Neural Network Meta-Modeling of Steam Assisted Gravity Drainage Oil Recovery Processes

Alali, Najeh; Pishvaie, Mahmoud Reza*[+]; Taghikhani, Vahid
Faculty of Chemical & Petroleum Engineering, Sharif University of Technology, Tehran, I.R. IRAN

ABSTRACT: *Production of highly viscous tar sand bitumen using Steam Assisted Gravity Drainage (SAGD) with a pair of horizontal wells has advantages over conventional steam flooding. This paper explores the use of Artificial Neural Networks (ANNs) as an alternative to the traditional SAGD simulation approach. Feed forward, multi-layered neural network meta-models are trained through the Back-Error-Propagation (BEP) learning algorithm to provide a versatile SAGD forecasting and analysis framework. The constructed neural network architectures are capable of estimating the recovery factors of the SAGD production as an enhanced oil recovery method satisfactorily. Rigorous studies regarding the hybrid static-dynamic structure of the proposed network are conducted to avoid the over-fitting phenomena. The feed forward artificial neural network-based simulations are able to capture the underlying relationship between several parameters/operational conditions and rate of bitumen production fairly well, which proves that ANNs are suitable tools for SAGD simulation.*

KEY WORDS: *Artificial Neural Network (ANN), Meta-modeling, Surrogate modeling, Enhanced oil recovery, Steam Assisted Gravity Drainage (SAGD).*

INTRODUCTION

There is considerable interest in effective oil recovery mechanisms for heavy oil and bitumen due to the decline of conventional oil reservoirs and the estimated magnitude of these resources worldwide (approximately 6 trillion barrels of heavy oil in place) [1, 2].

In order to deplete enormous amounts of immobile heavy oil, different alternatives have been proposed in the last three decades. Examples of these alternatives are cyclic steam stimulation (CSS), steam drive, in situ combustion, and Steam Assisted Gravity Drainage (SAGD) [3]. The latter could be effective even in reservoirs containing highly viscous oil or bitumen [4] and have proven to be economically viable in a variety of pilot and commercial recovery projects [5,6]. In the SAGD process, two parallel horizontal wells, one above the other, are utilized where the top well is considered as the steam injector and the bottom as the oil producer. When steam is continually injected into the top well the oil is heated up and forms a steam chamber which grows upward to the surroundings (Fig. 1).

In this process, heat exchange can occur by both conduction and convection mechanism, and also due to

* *To whom correspondence should be addressed.*
+ *E-mail: pishvaie@sharif.edu*
1021-9986/10/3/51 *15/$/3.50*

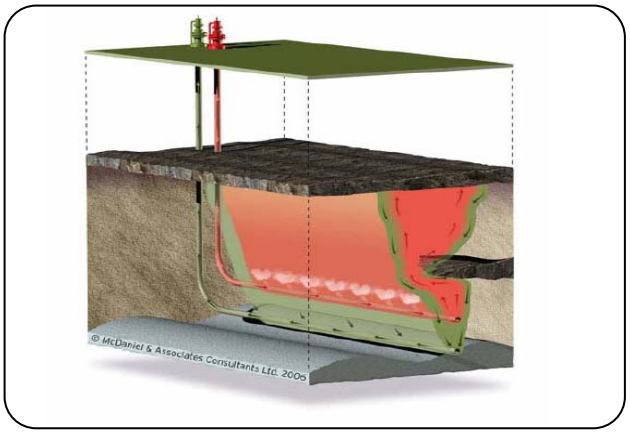

Fig. 1: SAGD principle, (www.mcdan.com/Images/SAGDInset.jpg).

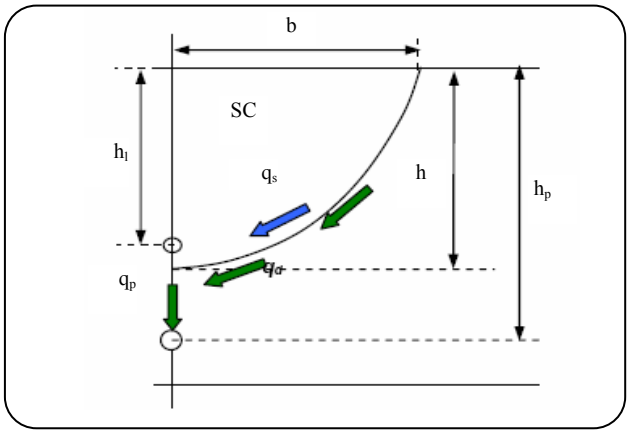

Fig. 2: Schematic diagram for the SAGD process. (SC: Steam Chamber).

the steam condensation. The SAGD process presents a significant advantage over compared to conventional continuous steam injection process. In continuous steam injection, oil is pushed to an area where its mobility is low, while in the SAGD process, oil is drained with a flow approximately parallel to the steam chamber, arriving at the producer well still warm and with high mobility. Fig. 2 provides a visual description of the process.

The performance of SAGD process can be significantly affected by the selection of the operational and geometric reservoir parameters [7,8], for instance, horizontal and vertical rock permeability, reservoir heterogeneity, oil reservoir thickness, and operational conditions such as, pre-heating policy, distance between wells, wells' length and steam rate, to name a few.

Obviously, the rigorous optimization and forecasting studies of SAGD processes is a complex task. The complexity is associated with being very time consuming, a potentially high number of parameters, and a nonlinear distributed parameter solution space. However, a major shortcoming of simulations based on first-principle modeling is the need for expert assistance any time a change is required in a model. Another drawback is encountering high computational load when conducting simulation runs through rigorous numerical simulators. Therefore, accurate, high-fidelity models are typically time-consuming and computationally expensive. In this context, the so-called metamodeling approach for analysis and optimization can play a very valuable role. A meta-model, response surface, and/or 'fast surrogate' model is an approximation of the input/output function implied by the underlying simulation model. It is usually a supplementary model that can be alternatively used to interpret a more detailed model. There are several techniques to provide a meta-model such as artificial neural networks. These connectionist models can act as a surrogate to the original rigorous model with relatively accurate and fast performance.

