Improving the Ensemble-Optimization Method Through Covariance-Matrix Adaptation

Journal:	*SPE Journal*
Manuscript ID:	SJ-0514-0042.R1
Manuscript Type:	Technical Paper
Date Submitted by the Author:	18-Aug-2014
Complete List of Authors:	Fonseca, Rahul; Delft University of Technology, Geoscience and Engineering Leeuwenburgh, Olwijn; TNO, Petroleum Geosciences Van den Hof, Paul; Eindhoven University of Technology, Electrical Engineering Jansen, Jan-Dirk; Delft University of Technology, Geoscience and Engineering
Keywords:	water flooding optimization, recovery optimization, ensemble optimization, covariance matrix adaptation, Brugge model

Improving the Ensemble-Optimization Method Through Covariance-Matrix Adaptation

R.M. Fonseca, Delft University of Technology; **O. Leeuwenburgh,** TNO; **P.M.J. Van den Hof,** Eindhoven University of Technology; and **J.D. Jansen,** Delft University of Technology

Summary

Ensemble optimization (referred to throughout the remainder of the paper as EnOpt) is a rapidly emerging method for reservoir-model-based production optimization. EnOpt uses an ensemble of controls to approximate the gradient of the objective function with respect to the controls. Current implementations of EnOpt use a Gaussian ensemble of control perturbations with a constant covariance matrix, and thus a constant perturbation size, during the entire optimization process. The covariance-matrix-adaptation evolutionary strategy is a gradient-free optimization method developed in the "machine learning" community, which also uses an ensemble of controls, but with a covariance matrix that is continually updated during the optimization process. It was shown to be an efficient method for several difficult but small-dimensional optimization problems and was recently applied in the petroleum industry for well location and production optimization. In this study, we investigate the scope to improve the computational efficiency of EnOpt through the use of covariance-matrix adaptation (referred to throughout the remainder of the paper as CMA-EnOpt). The resulting method is applied to the waterflooding optimization of a small multilayer test model and a modified version of the Brugge benchmark model. The controls used are inflow-control-valve settings at predefined time intervals for injectors and producers with undiscounted net present value as the objective function. We compare EnOpt and CMA-EnOpt starting from identical covariance matrices. For the small model, we achieve only slightly higher (0.7 to 1.8%) objective-function values and modest speedups with CMA-EnOpt compared with EnOpt. Significantly higher objective-function values (10%) are obtained for the modified Brugge model. The possibility to adapt the covariance matrix, and thus the perturbation size, during the optimization allows for the use of relatively large perturbations initially, for fast exploration of the control space, and small perturbations later, for more-precise gradients near the optimum. Moreover, the results demonstrate that a major benefit of CMA-EnOpt is its robustness with respect to the initial choice of the covariance matrix. A poor choice of the initial matrix can be detrimental to EnOpt, whereas the CMA-EnOpt performance is near-independent of the initial choice and produces higher objective-function values at no additional computational cost.

Introduction

Several studies have shown that there is considerable scope to improve the economic-life-cycle performance of oil fields through the use of formal optimization methods in conjunction with reservoir-simulation models. A very efficient way to perform such model-based life-cycle optimization is with the aid of gradient-based methods by which the gradient is obtained through an adjoint technique. For an overview of this approach and a large number of references, we refer to the review paper by Jansen (2011). The adjoint method is computationally very efficient, but, unfortunately, it is an intrusive method, requiring access to the simulator source code as well as extensive implementation efforts. Because it is practically impossible to access commercial simulator source codes for implementation of the adjoint, there is a need for alternative methods for model-based production optimization in which the simulator is treated as a black box.

One such method, EnOpt, was shown to achieve good results for a variety of different reservoir models. In EnOpt, the gradient of the objective function with respect to the vector of control variables is approximated by first evaluating the objective-function values for an ensemble of control vectors, chosen from a multi-Gaussian random distribution with a constant prescribed covariance matrix, and then estimating the optimal regression model with a least-squares approach. Predecessors to the EnOpt method were proposed by Lorentzen et al. (2006) and Nwaozo (2006), whereafter Chen (2008) and Chen et al. (2009) gave systematic descriptions of the method, as mostly used today. Thereafter, several publications addressed applications and computational aspects of the method; see, for example, Chaudhri et al. (2009), Chen and Oliver (2010), Su and Oliver (2010), Leeuwenburgh et al. (2010), and Chen and Oliver (2012). In a recent paper, Do and Reynolds (2013) demonstrate that EnOpt can be interpreted as a member of a broader class of approximate-gradient methods that also includes the simultaneous-perturbation stochastic approximation method. As an alternative to exact or approximate gradient-based optimization methods, one can revert to gradient-free methods such as genetic algorithms or evolutionary strategies, as developed in the "machine-learning" community. One of the latter, called the Covariance matrix adapted-evolutionary strategy (CMA-ES), which was developed by Hansen and coworkers (Hansen and Ostermeier 1996, 2001; Hansen 2006), was recently used for well-placement optimization by Ding (2008) and Bouzarkouna et al. (2011), in a flooding-optimization problem by Schulze-Riegert et al. (2011), and for a smart-well-optimization problem by Pajonk et al. (2011). The main idea in CMA-ES is to systematically adapt the variance of the control-vector sample in directions that have proved to be successful. In this paper, we propose an improvement to EnOpt in which the use of a constant covariance matrix throughout the optimization is replaced with a covariance matrix that is constantly adapted with the CMA-ES logic. We will refer to the resulting hybrid scheme as CMA-EnOpt. CMA-EnOpt combines the advantage of explicitly using gradient information to achieve faster convergence (EnOpt) with the continuous adaptation of the covariance matrix to improve the gradient estimate by means of improved sampling with "local" knowledge of the nature of the objective-function search space. In the remainder of this paper, we will first provide an introduction to CMA-EnOpt, followed by its application to a small synthetic 3D reservoir model and to a modified version of the Brugge benchmark model. A comparison of the results with those obtained by EnOpt will illustrate the advantages of CMA-EnOpt for relatively large-scale model-based production optimization.

Theory

In this section, we give a brief overview of the theoretical basis of CMA-EnOpt. We first define our objective function followed by an overview of EnOpt and the proposed modification. We

This paper (SPE 163657) was accepted for presentation at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18–20 February 2013, and revised for publication. Original manuscript received for review 7 January 2014. Revised manuscript received for review 19 August 2014. Paper peer approved 9 September 2014.

apply the usual expression for net present value as objective function J:

$$J=\sum_{k=1}^{K}\left(\frac{\{[(q_{o,k})\cdot r_o-(q_{wp,k})\cdot r_{wp}]-[(q_{wi,k})\cdot r_{wi}]\}\cdot\Delta t_k}{(1+b)^{t_k/\tau_t}}\right), \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

where $q_{o,k}$ is the oil-production rate in B/D, $q_{wp,k}$ is the water-production rate in B/D, $q_{wi,k}$ is the water-injection rate in B/D, r_o is the price of oil produced in USD/bbl, r_{wp} is the cost of water produced in USD/bbl, r_{wi} is the cost of water injected in USD/bbl, Δt_k is the difference between consecutive timesteps in days, b is the discount factor expressed as a fraction per year, t_k is the cumulative time in days corresponding to timestep k, and τ_t is the reference time period for discounting, typically 1 year.

EnOpt

EnOpt uses an ensemble of control vectors to approximate the gradient of the objective function J with respect to the (average) control vector. A single control vector is defined as

$$\mathbf{u}=(u_1 \quad u_2 \quad \cdots \quad u_N)^T, \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

where N is the number of control variables (e.g., bottomhole pressures, well rates, or valve settings) and the superscript T indicates the transpose. Thus, $\mathbf{u}$ is a vector with a number of elements N that typically equals the number of control timesteps times the number of control variables per timestep. In EnOpt, a multivariate, Gaussian-distributed ensemble ($\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$) is generated with a distribution mean $\tilde{\mathbf{u}}$ and a predefined distribution covariance matrix $\tilde{\mathbf{C}}$, where M is the ensemble size. (In this paper, matrices are indicated with bold capitals, vectors with bold lowercase letters, and scalars with italics.) In the remainder of this paper, we will use $\mathbf{u}$ instead of $\tilde{\mathbf{u}}$ to simplify the notation. During the iterative optimization process, $\mathbf{u}$ is updated until convergence, whereas $\tilde{\mathbf{C}}$ is kept fixed. To estimate the gradient, a mean-shifted ensemble matrix is defined as

$$\mathbf{U}=[\mathbf{u}_1-\overline{\mathbf{u}} \quad \mathbf{u}_2-\overline{\mathbf{u}} \quad \cdots \quad \mathbf{u}_M-\overline{\mathbf{u}}], \quad \ldots\ldots\ldots\ldots (3)$$

where

$$\overline{\mathbf{u}}=\frac{1}{M}\sum_{i=1}^{M}\mathbf{u}_i \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

is the ensemble mean (i.e., the sample mean that is an estimator of the distribution mean $\mathbf{u}$). [Note that in earlier publications we used the transposed version of $\mathbf{U}$. We modified our notation to bring it in line with that of textbooks such as Conn et al. (2009).] In our implementation of EnOpt, the ensemble members $\mathbf{u}_i$, $i=1, 2, \ldots, M$, are created with

$$\mathbf{u}_i=\overline{\mathbf{u}}+\tilde{\mathbf{C}}^{1/2}\mathbf{z}_i, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

where a Cholesky decomposition is used to calculate $\tilde{\mathbf{C}}^{1/2}$, and where $\mathbf{z}_i$ is drawn from a univariate Gaussian distribution. A mean-shifted objective-function vector is defined as

$$\mathbf{j}=[J_1-\overline{J} \quad J_2-\overline{J} \quad \cdots \quad J_M-\overline{J}]^T, \quad \ldots\ldots\ldots\ldots (6)$$

where values J_i correspond to the simulated response to control vectors $\mathbf{u}_i$, and where

$$\overline{J}=\frac{1}{M}\sum_{i=1}^{M}J_i. \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

If we would have an overdetermined case, that is, for $M>N$, and $\mathbf{U}$ of rank N, the approximate gradient with respect to the controls could be obtained as the least-squares solution for the regression coefficient vector:

$$\mathbf{g}=(\mathbf{U}\mathbf{U}^T)^{-1}\mathbf{U}\mathbf{j}. \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

