

Research Article

A Parallel Stochastic Framework for Reservoir Characterization and History Matching

**Sunil G. Thomas,^{1,2} Hector M. Klie,^{1,3} Adolfo A. Rodriguez,^{1,3}
and Mary F. Wheeler^{4,5}**

¹ *Institute for Computational Engineering and Sciences, The University of Texas at Austin,
Austin, TX 78712, USA*

² *Chevron ETC, San Ramon, CA 94583, USA*

³ *ConocoPhillips, Houston, TX 77252, USA*

⁴ *Institute for Computational Engineering and Sciences, Department of Aerospace Engineering and
Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712, USA*

⁵ *Department of Petroleum and Geosystems Engineering, The University of Texas at Austin,
Austin, TX 78712, USA*

Correspondence should be addressed to Sunil G. Thomas, sunil.g.thomas@gmail.com

Received 15 December 2010; Accepted 1 February 2011

Academic Editor: Shuyu Sun

Copyright © 2011 Sunil G. Thomas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The spatial distribution of parameters that characterize the subsurface is never known to any reasonable level of accuracy required to solve the governing PDEs of multiphase flow or species transport through porous media. This paper presents a numerically cheap, yet efficient, accurate and parallel framework to estimate reservoir parameters, for example, medium permeability, using sensor information from measurements of the solution variables such as phase pressures, phase concentrations, fluxes, and seismic and well log data. Numerical results are presented to demonstrate the method.

1. Introduction

Uncertainties associated with the measurement of subsurface medium properties such as rock permeability and porosity are widely known. Deterministic models of multiphase flow and transport through porous media require accurate estimate of such properties, since these affect the estimation and location of recoverable reserves, in addition to revealing true subsurface flow characteristics under various injection scenarios. Modern drilling instruments routinely have sensors installed at strategic locations in a reservoir. Specialized sensors are capable of measuring at a high local resolution, fluid, and rock properties [1, 2]. These advances together with 4d time-lapse seismic studies are revealing enormous potential in reducing uncertainty associated with reservoir characterization. Meanwhile new stochastic

optimization and statistical learning methods are emerging as promising tools to determining nontrivial correlations between data measurements and responses in addition to developing optimal reservoir exploration plans [3]. In this work, the existence of a number of prior realizations of the unknown data (i.e., estimates, e.g., from seismic studies) is assumed. The estimation and sampling is then performed at a fixed resolution using a parallel version of the SPSA [4] (simultaneous perturbation, stochastic approximation) algorithm. A multilevel approach, coupled to neural-network engines that enhance the solution by calculating sensitivities in the vicinity of the most promising solution was presented in [5, 6].

Suppose that a set of N realizations, \mathbf{x}_i , $i = 1, \dots, N$ is given of a reservoir property (e.g., permeability) where each $\mathbf{x}_i \in \mathbb{R}^M$, M being the number of grid elements. Using methods such as wavelet analysis [7] or the principle component analysis (PCA) method [8, 9], it is possible to determine a reduced set of basis functions that characterize a majority of the variability. In this work, the PCA method is adopted. Then, let the set of realizations be denoted by the matrix \mathbf{X} , given by

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}. \quad (1.1)$$

Further, let the empirical mean be given by

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i. \quad (1.2)$$

Next, let \mathbf{Y} denote the deviation of the data from the mean, given by

$$\mathbf{Y} = \mathbf{X} - \{\bar{\mathbf{x}}, \bar{\mathbf{x}}, \dots, \bar{\mathbf{x}}\}. \quad (1.3)$$

In the PCA method, the covariance matrix \mathbf{C} is required of the deviation of the data from the mean. The covariance matrix is given by

$$\mathbf{C} = E(\mathbf{Y}\mathbf{Y}^T). \quad (1.4)$$

Since each of the N realizations is in general equally likely, in (1.4), it is possible to further reduce the expression for covariance as $\mathbf{C} = E(\mathbf{Y}\mathbf{Y}^T) = (1/N)\mathbf{Y}\mathbf{Y}^T$. Thus, the covariance matrix \mathbf{C} is symmetric and positive definite. Hence, it is possible to compute the eigen decomposition of \mathbf{C} and the associated eigen values as follows

$$\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \quad \text{with } \mathbf{D} = \text{diag}(\{\lambda_i\}_{i=1}^M). \quad (1.5)$$

In (1.5), $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^M$, the matrix of eigen vectors of \mathbf{C} and λ_i are the corresponding eigen values. Let λ_i be arranged in descending order $\lambda_1 > \lambda_2 > \dots > \lambda_M$. Then each realization can be expressed to a desired level of accuracy as a linear combination of the first P eigen value/vector pairs ($P < M$). Infact, since \mathbf{C} is symmetric in addition to being positive definite, the eigen vectors of \mathbf{V} are orthogonal and thus, (1.5) can be expressed in terms of orthormal eigen vectors. In such a case, the eigen values coincide with the singular values.

Then, a well known result can be drawn upon from linear algebra, restated here for this special case without proof.

Theorem 1.1. *Suppose \mathbf{C} is an $M \times M$ matrix with singular values $\lambda_1 > \lambda_2 > \dots > \lambda_M > 0$. Then for any $P < M$, the matrix given by $\tilde{\mathbf{C}} \equiv \sum_{i=1}^P \lambda_i \mathbf{v}_i \mathbf{v}_i^T$ is the best “rank- P ” approximation of \mathbf{C} in the sense of the Frobenius norm, that is, $\tilde{\mathbf{C}}$ is the minimizer of $\|\mathbf{C} - \mathbf{Z}\|_{\mathcal{F}}$ where $\mathbf{Z} \in \mathbb{R}^{M \times M}$.*

In Theorem 1.1, recall that the Frobenius norm (which is an operator norm) of the matrix \mathbf{C} is given by $\|\mathbf{C}\|_{\mathcal{F}} \equiv \sqrt{\sum_{i=1}^M \lambda_i^2}$. Also, note that $\mathbf{v}_i \mathbf{v}_i^T$ denotes the outer product (sometimes referred to as the “tensor product”) of the vectors \mathbf{v}_i and \mathbf{v}_i^T . Therefore, the theorem serves to guarantee that the reduced basis corresponding to the subspace spanned by the first P eigen vectors is the best possible “rank- P ” choice. In other words, it suffices to determine coefficients α_j , such that the “true” property sought can be expressed as $\mathbf{x} \approx \bar{\mathbf{x}} + \sum_{j=1}^P \alpha_j \mathbf{v}_j$ in the subspace, $\text{span}\{\mathbf{v}_j\}_{j=1}^P$. Hence, the dimension of the system is greatly reduced. The coefficients α_j are actually determined by minimizing an objective function, using the SPSA method, described in the following section.