The meta- or surrogate modeling, required by simulation activities is very important and provides general guidelines for the development of ANN-based simulation metamodels [9]. Such guidelines were successfully applied in the development of many complex systems such as; estimating the Manufacturing Lead Times (MLT), dispatching system (planning system of transport routes), multicommodity network and bank system with several cash registers [9-11].

Amongst the others, *Queipo* presented a solution methodology called Neural Network-based Efficient Global Optimization (NEGO) for optimization of the geometrical and operational parameters in a SAGD process [22]. The solution methodology includes the construction of a ''fast surrogate'' of an objective function whose evaluation involves the execution of a time-consuming mathematical model (i.e. reservoir numerical simulator) based on neural networks. The parameters involved are only vertical spacing, injection pressure, steam-injected enthalpy, and sub-cooling.

In summary, the term 'simulation meta-model' refers to a simplified representation of a first-principle based simulation model, designed to approximate selected input-output mappings, whilst the term 'surrogate model' stresses on fast calculation of outputs, as required

essentially in intermediate computations of optimization and/or forecasting studies.

In this paper, all effective factors have been studied and their results for the first 10 years of production are considered as input parameters in the neural network.

The main objectives which are followed in this study are survey of parameters which are effective on primitive production in SAGD process, execution of sensitivity analysis of parameters relating to reservoir, for developing a meta-model neural network. All case studies are performed using the simulator 'STARS' from CMG (Computer Modeling Group, version 2006.10).

Methodology of Surrogate-based analysis and modeling

Computational modeling is a complex and lengthy process that begins with the examination and a set of scientific observations and the formation of explanatory hypothesis (either in deterministic or statistical sense), and ends with executable computer code designed to confirm or disprove these hypothesis [23].

The formalism for a conventional simulation includes a four-step methodology: 1) selection of system or control volume, 2) deriving the suitable governing equations, 3) selection of a solution technique to perform the simulation, and 4) computer coding to implement the algorithm of modeling/simulation. However, when the system is highly complex, the major computational load is due to step 3 which in turn is affected by step 2, i.e.; the formulation step. An alternative to the conventional and mathematical, rigorous or the so-called first-principle law formulation is surrogate- or meta-modeling. In other words, we may use black-box or grey-box neural network meta-modelers to surrogate the rigorous formulation instead. Obviously, the scheme differs somehow in formulation and model selection.

Preliminary study and sensitivity analysis of rigorous modeling of the desired system, i.e., SAGD process, construction of the surrogate model, i.e., the neural net meta-modeler structure, and training and/or model validation are three steps for the methodology of surrogated based modeling selected in this work.

The SAGD process rigorous model simulation

Base case

Specification of one of the reservoirs of Alberta State of Canada has been considered as a base case. Operational characteristics and well completion are changing in a way that one can create several models, obtain their results and compare them using a simulator. The base case is composed of a cube reservoir by the area of 10200 m^2 and 50 m thickness. Other specifications of the reservoir, liquid and rock are given in Table 1. The schematic plan of the base case is also shown in Fig. 3. Two horizontal wells by length of 850 m and radius of 8.6 cm are drilled in 235 and 247 m depths. From the upper well, steam is injected and from the lower one, oil is produced.

Table 1: Base case properties.

Variables	Values
Oil Gravity, API°	10.57
Reservoir coordinate in the X direction, m	120
Reservoir coordinate in the Y direction, m	850
Reservoir thickness, m	50
S_{gc}, %	5
S_{wi}, %	20
S_{or}, %	15
, mdk_x	3400
, mdk_y	800
A, m^2	102000
P_i, psi	2100
h, m	50
r_w, m	0.0875
L_p, m	850
Reservoir type	Heavy Oil Conventional Reservoir
S	0

The curves of PVT, permeability ratio and other properties such as water volume factor (B_w), density and viscosity of water phase, density, viscosity and gas solution for oil phase and also permeability ratio of water and oil phases have been provided and introduced to the simulator.

STARS simulator

STARS is a module belonging to the CMG software

Table 2: Different gridding properties.

Local Grid Refinement (LGR)	Block Number	Gridding Type
No	23250	Non Uniform Fine Grid
Yes	23250	Non Uniform Fine-Hybrid Grid
Yes	7250	Non Uniform Medium-Hybrid Grid
No	750	Uniform Coarse Grid
No	21700	Uniform Fine Grid
Yes	21700	Uniform Fine-Hybrid Grid
Yes	7500	Uniform Medium-Hybrid Grid

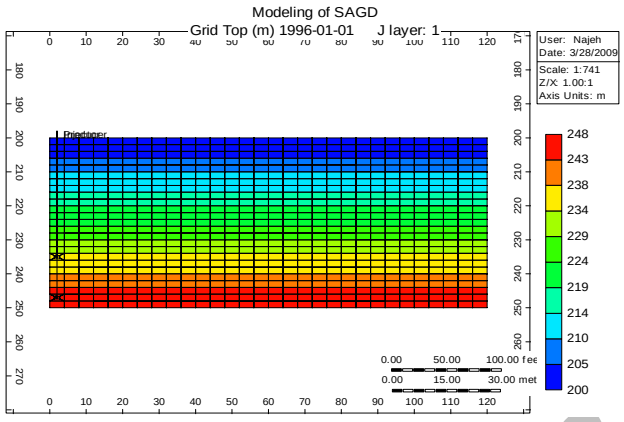

Fig. 3: Base case model (cross view).

