# Adaptation of the CPR Preconditioner for Efficient Solution of the Adjoint Equation

**Choongyong Han,** Chevron Energy Technology Company; **John Wallis,** Wallis Consulting Incorporated; and **Pallav Sarma, Gary Li, Mark L. Schrader,** and **Wen Chen,** Chevron Energy Technology Company

## Summary

It is well known that the adjoint approach is the most efficient approach for gradient calculation, and it can be used with gradient-based optimization techniques to solve various optimization problems, such as the production-optimization problem and the history-matching problem. The adjoint equation to be solved in the approach is a linear equation formed with the "transpose" of the Jacobian matrix from a fully implicit reservoir simulator. For a large and/or complex reservoir model, generalized preconditioners often prove impractical for solving the adjoint equation. Preconditioners specialized for reservoir simulation, such as constrained pressure residual (CPR), exploit properties of the Jacobian matrix to accelerate convergence, so they cannot be applied directly to the adjoint equation. To overcome this challenge, we have developed a new two-stage preconditioner for efficient solution of the adjoint equation by adaptation of the CPR preconditioner (named CPRA: CPR preconditioner for adjoint equation).

The CPRA preconditioner has been coupled with an algebraic multigrid (AMG) linear solver and implemented in Chevron's extended applications reservoir simulator (CHEARS®). The AMG solver is well known for its outstanding capability to solve the pressure equation of complex reservoir models; solving the linear system with the "transpose" of the pressure matrix is one of the two stages of construction of the CPRA preconditioner.

Through test cases, we have confirmed that the CPRA/AMG solver with generalized minimal residual (GMRES) acceleration solves the adjoint equation very efficiently with a reasonable number of linear-solver iterations. Adjoint simulations to calculate the gradients with the CPRA/AMG solver take approximately the same amount of time (at most) as do the corresponding CPR/AMG forward simulations. Accuracy of the solutions has also been confirmed by verifying the gradients against solutions with a direct solver. A production-optimization case study for a real field using the CPRA/AMG solver has further validated its accuracy, efficiency, and the capability to perform long-term optimization for large, complex reservoir models at low computational cost.

## Introduction

Practical production-optimization problems typically involve large, highly complex reservoir models; thousands of unknowns; and many nonlinear constraints, which makes the numerical calculation of gradients for the optimization process impractical. Adjoint models can be used very efficiently for the calculation of gradients (Brouwer and Jansen 2004). Direct coding of the adjoint model is, however, complex and time consuming, and the code is dependent on the forward model in the sense that it must be updated whenever the forward model is modified.

Sarma et al. (2005) proposed a new adjoint-construction procedure for production-optimization problems that makes it relatively easy to construct the adjoint equation with the additional advantage of making the adjoint-simulation code independent of the forward-simulation code. For a fully implicit forward model and specific forms of the cost function and nonlinear constraints used in production optimization, all information necessary for the adjoint run is calculated and stored during the forward simulation itself. The adjoint run then requires only the appropriate assembling of this information to calculate gradients of the cost function with respect to control variables at each control step. During the assembling procedure, the adjoint equation at each timestep should be solved for Lagrange multipliers using information obtained from forward simulation, such as Jacobians. Note that the Jacobian matrix for the adjoint simulation being available from the forward run for a fully implicit model was observed earlier by Li et al. (2003) for application to history-matching problems. However, unlike in the production-optimization problem, gradients of the history-matching error with respect to history-matching variables (e.g., permeability) are not available from the forward problem without significant modifications to the simulator code.

The adjoint equation at each timestep is a linear equation with the transpose of the Jacobian matrix evaluated at the convergence of that timestep. The adjoint equation with the transpose of the Jacobian matrix, unfortunately, cannot be solved by a linear solver using a preconditioner designed and optimized specifically for reservoir simulation, such as CPR (Wallis 1983; Wallis et al. 1985) or the nested-factorization method (Appleyard 1983). This is because the preconditioners exploit properties of the original Jacobian matrix to accelerate convergence.

The two-stage CPR preconditioner developed by Wallis (1983) has proved to be a very powerful preconditioner for reservoir simulation, especially when combined with an AMG linear solver (Cao et al. 2005). The AMG linear solver using the AMG preconditioner has shown its outstanding capability to solve the pressure equation of complex reservoir models in the construction of the CPR preconditioner. Therefore, there arises a strong desire to have a linear solver with the CPR preconditioner using the AMG solver (i.e., CPR/AMG solver) working for the adjoint equation, as well; however, there are other approaches proposed to solve similar adjoint equations outside of the petroleum industry (Schneider and Jimack 2005; Golub et al. 2008). To fulfill this demand, a new two-stage preconditioner has been developed for efficient solution of the adjoint equation by adaptation of the CPR preconditioner (named CPRA: CPR preconditioner for adjoint equation), and it has been coupled with an AMG linear solver using CHEARS®. The AMG solver is used to solve the linear system with the transpose of the pressure matrix, which is one of the two stages of construction of the CPRA preconditioner.

In this paper, mathematical derivation of the CPRA preconditioner will be presented first. Computational performance of a linear solver with the CPRA preconditioner using the AMG solver (i.e., CPRA/AMG solver) will then be shown using three test cases. Finally, a production-optimization case study for a real field using the CPRA/AMG solver will be presented to further validate its accuracy, efficiency, and capability to perform long-term optimization for large, complex reservoir models at low computational cost.

