0637

Development of Optimized History-Matched Models for Coalbed Methane Reservoirs

Anne Y. Oudinot, Advanced Resources International Inc.
Aiysha Sultana, Advanced Resources International Inc.
Reynaldo R. Gonzalez, Advanced Resources International Inc.
Scott R. Reeves, Advanced Resources International Inc.
Marco Wörmann, NuTech Solutions

ABSTRACT

Due to their high heterogeneity, coalbed methane (CBM) reservoirs as well as other non-conventional gas resources are often referred to as statistical plays. The productivity of CBM reservoirs depends upon a connected set of reservoir, completion and production characteristics making history-matching a tedious and time-consuming trial and error process. However, inverse modeling techniques have become a common practice for estimating unknown reservoir parameters. By minimizing the objective function quantifying numerical differences between actual data and those provided by reservoir simulation, optimal values of uncertain reservoir parameters can be generated. This technique has been implemented by coupling a CBM reservoir simulator with an external optimizer. In the process, thousands of simulations are automatically run in a series of iterations to explore the many combinations of unknown parameter values across their spectrum of uncertainty. At the end of each iteration, an error function is calculated to quantify the deviation between each simulation response and the observed data. The optimizer uses a genetic algorithm to then generate the next set of input parameters until a constant and acceptable value for the error is obtained. Tested through synthetic cases and applied to real field projects, this method has shown its efficiency and speed in history-matching reservoir problems and improving reservoir characterization and subsequent forecast results.

INTRODUCTION

Automated history-matching has been an on-going research area for years[1-3]. A model was previously built by coupling a fractured reservoir simulator with a Monte-Carlo modeling tool[4]. Through this process, thousands of simulations are run, varying the unknown input parameters per defined probability distributions. A least-square error function is then computed between the actual data and the simulated data and cases of interest are automatically high-graded. In many cases, the results of this stochastic-based simulation methodology have provided more than one history match to a simulation problem, showing the non-uniqueness of the process and adding insight into the range of parameters that impact the production response of the coalbed methane (CBM) reservoir being investigated.

In this work, a fractured reservoir simulator, *COMET3*[5-6], has been coupled to a commercially available optimizer (*ClearVu*®™ from NuTech Solutions) in order to rapidly find the best set of input data (optimal parameterization), which the reservoir model can employ to replicate field/well history. This process is completed by minimizing the difference (or objective function) computed between actual data and the simulated data. The authors hope to demonstrate through this work the efficiency of this process, which is fully automated and speeds up the time-consuming manual history-matching process.

PROCEDURE

Technique

The automated history matching is an iterative technique utilizing two software programs, the global optimizer (*ClearVu*) and a fractured reservoir simulator (*COMET3)*. To begin, a spreadsheet version of a *COMET3* input file is built in order to facilitate the variation of input parameters through the addition of formulas (Corey correlations for relative permeability curves, for example). Before the optimization process is started, the case is defined by generating possible ranges for uncertain input parameters (minima and maxima) as well as the total number of simulations to be run and the number of simulations per batch (also referred to as an iteration).

The current set of input parameters is generated by the optimizer based on the probability distributions previously defined by the user. These parameters are then inserted into the simulator input file in their appropriate locations and the spreadsheet is converted to a standard ASCII input file.

Simulations are executed one after the other and at the end of each iteration, the output data from each simulation is saved and compared to historical data. At this point, the corresponding error value is computed using Equations 1 to 3 below. Equation 1, weighted sum of squares of the difference between actual data and simulation data, is used most frequently and recognizes the goodness-of-fit measure for optimization purposes.

$$ErrorValue = \frac{\sum\left(\frac{(SimData - ActualData)^2}{2 * Variance}\right)}{NumberofData} \qquad (1)$$

With

$$Variance = \frac{\sum(ActualData - Mean)^2}{NumberofData} \qquad (2)$$

And

$$Mean = \frac{\sum ActualData}{NumberofData} \qquad (3)$$

These computed error values serve as objective (goodness-of-fit) function values and are evaluated by the optimizer, which produces a new set of input parameters. The next iteration can then be initialized. When several output parameters are to be matched, the objective function is computed separately for each parameter with the final error value being the sum (or some other combination) of the independent values. The workflow of the process is outlined in Figure 1. The algorithm used by the optimizer is described in more detail in the next section.

Optimization Algorithm

The optimizer consists of an "evolution strategy"[7], more exactly a (μ,λ)-evolutionary strategy. Evolutionary strategies were invented by H.P. Schwefel and I. Rechenberg in the late 1960s in Germany. The basic idea of such a strategy is to mimic biological evolution, also known as mutation-selection-mechanism (Darwinian Theory). Numerically, the great advantage of employing evolutionary strategy is that it is not necessary to calculate derivatives. Consequently, it is possible to deal with non-linear functions more easily than with any other optimization techniques.