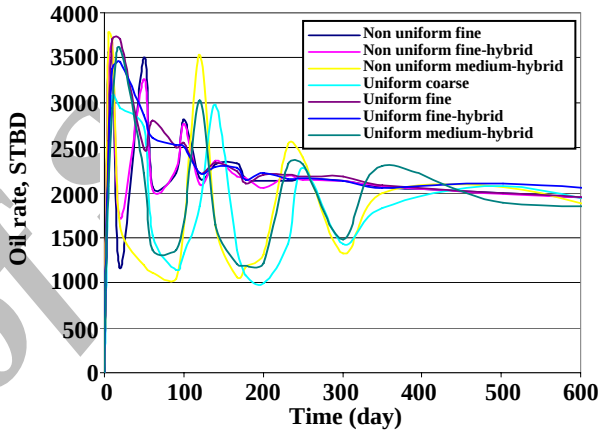

Fig. 4: Effect of different gridings over oil production rate.

simulators which simulates the operations regarding steam injection, thermal processes and complicated processes of enhanced oil recovery. STARS can be used for different kinds of liquid properties (PVT region), rock type and curves of permeability ratio related to each region in a reservoir.

Regardless of size and complexity of the reservoir under the study, STARS is a suitable tools for performing studies on the reservoir.

Optimal gridding

Prior to running several simulations, it is necessary to select the proper block sizes of problem domain. Seven cases with different gridding schemes were considered to select the model accuracy and computational load. The explanations of these models including the number of grid blocks are given in Table 2.

When *Peaceman* [24] brought his method about the relation among well and reservoir grids, he used uniform gridding. But as we know the use of non uniform gridding can increase the accuracy of calculations without increasing the period of program execution. Further, in simulation of some events (such as conning) and in special conditions in reservoirs (such as layered ones), the use of non uniform gridding is vital and necessary to preserve the stability and accuracy of simulation.

In Fig. 4, the production oil rate during 10 years of simulation is drawn regarding the number of gridding.

As can be observed from Fig. 4, by increasing the number of grids the rate of daily production in all cases except uniform fine grid and uniform fine hybrid in the first 300 days had fluctuations and it indicates the incompatibility of the model with gridding system. Between these two gridding types, uniform fine hybrid grid is preferred. This is the case in simulation of processes like SAGD that wells are horizontally drilled and further, by steam injection in a well, the pressure gradient and high temperature will be set up around it.

The number of grids in this model is 21700 (= 62×14×25) and for the grids where well is completed, are refined by local grid refinement method. Each grid is divided in radial type to two parts, in angular type to 4 parts and in axial type to one part.

The reservoir studied by STARS simulator is a model composed from many patterns which have been previously developed in the field. Therefore, boundary condition requirement for simulation of each pattern is no flow. In simulation of the flow inside the well, the Discretized Wellbore (DW) technique which can model the pressure and temperature drop inside the well has been considered. It makes the results reliable and acceptable.

Preheating

Effective initial heating of the cold oil is important for the formation of the steam chamber in gravity drainage processes [25]. To enhance the slow process of SAGD, an early-time preheating is performed in which steam may be circulated in both wells.

The results of the simulations show that it is possible to improve the initial amount of production of the reservoir through pre-heating step. Generally, cyclic vapor injection has yielded better results compared to vapor circulation in the wells.

Parametric Study

The methodology to set up a meta- or surrogate model initiates with identification of effective and key parameters of the process in a reduced manner. By reduced manner, we mean finding the minimum set of inputs and/or parameters which have the maximum effects on major outputs.

In this section we intend to study the evolving parameters and their effects on the oil production using the SAGD method. The parameters have been categorized into 4 groups, leading to totally 15 items enumerated as below:

Group 1 (reservoir parameters):

1- k_h / k_v ratio,

2- Initial pressure (p_i),

3- Thickness (h) and

4- Drainage area (A)

Group 2 (fluid parameters):

1- API gravity

2- Oil viscosity in the in initial Temperature of the reservoir (μ_{oi}),

3- Injecting vapor temperature (T_s),

4- Steam injection rate (q_{inj}) and

5- Steam quality (x)

Group 3 (reservoir rock-fluid parameters):

1- Critical Gas Saturation (S_{gc}) and

2- Residual oil saturation (S_{or})

Group 4 (well parameters):

1- distance between injecting and producing wells (d_{ip}),

2- well length (L_p),

3- Well radius (r_w),

4- Ratio of injecting well length to producing well (L_p / L_{inj}) and

5- Skin factor (s)

Each of above items was varied and deviated from the base case value, while the others remained constant at their original base case values. Totally 277 simulation sessions were run to study the relative effects of parameters on oil rate and its cumulative production (recovery factor). For all the scenarios, the time zero is considered when the preheating operation has been terminated. In other words, it was done to avoid the consideration of inherent time delay (due to preheating) prior to starting principal SAGD process. Amongst them, the most effective parameter was steam injection rate and the least one, was the API gravity parameter. However, due to space limit, only the corresponding graphs of steam injection rate are depicted in Figs. 5 and 6.

Neural network meta-modeler

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems. One can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. A trained network can perform the intended mapping of input space to output space in almost instantaneously fashion. Therefore, it can act as a meta-modeler instead of running a time-consuming and rigorous simulator or modeler. The neuron model and the architecture of a neural network describe how a network transforms its input into an output. This transformation can be viewed as a computation task.

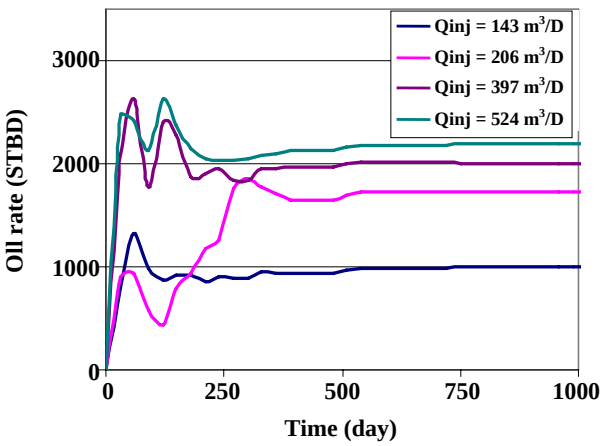

Fig. 5: Influence of Steam Injection Rate on Oil Rate via SAGD.