## Mathematical Derivation of the CPRA Preconditioner

**Definition of the Optimization Problem.** The optimization problem discussed in the preceding requires a sequence of control vectors $u^n$ be found for $n = 0, 1, \ldots, N-1$, where $n$ is the control-step index and $N$ is the total number of control steps, to maximize (or minimize) a performance measurement $J(u^0, \ldots, u^{N-1})$ (Sarma et al. 2005). The

This paper (SPE 141300) was accepted for presentation at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21–23 February 2011, and revised for publication. Original manuscript received for review 7 January 2011. Revised manuscript received for review 3 June 2012. Paper peer approved 11 June 2012.

optimization can be described very generally with the following mathematical formulation:

$$\max_{u^n}[J = \phi(x^N) + \sum_{n=0}^{N-1} L^n(x^{n+1}, u^n)] \ \forall n \in (0,..,N-1), \quad \text{(1)}$$

subject to

- $g^n(x^{n+1}, x^n, u^n) = 0, \quad \forall n \in (0,..,N-1).$
- $x^0 = x_0$ (initial condition).
- $c^n(x^{n+1}, u^n) \leq 0, \quad \forall n \in (0,..,N-1).$
- $Au^n \leq b, \quad \forall n \in (0,..,N-1).$
- $LB \leq u^n \leq UB, \quad \forall n \in (0,..,N-1).$

Here, $x^n$ refers to the dynamic states of the system, such as pressures, saturations, and compositions. The cost function $J$ consists of two terms. The first term, $\phi$, is only a function of the dynamic states of the last control step; in a production-optimization application, it could represent an abandonment cost, for example. The second term, which is a summation over all control steps, consists of the kernel $L^n$, known as the Lagrangian in control literature (Stengel 1994). For production optimization, it could include the oil and water rates or some function of the saturations (for sweep efficiency); for history-matching problems, it will represent the history-matching error. Because $L^n$ usually consists of well parameters or quantities that are functions of well parameters, it is written here in a fully implicit form.

The set of equations ($g^n$), together with the initial condition, define the dynamic system, which is basically the reservoir-simulation equations for each gridblock at each timestep. These equations relate the cost function $J$ to the controls $u^n$ through their mutual dependence on the states $x^n$. The last three equations of Eq. 1 refer to the additional constraints for the controls (i.e., nonlinear constraints, linear constraints, and bounds on controls). These will not be discussed in this paper, and interested readers are referred to Sarma et al. (2008) for details of these additional constraints.

**Adjoint Equation.** To solve the preceding optimization problem using gradient-based optimization algorithms, the gradient of $J$ with respect to $u^n$ is required. This can be calculated very efficiently using the adjoint method. The adjoint-model equations are obtained from the necessary conditions of optimality of the optimization problem, defined by Eq. 1. These necessary conditions of optimality are obtained from the classical theory of calculus of variations. For a relatively simple treatment of this subject, refer to Stengel (1994). The adjoint equations to be solved to calculate the gradient of the objective function with respect to the control variables are as follows (Sarma et al. 2005):

$$\lambda^{Tn} = -\left[\frac{\partial L^{n-1}}{\partial x^n} + \lambda^{T(n+1)}\frac{\partial g^n}{\partial x^n}\right]\left(\frac{\partial g^{n-1}}{\partial x^n}\right)^{-1}, \ \forall n \in (1, \cdots, N-1), \quad \text{(2a)}$$

and

$$\lambda^{TN} = -\left(\frac{\partial \Phi}{\partial x^N} + \frac{\partial L^{N-1}}{\partial x^N}\right)\left(\frac{\partial g^{N-1}}{\partial x^N}\right)^{-1}, \ \text{(final condition)}. \quad \text{(2b)}$$

Here, $\lambda$ denotes the Lagrange multipliers and $T$ denotes the transpose. Because $\lambda^n$ depends on $\lambda^{n+1}$, the Lagrange multipliers for the last timestep must be calculated first. It is for this reason that the adjoint equation is solved backward in time, and the adjoint simulation to obtain gradients of the cost function with respect to control variables with the Lagrange multipliers is often called the backward simulation. The required gradients of the cost function with respect to the control variables are given as

$$\frac{\partial J}{\partial u^n} = \frac{\partial J_A}{\partial u^n} = \left[\frac{\partial L^n}{\partial u^n} + \lambda^{T(n+1)}\frac{\partial g^n}{\partial u^n}\right], \ \forall n \in (0, \cdots, N-1). \quad \text{(3)}$$

Here, $J_A$ is the augmented cost function (Sarma et al. 2005). Please note that $\frac{\partial g^{n-1}}{\partial x^n}$ in Eq. 2 is actually the Jacobian matrix ($\overline{\overline{J}}$), evaluated at convergence at timestep $n$ in the forward reservoir simulation. Therefore, Eq. 2 at each timestep can be rewritten in the following simplified form:

$$A^T z = d, \quad \text{(4)}$$

where

$$A = \overline{\overline{J}}, \quad z = \lambda, \quad \text{and}$$

$$d = \begin{cases} -\left[\frac{\partial L^{n-1}}{\partial x^n} + \lambda^{T(n+1)}\frac{\partial g^n}{\partial x^n}\right]^T, \ \forall n \in (1, \cdots, N-1) \\ -\left[\frac{\partial \Phi}{\partial x^N} + \frac{\partial L^{N-1}}{\partial x^N}\right]^T, \ \text{(final condition)}. \end{cases} \quad \text{(5)}$$

Eq. 4 is a linear equation with the transpose of the Jacobian matrix. Because the Jacobian matrix is transposed, the adjoint equation cannot be solved directly by a linear solver using a preconditioner, such as the CPR preconditioner, which exploits properties of the original Jacobian matrix to accelerate convergence. Because the two-stage CPR preconditioner has proved to be a very powerful preconditioner for reservoir simulation, it has been adapted to solve the adjoint equation in this study and has been named the CPRA preconditioner.

**The CPR Preconditioner.** In this subsection, a brief description of the CPR preconditioner is provided, from which the CPRA preconditioner is derived. Let $A$ be the fully implicit reservoir matrix:

$$A = \begin{bmatrix} A_{ss} & A_{sp} \\ A_{ps} & A_{pp} \end{bmatrix}. \quad \text{(6)}$$

Here, subscript $p$ refers to pressure and subscript $s$ to the remaining unknowns.

