Improved and More-Rapid History Matching With a Nonlinear Proxy and Global Optimization

Generating reservoir-simulation models that match field production data has been a long-time industry challenge. Two workflows are presented for assisting history match. One workflow minimizes the misfit between simulated and historical data with a global optimizer. Another workflow trains a comprehensive nonlinear-proxy model with a small set of numerical simulations from experimental design to reduce the number of numerical simulations.

Introduction

History matching is, by its nature, an ill-posed optimization problem with many unknown reservoir parameters that could be adjusted to achieve a match against a relatively small amount of measured data at wells. The most common method of history match is to execute many simulations one at a time, changing a few parameters in a trial-and-error fashion. Often, a reservoir-simulation history match might require months of effort and many simulations to achieve a single model that neglects model nonuniqueness and often may not be a good predictor of future field performance.

Many automated history-match algorithms have been investigated. Early assisted history matching emphasized deterministic gradient-optimization algorithms that require complete derivatives of the production response with respect to reservoir parameters. Subsequently, the emphasis evolved to faster methods to compute the sensitivity coefficients, such as streamline simulation, with emphasis on preserving uncertainty through the integration of geostatistics. Recent developments with adjoint models have renewed interest in sensitivity-based algorithms. However, none of these methods have been taken up widely by practioners.

With black-box optimization, the number of required simulations to achieve a match can be very large, numbered in the hundreds of simulations. If a single simulation model takes many hours to execute, there is good incentive to reduce the number of simulations for the optimization, even with the use of a distributed-computing system.

This study used a global optimizer, while seeking to reduce the number of simulations with a fast proxy model. The use of proxy-regression models has been suggested to reduce the number of required simulations, but analytical-response models may not be sufficiently robust to represent the nonlinear reservoir production and pressure vs. time profile data. The use of neural networks as nonlinear-proxy models also has been suggested. In this paper, a history-match workflow is presented that has a robust, nonlinear neural-network proxy model to improve a history match and reduce the number of finite-difference simulations required.

This article, written by Technology Editor Dennis Denney, contains highlights of paper SPE 101933, "Improved and More-Rapid History Matching With a Nonlinear Proxy and Global Optimization," by ***A.S. Cullick,*** *SPE,* ***W.D. Johnson,*** *SPE, and* ***G. Shi,*** *Landmark Graphics, prepared for the 2006 SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 24–27 September.*

Assisted-History-Match Workflows

Assisted history matching is approached as a global-optimization problem—that is, a misfit function of observed vs. simulated production data is minimized by modifying reservoir and well parameters iteratively. Two related workflows are shown in **Fig. 1.** Workflow A is a direct approach to the optimization. It is a misfit function that is minimized by iterating the full reservoir-simulation model. History match is a nonunique-optimization problem, and the solution surface might have numerous local minima, many of which could be viable solutions. A stochastic-optimization algorithm is used that is based on scatter and tabu search. The solution search can require many simulations, which are conducted in a distributed-computing environment.

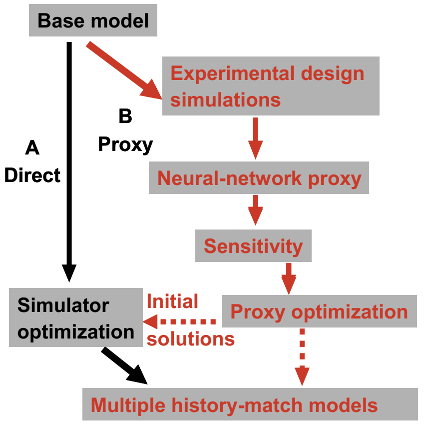

Fig. 1—Alternative workflows for history matching reservoir data. A: direct optimization with full simulator. B: optimization assisted with nonlinear proxy.

A potential problem with Workflow A is the time required. For example, a 200-iteration history match on a simulation model that runs in 6 hours on one processor requires 1,200 compute hours, or approximately 75 hours on a 16-node cluster. Alternative Workflow B uses a proxy model to provide parameter sets that reduce the required number of simulations.

Direct-Optimization Workflow. In Fig. 1, a base simulation model is used with an optimizer. The optimizer minimizes a misfit function by running the simulator through many simulation iterations while modifying the unknown reservoir

For a limited time, the full-length paper is available free to SPE members at ***www.spe.org/jpt****. The paper has not been peer reviewed.*

parameters. The optimizer uses a stochastic, scatter-search algorithm.

Proxy-Model Workflow. The alternative workflow in Fig. 1 reduces the number of simulations required by use of a proxy model of the simulator through a design of experiments. The proxy also identifies parameter sensitivities. If the proxy models the simulator response closely, it is used in a fast optimization to generate reservoir parameters for the match. Then the output is either validated back in the simulator or used as initial solutions to speed the simulation optimization.