2. Simultaneous Perturbation, Stochastic Approximation (SPSA)

Consider the problem of finding the root, θ^* , of the problem $\mathbf{g}(\theta) \equiv \nabla L(\theta) = 0$, where $L : \mathbb{R}^P \rightarrow \mathbb{R}$ is assumed to be a differentiable loss function that measures a weighted, time- and space-integrated error between observed measurements (of the phase pressures, concentrations, fluxes, seismic travel-times and well-log data) and computed solutions in some suitable norm. In this work, the expression for L takes the form

$$L(\theta) \equiv \sum_{\mathbf{f} \in \mathcal{D}} \sum_{i=1}^{N_T-1} \left[0.5 \left(\left\| w_{\mathbf{f},i} (\mathbf{f}_i^d - \mathbf{f}_i) \right\|_2^2 + \left\| w_{\mathbf{f},i+1} (\mathbf{f}_{i+1}^d - \mathbf{f}_{i+1}) \right\|_2^2 \right) \Delta t_i \right]. \quad (2.1)$$

Several terms in (2.1) which is written in concise terms, deserve mention. The functions $\mathbf{f} \in \mathcal{D}$ denote any property of interest in the calculations, $\mathbf{f}(\theta)$ which is a function of θ . The set of all such properties is denoted by \mathcal{D} . In this work, for the case of two-phase flow (and specifically the oil water model) $\mathcal{D} \equiv \{\mathbf{p}, \mathbf{N}, \mathbf{u}, \mathbf{q}, \tau\}$, where \mathbf{p} denotes the (oil)phase pressures, \mathbf{N} denotes the (oil)phase concentration, \mathbf{q} denotes the well-data (includes well-rates, bottom-hole pressures, and oil water ratios depending on the kind of well), \mathbf{u} denotes the phase fluxes and τ denotes seismic travel-times. Thus it can be seen that the right-hand side of (2.1) is a function of θ . The superscript d denotes measure data, while terms without the superscript denote computed solution. The index i denotes the time instant at which measurements are recorded and $w_{\mathbf{f},i}$ is a space- and time-dependent weight function on the property \mathbf{f} , assigned based on priority and relevance at a given time instant and location in space. Finally, N_T is the number of time instants in $[0, T]$ when data measurements are recorded and Δt_i denotes the time interval between such measurements.

In this problem, $\theta \equiv \{\alpha_1, \alpha_2, \dots, \alpha_P\}$ with α_i the coefficients in the expansion of the unknown (permeability) vector \mathbf{x} , as described in Section 1. Let θ_k denote the estimate for θ at the k th iterate. Then, the SPSA algorithm has the form

$$\theta_{k+1} = \theta_k - a_k \mathbf{g}_k(\theta_k), \quad (2.2)$$

where $\mathbf{g}_k(\theta_k)$ is a simultaneous perturbation, stochastic approximation of the gradient $\mathbf{g}(\theta_k)$ defined as follows. Let $\Delta_k \in \mathbb{R}^P$ be a vector consisting of $\{\pm 1\}$ values that are randomly generated using a Bernoulli distribution, that is, satisfying $E(\Delta_{k,i}) = 0$, $i = 1, 2, \dots, P$. Then $\mathbf{g}_k(\theta_k)$ is defined by the central-differences equation (component wise)

$$g_{k,i}(\theta_k) = \frac{L(\theta_{k,i} + c_k \Delta_{k,i}) - L(\theta_{k,i} - c_k \Delta_{k,i})}{2c_k \Delta_{k,i}}, \quad i = 1, 2, \dots, P. \quad (2.3)$$

From (2.3), it is evident the method is very cheap because it only involves two relatively inexpensive forward realizations (or simulations) for every iteration to update the estimated property. Here a_k, c_k are monotonically, decreasing sequences of positive scalars chosen according to the method prescribed in [4]. The relevant formulae are given by

$$a_k = \frac{a}{(A + k + 1)^\alpha}, \quad c_k = \frac{c}{(k + 1)^\gamma}, \quad (2.4)$$

where a, c, α, A , and γ are positive real numbers satisfying

$$0 < \alpha \leq 1, \quad \alpha = \gamma > 0.5, \quad \alpha > 2\gamma. \quad (2.5)$$

This ensures that some technical conditions are satisfied [4] which are in turn required for the convergence of the stochastic gradient to the steepest descent gradient. The choice of a, c, α, A and γ is to some extent case dependent and may require some experimentation. It is known that $\alpha = 1$ and $\gamma = 1/6$ are asymptotically optimum values but choosing smaller values, for example, $\alpha = 0.602$ and $\gamma = 0.101$ are found to be effective in practise. A common recommendation [4] is to set A equal to 5 to 10 percent of the maximum number of iterations allowed.

Spall [4] has shown that the method converges and can be regarded as stochastic analogue of the steepest descent method with similar rates of convergence. Therein, proofs of convergence of the method in a “stochastic sense” (i.e., in the sense of expectations) are presented. Similar proofs can also be found in [10]. For completeness, the basic steps of a proof are presented here to show that the expectation of the stochastic gradient $\mathbf{g}_k(\theta_k)$ equals the actual gradient $\mathbf{g}(\theta_k)$. Thus, it can be expected that the method will converge at a rate equal to the steepest descent method.

Theorem 2.1. *The expectation of the stochastic gradient $\mathbf{g}_k(\theta_k)$ of the SPSA method equals the true gradient $\mathbf{g}(\theta_k)$.*

Proof. A standard Taylor-series expansion yields that

$$L(\theta_k + c_k \Delta_k) = L(\theta_k) + c_k \sum_{j=1}^P \Delta_{k,j} \frac{\partial L(\theta_k)}{\partial \theta_{k,j}} + \mathcal{O}(c_k^2 \|\Delta_k\|^2), \quad (2.6)$$

and likewise

$$L(\theta_k - c_k \Delta_k) = L(\theta_k) - c_k \sum_{j=1}^P \Delta_{k,j} \frac{\partial L(\theta_k)}{\partial \theta_{k,j}} + \mathcal{O}(c_k^2 \|\Delta_k\|^2). \quad (2.7)$$

Subtracting (2.7) from (2.6), then rearranging the resulting terms and neglecting the higher-order terms, $\mathcal{O}(c_k^2 \|\Delta_k\|^2)$ yields

$$\frac{L(\theta_k + c_k \Delta_k) - L(\theta_k - c_k \Delta_k)}{2c_k} = \sum_{j=1}^P \Delta_{k,j} \frac{\partial L(\theta_k)}{\partial \theta_{k,j}} = \Delta_k^T \nabla L(\theta_k). \quad (2.8)$$

Next, define the inverse of Δ_k as

$$\Delta_k^{-1} = [\Delta_{k,1}^{-1}, \Delta_{k,2}^{-1}, \dots, \Delta_{k,P}^{-1}]. \quad (2.9)$$

Then, from $\Delta_{k,i}^{-1} = \Delta_{k,i}$ (recall that $\Delta_{k,i} = \pm 1$), it follows that

$$\Delta_k^{-1} = \Delta_k. \quad (2.10)$$

From (2.10), it follows that the stochastic gradient in (2.3) can be written as

$$\mathbf{g}_k(\theta_k) = \frac{L(\theta_k + c_k \Delta_k) - L(\theta_k - c_k \Delta_k)}{2c_k} \Delta_k = \Delta_k \frac{L(\theta_k + c_k \Delta_k) - L(\theta_k - c_k \Delta_k)}{2c_k}. \quad (2.11)$$