The algorithm is initialized with a set of λ individuals, where an individual represents a solution in the given search space. The optimizer can support the distribution of λ individuals in the search space completely arbitrarily or via initialization with a given start point. The latter can be used if an optimization shall be based on a known good solution.

The λ individuals are then evaluated -- meaning in this particular case that for each set of individuals, a complete set of simulation iterations are executed. Once completed, the objective value can be calculated as previously described. The next step in the optimization loop is the "selection" mechanism where the individuals are sorted according to their objective values. The population for the next generation is then created by taking the best μ individuals from the λ individuals (also called "comma-selection").

In order to exchange information about the topology between the individuals, the next applied operator is the recombination operator. Usually two individuals from these μ are taken randomly to create a new individual. Here an "intermediate recombination" is used, i.e. for each dimension the average of the values of the two participating parent individuals is calculated. This is done until λ offspring are generated.

Since the recombination is responsible for large moves in the search space the subsequent "mutation" is applied to make smaller moves and hence to search more locally. The value of each individual's dimension is mutated by adding a normal distributed term, calculated based on the parameters global sigma (σ) and local sigma (σ_i). These additional variables are called "step sizes" or "strategic variables" and belong to the individual's gene type as the objective variables (input variables) do. Due to the normal distribution, smaller changes are more likely than greater changes. The σ, σ_i itself are also underlying a normal distributed variation. By this means the step sizes change and should always be adapted to the current fitness landscape. This mechanism is called "self-adaptation."

Although an evolutionary algorithm is a very general and robust optimization technique, it is possible in each specific case to make the optimization more efficient and converge faster by applying known domain knowledge and good parameter settings.

SYNTHETIC EXAMPLE

Problem Description

A synthetic case was used to test the optimizer. The model is comprised of one well completed in one gas-shale layer, producing at a constant bottomhole pressure for 10 years with up to 15 unknown input variables (Table 1).

Simulated gas and water production rates are shown in Figure 2. In order to emulate reality, the same test was repeated with the addition of random noise added to the smoothed actual data as can be seen on Figure 3. These "noisy" data were obtained by randomly disturbing the smooth data at each time step using conventional statistical techniques.

Test Description

The first set of simulations was run, initially varying only two input parameters (permeability and porosity). The process was then repeated, with four input parameters (permeability, porosity, desorption pressure and the water relative permeability exponent from the Corey correlation) randomized. Then, four additional parameters were added to the list (skin factor, Langmuir Pressure, gas relative permeability exponent and maximum relative permeability to gas), for a total of eight variables. Finally, seven input parameters were added to the previous eight variables (initial water saturation, fracture spacing, water density, Langmuir Volume, sorption time, permeability anisotropy and irreducible water saturation), bringing the total variable population to fifteen. Each of these four cases was optimized using both the smoothed and noisy data sets.

Results

The objective function values for both smooth and noisy data cases are displayed in Figures 4 and 5, respectively. For the smoothed case with only two input parameter variations, a constant error value of 0.016 was reached after about 1,200 simulations. Approximately 2,000 simulations were necessary to obtain a constant error value of 0.017 varying four input parameters, while 6,000 simulations were necessary to obtain a constant error value of 0.029 varying eight parameters. Finally, 7,500 simulations were necessary to achieve a constant error value of 0.028 varying fifteen input parameters. From these cases, and as one might expect, the number of simulations necessary to achieve convergence increases with the number of input variables. Additionally, the stabilized objective function tends to increase. Even though the number of simulations necessary to reach convergence can seem quite high, the process was actually very fast as each simulation took on average one minute to be completed, which is very common for single well coalbed methane cases. Further, the process can proceed unattended, freeing-up valuable engineering resources.

Regarding the results for the noisy data on Figure 5, the Original Error line illustrates the deviation between the smoothed and noisy data. It should be pointed out that more simulations were necessary to achieve a constant error value for the eight variables case as opposed to the fifteen variables case. This is due to the fact that the error value got stuck into a local minimum and about 2,000 simulations were necessary before the optimizer could get out of it. This is a possible occurrence with this approach, and care must be taken to allow the process to continue if stuck in a local minima. As expected, the objective function value is generally higher for the noisy case than for the smoothed case as reservoir simulations usually provide quite smooth production output data. Of important note is that the optimized objective function between noisy data and simulation data is always lower than the difference between smoothed and noisy data. This indicates that alternative reservoir characterizations to the “actual” case can actually yield better history-matches. Thus history-matching noisy production will likely yield reservoir parameter values different than those measured.

Error values between the actual data and the optimized data for each case are provided in Tables 2 to 5. With a few number of varied input parameters, the error between the real data and the optimized data is quite low and acceptable. However, large discrepancies, as high as 73%, can be obtained by varying more input parameters, even with the smooth data. These results show that the optimal values found by optimization can be quite different from the actual input parameters, raising the question of the validity of an eventual forecast. It further illustrates the non-uniqueness of the history-matching process.

