Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Norwegian University of Science and Technology, Trondheim, Norway, May 31 - June 1, 2012

Estimation of Optimal Well Controls Using the Augmented Lagrangian Function with Approximate Derivatives ⋆⋆

Sy T. Do * **Fahim Forouzanfar** ** **Albert C. Reynolds** ***

* *McDougall School of Petroleum Engineering, The University of Tulsa, Tulsa, OK 74104 USA (e-mail: sy-do@ utulsa.edu).*
** *McDougall School of Petroleum Engineering, The University of Tulsa, Tulsa, OK 74104 USA (e-mail: fahim-forouzanfar@ utulsa.edu).*
*** *McDougall School of Petroleum Engineering, The University of Tulsa, Tulsa, OK 74104 USA (e-mail: reynolds@ utulsa.edu).*

Abstract: When efficient adjoint code for computing the necessary gradients is available, the augmented Lagrangian algorithm provides an efficient and robust method for constrained optimization. Here, we develop an augmented Lagrangian algorithm for constrained optimization problems where adjoint code is not available, and the number of optimization variables is so large that the approximation of gradients with the finite-difference method is not computationally feasible. Our procedure applies a pre-conditioned steepest ascent algorithm to maximize an augmented Lagrangian function which directly incorporates all bound constraints as well as all inequality and equality constraints. The pre-conditioned gradient of the augmented Lagrangian is estimated directly using a simultaneous perturbation stochastic approximation (SPSA) with Gaussian perturbations where the preconditioning matrix is a covariance matrix selected to impose a degree of temporal smoothness on the optimization variables, which, for the specific application considered here, are the well controls. Our implementation of this augmented Lagrangian method is applied to estimate the well controls which maximize the net present value (NPV) of production for the remaining life of a given oil reservoir.

Keywords: Optimal Well Controls, Constrained Production Optimization, SPSA.

1. INTRODUCTION

Our focus is on the production optimization step of closed-loop reservoir management (Brouwer and Jansen, 2004; Jansen et al., 2005; Peters et al., 2010). In this step, we wish to find well controls (well operating conditions) which maximize the net-present-value (NPV) of production (or cumulative oil production) over the presumed reservoir life. Here, we consider only the maximization of an NPV functional subject to the condition that all constraints are satisfied. Throughout, the well controls which solve this constrained optimization problem are referred to as the optimal well controls. These controls take the form of either specified flow rates or flowing bottom-hole pressures for all wells on each of a set of specified time intervals, the controls steps, which span the remaining life of the reservoir. When an ensemble of plausible reservoir models is available, it is best to do robust optimization (van Essen et al., 2009). As our focus is on the development and application of a reliable optimization procedure for estimating the optimal well controls, we only consider optimization based on a single reservoir model.

There is ample evidence that the optimal well control problem can be efficiently solved with a gradient-based algorithm (Brouwer and Jansen, 2004; Nævdal et al., 2006; Kraaijevanger et al., 2007; Sarma et al., 2008) with the gradient of the objective function that we wish to maximize computed by the adjoint method using an adjoint formulation similar to the one presented by Li et al. (2003) for history matching. When a gradient-based method for maximization of an objective function converges, it converges to a local maximum, and, in general, it is possible that such a local maximum corresponds to an objective function value which is much smaller than the global optimum. However, when the number of control variables in the production optimization problem is large, a gradient-based maximization algorithm at convergence typically results in a value of the objective function which is close to the global maximum. This is not a happenstance but is due to the fact that the hyper-surface of the objective function for the optimal control problem is characterized by hyper-plateaus along which the objective function is almost constant; the existence of these plateaus is implicitly implied by the results of van Essen et al. (2009).

Unfortunately, commercial simulators have limited capability for computing the gradients needed for general constrained optimization with the adjoint method. We should note, however, that there are cases where commercial code performs well; for example, when well controls are rates

⋆ This work was supported financially by the member companies of TUPREP.
⋆⋆Contribution to invited session "Closed-loop production optimization in reservoir engineering."

and the constraints are linear, all gradients needed to perform optimization can be obtained from the output of Eclipse 300 (Chen et al., 2010). When we are required to use a specific commercial code for the forward reservoir simulation run, but the code does not output all the gradients needed for the production optimization problem, it is necessary to apply an optimization algorithm which does not require an exact gradient computed from the adjoint method. There are a plethora of such so-called derivative-free optimization (DFO) algorithms; the performance of several of them, for the production optimization problem considered here, was compared in Zhao et al. (2011). However, Zhao et. al. only considered bound constraints. In essence, the objective of this paper is to show that with proper care in their implementation, derivative-free methods can be applied to the general constrained production optimization problem.