Then, using (2.8) in (2.11) yields

$$\mathbf{g}_k(\theta_k) = \Delta_k \Delta_k^T \nabla L(\theta_k). \quad (2.12)$$

The expectation of the stochastic gradient is then given by

$$E[\mathbf{g}_k(\theta_k)] = E[\Delta_k \Delta_k^T] \nabla L(\theta_k). \quad (2.13)$$

Observe that the form of the matrix $\Delta_k \Delta_k^T$ on the right-hand side of (2.13) is given by

$$\Delta_k \Delta_k^T = \begin{bmatrix} \Delta_{k,1}^2 & \Delta_{k,2} \Delta_{k,1} & \cdots & \Delta_{k,P} \Delta_{k,1} \\ \Delta_{k,1} \Delta_{k,2} & \Delta_{k,2}^2 & \cdots & \Delta_{k,P} \Delta_{k,2} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{k,1} \Delta_{k,P} & \Delta_{k,2} \Delta_{k,P} & \cdots & \Delta_{k,P}^2 \end{bmatrix}. \quad (2.14)$$

In (2.14), note that for any $j = 1, 2, \dots, P$, the only value that $\Delta_{k,j}^2$ can take is 1. Hence,

$$E\left[\Delta_{k,j}^2\right] = 1. \quad (2.15)$$

Further, because $\Delta_{k,i}$ and $\Delta_{k,j}$ are independent random variables (when $i \neq j$), it follows that

$$E\left[\Delta_{k,i}\Delta_{k,j}\right] = E\left[\Delta_{k,i}\right]E\left[\Delta_{k,j}\right] = 0, \quad \forall i \neq j. \quad (2.16)$$

From this, it follows also that

$$E\left[\Delta_k\Delta_k^T\right] = \mathbf{I}, \quad (2.17)$$

where \mathbf{I} denotes the $P \times P$ identity matrix. Applying (2.14)–(2.16) to (2.13) yields

$$E[\mathbf{g}_k(\theta_k)] = \nabla L(\theta_k) \equiv \mathbf{g}(\theta_k), \quad (2.18)$$

which is the desired result. □

3. A Parallel SPSA Algorithm

In this section, a parallel SPSA algorithm is described that runs several instances of the basic SPSA algorithm, one on each processor. This helps improve the convergence by widening the search space. Numerical tests were performed on various challenging problems on upto 256 processors. Sometimes, convergence is obtained in as few as 2 or 3 iterations. Each processor has its own copy of the vector θ_k , (i.e., the coefficients of the natural logarithm of permeability field), represented by say, θ_k^{id} . The random vectors Δ_k are generated on each processor and are not the same as those on other processors. This is easily achieved by using a different seed on each processor for the Bernoulli random number generator program that can be found, for example, in [11]. Thus each processor also has a copy of the stochastic gradient and updates θ_k according to (2.3).

Figure 1 shows the flow chart of the parallel SPSA algorithm for a single SPSA iteration step. Most of the steps in the figure are self-explanatory. The superscript *id* in θ_k^{id} represents the processor ID and *np* is the total number of processors. The main step that needs to be described is the box “All Gather mean/min”. Two approaches are implemented to gather θ_k^{id} from each processor and step to θ_{k+1}^{id} —the “mean” and “min” approaches. In the “mean” method, as the name indicates, the updated vector, θ_{k+1} broadcast to all processors for the next iteration is simply the mean of the processor updates, θ_{k+1}^{id} , that is,

$$\theta_{k+1} = \frac{1}{\text{np}} \sum_{i=1}^{\text{np}} \theta_{k+1}^{\text{id}}. \quad (3.1)$$

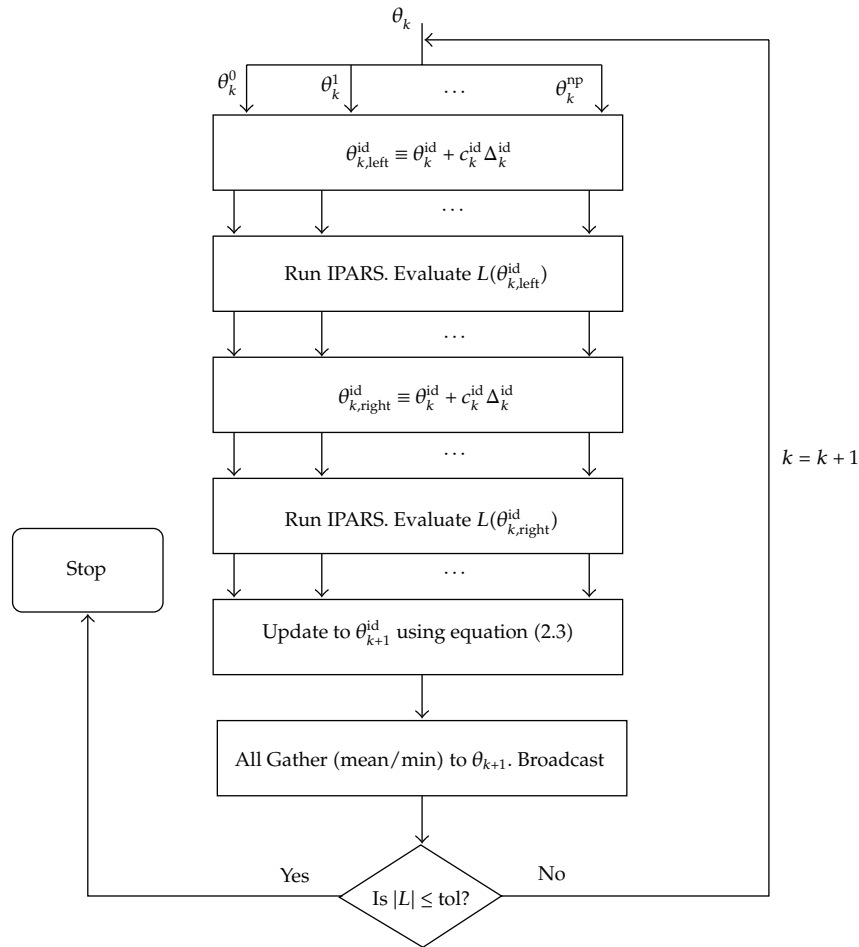


Figure 1: A parallel SPSA algorithm.

In the “min” method, again as the name indicates, first the processor with the least objective is identified. In other words, the index of that processor, id^{min} say, is defined first as

$$id^{min} \equiv \min_{1 \leq id \leq np} L(\theta_{k+1}^{id}). \tag{3.2}$$

Then the value of the vector $\theta_{k+1}^{id^{min}}$ on the processor id^{min} is broadcast to all others for the next iteration

$$\theta_{k+1} = \theta_{k+1}^{id^{min}}. \tag{3.3}$$

The mean method was observed to be more stable and robust in general while the min method, although faster, sometimes exhibited the tendency to get trapped in local minima. Each box with text “Run IPARS..” in the flow chart of Figure 1 is a forward run of the reservoir simulator IPARS [12] with the value of permeability being calculated from the

current perturbation of the value of vector, θ_k^{id} (i.e., left or right perturbation). It is also noted that in practise, the implementation takes the components of θ_k on each processor to be the coefficients of $\ln x$ in place of x . (i.e., natural log of permeability instead of permeability), since the permeabilities can vary by several orders of magnitude in most real-world problems. The permeability x can then be easily calculated from $\ln x$ using an exponential transformation.

