Application of derivative-free methodologies to generally constrained oil production optimisation problems

David Echeverría Ciaurri*, Obiajulu J. Isebor and Louis J. Durlofsky

Department of Energy Resources Engineering,
Stanford University,
367 Panama Street, Stanford, CA 94305-2220, USA
E-mail: echeverr@stanford.edu
E-mail: oisebor@stanford.edu
E-mail: lou@stanford.edu
*Corresponding author

Abstract: Oil production optimisation involves the determination of optimum well controls (well pressures, injection rates) to maximise an objective function such as net present value. These problems typically include physical and economic restrictions, which introduce general constraints into the optimisations. Cost function and constraint evaluations entail calls to a reservoir flow simulator. In many situations, gradient information is not available, so derivative-free (non-invasive, black-box) optimisation methods are of interest. This work entails a comparative study of several derivative-free methods applied to generally constrained production optimisation problems. The methods considered include generalised pattern search, Hooke-Jeeves direct search, and a genetic algorithm. Penalty function and filter-based methods are applied for constraint handling. Numerical results for optimisation problems of varying complexity highlight the relative advantages and disadvantages of these procedures. Several combinations of approaches are shown to perform well for the generally constrained cases considered.

Keywords: non-linear programming; derivative-free optimisation; oil production optimisation; reservoir simulation; closed-loop reservoir modelling.

Reference to this paper should be made as follows: Echeverría Ciaurri, D., Isebor, O.J. and Durlofsky, L.J. (2011) 'Application of derivative-free methodologies to generally constrained oil production optimisation problems', *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 2, No. 2, pp.134–161.

Biographical notes: David Echeverría Ciaurri received his Telecommunication Engineering degree and his MS in Applied Mathematics from the University of Zaragoza (Spain) in 1998 and 2002, respectively. In 2007 he obtained his PhD in Numerical Analysis from the University of Amsterdam (UvA). He has worked at the Dutch National Research Institute for Mathematics and Computer Science (CWI) and at the Lawrence Livermore National Laboratory (LLNL). Since 2007 he has been affiliated with the Smart Fields Consortium in the Department of Energy Resources Engineering at Stanford University. His research interests are efficient optimisation and modelling, and signal processing.

Obiajulu J. Isebor received his BS in Mechanical Engineering in 2007 from Idaho State University. In 2009, he obtained his MS in Petroleum Engineering from Stanford University, where he is currently pursuing his PhD in Energy Resources Engineering. His research interests include reservoir modelling and simulation, derivative-free optimisation and mixed-variable optimisation.

Louis J. Durlofsky is the Otto N. Miller Professor and Chairman in the Department of Energy Resources Engineering at Stanford University. He was previously affiliated with Chevron Energy Technology Company. He holds a BS from Penn State and MS and PhD from the Massachusetts Institute of Technology, all in Chemical Engineering. His research interests include optimisation for subsurface flow problems, upscaling of detailed geological models for flow simulation, reduced-order modelling, geological carbon sequestration, and general reservoir simulation.

1 Introduction

Oil and natural gas together currently provide about 60% of global primary energy, and forecasts indicate that these fossil fuels will continue to be significant contributors to global energy for decades to come. For this reason, significant effort in the oil and gas industry is being expended to optimise field operations in order to maximise oil recovery and/or profit. This has led to a great deal of interest in the idea of efficient closed-loop reservoir management (Jansen et al., 2005), of which production optimisation is a key component.

In the context of reservoir management, production optimisation entails maximising a particular objective function, such as cumulative oil produced or net present value (NPV), or minimising an objective function such as total water injected, by finding the optimal set of control variables. In this case the control variables correspond to the sequence (in time) of the well rates or well bottom-hole pressures (BHPs). Since the relationship between the reservoir dynamics and the control variables is in general non-linear, searching for the optimal set of controls is a very challenging task. In addition, the problem must usually be solved subject to operational constraints, such as maximum and minimum BHPs, maximum field water injection rates, maximum water cut (fraction of water in the produced fluid), etc. Thus, the problem is typically a non-linearly constrained optimisation.