To accomplish the aforementioned objective, we modify the augmented Lagrangian method used by Chen et al. (2010) to solve the general constrained optimization problem. In the procedure of Chen et al., all gradients were computed with the adjoint method and the bound constraints were enforced using a standard gradient projection method. The algorithm presented here modifies the Chen et al. procedure in two fundamental ways: (i) gradient projection is not used; instead bounds are enforced using a log-transform, and (ii) the pre-conditioned gradient of the augmented Lagrangian is approximated by a simultaneous perturbation stochastic approximation (SPSA), and then a preconditioned steepest-ascent algorithm is applied to estimate the optimal well controls subject to the constraints. In this methodology, a log-transform is applied to each of the original well control variables, and optimization is performed on the transformed variables. As discussed below, the log-transformation ensures that bound constraints are automatically satisfied so that the standard gradient projection technique used to enforce bound constraints is not needed. Although it will not be discussed further in this short paper, we note that because we only generate approximate gradients, the gradient projection method does not always yield reliable results. Thus, we were forced to find a way to avoid the use of gradient projection. Conceptually, bound constraints can be handled by simply truncating a control variable to the appropriate bound when the variable moves outside the bounds and setting that component of the gradient equal to zero, or by transforming bound constraints to inequality constraints. However, extensive computational experiments have established that handling bounds via the log-transform yields better final NPV's than are obtained using either of the other aforementioned two methods for dealing with bound constraints (Do, 2012).

To conclude the introduction, we note that SPSA (Spall, 2003) provides an efficient alternative to the finite-difference method for estimating the gradient of an objective (cost) function when the number of optimization variables is large. The SPSA method and a derivative of SPSA based on Gaussian perturbations have been successfully applied in a variety of petroleum engineering applications (Bangerth et al., 2006; Li and Reynolds, 2011). Also note that although we couple the SPSA method to compute the required gradients in an augmented Lagrangian formulation, it is also conceivable that the SPSA could be used to compute the required gradients in a sequential quadratic programming algorithm.

2. PROBLEM FORMULATION

The specific production optimization problem considered here pertains to estimating the optimal well controls when waterflooding an oil-reservoir. We assume that there is no gas injection and ignore the income from, or cost of disposal of produced gas so that the net-present-value (NPV) functional is defined by

$$J(w, y(w)) = \frac{\sum_{n=1}^{N}\left[\sum_{i=1}^{N_{\rm prod}}(r_{\rm o}\bar{q}^n_{{\rm o},i} - r_{\rm w}\bar{q}^n_{{\rm w},i}) - \sum_{i=1}^{N_{\rm winj}} r_{\rm winj}\bar{q}^n_{{\rm winj},i}\right]\Delta t^n}{(1+b)^{t^n/365}}. \quad (1)$$

Throughout, w is the vector of all well controls at all control steps; y is the vector of reservoir simulation primary variables at all simulation time steps; N is the total number of reservoir simulation time steps; $N_{\rm prod}$ is the total number of producers; $N_{\rm winj}$ is the total number of water injection wells; $r_{\rm o}$ is the oil revenue (\$/STB); $r_{\rm w}$ is the water disposal cost (\$/STB); $r_{\rm winj}$ is the water injection cost (\$/STB); $\bar{q}^n_{{\rm o},i}$ is the average oil production rate of the i^{th} producer (STB/day) during the n^{th} time step; $\bar{q}^n_{{\rm w},i}$ is the average water production rate of the i^{th} producer (STB/day) during the n^{th} time step; $\bar{q}^n_{{\rm winj},i}$ is the average water injection rate of the i^{th} injection well (STB/day) during the n^{th} time step; b is the annual discount rate; t^n is the cumulative time (days) up to the n^{th} time step; Δt^n (days) is the n^{th} simulation time step.