4. Numerical Results

Two examples are presented in this section. In the first example, a 2d heterogeneous permeability field derived from the 10th SPE project [13] is used to test the parallel parameter estimation and history matching implementation. The second example likewise determines the heterogeneous permeability field and performs history matching for an upscaled version of the synthetic “Brugge” field [14] test case.

For both tests, basis functions are first generated using sample realizations (see Section 1). A total of 8 basis functions are used in the expansion of permeability for the first example where an isotropic permeability field is assumed. A total of 10 basis functions for each direction (i.e., K_{xx} , K_{yy} , and K_{zz}) is assumed for the second example (this is a nonisotropic case). Both examples use the two-phase hydrology (oil and water), model as the base model in IPARS to perform history matching and parameter estimation. It is noted however that IPARS can handle general multiphase and compositional flow problems using various discretization methods. The reason for the choice here is that the inverse modeling framework is currently only supported with the single-phase and two-phase models of IPARS. Both problems presented here were tested on various parallel platforms including the Bevo2 and Ranger linux cluster at The University of Texas at Austin and Blue Gene cluster, IBM.

4.1. Example 1: Sensor Tests

In this example, the goal was to perform history matching and estimate the permeability field (assuming a known “true” permeability field). Knowledge of the “true” permeability field only serves to validate the final answer and is not required (and is in fact not known) in practise. Without this assumption, the parallel SPSA implementation can guarantee that the objective function is minimized, but that does not necessarily mean that the minimizer is the (unique) real permeability field. In practise, the objective functions can have several local minima, hence it is important to ensure that the algorithm not only minimized the objective function, but that it is also converging to the “true” permeability field.

In this example, different objective function combinations based on different sensor combinations were tested (by activating or deactivating the weight functions $w_{\ell,i}$ in (2.1)). The problem simulates an oil gas immiscible model using the hydrology model in IPARS. This is achieved by treating in the input, oil phase as the actual “gas-phase” and assigning the properties of gas to it. Likewise, the water-phase is treated in the input as the actual “oil phase”. Then the oil phase (actually “gas”) is injected to produce water (actually “oil”). While this may sound confusing, it is quite common in reservoir simulation to reuse existing two- or three-phase codes to solve problems in contexts they may not have been designed for. The full set of properties \mathcal{D} , is available for use in the objective function but as mentioned, some may be “turned-off” by setting the weights to zero. It is also noted that only the production well was included in the objective calculations. Table 1 summarizes the problem data describing this example.

Table 1: Parameter estimation: data for example 1, (based on 10th SPE permeability set).

Property	Value
Simulation end-time, days	1000.0
Grid dimensions	$100 \times 1 \times 20$
Physical dimensions, m^3	$2500 \times 25 \times 50$
Flowing phases modeled	Gas, oil
Reference gas density, lb/ft^3	0.06
Gas viscosity, cp	0.01
Reference oil density, lb/ft^3	43.68
Oil viscosity, cp	1.0
Gas compressibility, $1/psi$	2.0×10^{-4}
Oil compressibility, $1/psi$	1.0×10^{-6}
Initial (residual) gas pressure, psi	1000
Initial (residual) gas conc., lb/ft^3	0.006
Medium porosity	0.2
Number of wells	2 (gas-inj. and oil prod.)
Gas injection rate, lb/day	43.84
BHP of oil prod. well, psi	995.0

Since the “true” θ^* , that is, the coefficients corresponding to the actual permeability field are known, the problem is first run using the true permeability and the true solution (\mathbf{p}^* , \mathbf{N}^* , \mathbf{q}^* , \mathbf{u}^* , and τ^*) are recorded at discrete time instants and at all the grid elements (or faces in case of fluxes). Seismic travel times are recorded as the set of times it takes for seismic waves to travel from one end of the reservoir (from each element face on the “source” end) to the other end (each element face on the “target” end). These are then recorded as data. The SPSA iterations are then performed starting with an initial guess θ_0 that is obtained by randomly perturbing the “true” θ^* . A total of 1000 SPSA iterations were computed for this test, although the permeability field converged in very few (early) iterations.

Figure 2 shows the history matching versus time in days, of the oil phase (actually, “gas”) pressures in units of psi shown across a section through the middepth of the 2d reservoir. In this figure, it is noted that Figure 2(a) presents the match obtained when all sensors were active (i.e., the weights $w_{f,i} \equiv 1$ at all element locations and time instants). Figure 2(b) presents the match obtained when only the production well-sensor and the solution sensors (i.e., $w_{p,i}$, $w_{u,i}$, and $w_{N,i}$) at the midsection were active (i.e., at the location $x = 1250$ m and for all time instants). Figure 2(c) presents the match obtained when only the production well-sensor was active. The seismic sensor result was not included here because it was found to influence the history match in very insignificant amounts.

From the Figure 2, it is clear that sensors are not required at every grid element (and potentially not at every time instant as well) which is reassuring since installing such costly equipment as sensors at high areal densities can be very expensive and impractical. Figure 3 shows a similar match for the oil phase (actually “gas”) concentration in units of $lb/cu-ft$ across a section passing through the middepth of the reservoir. Once again a fairly good match is obtained with fewer measurements. As a demonstration of the flux matches, Figure 4 shows the history match obtained for oil phase fluxes (actually “gas”) at the same section as the previous cases.

