Sensitivity Calculation by Adjoint Method: An Application to Reservoir Simulation

Hao Wang*, Liaohe Oilfield Company, Petrochina, Panjin, China

Abstract

Scope of this paper is the sensitivity calculation by using adjoint method. We have a 9×9 two phase two-dimension quarter five spot model with one injector and one producer. The high uncertainty of the permeability field gives us the data mismatch in the model, and we set the permeability as model parameter (uncertainty). The data distribution we get is bottom-hole pressure at injector and water cut at producer for 10 time steps. To get the optimal permeability field, we first calculate the sensitivity coefficients for permeability by using adjoint method. We already have a forward simulator for this problem that is the fully implicit black-oil simulator. Hence, we need to extract necessary information from the forward simulator, i.e. Jacobian matrix, transmissibility, and accumulation etc. Advantage of the adjoint method is that it enables us to reduce the considerable amount of computation time for calculating the sensitivity matrix compared with gradient simulator method. The forward simulation we need at each minimization step is only one time for calculating the sensitivity. Then we minimize the objective function by Levenberg-Marquardt algorithm.

Introduction

The quality of the reservoir model, i.e., the degree to which it represents the actual reservoir, directly affects reservoir management. This model helps the manager to analyze the behaviors of the reservoir and also it is helpful in production forecasting and optimization. The reservoir model creates on the basis of valuable data obtained during the reservoir life. In the exploration phase, the reservoir model is constructed using 3D seismic data, geologic knowledge of the surrounding area, and log/core measurements from a few exploration wells. These kinds of data are called “static data

In the appraisal phase, drilling additional wells provide new information about the reservoir as a well test. Production data and 4D seismic data are available during the production time of a reservoir. These kinds of information are categorized as “dynamic data”. Conditioning reservoir model to the new information obtained about the reservoir is called history matching process, i.e. history matching is the process by which the geological model properties are modified to fit the production data. The objective of history matching is to build a reservoir model that integrates available data and yields production forecasts that are accurate. Incorrectly identifying structural features, such as fluvial channels, can have very serious consequences such as badly placed wells, by-passed hydrocarbon, and failure to find trapped hydrocarbons.

A reservoir model is described by many parameters, and each parameter can generate millions of pieces of data. Some parameters are specified per grid block (e.g., permeability and porosity) and others for the entire model or a particular layer (e.g. relative permeability and capillary pressure). Due to insufficiency of available data about the reservoir, history matching is an ill-posed problem. It means that it is possible to obtain reservoir models that honor observed measurements but provide incorrect predictions.

To deal with the ill-posed of history matching, the number of parameters has to be reduced. Also, parameterization preserves important geological features and their connectivity that has a significant effect on fluid flow within the reservoir.

DOI: 10.14800/IOGR.429
Received September 5, 2018; revised October 30, 2018; accepted November 8, 2018.
*Corresponding author: wanghao1202@petrochina.com.cn

Several methods have been investigated to reduce the number of unknown parameters. Jacquard and Jain (1965) used simple zonation approach to assign one value to a region of the reservoir. Other researchers (Jahns 1966; Bissel et al. 1994; Chavent and Bissell 1998) modified the Jacquard et al.'s method to adaptive one. Some authors (Grimstad et al. 2003; Sahni and Horne 2005) worked on different techniques for parameterization and history matching at different scales.

Another powerful approach that is suitable for history matching is KLT. KLT is mathematically defined as an orthogonal linear transformation that transforms a set of possibly correlated data into a smaller number of uncorrelated variables called principal components. KLT is theoretically the optimum transform for given data in least square terms. But for the standard KLT model, it is necessary to carry out an eigen-decomposition of the covariance matrix of the random field, which is expensive for large models.

In this research study, we use adjoint method to calculate the sensitivity matrix, thus to reduce the overall computation time during simulation. **Figure 1** shows the work flow for the history matching.

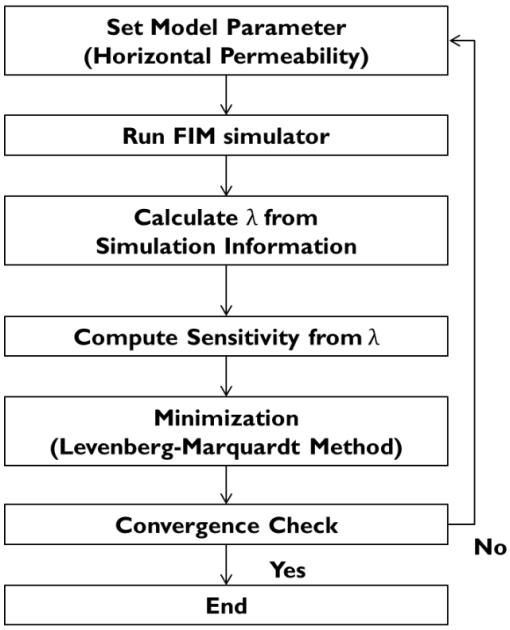

Figure 1—History matching flow chart.

Sensitivity Calculation based on Adjoint method

The simulator used in ajoint-based sensitivity calculation is based on fully implicit finite-difference method of the two-phase blackoil model. Several literatures refer to the derivation of adjoint system with fully implicit formulation (Wu et al. 1999; Li et al. 2003). In this chapter, I briefly show the formulation of adjoint method, used in the algorithm for sensitivity calculation. First equation derived from the finite-different equation is as follows.

$$\left[\nabla_{y^n}(f^n)^T\right]\lambda^n = -\left[\nabla_{y^n}(f^{n+1})^T\right]\lambda^{n+1} - \nabla_{y^n}\beta, \quad (1)$$

where λ^n is the vector of adjoint variables at timestep n, and given by

$$\lambda^n = [{\lambda_1}^n, {\lambda_2}^n, \dots, {\lambda_{2N}}^n]^T, \quad (2)$$

where N is number of gridblock. Returning to **Eq. 1**, $\left[\nabla_{y^n}(f^n)^T\right]$ is transpose of Jacobian matrix at n, which can be extracted from forward simulation. $\left[\nabla_{y^n}(f^{n+1})^T\right]$ is the gradient matrix (derivative of finite-difference equation at $n+1$ with respect to the primary variables at n). $\nabla_{y^n}\beta$ is the sensitivity matrix with respect to the primary variable. Using following initial and boundary conditions, Eq. 1 can be solved backwards in time for n=L, L-1, …, 1.

