Energy 224 Lecture: Calculating Sensitivity Coefficients for Gradient-based History Matching

Mehrdad Gharib Shirangi

November 27, 2012

History Matching as an Inverse Problem

In forward problems, the evolutionary state of a system is predicted from some predictive model, given auxiliary conditions and physical properties. Physical properties are referred to as model parameters. One example of a forward problem is to predict pressures and saturations of gridblocks and well rates of a reservoir versus time, given the initial state and the geometry of the reservoir, the rock property fields and fluid properties.

In an inverse problem, given a finite number of observed data which are functions of state variables, one aims to infer information about model parameters. Observed data contain measurement error.

History matching is a discrete inverse problem which is characterized by a finite number of model parameters. The history matching process consists of estimating reservoir properties through matching predicted data to reservoir production history (observed data).

It is well-known that large scale inverse problems are usually ill-posed as there are conceptually an infinite number of models that match the data. Three conditions for a well-posed problem are existence, uniqueness and stability of the solution or solutions.

Example

Forward problem: given the reservoir properties (permeability and porosity distribution, initial fluid contacts, fluid properties, etc.) calculate the cumulative oil and water production versus time.

Inverse problem: given the geostatistical data (prior knowledge) and production history, calculate the P10 and P90 of permeability/porosity value at the i, j, kth gridblock.

A Priori and A Posteriori Probability Density Functions

Here the N_m-dimensional vector of model parameters is denoted by m and it can include horizontal and vertical log permeability and porosity of gridblocks. Permeability fields are assumed to have a log-normal distribution, while porosity fields are assumed to have normal distribution.

The prior uncertainty in the model parameters is described by a pdf. If the prior pdf has a Gaussian distribution, it can be fully described by its mean and the covariance. To show a Gaussian prior distribution, with m denoting the random vector of model parameters, the following notation is used:

$$m \sim N(m_{\text{prior}}, C_M), \tag{1}$$

which means the random vector m has a normal (Gaussian) distribution with mean m_{prior} and covariance matrix C_M. The prior pdf is given by

$$f(m) = a_1 \exp\big[-\frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}})\big], \tag{2}$$

where a_1 is the normalizing constant. The N_d-dimensional column vector of predicted data d is related to the vector of model parameters m by

$$d = g(m). \tag{3}$$

In the problems of interest to us, $g(m)$ is the simulator output when it is ran with the given m, e.g. well by well phase rates.

If m is the vector of true model parameters, then d is referred to as true data. However, what is available is observed data which is the true data corrupted with measurement error. The difference between true data, d_{true}, and the corresponding vector of observed data, d_{obs}, represents measurement error ϵ^d, i.e.,

$$\epsilon^d = d_{\text{true}} - d_{\text{obs}}. \tag{4}$$

It is usually reasonable to assume that measurement error has a Gaussian distribution with mean zero and an $N_d \times N_d$ covariance matrix C_D. In other words, observed data is a random vector that has a Gaussian distribution with mean of $d_{\text{true}} = g(m_{\text{true}})$ and covariance matrix C_D. The Gaussian pdf is expressed as

$$f(d_{\text{obs}}|m) = f(\epsilon^d) = a_2 \exp\big[-\frac{1}{2}(d_{\text{obs}} - g(m))^T C_D^{-1}(d_{\text{obs}} - g(m))\big], \tag{5}$$

where a_2 is the normalizing constant. This pdf characterizes the uncertainty in observed data, d_{obs}, given the model parameters, m. As d_{obs} is given, Eq. 5 gives the likelihood of m given d_{obs} denoted by $L(m|d_{\text{obs}})$; the value of $L(m|d_{\text{obs}})$ is greater if the predicted data corresponding to the model m is closer to d_{obs}. If the data includes only production data (and not seismic data), then the $N_d \times N_d$ covariance matrix for measurement error is diagonal, i.e.,

$$C_D = \begin{bmatrix} \sigma_{d,1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{d,2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{d,N_d}^2 \end{bmatrix}, \tag{6}$$

where, $\sigma_{d,i}^2$, $i = 1, \dots N_d$, is the variance of the measurement error of the ith observed data.