One approach for dealing with non-linear constraints in production optimisation applications is to force each simulation to comply, if possible, with these operational constraints. This involves modifying the control variables during the course of the simulation. In practice, this is typically accomplished through use of heuristics, which are in general not optimal choices (this heuristic approach is used in commercial reservoir simulation codes; see, e.g., Schlumberger, 2008). Additionally, this approach may not be effective in situations where it is difficult to obtain controls that provide feasible solutions, such as when a maximum field water injection rate and a minimum field oil production target are imposed simultaneously.

Our goal in this work is to apply several promising derivative-free optimisation procedures to generally constrained production optimisation problems. Such problems can also be addressed using gradient-based methods. Derivative information can be estimated numerically in a straightforward manner, but this computation can be expensive and may lack robustness (e.g., in finite differencing, selection of the perturbation size and/or simulation tolerances can be problematic). The use of efficient adjoint-based techniques for computing the required gradients greatly reduces the computational effort, and such approaches have been applied for production optimisation by, e.g., Ramirez (1987), Brouwer and Jansen (2004), and Sarma et al. (2006). These procedures require extraction of information from the reservoir simulator during the course of the computations, and therefore are only feasible with full access to, and detailed knowledge of, the simulator source code. Even when such access exists, the effort associated with the development and maintenance of the adjoint code is significant. We note finally that most gradient-based strategies cannot avoid being trapped in local optima.

Derivative-free optimisation algorithms, by contrast, are straightforward to implement and most of them parallelise very naturally. These methods do not require the explicit calculation of gradients and use just the values obtained from function evaluations. These approaches can be divided into deterministic [e.g., generalised pattern search (GPS)] and stochastic (e.g., genetic algorithm) procedures. The stochastic component is usually included as a means for dealing with multiple local optima. Almost exclusively, deterministic algorithms in practice aim at local optimisation. The general performance of derivative-free methods depends strongly on the number of optimisation variables considered, and these methods have been used successfully in situations when this number is less than a few hundred.

Gradient-free optimisation methods have been applied in a number of areas. Examples include molecular geometry (Meza and Martinez, 1994), aircraft design (Booker et al., 1998; Marsden et al., 2007), hydrodynamics (Duvigneau and Visonneau, 2004; Fowler et al., 2008) and medicine (Oeuvray and Bierlaire, 2007; Marsden et al., 2008). Within the oil industry, most of the derivative-free schemes applied to date have been of the stochastic type (Harding et al., 1996; Cullick et al., 2003; Artus et al., 2006; Almeida et al., 2007), sometimes hybridised with a deterministic search (Bittencourt, 1997). Examples of purely deterministic strategies can be found in Carroll III (1990), and Echeverria and Mukerji (2009). It should be noted that none of these studies addressed non-linear constraint handling.

The derivative-free methods considered in this study are two pattern search schemes, namely GPS (Torczon, 1997; Audet and Dennis, 2002) and Hooke-Jeeves direct search (HJDS) (Hooke and Jeeves, 1961), and a genetic algorithm (Goldberg, 1989). Many existing derivative-free implementations can readily handle problems with only bound and linear constraints. As stated earlier, the production optimisation problem is a generally constrained problem, and it is common to have non-linear constraints. We will investigate various techniques for dealing with these general constraints including penalty functions (Nocedal and Wright, 2006), filter methods (Fletcher et al., 2006; Nocedal and Wright, 2006), and parameterless penalty functions for genetic algorithms (Deb, 2000).

The paper is structured as follows. In Section 2, we present the basic equations governing the flow of oil and water in subsurface formations as well as the optimisation problem statement. Next, in Section 3, all the optimisers considered will be briefly described for unconstrained optimisation problems. The extension to non-linearly constrained cases will be presented in Section 4. The schemes are applied in Section 5 to three production optimisation cases of varying complexity. We conclude the paper with some conclusions and recommendations.

