Computational Geosciences (2006) 10:119–136
DOI: 10.1007/s10596-005-9013-3

Calculating derivatives for automatic history matching

José Roberto P. Rodrigues

*PETROBRAS/CENPES/PDP/TR, Cidade Universitária, QD. 7, Ilha do Fundão,
Rio de Janeiro, RJ, 21941-598, Brazil.*
E-mail: jrprodrigues@petrobras.com.br

Received 30 November 2004; accepted 8 August 2005

Automatic history matching is based on minimizing an objective function that quantifies the mismatch between observed and simulated data. When using gradient-based methods for solving this optimization problem, a key point for the overall procedure is how the simulator delivers the necessary derivative information. In this paper, forward and adjoint methods for derivative calculation are discussed. Procedures for sensitivity matrix building, sensitivity matrix and transpose sensitivity matrix vector products are fully described. To show the usefulness of the derivative calculation algorithms, a new variant of the gradzone analysis, which tries to address the problem of selecting the most relevant parameters for a history matching, is proposed using the singular value decomposition of the sensitivity matrix. Application to a simple synthetic case shows that this procedure can reveal important information about the nature of the history-matching problem.

Keywords: automatic history matching, derivative calculation, parameter identification problem, gradzone analysis, adjoint method, truncated singular value decomposition, Lanczos method

1. Introduction

Automatic history matching is based on minimizing an objective function that quantifies the mismatch between observed and simulated data [1,4,9,11]. A priori information, such as geostatistics, can be included in this objective function [4,9]. Usually, the problem is formulated as a least-squares problem, with a strong non-linear character due to the complex nature of the dependence of the simulation results on the reservoir parameters. Among the possible methods for solving this minimization problem, gradient-based methods are generally reported as the most efficient ones [6]. According to the method selected, different derivative information has to be delivered. For instance, in the Gauss–Newton class of methods, the derivatives of the simulated data with respect to the parameters, which form the sensitivity matrix, are required, whereas for quasi-Newton, only the gradient of the objective function is needed, and in conjugate gradient methods, products of the sensitivity matrix and transpose sensitivity matrix with arbitrary vectors are requested [12,16].

A key point in this context is the interplay between the optimization algorithm and the procedure for calculating the derivatives. Two classes of methods are described in the literature for calculating derivative information within reservoir simulators: the forward method [1], also known as the gradient-simulator method, and the adjoint method [9–11]. In this work, a unified algebraic framework is presented for both methods, showing how the adjoint method can be derived directly from the forward method. Procedures for derivative-related calculation, including sensitivity matrix building, sensitivity matrix and transpose matrix vector products and gradient calculation, are reviewed and fully described, highlighting the similarities and common components shared by all of them.

To show how this framework for derivative calculation can be useful, even outside the context of optimization algorithms, it is discussed herein a new variant of the gradzone algorithm, proposed by Bissel [2] for grouping cells into regions for history-matching purposes. The singular value decomposition of the sensitivity matrix is used, instead of the eigenstructure of the Gauss–Newton approximation to the Hessian of the objective function. With the singular value decomposition, it is possible to have information also on simulator's response space, trying to identify which responses (wells, etc.) would be more affected by which parameters. In order to find the singular value decomposition, the Lanczos iterative method is employed [7,13,15], since it first approximates the largest singular values, thus being the ones carrying most of the information content of the problem. The Lanczos method does not need to build the sensitivity matrix because it only requires products of the relevant matrices and transpose matrices with arbitrary vectors, which can be efficiently calculated using the framework previously discussed for derivative calculation. Application to a simple synthetic five-spot problem is presented, showing how the singular value decomposition can reveal important information about the nature of the history-matching problem and help the engineer in deciding which are the relevant parameters for the match. In a companion paper [14], the idea of combining the truncated singular value decomposition of the sensitivity matrix with the Lanczos method is extended to propose a regularizing algorithm for solving the history-matching problem.

2. Derivative calculation in reservoir simulators

2.1. Preliminaries

In this section, the derivation of the forward and adjoint methods for derivative calculation is reviewed. A fully algebraic presentation is made, which shows how the adjoint method can be directly derived from the forward method. We start by introducing some notation and making some assumptions on the simulator.

Let $m \in \Re^p$ be the vector of reservoir parameters to be determined (porosities, permeabilities, etc.), $x^k(m) \in \Re^{m_k}$ be the vector of simulator primary variables at time level k (pressures, saturations, compositions at each reservoir cell plus well bottom-hole pressures at rate-constrained wells) and $\overline{m} = \max_k\{m_k\}$. The set of non-linear balance equations and well restrictions solved by the simulator to advance one step in time can be written as

$$f^k(x^k(m), x^{k-1}(m); m) = 0, \tag{1}$$

where $f^k : \Re^{m_k + m_{k-1} + p} \rightarrow \Re^{m_k}$, and it is assumed that x^0 represents given initial conditions. Let

$$J^k = \left[\frac{\partial f_i^k}{\partial x_j^k}\right]_{\substack{i=1,\dots,m_k\\ j=1,\dots,m_k}} \tag{2}$$

be the Jacobian matrix of f^k with respect to the primary variables at time level k, which is calculated by the simulator for solving (1) by Newton's method, and

$$J^{k-1,k} = \left[\frac{\partial f_i^k}{\partial x_j^{k-1}}\right]_{\substack{i=1,\dots,m_k\\ j=1,\dots,m_{k-1}}} \tag{3}$$

be the Jacobian matrix for the previous time level $k - 1$. Define

$$x(m) = \left[\left(x^1(m)\right)^T \left(x^2(m)\right)^T \cdots \left(x^N(m)\right)^T\right]^T, \tag{4}$$

the vector of primary variables for all time steps, $x(m) \in \Re^M$, and

$$f = \left[\left(f^1(x^1, x^0; m)\right)^T \left(f^2(x^2, x^1; m)\right)^T \cdots \left(f^N(x^N, x^{N-1}; m)\right)^T\right]^T, \tag{5}$$