For the derivation of Eq. 8, see any introductory linear-algebra textbook [e.g., Strang (2006)]. Eq. 8 can also be expressed as

$$\mathbf{g}=\mathbf{C}_{uu}^{-1}\mathbf{c}_{uJ}, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (9)$$

where

$$\mathbf{C}_{uu}=\frac{1}{M-1}(\mathbf{U}\mathbf{U}^T) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

and

$$\mathbf{c}_{uJ}=\frac{1}{M-1}(\mathbf{U}\mathbf{j}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

are the ensemble (sample) covariance matrix and cross-covariance vector, respectively (Chen 2008; Chen and Oliver 2010). (Note that the cross-covariance vector is often referred to as the cross-covariance matrix, in which case the vector is just interpreted as a 1D matrix.) In practical applications, however, we normally have an underdetermined case, that is, $M<N$. This implies that the matrix product $\mathbf{U}\mathbf{U}^T$ is rank-deficient such that we cannot directly compute its inverse and solve the associated system of equations with Eq. 8. Instead, we can compute the Moore-Penrose pseudoinverse with a singular-value decomposition (SVD) (e.g., Strang 2006). We note that this formulation was also described in Dehdari and Oliver (2012), whereas Do and Reynolds (2013) recently demonstrated that it is akin to what is known as a "simplex gradient" in, for example, Conn et al. (2009). Alternatively, Chen (2008) and Chen et al. (2009) propose to simply use

$$\mathbf{g}'=\mathbf{c}_{uJ}=\mathbf{C}_{uu}\mathbf{g} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (12)$$

instead of $\mathbf{g}$ [i.e., to use a regularized (smoothed) approximate gradient in the form of the cross-covariance $\mathbf{c}_{uJ}$]. Moreover, they propose to use a second premultiplication with $\mathbf{C}_{uu}$ as a further regularization that leads to

$$\mathbf{g}''=\mathbf{C}_{uu}\mathbf{c}_{uJ}=\mathbf{C}_{uu}\mathbf{C}_{uu}\mathbf{g}. \quad \ldots\ldots\ldots\ldots\ldots\ldots (13)$$

Alternatively, the premultiplication can be performed with $\tilde{\mathbf{C}}$, leading to

$$\mathbf{g}'''=\tilde{\mathbf{C}}\mathbf{c}_{uJ}=\tilde{\mathbf{C}}\mathbf{C}_{uu}\mathbf{g}. \quad \ldots\ldots\ldots\ldots\ldots\ldots (14)$$

Eqs. 12, 13, and 14 can be interpreted as modified approximate (regularized) gradients. Note that the regularized gradients $\mathbf{g}'$, $\mathbf{g}''$, and $\mathbf{g}'''$ are dimensionally inconsistent, in the sense that their elements do not have the same dimensions as those of $\mathbf{g}$. In our study, we applied Eq. 8 with an SVD on $\mathbf{U}$ with a truncation level of 0.999. Chen and Oliver (2012), Oliveira and Reynolds (2014), and Zhao et al. (2013) use Eq. 13 or Eq. 14 for the gradient estimate. Note that, when diagonal covariance matrices are used, Eqs. 13 and 14 act only as a scaling of the magnitude of the gradient and have no impact on the direction of the gradient. In addition, we believe there is no conclusive evidence to suggest that Eq. 13 or Eq. 14 is always better than Eq. 8, in particular, because the dimensional inconsistency of the expressions implies that the relevance of earlier experiences will be restricted to situations in which J and the elements of $\mathbf{u}$ have the same dimensions expressed in the same units systems as in the earlier case.

Update Rules

One can use the approximate gradient $\mathbf{g}$ from Eq. 8 in any gradient-based optimization algorithm. In our study, we used a simple steepest-ascent scheme according to

$$\mathbf{u}^{\ell+1}=\mathbf{u}^{\ell}+\alpha^{\ell}\mathbf{g}^{\ell}, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (15)$$

where the superscript $[\ell]$ is the iteration counter and $\alpha^{[\ell]}$ is a step length in the direction of the gradient. Note that, to ensure dimensional consistency, α must have dimensions, and its value will

therefore depend on the units system applied. If $\mathbf{u}$ and $\mathbf{g}$ both contain elements with different dimensions (e.g., when the elements of $\mathbf{u}$ are pressures and rates), an additional scaling of the gradient elements may be required. Following Oliveira and Reynolds (2014), we scaled the gradient by its infinity norm and then, for each iteration, used an initial step size $\alpha = 1$. Thereafter, we allowed for a maximum of three back-tracking steps, each time reducing the step size with a factor of one-half. In more-sophisticated optimization algorithms, an improved update direction (i.e., one different from $\mathbf{g}$) is determined by using optimization methods that make use of the second derivatives of J with respect to $\mathbf{u}$ (i.e., of the Hessian matrix, or, more commonly, of approximations to the Hessian). In particular, so-called quasi-Newton methods use gradient information of subsequent iterates to construct an approximate Hessian $\mathbf{H}^{\ell}$. The corresponding update rule then becomes

$$\mathbf{u}^{\ell+1} = \mathbf{u}^{\ell} + \alpha^{\ell}(\mathbf{H}^{\ell})^{-1}\mathbf{g}^{\ell}, \quad \ldots\ldots\ldots\ldots\ldots\ldots (16)$$

where the definition of $\mathbf{H}^{\ell}$ depends on the particular type of quasi-Newton method applied; see, for example, Nocedal and Wright (2006) or Luenberger and Ye (2010) for further details. Unlike Eqs. 13 and 14, Eq. 16 is dimensionally consistent if α is taken dimensionless, because $\mathbf{H}^{-1}$ acts as a natural scaling matrix for the gradient vector $\mathbf{g}$. Note that, as usual, in an actual implementation, computing the inverse is avoided, and a system of equations is solved instead. The gradient is the direction of a tangent (hyper) plane in a point touching the objective function, whereas the Hessian gives curvature information in that point (i.e., it defines a convex quadratic function). The basic idea underlying the various quasi-Newton methods is that the curvature information contained in the approximate Hessian is gradually increased by subsequent inclusion of gradient information from previous iterations. Although we do not use a quasi-Newton algorithm in the optimization examples in our study, the concept of the use of information from subsequent iterates to improve the estimate of the curvature of the objective function is an important aspect of covariance matrix adaptation-evolutionary strategy, and thus also of covariance matrix adapted-ensemble optimization. Moreover, we note that the use of the preconditioners $\mathbf{C}_{uu}$ and $\tilde{\mathbf{C}}$ in Eqs. 13 and 14 seems to play a role similar to that of the preconditioner $\mathbf{H}^{-1}$ in Eq. 16. However, $\mathbf{C}_{uu}$ and $\tilde{\mathbf{C}}$, unlike $\mathbf{H}^{-1}$, do not restore dimensional consistency.

Covariance-Matrix Adaptation

Covariance matrix adaptation-evolutionary strategy (CMA-ES) is a stochastic iterative optimization method in which the covariance matrix is updated at every iteration such that its largest principal direction (i.e., the eigenvector corresponding to its largest eigenvalue) is (approximately) realigned in the direction of the maximal increase of the objective function. CMA-ES uses two types of updates for the covariance matrix, as briefly explained later in this text. For a detailed overview, we refer to Hansen (2011).

Rank-μ Update. The motivation behind a rank-μ update is to use information obtained within one single iteration (i.e., one ensemble of random control vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M\}^{[\ell]}$ and their corresponding objective-function values $\{J_1, J_2, \ldots, J_M\}^{[\ell]}$ through selecting the "best" μ members (i.e., those corresponding to the μ highest objective-function values) out of the M ensemble members:

$$\tilde{\mathbf{C}}_{uu}^{\ell+1} = (1-c_{\mu})\tilde{\mathbf{C}}_{uu}^{\ell} + c_{\mu}\frac{1}{\mu}\tilde{\mathbf{U}}\tilde{\mathbf{U}}^{T}, \quad \text{with}$$
$$\tilde{\mathbf{U}} = [\,\mathbf{u}_1 - \mathbf{u}^{\ell} \quad \mathbf{u}_2 - \mathbf{u}^{\ell} \quad \cdots \quad \mathbf{u}_{\mu} - \mathbf{u}^{\ell}\,], \quad \ldots\ldots\ldots (17)$$

where $0 < c_{\mu} < 1$ is a learning rate and where the control vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$ have been ranked such that, for their corresponding objective-function values, it holds that $J_1 \geq J_2 \geq \cdots \geq J_{\mu} > J_{\mu+1} \geq \cdots \geq J_M$. Eq. 17 is called a rank-μ update because the matrix product $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^{T}$ is, at most, of rank μ. Note that we use the distribution mean $\mathbf{u}^{\ell}$ instead of the ensemble mean $\bar{\mathbf{u}}^{\ell}$. It can be shown that for $\mu = M$ and $c_{\mu} = 1$, $\tilde{\mathbf{C}}_{uu}^{\ell+1}$ would be an unbiased estimator of the distribution covariance $\tilde{\mathbf{C}}^{\ell+1}$ (Hansen 2006). However, because we typically choose $\mu < M$, the entries of the covariance matrix will be selectively influenced by the ensemble members corresponding to the μ highest objective-function values. The choice of the learning rate c_{μ} turns out to be crucial to the success of the optimization, as is demonstrated later. Hansen (2011) discusses strategies to determine optimal values for c_{μ} (and for learning rate c_1 introduced later) on the basis of the dimension of the problem. However, his test cases are of a relatively small dimension, whereas production-optimization problems typically have hundreds to thousands of control variables. The choice of μ is up to the user; in this study, we used $\mu = M/4$.

Rank-One Update. In Eq. 17, the covariance-matrix update is determined with the best objective-function values within one single iteration. It is also possible to update the covariance matrix with information from previous iterates, in a fashion similar to that of updating the Hessian in quasi-Newton methods. [For remarks about the relationship between the Hessian and the covariance matrix, see Hansen (2011).] The expression for such an update, derived in Hansen (2006), is given by

$$\tilde{\mathbf{C}}_{uu}^{\ell+1} = (1-c_1)\tilde{\mathbf{C}}_{uu}^{\ell} + c_1\mathbf{e}^{\ell+1}(\mathbf{e}^{\ell+1})^{T}, \quad \ldots\ldots\ldots\ldots (18)$$

where c_1 is again a learning rate and $\mathbf{e}$ is the "evolution path," which is a function of iterates $\mathbf{u}^{\ell}$ in earlier steps. Roughly speaking, $\mathbf{e}$ is obtained as a summation of previous iterates emphasizing the most recent iterates while gradually "forgetting" the earlier ones. For the exact definition of $\mathbf{e}$, see Hansen (2006). Because the outer product of two vectors results in a matrix of rank one, Eq. 18 is referred to as a "rank-one update". The rank-one update was shown to be particularly powerful when using small ensemble sizes with CMA-ES (Hansen 2006).