2 Governing equations and problem statement

The equations governing the flow of oil and water are formed by combining statements of mass conservation for the two components with an equation that relates phase fluxes to pressure gradient (this relationship is referred to as Darcy's law; see, e.g., Aziz and Settari, 1979; Gerritsen and Durlofsky, 2005). To simplify the current discussion, we neglect the effects of gravity, compressibility and capillary pressure (gravity and compressibility are included in the simulations performed in this work, though capillary pressure is neglected because its effects are insignificant for the cases considered). The flow problem can be expressed in terms of the so-called pressure and saturation equations, which are as follows:

$$\nabla \cdot [\lambda(S)\mathbf{k}(\mathbf{x}) \cdot \nabla p] = q, \tag{1}$$

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot [\mathbf{v} f(S)] = -q_w. \tag{2}$$

Here p designates pressure, $\mathbf{k}$ is the absolute permeability tensor (essentially a flow conductivity) of the rock, $\mathbf{x}$ is spatial location, S is water saturation (volume fraction of water within the pore space), λ is the total mobility (which accounts for the impact of saturation on flow rate), q is the total (oil + water) source term, ϕ is rock porosity (volume fraction of the void space), t is time, $\mathbf{v}$ is the total Darcy velocity, which is the sum of the oil and water Darcy velocities (Darcy velocity of a phase is defined as volumetric phase flow rate divided by area; thus it is essentially a superficial velocity), f is the fractional flow of water (water flux divided by total flux) and q_w is the water source term. The $\lambda(S)$ and $f(S)$ flow functions are generally determined experimentally using core samples. The total Darcy velocity is related to pressure gradient via $\mathbf{v} = -\lambda \mathbf{k} \cdot \nabla p$. Additional complexities, and another equation (the gas conservation equation), appear for oil-gas-water problems.

In reservoir simulators used in practice, the governing equations are discretised using a finite volume numerical procedure. In practical applications, simulation models may contain $O(10^5 \sim 10^6)$ grid blocks and may require several hundred time steps (the systems considered here are somewhat smaller). In addition, the discrete system of equations is non-linear and is solved using a Newton-Raphson procedure. Thus, the evaluation of reservoir performance is computationally demanding. See, e.g., Aziz and Settari (1979), and Gerritsen and Durlofsky (2005) for more details on the governing equations and numerical treatments. In this work we apply Stanford's general purpose research simulator (GPRS; Cao, 2002; Jiang, 2007) for all simulations.

The production optimisation problem can be formally stated as:

$$\min_{\mathbf{u} \in \Omega \subset \mathbb{R}^n} J(\mathbf{u}) \quad \text{subject to} \quad \mathbf{c}(\mathbf{u}) \leq 0, \tag{3}$$

where $J(\mathbf{u})$ is the objective function (e.g., negative of net present value ($-$NPV) or cumulative water produced), $\mathbf{u} \in \mathbb{R}^n$ is the vector of control variables (e.g., sequence of bottom-hole pressures or BHPs for each well), and $\mathbf{c}:\mathbb{R}^n \rightarrow \mathbb{R}^m$ represents the non-linear constraints in the problem. Bound and linear constraints are included in the set $\Omega \subset \mathbb{R}^n$. The objective function and constraint variables are computed using the output from the reservoir simulator, which renders function evaluations expensive.

In the production optimisation cases presented here, the problem involves the maximisation of (undiscounted) net present value (NPV) by adjusting the BHPs of water injection and production wells. The well flow rates could also be used as the optimisation variables. The specification of either BHPs or flow rates impacts the source terms q and q_w in (1) and (2), which drive the flow. Within the reservoir simulator, a well model is used to relate well flow rates to BHPs – see Cao (2002); Jiang (2007) for details. The objective function we seek to minimise is

$$J(\mathbf{u}) = -\mathrm{NPV}(\mathbf{u}) = -r_o Q_o(\mathbf{u}) + c_{wp} Q_{wp}(\mathbf{u}) + c_{wi} Q_{wi}(\mathbf{u}), \tag{4}$$

where r_o is the price of oil (\$/STB, where 'STB' refers to stock tank barrel and 1 STB = 0.1590 m^3), c_{wp} and c_{wi} are the costs of produced and injected water (\$/STB), respectively, and Q_o, Q_{wp} and Q_{wi} are the cumulative oil production, water production and water injection (STB) obtained from the simulator.