The NPV defined in Eq. (1) is a function of the well control vector w and the dynamic state vector y, which is the vector of variables solved for by the reservoir simulator, i.e., pressures and saturations. The well control variables include the water injection rate or the bottomhole pressure (BHP) of the injection wells and the production rate or the BHP of the production wells. The maximization of the NPV in Eq. (1) is usually subject to equality, inequality and bound constraints, respectively, given by

$$e_j(w, y(w)) = 0,\ j = 1, .., n_{\rm e}, \quad (2)$$

$$c_j(w, y(w)) \le 0,\ j = 1, .., n_{\rm ine}, \quad (3)$$

and

$$w_i^{\rm low} \le w_i \le w_i^{\rm up},\ i = 1, 2, ..., n_{\rm w}, \quad (4)$$

where $n_{\rm e}$, $n_{\rm ine}$ and $n_{\rm w}$, respectively, denote the number of equality, inequality and bound constraints. Requiring the field water injection rate to be equal to field liquid production rate is an example of a highly nonlinear equality constraint. Requiring the field and individual water cut to be less than a prescribed value is a nonlinear inequality constraint. The requirement that the sum of the rates at water injection wells be equal to a specified value represents a linear equality constraint. Constraints arise naturally due to the operational limits of the production and injection facilities.

2.1 Simple bound constraints

In the methodology presented here, a log-transform (Zhao et al., 2011) is applied to enforce the bound constraints. This transform is defined by

$$u_i = \ln\left(\frac{w_i - w_i^{\text{low}}}{w_i^{\text{up}} - w_i}\right). \quad (5)$$

From Eq. (5), it is apparent that as the value of the control variable w_i approaches its lower bound (w_i^{low}) from above, $u_i \to -\infty$, and when $w_i \to w_i^{\text{up}}$ from below, $u_i \to \infty$. Thus, there are no bound constraints on the components of the transformed control vector u. By doing optimization in terms of u, we avoid the use of the gradient projection method, which is typically used to enforce the bound constraints when using an augmented Lagrangian approach for optimization. Avoiding the gradient projection method is crucial to the procedure we present here because we do not calculate gradients accurately enough to ensure that the gradient projection method will be reliable. The log-transform is a one-to-one mapping of the bounding interval for w_i onto $(-\infty, \infty)$ and can be inverted using

$$w_i = \frac{\exp(u_i) w_i^{\text{up}} + w_i^{\text{low}}}{1 + \exp(u_i)} = \frac{w_i^{\text{up}} + w_i^{\text{low}} \exp(-u_i)}{1 + \exp(-u_i)}. \quad (6)$$

Our optimization procedure is applied in the transformed domain, but at each iteration, we need to invert to the original w-domain in order to run the simulator to obtain the results needed to calculate the NPV functional and to determine if the nonlinear constraints are violated. Throughout, we let N_u denote the dimension of the control vector u, which is a column vector.

3. AUGMENTED LAGRANGIAN FUNCTION

The augmented Lagrangian function has proved to be successful for both nonlinear and linear constraint problems. This method can reduce the possibility of ill conditioning by introducing explicit Lagrange multiplier estimates (Conn et al., 1992). In this study, we apply the log-transformation to handle the bound constraints. Then we incorporate all constraints into an augmented Lagrangian function for the new unbounded control variables (Wang and Spall, 2008). The augmented Lagrangian function β is defined by

$$\begin{aligned}\beta(u, y(u), \mu, \lambda) = \\ J(y,u) - \sum_{j=1}^{n_{\text{e}}} \lambda_{\text{e},j} e_j(u, y(u)) - \frac{1}{2\mu} \sum_{j=1}^{n_{\text{e}}} s_{\text{e},j} e_j^2(u, y(u)) \\ - \sum_{j=1}^{n_{\text{ine}}} \lambda_{\text{c},j} \left[\max\left\{c_j(u, y(u)), -\lambda_{\text{c},j} \frac{\mu}{s_{\text{c},j}}\right\}\right] \\ - \frac{1}{2\mu} \sum_{j=1}^{n_{\text{ine}}} s_{\text{c},j} \left[\max\left\{c_j(u, y(u)), -\lambda_{\text{c},j} \frac{\mu}{s_{\text{c},j}}\right\}\right]^2,\end{aligned} \quad (7)$$