Combined-Rank Update. Combining Eqs. 17 and 18, one obtains the update rule

$$\tilde{\mathbf{C}}_{uu}^{\ell+1} = (1-c_{\mu}-c_1)\tilde{\mathbf{C}}_{uu}^{\ell} + c_{\mu}\underbrace{\frac{1}{\mu}\tilde{\mathbf{U}}\tilde{\mathbf{U}}^{T}}_{\text{rank-}\mu\text{ update}} + c_1\underbrace{\mathbf{e}^{\ell+1}(\mathbf{e}^{\ell+1})^{T}}_{\text{rank-one update}} \quad \ldots\ldots\ldots\ldots\ldots (19)$$

Eq. 19 uses information within one iteration as well as information from previous iterations. Hansen (2011) suggests that the former is more important when using a larger ensemble and that the latter is more important when using smaller ensembles. Several variations to Eq. 19 were proposed; see, for example, Ros and Hansen (2008) and Arnold and Hansen (2010) wherein the off-diagonal elements are set to zero and only the diagonal elements (i.e., the variances of the covariance matrix) are updated:

$$\tilde{C}_{uu,ii}^{\ell+1} = (1-c_{\mu}-c_1)\tilde{C}_{uu,ii}^{\ell} + c_{\mu}\underbrace{\frac{1}{\mu}\tilde{U}_{ii}\tilde{U}_{ii}^{T}}_{\text{rank-}\mu\text{ update}} + c_1\underbrace{e_i^{\ell+1}(e_i^{\ell+1})^{T}}_{\text{rank-one update}}, i = 1, \ldots, M. \quad \ldots\ldots\ldots (20)$$

CMA-EnOpt

In EnOpt, a constant-distribution covariance matrix $\tilde{\mathbf{C}}$ is used which, for uncorrelated controls of the same type, is typically chosen as a diagonal covariance matrix with equal diagonal elements σ^2. Often, a time correlation is imposed on the controls, resulting in a block-diagonal matrix with each block corresponding to the

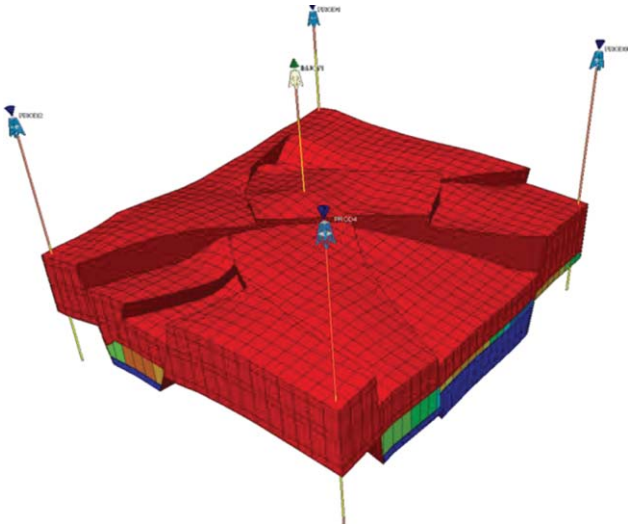

Fig. 1—Five-spot reservoir model. The colors indicate the initial water saturation.

control timesteps of a single well or inflow control valve. The value of the standard deviation σ can have a strong influence on the quality of the approximate gradient and therefore on the performance of the optimization algorithm. However, there is no well-defined method to choose this value. (The same is true for the choice of the time-correlation length.) Thus, we propose to use the covariance-adaption strategy described previously to gradually improve the distribution-covariance matrix in EnOpt, leading to CMA-EnOpt.

Algorithm:

1. Set $\ell = 0$. Choose an initial control vector $\mathbf{u}^0$ and evaluate the corresponding objective-function value.
2. Generate an ensemble of randomly perturbed controls around $\mathbf{u}^0$ from an initial, user-defined, diagonal covariance matrix $\tilde{\mathbf{C}} = \sigma^2\mathbf{I}$ or block-diagonal covariance matrix $\tilde{\mathbf{C}}$.
3. Run a reservoir simulation for every member of the perturbed control ensemble and calculate the corresponding objective-function values with Eq. 1.
4. Compute the EnOpt gradient $\mathbf{g}^\ell$ with Eq. 8.
5. If the optimization-stopping criterion, or the maximum allowed number of iterations, is achieved, stop. Otherwise, set $\ell = \ell + 1$.
6. Determine a step size α^ℓ and compute an updated control vector $\mathbf{u}^{\ell+1}$ with Eq. 15 and the corresponding objective-function value.
7. Compute the updated covariance matrix $\tilde{\mathbf{C}}_{uu}^{\ell+1}$ with either Eq. 19 or Eq. 20.
8. Regenerate an ensemble of randomly perturbed controls around $\mathbf{u}^{\ell+1}$ from the updated covariance matrix $\tilde{\mathbf{C}}_{uu}^{\ell+1}$.
9. Go to Step 3.

3D Synthetic Reservoir Model

Advances in technology have led to an increase in the application of inflow-control valves (ICVs) to regulate flow rates and maintain pressure in the reservoir. We consider a control problem where ICV settings of injection and production wells in a 3D synthetic reservoir model are manipulated to optimize waterflooding over the producing life of the reservoir. The model, illustrated in **Fig. 1,** consists of $25 \times 32 \times 5 = 4{,}000$ gridblocks. The approximate size of the gridblocks is $110 \times 90 \times 20$ m, such that the model represents a volume of $2.5 \times 3.5 \times 0.1$ km. The geological structure consists of uplifted blocks, separated by faults. The reservoir is produced with an inverted five-spot well pattern (i.e., four producers at the corners of the grid with an injector in the center). The reservoir is divided into five layers with different horizontal permeabilities. There is a sealing fault on the northwestern side of the block, close to Producer 1. **Table 1** lists the reservoir and fluid properties of the model.

A Corey model with exponents equal to 2 for both oil and water is used for the relative permeabilities in which the connate water saturation is 0.2, the residual oil saturation is 0.3, and the endpoint relative permeabilities to oil and water are 0.8 and 0.4, respectively. No capillary pressures are included. The wells penetrate all five layers with an ICV in every layer, resulting in a total of 25 controls per timestep. The producing life of the reservoir is divided into 15 time intervals of 1 year (365 days) each, which results in a total of $15 \times 25 = 375$ controls to be optimized. Water is injected at a constant pressure of 300 bar, and the production wells are operated at a minimum pressure of 15 bar. We used an oil price $r_o =$ USD 130/m^3, water-production costs $r_{wp} =$ USD 25/m^3, and water-injection costs $r_{wi} =$ USD 6/m^3. The discount rate b was set to zero. Well-index multipliers were used to model the ICVs in the simulator with bounds of 1×10^{-4} and unity. For the simulation of the model, we used a commercial, fully implicit finite-difference black-oil simulator (Eclipse 2011).

Comparison Between EnOpt and CMA-EnOpt

We performed several comparisons between EnOpt and CMA-EnOpt to optimize the inflow-control-valve (ICV) settings with the aim to maximize net present value as defined in Eq. 1. The starting point for the optimization was an initial control vector with values equal to unity. Thus, all the ICVs were fully open as a starting strategy. The initial value of σ, for use in a diagonal covariance matrix, was chosen equal to 0.1, and we used a fixed ensemble size of 50 samples. Random control values outside the range $[1 \times 10^{-4}, 1]$ were simply reset to their bounds. We first compare the results with only a diagonal update for the covariance matrix; see Eq. 12. The optimization was allowed to run for 50 iterations that usually resulted in a near-horizontal (i.e., nearly converged) objective-function graph for the EnOpt method; see **Fig. 2.**

Note that in the use of approximate gradients and inexact line-search techniques, it is very well possible that the curves do not

Property	Values	Units
Porosity	0.2	–
Permeability (Layers 1 through Layer 5)	100, 300, 50, 600, 100	md
Reservoir pressure at 1950 m	200	bar
Density of oil	800	kg/m^3
Density of water	1000	kg/m^3
Temperature	77	°C
Oil compressibility at 200 bar	4×10^{-5}	1/bar
Water compressibility at 200 bar	4×10^{-5}	1/bar
Rock compressibility	0	1/bar
Viscosity of oil at 1 bar	2	cp
Viscosity of water at 1 bar	0.5	cp

Table 1—Reservoir and fluid properties.

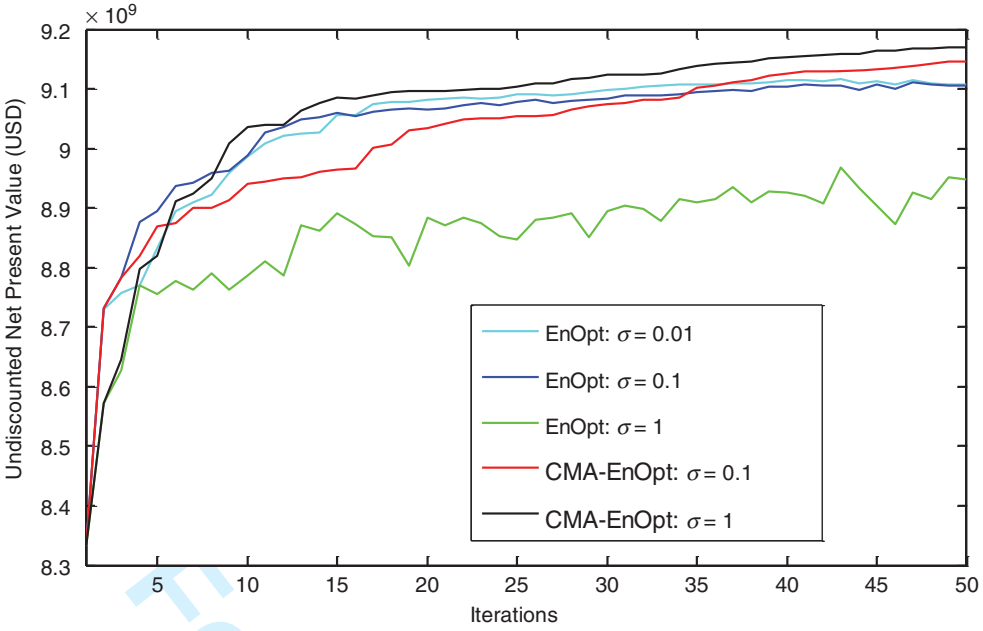

Fig. 2—Comparison of optimization performance for different initial covariance matrices.

always increase monotonically. We have chosen to allow for this to happen. Although monotonicity could be enforced, this would induce a higher computational cost and an increased risk of becoming trapped in a local optimum. We used various settings of the initial-distribution covariance matrix (i.e., of the standard deviation σ) to generate the initial ensemble, leading to different optimization results; see Fig. 2. The best EnOpt run resulted in an objective-function value of USD 9.1×10^9 whereas the CMA-EnOpt run for the same initial covariance distribution achieved a slightly higher value of USD 9.15×10^9 (i.e., 0.7% higher). An illustration of the corresponding ICV settings for one of the wells is presented in **Fig. 3,** which shows the differences in the control strategies between EnOpt (blue line) and CMA-EnOpt (red line). The major difference is in the nature of the controls: CMA-EnOpt obtains controls that switch almost completely between the upper and lower bounds. We note that such (near-) bang-bang controls are, in some optimization problems, the optimal solutions, and are often easier to implement in practice. As shown by Zandvliet et al. (2007), a locally optimal solution of an optimal control problem in which the system equations and the integral term in the objective function are both linear in the controls, and that has only simple bound constraints or linear constraints on the controls, is necessarily a bang-bang solution (possibly in combination with singular arcs, i.e., areas in which the solution is not bang-bang). In our example, we have indeed only bound constraints on the inputs, that is, on the ICV settings, while also the integral term in the objective function and the system equations are linear in the controls (i.e., the problem is control-affine). One can see the latter by considering that the oil- and water-flow rates q_o and q_{wp}, respectively, that appear in the objective function in Eq. 1 are obtained with the aid of a well model that can be written, for a single well, in general form as (Jansen 2013)

$$\begin{bmatrix} q_o \\ q_{wp} \end{bmatrix} = \begin{bmatrix} f_o \\ f_w \end{bmatrix} I_{\text{well}} \left(p_{\text{well}} - p_{gb} \right) \beta, \quad \ldots\ldots\ldots\ldots\ldots (21)$$