3 Derivative-free optimisation

In this section we describe the derivative-free optimisation methods considered in this work – GPS, HJDS, and a genetic algorithm. Constraint handling procedures are described below in Section 4.

3.1 Pattern search methods

Pattern search methods are optimisation procedures that evaluate the cost function in a stencil-based fashion. This stencil is sequentially modified as iterations proceed. The recent popularity of these schemes is due in part to the development of mathematically sound convergence theory (Kolda et al., 2003; Conn et al., 2009) and to the widespread availability of parallel computing resources.

3.1.1 Generalised pattern search

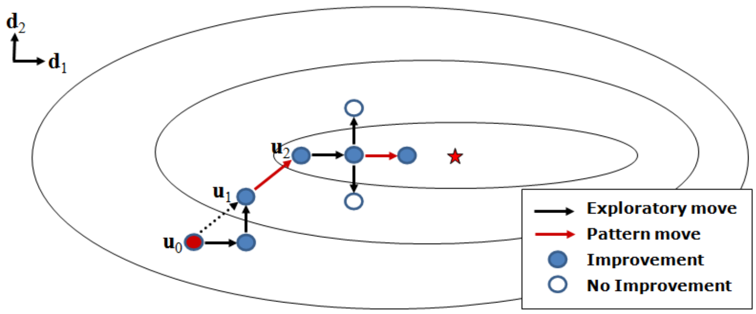
GPS (Torczon, 1997; Audet and Dennis, 2002) refers to a whole family of optimisation methods. In essence, GPS relies on polling (local exploration of the cost function; see Audet and Dennis, 2002), and it works as follows. At any particular iteration a stencil is centred at the current solution. The stencil comprises a set of directions such that at least one is a descent direction (this is called a generating set; see Kolda et al., 2003). If some of the points in the stencil represent an improvement in the cost function, the stencil is moved to one of these new solutions. Otherwise, the stencil size is decreased. The optimisation progresses until some stopping criterion is satisfied (typically, a minimum stencil size). In GPS the stencil orientation remains the same at each iteration, and typically induces a coordinate or compass search. GPS can be further generalised by polling in an asymptotically dense set of directions (this set varies with iteration). The resulting algorithm is the mesh adaptive direct search (MADS; Audet and Dennis, 2006).

The GPS method parallelises naturally since, at a particular iteration, the objective function evaluations at the polling points can be accomplished in a distributed fashion. The method typically requires on the order of n function evaluations per iteration (where n is the number of optimisation variables).

3.1.2 Hooke-Jeeves direct search

The HJDS (Hooke and Jeeves, 1961) is another pattern search method. HJDS is based on two types of moves: exploratory and pattern. These moves are illustrated in Figure 1 for some optimisation iterations in $\mathbb{R}^2$.

Figure 1 Illustration of exploratory and pattern moves in HJDS (see online version for colours)



Note: The star represents the optimum.