The initial condition and boundary condition are fixed.

$$dy^0 = 0. \quad (3)$$

The boundary condition is as follows.

$$\lambda^{L+1} = 0 \quad \text{.....(4)}$$

The gradient matrix, $\left[\nabla_{y^n}(f^{n+1})^T\right]$ is the derivative of accumulation term at n, because all terms in f^{n+1} are independent to the primary variables at n except for the accumulation at n as shown in **Eq. 5**.

$$\frac{\partial f^{n+1}}{\partial y^n} = \frac{\partial\left\{(Ttans\times\nabla p)^{n+1}-\left(Accum^{n+1}-Accum^n\right)-(Sink/Source)^{n+1}\right\}}{\partial y^n} \quad \text{.....(5)}$$

Hence,

$$\frac{\partial(Accum_{o,i,j}{}^n)}{\partial p_{i,j}^n} = \Delta x_{i,j}\Delta y_{i,j}\Delta z_{i,j}\frac{1}{\Delta t_{n-1}}S_{o,i,j}{}^n\left(\frac{\frac{\partial\varphi_{i,j}^n}{\partial p_{i,j}^n}B_{o,i,j}{}^n-\varphi_{i,j}^n\frac{\partial B_{o,i,j}{}^n}{\partial p_{i,j}^n}}{B_{o,i,j}{}^{n^2}}\right), \quad \text{.....(6)}$$

$$\frac{\partial(Accum_{w,i,j}{}^n)}{\partial p_{i,j}^n} = \Delta x_{i,j}\Delta y_{i,j}\Delta z_{i,j}\frac{1}{\Delta t_{n-1}}S_{w,i,j}{}^n\left(\frac{\frac{\partial\varphi_{i,j}^n}{\partial p_{i,j}^n}B_{w,i,j}{}^n-\varphi_{i,j}^n\frac{\partial B_{w,i,j}{}^n}{\partial p_{i,j}^n}}{B_{w,i,j}{}^{n^2}}\right), \quad \text{.....(7)}$$

$$\frac{\partial(Accum_{o,i,j}{}^n)}{\partial S_{w,i,j}^n} = -\Delta x_{i,j}\Delta y_{i,j}\Delta z_{i,j}\frac{1}{\Delta t_{n-1}}\left(\frac{\varphi_{i,j}^n}{B_{o,i,j}{}^n}\right), \quad \text{.....(8)}$$

$$\frac{\partial(Accum_{w,i,j}{}^n)}{\partial S_{w,i,j}^n} = \Delta x_{i,j}\Delta y_{i,j}\Delta z_{i,j}\frac{1}{\Delta t_{n-1}}\left(\frac{\varphi_{i,j}^n}{B_{w,i,j}{}^n}\right) \quad \text{.....(9)}$$

The gradient matrix, $\left[\nabla_{y^n}(f^{n+1})^T\right]$ form the following matrix

$$\nabla_{y^n}(f^{n+1})^T = \frac{\partial(Accum^n)^T}{\partial y^n} = \begin{bmatrix} \frac{\partial(Accum_o{}^n)^T}{\partial p^n} & \frac{\partial(Accum_w{}^n)^T}{\partial p^n} \\ \frac{\partial(Accum_o{}^n)^T}{\partial S_w{}^n} & \frac{\partial(Accum_w{}^n)^T}{\partial S_w{}^n} \end{bmatrix} \quad \text{.....(10)}$$

Because β is assumed as bottomhole pressure at injector and water cut at producer in this project, $\nabla_{y^n}\beta$ can be analytically calculated by using following formulation

$$WCT = \frac{k_{rw}\mu_o B_o}{k_{ro}\mu_w B_w + k_{rw}\mu_o B_o}, \quad \text{.....(11)}$$

$$p_{wf,i,j} = p_{i,j} + \frac{ln\left(r_{o,i,j}/r_{w,i,j}\right)+s_{i,j}}{(2\pi)1.127\times10^{-3}hk_{i,j}}\left(\frac{B_{m,i,j}\mu_{m,i,j}}{k_{rm,i,j}}\right)q_{m,i,j}, \quad \text{.....(12)}$$

$$r_{o,i,j} = 0.14036\sqrt{\Delta x^2+\Delta y^2} \quad \text{.....(13)}$$

The adjoint system is ended up computing as shown in **Figure 2**. The sensitivity coefficients for *J* are given by,

$$\nabla_m J = \nabla_m\beta + \sum_{n=1}^{L}[\nabla_m(f^n)^T](\lambda^n) \quad \text{.....(14)}$$

We already know λ^n from the computation of **Eq. 1**. The gradient matrix, $[\nabla_m(f^n)^T]$ is $M \times 2N$ sparse matrix as follows

$$\nabla_k(f^n)^T = \nabla_k[(Ttans\times\nabla p)^n]^T = \begin{bmatrix} \frac{\partial f_{o,1}{}^n}{\partial k_1} & \frac{\partial f_{w,1}{}^n}{\partial k_1} & \frac{\partial f_{o,2}{}^n}{\partial k_1} & \frac{\partial f_{w,2}{}^n}{\partial k_1} & \cdots & \frac{\partial f_{o,N}{}^n}{\partial k_1} & \frac{\partial f_{w,N}{}^n}{\partial k_1} \\ \frac{\partial f_{o,1}{}^n}{\partial k_2} & \frac{\partial f_{w,1}{}^n}{\partial k_2} & \frac{\partial f_{o,2}{}^n}{\partial k_2} & \frac{\partial f_{w,2}{}^n}{\partial k_2} & & \frac{\partial f_{o,N}{}^n}{\partial k_2} & \frac{\partial f_{w,N}{}^n}{\partial k_2} \\ \vdots & & \vdots & & & \vdots & \\ \frac{\partial f_{o,1}{}^n}{\partial k_M} & \frac{\partial f_{w,1}{}^n}{\partial k_M} & \frac{\partial f_{o,2}{}^n}{\partial k_M} & \frac{\partial f_{w,2}{}^n}{\partial k_M} & \cdots & \frac{\partial f_{o,N}{}^n}{\partial k_M} & \frac{\partial f_{w,N}{}^n}{\partial k_M} \end{bmatrix} \quad \text{.....(15)}$$

Figure 3 shows the computation of sensitivity matrix.

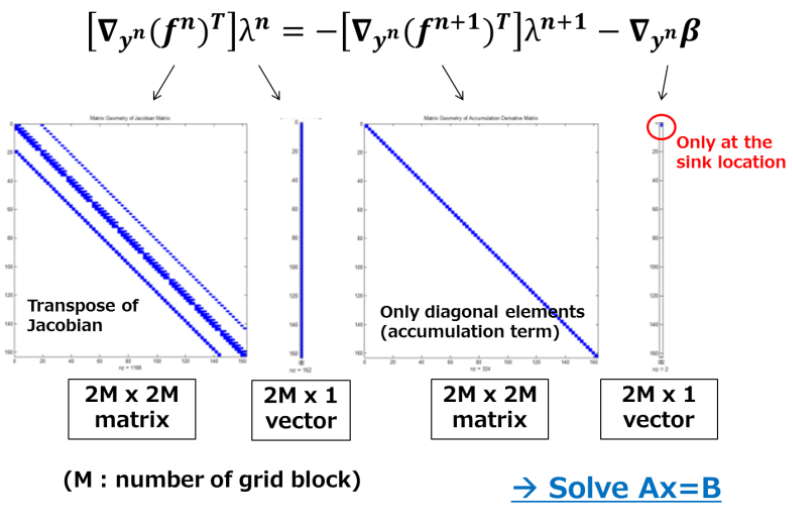

Figure 2—Computation of adjoint system.