As a consequence, the results from the optimization were then used to forecast water and gas production over a 20 year period for all cases (smoothed and noisy data with different sets of optimized values). The results are shown in Figures 6 and 7 and are compared to the forecasted actual data. It can be seen that the smoothed and noisy data give similar forecasts. The smoothed, fifteen variable case, however, happens to be the case with the highest differences between real input parameter values and optimized values. Nevertheless, even when widely varying so many parameters, the forecast still maintains a similar character to the previous cases. Overall, the results of these forecasts are quite acceptable, and thus even with “incorrect” reservoir characterizations from history-matching, production forecasting can still be reliable.

CASE STUDY

Problem Description

The case study involved history matching the gas and water production data for a coalbed methane well completed in one layer. The well had about two months of production data.

Set Up

The simulation input file for the model was built based on existing reservoir parameters, such as coal depth and coal thickness. Likely ranges were established for seventeen input parameters that were not measured and are shown in Table 6. The well was produced using a minimum bottomhole pressure constraint.

Results

Actual production data versus optimized production data are shown in Figure 8. A very good match was obtained as the optimized data fall in the median of the actual noisy data. The evolution of three input parameters (permeability, porosity and initial water saturation) as well as the objective function during the optimization process is shown in Figure 9. The input parameters start well spread within the range defined, which quickly narrows down to a single value. Convergence required about 700 simulation runs, which were performed under 20 hours of (unsupervised) computer time. This process yielded superior results in less time than would have been required using conventional (normal) history-matching procedures.

CONCLUSIONS

- A means of automatic history matching has been developed by the combination of a reservoir simulator and an optimizer for characterization of CBM reservoirs.

- Even though the process is viable and provides acceptable matches and production forecasts, the number of input parameters should be kept reasonable as large discrepancies can be obtained between actual and optimized data. However, the reservoir characterization does not need to be precise for forecasting purposes.

- This method assumes that a unique solution to the problem exists but it is fully recognized and accepted that the solution is non-unique.

- While requiring many iterations of the simulator, the process is quite efficient and greatly leverages scarce engineering resources.

REFERENCES

1. Silva, F.P.T., Rodrigues, J.R.P., Paraizo, P.L.B, Romeu, R.K., Peres, A.M.M., Oliveira, R.M. and Pinto, L.B. (Petrobras) and Maschio, C. (Unicamp), “Novel Ways of Parameterizing the History Matching Problem” SPE 94875 presented at the 2005 SPE Latin American and Caribbean Petroleum Engineering Conference, Rio de Janeiro, Brazil, 20-23 June.

2. Kabir, C.S. (Chevron Texaco Overseas Petroleum), Chien, M.C.H. (Focus Simulation Inc.), and Landa, J.L. (Chevron Texaco EPTC), “Experiences with Automated History Matching” SPE 79670 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, Feb. 3-5, 2003.

3. Tavassoli, Z., Carter, J.N. and King, P.R. (Imperial College), “Errors in History Matching” SPE Paper 86883, published in the SPE Journal, Sept. 2004.

4. Oudinot, A.Y., Koperna, G.J. and Reeves, S.R. (Advanced Resources International Inc.), “Development of a Probabilistic Forecasting and History Matching Model for Coalbed Methane Reservoirs” presented at the 2005 International Coalbed Methane Symposium in Tuscaloosa, Alabama, May 25-26.

5. Sawyer, W. K., Paul, G. W., and Schraufnagel, R. A., “Development and Application of a 3D Coalbed Simulator,” paper CIM/SPE 90-119 presented at the 1990 CIM/SPE International Technical Conference, Calgary, June 10-13.

6. Paul, G. W., Sawyer, W. K., and Dean, R. H., “Validation 3D Coalbed Simulators,” paper SPE 20733 presented at SPE Annual Technical Conference and Exhibition, Houston, Sept. 23-26, 1990.

7. Baeck, T.; Fogel, D.B., Michalewicz, Z.: “Handbook of Evolutionary Computation”; Oxford University Press ; New York, Oxford ; 1997

Table 1: Synthetic Case Simulation Input Data

	Actual Value	Minimum	Maximum	Units
Average Permeability	5.5	0.1	500	mD
Porosity	0.006	0	0.3	fraction
Pd/Pi	1.00	0.5	1	fraction
KrwExp	1.4	1	3	n/a
KrgMax	0.82	0.35	1	n/a
KrgExp	2.2	1	3	n/a
Skin	-2.0	-4	0	n/a
Langmuir Pressure	354	100	500	psi
Irreducible Water Saturation	0.18	0.05	0.35	n/a
Permeability Anisotropy	4.3	1	10	n/a
Sorption Time	1223	10	3650	days
Langmuir Volume	7.55	1	15	scf/ft3
Initial Water Saturation	1.00	0	1	n/a
Water Density	50.9	40	60	lb/ft3
Fracture Spacing	41	5	125	inches