Base Case

A regional reservoir-simulation model and a subset of field historical oil-production data covering the period 1995 to 2002 were used as a base model. The area production was approximately 17 500 std m^3/d in 1997. Early simulation models were used for development planning, and a history match against formation-pressure and water-production data was reported, with the key reservoir parameters being fault and mudstone transmissibility.

A high-resolution simulation model was coarsened to 50,000 grid cells, with 44,640 active cells and 15,537 nonneighbor-fault connections. The model has approximately 1.41×10^8 stock-tank m^3 oil in place and area dimensions of approximately 4.5 km×12.9 km, and it is a heterogeneous model for the porosity, permeability, and net-to-gross thickness. The original oil/water contact was approximately 1623 m subsea, with a pressure of approximately 17 000 kPa. **Fig. 2** is a snapshot of the regional model showing the wells in the model grid and illustrates the rock pore volume (red is relatively low pore volume, and green is relatively higher pore volume).

The oil-production-rate data were available for 43 production wells and water-injection rates for 16 water-injection wells. Pressure data were not available, and water-production data were incomplete. The oil-production and water-injection rates were set as flow targets in the simulation model.

Case Studies

Two case studies are presented. Both cases use the base reservoir-simulation model. The studies differ as to the number and types of unknown parameters, the experimental design, and the rigor of neural-network training.

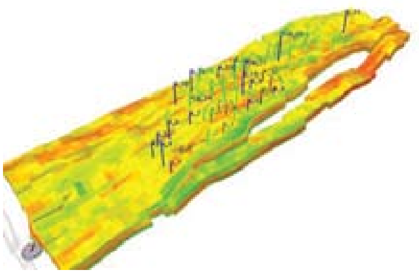

Fig. 2—Regional reservoir-simulation model of the Wytch Farm field.

Case 1. For the purpose of developing the history-match methodology, 13 reservoir and well parameters were selected arbitrarily as unknowns, to be varied over a range of values.

Historical Production Data. Detailed per-well water-production data were not available. Therefore, a test "history" data set was generated with one set of randomly selected values for the 13 parameter unknowns in a simulation, along with the observed oil-production and water-injection data as constraints. The test data set was designated as the production "history." Optimization history matches were compared with these history data.

Objective, or Misfit, Function. The objective function in history matching minimizes the misfit between simulated and historical data. One measure of the misfit is the squared difference between simulated and historical data. To emphasize the importance of a particular data point or time, a weight was assigned. To consider data with quite different absolute ranges (e.g., fluid-production rate, pressure, or water cut), a scaling factor was used. The misfit function is detailed in the full-length paper.

The objective function for this study included the water-production rates for 11 wells (out of the 43 total wells). The weights were adjusted to weight higher-water-rate producers more than very-low-rate producers.

Direct Optimization. First, Workflow A of Fig. 1 was used. The optimizer solver used standard and heuristic global-search methods (metaheuristics). The optimizer does not use the complex physical-model equations in the numerical simulator; the flow simulator handles the nonlinear physical-model production flow and constraints. The optimizer tracks multiple "stacks" of reference-set solutions and previous solutions that improved the objective. Thus, the algorithm is very efficient for executing in parallel and takes advantage of grid computing. As with any optimizer that uses a global approach, the optimizer does not converge to a single mathematical "provable" optimum. It improves the objective progressively by finding "better" solutions as the procedure progresses. Thus, stopping criteria were imposed as a specified number of iterations or as some amount of decrease in misfit. Multiple solutions that reduce the misfit function by some amount are the result.

Results. The optimizer reduced the objective substantially after approximately 100 iterations, but then tried some other solutions, finally finding good convergence after 190 iterations. The last approximately 50 iterations were taken as "good" solutions because their misfit values were less than 0.5 vs. early misfit values of 5 or greater. The good solutions were clustered near the historical data, and the others formed a broad band of poor solutions.

Experimental Design. For Workflow B of Fig. 1, the first step was an experimental design over the parameter range. Combinations of the input-parameter values were constructed such that the maximum information could be obtained from the minimum number of simulation runs. A full-factorial design has all possible combinations. For 13 parameters, with three value levels for each, there are 3^{13} experiments. There are many different experimental-design algorithms, the purpose of which is to reduce significantly the number of experiments needed for sensitivity analysis or response modeling. A nearly orthogonal array (NOA) design, which is very similar to an orthogonal array, was used, except a small number of columns are not pairwise orthogonal. An NOA can handle many factors with mixed levels. The minimum number of experiments that is required to conduct the method can be calculated with the degrees-of-freedom approach.

Neural-Network-Response Model. In contrast to a regression model, an artificial-neural-network model can handle the high dimensionality through hidden layers, which link input parameters to outputs through nonlinear transfer functions with weights. Many properties of artificial neural networks make them useful as optimization models. They execute very rapidly, are continuously differentiable, and are universal approximators. It is this last property that is important for the proxy. A neural network is able to capture and represent complex input-to-output relationships.