$f : \Re^{M+p} \rightarrow \Re^M$, so that the functional relationship defining the simulator is written as

$$f(x(m); m) = 0. \tag{6}$$

Matrices of partial derivatives are defined as

$$D_x f = \left[\frac{\partial f_i}{\partial x_j}\right]_{\substack{i=1,\dots,M\\ j=1,\dots,M}}, D_m f = \left[\frac{\partial f_i}{\partial m_j}\right]_{\substack{i=1,\dots,M\\ j=1,\dots,p}} \text{ and } D_m x = \left[\frac{\partial x_i}{\partial m_j}\right]_{\substack{i=1,\dots,M\\ j=1,\dots,p}}. \tag{7}$$

The simulator response to be compared with the observed data, which can be, for instance, the water rate or bottom-hole pressure in a particular well, is not given directly by (6), but is obtained after the calculation of primary variables, and can be described as

$$y_i^{sim} = g_i^{sim}\left(x^k(m); m\right), i = 1, \ldots, n_o, \tag{8}$$

where k is the time level when the ith observation was taken and n_o is the number of observed data. Let $g^{sim} : \Re^{M+p} \rightarrow \Re^{n_o}$,

$$g^{sim}(x(m); m) = \left[g_1^{sim}(x(m); m)\ g_2^{sim}(x(m); m) \cdots g_{n_o}^{sim}(x(m); m)\right]^T, \tag{9}$$

be the vector of observed data and

$$D_x g^{sim} = \left[\frac{\partial g_i^{sim}}{\partial x_j}\right]_{\substack{i=1,\ldots,n_o \\ j=1,\ldots,M}} \text{ and } D_m g^{sim} = \left[\frac{\partial g_i^{sim}}{\partial m_j}\right]_{\substack{i=1,\ldots,n_o \\ j=1,\ldots,p}} \tag{10}$$

be matrices of partial derivatives defined similarly to (7). The matrix of (total) derivatives of g_i^{sim}, $i = 1, \ldots, n_o$, with respect to the parameters is often called sensitivity matrix and is defined as

$$G = \left[\frac{dg_i^{sim}}{dm_j}\right]_{\substack{i=1,\ldots,n_o \\ j=1,\ldots,p}}. \tag{11}$$

Given a matrix $A \in \Re^{r\times s}$, $A_{.,j} \in \Re^r$ will denote the jth column of A, $1 \leq j \leq s$.

2.2. Forward method

It will be shown how the forward method can be used to calculate the sensitivity matrix as well as the product of the sensitivity matrix with an arbitrary vector $v \in \Re^p$.

Differentiating (8) with respect to parameter m_j,

$$\frac{dg_i^{sim}}{dm_j} = \sum_{k=1}^{M} \frac{\partial g_i^{sim}}{\partial x_k} \frac{\partial x_k}{\partial m_j} + \frac{\partial g_i^{sim}}{\partial m_j}. \tag{12}$$

To calculate the derivatives of primary variables with respect to the parameters, differentiate (6), using (7), to obtain

$$(D_x f)(D_m x) + D_m f = 0, \tag{13}$$

from what follows

$$D_m x = -(D_x f)^{-1}(D_m f), \tag{14}$$

which requires the solution of p linear systems involving matrix $D_x f$. Using the time-level partition previously introduced and recalling (1), the structure of $D_x f$ can be represented as

$$D_x f = \begin{bmatrix} J^1 & & & \\ J^{1,2} & J^2 & & \\ & \ddots & \ddots & \\ & & J^{N-1,N} & J^N \end{bmatrix}, \tag{15}$$

where J^k and $J^{k-1,k}$ are defined by (2) and (3). The derivatives of primary variables with respect to parameters are obtained marching forward in time,

$$\begin{aligned} J^1 (D_m x)^1 &= -(D_m f)^1 \\ J^k (D_m x)^k &= -J^{k-1,k}(D_m x)^{k-1} - (D_m f)^k, k = 2, \ldots, N \end{aligned}, \tag{16}$$

requiring the solution of p linear systems. Substituting into (12) provides the desired derivatives.

Since a system of order equal to the number of primary variables in the simulator has to be solved for each parameter, the total cost for calculating the sensitivity matrix using the forward method is proportional to the number of parameters. In terms of storage, it requires one array of order $\overline{m} \times p$ for $D_m x$, one array of order $\overline{m}$ for the right-hand side of (16) and a sparse matrix for $J^{k-1,k}$. The sparsity structure of $J^{k-1,k}$ depends on the simulator's time-advancing scheme and is discussed in [14].

Now, it will be shown how the forward method can be used to calculate the product Gv, where $v \in \Re^p$ is an arbitrary vector. Using (7) and (10), one sees that

$$G = \left(D_x g^{sim}\right)(D_m x) + D_m g^{sim}, \tag{17}$$

which, after substitution of (14), results into

$$G = -\left(D_x g^{sim}\right)(D_x f)^{-1}(D_m f) + D_m g^{sim}, \tag{18}$$

so that

$$Gv = -\left(D_x g^{sim}\right)(D_x f)^{-1}(D_m f)v + \left(D_m g^{sim}\right)v. \tag{19}$$

Recalling the structure of $D_x f$ (15), it is clear that the product $z = (D_x f)^{-1}(D_m f)v$ can be obtained marching forward in time,

$$\begin{array}{l} J^1 z^1 = (D_m f)^1 v \\ J^k z^k = -J^{k-1,k} z^{k-1} + (D_m f)^k v, k = 2, \ldots, N \end{array}. \tag{20}$$

After z has been calculated, the components of Gv can be calculated using

$$(Gv)_i = -\sum_{k=1}^{M} \frac{\partial g_i^{sim}}{\partial x_k} z_k + \sum_{j=1}^{p} \frac{\partial g_i^{sim}}{\partial m_j} v_j, i = 1, \ldots, n_o. \tag{21}$$

A single linear system per time step has to be solved, so that the computational cost is approximately constant regardless of the number of parameters or observed data. The storage requirements are limited to one array of size $\overline{m}$ for holding z and a sparse matrix for $J^{k-1,k}$.