Table 2: Synthetic Case – Two Variables Results

Parameter	Units	Actual Value	Smooth Data Optimized Value	Noisy Data Optimized Value	Smooth Data % Error	Noisy Data % Error
Average Permeability	mD	5.5	5.5	5.5	0.02	0.15
Porosity	fraction	0.01	0.01	0.01	0.02	2.98

Table 3: Synthetic Case – Four Variables Results

Parameter	Units	Actual Value	Smooth Data Optimized Value	Noisy Data Optimized Value	Smooth Data % Error	Noisy Data % Error
Average Permeability	mD	5.5	6.0	6.4	9.2	16.5
Porosity	fraction	0.01	0.01	0.01	8.5	8.2
Pd/Pi	fraction	1.00	0.88	0.99	11.7	0.5
KrwExp	n/a	1.4	1.2	1.9	17.9	36.2

Table 4: Synthetic Case – Eight Variables Results

Parameter	Units	Actual Value	Smooth Data Optimized Value	Noisy Data Optimized Value	Smooth Data % Error	Noisy Data % Error
Average Permeability	mD	5.5	6.2	8.4	12.6	51.5
Porosity	fraction	0.01	0.01	0.01	0.4	0.7
Pd/Pi	fraction	1.00	0.85	0.72	14.5	28.3
KrwExp	n/a	1.4	1.0	1.3	27.9	10.6
KrgMax	n/a	0.82	0.91	1.00	11.2	21.4
KrgExp	n/a	2.2	1.9	2.1	14.6	5.2
Skin	n/a	-2.0	-1.3	-1.5	34.3	24.9
Langmuir Pressure	psi	354	371	209	4.9	41.1

Table 5: Synthetic Case – Fifteen Variables Results

Parameter	Units	Actual Value	Smooth Data Optimized Value	Noisy Data Optimized Value	Smooth Data % Error	Noisy Data % Error
Average Permeability	mD	5.5	9.0	7.7	63.5	39.8
Porosity	fraction	0.01	0.01	0.01	10.7	5.2
Pd/Pi	fraction	1.00	0.76	0.84	24.2	15.5
KrwExp	n/a	1.4	2.4	1.6	73.0	12.3
KrgMax	n/a	0.82	0.89	0.71	7.9	13.0
KrgExp	n/a	2.2	2.1	2.2	7.7	0.4
Skin	n/a	-2.0	-2.0	-1.5	0.5	27.3
Langmuir Pressure	psi	354	287	340	18.9	3.9
Irreducible Water Saturation	n/a	0.18	0.15	0.14	17.5	22.7
Permeability Anisotropy	n/a	4.3	4.4	3.7	1.0	15.5
Sorption Time	days	1223	1182	668	3.3	45.4
Langmuir Volume	scf/ft3	7.5	10.2	7.8	34.9	3.1
Initial Water Saturation	n/a	1.00	0.60	0.66	39.6	33.7
Water Density	lb/ft3	50.9	50.8	50.7	0.1	0.4
Fracture Spacing	in	41	65	61	58.6	48.1

Table 6: Case Study Simulation Variables Input Ranges

Parameter	Min	Max	Units
Initial Water Saturation	70	100	%
Permeability	0.1	100	mD
Porosity	0.1	5	%
Sorption Time	20	80	days
Pore Compressibility	1.00E-05	1.00E-03	1/psi
Matrix Compressibity	1.00E-07	1.00E-05	1/psi
Irreducible Water Saturation	10	50	%
KrgMax	0.5	1	n/a
KrgExp	2	6	n/a
KrwExp	2	6	n/a
Skin	-10	10	n/a
VL CH4	200	800	scf/ton
VL CH4 / PL CH4	1	1.5	scf/ton/psi
VL CO2 / VL CH4	1.5	2.5	
PL CH4 / PL CO2	1	1.25	
VL CH4 / VL N2	1.3	2.2	
PL CH4 / PL N2	0.2	0.8	