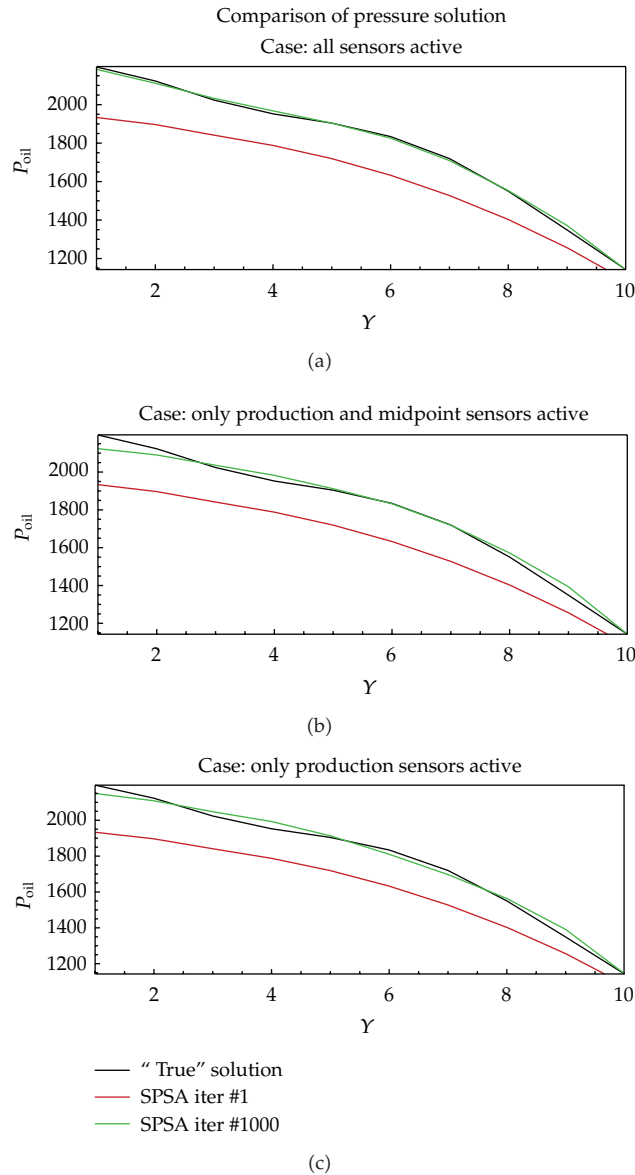


Figure 2: Example 1: history matching of oil phase pressures based on sensor choices.

It is observed that flux match is poorer when fewer sensors are used (as opposed to the quality of the pressure and concentration history matches). Finally, the production well-data history matches versus time in days of oil production (actually “gas”) in mscf/day and oil gas (actually “water-gas”) ratio and permeability estimation (for the case when only the midsection and production well sensors are active) is shown in the Figures 5, 6, and 7, respectively. For the permeability estimate, it is pointed out that the legend spans 50 to 900 mD to allow consistent comparison. From these graphs, the convergence of the parallel SPSA method can be seen to be very effective.

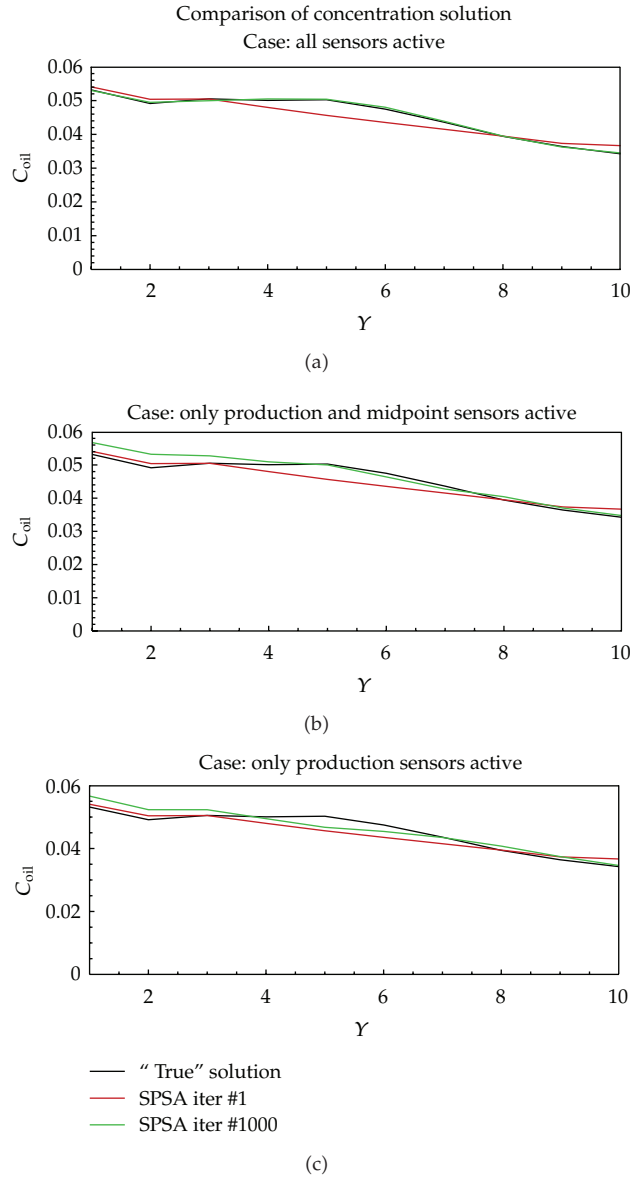


Figure 3: Example 1: history matching of oil phase concentrations based on sensor choices.

4.2. Example 2: Brugge Field Test

In this example, a synthetic reservoir referred to as the Brugge field is used for the purpose of history matching and permeability estimation. This synthetic field was constructed by the Norwegian research center, TNO in February 2008 for a comparative case study on history matching and reservoir characterization. It consisted of 104 upscaled realizations of a 3-D geological model with well-log data from 30 wells with fixed spatial positions; first 10 years production history; inverted time-lapse seismic data in terms of pressures and saturations as well as economic parameters for oil, water and discount rates. A more detailed description

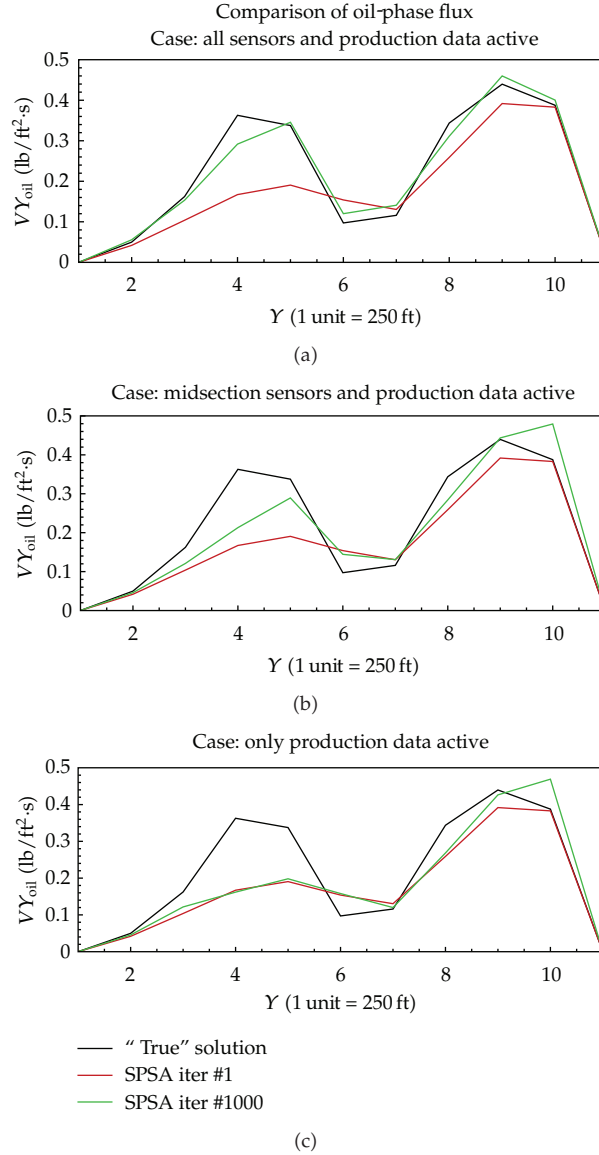


Figure 4: Example 1: history matching of oil phase fluxes based on sensor choices.

as well as the data from the field can be found at [14]. This problem is very challenging for several reasons.