where $\lambda_{\text{e},j}$ and $\lambda_{\text{c},j}$, respectively, denote the Lagrange multipliers associated with the j^{th} equality constraint and the jth inequality constraint; the argument λ of β denotes the vector of all Lagrange multipliers; $s_{\text{e},j}$'s and $s_{\text{c},j}$'s, respectively, denote scaling factors for the equality and inequality constraints and μ is the penalty parameter. The scaling factors, $s_{\text{e},j}$ and $s_{\text{c},j}$, are introduced so that the scaled constraints are of roughly the same magnitudes because poor scaling tends to adversely affect the convergence rate of a maximization algorithm. Following Chen et al. (2010), we set $s_{\text{e},j} = 1/E_j^2$ for $j = 1, \cdots n_{\text{e}}$ and $s_{\text{c},j} = 1/C_j^2$, for $j = 1, \cdots n_{\text{ine}}$, where E_j and C_j, respectively, are the nonzero constraint values that reflect the magnitude of the j^{th} equality constraint and the j^{th} inequality constraint, respectively; see Chen et al. (2010) and the Brugge example considered later for additional discussion. The utility of the augmented Lagrangian function is based on the fact that, under appropriate conditions, a local solution of the constrained optimization problem defined by: maximize the NPV functional $J(y, u)$ subject to the constraints of Eqs. 2 and 3 is a local maximum of the augmented Lagrangian function $\beta(u, y(u), \mu, \lambda)$. Thus, we seek a local maximum of the augment Lagrangian function, and again under appropriate conditions, this maximum will provide a good approximation of the local solution of the constrained optimization. Precise conditions under which we can generate a sequence of $(u_\ell, \mu_\ell, \lambda_\ell)$, which maximize a sequence of augmented Lagrangian functions and have the sequence of u_ℓ converge to a vector of controls that satisfy the constrained optimization problem are somewhat technical and can be found in Conn et al. (1992, 2000). One advantage of the augmented Lagrangian method over the pure-penalty method is that with the augmented Lagrangian method, we do not need to drive the penalty parameter to zero to obtain convergence. Thus, we can avoid the ill-conditioning that can occur when the penalty parameter becomes too small. Conceptually, with the augmented Lagrangian approach, we can decrease the penalty parameter until we are close to a solution of the constrained optimization problem and then, from that point on, keep the value of the penalty parameter fixed and adjust only the Lagrange multipliers from iteration to iteration to attempt to find (u^*, λ^*) which satisfy the first order Karush−Kuhn−Tucker optimality conditions (Conn et al., 2000).

As in the standard implementation of the augmented Lagrangian method, the optimization process involved an inner loop where the the maximization of the augmented Lagrangian is done with fixed λ and μ and an outer loop where, depending on the magnitude of the constraint violation, either all the Lagrange multipliers are modified or the penalty parameter is modified. The standard augmented Lagrangian method is gradient-based, and in the inner loop the augmented Lagrangian function with fixed λ and μ is maximized, subject to the bound constraints, using a gradient projection method. As there are no bound constraints on the transformed control vector u, in our inner loop we simply maximize the augmented Lagrangian directly, i.e., letting λ_ℓ and μ_ℓ denote the values of Lagrange multipliers and penalty parameter at outer loop ℓ at the subsequent inner loop, λ_ℓ and μ_ℓ are held fixed and we maximize $\beta(u, y(u), \mu_\ell \lambda_\ell)$, which is simply denoted by $\beta_\ell(u)$ throughout. The other modification we make to the standard implementation is that we approximate the gradient of the augmented Lagrangian using the modified SPSA introduced by Li and Reynolds (2011). We denote this algorithm by G-SPSA and provide details on its implementation later in the paper. The G-SPSA has the main desirable features similar to those of the SPSA, i.e., the approximate gradient generated with G-SPSA gives an uphill direction for sufficiently small perturbation size, and the expectation of the G-SPSA gradient is equal to a smoothing covariance matrix times the true gradient with a bias in the approximation that goes to zero as the perturbation size goes to zero. We expect the computational efficiency of an augmented Lagrangian technique that uses

any approximate gradient of the augmented Lagrangian function to be at least one to two orders of magnitude less than a comparable implementation that uses an accurate gradient generated with an adjoint method (Brouwer and Jansen, 2004; Kraaijevanger et al., 2007). However, the algorithm presented here can be applied to any augmented Lagrangian function and can easily be coupled with any reservoir simulator as it does not require adjoint code.

In addition to using a log-transform to remove the bounds and using G-SPSA to approximate the gradient of the augmented Lagrangian function, there is one other difference between our implementation and the standard augmented Lagrangian algorithm (Conn et al., 1992, 2000). Specifically, we do not terminate the inner loop based on the magnitude of the gradient of the augmented Lagrangian; instead the following convergence criteria are used:

$$\Delta\beta_{k,\ell} \equiv \frac{|\beta_\ell(u_{k+1}) - \beta_\ell(u_k)|}{\max(|\beta_\ell(u_{k+1})|, 1)} < \epsilon_\ell, \tag{8}$$

and

$$\Delta w_{k,\ell} \equiv \frac{\|w_{k+1} - w_k\|_2}{\max(\|w_{k+1}\|_2, 1)} < \xi_\ell, \tag{9}$$

where k denotes the inner-loop iteration index, ℓ is the outer-loop iteration index, and ϵ_ℓ and ξ_ℓ are the inner loop convergence criteria which decrease as ℓ increases in the standard way.

