gaussian distributed. By making prediction from a number of best-of-history individuals from 10 independent runs, the prediction uncertainty could be estimated. Further details can be found in Ref. [6].

2.4 The adaptive genetic Metropolis–Hastings algorithm

Experience shows that ordinary Metropolis–Hastings algorithms with proposals drawn from the prior distribution converge too slowly in history matching. The probability of drawing a reservoir that fits the production data well enough is too small. In the adaptive Metropolis–Hastings algorithm, all earlier proposals are used to generate new proposals, see Holden [7]. In this way, the hope is to speed up convergence by making better proposals.

The proposals are inspired by the genetic algorithm. First, a population is generated from the prior distribution conditioned to well observations. New proposals are generated as linear combinations of two individuals of the population and one realization from the prior (mutation) in addition to a kriged field. The linear combinations are chosen such that the proposal gets the same mean and variance as the prior conditioned to well data. The new proposal is taken into the population with probability proportional to the likelihood function. In this way, the population changes during the iterations, and the individuals give better and better match to the production data. The chain is generated using Metropolis–Hastings acceptance criteria with proposals generated from the current population.

In theory, this method samples correctly from the posterior density (2). However, the acceptance rate is hard to evaluate exactly because the prior distribution must be evaluated. Therefore we only calculate the acceptance rate approximately correct, but we believe that we will be close to the correct posterior distribution.

3. RESULTS AND CONCLUSIONS

We compared the total oil production predicted by the different methods with the ‘true’ production given by running a reservoir simulator on the ‘true’ petrophysical fields (Table 2). In the first columns the Eclipse simulator was used on the true reservoir for generating both production data and predictions, while the More simulator was used by the GA and AGA algorithms and Athos was used by the Oliver method. In the middle columns Eclipse was replaced by More. In the last columns the ’true’ reservoir was generated from the Gaussian prior and the More simulator was used. The reservoir was picked such that the total production was similar to the other case. The uncertainties are estimated from a limited number of runs. Most of the uncertainties are small, considerably smaller than the expected precision of the simulator. The uncertainties are larger when prediction production from new wells.

Note that it is only minor differences between the AGA and GA algorithms and between using Eclipse and More on the ’true’ reservoir. The slightly lower values in the GA algorithm compared with the AGA are believed to mainly be due to the mutation swapping in the GA algorithm that

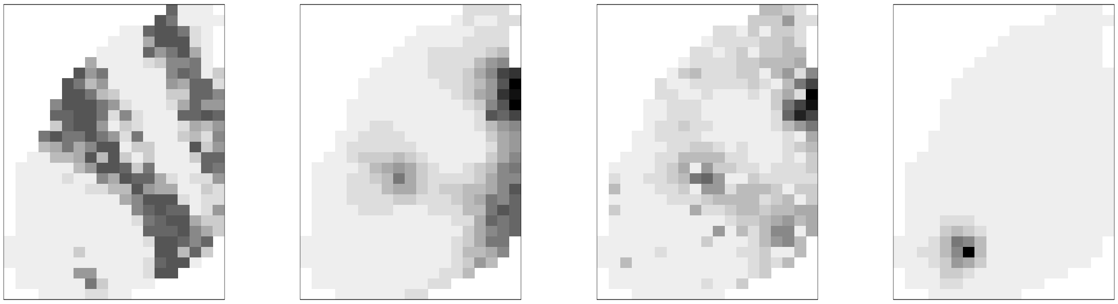

Figure 1: Horizontal permeability in layer 1 for the 'true' reservoir (left) and samples from the AGA, GA and Oliver methods.

reduces effective permeability. The AGA and GA algorithms converged faster when the More simulator was used. For this particular reservoir the Athos simulator seems to give lower values for the production than Eclipse and More, see Floris *et al.* [1].