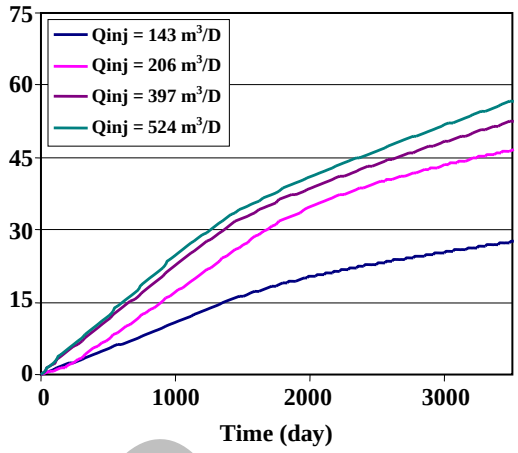

Fig. 6: Influence of Steam Injection Rate on Recovery Factor via SAGD.

To define the problem (SAGD performance) in the context of neuro-computing, a set of input vectors and a set of corresponding target vectors (the correct output vectors for each of the input vectors) should be arranged into an appropriate database. However, the main objective is such that the trained net-modeler should be capable of predicting next month recovery factor for any probable combination of early-mentioned parameters. This has been done instead of running a professional, complex and time-consuming simulator such as STARS.

Neural network topology

There are two basic types of input vectors: those that occur concurrently (at the same time or in no particular time sequence), and those that occur sequentially in time. For concurrent vectors, the order is not important, whereas for sequential vectors, the order in which the vectors appear is important. Concurrent inputs are appropriate for static networks while the sequential inputs are suitable for pure dynamic networks. However, the proposed network is specially designed in a mixed static-dynamic network. Therefore, the input data structure comprises of two parts; 16 parameters of well, reservoir and fluid properties, all taking part as factors or affecting parameters and three sequential producing well recovery factors of last three months, representing the dynamic feature of the SAGD process. This leads totally to a 19-dimensional input vector. The corresponding target vector (outputs) includes the current time instance of recovering factor. In summary, the proposed network maps the 19-dimensional input (16 parameters and the three past values of recovery factors) into a 1-dimensional output (current RF) space, as below:

Input vector: API^o , q_{inj}, T_s, x, μ_{oi}, A, p_i, k_h / k_v, h, S_{gc}, S_{or}, d_{ip}, L_p / L_{inj}, L_p, r_w, s, RF_{k-1}, RF_{k-2} and RF_{k-3}

Output vector: RF_k

Data preparation for training

Back-propagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Standard back-propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term back-propagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

Neural network structure includes 19 inputs, 35 neurons in hidden layer and one-element output. It should be remarked that the selection of 35 neurons is primitive and it will be reduced systematically to 6 neurons, eventually. Activation function in hidden layer neurons is tansig and in output layer is purelin and finally it applies Error Back Propagation algorithm and training method of Levenberg-Marquardt.

Training

Once the network weights and biases are initialized, the network is ready for training. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function.

To train the proposed neural network, we used MATLAB software (NNET toolbox) and after determination of topology as recommended, we prepared the training vector pairs into two matrixes (input matrix and target matrix). Further, we should determine some parameters related to training algorithm.

Improving Generalization

One of the problems that occur during neural network training is called over-fitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations.

One method for improving network generalization is to use a network that is just large enough to provide an adequate fit. The larger network we use, the more complex the functions the network can create. If a small enough network is used, it will not have enough power to over-fit the data. Unfortunately, it is difficult to know beforehand how large a network should be for a specific application. However, there are two other methods for improving generalization that are implemented in NNet Toolbox™ software: early stopping and regularization.

In the early stopping algorithm (the default technique in NNet Toolbox) the available data is divided into three subsets. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to over-fit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations (epochs), the training is stopped.

The test set error is not used during training, but it is used to compare different models. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this might indicate a poor division of the data set. However, according to the hybrid structure (static/dynamic) of the proposed network, it is very difficult to find an appropriate dividing criterion to separate the training, validation and test data set. Therefore, this technique was put aside in this work.

Another method for improving generalization is called regularization. This involves modifying the performance function, which is normally chosen to be the sum of squares of the network errors on the training set.

The typical performance function used for training feed-forward neural networks is the mean sum of squares of the network errors:

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \quad (1)$$

It is possible to improve generalization if the performance function is modified by adding a term that consists of the mean of the sum of squares of the network weights and biases:

$$msereg = \gamma mse + (1-\gamma) msw$$

where γ is the performance ratio, and

$$msw = \frac{1}{n}\sum_{i=1}^{n}(w_i)^2 \quad (2)$$

Using this performance function causes the network to have smaller weights and biases, and this forces the network response to be smoother and less likely to over-fit.

The problem with regularization is that it is difficult to determine the optimum value for the performance ratio parameter. If this parameter is assigned too large, we might get over-fitting. If the ratio is too small, the network does not adequately fit the training data. Therefore, it is desirable to determine the optimal regularization parameters in an automated fashion. One approach to this process is the Bayesian framework of *MacKay* [26]. In this framework, the weights and biases of the network are assumed to be random variables with specified distributions. The regularization parameters are related to the unknown variances associated with these distributions. We can then estimate these parameters using statistical techniques. A detailed discussion of the use of Bayesian regularization, in combination with Levenberg-Marquardt training, can be found in [27].

One feature of this algorithm is that it provides a measure of how many network parameters (weights and biases) are being effectively used by the network.

The algorithm generally works best when the network inputs and targets are scaled so that they fall approximately in the range [-1, 1]. This is the case for the proposed network here.