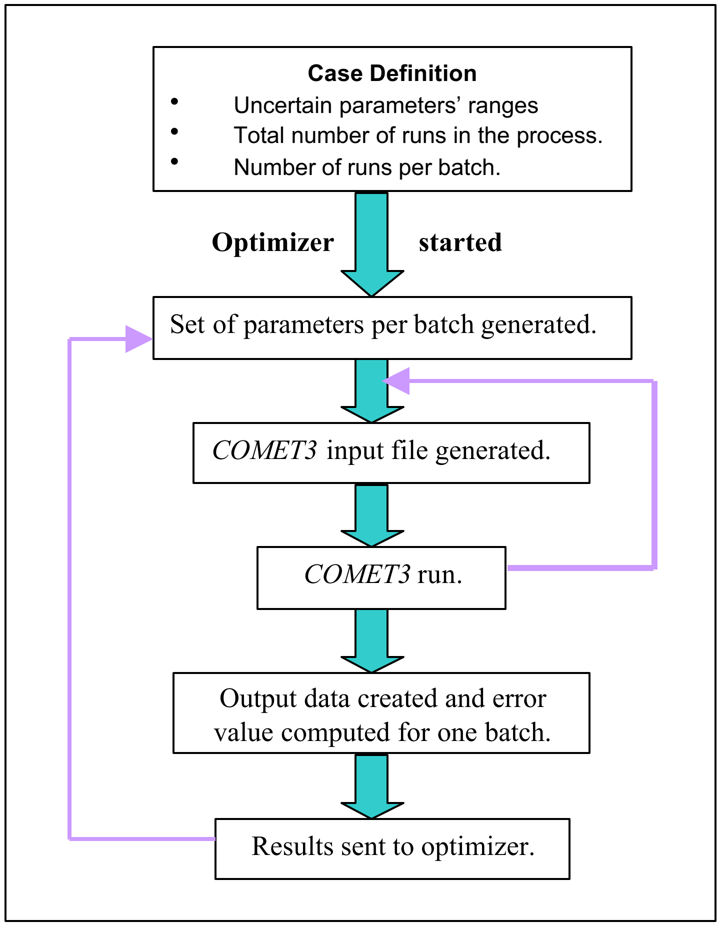


Figure 1: Model Flow Chart

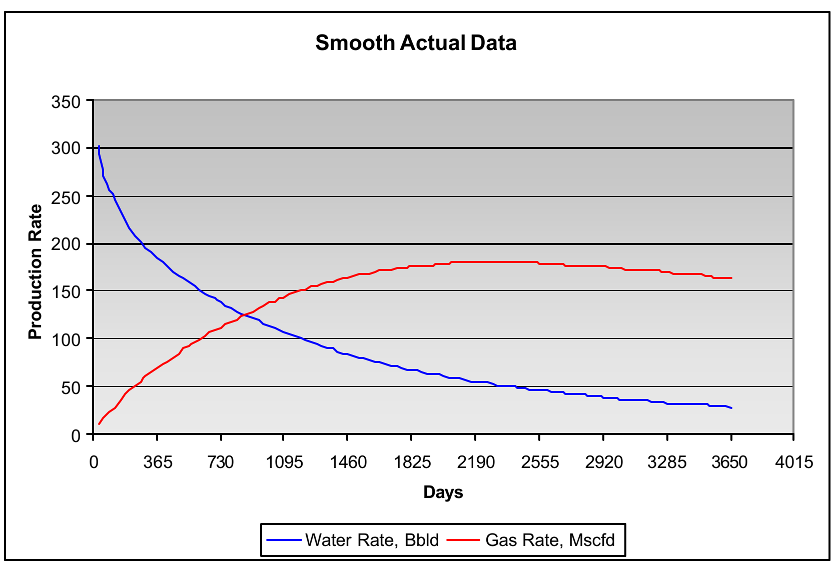


Figure 2: Synthetic Case Smooth Actual Data

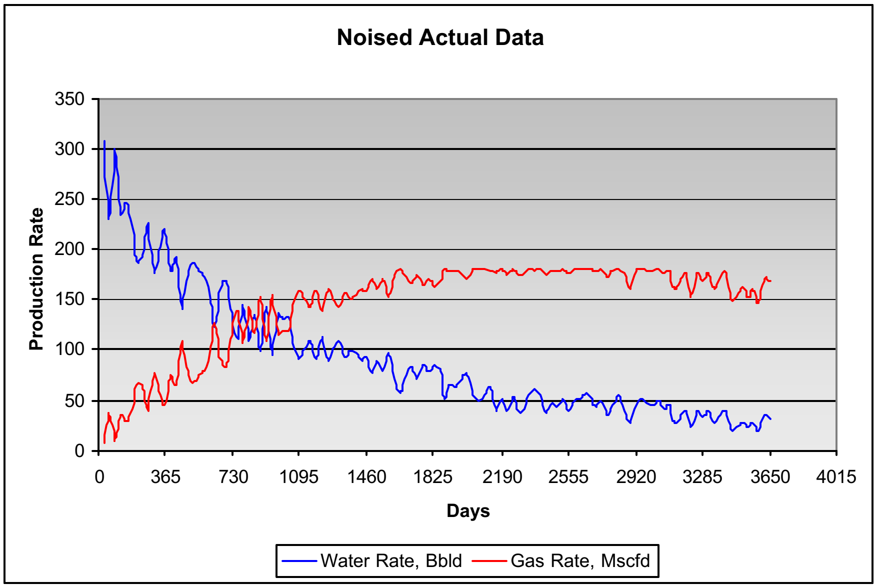


