Petroleum Science and Technology
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/lpet20

A New Approach for the Development of Fast-analysis Proxies for Petroleum Reservoir Simulation

M. H. Sefat [a], K. Salahshoor [b], M. Jamialahmadi [a] & B. Moradi [c]

[a] Petroleum Department, Petroleum University of Technology, Ahwaz, Iran

[b] Petroleum University of Technology, Tehran, Iran

[c] R&D, Iranian Central Oil Fields Company, Tehran, Iran

Version of record first published: 12 Jul 2012

Petroleum Science and Technology, 30:1920–1930, 2012

ISSN: 1091-6466 print/1532-2459 online
DOI: 10.1080/10916466.2010.512885

Taylor & Francis
Taylor & Francis Group

A New Approach for the Development of Fast-analysis Proxies for Petroleum Reservoir Simulation

M. H. SEFAT,[1] K. SALAHSHOOR,[2] M. JAMIALAHMADI,[1] AND B. MORADI[3]

[1]Petroleum Department, Petroleum University of Technology, Ahwaz, Iran
[2]Petroleum University of Technology, Tehran, Iran
[3]R&D, Iranian Central Oil Fields Company, Tehran, Iran

Abstract *The bottleneck of all processes that are using field-scale numerical simulators is the computationally expensive objective function evaluation. Hence, always a gap exists between simulation runs and real-time processing. In this study, a new approach is presented that uses online-adaptive artificial neural networks to develop proxies that mimic the behavior of the actual reservoir simulator. In this approach, initially Latin hypercube sampling is used and then an intelligent sample selection algorithm is developed to improve the online network prediction. The cited approach improves the surrogate model development in two directions. First, proxies can be used while they are developing and, second, samples are selected intelligently and this reduces computational cost.*

Keywords ANN, artificial intelligence, artificial neural network, computational cost, Latin hypercube sampling, proxy model, reservoir simulation, surrogate model

1. Introduction

Nowadays, reservoir simulation has become the main source of information in all phases of field development. Reservoir models are used for sensitivity analysis, risk analysis, history matching, field development planning, and production optimization. The bottleneck of all these processes, which use numerical simulation, is the computationally expensive objective function evaluation, which, in fact, is a field-scale numerical simulator. The run time increases with increasing the size and complexity of the reservoir models. Recent progress in computational hardware and software developments help to a certain degree but they cannot close the gap existing between simulation runs and real-time processing (Mohaghegh et al., 2006). Therefore, engineers are still looking for a way to reduce the computational load related to simulation studies. For this reason application of computationally efficient proxy-models has gained a lot of attention.

Proxy can simplify models with lower confidence levels in some outputs and act as alternative for the numerical simulator in several procedures. The terms *response surface model*, *meta-model*, and *surrogate model* are sometimes used as alternatives to proxy-model. However, proxy-model seems to be more accepted in the petroleum industry

Address correspondence to M. H. Sefat, Golestan Alley, Aref Street, Baharestan Avenue, Isfahan, P.O. Box: 8196896315, Iran. E-mail: Morteza.haghighat@gmail.com

and will be used in this paper (Zubarev, 2009). Typical application areas in reservoir simulation include

- Sensitivity analysis of uncertainty variables,
- Probabilistic forecasting and risk analysis (see Yeten et al., 2005),
- Conditioning of a simulation model to historically observed data (history matching; see Cullick et al., 2006; Junker et al., 2006; Slotte and Smorgrav, 2008), and
- Field development planning and production optimization (see Pan and Horne, 1998; Guyaguler et al., 2000; Badru and Kabir, 2003; Ozdogan et al., 2005; Yeten, 2007).

This work presents a new approach for developing fast-analysis proxies. Online-adaptive artificial neural networks (ANNs) are used while Latin hypercube sampling (LHS) and an intelligent sample selection algorithm search and locate regions with poor prediction quality of the ANN and generate new samples in those regions. This approach improves the surrogate model development in two directions:

1. Online-adaptive ANNs are used with incremental training: These ANNs can be used while they are developing and error generally decreases as number of sample points increases.
2. Incorporating intelligent experimental design criteria: This intelligent system generates valuable data in regions with poor prediction quality of the ANN. As reservoir simulator should be run for each sample point, this leading to reduction of computational cost.