According to the Bayes Theorem (Tarantola, 1987; Oliver et al., 2008), the posterior pdf of the model parameters conditional to the observed data is proportional to the product of the prior pdf and the likelihood function for the model parameters:

$$f(m|d_{\text{obs}}) \propto f(m) L(m|d_{\text{obs}}). \tag{7}$$

Using Eqs. 2 and 5 in Eq. 7, the posterior pdf of model parameters conditioned to observed data can be written as

$$f(m|d_{\text{obs}}) = a \exp(-O(m)), \tag{8}$$

where a is the normalizing constant and $O(m)$ which is referred to as the total objective function, is

$$\begin{aligned} O(m) = \frac{1}{2}(m - m_{\text{prior}})^T C_M^{-1}(m - m_{\text{prior}}) + \frac{1}{2}(g(m) - d_{\text{obs}})^T C_D^{-1}(g(m) - d_{\text{obs}}) \\ = O_m(m) + O_d(m). \end{aligned} \tag{9}$$

The total objective function has two parts, the model mismatch part, O_m, and the data mismatch term, O_d. The model mismatch part, which comes from the prior pdf, provides regularization to avoid unrealistic changes in model parameters.

The Sensitivity Matrix

One way to minimize the objective function of Eq. 9 is using the gradient based algorithms, in particular the Gauss-Newton method. In the Gauss-Newton method of minimizing the objective function $O(m)$, given by Eq. 9, the search direction δm^{l+1} at the lth iteration, is generated from

$$H_l \delta m^{l+1} = -\nabla O^l, \tag{10}$$

where ∇O^l is the gradient of $O(m)$ evaluate at m^l, given by

$$\nabla O^l = C_M^{-1}(m^l - m_{\text{prior}}) + G_l^T\{C_D^{-1}(g(m^l) - d_{\text{obs}})\}, \tag{11}$$

and H_l is the GN-Hessian matrix, given by

$$H_l = C_M^{-1} + G_l^T C_D^{-1} G_l. \tag{12}$$

G is the matrix of total derivatives of the predicted data, g_i, $i = 1, 2, \ldots, N_d$, with respect to the model parameters, which is called the sensitivity matrix (sometimes Jacobian) and is defined as

$$G = \left[G_{i,j}\right] = \left[\frac{\partial g_i}{\partial m_j}\right], \tag{13}$$

for $i = 1, 2, \ldots, N_d$ and $j = 1, 2, \ldots, N_m$. Thus G is an $N_d \times N_m$ matrix. Here and in the rest of this thesis, G denotes the sensitivity matrix, given by

$$G = \begin{bmatrix} \frac{\partial g_1}{\partial m_1} & \frac{\partial g_1}{\partial m_2} & \cdots & \frac{\partial g_1}{\partial m_{N_m}} \\ \frac{\partial g_2}{\partial m_1} & \frac{\partial g_2}{\partial m_2} & \cdots & \frac{\partial g_2}{\partial m_{N_m}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_{N_d}}{\partial m_1} & \frac{\partial g_{N_d}}{\partial m_2} & \cdots & \frac{\partial g_{N_d}}{\partial m_{N_m}} \end{bmatrix}, \tag{14}$$

where the (i, j) entry of the matrix G, denoted as $\frac{\partial g_i}{\partial m_j}$, is the total derivative of the ith predicted data with respect to the jth model parameter.

The product of G times an arbitrary N_d-dimensional vector v, which is a linear combination of the columns of G, can be computed with the "gradient simulator method" Anterion et al. (1989). The product of G^T times an arbitrary N_m-dimensional vector u, which is a linear combination of the rows of G, can be computed with the "adjoint method" (Li et al., 2003; Zhang and Reynolds, 2002). These methods are discussed in Oliver et al. (2008).