2.3. Adjoint method

The starting point for deriving the adjoint method for calculating G is the observation that only a few $\frac{\partial x_k}{\partial m_j}$ are necessary in (12), since $\frac{\partial g_i^{sim}}{\partial x_k}$ is non-zero only for those k for which x_k is a primary variable that has an effect on observed data i (same time step and well block). From (14), it can be written that

$$\frac{\partial x_k}{\partial m_j} = -\sum_{q=1}^{M} (D_x f)^{-1}_{kq} (D_m f)_{qj}, \tag{22}$$

which, after substitution in (12) and changing the order of summation, results into

$$\begin{aligned} \frac{dg_i^{sim}}{dm_j} &= -\sum_{k=1}^{M} \frac{\partial g_i^{sim}}{\partial x_k} \sum_{q=1}^{M} (D_x f)^{-1}_{kq} (D_m f)_{qj} + \frac{\partial g_i^{sim}}{\partial m_j} = \\ &= -\sum_{k=1}^{M} \sum_{q=1}^{M} (D_x f)^{-1}_{kq} \frac{\partial g_i^{sim}}{\partial x_k} (D_m f)_{qj} + \frac{\partial g_i^{sim}}{\partial m_j} = \\ &= -\sum_{q=1}^{M} \sum_{k=1}^{M} (D_x f)^{-T}_{qk} \frac{\partial g_i^{sim}}{\partial x_k} (D_m f)_{qj} + \frac{\partial g_i^{sim}}{\partial m_j}. \end{aligned} \tag{23}$$

Introducing a vector of "adjoint states" given by

$$\lambda = -(D_x f)^{-T} \left(D_x g_i^{sim}\right) \tag{24}$$

and substituting into (23), results into

$$\frac{dg_i^{sim}}{dm_j} = \sum_{q=1}^{M} \lambda_q (D_m f)_{qj} + \frac{\partial g_i^{sim}}{\partial m_j} = \lambda^T (D_m f)_{\cdot,j} + \frac{\partial g_i^{sim}}{\partial m_j}. \tag{25}$$

The adjoint method is comprised of two steps: first, finding λ by solving (24), then using (25) to calculate the desired derivatives. As before, using the time level partition, we have

$$(D_x f)^T = \begin{bmatrix} (J^1)^T & (J^{1,2})^T & & \\ & (J^2)^T & \ddots & \\ & & \ddots & (J^{N-1,N})^T \\ & & & (J^N)^T \end{bmatrix}, \tag{26}$$

so that λ is obtained marching backward in time,

$$\left.\begin{aligned} (J^N)^T \lambda^N &= \left(D_x g_i^{sim}\right)^N \\ \left(J^k\right)^T \lambda^k &= -\left(J^{k,k+1}\right)^T \lambda^{k+1} + \left(D_x g_i^{sim}\right)^k, k = N-1, \ldots, 1 \end{aligned}\right\}. \tag{27}$$

Since a system of order equal to the number of primary variables in the simulator has to be solved for each observed data, the total cost for calculating the sensitivity matrix using the adjoint method is proportional to the number of observed data. In terms of storage, it requires one array of order $\overline{m} \times n_o$ for λ, one array of order $\overline{m}$ for z and a sparse matrix for $J^{k-1,k}$.

Given an arbitrary vector $v \in \Re^{n_o}$, it will be shown how the adjoint method can be used to calculate the product $G^T v$. Transposing (18), one obtains

$$G^T = -(D_m f)^T (D_x f)^{-T} \left(D_x g^{sim}\right)^T + \left(D_m g^{sim}\right)^T, \tag{28}$$

from what

$$G^T v = -(D_m f)^T (D_x f)^{-T} \left(D_x g^{sim}\right)^T v + \left(D_m g^{sim}\right)^T v \tag{29}$$

follows. From (26), it is clear that the product $z = (D_x f)^{-T} (D_m g^{sim})^T v$ can be obtained marching backward in time,

$$\left.\begin{aligned} (J^N)^T z^N &= \left(\left(D_m g^{sim}\right)^N\right)^T v \\ \left(J^k\right)^T z^k &= -\left(J^{k,k+1}\right)^T z^{k+1} + \left(\left(D_m g^{sim}\right)^k\right)^T v, k = N-1, \ldots, 1 \end{aligned}\right\}. \tag{30}$$

After z has been calculated, the components of $G^T v$ can be calculated using

$$\left(G^T v\right)_j = -\sum_{k=1}^{M} \frac{\partial f_k}{\partial m_j} z_k + \sum_{i=1}^{n_o} \frac{\partial g_i^{sim}}{\partial m_j} v_i,\ j = 1, \ldots, p. \tag{31}$$

Similarly to the forward method for sensitivity matrix vector product, a single linear system solution per time step is required, so that the computational cost is approximately constant regardless of the number of parameters or observed data. The storage requirements are one array of size $\overline{m}$ for holding z and a sparse matrix for $J^{k-1,k}$.