	S3 Eclipse		S3 More		Gaussian reservoir	
	6 wells	11 wells	6 wells	11 wells	6 wells	11 wells
True	3.87	5.33	3.85	5.15	3.85	4.81
AGA	3.87 ± 0.02	4.85 ± 0.04	3.88 ± 0.02	4.96 ± 0.10	3.87 ± 0.004	4.71 ± 0.05
GA	3.81 ± 0.04	4.69 ± 0.09	3.82 ± 0.03	4.75 ± 0.10	3.78 ± 0.05	4.57 ± 0.14
Oliver	3.57 ± 0.11	4.59 ± 0.17	N.A.	N.A.	N.A.	N.A.

Table 2: Comparison of predicted oil production. Total oil production (10^3 m^3 of oil) with $1\,\sigma$ uncertainties.

When using the same set of wells as in the matching period, both the GA and the AGA method gave predictions agreeing with the true value within the uncertainty. However, when new wells were added, the prediction fell significantly below the true value for the non-Gaussian 'true' reservoir. The mismatch can be explained by several factors. The most important factor is the Gaussian prior. We now know that the 'true' reservoir has non-Gaussian distributions of permeability and porosity with a top at high-permeable cells, see Figure 1. This distribution was impossible to match with our Gaussian assumptions. The infill wells were positioned in high permeable areas not penetrated by the other wells. For the Gaussian 'true' reservoir, the predictions are acceptable also with infill wells recognising that predicting production in new wells is a much harder problem.

Figure 1 illustrating layer 1 in realisations from the different methods, shows some of the properties of the methods. The AGA method gives a field in accordance with the Gaussian prior, the GA algorithm has considerable white noise from the mutation swapping and the Oliver method tend to have some extreme values.

ACKNOWLEDGEMENTS

We are grateful to Institut Français du Pétrole and Beicip-Franlab, France, who kindly let us use their code ATHOS for production simulation and optimisation. We thank Håkon Tjelmeland for use of his code for Gaussian fields, and for valuable discussions. Thanks go to Smedvig Technologies for support and for donating a license for the Modular Oil Reservoir Evaluation programme (More). HHS especially wants to thank David Ponting of Smedvig Technologies for his kind assistance in setting up the model files for More. The software for this work used the GAlib [8] genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology. The European Commission is acknowledged for financial support through the Punq project in the Joule-III non-nuclear energy programme.

REFERENCES

1. FLORIS, F.J.T., BUSH, M.D., CUYPERS, M., ROGGERO, F. and SYVERSVEEN, A.R. – Comparison of production forecast uncertainty quantification methods: an integrated study, *work in progress*, (1999).

2. OLIVER, D.S., HE, N. and REYNOLDS, A.C. – Conditioning Permeability Fields to Pressure Data, *Proceedings of the Fifth European Conference on the Mathematics of Oil Recovery*, eds. HEINEMANN, Z.E. and KRIEBERNEGG, M., Mining University Leoben (1996), pp. 259–269.

3. OMRE, H., TJELMELAND, H. and HEGSTAD, B.K. – Punq Project 1997. Summary of contribution from NTNU, Trondheim, Norway (1997), internal project report.

4. TJELMELAND, H. – A note on the Bayesian approach to history matching of reservoir characteristics, *Proceedings of IAMG'97: The Third Annual Conference of the International Association for Mathematical Geology*, ed. PAWLOWSKY-GLAHN, V., International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain (1997), vol. 2, pp. 772–777.

5. OMRE, H., TJELMELAND, H. and WIST, H.T. – Uncertainty in history matching—model specifications and sampling algorithms, *Tech. Rep. Statistics 6/99*, Deparment of Mathematics, Norwegian University of Science and Technology, Trondheim, Norway (1999).

6. SOLENG, H.H. – Oil reservoir production forecasting with uncertainty estimation using genetic algorithms, *Proceedings of the 1999 Congress on Evolutionary Computing* (1999), accepted for publication.

7. HOLDEN, L. – Adaptive chains, *Tech. Rep. SAND 11/98*, Norwegian Computing Center, Oslo, Norway (1998).

8. WALL, M. – *GAlib: A C++ Library of Genetic Algorithm Components*, MIT, Mechanical Engineering Department, Massachusetts Institute of Technology (Aug. 1996), http://lancet.mit.edu/ga.