DISCUSSION

In this research we produced and simulated the data of 277 injecting and producing pair well by CMG software and applied these results as input data to the neural network. The neural network structure and the way we allocated input and target data are mentioned in previous parts. In this part we are involved with analysis and optimization of neural network parameters. The neural network should not be affected by common errors of training like generalization and memorization.

In this research, first a neural network with 19 inputs and 35 neurons in hidden layer and 1 neuron in output layer is considered. Error threshold and epochs number are determined 0.001 and 100 times, respectively. Obtained results confirmed that neural network was well trained and in test stage provided very accurate results. Figs. 7 to 9 show neural network results.

Fig. 7 is a result of neural network training process. It shows that the training is done with a good performance by comparison between train and validation data which are divided before by MATLAB. According to performance result (0.00121388) and predefined value of error tolerance (as 0.001), the neural network is definitely trained well.

Fig. 8 is a comparison diagram of neural network output and real data related to test cases, where the NN checked for test data which is divided before using MATLAB (55 Test cases). This Figure compares the result created by NN (Outputs) and the real data were obtained by CMG (Targets) indicating a good agreement.

Fig. 9 is a comparison of recovery factor obtained from neural network and real one obtained from CMG for one of the 55 test cases. The case shown here corresponds to the results obtained by NN for 120 months (10 years) and to those obtained by CMG. The results of other cases are similar.

By verifying neural network weights, it is found that the 35-neuron network had only considered some

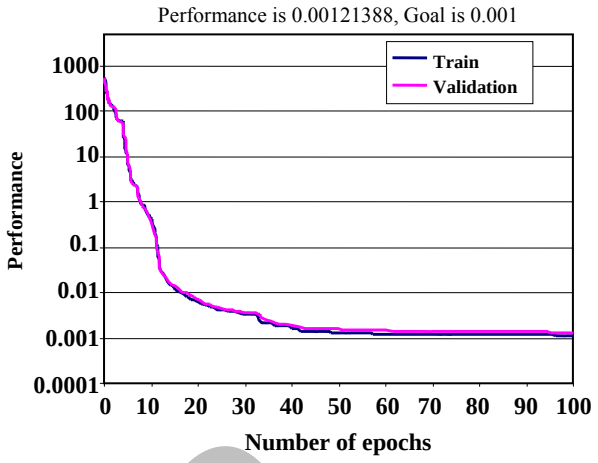

Fig. 7: Neural network training diagram with 35 neurons in hidden layer.

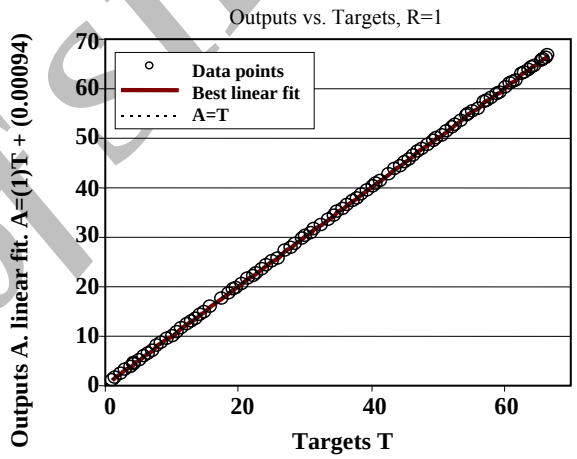

Fig. 8: Comparison diagram of neural network output and real data related to test cases.

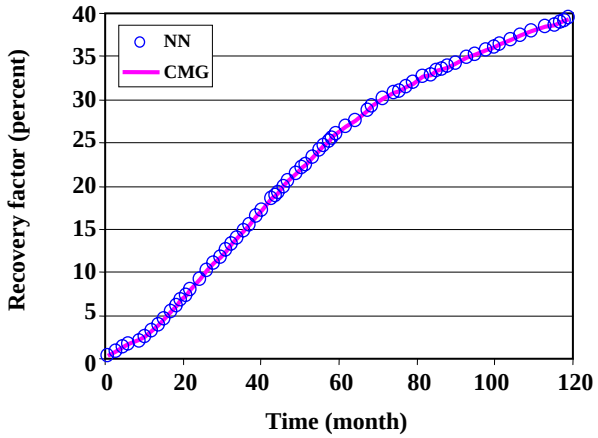

Fig. 9: Comparison of recovery factor obtained from neural network and real one obtained from CMG for one of the 55 test cases.

Table 3: The effect of node numbers in empowering the generalization.

Number of Nodes	Number of Parameters	Num. of effective Effective Parameters	% of efficiency
5	108	83	76.8
6	129	123	95.3
7	150	124	82.6
8	171	144	84.2
9	192	140	72.9
10	213	168	78.8
11	234	170	72.5
12	255	124	48.6
13	276	113	40.9
14	297	224	75.4
15	318	149	46.8
16	339	298	87.9
17	360	213	59.1
18	381	224	58.8
19	402	194	48.2
20	423	227	53.6
21	444	93	20.9
22	465	93	20.0
23	486	103	21.2
24	507	83	16.4
25	528	102	19.3
26	549	106	19.3
27	570	252	44.2
28	591	100	16.9
29	612	95	15.5
30	633	322	50.8
31	654	88	13.4
32	675	114	16.8
33	696	103	14.8
34	717	325	45.3
35	738	147	19.9

particular inputs. In other words, the networks may be involved in memorization error due to overtraining. By reduction of neuron number and also, the epoch number this problem can be solved. Therefore, the results led us toward the study of generalization power and over-fitting mirage. The results of training experiments, equipped by Bayesian regularization for several candidates of hidden layer nodes are given in Table 3. As it is clear, there is an optimum number of hidden layer (16 neurons in the hidden layer) at which the most generalization power occurs. It should be remarked that the *mse* value for all results was around 0.001, approximately.