3. Application to gradzone analysis

3.1. An overview of the history-matching problem

When solving an automatic history-matching problem, we are interested in the objective function

$$O(m) = \frac{1}{2} r^T W r, \tag{32}$$

where r is a vector of residuals given by

$$r(m) = \left[g_1^{sim}(m) - y_1^{obs} \;\; g_2^{sim}(m) - y_2^{obs} \;\; \cdots \;\; g_{n_o}^{sim}(m) - y_{n_o}^{obs} \right]^T, \tag{33}$$

being y_i^{obs} the observed data, and W is a weighted matrix assumed to be diagonal. A priori information, such as geostatistics, can also be included in this objective function (see discussion at the end of section 3.2 and [4,9]). The gradient of $O(m)$ is given by

$$\nabla O(m) = G^T W r. \tag{34}$$

Equation (34) shows that, making $v = Wr$, the transpose sensitivity matrix times vector procedure discussed in section 2.3 can be used to calculate $\nabla O(m)$.

3.2. Description of the gradzone procedure

To show how the framework presented in section 2 for derivative calculation can be useful, a new variant of the gradzone analysis will be discussed. This algorithm was proposed by Bissel [2] and tries to address the problem of parameter selection in a history-matching process. Note that, in principle, the values of the petrophysical properties in each grid cell can be treated as parameters to be determined, which makes the number of potential parameters much larger than the number of observations. The gradzone analysis, as originally proposed, uses the spectrum of

G^TWG for suggesting to the engineer how to group cells into regions for history-matching purposes in order to keep the number of parameters manageable by an optimization algorithm.

By defining one parameter for each grid cell, it takes the largest eigenvalue of G^TWG, associates the components of its eigenvector with each grid cell, and those components greater than a certain threshold are grouped together. Positive and negative components are treated separately, so that the first eigenvalue can generate the first two parameter regions. The second-largest eigenvalue is treated similarly and so on, until a sufficiently small eigenvalue is reached. Since this procedure was coupled with a forward method for calculating G, it was not possible to handle G^TWG matrix for a large number of parameters, and, to make the procedure viable, a set of sample cells is defined and the eigenvector components are interpolated in order to generate the eigencomponents for the whole grid.

The motivation for this algorithm, as described in [2], is based on the fact that G^TWG is the Gauss–Newton approximation to the Hessian of the objective function (32), and, in order for its minimization to proceed smoothly, G^TWG should have good mathematical properties, such as diagonal dominance. Now, another justification for the gradzone algorithm will be presented, which will lead to a new implementation. Let m_o be a point in the parameter space and consider a linear approximation for g^{sim} around m_o,

$$g^{sim}(m) \approx g^{sim}(m_o) + G\delta m, \tag{35}$$

where $\delta m = m - m_o$ and G is calculated at m_o. Substituting (35) into (32), a linear approximation to the objective function is found,

$$\begin{aligned} O(m) &\approx \frac{1}{2}(G\delta m + r_o)^T W(G\delta m + r_o) \\ &= \frac{1}{2}\left(W^{1/2}G\delta m + W^{1/2}r_o\right)^T\left(W^{1/2}G\delta m + W^{1/2}r_o\right), \end{aligned} \tag{36}$$

where r_o is the vector of residuals (33) calculated at $m = m_o$. The problem of minimizing (36) is a linear least-squares problem, whose the least norm solution can be expressed in terms of the singular value decomposition of $W^{1/2}G$ [3,7],

$$\delta m = -\sum_{i=1}^{n_{sv}} \frac{u_i^T W^{1/2} r_o}{\sigma_i} v_i, \tag{37}$$

where σ_i are the singular values of $W^{1/2}G$, indexed in decreasing order, u_i are the left singular vectors, v_i are the right singular vectors and n_{sv} is the number of non-zero singular values. Expression (37) reveals that the most relevant information about the solution is contained in the largest singular values and corresponding singular vectors, since small singular values tend to amplify noise in the measurements [8]. Since the singular values of $W^{1/2}G$ are the positive square roots of the eigenvalues of G^TWG, and the right singular values of $W^{1/2}G$ are the eigenvectors of G^TWG [3], the gradzone analysis can also be interpreted in terms of the singular value decomposition. The use of the singular value decomposition, however, has one advantage over the eigendecomposition of G^TWG: associating the components of the left singular values with the observed data, it is possible to have some information also on the simulator's response space, trying to identify which responses (wells, etc.) would be more affected by which parameters.

Another observation is that (37) is the basis for the truncated singular value decomposition [3,15], a well-known regularization technique for ill-posed problems, where only the first few largest singular values are considered in the summation in the equation. This observation is the basis for an extension to this algorithm, which uses the singular value decomposition to solve a sequence of problems similar to the minimization of (36), aiming to solve the history-matching problem. This approach is further discussed in [14].

It remains to be discussed how to calculate the singular value decomposition of $W^{1/2}G$. In this work, the use of the Lanczos method is proposed [7,13–15], which is convenient for two main reasons. First, it iteratively approximates firstly the largest singular values and corresponding singular vectors, which are the ones we are seeking for. Second, the Lanczos method does not need to build $W^{1/2}G$ because it only requires products of $W^{1/2}G$ and $G^TW^{1/2}$ with arbitrary vectors, which, as shown in section 2, can be done at a cost proportional to a single simulation.