Finally, we tried to compare this new approach with conventional method of proxy development. The effectiveness and efficiency of the proposed approach is demonstrated using a synthetic model.

2. Theoretical Background

2.1. Latin Hypercube Sampling

In the context of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. A Latin hypercube is the generalization of this concept to an arbitrary number of dimensions. When sampling is a function of N variables, the range of each variable is divided into M equally probable intervals. Then, M sample points are generated and placed to satisfy the Latin hypercube requirements. This sampling scheme does not require more samples for more dimensions (variables) and this independency is one of its main advantages.

2.2. Surrogate Model

For many real world problems, a single simulation can take many minutes, hours, or even days to complete. As a result, routine tasks such as design optimization, design space exploration, sensitivity analysis and what-if analysis become impossible since they require thousands or even millions of simulation evaluations. One way of alleviating this burden is by constructing a cheap-to-evaluate surrogate model $\hat{F}$ that emulates the expensive response of some black box **F**.

Let **F(x)** to be some continuous quality, cost or performance metric of a product or process defined by a k-vector of design variables $x \in D \in R^K$. In what follows, **D** is referred as the design space or design domain. Beyond the assumption of continuity, the only insight that can be gained from **F** is through discrete observations or samples indicating as follows:

$$x^i \rightarrow y^i = F(x^i)|i = 1, 2, \ldots, n \tag{1}$$

These data are expensive to obtain and, therefore, must be used sparingly. The objective is to use this sparse set of samples to construct an approximate $\hat{F}$, which can then be used to make a cheap performance prediction for any design for $x \in D$ (Forrester et al., 2008).

Two different applications of surrogate models can be considered namely, design optimization and design space approximation. In surrogate model–based optimization, an initial surrogate is constructed using some of the available budget of expensive experiments and/or simulations. The remaining experiments/simulations are run for designs where the surrogate model prediction may have promising performance. In design space approximation, the surrogate is tuned to mimic the underlying model as closely as needed over the complete design space. Optimization can still take place as a postprocessing step (Queipo et al., 2005).

The scientific challenge of surrogate modeling is the generation of a surrogate that is as accurate as possible, using the least possible simulation evaluations. The accuracy of the surrogate depends on the number and location of samples (expensive experiments or simulations) in the design space. In reservoir simulation, we are dealing with highly nonlinear output. Therefore, input dataset with experiments uniformly distributed over the uncertainty domain may not be sufficient for construction of an adequate proxy-model. To overcome this limitation, several techniques have been proposed (see Jones et al., 1998; Wang, 2003; Li and Friedmann, 2005).

The most popular surrogate models are based on polynomial response surfaces, Kriging, support vector machines, and ANNs. In this paper, surrogate models (proxies) are developed using online-adaptive ANNs to approximate the design space with more efficiency. These proxies can be used as a substitute for actual reservoir simulator in optimization, sensitivity analysis, and so on.

3. Methodology

In this work, it we did not try to develop a general and multioutput proxy to mimic all behavior of the actual reservoir simulator. Instead of this complex and nontrivial work, one proxy is developed for every output variable of interest. For example, in an optimization or approximation problem in which cumulative oil production and cumulative water production are objective functions, two proxies are developed. This makes training and prediction of the ANN easier. For developing each proxy, the following steps were considered (see Figure 1).

3.1. Define Input Variables

Sometimes, the input variables are obvious, for example, in optimization of injection rate in a water injection process. The input variables are the injection rates of injectors and all other variables are fixed. But for problems with lots of undefined inputs, sensitivity analysis is needed in order to select some important variables. Variable range should be defined because this will improve the training of the ANN and makes the intelligent sample selection possible.