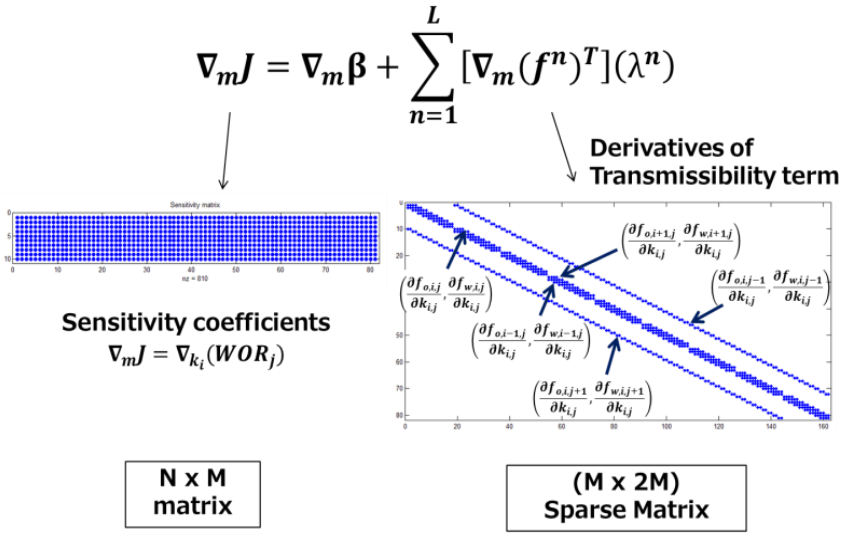

Figure 3—Computation of Sensitivity Matrix.

Minimization

The Levenberg–Marquardt algorithm, which was independently developed by Kenneth Levenberg and Donald Marquardt, provides a numerical solution to the problem of minimizing a nonlinear function. It is fast and has stable convergence. In the artificial neural-networks field, this algorithm is suitable for training small- and medium-sized problems.

The Levenberg–Marquardt algorithm blends the steepest descent method and the Gauss–Newton algorithm. Fortunately, it inherits the speed advantage of the Gauss–Newton algorithm and the stability of the steepest descent method. It's more robust than the Gauss–Newton algorithm, because in many cases it can converge well even if the error surface is much more complex than the quadratic situation. Although the Levenberg Marquardt algorithm tends to be a bit slower than Gauss–Newton algorithm (in convergent situation), it converges much faster than the steepest descent method. The basic idea of the Levenberg–Marquardt algorithm is that it performs a combined training process: around the area with complex curvature, the Levenberg–Marquardt algorithm switches to the steepest descent algorithm, until the local curvature is proper to make a quadratic approximation; then it approximately becomes the Gauss–Newton algorithm, which can speed up the convergence significantly.

Gauss–Newton Algorithm. The relationship between Hessian matrix *H* and Jacobian matrix *J* can be rewritten as,

$$H = J^T J \quad (16)$$

Levenberg's Contribution. Replace the above equation by a "damped version",

$$H = J^T J + \lambda I, \quad (17)$$

Where λ is always positive, called combination coefficient. *I* is the identity matrix
Levenberg's algorithm has the disadvantage that if the value of damping factor, λ, is large, inverting $J^T J + \lambda I$ is not used at all.

Marquardt Modification. Replaced the identity matrix, *I*, with the diagonal matrix consisting of the diagonal elements of $J^T J$, resulting in the Levenberg–Marquardt algorithm

$$H = J^T J + \lambda diag(J^T J) \quad (18)$$

From above equation, one may notice that the elements on the main diagonal of the approximated Hessian matrix will be larger than zero. Therefore, with this approximation, it can be sure that matrix *H* is always invertible.

For our problem, we drive the objective function without prior information

$$O(m) = \frac{1}{2}[g(m) - d_{obs}]^T {C_D}^{-1}[g(m) - d_{obs}], \quad (19)$$

$$O(m + \delta m) = O(m) + J^T e\delta m + \frac{1}{2}\delta m^T J^T J\delta m \quad (20)$$

Set

$$\frac{\partial O(m+\delta m)}{\partial \delta m} \approx 0 \quad (21)$$

We have

$$J^T J\delta m = -J^T e \quad (22)$$

Hence, apply Levenberg-Marquardt Algorithm, we get

$$\delta m = -[J^T J + \lambda diag(J^T J)]^{-1}(J^T e), \quad (23)$$

Where,

$$O(m) = \frac{1}{2}e^T e, \quad (24)$$

$$e = C_D^{1/2}[g(m) - d_{obs}], \quad (25)$$

$$J = \frac{\partial e}{\partial m}, \quad (26)$$

$$J = C_D^{-1/2}\frac{\partial g(m)}{\partial m} = C_D^{-1/2} G_l, \quad (27)$$

$$J^T J = G_l^T C_D^{-1} G_l \quad (28)$$

The simplest way to obtain the correction δm is to use Cholesky decomposition on the linear system. The main advantage of the nodal equations is speed.

$$m^{l+1} = m^l + \alpha\delta m^{l+1}, \quad (29)$$

Where, $\boldsymbol{l}$ is iteration number.

The (non-negative) damping factor, $\boldsymbol{\lambda}$, is adjusted at each iteration.

$$O(m^{l+1}) > O(m^l) \rightarrow \lambda^{l+1} = \rho\lambda^l, \quad (30)$$

$$O(m^{l+1}) < O(m^l) \rightarrow \lambda^{l+1} = \frac{\lambda^l}{\rho}, \quad (31)$$

where $\boldsymbol{\rho} > 1$; $\boldsymbol{\lambda^1}$ is between $\sqrt{\frac{O(m^0)}{N_d}}$ and $\frac{O(m^0)}{N_d}$; f $\boldsymbol{\lambda} = 0$, we have Gauss-Newton Method.

If the reduction of objective function is rapid, a smaller value can be used, bringing the algorithm closer to the Gauss-Newton Algorithm, whereas if one iteration gives insufficient reduction in the residual, $\boldsymbol{\lambda}$ can be increased, giving a step closer to the gradient descent direction.