The computational domain is a full 3d domain, it is irregular in geometry with a geologic fault as shown in Figure 8. The permeability field is anisotropic and very heterogeneous and hence, many more unknowns have to be estimated in this case. Finally, there are 30 wells driving the flow in this oil water problem (actually oil water!). There are 10 water injection wells and 20 oil production wells that can be shut-in based on rate or bottom-hole constraints. All these combine to make this a fairly challenging problem. Table 2 summarizes the data that describe the problem that was solved. It is noted that well #19

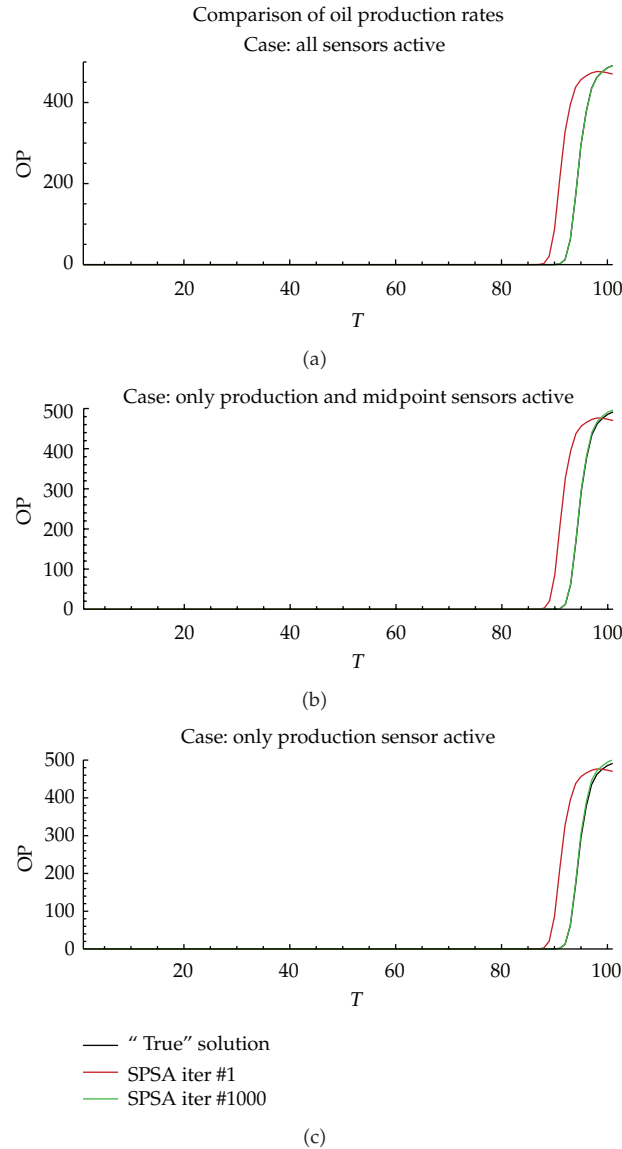


Figure 5: Example 1: history matching of well-data based on sensor choices.

(a production well) was the only bottom-hole pressure specified well without any constraint. All other wells were rate specified with a bottom-hole pressure constraint for shut-in. It is also noted that different injectors start injecting water at different times (earliest at $t = 600$ days) while the producers start producing as early as $t = 0$ days.

The first challenge with this problem was in accurately modeling the geometry of the domain with the faults. A stair-stepped approximation was once again employed to treat the geometry in keeping with the stencil resulting from mixed FEM [15]. A bounding box grid of $35 \times 21 \times 57$ was used and suitably interpolated to get values of the properties at corners of the box that intersect with the actual domain. For all other points, negative values are

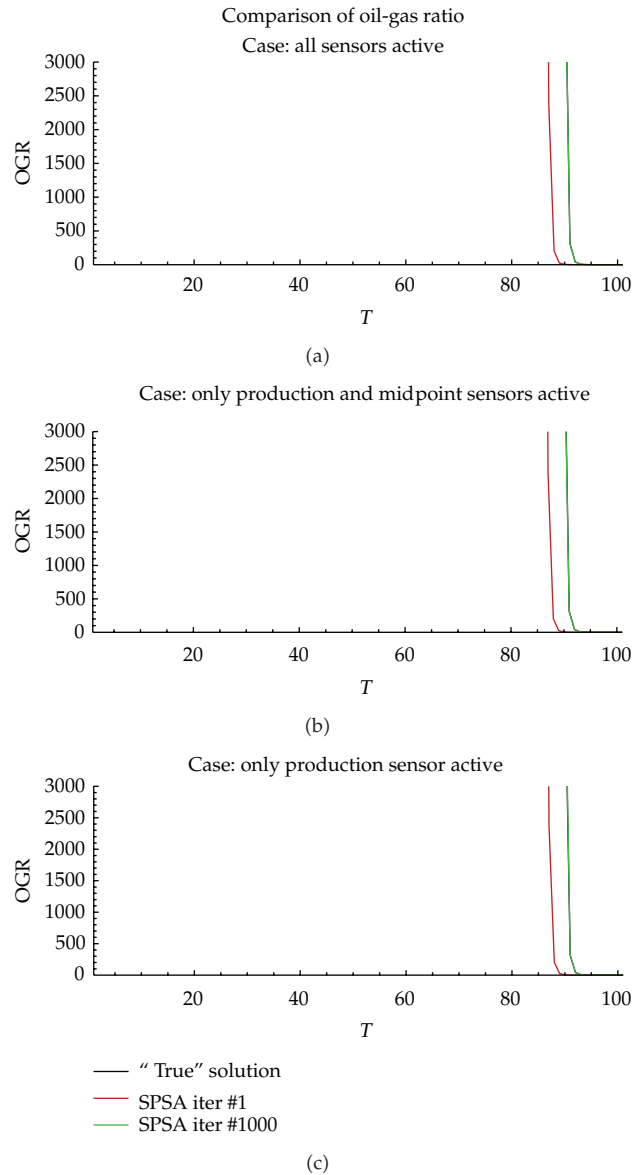


Figure 6: Example 1: history matching of well-data based on sensor choices.

assigned and these are used to keyout the elements that are actually inactive in the bounding box (i.e., for the corners that do not intersect the actual domain). Figure 8 shows the true and approximated geometries with the initial water saturation and mesh superimposed. The well locations are indicated by colored spheres, black for injectors and orange for producers.