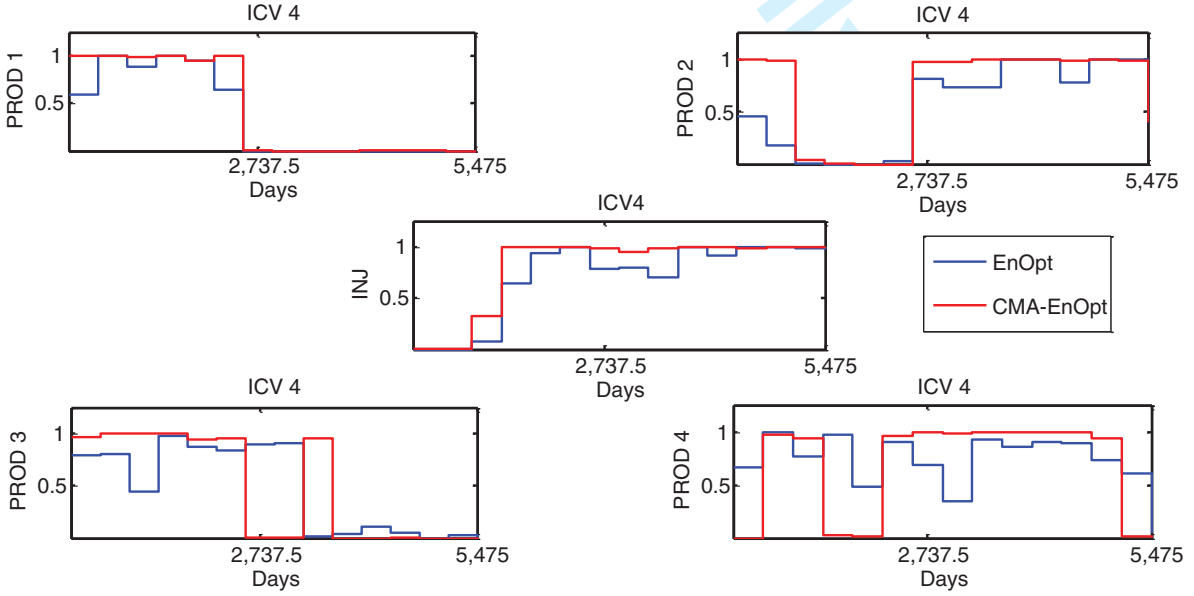

Fig. 3—Optimal control settings for ICV 4 as computed by EnOpt (blue) and CMA-EnOpt (red) with initial $\sigma = 0.1$ for both cases.

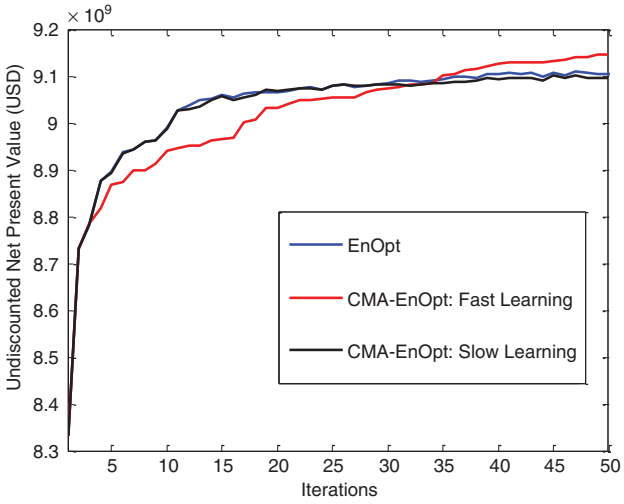

Fig. 4—Impact of optimization method and learning rates: EnOpt (blue curve) and CMA-EnOpt with low learning rates (0.5%, black curve) and with high learning rates (25%, red curve).

where f_o and f_w are fractional flows, which are functions of the water saturation in the well gridblock (a state variable); I_{well} is the well index, which, through its dependence on relative permabilities, is also a function of the gridblock's water saturation; p_{well} is the wellbore pressure, p_{gb} is the well gridblock pressure (also a state variable); and β is the dimensionless ICV opening (well-index multiplier; i.e., the control variable). For the water-injection rate q_{wi} in Eq. 1, a similar expression holds but with $q_o = 0$ and $f_w = 1$. The full analysis of necessary and sufficient conditions for bang-bang control would require rewriting the optimization problem in terms of an optimal switching time with the aid of an adjoint solution, which we therefore did not pursue for this example. We refer to Zandvliet et al. (2007) for detailed information on the conditions for bang-bang control.

Improved Robustness

Exact gradient-based methods (in combination with an exact line search) are inherently local methods because they always result in uphill directions, unless forced to take steps in other directions to search for other optima. Most gradient-free methods were shown to possess more-global search characteristics. In the EnOpt method, one may choose the initial-distribution covariance matrix to mimic the desired behavior: Large variances result in a more-global search strategy and small variances in a more-local one. We tested EnOpt and CMA-EnOpt with different initial starting values for the standard deviation σ. Fig. 2 illustrates that the choice of the initial covariance matrix has a significant impact on the performance of EnOpt, because the matrix remains constant throughout the optimization. When using $\sigma = 1$ for EnOpt (green line), the algorithm achieves poor results, probably because the sampling takes place in a too-large area, which leads to poor approximations of the gradient. For $\sigma = 0.1$ (blue line) and $\sigma = 0.01$ (light-blue line), better results are achieved. For CMA-EnOpt (red and black lines), the initial starting covariance matrices have relatively little impact on the final objective-function values, and both runs achieve better results than EnOpt. When starting with $\sigma = 1$, CMA-EnOpt achieves a slightly higher objective-function value than when starting with $\sigma = 0.1$. However, if the algorithm is continued for more iterations, we observe (not shown here) that both the curves (red and black) achieve very similar objective-function values. For this example, we therefore find that CMA-EnOpt performs better than EnOpt with an optimized-distribution covariance matrix, and much better than EnOpt with a poor initial guess for the covariance. The main benefit of CMA-EnOpt, therefore, appears to be its robustness with respect to initial choices of the covariance matrix.

Learning Rates

CMA-EnOpt contains several parameters that require user-defined values (particularly, the learning rates c_μ and c_1). **Fig. 4** illustrates that if we choose the learning rates too low, the advantage of CMA-EnOpt over EnOpt is negligible, if any. Higher learning rates are seen to achieve significantly better results. Here, a high learning rate (fast learning) corresponds to a 75 to 25% update rule with $c_\mu = 0.20$ and $c_1 = 0.05$. In the case $(c_\mu + c_1) \times 100\% = 25\%$, new information is incorporated into the covariance matrix every iteration. A low learning rate (slow learning) corresponds to a 99.5 to 0.5% update rule with $c_\mu = 0.004$ and $c_1 = 0.001$; in the case $(c_\mu + c_1) \times 100\% = 0.05\%$, new information is incorporated. These results were obtained when we only updated the diagonal elements of the covariance matrix. Hansen (2011) reports, for covariance matrix adaptation-evolutionary strategy, that if the full covariance matrix is updated, high learning rates can have a detrimental impact on the optimization because they may lead to covariance-matrix degeneration (i.e., to a situation in which the elements of the covariance matrix become so small that sampling takes place in a too-small area).

Fig. 5 illustrates estimated standard deviations σ for two control variables (i.e., the square root of the corresponding diagonal values of the distribution covariance matrix) for different learning rates. We observe from Fig. 5 that different control variables have different optimal standard deviations at each iteration. The CMA-EnOpt algorithm generally results in a gradual decrease of the standard deviation of a control when approaching the optimum. For Control 19 in Fig. 5, the magnitude of the corresponding gradient is an order higher than that of Control 299, indicating that either Control 19 is still much farther from the optimum than Control 299, or that the objective function has a much higher curvature in the direction of Control 19. We note that of the 375 controls, only five controls have standard deviations similar to Control 19 (i.e., standard deviations higher than the initial standard deviation) with Control 19 being the highest, whereas the remaining 370 controls have much lower standard deviations. The low learning rates, on the basis of the recommendations for small-sized problems as described in Hansen (2011) and Hansen (2006), lead to negligible changes in the standard deviation (black line, overlapped by the blue line in Fig. 5). In this case, CMA-EnOpt behaves similarly to EnOpt with a fixed standard deviation (blue line) for all controls. Thus, learning rates have a significant impact on the performance of the optimization algorithm; if chosen too conservatively, the advantage of CMA-EnOpt over EnOpt will be negligible.

Correlations and Block-Diagonal Update

Imposing correlations over control times (effectively imposing smoothness on the control solution) may lead to an improved efficiency of the EnOpt algorithm when using many control timesteps; see, for example, Chen et al. (2009), Leeuwenburgh et al. (2010), and Oliveira and Reynolds (2014). The CMA-EnOpt results presented thus far were obtained with a diagonal update of $\tilde{\mathbf{C}}_{uu}$, with Eq. 20, without imposing smoothness on the controls. **Fig. 6** illustrates the impact of introducing nonzero correlations on the control perturbations over time with the spherical correlation function as defined in Zhao et al. (2013) in which the correlation length is set equal to the total number of control timesteps. This results in the red curve; that is, a diagonal CMA-EnOpt update with an additional correlation (smoothing). The black curve represents the case without correlation. Although, in this case, the impact of imposing a correlation over time only marginally increases the objective-function value, the red curve is nearly always above the black curve, which suggests an improved computational efficiency.