It is also possible to include prior information in the gradzone analysis, as presented in [4]. Now, it is shown how this can be done in the framework discussed above. When using prior information, the objective function (32) is augmented with a quadratic term, expressing some known information about the solution, like smoothness or spatial correlation, as described by geostatistics [4,9]. After linearization, the history-matching process can be expressed as the minimization of the objective function

$$O(m) = \frac{1}{2}\left(W^{1/2}G\delta m + W^{1/2}r_o\right)^T\left(W^{1/2}G\delta m + W^{1/2}r_o\right) + \frac{\alpha}{2}(\delta m)^T\Gamma\delta m, \tag{38}$$

where Γ is a matrix expressing prior information, which is assumed to be symmetric positive definite, and α is a real number expressing the relative weight between the prior and mismatch terms. Let

$$\Gamma = LL^T \tag{39}$$

be the Choleski decomposition of Γ and introduce the change of variables

$$\delta\widetilde{m} = L^T\delta m. \tag{40}$$

Substituting (40) into (38),

$$O(m) = \frac{1}{2}\left(W^{1/2}GL^{-T}\delta\widetilde{m} + W^{1/2}r_o\right)^T \left(W^{1/2}GL^{-T}\delta\widetilde{m} + W^{1/2}r_o\right) + \frac{\alpha}{2}(\delta\widetilde{m})^T\delta\widetilde{m}, \tag{41}$$

whose solution can be expressed in terms of the singular value decomposition of the matrix $A = W^{1/2}GL^{-T}$ [14,15]. Thus, to take into account prior information in the gradzone analysis, the relevant quantities are the singular vectors associated with the largest singular values of A. Once the Choleski decomposition (39) is obtained, the Lanczos procedure has to be modified so as to make the matrix products with A, instead of $W^{1/2}G$. This can be achieved without building A. If the product $x = As$ for an arbitrary vector $s \in \Re^p$ is needed, three steps are required. First, it obtains $y = L^{-T}s$ by solving the upper triangular system $L^Ty = s$, then it applies the forward method from section 2.2 to get $\overline{x} = Gy$ and finally multiplies by $W^{1/2}$ to obtain $x = W^{1/2}\overline{x}$. Analogously, if $s \in \Re^{n_o}$ and $x = A^Ts = L^{-1}G^TW^{1/2}s$ is desired, it starts by multiplying $W^{1/2}$ by s, obtaining $\overline{s} = W^{1/2}s$, then it applies the adjoint method (section 2.3) to get $y = G^T\overline{s}$ and then solves $Lx = y$.

3.3. A numerical example

The derivative calculation procedures were implemented in a full-featured adaptive implicit black-oil simulator [5]. Discussion on some issues related to the implementation of these methods in existing codes can be found in [14]. In this section, an application of the gradzone analysis to a simple synthetic five-spot case is shown. It consists of a 23 × 23 × 5 regular mesh with grid block dimensions of 50 × 50 × 5 m. An injection well is located at the centre of the mesh and four producers at the corners, all wells perforated in the five layers. A reservoir model with four regions with different transmissibility multipliers as shown in figure 1 was simulated for 10 years to generate water production rate at the producers. A 10% noise level was added to the simulated values, and the resulting water rates were used as observed data. The gradzone analysis was applied to a homogeneous reservoir model with transmissibility multipliers equal to 1. This homogeneous model will be referred to as the starting model. Due to gravitational segregation, water moves preferentially through the bottommost layers, and therefore, the information extracted from the water rate history is concentrated in these layers. Table 1 presents the sum of squares of the components for the first eight right singular vectors for each layer. Around 75% of the values are concentrated in the fifth layer for the first four singular vectors. For the next four singular vectors, around half of the values are concentrated in the fourth layer and around 30% in the fifth. Figure 2 shows a map of the components for the first right singular vector for the fifth layer. Note that the components are greater in the region linking the injector and producer PROD4. Associating the components of the first left

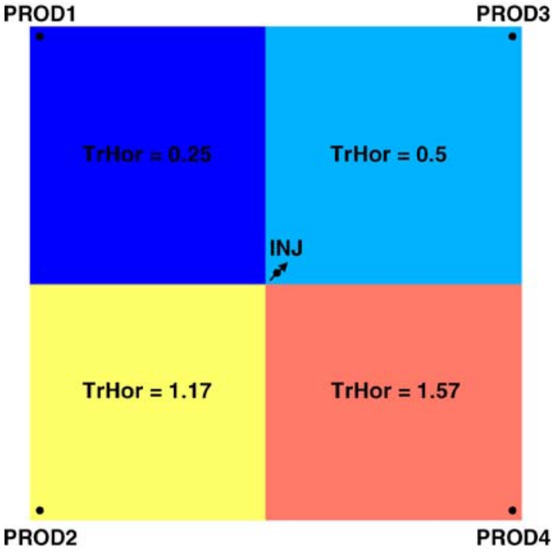

Figure 1. *Top view* of the reservoir model used to generate water rate history. TrHor is the value of horizontal transmissibility multiplier used in each of the four constant properties regions.

Table 1
Sum of the squares of the components of the first eight right singular vectors for each layer of the grid.

	Sum of squares of the components per layer							
	1	2	3	4	5	6	7	8
Layer 1	0.04	0.04	0.04	0.05	0.03	0.04	0.04	0.07
Layer 2	0.05	0.05	0.05	0.04	0.04	0.04	0.04	0.06
Layer 3	0.06	0.06	0.05	0.03	0.05	0.05	0.05	0.05
Layer 4	0.10	0.08	0.08	0.03	0.56	0.56	0.57	0.57
Layer 5	0.75	0.77	0.77	0.84	0.32	0.31	0.29	0.25

singular vector with the observed data results in the plot shown in figure 3, where, in accordance with figure 2, the most pronounced components are related to well PROD4. A similar map for the components of the second right singular vector is shown in figure 4. The largest components are in the region linking injector and producer PROD2, with smaller components in the regions between the injector and PROD1 and PROD3 and almost null components in the region between injector and PROD4. The plot of the components for the second left singular vector in figure 5 follows this trend: the larger components in absolute value are related to PROD2, with smaller values for PROD1 and PROD3 and almost zero components for PROD4. For the third singular vectors, a similar correspondence between the most significant components in the right and left singular vectors are observed. Based on the infor-