CONCLUSIONS

In the present work, a new meta- and surrogated model has been designed through a delegated artificial neural network. The sensitivity analysis indicates that SAGD meta-modeler is applicable to the most of heavy oil reservoirs. In the proposed scheme, a new hybrid (static/dynamic) structure of neural network has been

used and also the optimal selection of hidden layer neurons has been selected such that the over-fitting has been avoided. The major work was focused on both designing and training of the proposed NNet-modeler. The training and validation of proposed neural network are deliberately assessed to avoid the common errors like generalization, over-fitting and memorization.

In this work, for designing a neural network based meta-modeler for SAGD processes, some general guidelines have been provided.

As a future work, the proposed scheme can be extended to cope with transient behavior of SAGD processes.

Nomenclature

B_w	Water volume factor, Rbbl/STB
S_{wi}	Initial water saturation, %
k_x	Reservoir permeability in x direction, md
k_y	Reservoir permeability in y direction, md
k_z	Reservoir permeability in z direction, md
d_{ip}	Injector-Producer distance, m
S	Skin factor
S_{or}	Residual oil saturation, %
S_{gc}	Critical gas saturation, %
k_h	Horizontal permeability, md
k_v	Vertical permeability, md
L_{inj}	Injection well length, m
L_p	Production well length, m
A	Drainage area, m^2
h	Reservoir thickness, m
μ_{oi}	Viscosity in the initial temperature, cp
T_s	The temperature of vapor, °C
p_i	Initial pressure, psi
q_{inj}	The rate of injecting vapor, m^3/day
x	Steam quality
r_w	Well radius, m
I	Inputs vector
T	Outputs vector
D	Unnormalized initial data
E	Neural network error

APPENDIX - TEMPERATURE DEPENDENCY OF VISCOSITY AND CONDUCTIVITY

Viscosity

Water Phase

Water phase viscosity tends to be relatively constant at 1 cp, and decreases as far as 0.1 cp at 300°C. The main purpose of allowing entry of water viscosity data is to account for the various brine concentrations encountered in different reservoirs.

The STARS model has the following data entry options:

1. Use internal table of μw versus T, with a possible dependence on salt concentration

which can be significant.

2. Use the correlation μw = a • exp (b/T), where T is in absolute degrees.

3. Enter directly a table of μw versus T.

Gas Phase

Gas phase viscosities usually are much smaller than liquid phase values, and hence will tend to dominate the flow if gas phase is mobile. As a consequence, pressure gradients may be high when gas is immobile, but will certainly be low when gas is flowing. Gas phase viscosities have values around 0.01 cp.

The STARS model has the following gas viscosity options:

1. Correlation

μg = 0.0136 +3.8 *10^(-5)* T

T in deg C; μg depends only on temperature, and not on composition or p. This

gives μg = 0.014 cp at 20°C and μg = 0.025 at 300°C.

This option is very cheap to use and is very often quite sufficient, for the reason

that the compositional effects on μg are small.

2. Correlation

μgi = ai * T^bi

(T in absolute degrees) for each component. The phase viscosity is

$$\mu_g = \frac{\sum_{i=1}^{n_c} w_i \mu_{gi}}{\sum_{i=1}^{n_c} w_i}$$

Where:

$$w_i = y_i \sqrt{M_i}$$

because viscosities μgi for different components are quite similar, and the T dependence is not strong, the first option mentioned above is often sufficient.

3. Same as option (2), but with an additional correction to μg for high pressure.

Oil Phase

In many steam-injection processes the μo-versus-T function will contain much of the nonlinearity in the flow equations. This is because μo can decrease by several orders of magnitude over only a modest temperature increase. Therefore, it is crucial that the temperature dependence of μo be represented adequately. In addition, the effect of soluble gas components on μo can be significant, and so must be accounted for.

The oil phase viscosity is obtained by a logarithmic mixing rule:

$$\ln(\mu_o) = \sum_{i=1}^{n_c} xi \ln(\mu_{oi})$$

For liquids the component value μoi can be measured directly or estimated from correlations or tables. However, for a soluble gas such as methane, a measured value for the liquid phase may be difficult to find.

In this case, as was the case for oil phase density calculations, μoi must be regarded as the contribution of the soluble gas toward the viscosity of the liquid mixture.

Figure below shows that plotting ln(μo) versus mole fraction for various values of gas content will yield a value for the gas component μo2 by extrapolating x2=1.

Example: Suppose a dead oil component has a viscosity of μo1=1000 cp. When some soluble gas is added and mixed thoroughly, the live oil viscosity is found to be μo=300 cp, and the mole fraction is calculated to be x2=0.20. The equation for μo2 is

$$\ln(300) = 0.80\ln(1000) + 0.20 \ln(\mu o2)$$

So:

$$\ln(\mu o2) = 0.1776 \text{ and } \mu o2 = 1.19cp$$

In the case of a soluble gas, μo2 is the viscosity of a hypothetical liquid composed of 100% solution gas. Note that μo2 is NOT the viscosity of solution gas in the gas phase. Note also that the above sample calculation must be done at other temperatures, in order to obtain the dependence of μo2 on T.

The component values μoi may be specified using one of the following two options.

1. Correlation μoi = ai . exp (bi/T), where T is in absolute degrees.

Coefficients can be calculated from two points on the curve. For example, a heavy oil component may have

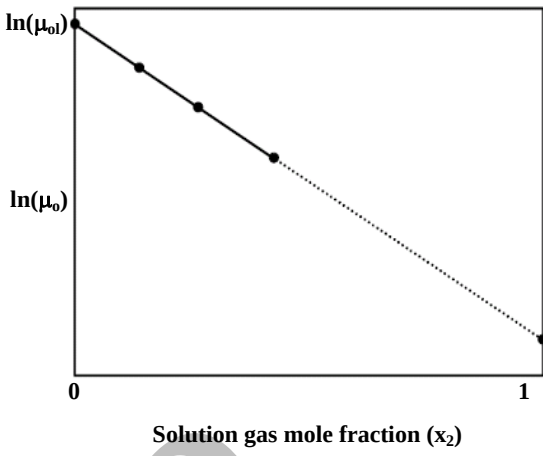

Fig. A1: Calculation of liquid viscosity for solution gas component.