The iteration starts with a base point $\mathbf{u}_0$ and a given step size. During the exploratory move, the objective function is evaluated at successive perturbations of the base point in the search (coordinate) directions. All the directions are polled sequentially and in an opportunistic way. This means that if $\mathbf{d}_1 \in \mathbb{R}^n$ is the first search direction, the first function evaluation is at $\mathbf{u}_0 + \mathbf{d}_1$. If this represents an improvement in the cost function, the next point polled will be, assuming $n > 1$, $\mathbf{u}_0 + \mathbf{d}_1 + \mathbf{d}_2$, where $\mathbf{d}_2$ is the second search direction. Otherwise the point $\mathbf{u}_0 - \mathbf{d}_1$ is polled. Upon success at this last point, the search proceeds with $\mathbf{u}_0 - \mathbf{d}_1 + \mathbf{d}_2$, and alternatively with $\mathbf{u}_0 + \mathbf{d}_2$. The exploration continues until all search directions have been considered. If after the exploratory step no improvement in the cost function is found, the step size is reduced. Otherwise, a new point $\mathbf{u}_1$ is obtained, but instead of centring another exploratory move at $\mathbf{u}_1$, the algorithm performs the pattern move, which is an aggressive step that moves further in the underlying successful direction. After the pattern move, the next polling centre $\mathbf{u}_2$ is set at $\mathbf{u}_0 + 2(\mathbf{u}_1 - \mathbf{u}_0)$. If the exploratory move at $\mathbf{u}_2$ fails to improve upon $\mathbf{u}_1$, a new polling is performed around $\mathbf{u}_1$. If this again yields no cost function decrease, the step size is reduced, keeping the polling centre at $\mathbf{u}_1$.

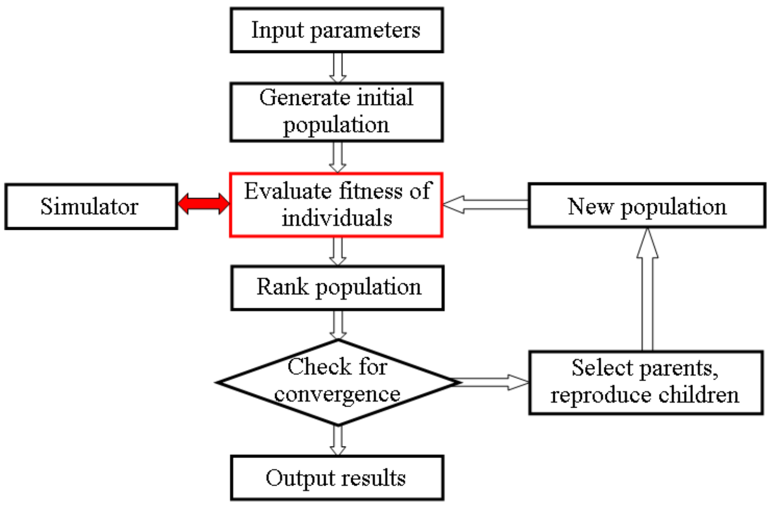
Notice the clear serial nature of the algorithm. This makes HJDS a reasonable pattern search option when distributed computing resources are not available. Because of the pattern move, HJDS may also be beneficial in situations where an optimum is far away from the initial guess.

The first stages of pattern search techniques use a relatively large stencil size. This feature enables them to avoid some local minima, and this may provide these algorithms with some robustness against noisy cost functions. Pattern search methods (and genetic algorithms as well) can be accelerated through the use of inexpensive surrogates, which can be highly useful given the large number of objective function evaluations that are typically required.

3.2 Genetic algorithms

Genetic algorithms (GAs) are stochastic search techniques that are based on the theory of natural selection. These algorithms perform a global search by first generating randomly a set of possible solutions (a population) and then evaluating the fitness (i.e., objective function) of all the individuals in this population. Individuals are then ranked, after which certain operators (typically selection, crossover and mutation) are applied to generate a new population. The selection operator chooses as parents the individuals with the best objective function values. After selection, the crossover operator combines the parents to produce children (next population of individuals). During mutation, a specific part (e.g., bit or element) of an individual is probabilistically modified. The general workflow followed in this work for optimisation using a genetic algorithm is shown in Figure 2. Refer to Goldberg (1989) for a more detailed description of genetic algorithms.

Figure 2 General workflow for optimisation using a genetic algorithm (see online version for colours)



One of the most important parameters in a GA is the population size. With a proper population size, GAs can explore complex non-smooth search spaces with multiple local optima and may as a result identify promising regions in the search space. This often requires a large population size and thus, a correspondingly high number of function evaluations. However, as in GPS, the simulations required for all of the individuals in the population can be readily performed in a distributed manner.