The whole sensitivity matrix, G, can be formed by either N_m applications of the gradient simulator method or N_d adjoint solutions. If both N_m and N_d are very large, forming the matrix G can be computationally very expensive; because of this fact, a direct application of the GN algorithm to large scale problems is not efficient.

For the nonlinear conjugate gradient method (Makhlouf et al., 1993; Chu et al., 2000; Kalita, 2000), and for the quasi-Newton methods, explicit computation of the full sensitivity matrix is not necessary. Both algorithms use the gradient information, and to apply them, only the product of G times a vector and/or the product of G^T times a vector are required. Among the quasi-Newton methods, the limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) method has been successfully applied to history matching problems (Zhang and Reynolds, 2002; Gao and Reynolds, 2006).

Adjoint Method

Consider the simulation equations:

$$f^{n+1} = f^{n+1}(y^{n+1}, y^n, m) = 0, \quad n = 0, 1, 2, \ldots L-1 \tag{15}$$

y^n denotes the N_y-dimensional vector of primary variables that we solve for in the simulator at the nth time-step (t_n), i.e. pressure, saturation or R_s of all gridblocks in a black oil simulator. Note that for the fully implicit simulator, y^n contains the converged values of primary variables computed at the last Newton-Raphson iteration. L is the number of time-steps. Eq. 15 represents all the N_y equations that we solve to obtain y^{n+1} after having obtained y^n. m denotes the N_m-dimensional vector of model parameters, e.g. log-permeability and porosity of all gridblocks.

Our objective is to calculate the sensitivity of β with respect to all model parameters, i.e., $\partial\beta/\partial m_i,\ i = 1, \dots N_m$. The derivation here follows that of (Li et al., 2003; Oliver et al., 2008).

β is a scalar function of predicted data,

$$\beta = \beta(y^1, y^2, \dots, y^L, m). \tag{16}$$

Adjoint all the simulator equations $f^1, f^2, \dots f^L$ using vector of Lagrange multipliers, $\lambda^n,\ n = 1, \dots, L+1$:

$$J = \beta + \sum_{n=0}^{L} \lambda_{n+1}^T f^{n+1}(y^{n+1}, y^n, m). \tag{17}$$

Note that each λ_{n+1} is a N_y-dimensional column vector, i.e.,

$$\lambda^{n+1} = [\lambda_{n+1,1}, \lambda_{n+1,2}, \dots, \lambda_{n+1,N_y}]^T. \tag{18}$$

There are a total of $N_y \times (L+1)$ flow equations, and for each flow equation there is one Lagrange multiplier.

Take the total derivative:

$$\begin{aligned} dJ = d\beta + \sum_{n=0}^{L} \lambda_{n+1}^T df^{n+1} = d\beta + \sum_{n=0}^{L} \lambda_{n+1}^T [\nabla_{y^{n+1}} (f^{n+1})^T]^T dy^{n+1} \\ + \sum_{n=0}^{L} \lambda_{n+1}^T [\nabla_{y^n} (f^{n+1})^T]^T dy^n + \sum_{n=0}^{L} \lambda_{n+1}^T [\nabla_m (f^{n+1})^T]^T dm. \end{aligned} \tag{19}$$

One can rewrite the first summation as

$$\begin{aligned} &\sum_{n=0}^{L} \lambda_{n+1}^T [\nabla_{y^{n+1}} (f^{n+1})^T]^T dy^{n+1} \\ &\qquad = \lambda_{L+1}^T [\nabla_{y^{L+1}} (f^{L+1})^T]^T dy^{L+1} + \sum_{n=0}^{L-1} \lambda_{n+1}^T [\nabla_{y^{n+1}} (f^{n+1})^T]^T dy^{n+1} \\ &= \lambda_{L+1}^T [\nabla_{y^{L+1}} (f^{L+1})^T]^T dy^{L+1} + \sum_{n=1}^{L} \lambda_n^T [\nabla_{y^n} (f^n)^T]^T dy^n = \sum_{n=1}^{L} \lambda_n^T [\nabla_{y^n} (f^n)^T]^T dy^n, \end{aligned} \tag{20}$$

where we have set $\lambda_{L+1} = 0$.