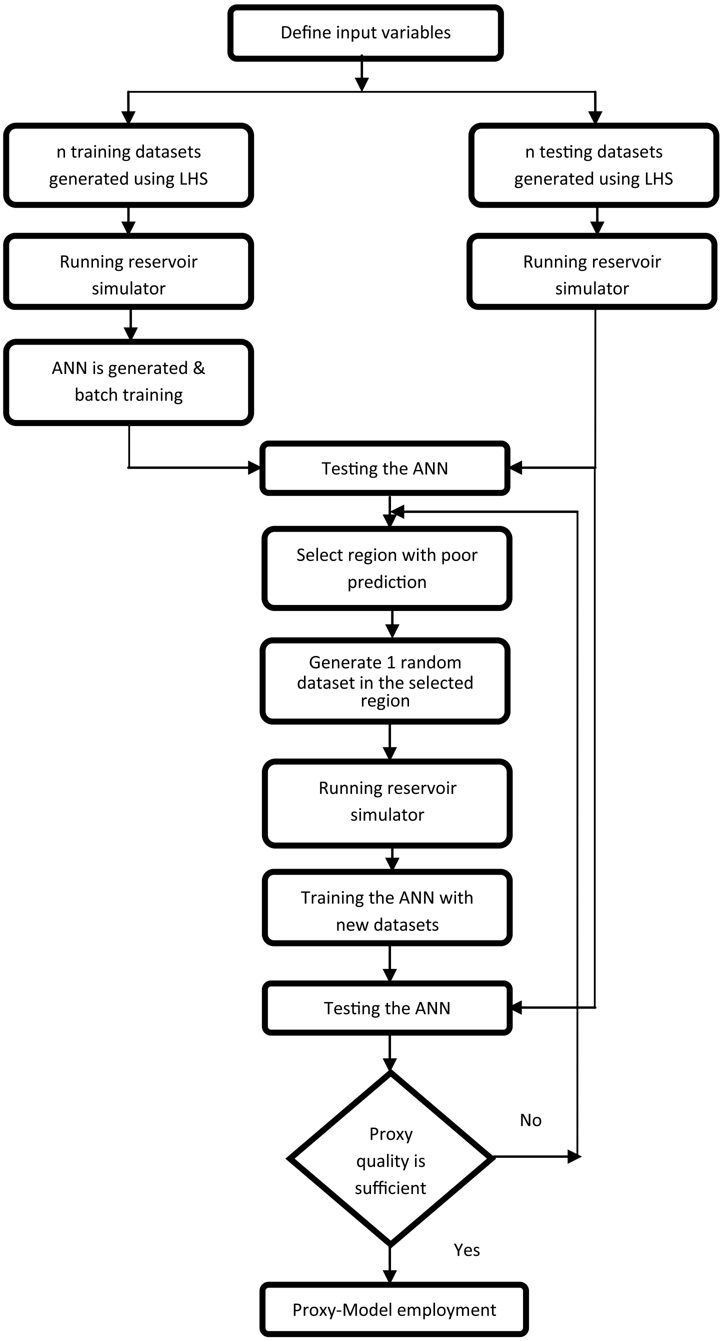

Figure 1. Steps for developing a proxy.

3.2. Generating Training and Testing Data Sets Using LHS

In this step, LHS is used to generate some initial datasets for training and testing the ANN. As they are only used for initial training, it is not necessary to generate large number of datasets.

3.3. Running Reservoir Simulator

All data generation steps only generate input data, therefore, a function is written to evaluate reservoir simulator for these inputs and calculating the output variable of interest.

3.4. Generating ANN and Batch Training

A multilayer perceptron (MLP) network is generated (Demuth and Beale, 2002). The already generated training datasets are used for batch training of the ANN.

3.5. Testing the ANN

The testing datasets, generated from LHS, are used for testing the ANN. The error for each testing datum is calculated, this making possible to find the dataset with the highest value of error.

3.6. Select Region With Poor Prediction

Using LHS, the design domain is divided into several subdomains. In the previous step, the dataset with the highest value of error was found, hence making possible to find its corresponding subdomain. Naturally, this will indicate the poor prediction region.

3.7. Training the ANN With New Datasets

In this step, the new dataset (generated randomly in the poor prediction region) is added to the previous training dataset to be used for updating the ANN. Batch training is used, because of its better performance and enough time for training.

3.8. Testing Proxy Quality

The testing datasets that have already been generated from LHS are used for testing the developed ANN. The mean value of error is treated as a measure to be calculated and compared with some user-defined value. If the proxy quality is sufficient it will be ready to be used, otherwise, the algorithm will be repeated to enhance the proxy quality (see Figure 1).