4 Generally constrained derivative-free optimisation

We now describe non-linear constraint handling techniques that can be combined with the optimisation methods presented in Section 3.

4.1 Penalty functions

The penalty function method (cf. Nocedal and Wright, 2006) for general optimisation constraints involves modifying the objective function with a penalty term that depends on the constraint violation $h : \mathbb{R}^n \rightarrow \mathbb{R}$. The original optimisation problem in (3) is thus modified as follows:

$$\min_{\mathbf{u}\in\Omega\subset\mathbb{R}^n} \; J\left(\mathbf{u}\right) + \rho\, h\left(\mathbf{u}\right), \tag{5}$$

where $\rho > 0$ is a penalty parameter. In the examples below, the modified optimisation problem still has (bound) constraints, but they are straightforward to handle.

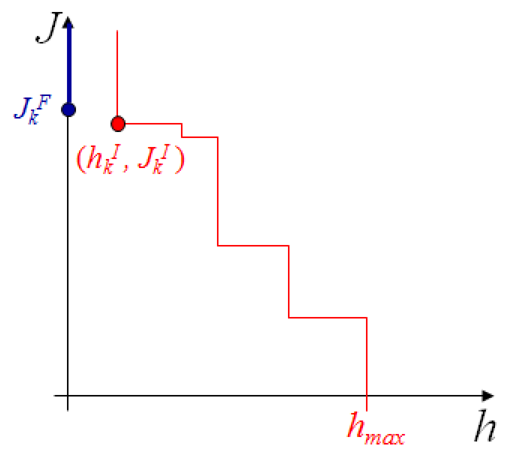
If the penalty parameter is iteratively increased (tending to infinity), the solution of (5) converges to that of the original problem in (3). However, in certain cases, a finite (and fixed) value of the penalty parameter ρ also yields the correct solution (this is the so-called *exact* penalty; see Nocedal and Wright, 2006). For exact penalties, the modified cost function is not smooth around the solution (Nocedal and Wright, 2006), and thus, the corresponding optimisation problem can be significantly more involved than that in (5). Our approach here is based on sequentially increasing the penalty parameter, as described in Nocedal and Wright (2006). This avoids a potentially expensive tuning process associated with the search for the exact penalty. However, the eventual high value of the penalty parameter can lead to numerical complications.

In this work, and within the penalty function context, we apply $h\left(\mathbf{u}\right) = ||\mathbf{c}^{+}\left(\mathbf{u}\right)||_1$ and $h\left(\mathbf{u}\right) = ||\mathbf{c}^{+}\left(\mathbf{u}\right)||_2^2$, with $\mathbf{c}^{+} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ defined as $c_i^{+}\left(\mathbf{u}\right) = \max\{0, c_i\left(\mathbf{u}\right)\}$, for the examples in Section 5.2 and in Section 5.3, respectively (normalising the constraints can be beneficial since they are all weighted equally in the penalty term). The first choice for h is consistent with Griffin and Kolda (2007), where non-quadratic penalties have been suggested for pattern search techniques. Since in our experiments in Section 5.2 quadratic and non-quadratic penalties performed similarly, we only tested a quadratic penalty function in the case in Section 5.3. The optimisations presented in Griffin and Kolda (2007) are, however, much simpler than those considered here, so the recommendations given might not be applicable to our problems. In future research, it will be useful to explore further the performance of different penalty functions for practical optimisation problems.