Figure 3: Synthetic Case Noised Actual Data

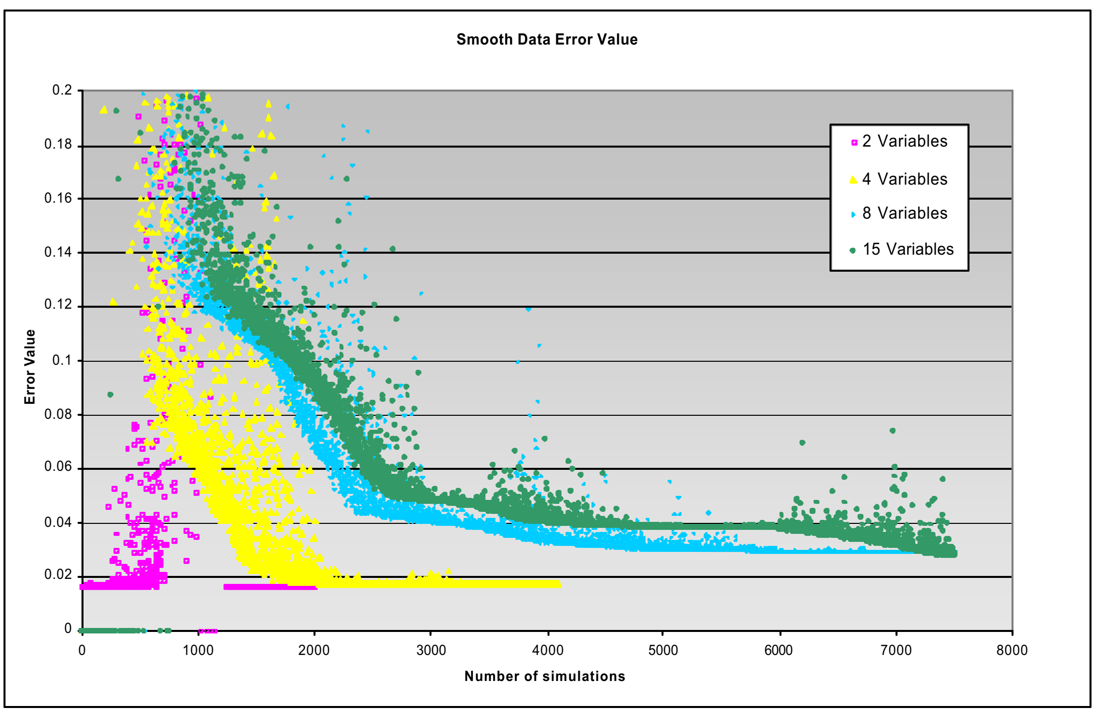


Figure 4: Synthetic Case Smooth Data Error Value

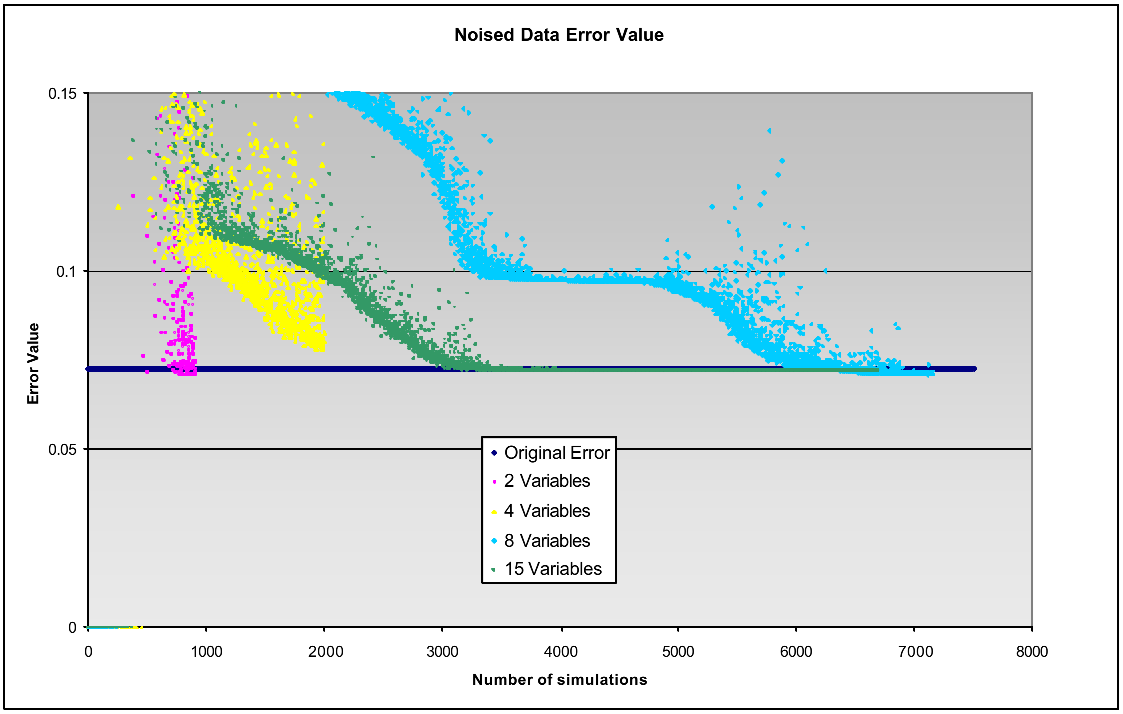
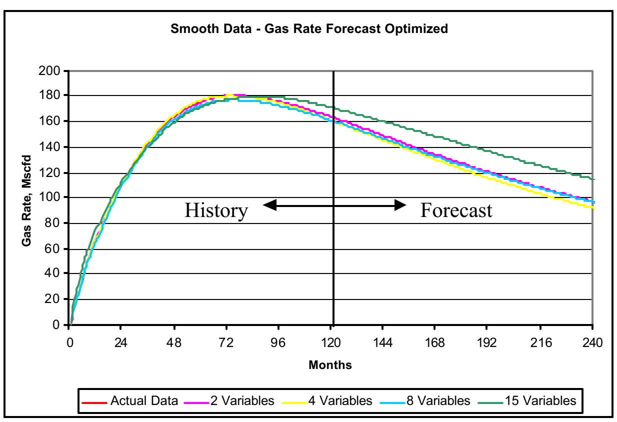