Result

Figure 4 and **5** show the WCT and BHP sensitivity comparison between our adjoint sensitivity calculation and sensitivity calculated using perturbation. We are able to capture the trend of both WCT and BHP, the relative difference is generally less than 10%.

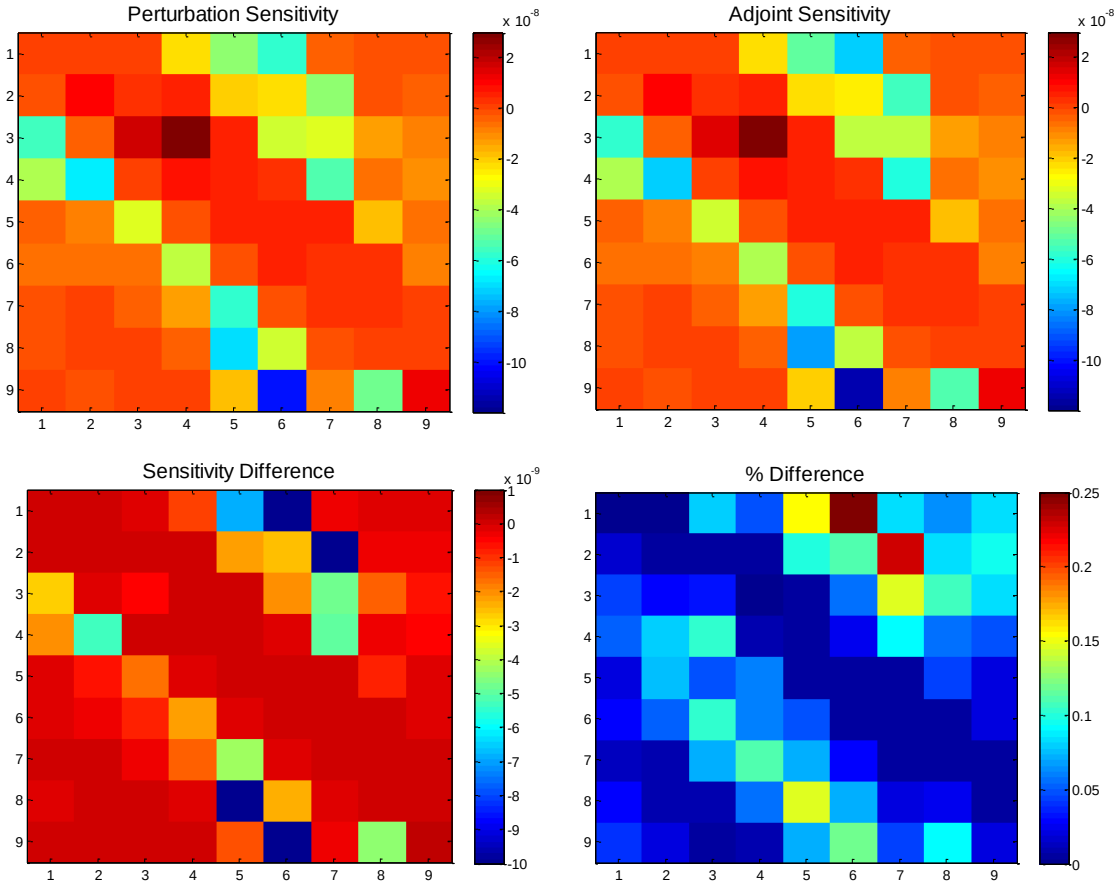

Figure 4—Sensitivity comparison of WCT.

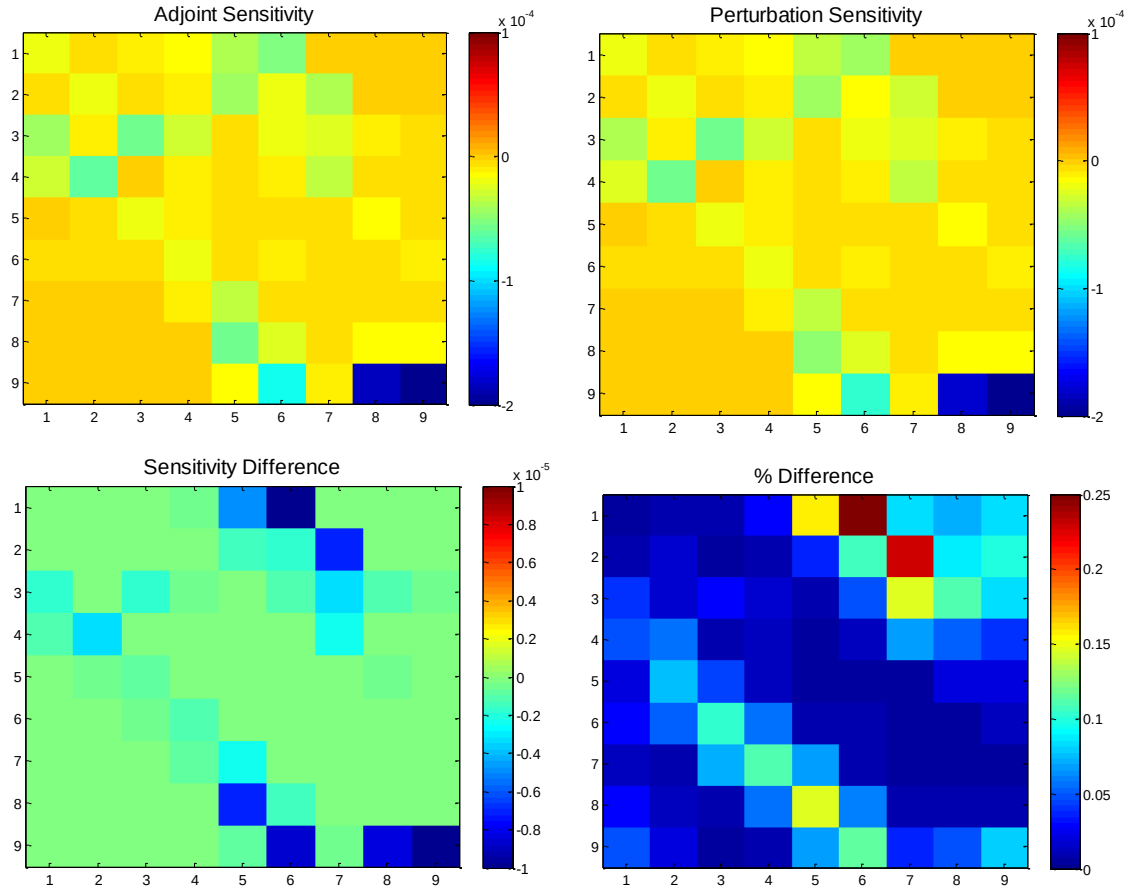

Figure 5—Sensitivity comparison of BHP.