To obtain those results, an arbitrary correlation length was chosen. The correlation length can have a significant impact on the results in different problems, but unfortunately there is no predefined way of knowing the ideal correlation length, as shown in

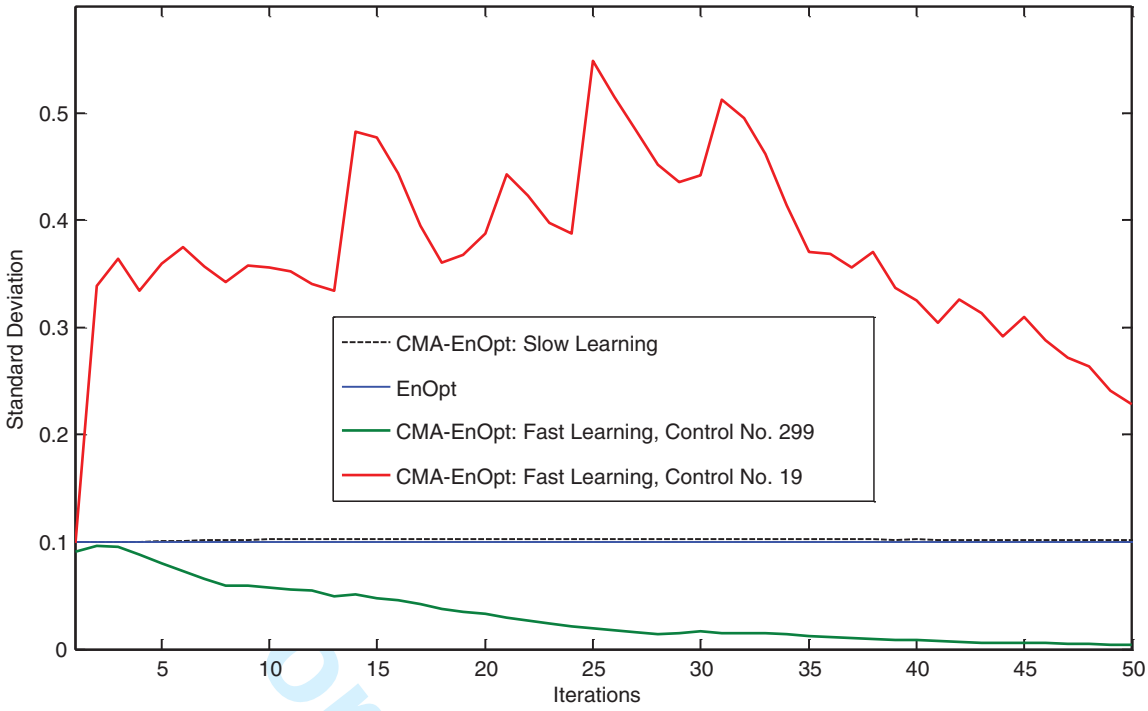

Fig. 5—Standard deviations for two control variables for different optimization methods and learning rates: CMA-EnOpt (high learning rates: red and green curves), CMA-EnOpt (low learning rates: black curve), and EnOpt (blue curve, nearly overlapping the black curve).

Oliveira and Reynolds (2014). The efficiency of using such a correlation function in conjunction with the diagonal-update-based CMA-EnOpt algorithm raises the question as to whether it is possible to estimate the optimal correlation between the controls with the CMA-EnOpt method. To investigate this, we introduce a block-diagonal update of $\tilde{\mathbf{C}}_{uu}$ for the CMA-EnOpt method with Eq. 19 in which off-diagonal covariance updates also are allowed, but only for cross-covariances between controls belonging to the same inflow-control valve (ICV). Thus, instead of using an artificial or predefined correlation function and length to impose smoothness on the controls, we allow the CMA-EnOpt algorithm to estimate the optimal cross-correlation between the controls; that is, each ICV is associated with its own 15×15 covariance matrix (in which 15 is the total number of control timesteps), starting from a diagonal covariance matrix. The blue curve in Fig. 6 represents this "block-diagonal" update, which performs better than both the diagonal update (black curve) and the diagonal update with smoothing (red curve). As a comparison, we also introduced time-correlations (smoothing) in EnOpt with the aid of a (constant) block-diagonal covariance matrix $\tilde{\mathbf{C}}$. With different correlation lengths (green and yellow lines), we achieve higher objective-function solutions in the earlier iterations but solutions

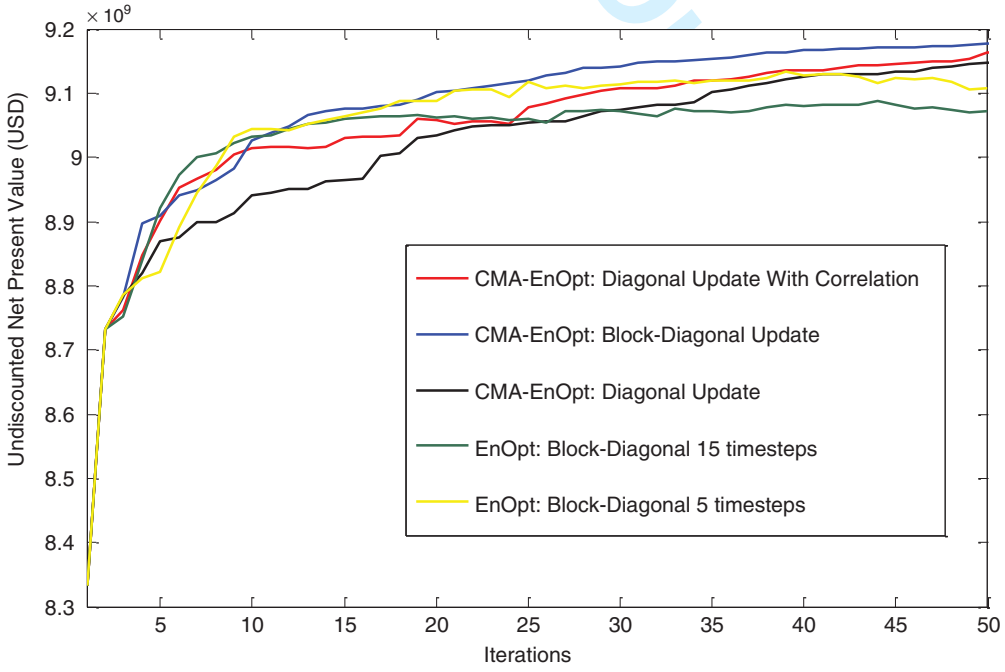

Fig. 6—Impact of imposing time correlations.

Gradient Formulation for EnOpt	Gradient Formulation for CMA-EnOpt	EnOpt: Block-Diagonal, 5-Timestep Correlation	EnOpt: Block-Diagonal, 15-Timestep Correlation	CMA-EnOpt Block-diagonal	CMA-EnOpt Diagonal, 5-Timestep Correlation	CMA-EnOpt Diagonal, 15-Timestep Correlation
$\mathbf{g}$	$\mathbf{g}$	9.102×10^{9}	9.073×10^{9}	9.175×10^{9}	9.165×10^{9}	9.14×10^{9}
$\tilde{\mathbf{C}}\,\mathbf{g}$	$\tilde{\mathbf{C}}_{uu}\,\mathbf{g}$	9.103×10^{9}	9.035×10^{9}	9.001×10^{9}	9.132×10^{9}	9.05×10^{9}
$\tilde{\mathbf{C}}\;\tilde{\mathbf{C}}\,\mathbf{g}$	$\tilde{\mathbf{C}}_{uu}\;\tilde{\mathbf{C}}_{uu}\,\mathbf{g}$	9.088×10^{9}	9.018×10^{9}	8.983×10^{9}	9.112×10^{9}	8.925×10^{9}

Table 2—Effect of regularization. All results are expressed as objective function value in $.

with lower objective-function values at the final iteration compared with CMA-EnOpt. Also, in Fig. 4, EnOpt seems to perform better early in the optimization, which suggests using EnOpt early and switching to CMA-EnOpt only later during an optimization run. However, it is unknown in advance what is the correct point during the optimization to switch between the two methods, and we did not pursue this possibility. Finally, we checked the influence of regularization by (double) premultiplication of the gradient $\mathbf{g}$ with $\tilde{\mathbf{C}}$ (constant, for EnOpt) or with $\tilde{\mathbf{C}}_{uu}$ (updated after every iteration, for CMA-EnOpt), similar to the premultiplications in Eqs. 12, 13, and 14. **Table 2** depicts the results, and it follows that, for this particular example, regularization generally does not have a positive effect and, in nearly all cases, results in somewhat lower objective-function values.

Full-Matrix Update

In the Theory section, we discussed the similarity between CMA-EnOpt and quasi-Newton methods, with an adapted covariance matrix that is approximately similar to the inverse of the Hessian matrix. So far, however, we did not use the full covariance matrix for two reasons:

- Bouzarkouna et al. (2011) and Ros and Hansen (2008) show that for certain types of objective functions, such as net present value (NPV), which can be decomposed into a sum of the individual NPVs from each well, an uncorrelated (diagonal) covariance matrix achieves better solutions in comparison to using a full covariance matrix.
- Because the updated covariance matrix is used to sample a new ensemble of controls for a gradient estimate of the next iteration, either an eigen or Cholesky decomposition of the covariance matrix is needed to generate the ensemble of controls, which can be computationally demanding especially in problems of a high dimension.

In view of these points we, so far, used diagonal or block-diagonal updates. This approach is supported by the results shown in **Fig. 7,** which compares the various covariance-matrix update types available with CMA-EnOpt for this model.

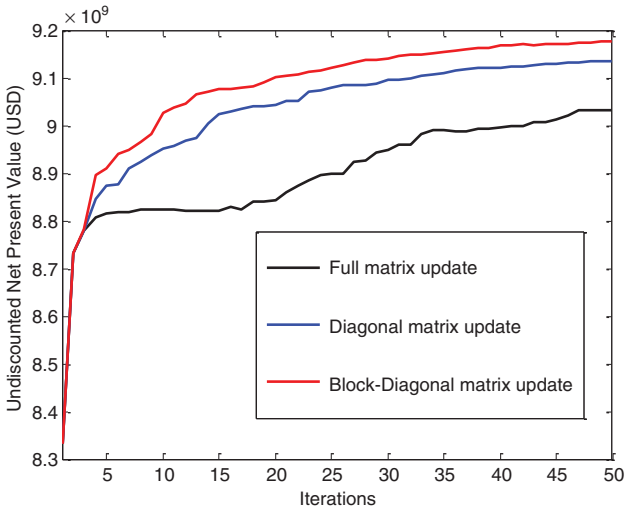

Fig. 7—Comparison of the impact of different matrix updates.

We observe that with the full-matrix update (black curve), the achieved solutions are inferior to those resulting from the diagonal update (blue curve), which we believe is a result of the introduction of "unnecessary" correlations between controls that should be, logically, uncorrelated (e.g., wells on opposite corners of the five-spot pattern). The block-diagonal update (red curve) of the covariance matrix achieves the best results. There are a number of advantages associated with the use of a block-diagonal update in this example:

- The obtained objective-function values are higher than for other update types.
- The solutions are somewhat smoother, as shown in **Fig. 8.**
- There is no need to define a correlation length.
- The computational burden of decomposing a large matrix is avoided.

In addition, some of the rotational aspects of the covariance matrix are retained, which is known to be especially useful when dealing with objective-function search spaces that contain ridges, in which case the elongated sampling area may align itself with a ridge.

Update Types

In the CMA-EnOpt equations written previously, there are two different sources of information used for the matrix update: rank-one and rank-μ updates. Although the literature on covariance matrix adaptation-evolutionary strategy suggests that the former is more important in smaller ensembles and the latter in larger ensembles, we investigated which update is more useful for our case. **Fig. 9** illustrates the effect of the different matrix-update types, and shows that for our problem the rank-μ update (blue curve) performs significantly better than the rank-one update (green curve), in terms of objective-function value and computational efficiency. In fact, for this model and learning rates, the rank-one update has a detrimental effect on the optimization, as one can see in the combined-rank update (black curve), the result of which lies between the different rank updates. We note that further testing is required to obtain full insight into the relationships between various aspects of the CMA-EnOpt algorithm, such as optimal learning rates and update strategies.