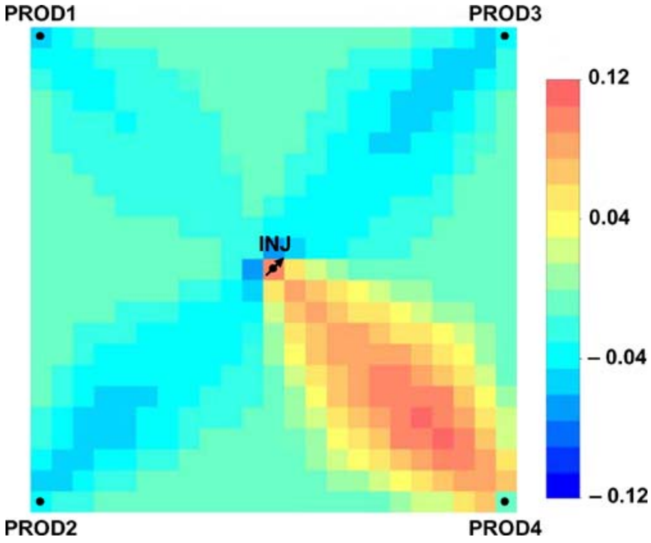

Figure 2. First right singular vector for the fifth layer.

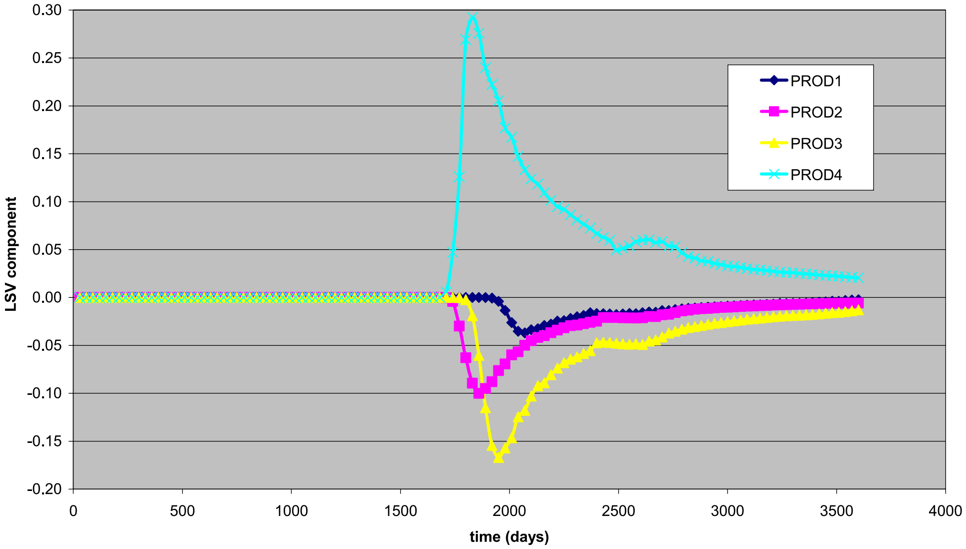

Figure 3. First left singular vector.

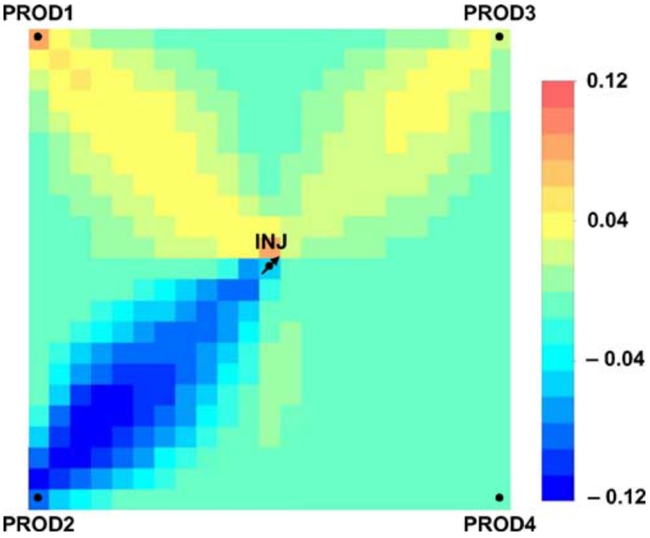

Figure 4. Second right singular vector for the fifth layer.

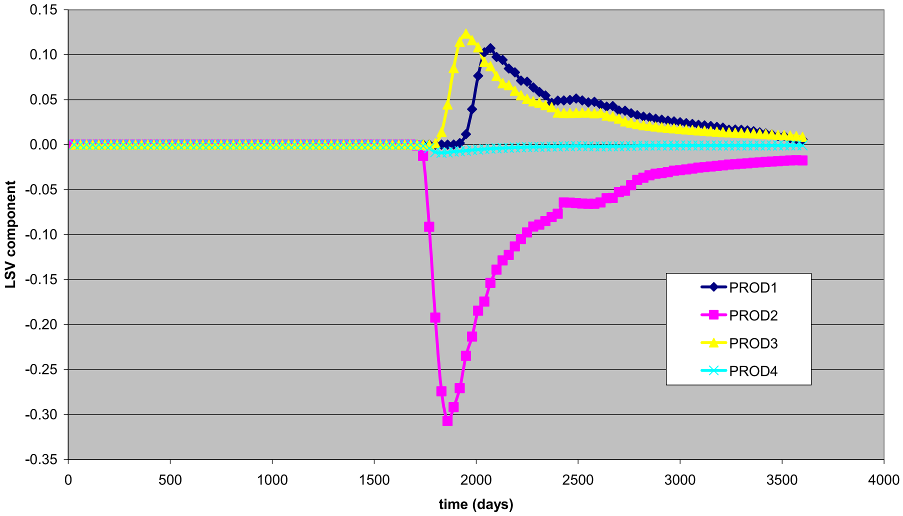

Figure 5. Second left singular vector.

mation of these three singular triplets, a gradzone analysis was made, using as threshold parameter a value of 0.4 times the largest (in absolute value) component of each right singular vector. Figure 6 shows the resulting six zones in the fifth layer. Note that there is a trend of generating regions linking the injector and the producers.