4. Application to a Synthetic Model

In order to test the method, a synthetic reservoir model is simulated using Eclipse-100 reservoir simulator. The schematic diagram of reservoir model is shown in Figure 2 and

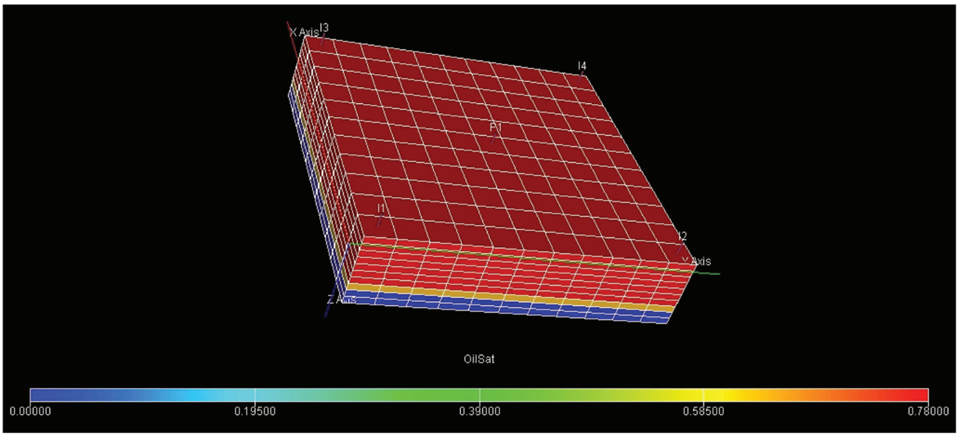

Figure 2. 3-D reservoir model. (color figure available online)

the properties of the reservoir are summarized in Table 1. This model includes four water injection wells and one production well (five-spot pattern). The reservoir is heterogeneous, hence, finding optimum injection rate of each well to maximize oil production with minimum water cut in a 10-year period is our objective in optimization. The total oil production and water cut are required as functions of injection rates to calculate the objective function. As we have two outputs, two proxies should be developed for this problem. In order to compare the proposed method with conventional method, two cases are considered with the same ANN structure.

4.1. Structure of the ANN

For modeling purpose, a fully connected two layer feedforward MLP network with four inputs, one hidden layer with hyperbolic tangent activation function (**f**) with 20

Table 1
Reservoir properties

Property	Value	Property	Value
X dimension	11	Porosity, fraction	0.25
Y dimension	11	Permeability Ave., mD	50
Z dimension	10	No. of sealing fault	2
Dx, ft	400	Swi	0.22
Dy, ft	400	Rock compressibility	4e-5
Dz, ft	100	OWC depth, ft	6,800
Top, ft	6,000	Oil density, lb/ft^3	45
Well bore diameter, ft	0.5	Water density, lb/ft^3	63.02
Injection well completion, ft	6,800–7,000	Prod well completion, ft	6,500–6,700
Active phases	Dead oil, water		

hidden neurons, and one output layer with linear activation functions (**F**) with one output neuron is used. Mathematically, the MLP network can be presented by the following functionality:

$$\hat{y}_i(w, W) = F_i\left(\sum_{j=1}^{q} W_{ij} h_j(w) + W_{i0}\right)$$
$$= F_i\left(\sum_{j=0}^{q} W_{ij} f_j\left(\sum_{l=1}^{m} w_{jl} z_l + w_{j0}\right) + W_{i0}\right) \tag{2}$$

The weights (**w** and **W** or $\boldsymbol{\theta}$) are the adjustable parameters of the network, being determined from a set of examples through the process called training. The examples, or the training data, include a set of inputs, **u(t)**, and their corresponding desired outputs, **y(t)**. The training set is specified as

$$Z^N = \{[u(t), y(t)] | t = 1, \ldots, N\} \tag{3}$$

The objective of training is then to determine a mapping from the set of training data to the set of possible weights $\mathbf{Z^N} \rightarrow \boldsymbol{\theta}$ so that the network will produce predictions $\hat{y}(t)$ that are as close as possible to the true output values of **y(t)**. The prediction error approach based on a measure of closeness in terms of a mean square error criterion is the strategy being utilized here:

$$V_N(\theta, Z^N) = \frac{1}{2N}\sum_{t=1}^{N}[y(t) - \hat{y}(t|\theta)]^T [y(t) - \hat{y}(t|\theta)] \tag{4}$$