To solve the linear equation $Ax = b$ for the forward reservoir simulation, the true implicit-pressure/explicit-saturation (Wallis et al. 2009) pressure reduction is performed using the matrix $R$:

$$R = \begin{bmatrix} I & 0 \\ -Colsum(A_{ps})Colsum^{-1}(A_{ss}) & I \end{bmatrix}, \quad \text{(7)}$$

where $Colsum(B)$ ($B$ is any blocked matrix in which the submatrix blocks are all the same size) is defined to be the block diagonal matrix, having as the $i$th main diagonal block the sum of all the blocks in the $i$th block column of $B$. Then,

$$\tilde{A}x = \tilde{b}, \quad \text{(8)}$$

where $\tilde{A} = RA$ and $\tilde{b} = Rb$.

The CPR pressure matrix is

$$A_p = C^T \tilde{A}\, C, \quad \text{(9)}$$

where $C$ is the $(n_{\text{cell}} \times n_{\text{eqn}}) \times n_{\text{cell}}$ pressure-prolongation matrix ($n_{\text{cell}}$ is the number of cells, and $n_{\text{eqn}}$ is the number of equations per cell). The CPR preconditioning matrix for $\tilde{A}$ is given by (Wallis 1983):

$$M_{CPR}^{-1} = \tilde{M}^{-1}(I - \tilde{A}C A_p^{-1} C^T) + C A_p^{-1} C^T, \quad \text{(10)}$$

where $\tilde{M}$ is an approximate factorization for $\tilde{A}$.

**Derivation of the CPRA Preconditioner.** Now the CPR preconditioner is adapted to solve the adjoint equation (Eq. 4), which is the same as

$$(\tilde{A})^T y = d, \quad \text{(11)}$$

where $y = (R^T)^{-1} z$.

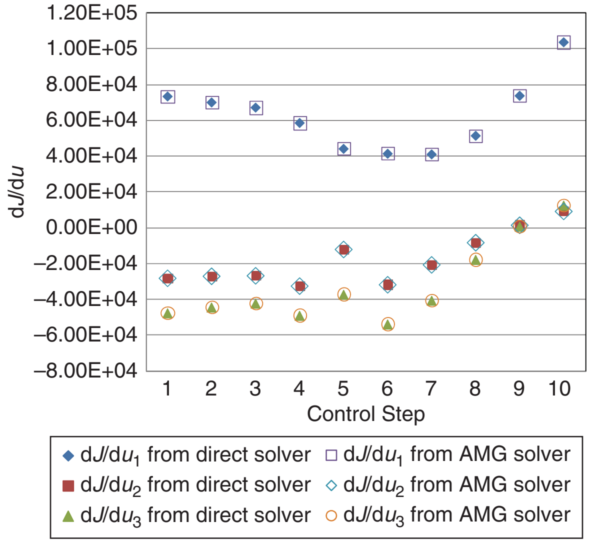

**Fig. 1—Comparison of d*J*/d*u* between the direct solver and the CPRA/AMG solver.**

Here, the CPRA preconditioner $M_{CPRA}^{-1}$ is proposed to solve Eq. 11 as follows:

$$M_{CPRA}^{-1} = M_{CPR}^{-T} = [I - C(A_p^T)^{-1} C^T (\tilde{A})^T](\hat{M})^{-1} + C(A_p^T)^{-1} C^T, \quad \ldots\ldots\ldots\ldots (12)$$

where $\hat{M}$ is an approximate factorization of $(\tilde{A})^T$. Because the CPRA preconditioner needs to be applied to a transpose of $\tilde{A}$, it is proposed that it is obtained by transposing the CPR preconditioner. By rearranging Eq. 12, we obtain

$$M_{CPRA}^{-1} = C(A_p^T)^{-1} C^T [I - (\tilde{A})^T (\hat{M})^{-1}] + (\hat{M})^{-1}. \quad \ldots (13)$$

After analyzing Eq. 13, the following noble two-stage preconditioning algorithm for the residual $r[= d - (\tilde{A})^T y]$ has been developed to construct the CPRA preconditioner for $(\tilde{A})^T$:

1. Solve $\hat{M} x_f = r$ for $x_f$. (First stage: Apply incomplete LU [ILU($k$)] to $(\tilde{A})^T$ to solve the full system.)
2. $r_m = r - (\tilde{A})^T x_f$, (modify the residual).
3. $r_p = C^T r_m$, (shrink the residual).
4. Solve $(A_p)^T x_p = r_p$ for $x_p$. (Second stage: Solve the transpose of the pressure system using the AMG preconditioner.)
5. $r_t = C x_p + x_f$, (obtain the true residual).

Steps 1 through 5 are repeated using the GMRES algorithm (Saad and Schultz 1986) until converged solution $y$ is obtained. With the converged solution $y$, the solution $z$ to Eq. 4 can be obtained by $z = R^T y$.

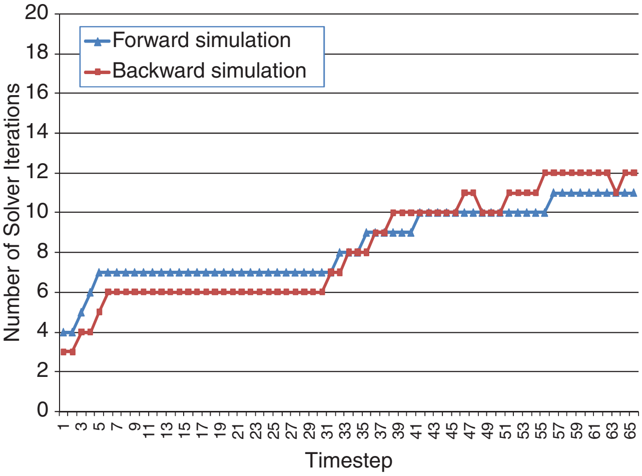

**Fig. 2—Number of linear-solver iterations, with timestep, in the forward and backward simulations for Case 1.**

## Validation and Performance of the CPRA/AMG Solver

This new CPRA algorithm has been implemented with the AMG solver using CHEARS$^{\text{TM}}$. The following three cases have been tested to confirm the performance of the new algorithm:

- Case 1—Synthetic case, 10×10×3 (= 300) grids with 264 active grids, BLACKOIL option, two water injectors and one producer
- Case 2—Field P case, 86×100×8 (= 68,800) grids with 39,265 active grids, BLACKOIL option, five water injectors and five producers
- Case 3—Field L case, 96×73×150 (= 1,051,200) grids with 164,281 active grids, BLACKOIL option, 11 water injectors and 15 producers

While testing the performance of the new algorithm, it has also been validated that the gradient of the objective function with respect to control variable $i$ (= d$J$/d$u_i$) from the CPRA/AMG solver is exactly the same as that from the direct solver **(Fig. 1).** Case 1 was used for the validation because the adjoint equation of the case is relatively easy for the direct solver to solve because of its smaller linear-system size.