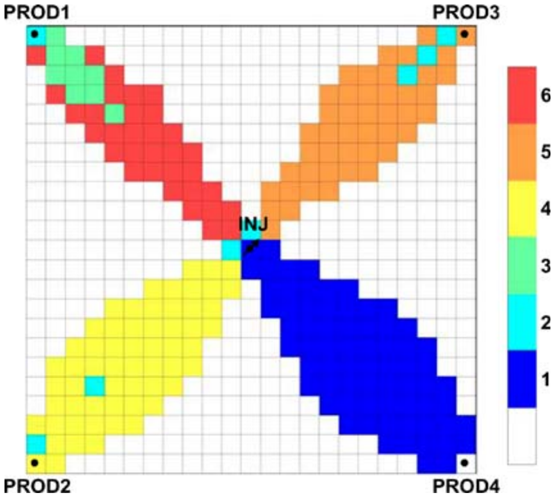

Figure 6. Zone map for the fifth layer generated using the first three singular vectors.

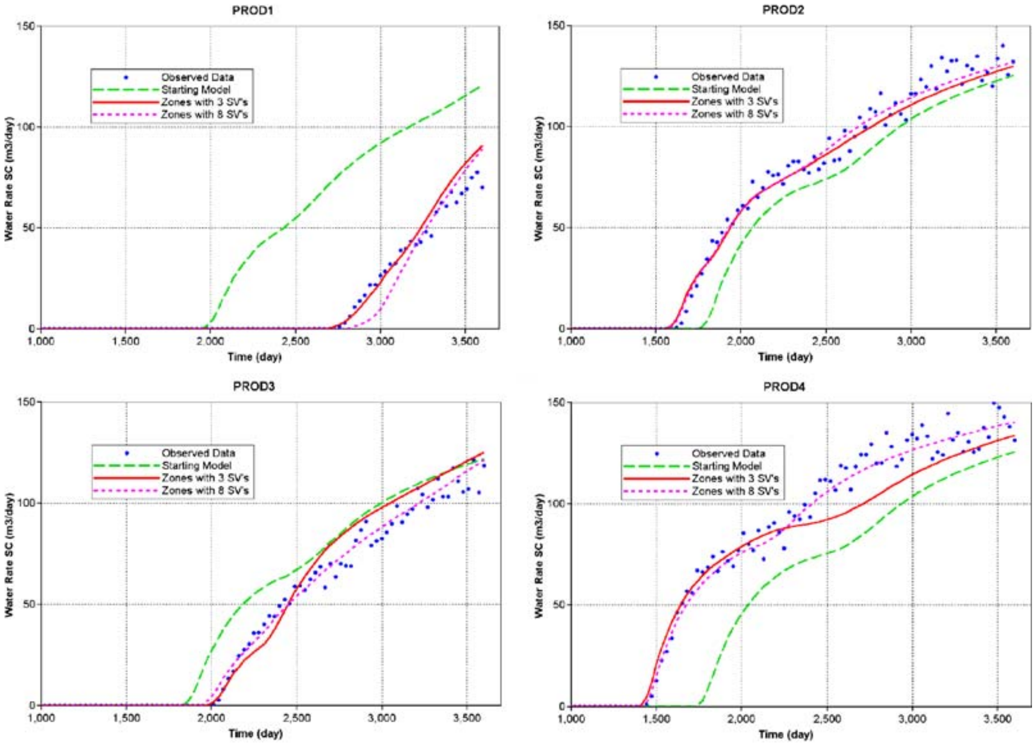

Figure 7. Water rates for the producers. *Dots* are the history rates, *dash line* is the starting model rates, *continuous line* is the result of the match using the zones defined by the first three singular vectors and *short dash line* is the result of the match using the zones defined by the first eight singular vectors.

These regions were used in an automatic history-matching engine, using one as initial guess for the transmissibility multipliers. No attempt was made to try a better clustering of the zones, e.g., eliminating the zone consisting of isolated cells near PROD2 and PROD3. Figure 7 shows the final match for each of the wells together with the starting model water rates. In all wells, a significant improvement in the match was reached, particularly near the breakthrough time.

These results can be better understood looking at figures 3 and 5: there is a peak in the components of the left singular vectors close to the breakthrough time in the starting model with the components decaying for larger times. In order to see if further improvement could be achieved, the subsequent singular triplets were examined. Figure 8 shows a plot of the fifth left singular vector. The larger components (in absolute value) are in the second half of the water production period, so that a better match could be expected for this period if the information given by the corresponding right singular vector were used for generating the zones. The subsequent singular vectors up to the eighth follow the same trend. The same threshold criterion was used in a new gradzone analysis with information from the first eight singular vectors. Thirteen zones were generated. Apart from an isolated cell in the fifth layer, the additional zones were defined in the fourth layer. As before, an automatic history-matching procedure was applied to the parameters exactly as generated by the gradzone analysis, and the resulting matching is shown in figure 7. Note the improvement in the match for the late times of the history in wells PROD2, PROD3 and