Figure 5: Synthetic Case Noisy Data Error Value



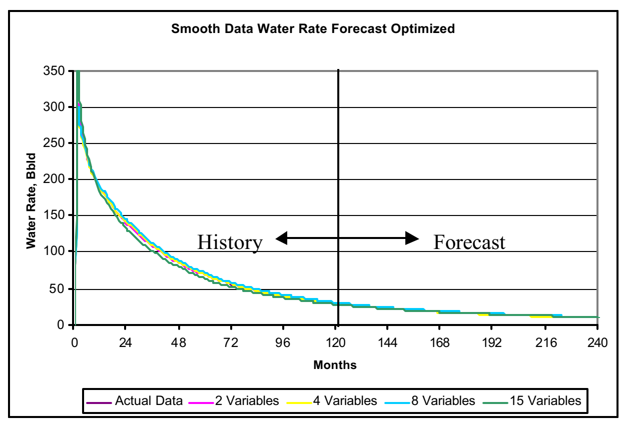


Figure 6: Smooth Data Optimized Forecast

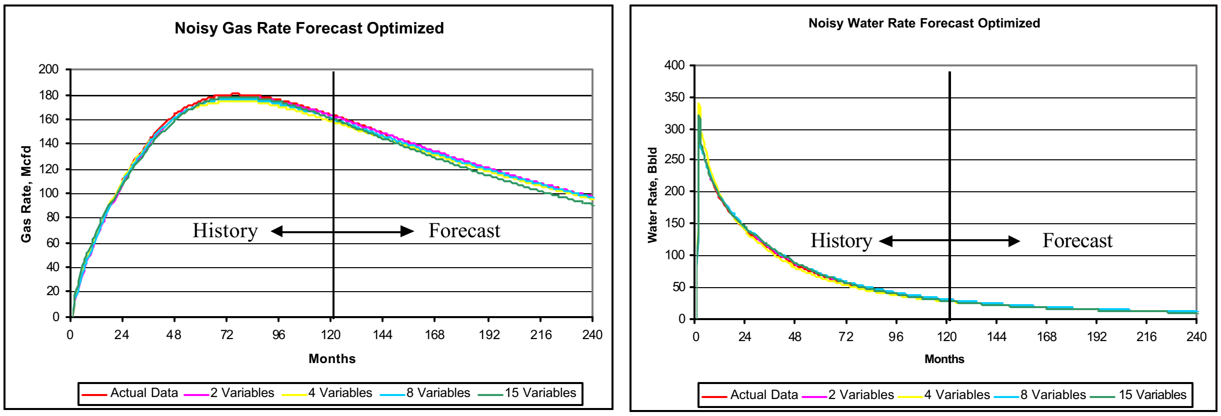


Figure 7 Noisy Data Optimized Forecast