4. THE SPSA ALGORITHM

For the inner loop maximization of the augmented Lagrangian function $\beta(y(u), u, \mu, \lambda)$, with fixed values of μ and the components of λ, we used the following preconditioned steepest ascent algorithm:

$$u_{k+1} = u_k + a_k \frac{\overline{\widehat{g}}_k}{\|\overline{\widehat{g}}_k\|_\infty}, \tag{10}$$

where k is the iteration index and a_k is the step size. In Eq. 10, $\overline{\widehat{g}}_k$ is a preconditioned steepest-ascent search direction which approximates $C_U \nabla_u \beta(y(u_k), u_k, \mu_\ell, \lambda_\ell)$, where ℓ denotes the outer-loop iteration index, and throughout the inner-loop iteration, λ_ℓ and μ_ℓ are fixed at the values obtained in the last outer-loop iteration. The procedure for calculating $\overline{\widehat{g}}_k$ is discussed next.

We first define a $N_u \times N_u$ covariance matrix for the controls which has the form

$$C_U = \begin{bmatrix} C_{U^1} & 0 & 0 & 0 \\ 0 & C_{U^2} & 0 & 0 \\ . & . & . & . \\ 0 & 0 & 0 & C_{U^{n_w}} \end{bmatrix}, \tag{11}$$

where n_w is the number of wells at which we wish to optimize the well controls and $C_{U^\ell}, \ell = 1, 2, ...n_w$ is the covariance matrix used to force some degree of temporal smoothness on the well controls of well ℓ. The subvector of u that contains all controls for well ℓ is denoted by u^ℓ, and the components of u^ℓ are denoted by u^ℓ_m, $m = 1, 2, \cdots n_j$. We assume that these components are ordered such that u^ℓ_m's correspond to consecutive control steps in time, i.e., u^ℓ_m represents the well control for the mth control interval, and the control for the next control time interval is u^ℓ_{m+1}. Here, we use a spherical covariance function to define the entries of each C_{U^j}, i.e., the entry in the ith row and jth column of C_{U^ℓ} is given by

$$c^\ell_{i,j} = \begin{cases} \sigma^2_\ell \left[1 - \dfrac{3|i-j|}{2N^\ell_s} + \dfrac{1}{2}\left(\dfrac{|i-j|}{N^\ell_s}\right)^3\right], & |i-j| \le N^\ell_s, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Here, i and j refer to the ith and jth control steps, respectively, and N^ℓ_s is the number of control steps over which we wish the control at well ℓ to be correlated. Finally, σ_ℓ is the standard deviation of the control of well ℓ. It is clear that if we generate a sample of the control vector u from a Gaussian distribution with covariance matrix C_U, then the controls for well ℓ will tend to be correlated in time with a correlation length of N^ℓ_s time steps. In our computational algorithm, C_U is used to generate the perturbation for calculating the G-SPSA gradient.

Given u^k in Eq. 10, we compute the search direction by averaging a set of G-SPSA gradients where each G-SPSA gradient used in the average is computed as follows: first we generate a sample Z_k of the normal distribution with covariance C_U and mean equal to the N_u dimensional column vector, i.e., Z_k is a sample of $N(0, C_U)$ (Li and Reynolds, 2011); then we calculate a G-SPSA gradient as

$$\widehat{g}_k = \frac{\beta_\ell(u_k + c_k Z_k) - \beta(u_k)}{c_k} Z_k, \tag{13}$$

where the scalar c_k is the "perturbation size." As shown by Li and Reynolds (2011), for sufficiently small c_k, $\widehat{g}_k$ points in an uphill direction from u_k and the difference between the expectation of $\widehat{g}_k$ and $C_U \nabla \beta_\ell(u_k)$ converges to zero as $c_k \to 0$ provided the second derivatives of $\beta_\ell(u)$ with respect to u are continuous and bounded. Because the expectation of $\widehat{g}_k$ is equal to the true preconditioned gradient, $C_U \nabla \beta_\ell(u_k)$, the "preconditioned steepest-ascent" algorithm of Eq. 10 is both more robust and more computationally efficient if, instead of simply setting the search direction equal to $\widehat{g}_k$, we compute the search direction as an average of a few G-SPSA gradients. Thus, we generate M G-SPSA gradients, $\widehat{g}_{k,j}(u_k)$, $j = 1, 2, \cdots M$, by the procedure introduced above and then calculate

$$\overline{\widehat{g}}_k(u_k) = \frac{1}{M} \sum_{j=1}^{M} \widehat{g}_{k,j}(u_k) \tag{14}$$

as the search direction in Eq. 10. In Eq. 10, we have normalized the average stochastic gradient by dividing by its infinity norm as our experience indicates that it is easier to specify the step sizes a_k when this normalization is done.