Similarly, as $\lambda_{L+1} = 0$, the third summation in (19) can be written as

$$\sum_{n=0}^{L} \lambda_{n+1}^T [\nabla_m (f^{n+1})^T]^T dm = \sum_{n=1}^{L} \lambda_n^T [\nabla_m (f^n)^T]^T dm. \tag{21}$$

Using (20) and (21) in (19) and expanding $d\beta$, one can obtain

$$dJ = (\nabla_m \beta)^T dm + \sum_{n=1}^{L} (\nabla_{y^n} \beta)^T dy^n + \sum_{n=1}^{L} \lambda_n^T [\nabla_{y^n} (f^n)^T]^T dy^n + \lambda_1^T [\nabla_{y^0} (f^1)^T]^T dy^0 + \sum_{n=1}^{L} \lambda_{n+1}^T [\nabla_{y^n} (f^{n+1})^T]^T dy^n + \sum_{n=1}^{L} \lambda_n^T [\nabla_m (f^n)^T]^T dm, \tag{22}$$

$$dJ = \{(\nabla_m \beta)^T + \sum_{n=1}^{L} \lambda_n^T [\nabla_m (f^n)^T]^T\} dm + \lambda_1^T [\nabla_{y^0} (f^1)^T]^T dy^0 + \sum_{n=1}^{L} \{(\nabla_{y^n} \beta)^T + \lambda_n^T [\nabla_{y^n} (f^n)^T]^T + \lambda_{n+1}^T [\nabla_{y^n} (f^{n+1})^T]^T\} dy^n. \tag{23}$$

Hence,

$$\frac{dJ}{dm} = \{(\nabla_m \beta)^T + \sum_{n=1}^{L} \lambda_n^T [\nabla_m (f^n)^T]^T\} + \lambda_1^T [\nabla_{y^0} (f^1)^T]^T \frac{dy^0}{dm} + \sum_{n=1}^{L} \{(\nabla_{y^n} \beta)^T + \lambda_n^T [\nabla_{y^n} (f^n)^T]^T + \lambda_{n+1}^T [\nabla_{y^n} (f^{n+1})^T]^T\} \frac{dy^n}{dm}, \tag{24}$$

where, y^0 is the initial reservoir condition; if the initial condition is fixed and independent of m, $\frac{dy^0}{dm} = 0$. One can choose the Lagrange multipliers such that the coefficient of $\frac{dy^n}{dm}$ is zero, i.e.

$$[\nabla_{y^n} (f^n)^T] \lambda_n = -\nabla_{y^n} \beta - \nabla_{y^n} (f^{n+1})^T \lambda_{n+1}. \tag{25}$$

Eq. 25 is the adjoint system of equations. To solve, set $\lambda_{L+1} = 0$, and solve (25) sequentially for $n = L, L-1, L-2, \ldots, 1$. After calculating the Lagrange multipliers, sensitivity of β can be computed from

$$\frac{dJ}{dm} = \nabla_m \beta + \sum_{n=1}^{L} \nabla_m (f^n)^T \lambda_n, \tag{26}$$

where $\frac{dJ}{dm}$ is an N_m-dimensional vector.

Computing the Gradient Using Adjoint

Adjoint can compute the sensitivity of β wrt all model parameters. If we set $\beta = g(m)^T v$, then $\nabla_m \beta = G^T v$, where G is the sensitivity matrix. In order to compute the gradient of the data mismatch term given by $G^T \{C_D^{-1} (g(m^l) - d_{\text{obs}})\}$, simply set $v = C_D^{-1} (g(m^l) - d_{\text{obs}})$, and calculate the adjoint solution for $\beta = g(m)^T v$.