Comparison With Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES)

The motivation for CMA-EnOpt was derived from the CMA-ES algorithm given in Hansen (2006). Therefore, an obvious next step would be the comparison of EnOpt and CMA-EnOpt with CMA-ES (i.e., with the evolutionary strategy that formed the basis for the covariance-matrix adaptation strategy in CMA-EnOpt). Initial comparisons, illustrated in **Fig. 10,** indicate that the CMA-ES results are inferior to those of the two EnOpt varieties. CMA-ES achieves results that are approximately 2.5% lower than CMA-EnOpt for the diagonal update and 1.4% lower for the full-matrix update, while also EnOpt achieves better results than CMA-ES for the same computational burden (i.e., 50 optimization iterations). Therefore, explicitly using (approximate) gradient information seems to pay off for this problem. (Note: Tuning parameters such as ensemble size, initial covariance matrix, and learning rates are exactly the same for all the results in

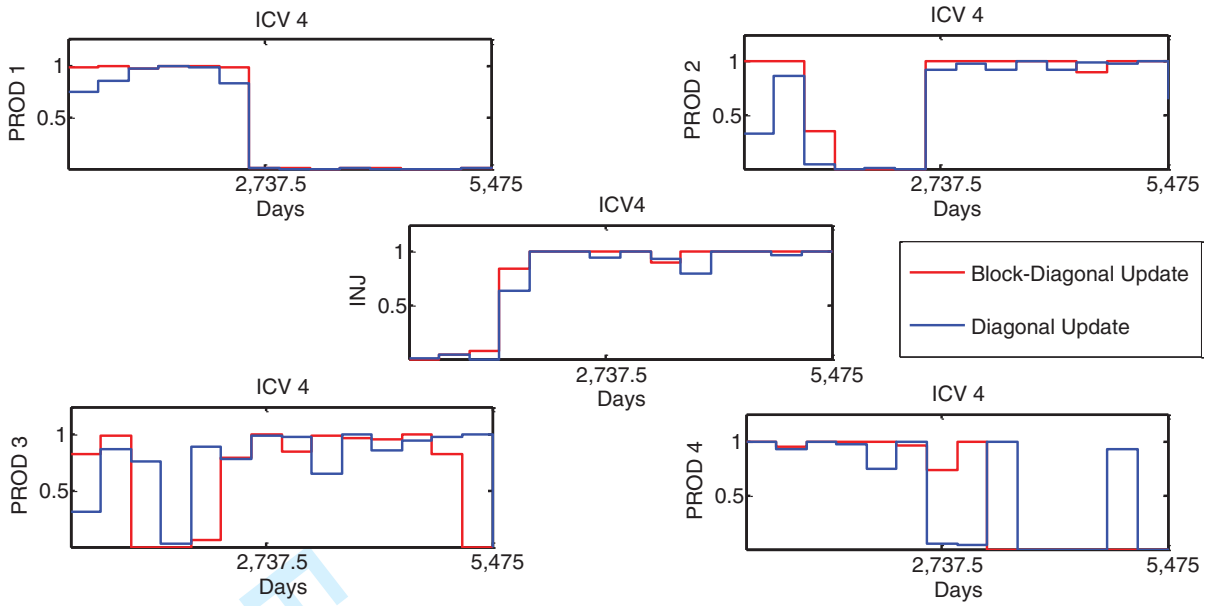

Fig. 8—Optimal control settings for ICV 4, corresponding to different CMA-EnOpt update types.

Fig. 10.) However, further numerical comparisons are required to evaluate the strength and weaknesses of the three methods.

Effect of Different Random-Number Sequences

The results shown in this paper are based on single runs starting from the same initial random-number seed. A comparison of stochastic methods requires multiple sets of comparisons with different random-number sequences. Chen and Oliver (2012) have shown that sizeable variability in the results for different random seeds exists for cases in which smaller ensembles are used to estimate the gradient. When they use a larger ensemble size, which is still smaller than the size used in our paper, the variability is reduced. We have run our experiments with five different random seeds, the results of which are displayed in **Fig. 11.** We observe the trends to be similar to the earlier results in our paper (i.e., CMA-EnOpt always outperforms EnOpt). In addition, there is a marginal reduction in variability when using CMA-EnOpt, but we may need more runs to conclude definitively in this regard. Note the scale of the *y*-axis in Fig. 11: The range of net present values that defines the variability is rather small, so even for EnOpt, the variability is not very large, which is in accordance with the results shown in Chen and Oliver (2012).

Modified Brugge Model

To test our results on a more-complex and challenging reservoir model, we selected a modified version of the Brugge benchmark model (Peters et al. 2010, 2013). The model, shown in **Fig. 12,** consists of 60,048 gridblocks. It represents a segment of an anticlinal structure with one major fault. The reservoir is produced with a peripheral well pattern (i.e., 10 injectors on the down-flank of the structure and 20 producers toward the top of the structure). The reservoir is divided into nine layers with varying permeabilities and porosities. All rock and fluid properties were chosen identical to those in the original Brugge model (Peters et al. 2010, 2013). However, the oil/water contact has been lowered to 1780 m (compared with an original contact depth of 1678 m), thus increasing the stock-tank oil initially in place of the model, while

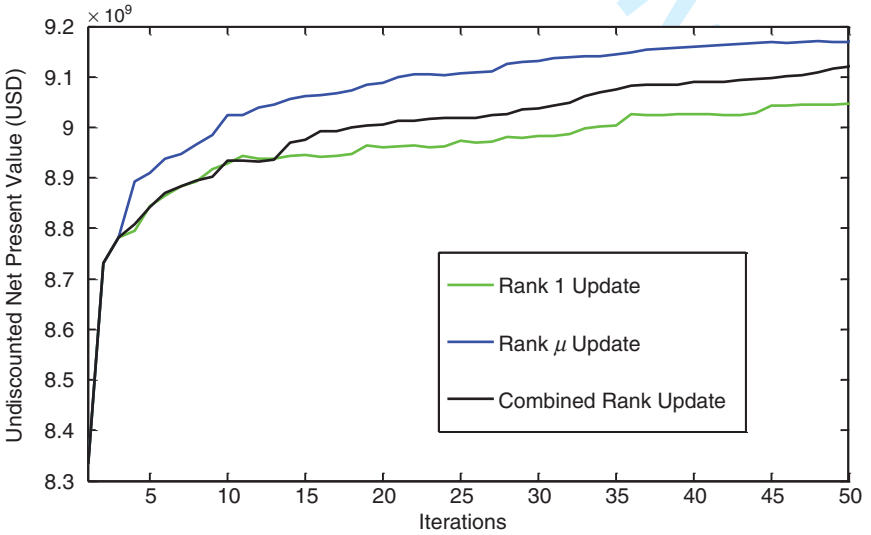

Fig. 9—Comparison of the impact of different rank updates. Rank-1 update: $c_\mu = 0$ and $c_1 = 0.05$. Rank-μ update: $c_\mu = 0.20$ and $c_1 = 0$. Combined update: $c_\mu = 0.20$ and $c_1 = 0.05$.

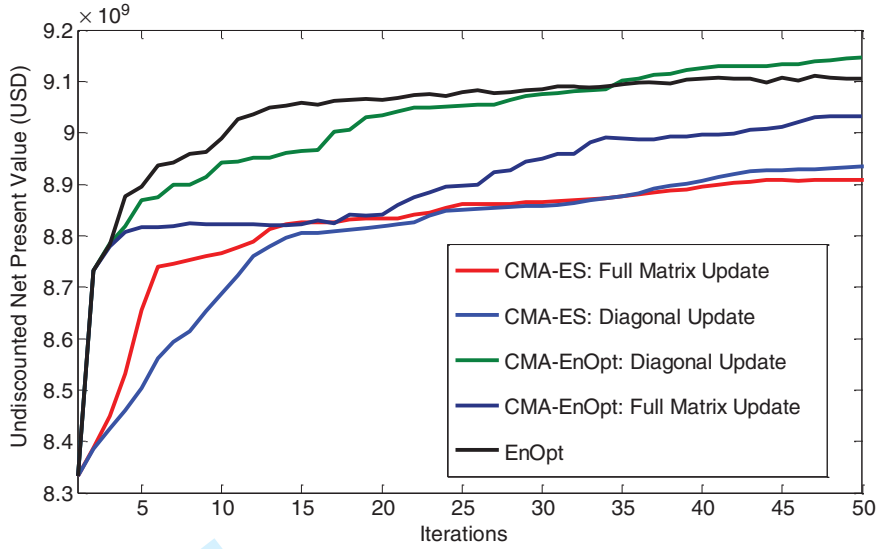

Fig. 10—Performance comparison of CMA-ES with CMA-EnOpt on the five-spot model with a full-matrix update (red, purple) and a diagonal-matrix update (blue, green).

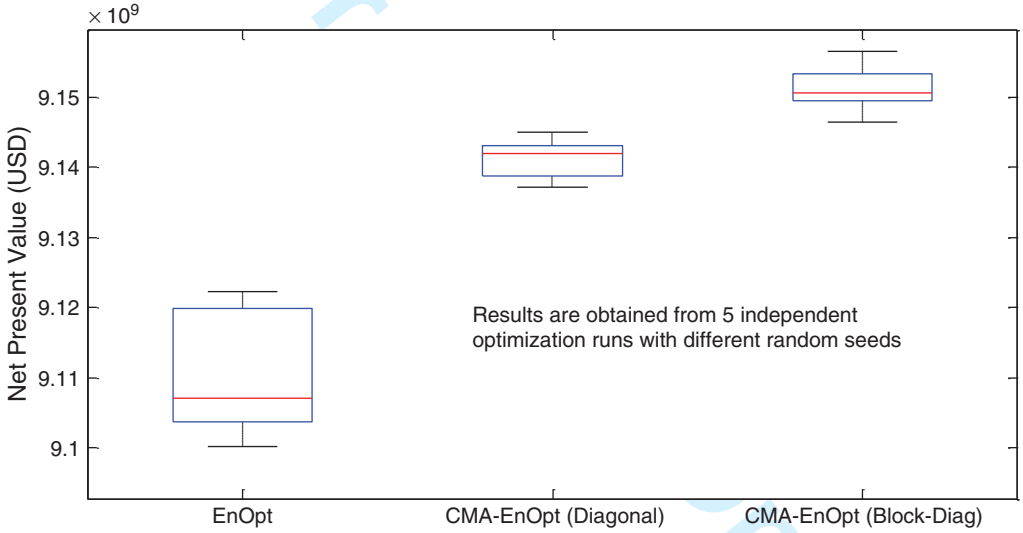

Fig. 11—Box plot to illustrate the effect of different random seeds on the stochastic ensemble-optimization method. Note the scale of the *y*-axis, which implies that the net-present values are quite close. Red lines: median. Whiskers (black): maximum and minimum values. Box edges (blue): 25 and 75%.