Following Spall (2003), the step size a_k and perturbation size c_k are defined by the following equations:

$$a_k = \frac{a}{(k + A + 1)^\alpha}, \tag{15}$$

$$c_k = \frac{c}{(k+1)^\gamma}, \tag{16}$$

where a, A, c, α and γ are positive real numbers which satisfy $A \ge 0, \alpha - 2\gamma > 0$ and $3\gamma - \alpha/2 > 0$. The choice of these parameters can sometimes have a fairly significant effect on the performance of the SPSA algorithm. Our guidelines (Do, 2012) for choosing parameters is given below.

- Set $\alpha = 0.602$ and $\gamma = 0.101$; these choices are based on a theoretically-based recommendation of Spall (2003).

- Set the value of A equal to 10% of the maximum number of iterations allowed, $k_{\max}$.
- a_0 is equal to the maximum change in any component of u that is allowed at the first iteration. In the log-transformed domain, we set $a_0 = 1.5$.
- After choosing A and a_0, a is computed such that at $k = 0$, a satisfies $a_0 = a/(1 + A)^{\alpha}$.
- Specify the minimum allowed perturbation size $c_{\min}$ and find c by solving $c_{\min} = c/(k_{\max} + 1)^{\gamma}$.

Note that Eq. 16 specified how the perturbation sizes used to compute G-SPSA gradients decrease during iteration. It is important that c_k not become so small that all correct significant digits are lost when computing the difference in the numerator of Eq. 13.

5. COMPUTATIONAL RESULTS

The Brugge field is a synthetic reservoir developed by TNO (Peters et al., 2010) as a benchmark study to test different algorithms in the closed-loop reservoir management. The original model was constructed with approximately 20 million gridblocks and then upscaled to a 450,000 gridblock model, which is used as the true reservoir to provide observation data for history matching. The true case was used to construct data such as well logs and facies maps. According to this information, 104 geological realizations were upscaled to a 60,048 gridcell model and provided to participants. The simulation model consists of nine layers, each with 139×48 gridblocks. The total number of active gridblocks is 44,550. Details of the structure and geology of the Brugge field can be found in Peters et al. (2010).

Here, we consider only the production optimization step of the closed-loop reservoir management problem. There are 30 vertical wells in the Brugge reservoir, including 20 smart producing wells and 10 smart water injection wells. Each well has multiple segments that can be controlled individually. We optimize the well controls for years 10 through 30 based on the mean model obtained by Chen et al. (2010) using the ensemble Kalman filter with covariance localization to assimilate production data for the first ten years of the reservoir life. This example is identical to the production optimization problem for years 10-30 solved with an adjoint-gradient method in Chen et al. (2010).

The production period of the reservoir is divided into 40 control steps, i.e., each control step is 182.5 days. There are 84 control variables for each control step. These control variables are the liquid production rate at each individual segment of the production wells and the water injection rate at each individual segment of the injection wells. The total number of control variables is $40 \times 84 = 3360$. The maximum liquid production rate of each producer segment is 477 m^3/D (3000 STB/D), and the maximum injection rate of each injector segment is 636 m^3/D (4000 STB/D). The minimum value for the rate of each segment in a production or an injection well is 0 m^3/D. The minimum bottomhole pressure (BHP) constraint for a producer segment is 4997.32 kpa (725 psi); the maximum BHP constraint for an injection well segment is 17997.25 kpa (2611 psi). The BHP nonlinear constraints are considered reactively by inputting them directly into the simulator data file. In addition to the bound constraints, we have a large number of inequality constraints. The total liquid production rate of the three segments of each production well j must be less than or equal to 477 m^3/D, i.e., $q_{L_j,1} + q_{L_j,2} + q_{L_j,3} \leq 477$ m^3/D, $j = 1, 2, \ldots, 20$. Similarly the water injection rates at three segments of each injector must satisfy $q_{inj_j,1} + q_{inj_j,2} + q_{inj_j,3} \leq 636$ m^3/D, $j = 1, 2, \ldots, 10$. These 1200 linear inequality constraints are incorporated into the augmented Lagrangian function of Eq. (7). In Eq. (1), the oil price is $r_{\mathrm{o}} = \$503.14/\mathrm{m}^3$ ($80.0/STB), and both the water disposal and injection costs are $\$31.45/\mathrm{m}^3$ ($5.0/STB). The annual discount rate is 10%.