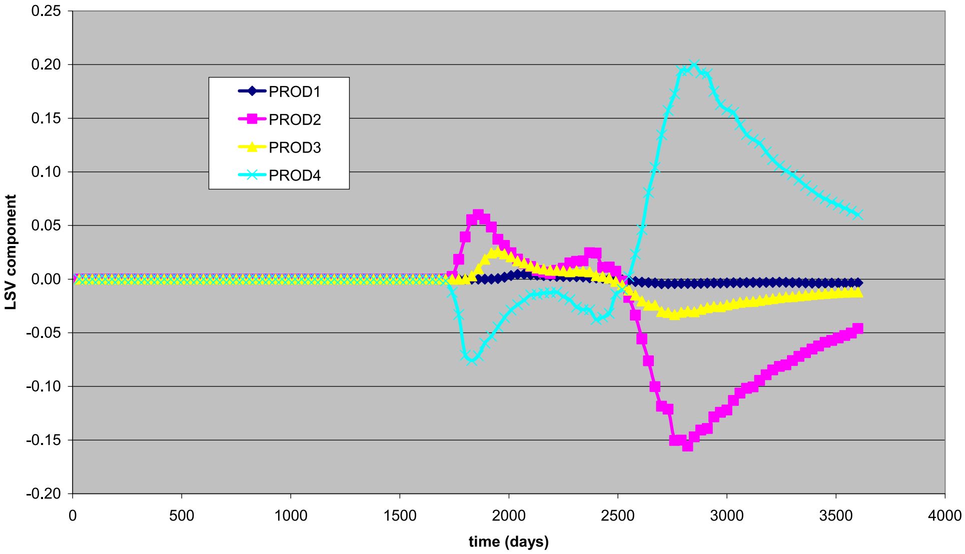

Figure 8. Fifth left singular vector.

PROD4. Well PROD1, on the other hand, shows degradation in the match for the first half of the history.

4. Conclusions

Procedures for derivative calculation in reservoir simulators were discussed and presented using a unified approach. The presentation highlights similarities and common components shared by all of them. The algorithms were applied to implement a new variant of the gradzone analysis for grouping cells into parameters for history-matching purposes. This variant uses the singular value decomposition of the sensitivity matrix, calculated by the Lanczos iterative method. Results of a simple five-spot synthetic case show that the singular value decomposition can be a useful tool for revealing information on the nature of the history-matching problem.

Acknowledgments

The optimization package used in the history-matching engine was developed by Dr. Andrew Conn and Dr. Andreas Waechter from IBM T.J. Watson Research Center, in the context of a joint cooperation agreement between PETROBRAS and IBM. The author wishes to express his gratitude to Dr. Dave Collins, from Computer Modelling Group, for his help with the simulator's code. The support from PETROBRAS management is also greatly appreciated.

References

[1] F. Anterion, R. Eymard and B. Karcher, Use of parameter gradients for reservoir history matching, in: *Proc. of the SPE Symposium on Reservoir Simulation*, Houston, TX, USA, paper SPE 18433 (6–8 February 1989).

[2] R. Bissel, Calculating optimal parameters for history matching, in: *Proc. of the 4th European Conference on the Mathematics of Oil Recovery*, Roros, Norway (7–10 June 1994).

[3] A. Bjorck, *Numerical Methods for Least Squares Problems* (SIAM, Philadelphia, 1996).

[4] B. Brun, O. Gosselin and J.W. Barker, Use of prior information in gradient-based history-matching, in: *Proc. of the SPE Symposium on Reservoir Simulation*, Houston, TX, USA, paper SPE 66353 (11–14 February 200).

[5] Computer Modelling Group, *IMEX 2004 User's Guide* (Calgary, Canada, 2004).

[6] G. Gao, G. Li and A.C. Reynolds, A stochastic optimization algorithm for automatic history matching, in: *Proc. of the SPE Annual Technical Conference and Exhibition*, Houston, TX, USA, paper SPE 90065 (26–29 September 200).

[7] G.H. Golub and C.F. Van Loan, *Matrix Computations* (John Hopkins University Press, Baltimore, 1989).

[8] M. Hanke and P.C. Hansen, Regularization methods for large-scale problems, Surv. Math. Ind. 3 (1993) 253–315.

[9] R. Li, A.C. Reynolds and D.S. Oliver, History matching of three-phase flow production data, in: *Proc. of the SPE Symposium on Reservoir Simulation*, Houston, TX, USA, paper SPE 66351 (11–14 February 2001).

[10] R. Li, A.C. Reynolds and D.S. Oliver, Sensitivity coefficients for three-phase flow history matching, J. Can. Pet. Technol. 42(4) (2003) 70–77.

[11] M. Mezghani, *Caractérisation des Réservoirs Hétérogènes:Méthode des Sentinelles et Quantification des Incertitudes*, Ph.D. diss., Institut Français du Pétrole, Paris, France (1999).

[12] J. Nocedal and S.J. Wright, *Numerical Optimization* (Springer, New York, 1999).

[13] D.P. O'Leary and J.A. Simmons, A bidiagonalization–regularization procedure for large scale discretizations of ill-posed problems, SIAM J. Sci. Statist. Comput. 2(4) (1981) 474–489.

[14] J.R.P. Rodrigues, Calculating derivatives for history matching in reservoir simulators, in: *Proc. of the SPE Symposium on Reservoir Simulation*, The Woodlands, USA, paper SPE 93445 (2005).

[15] C.R. Vogel and J.G. Wade, Iterative SVD-based methods for ill-posed problems, SIAM J. Sci. Comput. 15(3) (1994) 736–754.

[16] F. Zhang and A.C. Reynolds, Optimization algorithms for automatic history matching of production data, in: *Proc. of the 8th European Conference on the Mathematics of Oil Recovery* (2002).