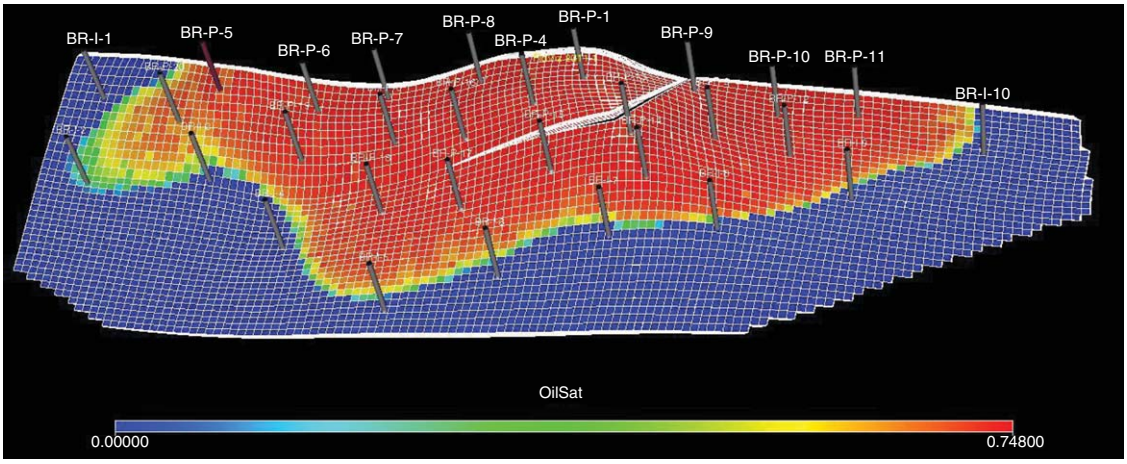

Fig. 12—Modified Brugge reservoir model. The colors indicate the initial oil saturation.

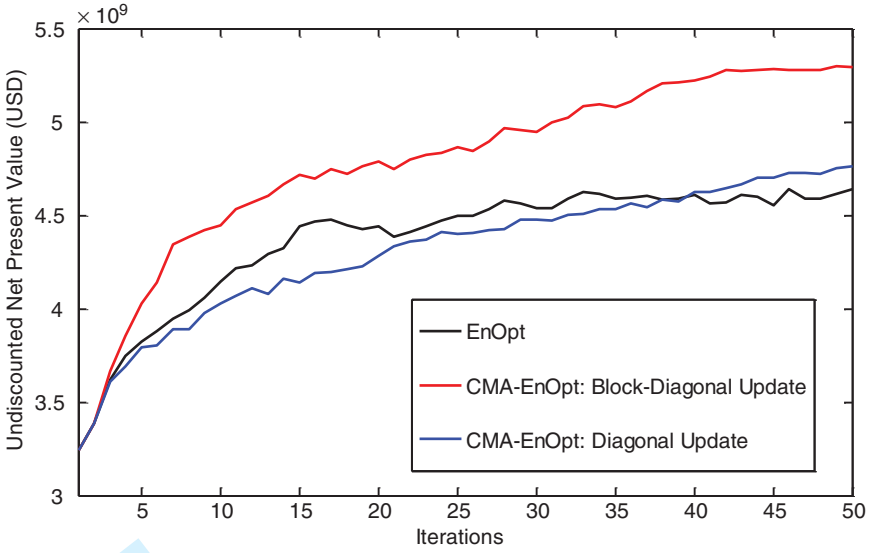

Fig. 13—Comparison of different methods for the modified Brugge model.

the corresponding well locations are also adjusted, as shown in Fig. 12. We consider a control problem in which inflow-control-valve (ICV) settings of injection and production wells are manipulated to optimize waterflooding over the producing life of the reservoir, which is set to 20 years, or 7,200 days.

The wells penetrate all nine layers with three ICVs in every well (except where bottom ICVs would be placed in the water leg), resulting in a total of 87 controls per timestep. The producing life of the reservoir is divided into 20 time intervals of 1 year (360 days) each, which results in a total of 1,740 controls to be optimized. Water is injected at a constant rate of 3000 m^3/d, and the production wells are operated at a maximum liquid rate of 1200 m^3/d. The bottomhole-pressure limits were 190 bar for injectors and 40 bar for producers. All wells are assumed to be operational from the start of field production. We used an oil price $r_o = \text{USD } 140/m^3$, a water-production cost $r_{wp} = \text{USD } 30/m^3$, and a water-injection cost $r_{wi} = \text{USD } 10/m^3$. The discount rate b was set to zero. Well-index multipliers were used to model the ICVs in the simulator with bounds of 1×10^{-4} and unity. The starting point for the optimization was an initial control vector with values equal to unity. The initial value of σ was equal to 0.1, and we used a fixed ensemble size of 50 samples. Perturbed control values outside the range $[1 \times 10^{-4}, 1]$ were simply reset to their bounds. We used a commercial fully implicit finite-difference black-oil simulator (Eclipse 2011).

Results

Because this model is significantly more complex and larger than the small model discussed previously, we chose to perform only a limited number of optimization experiments. **Fig. 13** shows a comparison between EnOpt (black curve), CMA-EnOpt with only diagonal updates (blue curve), and CMA-EnOpt with block-diagonal updates (red curve). The optimization was run for 50 iterations (i.e., 2,500 reservoir simulations). User-defined parameters and initial strategies were the same for all algorithms. Although neither of the CMA-EnOpt methods had yet converged to an optimum after 50 iterations, CMA-EnOpt with block-diagonal updates clearly outperformed EnOpt, with an increase of 10.5% in objective-function value for the same number of reservoir simulations.

The results shown in Fig. 13 confirm the results obtained from the simple five-spot model. The increased robustness to the choice of the initial covariance matrix is seen to similarly apply for this case. In addition, we observe that the relative increase in objective-function value for the complex Brugge model is significantly higher than for the simple five-spot model.

Robustness to the Initial Covariance Matrix

The choice of a good covariance matrix in EnOpt is a matter of trial and error, which, for the modified Brugge case, requires a significant computational effort. The main idea of CMA-EnOpt is to avoid the trial-and-error procedure and still achieve results of practical importance. We reran the simulations for the modified Brugge example for two additional initial choices of the covariance matrix, the results of which are summarized in **Table 3.** We observe that, irrespective of the initial choice of the covariance matrix, the trend in the results (i.e., CMA-EnOpt performing better than EnOpt) is similar to that in Fig. 13. We also observe that the initial choice of the covariance matrix still plays a role in the optimization, but that we achieve better results for CMA-EnOpt. The EnOpt (block-diagonal) result is from an experiment with a constant covariance matrix with a correlation length of five timesteps for the correlation function described in Zhao et al. (2013). Varying the correlation length may lead to superior or inferior results, which is not known a priori and hence would require trial-and-error experiments. This has not been pursued because the experiments are computationally expensive. For the three different choices of the initial covariance matrix with block-diagonal covariance matrices (i.e., time-correlated covariance matrices), CMA-EnOpt has performed better than EnOpt for two choices of covariance matrices whereas, for the other choice, it is marginally inferior. Note: Because of the significant computational effort required for this model, the effect of different random seeds on the results was not tested. The stopping criterion for the

Initial Sigma	EnOpt	CMA-EnOpt (Diagonal)	EnOpt (Block-Diagonal)	CMA-EnOpt (Block-Diagonal)
1	4.58×10^9	4.97×10^9	5.33×10^9	5.42×10^9
0.1	4.6×10^9	4.63×10^9	5.26×10^9	5.23×10^9
0.01	4.49×10^9	4.45×10^9	4.87×10^9	5.32×10^9

Table 3—The effect of different choices of the initial covariance matrix on the optimization for different methods. All results are expressed as objective-function value in USD.

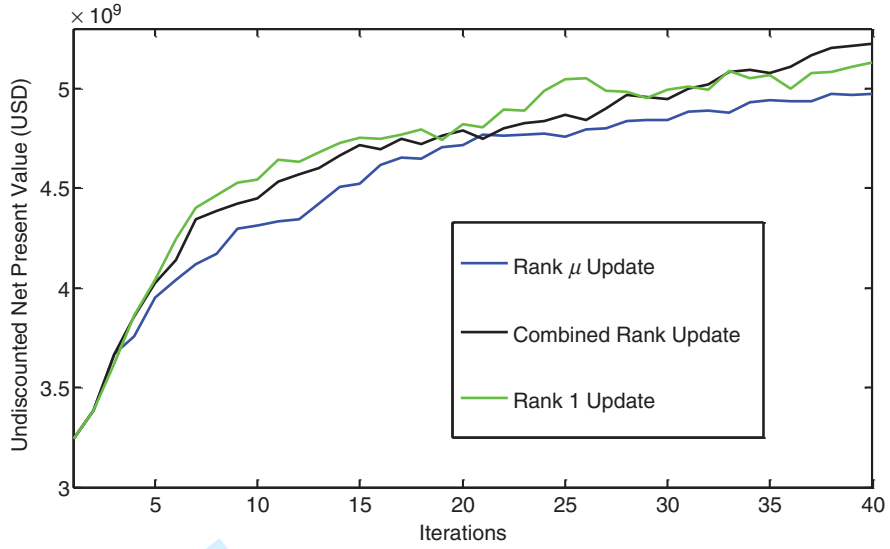

Fig. 14—Comparison of the rank updates for the modified Brugge model.

experiments was the maximum number of iterations, in this case, 40 iterations.

Update Comparison

In the simple five-spot example, the rank-μ update for covariance matrix adaptation-ensemble optimization with a block-diagonal update scheme performed best. We performed a similar experiment on the more-complex modified Brugge case, and observed different results, as illustrated in **Fig. 14.** For this model, the rank-one (green line) update performs better than the rank-μ (blue line) update, whereas after 40 iterations, the combined-rank (black line) update performs best. Note that the rank-μ update has a higher weight (0.20) than the rank-one update (0.05), similar to the values used in the previous example.

Thus, for the Brugge model, the combined-rank update (black line in Fig. 14) eventually achieves the highest objective-function value whereas the rank-μ-only update achieves the worst solution for this case. More experience is needed to arrive at general recommendations on which type of update is most suited for a particular case.

Conclusions

- A comparison between CMA-EnOpt and EnOpt for a simple five-spot model showed consistently (somewhat) higher objective-function values and modest speedups for CMA-EnOpt, depending on the choice of user-defined parameters in both algorithms.
- The major benefit of CMA-EnOpt is its robustness with respect to the initial choice of the covariance matrix. A poor choice of the initial matrix can be detrimental to EnOpt, whereas the CMA-EnOpt performance is near-independent of the initial choice.
- Learning rates are crucial for the success of CMA-EnOpt. For both the simple five-spot model and the modified Brugge model, a 75 to 25% update rule proved to be successful.
- For the simple five-spot model, the methods that explicitly use gradient information (EnOpt and CMA-EnOpt) performed better than the method that does not do so (covariance matrix adaptation-evolutionary strategy). (Not tested for the modified Brugge model).
- A comparison between CMA-EnOpt and EnOpt for the modified Brugge model revealed slightly lower to significantly higher (–1% to +9%) objective-function values, depending on choice of user-defined parameters in both algorithms.
- Updating a block-diagonal (i.e., time-correlated) covariance matrix leads to significant improvements in the results as well as in the efficiency of the algorithm, compared with using a prescribed correlation (smoothing) and compared with updating either diagonal elements only or updating the full matrix.
- The different rank updates play different roles in the success of the optimization; for the simple five-spot model, the rank-μ update performed much better than the rank-one update, with a combined-rank update ending up between these. For the complex Brugge model, however, the rank-μ update performed worst, whereas the combined-rank update performed best. Further experience is needed to arrive at general recommendations.
- Robustness to the choice of the initial covariance matrix and higher objective-function values are the main advantages of CMA-EnOpt over EnOpt.