Figure 6 and **7** shows the model calibration history and results. We are able to match both BHP and WCT quite well. The updated permeability field keeps the high permeability region in the center. It does not give quite similar results mainly due to the non-uniqueness of solution.

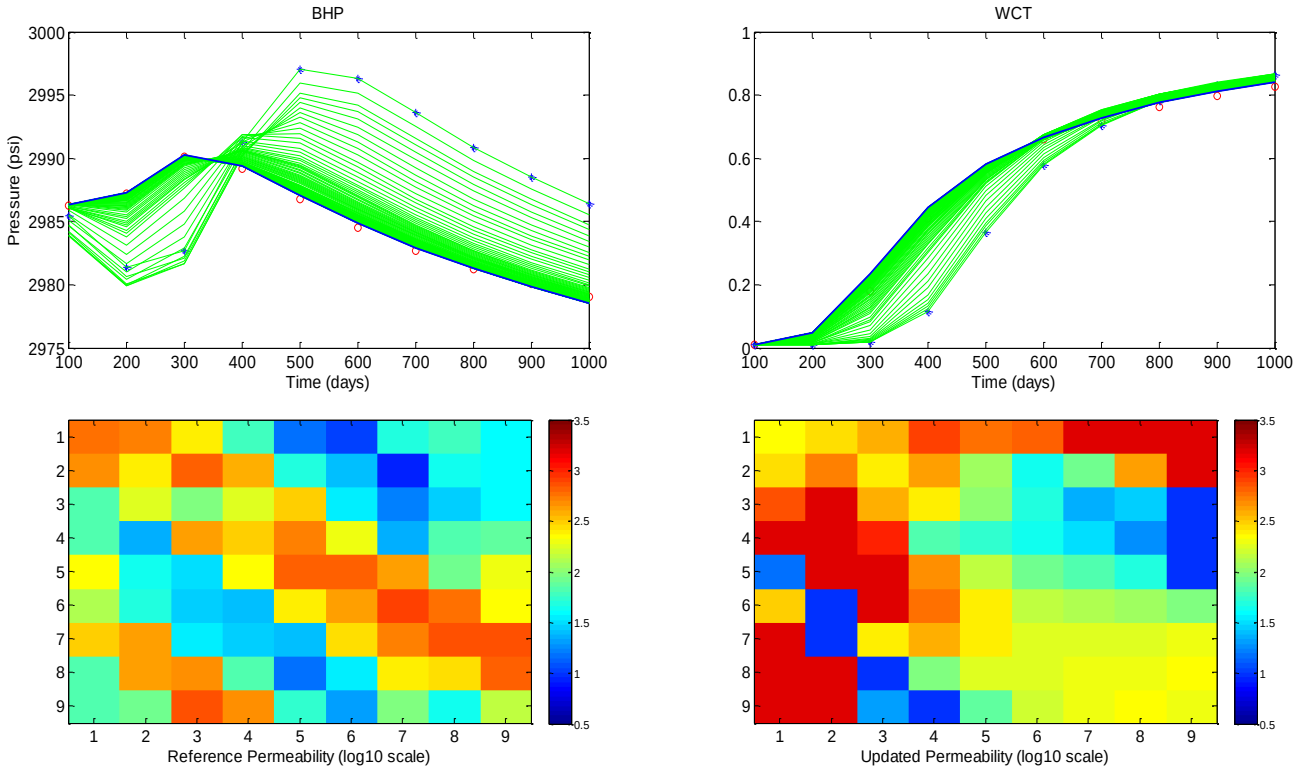

Figure 6—History of model calibration (BHP and WCT).

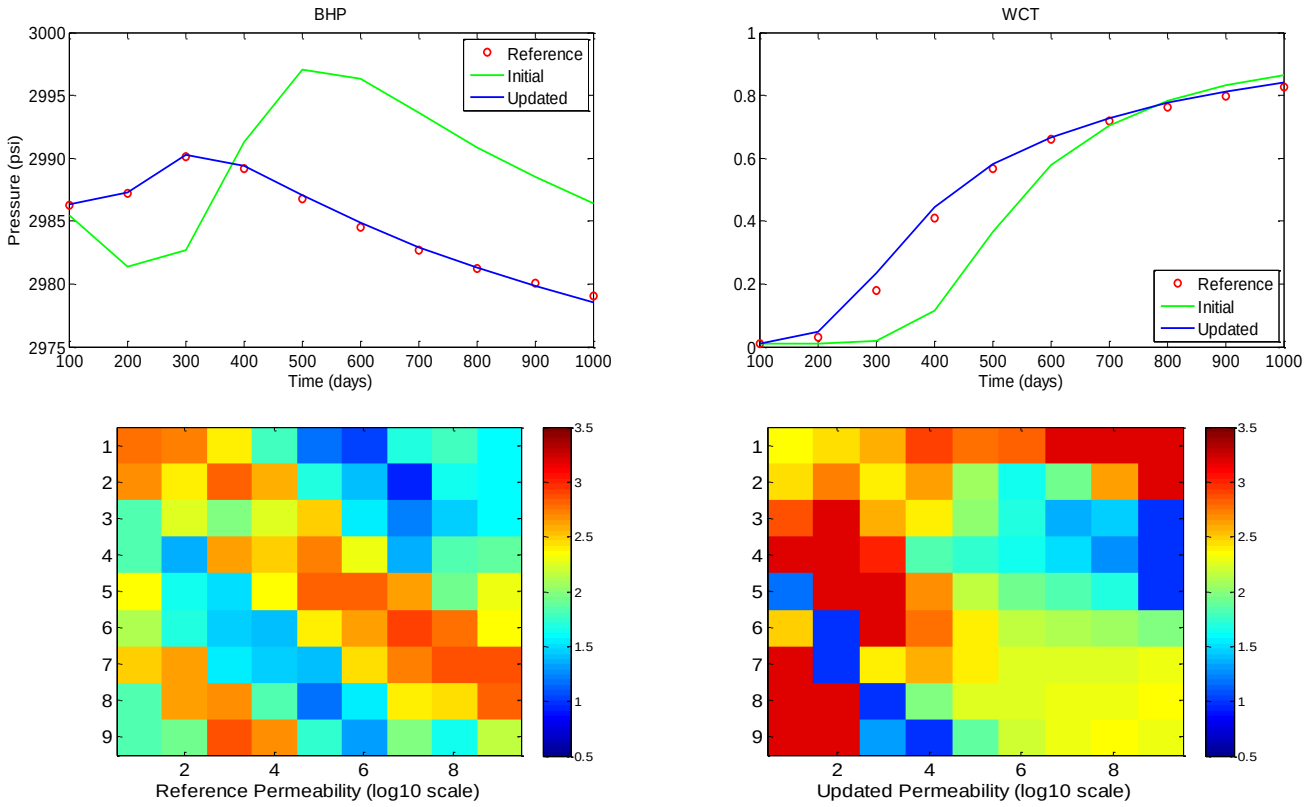

Figure 7—Model calibration results (BHP and WCT).