One way to check the performance of a preconditioner is to check the number of iterations of a linear solver using the preconditioner to solve a linear equation. **Figs. 2 through 4** show the number of iterations required by the CPR/AMG solver to solve

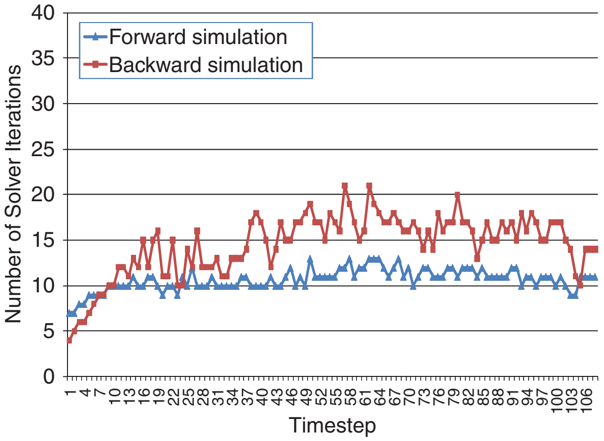

**Fig. 3—Number of linear-solver iterations, with timestep, in the forward and backward simulations for Case 2.**

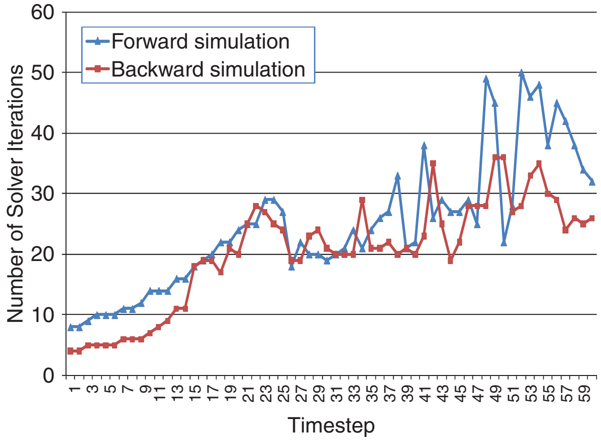

**Fig. 4—Number of linear-solver iterations, with timestep, in the forward and backward simulations for Case 3.**

$\tilde{A}x = \tilde{b}$ at each timestep in the forward simulation for each case. The number of iterations required by the CPRA/AMG solver to solve $(\tilde{A})^T y = d$ at each timestep in the backward simulation is also shown in the figures. Here, $\tilde{A}$ is the Jacobian matrix evaluated at the convergence of the Newtonian iteration at a timestep. The same tolerance parameter values are used to solve both $\tilde{A}x = \tilde{b}$ and $(\tilde{A})^T y = d$. Ideally, with the same tolerance, the number of iterations required by the CPRA/AMG solver in the backward simulation should not be much different from that of the CPR/AMG solver in the forward simulation. This is because the eigenvalues of the CPRA preconditioned matrix are identical to the eigenvalues of the CPR preconditioned matrix. The asymptotic convergence rate should be the same regardless of right-hand-side values, and if CPR is working well for a matrix, CPRA should be working equally as well for the transpose of the matrix. This can be well confirmed in Figs. 2 through 4.

Please note that in Fig. 2, the number of iterations required by the CPR/AMG solver in the forward simulation seems quite large for the small size of the problem, which has 10×10×3 grids. This is because tight tolerance for the backward simulation was used for the forward simulation for fair comparison. A very small maximum relative residual of $10^{-9}$ was used as the tolerance to determine convergence of both CPR/AMG and CPRA/AMG solvers for all three test cases. During this study, it was found that numerical solution to the adjoint equation is highly dependent on tolerance, and tight tolerance should be used to obtain reasonable solution. On the basis of our experience, it is recommended that a maximum relative residual equal to or less than $10^{-9}$ be used to solve the adjoint equation.

For the purpose of comparison, an ILU preconditioner with GMRES algorithm was also attempted for the backward simulations. For the first test case, the solver provided the exact same d$J$/d$u_i$ as the direct solver when the infill-level value was increased with tightened tolerance. However, this required very high computational cost without the benefit of using the preconditioner. For the second and third test cases, the solver gave incorrect d$J$/d$u_i$ because of convergence failure, even with relatively large infill values requiring significant run time. Because the CPRA/AMG solver was free from these problems for the test cases, it was tried on a field-production-optimization problem next, which is discussed in the following.

## Application to a Production-Optimization Problem

The gradients obtained from CHEARS® are used with a sequential-quadratic-programming (SQP) algorithm (Rao 2009) to solve the production-optimization problem described by Eq. 1. The main reason for including such an optimization example is that the ultimate goal of developing CPRA is to solve such optimization problems efficiently. The simulation model used is that of a real reservoir and is chosen because it is fairly large, such that direct and other simple solvers using conventional preconditioners cannot be used for this problem because of extremely high computational cost. In fact, the direct solver failed on this model because of memory requirements, and the development of CPRA was indeed motivated by this example. As such, this is a good example to demonstrate the benefit of CPRA. This model was also used by Chaudhri et al. (2009) to test ensemble-based optimization, so these results allow us to compare adjoint-based optimization with ensemble-based optimization. For details of ensemble-based optimization, readers are referred to Chen et al. (2009).