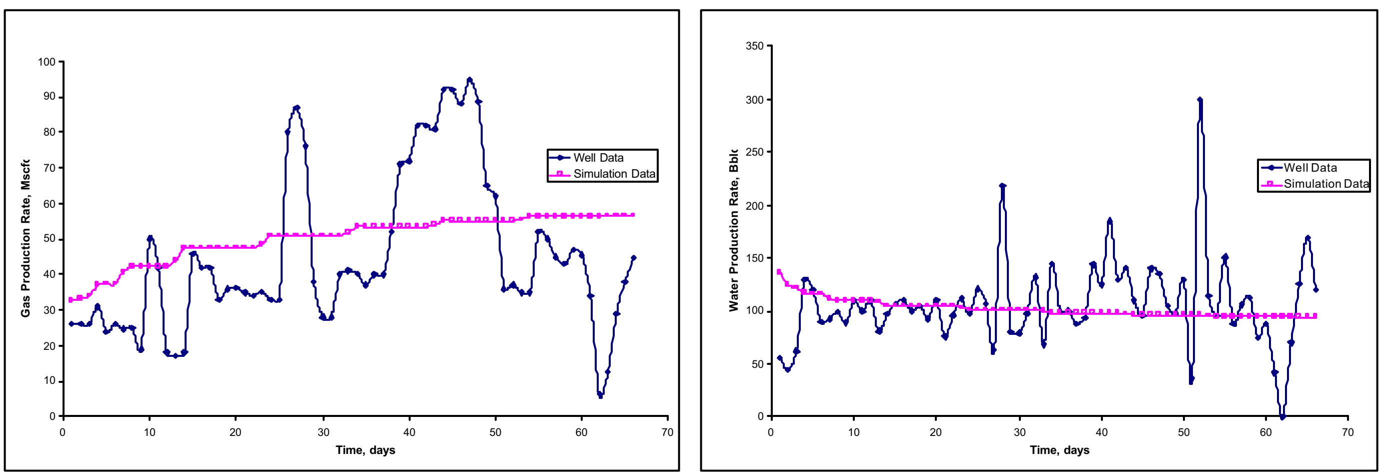


Figure 8: Case Study Optimization Results

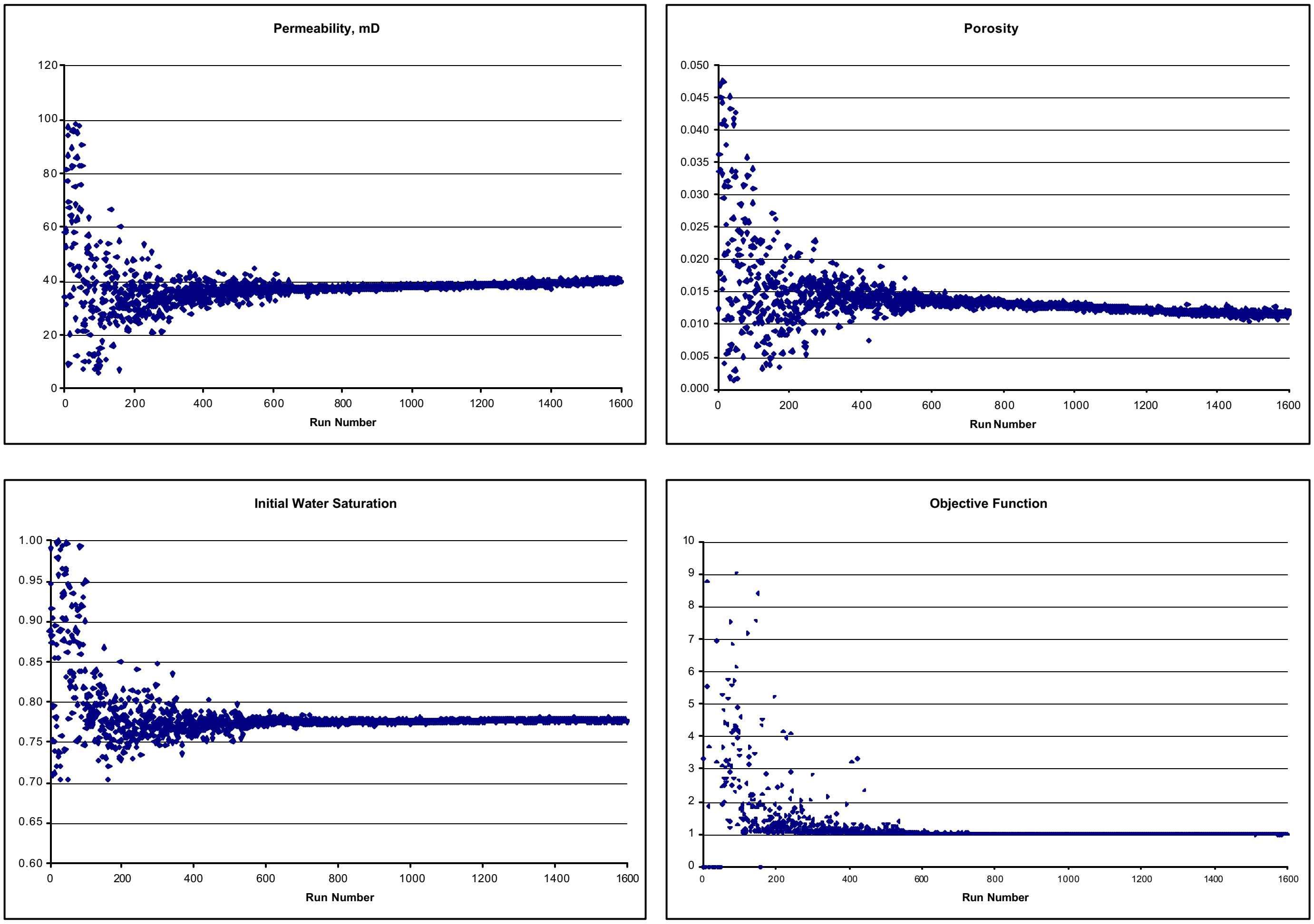


Figure 9: Evolution of Input Parameters and Objective Function