A parallel SPSA history matching simulation was performed on the Brugge field for $P = 10$ unknowns in each direction of anisotropy (i.e., a total of 30 unknowns) and a subset ($M = 16$) of the realizations. For proof of concept, the mean permeability was assumed to be the true permeability and data (measurements) derived from the solution corresponding to the mean at prescribed intervals of time at all grid elements. Again, it is noted that this

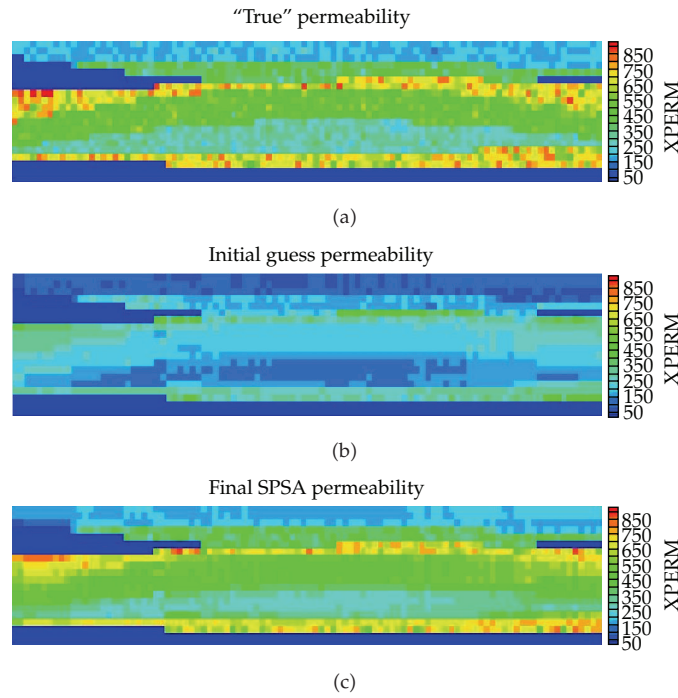


Figure 7: Example 1: permeability estimation using SPSA.

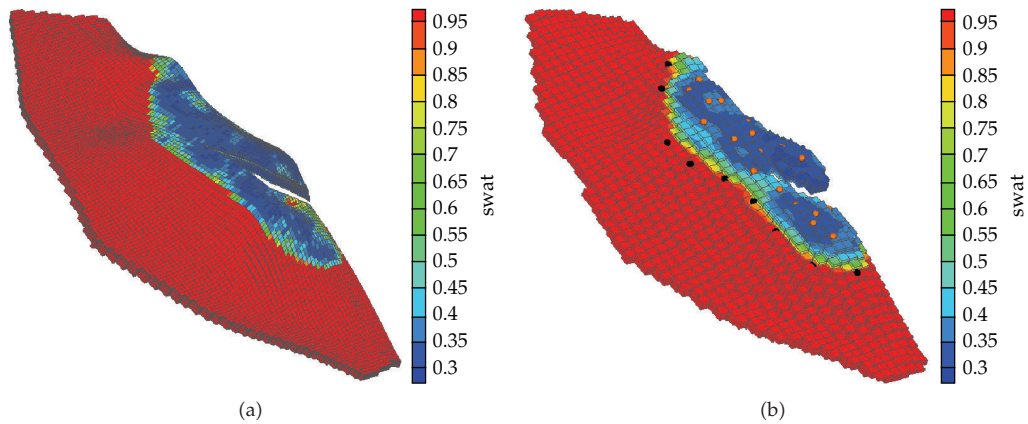
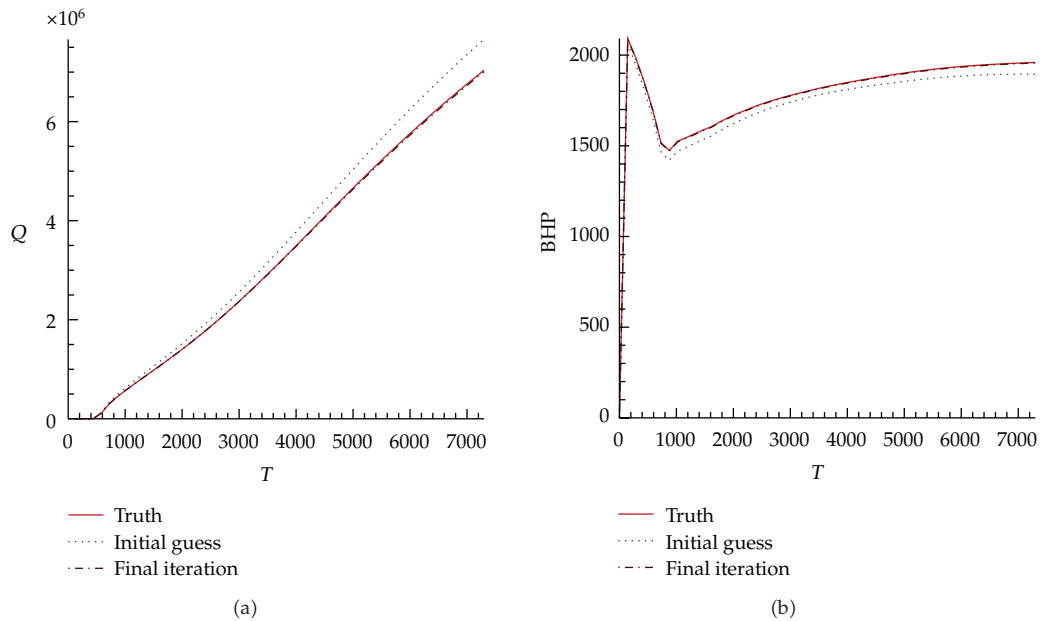


Figure 8: Example 2 (Brugge field): true (a), approximate (b) geometry with initial water saturation and grid superimposed.

is strictly not a required step, since ideally history matching is the only goal in such studies while estimated permeability is a by-product. But assuming a "true" permeability, allows one to test the effectiveness of the applied algorithm. Also, it is assumed that the history matching is performed for 20-year simulation period. Figure 10 shows the permeability estimate when the production data and solution (pressures and concentrations) measured at intervals of 5 grid elements in either areal direction are included in the loss function. The legend again spans 50 to 900 mD for all iterations for ease of comparison.

Table 2: Parameter estimation: data for example 2 (Brugge field test).

Property	Value
Simulation end-time, years	20.0
Grid dimensions	$21 \times 57 \times 35$
Physical dimensions, ft^3	$21000 \times 57000 \times 1750$
Flowing phases modeled	Oil, water
Reference oil density, lb/ft^3	56.0
Oil viscosity, cp	1.29
Reference water density, lb/ft^3	62.6
Water viscosity, cp	0.32
Oil compressibility, $1/\text{psi}$	9.26×10^{-6}
Water compressibility, $1/\text{psi}$	3.0×10^{-6}
Initial oil pressure, psi	≈ 2500
Initial oil conc., lb/ft^3	Residual-water (0.2)
Medium porosity	≈ 0.2
Number of wells	30 (10 water-inj. and 20 oil prod.)
Water injection rate, lb/day	4000.0
Oil-prod rate, lb/day	2000.0

**Figure 9:** Example 2 (Brugge field test): well 19 (a) production curves and well 21 (b) BHP history match.

Since the physical dimensions of the Brugge formation was huge, this was a reasonable areal density (about 1 sensor every 4 square km) for sensor locations. It is observed that even though the initial guess was very far from the “target” or “true” permeability, the algorithm “covers” in as few as 5–10 iterations. To fix ideas, it is noted that the initial permeability error is more than 60% (measured in an l_2 -norm) and the normalized objective function value is 0.84 approximately. At the end of 10 iterations the permeability error is reduced to 25% and