The reservoir field considered for this study is an offshore Angola field, which is also used for Case 3 in the validation and performance of the CPRA/AMG solver in the preceding section. The average depth of the reservoir is 7,000 ft, and the initial pressure is approximately 3,300 psia at a datum depth of 7,250 ft. The bubblepoint pressure of the reservoir is assumed to be approximately 2,195 psia. Thus, the reservoir is initially undersaturated and has no gas cap. The reservoir temperature is 130°F. The field has an average porosity of approximately 31% and average permeability of 900 md. The oil/water contact is located at a depth of 7,973 ft. The simulation model has grid dimensions of 96×73×150. The size of a gridblock in the $x$-direction is approximately 245 ft and varies between 190 ft and 335 ft in the $y$-direction. The thickness of an individual layer shows a wide variation between 0.1 and 275 ft. There are 15 producers and 11 injectors in the simulation model. The completion interval is different for different wells. The reservoir properties (e.g., porosity and permeability) are assumed to be known without any uncertainty.

Total fluid-production (reservoir volume) rate is the primary constraint on the producers, whereas the injectors are constrained by the total water-injection (reservoir volume) rate. The producers and the injectors are also constrained by the bottomhole-pressure (BHP) limits. The producers are constrained by a minimum BHP of 2,240 psia, whereas the injectors are constrained by a maximum BHP of 4,700 psia. Additionally, the producers and injectors have group constraints of total fluid-production rate and total water-injection rate, respectively. The BHP constraints are simple bound constraints and are handled directly by the optimizer. The other constraints are nonlinear constraints and are handled by the simulator itself (Sarma et al. 2008). For this field, we assume that the water injection is started from Day 1 and is continued throughout the life of the waterflooding project (3,960 days). The objective of this study is to optimize the net present value (NPV) of the reservoir over the project life. The control variables are the BHPs, and the control-step size for both producers and injectors is considered to be 180 days, which results in 22 control steps per well for the optimization problem. Therefore, for a total of 26 wells (15 producers and 11 injectors), we need to optimize a set of 572 control variables. Some of the other parameters used in the optimization procedure include oil price of USD 70/bbl, water-disposal cost of USD 10/bbl, discount rate of 0% per year, and an ensemble of 40 realizations of control variables.

To understand the benefit of any optimization process, it is usual to compare the optimization results with a base or reference case. In the case of production optimization, such a base case would be a reasonable production strategy that an engineer would devise given a simulation model and a set of constraints. In general, although an engineer's intuition and knowledge about a reservoir plays an important role in determining optimum or near-optimum well locations, it is very difficult to understand the implications of varying well controls, such as rates, manually on the optimization process. It is thus usual for engineers to specify constant production/injection rates or BHPs until some detrimental reservoir response, such as breakthrough, is observed.

For the purpose of this case study, the base (or reference) case is a constant-production-/injection-rate strategy (unless other constraints are hit). The group constraint of total fluid-production rate is distributed equally among all producers. Therefore, in the base case, all the producers are allowed to produce at an equal production rate throughout the waterflooding project. In a similar manner, the group constraint of total water-injection rate is distributed equally among all injectors, allowing each injector to inject water at an equal rate.

**Fig. 5** shows the increase in NPV (normalized with respect to the maximum NPV obtained), with iterations for SQP/adjoint with CPRA/AMG solver (green line), and also shows the NPV from two ensemble-based approaches demonstrated in Chaudhri et al. (2009). The red line is from the original ensemble-based optimization (EnOpt), as described in Chen et al. (2009), and the blue line is from the conjugate gradient-based-ensemble optimization (CGEnOpt), as described in Chaudhri et al. (2009). All methods lead to a significant increase in NPV (approximately 17% more than the base case). Further, we observe that the SQP/adjoint reaches close to the maximum NPV after three iterations, which requires eight simulations. Both ensemble-based methods require on the order of 600 simulations to reach the same level of NPV. This clearly proves the accuracy, efficiency, and the capability of the CPRA/AMG solver to perform long-term optimization for large, complex reservoir models at low computational cost. This also demonstrates the benefit of adjoint-based optimization over ensemble-based optimization for large-scale models, for which hundreds of simulations are currently not practically possible. Please note that a direct

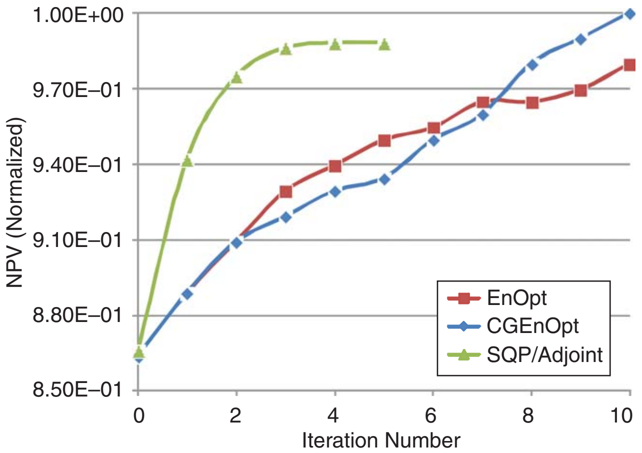

**Fig. 5—NPV vs. iteration number for ensemble-based optimization (from Chaudhri et al. 2009) and SQP/adjoint.**

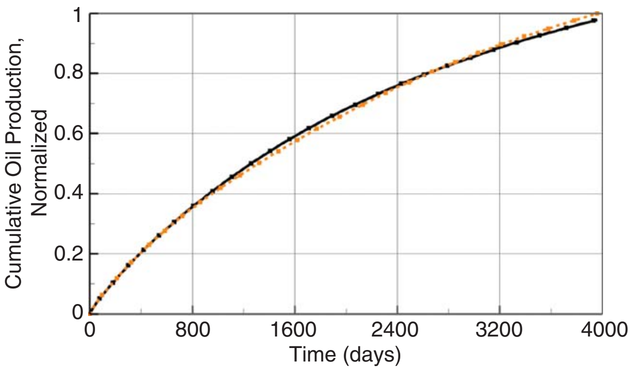

**Fig. 6—Cumulative oil production for the base case (black) and the optimized case (orange).**

solver available in CHEARS® was not able to solve the linear system of this problem, so more-efficient reservoir-simulation-specific solvers are required for such problems.