Figure 8 shows the history of objectives. The data misfit decrease dramatically, even in the semilog plot.

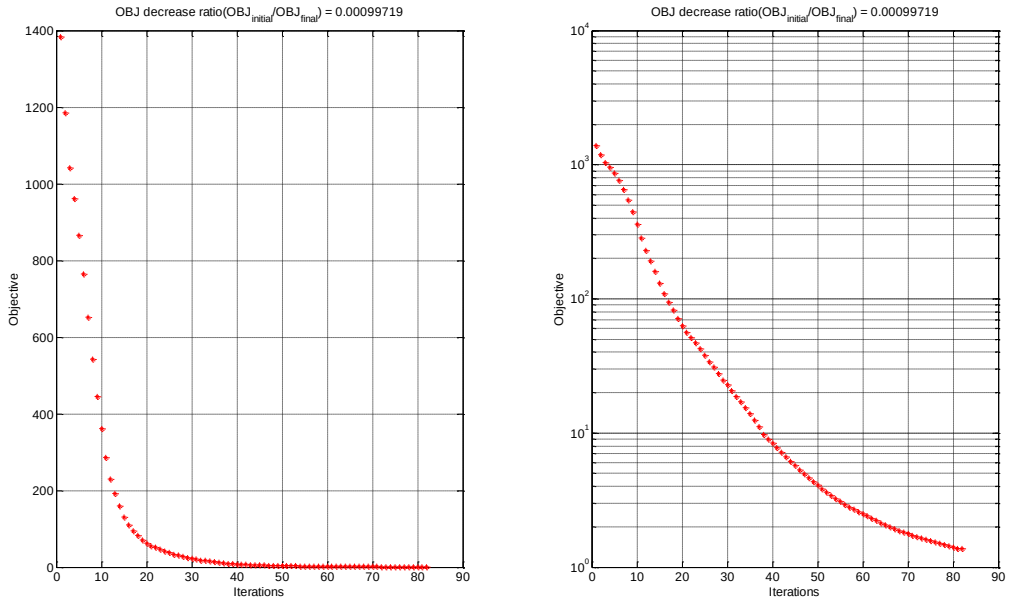

Figure 8—History of objective (normal and semilog coordinate).

Physical Explanation of Sensitivity

The effect of permeability change on the BHP and WOR is studied. **Figure 9** shows the sensitivity of BHP with the permeability.

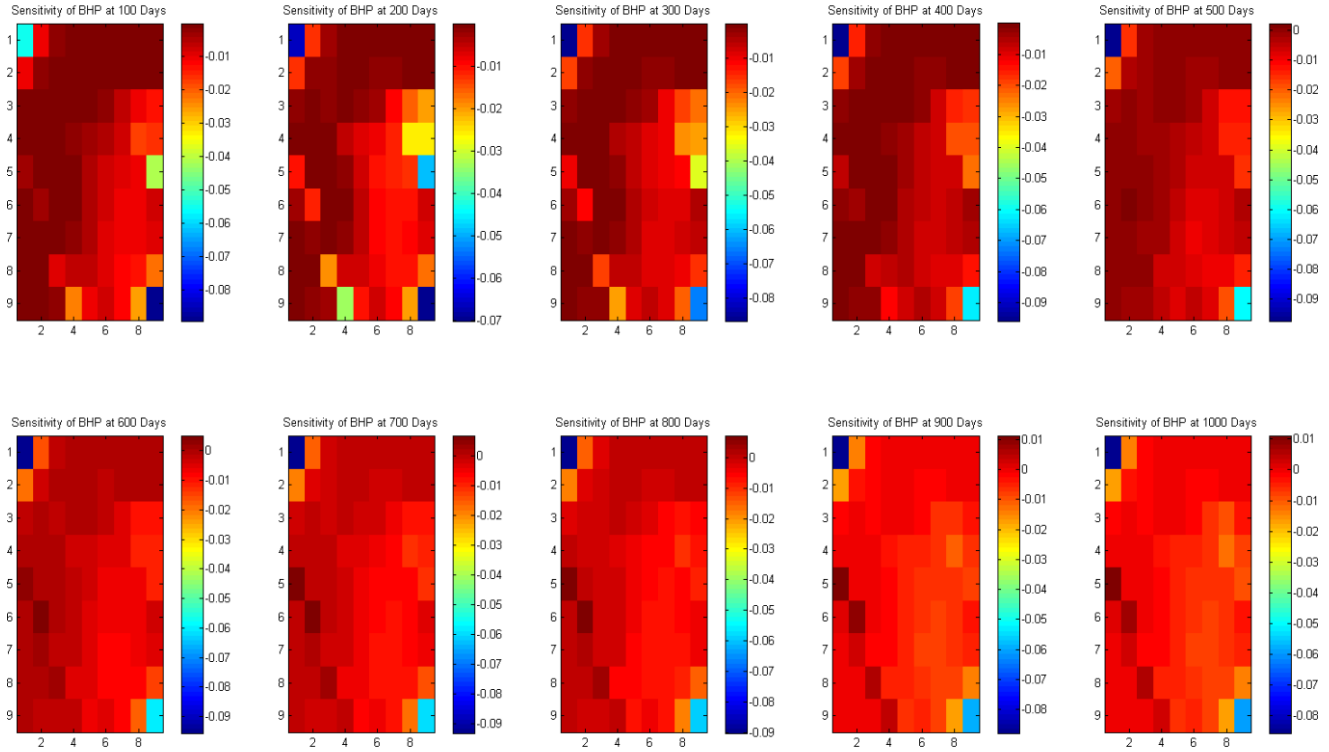

Figure 9—Sensitivity of BHP.

We can see from the colorbar that most the grid cells have negative value, which means if the permeability increases, the BHP decreases. It is easy to understand that higher permeability is more conductive for water than lower permeability. Besides, with the time increasing, the cells around the producer are becoming more sensitive to the BHP. **Figure 10** shows the sensitivity of WOR with the permeability.