4.2 Filter method

The penalty function approach is easier to implement than other more sophisticated constraint handling techniques (e.g., augmented Lagrangian methods; see Nocedal and Wright, 2006) and can be used successfully for many applications. However, as discussed above, the sequential increase of the penalty parameter can be problematic and/or time consuming. Filter methods (Fletcher et al., 2006; Nocedal and Wright, 2006) provide a somewhat cleaner means of handling general constraints. Using filters, the original problem (3) is viewed as a bi-objective optimisation: besides minimising the cost function $J(\mathbf{u})$, one also seeks to reduce the constraint violation $h(\mathbf{u})$. In this work the constraint violation associated to the filter method in all cases is $h(\mathbf{u}) = ||\mathbf{c}^+(\mathbf{u})||_2^2$. The concept of dominance, crucial in multi-objective optimisation, is defined as follows: the point $\mathbf{u}_1 \in \mathbb{R}^n$ dominates $\mathbf{u}_2 \in \mathbb{R}^n$ if and only if either $J(\mathbf{u}_1) \leq J(\mathbf{u}_2)$ and $h(\mathbf{u}_1) < h(\mathbf{u}_2)$, or $J(\mathbf{u}_1) < J(\mathbf{u}_2)$ and $h(\mathbf{u}_1) \leq h(\mathbf{u}_2)$. A filter is a set of pairs $(h(\mathbf{u}), J(\mathbf{u}))$, such that no pair dominates another pair. In practice, a maximum allowable constraint violation $h_{\max}$ is specified. This is accomplished by introducing the pair $(h_{\max}, -\infty)$ in the filter. An idealised filter (at iteration k) is shown in Figure 3.

Figure 3 An idealised (pattern search) filter at iteration k (see online version for colours)



A filter can be understood as essentially an add-on for an optimisation procedure. The intermediate solutions proposed by the optimisation algorithm at a given iteration are accepted if they are not dominated by any point in the filter. The filter is updated at each iteration based on all the points evaluated by the optimiser. We reiterate that the optimisation search is enriched by considering infeasible points, though the ultimate solution is intended to be feasible (or very nearly so). Filter methods are often observed to lead to faster convergence than methods that rely only on feasible iterates.

Pattern search optimisation techniques have been previously combined with filters (Audet and Dennis, 2004). In HJDS, the filter establishes the acceptance criterion for each (unique) new solution. For schemes where, in each iteration, multiple solutions can be accepted by the filter (such as in GPS), the new polling centre must be selected from the set of validated points. When the filter is not updated in a particular iteration (and thus, the best feasible point is not improved), the pattern size is decreased. As in Audet and Dennis (2004), when we combine GPS with a filter, the polling centre at a given iteration will be the feasible point with lowest cost function or, if no feasible points remain, it will be the infeasible point with lowest constraint violation. These two points, $(0, J_k^{\mathrm{F}})$ and $(h_k^{\mathrm{I}}, J_k^{\mathrm{I}})$, respectively, are shown in Figure 3 (it is assumed that both points have just been accepted by the filter, and thus, it makes sense to use one of them as the new polling centre). Refer to Audet and Dennis (2004), and Abramson (2007) for more details on pattern search filter methods.

4.3 Parameterless penalty function for genetic algorithms

Within the context of genetic algorithms, there are alternatives to penalty functions that handle general constraints through use of approaches borrowed from multi-objective optimisation. In this work we apply the parameterless penalty method described in Deb (2000). Other approaches, such as that of Surry et al. (1995), could also be considered. The parameterless penalty method for GAs is based on the tournament selection operator. Within tournament selection, two individuals are compared in terms of their cost function, and the one with lowest value is kept for the next generation. The parameterless penalty method modifies the cost function for infeasible individuals within a given population to $J_{\max} + h(\mathbf{u})$, where $J_{\max}$ is the cost function value corresponding to the worst feasible individual in the population. The constraint violation considered here is $h(\mathbf{u}) = ||\mathbf{c}^{+}(\mathbf{u})||_1$. The cost function used for feasible individuals remains $J(\mathbf{u})$. Thus, when two feasible (infeasible) solutions are compared, the one with the lowest cost function (constraint violation) is selected. A feasible individual always prevails over an infeasible one. In this algorithm, some individuals are optimised based on $J(\mathbf{u})$ and others with respect to $h(\mathbf{u})$, with a trend that favours feasibility. In contrast to filter methods, a slightly infeasible individual with a low cost function can in some cases be discarded (e.g., in a tournament against any feasible solution).