In terms of computation time for this problem, *X* minutes was required for each forward simulation (here, *X* minutes includes time to write output required for the adjoint simulation) and 0.35*X* minutes for the adjoint simulation at a tolerance of $10^{-9}$. For the ensemble optimization, a lower tolerance can be used because an adjoint is not solved and, using a tolerance of $10^{-6}$, 0.63*X* minutes was required for each forward simulation.

**Fig. 6** shows cumulative oil production (normalized) for base (black) and optimized (orange) cases. We observe that overall, at the end of the optimization period, there is only a slight improvement in cumulative oil production. The significant increase in NPV thus is a result of a significant decrease in water injection and production (normalized), as seen in **Fig. 7.** We see that for approximately 500 days in the beginning, the water-injection and -production rates for the optimized case are similar to those of the base case, and after that, both injection and production decrease significantly, leading to the increase in NPV.

**Fig. 8** shows BHP profiles for some of the injectors and producers for the base (blue) and optimized (orange) cases. Fig. 8a shows the BHP profiles typical of most producers. Although it is difficult to determine the reason behind the nature of the optimized profiles because of interaction with various constraints, it seems that the optimization is trying to open up the producers slowly over time, until minimum BHP is reached. On the other hand, some wells that produce a large amount of water are shut down, as shown in Fig. 8b. For most injectors, as seen in the Fig. 8c, the

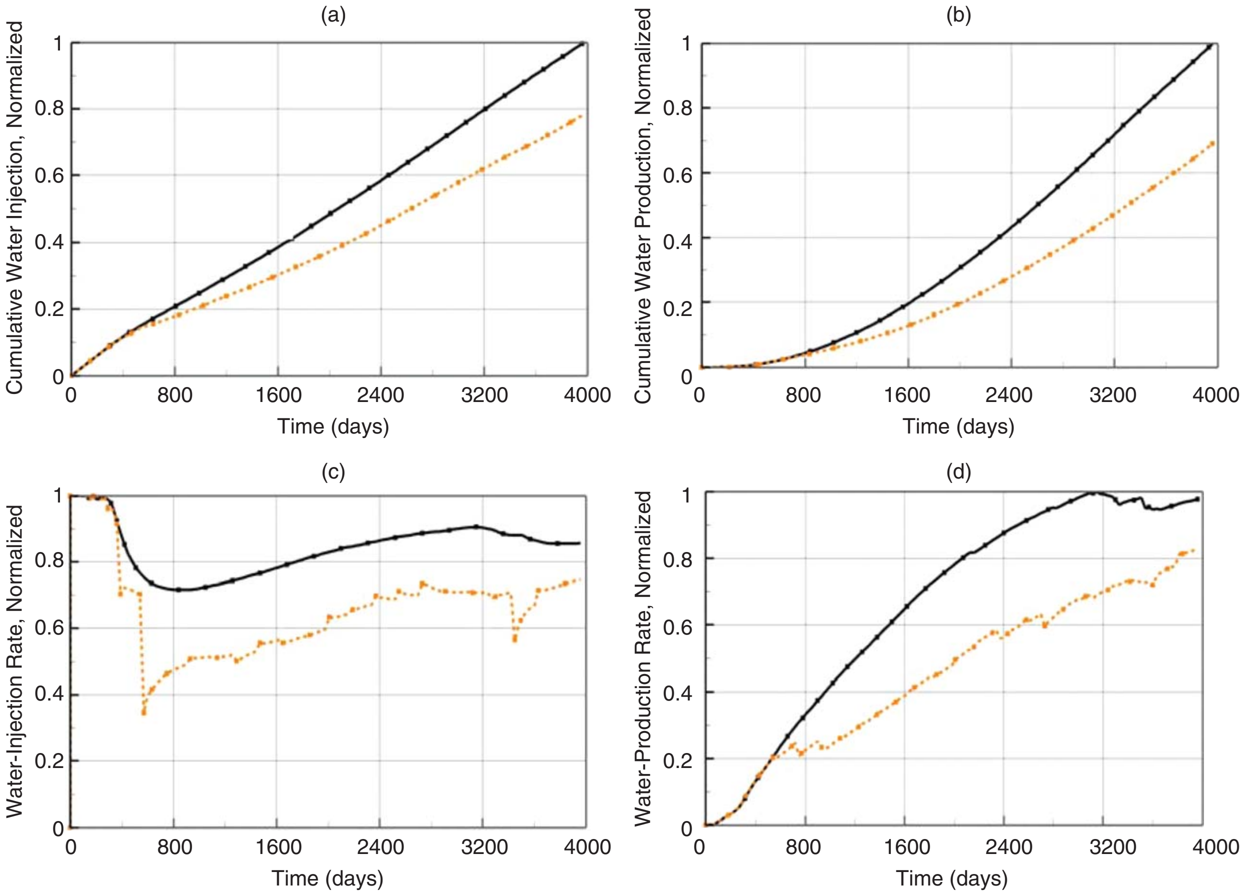

**Fig. 7—Cumulative water-injection and -production rates [(a) and (b), respectively] and water-injection and -production rates [(c) and (d), respectively] for the base case (black) and the optimized case (orange).**