The neural-network fitting algorithm solves a nonlinear least-squares problem on the hidden-layer coefficients. The supervised training is done with a set of inputs, which comprises the reservoir unknowns, and a set of outputs, which includes such terms as the production profiles and pressures from the simulator. The training is done with a gradient-descent algorithm. The neural network selects relevant terms of any order, and the number of weights is the number of hidden layers, which is determined during the fitting. Thus, the use of coefficients is judicious. Initially, all hidden layers are linear and behave similarly, and the dimension is roughly one. As training progresses, the dimensionality increases (i.e., the effective number of hidden layers). Weights are initialized near zero so that hidden units begin in linear regions. Training increases the weights' magnitudes, and hidden layers become more nonlinear.

History-Match Optimization From Neural Network. The neural network is an analytical model with analytic derivatives; thus, a gradient optimizer (i.e., a large-scale generalized reduced-gradient algorithm) solves the neural network for optimal parameter sets. The neural-network optimization solves in fractions of a second.

The neural-network optimization provided initial guess values for the parameters to the full simulator optimization. Multiple initial solutions from the neural network were generated with multiple initial parameters. The multiple solutions were added to the suite of initial values already available from the design runs. Use of the neural-network initial values reduced the optimization iterations needed and resulted in lower final objective-misfit values than did the direct-simulation optimization.

The optimal-parameter values cluster as expected. Generally, better reservoir-parameter values were obtained with the neural network.

Case 2. Case 2 is an extension of Case 1 with additional uncertain parameters. The goal was to determine whether adding parameters, reducing the number of design simulations, and accepting a lower quality of neural-network training would deteriorate the overall process. Uncertain parameters on relative permeability, fault transmissibility, and volume by layer were added. Executing Workflow A for Case 2 resulted in reasonable convergence in approximately 250 simulations.

A multiple-level experimental design was used. Some parameters were assigned two levels on the basis of an assumption of approximate linearity, and the remaining parameters were assigned three levels. The minimum number of simulations was 48.

Neural-Network Training. The neural-network training had an overall root-mean-square error of 0.95. This error was not as good as for Case 1 and reflects the smaller design set that was used relative to the number of parameters. And unlike Case 1, for which the neural-network profiles overlay the simulation profiles in the design set, Case 2 profiles only approximate the design-set profiles.

Optimization With Neural-Network Inputs. The neural network was used in a stochastic optimization of the misfit function. Multiple solution sets were generated by initializing the optimization with random parameter sets. The parameter sets for 10 of the lowest misfit values were selected and used in a simulation optimization, along with the 48 design runs, as initial values. Focusing on the first 50 simulations showed that the first 10 misfit values from the neural network were all less than a value of 1.0, whereas without the neural-network inputs, the first 10 simulations have only two values less than 1.0. With the neural-network inputs, the number of low misfit values was 29 out of the first 50 simulations, whereas without the inputs, there are only eight low values in the first 50 simulation runs. With the benefit of the neural network, there was a cluster of simulations between 90 and 140 with a misfit of 0.45 or less. These compare with a cluster between 220 and 275 simulations without the benefit of the neural network. The neural network reduced the number of required iterations by approximately 25 to 35% to achieve equivalent low misfit values.

Twenty-eight solutions with misfit values less than 0.4 were selected as the low-misfit subset. Oil rate was a constraint in the simulator and was not a term in the misfit function. For the 28 low-misfit solutions, the parameter values cluster similarly to those observed for Case 1, with the addition of the relative permeability parameters (for the logistic relative permeability function), fault transmissibility, and layer volume.

Discussion

The proxy model provided a robust nonlinear model. The methodology appeared robust and quite efficient compared with a conventional manual history match, and it complements the global-optimization workflows. Two cases demonstrated the potential to reduce significantly the number of simulations by use of a proxy compared with not having the proxy.

For simulations that take many hours to execute, a 25% reduction in the required number of simulations can be significant. The neural network also provided sensitivities. A further efficiency improvement might be realized when a model needs updating. History matching is not a one-time process; it must be repeated many times over a field's life, often requiring a lengthy effort. A neural network can initiate a model update by providing initial solution-parameter sets. A model update might involve new reservoir or well parameters. A neural network can be updated quickly by combining a previous model with a model having new parameters and data.

As with any history-match procedure, history match assisted with an optimizer and neural network cannot replace the importance of choosing parameters that represent uncertainties of the reservoir, and that truly might affect the performance. If some critically important characteristic of the reservoir is left out of the analysis or is assumed to have a range that does not include valid values, then history matching will not be successful, no matter what process is used. It also is important to select judiciously the critical reservoir responses that must be matched (i.e., those included in the misfit function such as fluid rates, water cut, pressure, and gas/oil ratio). Post-analysis of the results of the history match is a key to ensure that the parameter values make sense. **JPT**