The initial value for the injection rate of each injection well segment is 212 m^3/D (1333.3 STB/D), and the initial value for the liquid production rate of each production well segment is 111.3 m^3/D (700 STB/D). We set the scaling factors for inequality constraints $s_{c_i} = 1/C_i^2$. For the total liquid production rate constraints of the producers, $C_i = 477$, $i = 1, 2, \cdots, 800$ and for the total injection rate of the injectors, $C_i = 636$, $i = 801, 802, \cdots, 1200$. The initial Lagrange multipliers are set equal to 0 and the initial penalty parameter is set equal to 10^{-7}. The initial values of the convergence tolerances in Eqs. (8) and (9), respectively, are set equal to $\epsilon_0 \leq 0.005$, $\xi_0 \leq 0.05$, and these values decrease very slowly from iteration to iteration. The algorithm terminates when $\Delta\beta_{k,\ell} \leq 0.002$, $\Delta u_{k,\ell} \leq 0.02$ and the sum of all constraint violations is less than or equal to one percent of the minimum C_i, i.e., when the sum of all constraint violations is less than or equal to 4.76 m^3/D (30 STB/D). The search direction in Eq. 10 is computed from Eq. 14 with $M = 10$. Eq. 12 with $\sigma_\ell^2 = 1.0$ and $N_s^\ell = 40$ is applied for all ℓ to generate the covariance matrix C_U which we use to generate perturbations for calculation of stochastic gradients and for promoting temporal smoothness of the well controls at each individual well. The maximum number of iterations allowed is set equal to the number of controls, i.e., set equal to 3,360. As discussed in the guidelines following Eq. 16, we set $\alpha = 0.602$ and $\gamma = 0.101$, $A = 336$, $a_0 = 1.5$ and $c_{\min} = 0.05$ so that in $a = 50$ and $c = 0.113$ when applying Eqs. 15 and 16.

The G-SPSA-based augmented Lagrangian algorithm converged after 2882 simulation runs using three outer loop iterations. The NPV increased from $\$3.04 \times 10^9$ at the initial guess for optimal well controls to $\$4.25 \times 10^9$ at convergence. Using an augmented Lagrangian approach with gradients computed by the adjoint method, Chen et al. (2010) achieved a realized NPV of $\$4.17 \times 10^9$, which is about 2% lower than our final NPV value. The Chen et al. result, however, only required 30 reservoir simulation runs, two orders of magnitude fewer simulation runs than we required to achieve our result. It is important to realize, however, the preconditioned steepest ascent algorithm does not require adjoint capability, and to the best of our knowledge, no commercial reservoir simulator provides the user the gradient of general nonlinear constraints. The constraints for the Brugge case are linear. The G-SPSA-based augmented Lagrangian method presented here has been applied to problems with general nonlinear constraints, Do (2012). The augmented Lagrangian function and NPV are plotted versus the number of reservoir simulation runs in Fig. 1, and the violations for all the group liquid

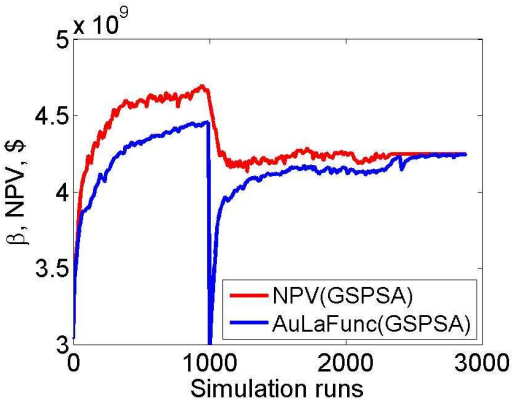

Fig. 1. β and NPV vs, simulation runs.

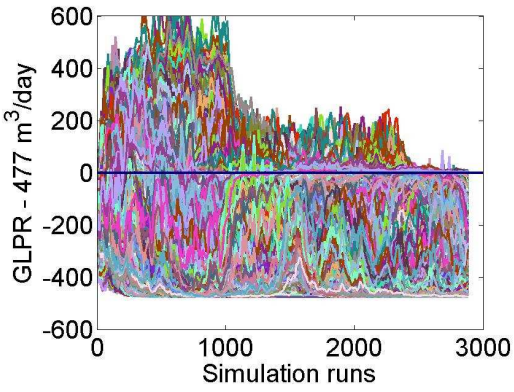

Fig. 2. Violations of the group liquid production rate vs, simulation runs.