The weights are then found as

$$\hat{\theta} = \arg_\theta \min V_N(\theta, Z^N) \tag{5}$$

The iterative steepest decent method is used to minimize the error defined by Eq. (4). This scheme can be expressed as:

$$\theta^{(i+1)} = \theta^{(i)} + \mu^{(i)} f^{(i)} \tag{6}$$

where $\boldsymbol{\theta}^{(\mathbf{i})}$ specifies the ith iterate number, $\mathbf{f}^{(\mathbf{i})}$ is the search direction and $\boldsymbol{\mu}^{(\mathbf{i})}$ is the step size.

A large number of training algorithms exists, each of which is characterized by the way in which search direction and step size are selected. Levenberg-Marquardt method is used in this work.

5. Results and Discussion

To illustrate the performance of the proposed method two cases are presented here.

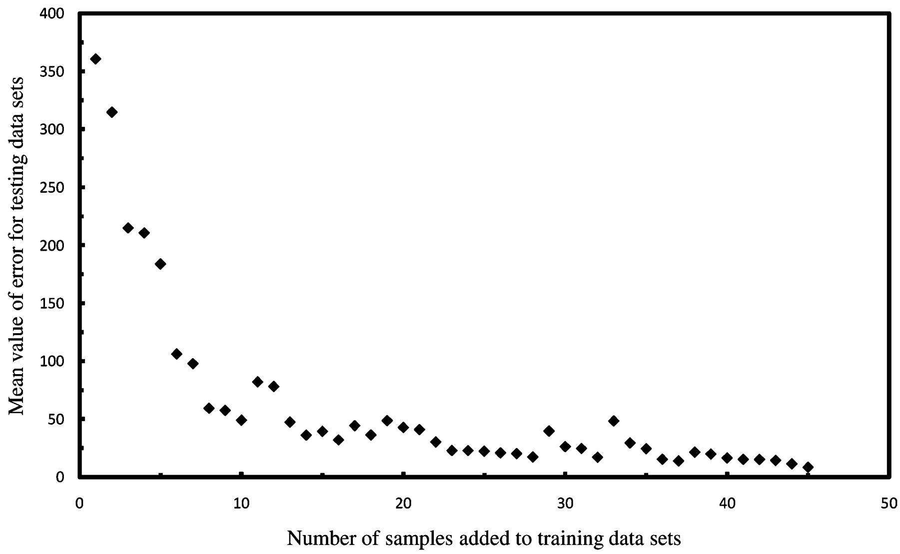

Figure 3. Case 1: The mean value of error for testing dataset versus number of samples added to training dataset.

5.1. Case 1: Online-adaptive ANN Scheme Integrating LHS and Intelligent Sample Selection

The injection rates of four injection wells are variables and total field oil production is the objective function. It was assumed that the injection rate varies from 1,000 stb/day to 1,000,000 stb/day for each injection well and the reservoir was simulated for 10 years of production. Twenty datasets were generated using two LHS (i.e., 10 for training and 10 for testing the ANN). The network was trained by 10 training datasets and then was tested using 10 testing datasets. The calculated mean value of error for testing datasets was about 360% in this test step.

After this initial training, the intelligent sample selection algorithm starts to work. New samples are generated in regions with the highest value of error to be added to the training datasets. Consequently, the value of error is decreased as new samples are added. After each step, the mean value of error is calculated. A graph of the mean value of error for testing dataset versus number of samples added to the training dataset is shown in Figure 3. Results illustrate that the prediction error of about 8% is achieved in this method with 55 training datasets.

5.2. Case 2: Offline ANN Scheme With LHS

In this case, the previous simulation study is repeated under the same conditions with the offline ANN scheme with LHS. A total of 65 datasets were generated using two LHS (i.e., 55 for training and 10 for testing the ANN) and the reservoir was simulated for each of these 65 datasets.