μoi = 10^4 cp at 20°C(= 293K)
μoi = 10cp at 300°C(= 573K)
Just solve
10^4 = ai . exp(bi / 293)
10 = ai . exp(bi / 573)
for ai and bi, which are

$$bi = \frac{\ln(10^4) - \ln(10)}{1/293 - 1/573} = 4141.9 \text{ k}$$

ai= 10^4 / exp(bi / 293)= 7.256* 10^3 cp
Check at the other point:
At T = 300 C, μoi = ai • exp (bii/573) = 1.0 cp.

2. Enter directly a table of μoi versus T. Interpolation between entries is done using the correlation described in option (1) above.

Nonlinear Viscosity Mixing

For situations where the normal mixing rule for phase viscosities does not appear adequate, an approach based on a nonlinear mixing rule for a key component can be employed.

Basically, the linear logarithmic mixing rule

$$\ln \mu = \sum_i x_i \ln \mu_i$$

is replaced by a nonlinear function
xa →f(xa)
for one key component "a". The requirement that the pseudo-composition still to 1 yields a condition on the normalizing factor N:

$$f(x_a) + N\sum_{i \neq a} x_i = 1 \Rightarrow N = \frac{1 - f(x_a)}{1 - x_a}$$

so that the modified logarithmic mixing rule becomes

$$\ln \mu = f\left(x_a\right) \ln \mu_a + \frac{1 - f\left(x_a\right)}{1 - x_a} \sum_{x \neq a} x_i \ln \mu_i$$

If f(xa) is linear then this form reduces to the original mixing rule. The function f(xa) can in principle be quite general such that the nonlinear interval can be in any region between xa = 0.0 and xa = 1.0.

Nonlinear mixing function for key component "a" illustrating the range of values (between x_{al} and x_{au}) over which the nonlinear function is assumed to apply and the method by which the function is input into the simulator.

Fig. A2 shows the general form of this function and illustrates that outside the specified range, a linear function is assumed. This figure also illustrates the manner in which the nonlinear mixing function is entered - namely that the values of this function over 10 indicates equally spaced intervals between x_{al} and x_{au}.

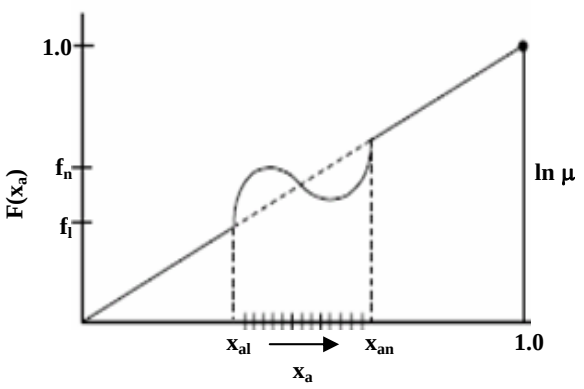

Fig. A2: General functionality of viscosity.

Thermal conductivity:

Thermal conductivity determines the flow term KΔT due to diffusion of energy from a region of high temperature to low temperature. The only other way for energy to flow in situ is by convection. In field-scale steam problems convection usually dominates conduction, at least in the direction of flow. In field-scale combustion, the temperature profile at the fire front can be determined largely by conduction, but this temperature profile is almost never resolved because the grid blocks used are too large. For these reasons, conduction is rarely a major mechanism in field-scale problems. Conduction can play a significant role in both steam and combustion at the laboratory scale, since the length scale is much smaller than in the field.

The following are options for calculating an overall thermal conductivity from phase values. In each the porosity is fluid porosity φ_f.

Linear Mixing

Thermal conductivities are weighted by volume

$$\kappa = \varphi * [Sw\ \kappa w + So\ \kappa o + Sg\ \kappa g\ ] + (1 - \varphi) * \kappa r$$

Nonlinear Mixing

The thermal conductivities are weighted using the correlation of Anand et al. The liquid-rock mixed value is:

$$k_{L-r} = k_L \times a^b$$

where

$$k_L = \left(S_o k_o + S_w k_w\right) / \left(S_o + S_w\right)$$

$$a = k_r / k_L$$

$$b = 0.28 - 0.757 \times \log_{10} \varphi - 0.075 \times \log_{10} a$$

The gas-rock mixed value is:

$$k_{g-r} = k_g \times c^d$$

Where:

$$c = k_r / k_g$$

$$d = 0.28 - 0.757 \times \log_{10} \varphi - 0.057 \times \log_{10} c$$

The gas-liquid-rock mixed value is:

$$k_{g-L-r} = \left(1 - e\right) \times k_{g-r} - e \times k_{L-r}$$

Where:

$$e = \sqrt{S_w + S_o}$$

Temperature Dependence

This modification of Somerton et al accounts for the observed change in thermal conductivity as temperature is increased. In the STARS model this modification may be done after the mixed liquid-gas-rock value has been calculated. The unit of κ is J/m-day-K.

$$k = a - 1.7524 \times 10^{-5} \left(T - T_r\right) \times \left(a - 119616\right) \times b \times c$$

Where:

$$a = k_{g-L-r}$$

$$b = a^{-0.64}$$

$$c = a \times d^e + 110644.8$$

$$d = 1.8 \times 10^{-3} \times T$$

where T is in K, and:

$$e = -3.6784 \times 10^{-6} \times a$$

Received : May 19, 2009 ; Accepted : Mar. 1, 2010

REFERENCES

[1] Godec M.L., "Characterization and Potential of U.S. Heavy Oil." Proceedings of the 5th UNITAR International Conference on Heavy Crude Oil and Tar Sands, Caracas, Venezuela, Aug. 4-9 (1991).