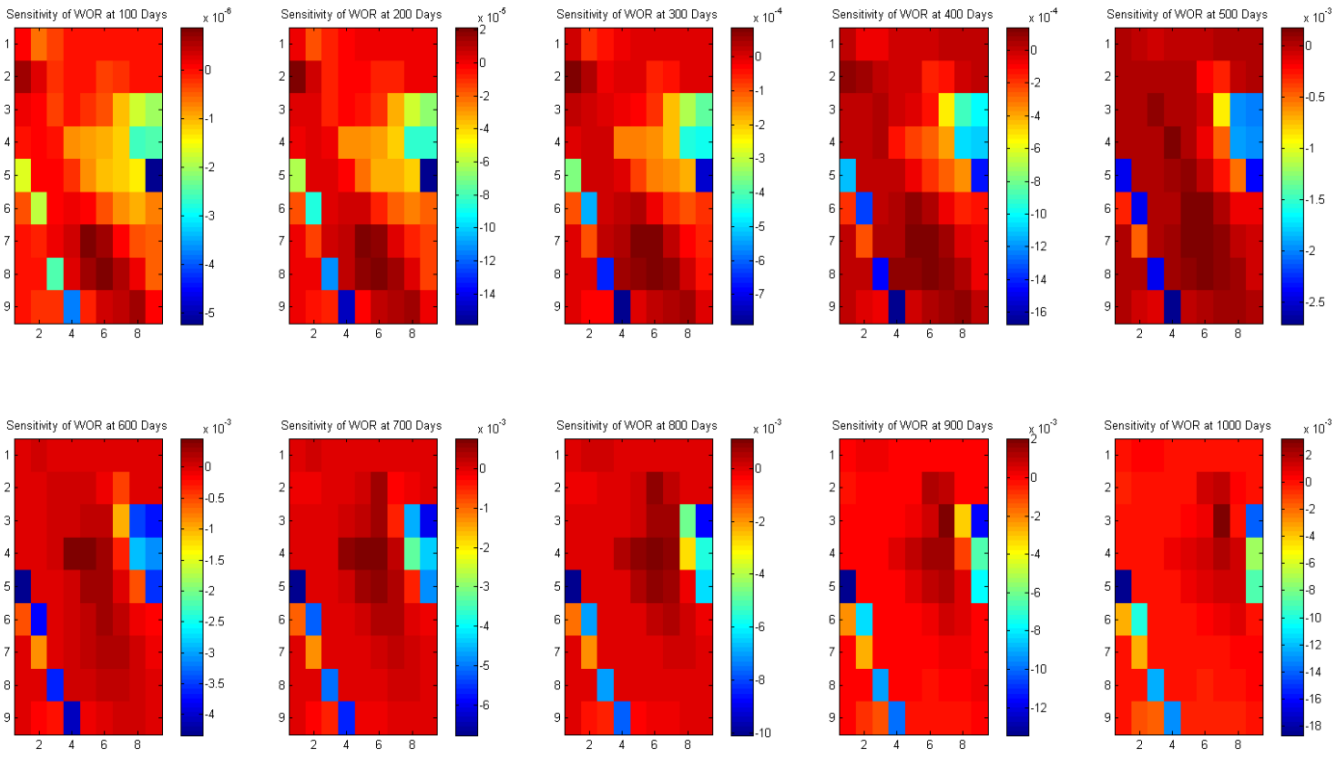

Figure 10—Sensitivity of WOR.

We can see there is a channel connecting injector and producer. The sensitivity of these channel cells are positive, which means if the permeability increases in these areas, the WOR increases. More water tends to flow in this channel. The sensitivity of cells in the edge part is negative, meaning higher permeability in these areas can decrease the WOR. More water flow to this area.

Effect of α and λ in the LM minimization process. α is the searching step in the LM algorithm. Small α results in more iterations while large α may lead to convergence failure. Large searching step may make the derivative stepping cross the minimization point. **Figure 11** shows the comparison of history matching process between α=0.01 and 3.

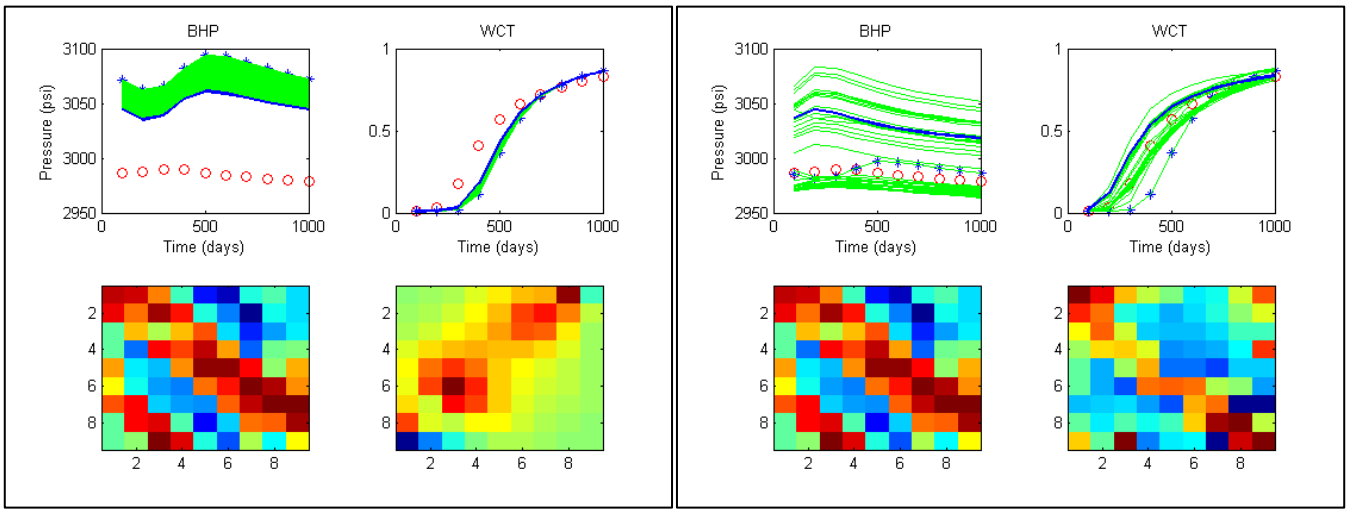

Figure 11—Comparison of history matching process between α=0.01 and 3.

The left part of Figure 11 shows the history matching process with α equals 0.01. We can see the converging process is slow. The blue line is the final curve. It is limited by the iteration numbers. If given sufficient iterations, it can converge. The right part shows history matching process with α equals 3. We can see the matching curve jumps around, and the convergence is failed.

Figure 12 below shows the objective function behavior with iteration numbers for nine different α. We can see if α equals 0.2, the objective function converges slowly and smoothly, although 68 iterations are needed. But if the α equals 3, the objective function fluctuates without tendency to converge. These are two extreme cases. Additionally, if α equals 1.7, it converges very fast with only 8 iterations.