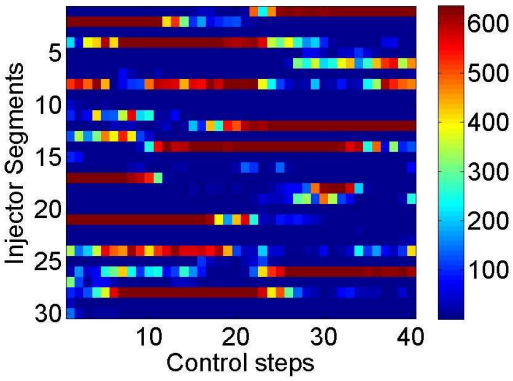

Fig. 3. The estimated optimal injection segment controls.

production rate (GLPR) constraints are shown in Fig. 2, where each individual curve corresponds to a specific well at a specific control step. At convergence, all the GLPR constraints are satisfied within the small tolerance specified. Although not shown, at convergence, the inequality constraints on injectors are also satisfied to within the tolerance specified. The estimated optimal well controls at the thirty injector well segments are shown in Fig. 3. Note that at the end of the reservoir life, most injector well segments operate at or close to the lower bound of 0 STB/D.

REFERENCES

Bangerth, W., Klie, H., Wheeler, M., Stoffa, P., and Sen, M. (2006). On optimization algorithm for the reservoir oil well placement problem. *Computational Geosciences*, 10, 303–319.

Brouwer, D. and Jansen, J. (2004). Dynamic optimization of water flooding with smart wells using optimial control theory. *SPE Journal*, 9(4), 391–402.

Chen, C., Li, G., and Reynolds, A.C. (2010). Closed-loop reservoir management on the Brugge test case. *Computational Geosciences*, 14(4), 691–703.

Conn, A.R., Gould, N., and Toint, P. (2000). *Trust-Region Methods*. SIAM, Philadelphia.

Conn, A., Gould, N., and Toint, P. (1992). *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*. Springer-Verlag, New York.

Do, S. (2012). *Application of SPSA-Type Algorithms to Production Optimization*. Ph.D. thesis, The University of Tulsa, Tulsa, Oklahoma.

Jansen, J., Brouwer, D., Naevdal, G., and van Kruijsdijk, C. (2005). Closed-loop reservoir management. *First Break*, 23, 43–48.

Kraaijevanger, J.F.B.M., Egberts, P.J.P., Valstar, J.R., and Buurman, H.W. (2007). Optimal waterflood design using the adjoint method. In *Proceedings of the SPE Reservoir Simulation Symposium*, SPE 105764, 15.

Li, G. and Reynolds, A.C. (2011). Uncertainty quantification of reservoir performance predictions using a stochastic optimization algorithm. *Computational Geosciences*, 15(3), 451–462.

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

Nævdal, G., Brower, D.R., and Jansen, J.D. (2006). Waterflooding using closed-loop control. *Computational Geosciences*, 10(1), 37–60.

Peters, L., Arts, R., Brouwer, G., Geel, C., Cullick, S., Lorentzen, R., Chen, Y., Dunlop, K., Vossepoel, F., Xu, R., Sarma, P., Alhuthali, A., and Reynolds, A. (2010). Results of the Brugge benchmark study for flooding optimisation and history matching. *SPE Reservoir Evaluation & Engineering*, 13(3), 391–405.

Sarma, P., Chen, W., Durlofsky, L., and Aziz, K. (2008). Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reservoir Evaluation & Engineering*, 11(2), 326–339.

Spall, J.C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, Hoboken, NJ.

van Essen, G., Zandvliet, M., den Hof, P.V., Bosgra, O., and Jansen, J. (2009). Robust waterflooding optimization of multiple geological scenarios. *SPE Journal*, 14(1), 202–210.

Wang, I.J. and Spall, J.C. (2008). Stochastic optimisation with inequality constraints using simultaneous perturbations and penalty functions. *International Journal of Control*, 81(8), 1232–1238.

Zhao, H., Chen, C., Do, S., Li, G., and Reynolds, A. (2011). Maximization of a dynamic quadratic interpolation model for production optimization. In *Proceedings of the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21–23 February*, SPE 141317.