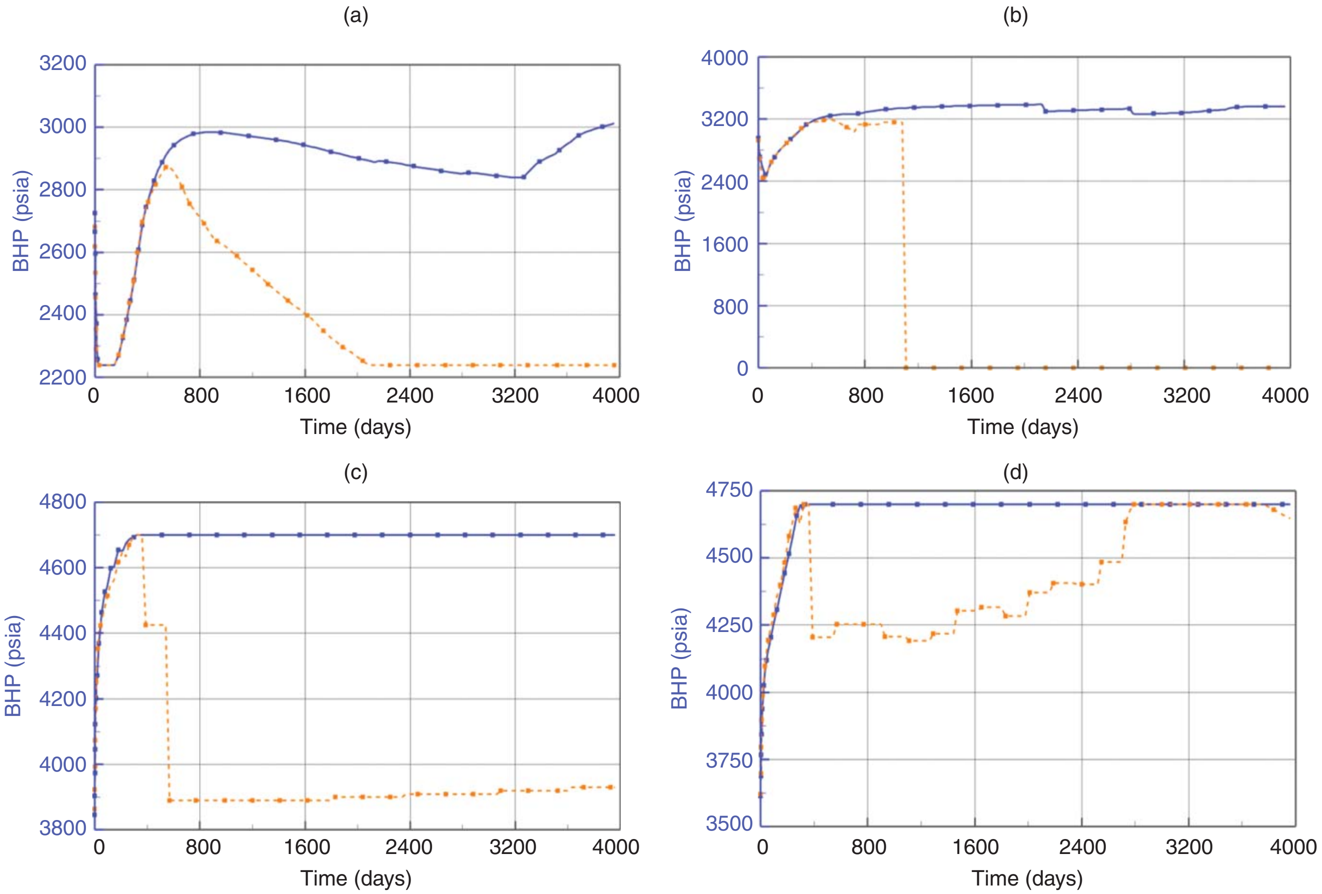

**Fig. 8—(a) Shows the typical BHP profiles for most producers for the base case (black) and the optimized case (orange), (b) shows that some producers are shut down after optimization, (c) shows the typical injector BHP profiles for most injectors, and (d) shows that some injectors continue to inject in late time.**

BHPs hit the lower boundary after approximately 500 days, which is in agreement with the reduction in total injection. Some injectors do, however, continue injecting at the upper boundary of the BHP throughout the optimization period, as seen in Fig. 8d.

## Conclusion

A new two-stage preconditioner, named the CPRA preconditioner, has been developed to solve the adjoint equation with the transpose of the Jacobian matrix by adaptation of the CPR preconditioner and has been implemented in Chevron's in-house reservoir simulator, CHEARS®. The AMG solver is used to solve the linear system with the transpose of the pressure matrix, which is one of the two stages of construction of the CPRA preconditioner. Through test cases, it has been confirmed that the CPRA/AMG solver with GMRES acceleration solves the adjoint equation very efficiently, with a reasonable number of linear-solver iterations. Adjoint simulations to calculate the gradients with the CPRA/AMG solver take approximately the same amount of time (at most) as the corresponding CPR/AMG forward simulations. Accuracy of the solutions has also been confirmed by verifying the gradients against solutions with a direct solver. A production-optimization case study for a real field using the CPRA/AMG solver has validated further the accuracy, efficiency, and capability of the CPRA/AMG solver to perform long-term optimization for large, complex reservoir models at low computational cost. The new algorithm to create the CPRA preconditioner and the coupling of it with the AMG solver for efficient solution of the adjoint equation are now patent pending.

## Nomenclature

$c$ = nonlinear-constraint equation
$C$ = pressure-prolongation matrix
$g$ = reservoir-simulation equation
$J$ = cost function
$J_A$ = augmented cost function
$\bar{\bar{\bar{J}}}$ = Jacobian matrix
$L$ = Lagrangian
$M_{CPR}^{-1}$ = CPR preconditioning matrix for Matrix $\tilde{A}$
$M_{CPRA}^{-1}$ = CPRA preconditioning matrix for Matrix $(\tilde{A})^T$
$\tilde{M}$ = approximate factorization of Matrix $\tilde{A}$
$\hat{M}$ = approximate factorization of Matrix $(\tilde{A})^T$
$n$ = control-step index
$n_{\text{cell}}$ = number of cells
$n_{\text{eqn}}$ = number of equations per cell
$N$ = total number of control steps
$r$ = residual vector
$R$ = matrix used for pressure reduction
$u$ = control vector
$x$ = dynamic states of the system
$\lambda$ = Lagrange multiplier
$\phi$ = function of the dynamic states of the last control step

### Superscript

$T$ = transpose

### Subscripts

$p$ = pressure
$s$ = unknowns, excluding pressure

## Acknowledgments

The authors would like to thank Chevron for permission to publish this work.