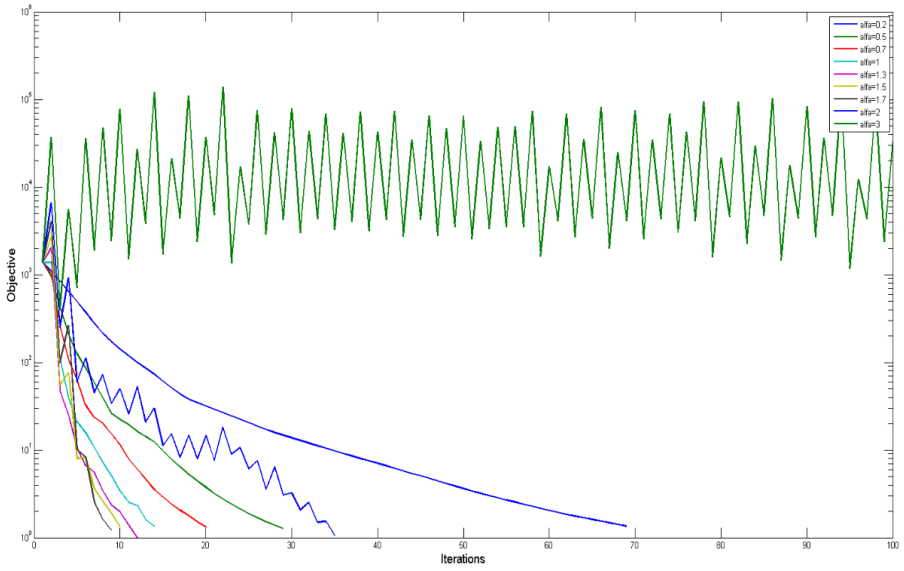

Figure 12—Objective function behavior with iteration numbers for different α.

λ in the LM algorithm is the perturbation added to the original function, to avoid the singular matrix. If λ is too small, the original function may still be singular, while if λ is big, more iterations will be needed to converge the objective function. **Figure 13** below shows the trend.

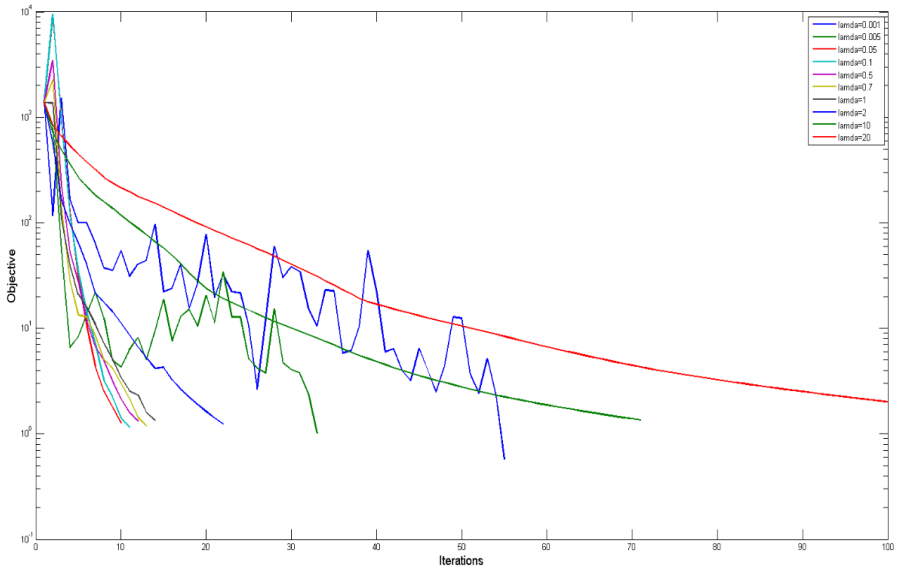

Figure 13—Objective function behavior with iterations for different λ.

We can see if the λ equals 0.001, the objective function fluctuates a lot, and the code gives the warning that the objective matrix is badly conditioned. The result is not correct. For λ of 10, it converges slowly and smooth, but 72 iterations are needed. For λ of 20, it does not converge with the limited iterations, but it will converge given sufficient iterations. Additionally, λ of 0.5 gives very fast convergence. We can see both α and λ affect the history matching process, and good combination of them is desired.

Conclusions

1. We successfully incorporate the sensitivity calculation with adjoint method into the software, and the sensitivity results are quite close to the results calculated by perturbation.
2. The adjoint method only need one-time simulation to give the sensitivity calculation, instead of M times (the number of parameters), which show great advantages in computational efficiency.
3. We can carry out history matching using Levenberg-Marquardt method to match both WCT and BHP. We are also able to keep the high permeability trend of the permeability field.
4. More suitable search step lengths and perturbation in Levenberg-Marquardt method will give faster converge. Integrated with adjoint method, it is a fast and accurate history matching workflow.

Conflicts of Interest

The author(s) declare that they have no conflicting interests.

References

Wu, Z., Reynolds, A.C., and Oliver, D.S. 1999. Conditioning Geostatistical Models to Two-Phase Production Data. Paper presented at SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, September 27-30. SPE-56855-MS.

Li, R., Reynolds, A.C., and Oliver, D.S. 2003. History Matching of Three-Phase Flow Production Data. *SPE Journal* **8**(4):25-31. SPE-87336-PA.

Oliver, D.S., Reynolds, A.C., and Liu, N. 2008. Inverse Theory for Petroleum Reservoir Characterization and History Matching: Front Matter. *Journal of Affective Disorders* **2008**(175):488-493.

Jacquard, P. 1965. Permeability Distribution from Field Pressure Data. *SPE Journal* **5**(4): 281-294. SPE-1307-PA.

Jahns, O.H. 1966. A Rapid Method for Obtaining a Two Dimensional Reservoir Description from Well Pressure Response Data. *SPE Journal* **6**(4): 315-327.SPE-1473-PA

Chavent, G. and Bissell, R. 1998. *Indicator for the Refinement of Parameterization in Inverse Problems in Engineering Mechanics*. Ed. M. Tanaka and G.S. Dulikravich, 309-314. Oxford, UK: Elsevier Science.

Grimstad, A.A., Mannseth, T., Nævdal, G., et al. 2003. Adaptive Multiscale Permeability Estimation. *Computational Geosciences* **7**(1): 1-25.

Sahni, I. and Horne, R.N. 2005. Multiresolution Wavelet Analysis for Improved Reservoir Description. *SPE Reservoir Evaluation & Engineering* **8**(1): 53-69. SPE-87820-PA.

Hao Wang is a petroleum engineer at Liaohe Oilfield Company, a subsidiary of PetroChina. Wang's specialties include viscous oil development technologies, like SAGD and steam drive. Wang holds a bachelor's degree in petroleum engineering from China University of Petroleum, Beijing, and master's degree in petroleum engineering from Northeast Petroleum University.