[2] Mahmood S.M., Olsen D.K., Thomas C.P., "Heavy Oil Production from Alaska", Proceedings of the 6th UNITAR International Conference on Heavy Crude Oil and Tar Sands, Houston, TX, Feb. 12-17 (1995).

[3] Butler R.M., Steam Assisted Gravity Drainage: Concept, Development, Performance and Future, *J. of Canadian Petroleum Technology*, **33** (1994).

[4] Jiang Q., Butler R.M., Yee C.T., "Development of the Steam and Gas Push (SAGP) Process", The 7th UNITAR International Conference of Heavy Crude and Tar Sands, Oct. 27-30, Beijing, China. Paper 59 (1998).

[5] Butler R.M., "Horizontal Wells for the Recovery of Oil, Gas and Bitumen. Petroleum Society Monograph", Prentice-Hall, Englewood Cliffs, NJ, **2**, pp. 169-200 (1994).

[6] Mendoza H.A., Finol J.J., Butler R.M., "SAGD, Pilot Test in Venezuela", Paper *SPE 53687*, presented at the Latin American and Caribbean Petroleum Engineering Conference Held in Caracas, Venezuela, April 21–23 (1999).

[7] Butler R.M., "Thermal Recovery of Oil and Bitumen". Prentice-Hall, Englewood Cliffs, NJ USA, p. 285-359 (1991).

[8] Joshi S.D., "A Laboratory Study of Thermal Oil Recovery Using Horizontal Wells," Paper SPE/DOE 14916 Presented at the SPE/DOE Fifth Symposium on Enhanced Oil Recovery Held in Tulsa, OK, April 20-23 (1986).

[9] Fonseca D.J., Navaresse D.O., Moynihan G.P., "Simulation Metamodeling Through Artificial Neural Networks. Department of Industrial Engineering, The University of Alabama", Box 870288, Tuscaloosa, AL 35487-0288, USA. (2003).

[10] "Engineering Applications of Artificial Intelligence", Volume 16, Issue 3, Pages 177-183 (2003).

[11] Afonin P.V., Derjabkina V.V., Kozhukhova A.A., Lamskova O.Y., The Design and Analysis of Complex System Neural Network Metamodels, *Journal Optical Memory & Neural Networks*, **16**(3), p. 154 (2007).

[12] Singhal A., Das S.K., Leggitt S.M., Kasraie M., Ito Y., "Screening of Reservoir for Exploitation by Application of Steam Assisted Gravity Drainage/ Vapex processes". Paper SPE 37144, presented at the SPE International Conference on Horizontal Well Technology held in Calgary, Alberta, Canada, Nov. 18-20 (1996).

[13] Edmunds N.R., Suggett J.C., "Design of a Commercial SAGD Heavy Oil Project", Paper SPE 30277, Presented at the SPE International Heavy Oil Symposium held in Calgary, Alberta, Canada, June 19-21 (1995).

[14] Butler R.M., Rise of Interfering Steam Chambers. JCPT 26 (3), 70-75, May- June. Butler, R.M., 1994. "Horizontal Wells for the Recovery of Oil, Gas and Bitumen. Petroleum Society Monograph", vol. 2. Prentice-Hall, Englewood Cliffs, NJ, pp. 169-200 (1987).

[15] Butler R.M., A New Approach to the Modeling of Steam Assisted Gravity Drainage, *JCPT* **24** (3), 42-51, May-June (1985).

[16] Reis J.C., A Steam-Assisted Gravity Drainage Model for Tar Sands: Linear Geometry, *JCPT*, **31** (10), 14-20, December (1992).

[17] Scott Ferguson F.R., Butler R.M., Steam-Assisted Gravity Drainage Model Incorporating Energy Recovery from a Cooling Stream Chamber, *JCPT*, **27**(5), 75-83, September-October (1988).

[18] Yang G., Butler R.M., Effects of Reservoir Heterogeneities on Heavy Oil Recovery by Steam-Assisted Gravity Drainage, *JCPT*, **31**(8), 37-43, October (1992).

[19] Nasr T.S., Law D.H.-S., Golbeck H., Korpany G., Counter Current Aspect of the SAGD Process, *JCPT*, **39**(1), 41-47, January (2000).

[20] Kamath V.A., Hatzignatiou D.G., "Simulation Study of Steam Assisted Gravity Drainage Process in Ugnu Tar Sand Reservoir", Paper SPE 26075, Presented at the Western Regional Meeting Held in Anchorage, AK, USA, May 26-28 (1993).

[21] Kisman K.E., Yeung K.C., "Numerical Study of the SAGD Process in the Burnt Lake Oil Sands Lease", Paper SPE 30276, Presented at the International Heavy Oil Symposium Held in Calgary, Alberta, Canada, June 19-21 (1995).

[22] Queipo N.V., Goicochea J.V., Pintos S., Surrogate Modeling-Based Optimization of SAGD Processes. *Journal of Petroleum Science and Engineering*, **35**, p. 83 (2002).

[23] Keller R.M., Dungan J.L., Meta-Modeling: a Knowledge-Based Approach to Facilitating Process Model Construction and Reuse, *Ecological Modeling,* **119**, p. 89 (1990).

[24] Peaceman D.W., Exxon Production Research Co. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability, *SPE Journal*, **23**(3), p. 531 (1983).

[25] Thomas S., Enhanced Oil Recovery-An Overview, *Oil & Gas Science and Technology Journal* – Rev. IFP, **63**(1), p. 9 (2008).

[26] MacKay, D.J.C., Bayesian interpolation, *Neural Computation*, **4**(3), p. 415 (1992).

[27] Foresee F.D., Hagan, M.T., Gauss-Newton Approximation to Bayesian Regularization, Proceedings of the International Joint Conference on Neural Networks, p. 1930 (1997).