## References

Appleyard, J.R. 1983. Nested Factorization. Presented at the SPE Reservoir Simulation Symposium, San Francisco, California, USA, 15–18 November. SPE-12264-MS. http://dx.doi.org/10.2118/12264-MS.

Brouwer, D.R. and Jansen, J.-D. 2004. Dynamic Optmization of Water Flooding With Smart Wells Using Optimal Control Theory. *SPE J*. **9** (4): 391–402. SPE-78278-PA. http://dx.doi.org/10.2118/78278-PA.

Cao, H., Tchelepi, H.A., Wallis, J. et al. 2005. Parallel Scalable Unstructured CPR-Type Linear Solver for Reservoir Simulation. Presented at the SPE Annual Technical Conference and Exhibition, Dallas, 9–12 October. SPE-96809-MS. http://dx.doi.org/10.2118/96809-MS.

Chaudhri, M.M., Phale, H.A., and Liu, N. 2009. An Improved Approach For Ensemble Based Production Optimization: Application To A Field Case. Presented at the EUROPEC/EAGE Conference and Exhibition, Amsterdam, 8–11 June. SPE-121307-MS. http://dx.doi.org/10.2118/121307-MS.

Chen, Y., Oliver, D.S., and Zhang, D. 2009. Efficient Ensemble-Based Closed-Loop Production Optimization. *SPE J*. **14** (4): 634–645. SPE-112873-PA. http://dx.doi.org/10.2118/112873-PA.

Golub, G.H., Stoll, M., and Wathen, A. 2008. Approximation of the Scattering Amplitude and Linear Systems. *Electron. Trans. Numer. Anal.* **31** (2008): 178–203.

Li, R., Reynolds, A.C., and Oliver, D.S. 2003. History Matching of Three-Phase Flow Production Data. *SPE J*. **8** (4): 328–340. SPE-87336-PA. http://dx.doi.org/10.2118/87336-PA.

Rao, S.S. 2009. *Engineering Optimization: Theory and Practice*, fourth edition. New York: John Wiley & Sons.

Saad, Y. and Schultz, M.H. 1986. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. and Stat. Comput*. **7** (3): 856–869. http://dx.doi.org/10.1137/0907058.

Sarma, P., Aziz, K., and Durlofsky, L.J. 2005. Implementation of Adjoint Solution for Optimal Control of Smart Wells. Presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 31 January–2 February. SPE-92864-MS. http://dx.doi.org/10.2118/92864-MS.

Sarma, P., Chen, W.H., Durlofsky, L.J. et al. 2008. Production Optimization With Adjoint Models Under Nonlinear Control-State Path Inequality Constraints. *SPE Res Eval & Eng* **11** (2): 326–339. SPE-99959-PA. http://dx.doi.org/10.2118/99959-PA.

Schneider, R. and Jimack, P.K. 2005. Efficient preconditioning of the discrete adjoint equations for the incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids* **47** (10-11): 1277–1283. http://dx.doi.org/10.1002/fld.874.

Stengel, R.F. 1994. *Optimal Control and Estimation*. Mineola, New York: Dover Books on Advanced Mathematics, Dover Publications.

Wallis, J., Tchelepi, H.A., and Cao, H. 2009. Apparatus, method and system for improved reservoir simulation using a multiplicative overlapping Schwarz preconditioning for adaptive implicit linear systems. US Patent No. 7,516,056.

Wallis, J.R. 1983. Incomplete Gaussian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration. Presented at the SPE Reservoir Simulation Symposium, San Francisco, California, USA, 15–18 November. SPE-12265-MS. http://dx.doi.org/10.2118/12265-MS.

Wallis, J.R., Kendall, R.P., and Little, T.E. 1985. Constrained Residual Acceleration of Conjugate Residual Methods. Presented at the SPE Reservoir Simulation Symposium, Dallas, 10–13 February. SPE-13536-MS. http://dx.doi.org/10.2118/13536-MS.

**Choongyong Han** is a reservoir simulation development engineer at the Chevron Energy Technology Company. e-mail: cyhan@chevron.com. His main interests in reservoir engineering are reservoir simulation, enhanced oil recovery, and reservoir description. Han used to work as a research associate in the Center for Petroleum and Geosystems Engineering at the University of Texas at Austin. Han holds a PhD degree from Seoul National University. He is a member of the SPE Editorial Review Committee.

**John Wallis** is president of Wallis Consulting Company.

**Pallav Sarma** currently works for Chevron Energy Technology Company as a senior research scientist on the Reservoir Simulation Research team, where he has been since 2006. Previously, Sarma worked as a reservoir engineer for Schlumberger. His research interests include reservoir simulation and optimization, optimal-control theory, statistical-pattern recognition, artificial intelligence, probability theory, stochastic processes, and geostatistics. Sarma holds a bachelor's degree in petroleum engineering from the Indian School of Mines and a PhD degree in petroleum engineering, with a minor in management science and engineering, from Stanford University. He has received several scholarships and awards, including the Dantzig award, the Miller and Ramey Fellowships at Stanford, an SPE scholarship, and a SIAM award for excellence in research.

**Gary Li** is a senior research consultant for the Reservoir Simulation Development team at Chevron Energy Technology Company in Houston. Li holds a PhD degree in mechanical engineering from the University of California, Berkeley.

**Mark L. Schrader** is a team leader for the Reservoir Simulation Development team at Chevron Energy Technology Company in Houston.

**Wen Chen** is a Chevron fellow at Chevron Energy Technology Company in San Ramon, California, USA. His research interests include parallel reservoir simulation, numerical methods, optimization and control, automatic history matching, and thermal-recovery processes. Chen holds a PhD degree in chemical engineering from the California Institute of Technology. In 2007, he was honored as an SPE Distinguished Member.