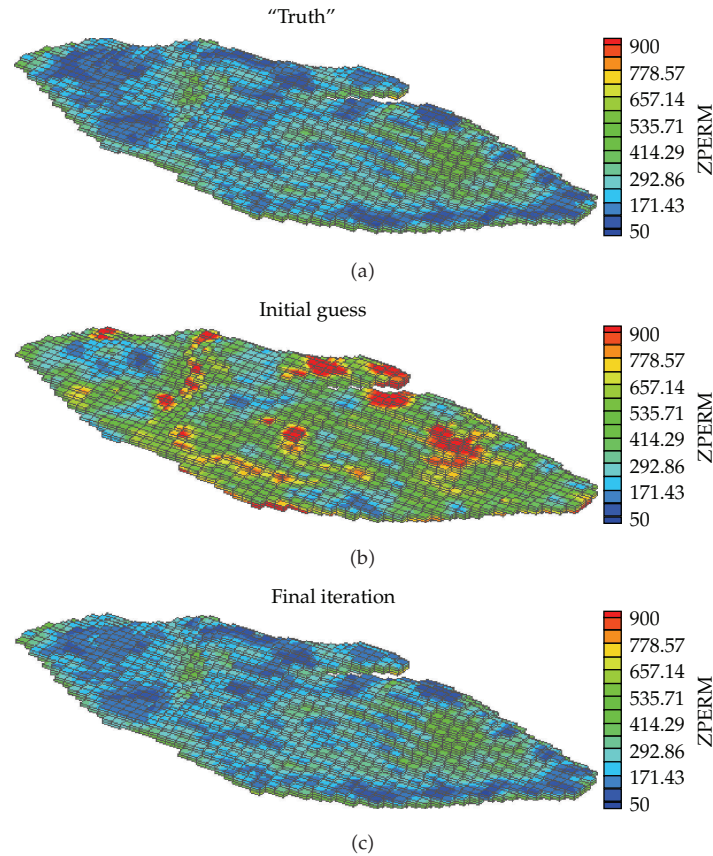


Figure 10: Example 2 (Brugge field test): estimate of permeability with SPSA iteration number.

the objective function to 0.076! At the end of 50 iterations, the permeability error is reduced to 13% and the objective to 0.016. Finally, at the end of 200 iterations, the permeability error is reduced to 3.5% while the objective is reduced to 0.001! These numbers are very encouraging given the problem at hand.

Figure 9 shows the history match versus time in days obtained with respect to bottom-hole pressures (in psi, shown for rate specified producing well #21) and cumulative production rate (in lb/day, shown for bottom-hole pressure specified producing well #19). The matches obtained for all wells were equally good. These parallel runs were performed on up to 64 processors on the Bevo2 cluster at ICES as well as on the Ranger cluster (at TACC, The University of Texas at Austin) on up to 512 processors.

4.3. Conclusions

This paper presents a robust and efficient parallel stochastic framework for subsurface reservoir characterization and history matching problems using sensor measurements of data from general multiphase flow scenarios. The parallel algorithm involves triggering multiple SPSA (simultaneous perturbation, stochastic approximation) instances, one on each processor which uses its own realization of the estimated property based on its own seed. At the end of each iteration, two options are available to compute the starting value of the

estimated property for the ensuing iteration; one based on the minimum objective across all processors and the other based on the mean of the properties across all processors. It is generally observed that a very accurate history match is obtained in as few as 5 to 10 iterations even for fairly general and complex problems. If a “true” permeability is assumed, a very accurate permeability match is also attained in as few iterations. The algorithm itself is shown to converge at a rate equivalent to that of the steepest descent method in the sense of expectations. A natural extension of this work would be to support parallel domain decomposition in each SPSA “instance” of the algorithm, so that the method not only takes advantage of a wider search space but also utilizes a divide and conquer strategy within each search path. This is perhaps the optimal strategy for such problems and may form the subject of a future work.

Acknowledgments

A portion of this paper was supported by the United States Department of Energy (DoE) through the Office of Basic Energy Sciences, Office of Science. The Center for Frontiers of Subsurface Energy Security (CFSES) is a DOE Energy Frontier Research Center, set up under contract no. DE-SC0001114. The authors gratefully acknowledge the financial support provided by the NSF-CDI under Contract no DMS 0835745 and DOE Grant no. DE-FGO2-04ER25617. Finally, the first author thanks Dr. G. Pencheva and Dr. T. Wildey for their help with input data used in the second numerical example.

References

- [1] R. Versteeg, M. Ankeny, J. Harbour et al., “A structured approach to the use of near-surface geophysics in long-term monitoring,” *Leading Edge*, vol. 23, no. 7, pp. 700–703, 2004.
- [2] D. E. Lumley and M. J. Blunt, “Time-lapse seismic reservoir management,” *Geophysics*, vol. 66, pp. 50–53, 2001.
- [3] H. Kile, W. Bangerth, M. F. Wheeler, M. Parashar, and V. Matossian, “Parallel well-location optimization using stochastic algorithms on the grid computational framework,” in *Proceedings of the 9th European Conference on the Mathematics of Oil Recovery (ECMOR '04)*, Cannes, France, 2004.
- [4] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience, Hoboken, NJ, USA, 2003.
- [5] R. E. Banchs, H. Klie, A. Rodriguez, S. G. Thomas, and M. F. Wheeler, “A neural stochastic multiscale optimization framework for sensor-based parameter estimation,” *Integrated Computer-Aided Engineering*, vol. 14, no. 3, pp. 213–223, 2007.
- [6] H. Klie, A. Rodriguez, S. G. Thomas, M. F. Wheeler, and R. Banchs, “Assessing the value of sensor information in 4-D seismic history matching,” *SEG Technical Program Expanded Abstracts*, vol. 25, no. 1, pp. 3245–3249, 2006.
- [7] P. Lu, *Reservoir parameter estimation using wavelet analysis*, Ph.D. thesis, Stanford University, Palo Alto, Calif, USA, 2001.
- [8] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Physical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.
- [9] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2002.
- [10] A. C. Reynolds, G. Gao, and G. Li, “A stochastic optimization algorithm for automatic history matching,” in *Proceedings of the SPE Annual Technical Conference and Exhibition*, Houston, Tex, USA, September 2004.
- [11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

- [12] J. Wheeler et al., "Integrated parallel and accurate reservoir simulator," Tech. Rep. TICAM01-25, Institute of Computational Engineering and Sciences, The University of Texas at Austin, Austin, Tex, USA, 2001.
- [13] M. A. Christie and M. J. Blunt, "Tenth SPE comparative solution project: a comparison of upscaling techniques," *SPE Reservoir Evaluation and Engineering*, vol. 4, no. 4, pp. 308–316, 2001.
- [14] E. Peters, R. J. Arts, G. K. Brouwer et al., "Results of the brugge benchmark study for flooding optimization and history matching," *SPE Reservoir Evaluation and Engineering*, vol. 13, no. 3, pp. 391–405, 2010.
- [15] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, vol. 15 of *Springer Series in Computational Mathematics*, Springer, New York, NY, USA, 1991.