Remarks

First, note that primary variables of simulation needed to compute the gradient from the adjoint solution are saved during the previous simulation run on disk using unformatted direct access I/O which uses the binary format to store data record by record. This can be an issue when running large reservoir models with many time steps.

Also note that the matrices used in calculating Lagrange multipliers are sparse, and they are saved in sparse format.

Direct Method

The direct method, also known as the gradient simulator method, can be used to calculate the sensitivity of all predicted data wrt one specific model parameter. Consider the simulation equations:

$$f^{n+1} = f^{n+1}(y^{n+1}, y^n, m) = 0, \quad n = 0, 1, 2, \ldots L-1 \tag{27}$$

$$df^{n+1} = [\nabla_{y^{n+1}}(f^{n+1})^T]^T dy^{n+1} + [\nabla_{y^n}(f^{n+1})^T]^T dy^n + [\nabla_m(f^{n+1})^T]^T dm = 0 \tag{28}$$

For each model parameter α_i, which can be log-permeability or porosity of a gridblock, one can solve

$$[\nabla_{y^{n+1}}(f^{n+1})^T]^T \frac{dy^{n+1}}{d\alpha_i} = -[\nabla_{y^n}(f^{n+1})^T]^T \frac{dy^n}{d\alpha_i} - [\nabla_m(f^{n+1})^T]^T \frac{dm}{d\alpha_i}, \tag{29}$$

to calculate $\frac{dy^{n+1}}{d\alpha_i}$ sequentially for $n = 1, 2, \ldots, L$. Note that $\frac{dy^0}{d\alpha_i} = 0$. Then for any predicted data, β, the sensitivity is calculated as

$$\frac{d\beta}{d\alpha_i} = \frac{\partial\beta}{\partial\alpha_i} + \sum_{n=0}^{L-1} \frac{\partial\beta}{\partial y^{n+1}} \frac{dy^{n+1}}{d\alpha_i}, \tag{30}$$

Direct method can compute the product of G times an arbitrary vector.

References

Anterion, F., B. Karcher, and R. Eymard, Use of parameter gradients for reservoir history matching, in *Proceedings of the 10th SPE Reservoir Simulation Symposium, Houston, Texas, 6–8 February*, SPE 18433, 1989.

Chu, L., M. Komara, and R. A. Schatzinger, An efficient technique for inversion of reservoir properties using iteration method, *SPE Journal*, **5**(1), 71–81, 2000.

Gao, G. and A. C. Reynolds, An improved implementation of the LBFGS algorithm for automatic history matching, *SPE Journal*, **11**(1), 5–17, 2006.

Kalita, R., *Conditioning a Three Dimensional Reservoir Model to Gas Production Data*, M.S. thesis, The University of Tulsa, Tulsa, Oklahoma, 2000.

Li, R., A. C. Reynolds, and D. S. Oliver, History matching of three-phase flow production data, *SPE Journal*, **8**(4), 328–340, 2003.

Makhlouf, E. M., W. H. Chen, M. L. Wasserman, and J. H. Seinfeld, A general history matching algorithm for three-phase, three-dimensional petroleum reservoirs, *SPE Advanced Technology Series*, **1**(2), 83–91, 1993.

Oliver, D. S., A. C. Reynolds, and N. Liu, *Inverse Theory for Petroleum Reservoir Characterization and History Matching*, Cambridge University Press, Cambridge, UK, 2008.

Tarantola, A., *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*, Elsevier, Amsterdam, The Netherlands, 1987.

Zhang, F. and A. C. Reynolds, Optimization algorithms for automatic history matching of production data, in *Proceedings of 8th European Conference on the Mathematics of Oil Recovery, Freiberg, Germany, 3–6 September*, 2002.