Nomenclature

b = discount rate
c = learning rate
$\tilde{\mathbf{C}}$ = constant-distribution covariance matrix
$\tilde{\mathbf{C}}_{uu}$ = updated-distribution covariance matrix
$\mathbf{c}_{uJ}$ = ensemble cross-covariance vector
$\mathbf{C}_{uu}$ = ensemble covariance matrix
e = element of evolution path vector
$\mathbf{e}$ = evolution path vector
f = fractional flow
$\mathbf{g}$ = gradient vector
$[\ell]$ = iteration counter
$\mathbf{H}$ = Hessian matrix
I = index
$\mathbf{j}$ = vector of mean-shifted objective-function values
J = objective-function value
$\overline{J}$ = mean objective-function value
k = timestep counter
K = total number of timesteps
M = number of ensemble members
N = number of control variables
p = pressure
q = flow rate
r = price per unit volume
t = time
u = control variable
$\mathbf{u}$ = vector of control variables
$\overline{\mathbf{u}}$ = ensemble mean
$\tilde{\mathbf{u}}$ = distribution mean
$\mathbf{U}$ = matrix of ensemble mean-shifted control vectors
$\tilde{\mathbf{U}}$ = matrix of distribution mean-shifted control vectors
α = step size
β = dimensionless inflow-control-valve opening (well-index multiplier)

μ = number of "best" ensemble members
τ = reference time for discounting

Subscripts

gb = gridblock
o = oil
w = water
wp = produced water
wi = injected water

Acknowledgments

This research was carried out within the context of the ISAPP Knowledge Centre. ISAPP (Integrated Systems Approach to Petroleum Production) is a joint project of TNO, Delft University of Technology, Eni, Statoil, and Petrobras. We acknowledge Schlumberger for providing multiple academic Eclipse licenses for this work.

References

Arnold, D.V. and Hansen, N. 2010. Active Covariance Matrix Adaptation for the (1+1)-CMA-ES. In *Proc., 12th Annual Conference on Genetic and Evolutionary Computation. GECCO'10.* Portland, Oregon, USA, 7–11 July.

Bouzarkouna, Z., Ding, Y.D., and Auger, A. 2011. Partially Separated Meta-models With Evolution Strategies for Well Placement Optimization. Presented at the SPE EUROPEC/EAGE Annual Conference and Exhibition, Vienna, Austria, 23–26 May. SPE-143292-MS. http://dx.doi.org/10.2118/143292-MS.

Chaudhri, M.M., Phale, H.A., Liu, N. et al. 2009. An Improved Approach for Ensemble-Based Production Optimization. Presented at the SPE Western Regional Meeting, San Jose, California, 24–26 March. SPE-121305-MS. http://dx.doi.org/10.2118/121305-MS.

Chen, Y. 2008. *Efficient Ensemble-Based Reservoir Management*. PhD Thesis, University of Oklahoma, USA.

Chen, Y. and Oliver, D.S. 2010. Ensemble-Based Closed-Loop Optimization Applied to Brugge Field. *SPE Res Eval & Eng* **13** (1): 56–71. SPE-118926-PA. http://dx.doi.org/10.2118/118926-PA.

Chen, Y. and Oliver, D.S. 2012. Localization of Ensemble-Based Control-Setting Updates for Production Optimization. *SPE J.* **17** (1): 122–136. SPE-125042-PA. http://dx.doi.org/10.2118/125042-PA.

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.

Conn, A.R., Scheinberg, S., and Vicente, L.N. 2009. *Introduction to Derivative-Free Optimization*. Philadelphia: Siam.

Dehdari, V. and Oliver, D.S. 2012: Sequential Quadratic Programming for Solving Constrained Production Optimization–Case Study From Brugge Field. *SPE J.* **17** (3): 874–884. SPE-141589-PA. http://dx.doi.org/10.2118/141589-PA.

Ding, Y. 2008. Optimization of Well Placement Using Evolutionary Algorithms. Presented at the Europec/EAGE Conference and Exhibition, Rome, Italy, 9–12 June. SPE-113525-MS. http://dx.doi.org/10.2118/113525-MS.

Do, S.T. and Reynolds, A.C. 2013: Theoretical Connections Between Optimization Algorithms Based on an Approximate Gradient. *Computational Geosci.* **17** (6): 959–973. http://dx.doi.org/10.1007/s10596-013-9368-9.

Eclipse. 2011. http://www.slb.com/services/software/reseng/eclipse.

Hansen, N. 2006. The CMA Evolution Strategy: A Comparing Review. In *Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms*, ed. J.A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, 75–102, Springer.

Hansen, N. 2011. The CMA Evolution Strategy: A Tutorial. http://www.lri.fr/~hansen/cmatutorial110628.pdf.

Hansen, N. and Ostermeier, A. 1996. Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In *Proc., 1996 IEEE International Conference on Evolutionary Computation*.

Hansen, N. and Ostermeier, A. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* **9** (2): 159–195.

Jansen, J.D. 2011. Adjoint-Based Optimization of Multi-Phase Flow Through Porous Media—A Review. *Computers & Fluids* **46** (1): 40–51. http://dx.doi.org/10.1016/j.compfluid.2010.09.039.

Jansen, J.D. 2013. *A Systems Description of Flow Through Porous Media*. SpringerBriefs in Earth Sciences, New York: Springer. http://dx.doi.org/10.1007/978-3-319-00260-6.

Leeuwenburgh, O., Egberts, P.J.P., and Abbink, O.A. 2010. Ensemble Methods for Reservoir Life-Cycle Optimization and Well Placement. Presented at the SPE/DGS Saudi Arabia Section Technical Symposium and Exhibition, Al-Khobar, Saudi Arabia, 4–7 April. SPE-136916-MS. http://dx.doi.org/10.2118/136916-MS.

Lorentzen, R.J., Berg, A., Naevdal, G. et al. 2006. A New Approach for Dynamic Optimization of Waterflooding Problems. Presented at the Intelligent Energy Conference and Exhibition, Amsterdam, The Netherlands, 11–13 April. SPE-99690-MS. http://dx.doi.org/10.2118/99690-MS.

Luenberger, D.G. and Ye, Y. 2010. *Linear and Nonlinear Programming, third edition*. New York: Springer.

Nocedal, J. and Wright, S.J. 2006. *Numerical Optimzation,* second edition. New York: Springer.

Nwaozo, J. 2006. *Dynamic Optimization of a Wateflooding Reservoir*. MSc Thesis, University of Oklahoma, USA.

Oliveira, D.F. and Reynolds, A.C. 2014. An Adaptive Hierarchical Algorithm for Estimation of Optimal Well Controls. *SPE J.* **9** (4): 391–402. SPE-163645-PA. http://dx.doi.org/10.2118/163645-PA.

Pajonk, O., Schulze-Riegert, R., Krosche, M. et al. 2011. Ensemble-Based Water Flooding Optimization Applied to Mature Fields. Presented at the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain, 25–28 September. SPE-142621-MS. http://dx.doi.org/10.2118/142621-MS.

Peters, L., Arts, R.J., Brouwer, G.K. et al. 2010. Results of the Brugge Benchmark Study for Flooding Optimization and History Matching. *SPE Res Eval & Eng* **13** (3): 391–405. SPE-119094-PA. http://dx.doi.org/10.2118/119094-PA.

Peters, E., Chen, Y., Leeuwenburgh, O. et al. 2013. Extended Brugge Benchmark Case for History Matching and Water Flooding Optimization. *Computers & Geosci.* **50:** 16–24. http://dx.doi.org/10.1016/j.cageo.2012.07.018.

Ros, R. and Hansen, N. 2008. A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity. *Proc., 10th International Conference on Parallel Problem Solving From Nature–PPSN X Workshop, Dortman, Germany,* September, 296–305.

Schulze-Riegert, R., Bagheri, M., Krosche, M. et al. 2011. Multiple-Objective Optimization Applied to Well Path Design Under Geological Uncertainty. Presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 21–23 February. SPE-141712-MS. http://dx.doi.org/10.2118/141712-MS.

Strang, G. 2006. *Linear Algebra and Its Applications*, fourth edition. Pacific Grove: Thomson Brooks/Cole.

Su, H.-J. and Oliver, D.S. 2010. Smart Well Production Optimization Using an Ensemble-Based Method. *SPE Res Eval & Eng* **13** (6): 884–892. SPE-126072-PA. http://dx.doi.org/10.2118/126072-PA.

Zandvliet, M.J., Bosgra, O.H., Van den Hof, P.M.J. et al. 2007. Bang-Bang Control and Singular Arcs in Reservoir Flooding. *J. Petrol. Sci. & Eng.* **58:** 186–200. http://dx.doi.org/10.1016/j.petrol.2006.12.008.

Zhao, H., Chen, C., Do, S.T. et al. 2013. Maximization of a Dynamic Quadratic Interpolation Model for Production Optimization. *SPE J.* **18** (6): 1012–1025. SPE-141317-PA. http://dx.doi.org/10.2118/141317-PA.

Rahul-Mark Fonseca is a PhD degree candidate at Delft University of Technology working on ensemble-based production optimization and reservoir management. He holds an MS degree in petroleum engineering from Delft University of Technology.

Olwijn Leeuwenburgh is a research scientist at TNO. He previously worked for several years in the field of ocean and climate simulation and forecasting with a focus on data assimilation. Leeuwenburgh's current interests include mathematical

aspects of reservoir modeling, especially ensemble methods for assisted history matching, proxy modeling, and life-cycle optimization, and the integration of disciplines with the aim of optimal reservoir management. He holds an MSc degree in geodetic engineering from Delft University of Technology and a PhD degree in geophysics from the University of Copenhagen.

Paul Van den Hof is Professor of Control Systems in the Department of Electrical Engineering at Eindhoven University of Technology, Eindhoven, the Netherlands, and scientific director of the Dutch Institute of Systems and Control. His current research interests are in system identification and model-based control and optimization, with applications in several technology domains, including petroleum-reservoir-engineering systems.

Jan-Dirk Jansen is Professor of Reservoir Systems and Control, and head of the Department of Geoscience and Engineering at Delft University of Technology, the Netherlands. His current research interests are the use of systems and control theory for production optimization and reservoir management. Previously, Jansen worked for Shell International in research and operations. He holds MSc and PhD degrees from Delft University of Technology.