In this case the network is trained offline by 55 training datasets and is tested using 10 testing datasets. In order to remove the effect of randomness in training, the ANN is trained 20 times and after each iteration the mean value of error is calculated for testing

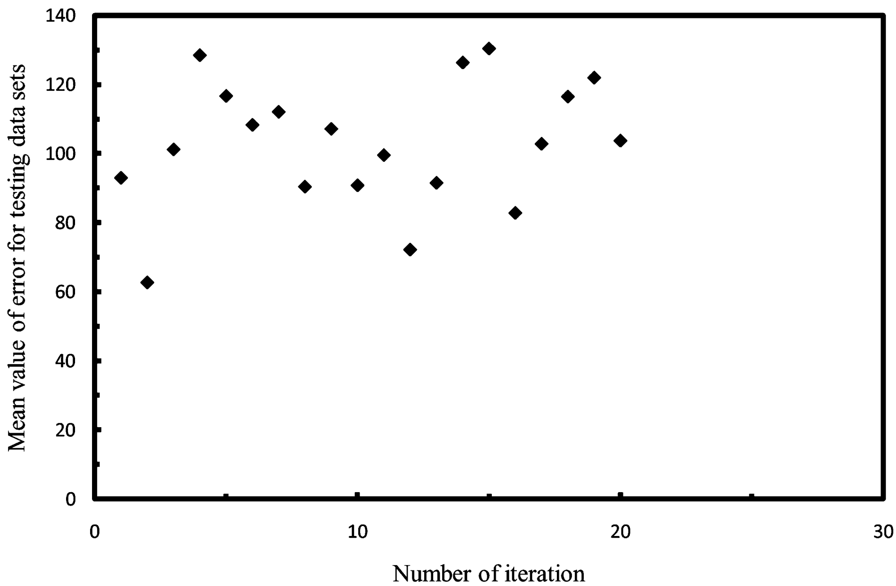

Figure 4. Case 2: The mean value of error for testing datasets for 20 iterations.

datasets. The results are summarized in Figure 4. As shown the average error of about 102% is obtained in this method.

The performances of the two schemes are compared in Figure 5, where the mean average errors are presented as a function of number of samples added to the training datasets. The results illustrate that the prediction of the two methods differ greatly. In case 1, the error is decreased and gets asymptote to a value about 8% after 45 runs of

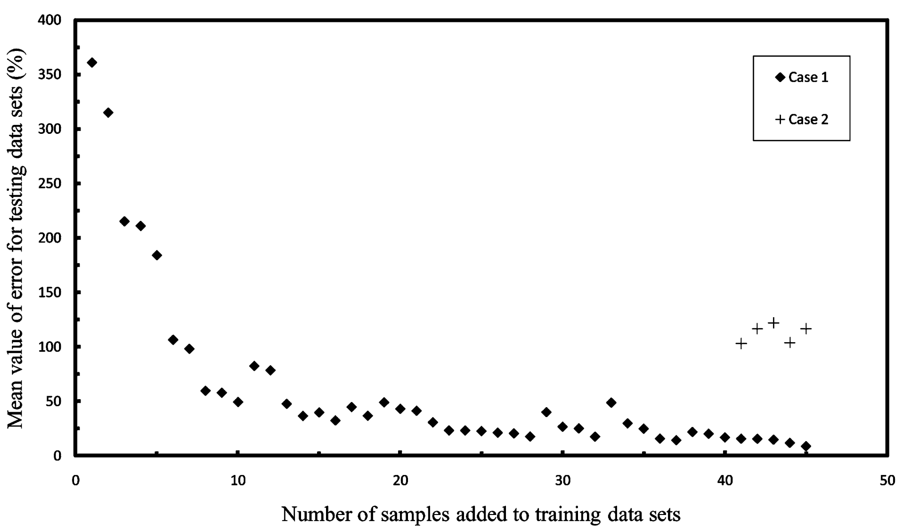

Figure 5. Comparison between the mean value of error for testing datasets for cases 1 and 2 with same computation power.

the reservoir simulator while in case 2 the error is almost remained constant on 100% after 55 runs of the reservoir simulator. This clearly demonstrates that the proposed approach is able to improve the simulation results and yet decreases the computational time considerably.

6. Conclusions

A new method has been developed for fast analysis of oil reservoirs. The proposed method has been comparatively tested for five spots water injection pattern with conventional offline ANN scheme with LHS using the same trained ANN. It is clearly observed that the proposed scheme improves the results and yet decreases the computational time considerably. However, with the same computational power, the resulting prediction error for the conventional offline ANN scheme with LHS becomes very high and remains constant if the training of the network is performed offline without any supervision on samples. Furthermore, in the conventional offline training approach, the network can only be used after all the simulations are executed, while the proposed approach allows the application of the model at every recursive training stage.

References

Badru, O., and Kabir, S. (2003). Well placement optimization in field development. SPE 84191, *SPE Annual Technical Conference and Exhibition*, October 5–8, Denver, Colorado.

Cullick, A. S., Johnson, D., and Shi, G. (2006). Improved and more-rapid history matching with a nonlinear proxy and global optimization. SPE 101993, *SPE Annual Technical Conference and Exhibition*, September 24–27, San Antonio, Texas.

Demuth, H., and Beale, M. (2002). *Neural Network Toolbox: User's Guide*. Natick, MA: The MathWorks.

Forrester, A., Sóbester, A., and Keane, A. (2008). *Engineering Design via Surrogate Modeling: A Practical Guide*. London, England: Wiley.

Guyaguler, B., Horne, R. N., Rogers, L., and Rosenzweig, J. J. (2000). Optimization of well placement in a Gulf of Mexico waterflooding project. *SPE Res. Eval. Eng.* 5:229–236.

Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *J. Glob. Optimiz.* 13:455–492.

Junker, H. J., Plas, L., Dose, T., Dea, A. G., and Little, A. J. (2006). Modern approach to estimation of uncertainty with dynamic reservoir simulation. SPE 103340, *SPE Annual Technical Conference and Exhibition*, September 24–27, San Antonio, Texas.

Li, B., and Friedmann, F. (2005). *Novel multiple resolution design of experiment/response surface methodology for uncertainty analysis of reservoir simulation*. SPE 92853.

Mohaghegh, S. D., Modavi, A., and Hafez, S. (2006). Development of surrogate reservoir models (SRM) for fast-track analysis of complex reservoirs. SPE 99667, *SPE Intelligent Energy Conference and Exhibition*, April 11–13, Amsterdam, the Netherlands.

Ozdogan, U., Sahni, A., and Yeten, B. (2005). Efficient assessment and optimization of a deepwater asset development using fixed pattern approach. SPE 95792, *SPE Annual Technical Conference and Exhibition*, October 9–12, Dallas, Texas.

Pan, Y., and Horne, R. N. (1998). Improved methods for multivariate optimization of field development scheduling and well placement design. SPE 49055, *SPE Annual Technical Conference and Exhibition*, September 27–30, New Orleans, Louisiana.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., and Vaidyanathan, R. (2005). Surrogate-based analysis and optimization. *Prog. Aerospace Sci. J.* 41:1–28.

Slotte, P. A., and Smorgrav, E. (2008). Response surface methodology approach for history matching and uncertainty assessment of reservoir simulation models. SPE 113390, *Europec/EAGE Conference and Exhibition*, June 9–12, Rome, Italy.

Wang, G. G. (2003). Adaptive response surface method using inherited Latin hypercube design points. *J. Mech. Des.* 125:210–220.

Yeten, B. (2007). Optimization of field development. *9th International Forum on Reservoir Simulation*, Abu Dhabi, UAE.

Yeten, B., Castellini, A., Guyaguler, B., and Chen, W. H. (2005). A comparison study on experimental design and response surface methodologies. SPE 93347, *SPE Reservoir Simulation Symposium*, January 31–February 2, The Woodlands, Texas.

Zubarev, D. I. (2009). Pros and cons of applying proxy-models as a substitute for full reservoir simulations. SPE 124815, *SPE Annual Technical Conference and Exhibition*, October 4–7, New Orleans, Louisiana.