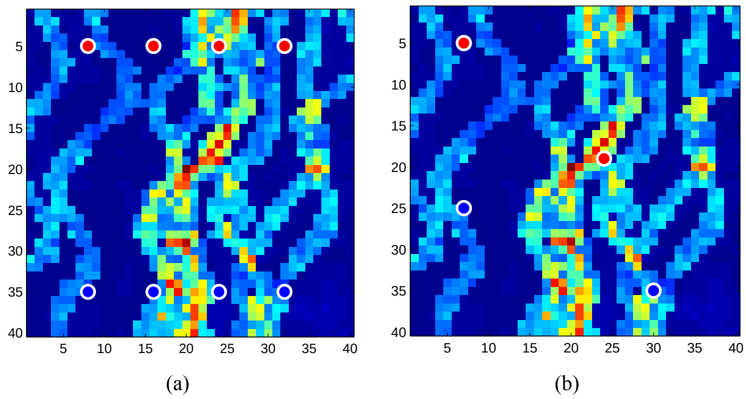
5 Oil production optimisation cases

The methods described in the preceding sections will now be applied to synthetic (though challenging) production optimisation problems with general constraints. These cases resemble actual reservoir models in terms of their property (permeability and porosity) variation and operating specifications and constraints, though they lack some complications (e.g., faults) and contain fewer grid blocks and wells than most models used in practice. A gradient-based method, sequential quadratic programming (SQP; see Nocedal and Wright, 2006), is also considered to enable comparisons between the various approaches. The optimisation variables are, in all cases, well BHPs. The first case has 80 optimisation variables and only bound constraints, while the other two examples are generally constrained production optimisations. The second case involves 20 optimisation variables and five general constraints and the third example deals with 100 optimisation variables and 12 non-linear constraints. In all cases we consider two-phase oil-water systems.

5.1 Production optimisation with bound constraints

The reservoir in this case is represented on a two-dimensional 40×40 grid. Eight wells (four injectors and four producers) drive the flow. The wells are arranged in a line drive pattern as shown in Figure 4(a). The production time frame is 3,000 days. The BHPs of each well are updated every 300 days (there are a total of ten control intervals). The BHP is held constant over each control interval. The total number of optimisation variables in this problem is therefore 80. The main problem parameters are shown in Table 1. The optimisation bounds are indicated by means of the injector and producer BHP ranges. More details on the problem setup can be found in Isebor (2009).

Figure 4 Well configurations and geological model considered in (a) Section 5.1 and (b) Section 5.2 (see online version for colours)



Notes: Injection and production wells are represented as blue and red circles, respectively. Grid blocks are coloured to indicate value of permeability (red and blue indicate high and low permeability, respectively).

Table 1 Optimisation parameters for the cases in Section 5

	Section 5.1	*Section 5.2*	*Section 5.3*
r_o	\$80/STB	\$50/STB	\$50/STB
c_{wp}	\$36/STB	\$10/STB	\$10/STB
c_{wi}	\$18/STB	\$5/STB	\$5/STB
Injector BHP range	6,000–9,000 psi	6,000–10,000 psi	6,500–12,000 psi
Producer BHP range	2,500–4,500 psi	500–4,500 psi	500–5,500 psi
Maximum total water injection rate	-	1,000 STB/day	15,000 STB/day
Minimum total oil production rate	-	450 STB/day	3,000 STB/day
Maximum total fluid production rate	-	2,000 STB/day	10,000 STB/day
Maximum water cut at any production well	-	0.5	0.7

The specifications for the optimisers applied to this problem are as follows. For the SQP algorithm, the gradients are estimated numerically by second-order finite differencing (with a perturbation size of 0.01 psi). Although much less efficient than adjoint procedures, numerical derivatives, when implemented in a distributed computing framework or when combined with surrogates, can still be appealing in practical applications. However, the selection of the perturbation size and the associated tuning of the simulator settings can be troublesome. The initial guess for SQP, GPS and HJDS is the centre of the optimisation domain (BHP values of 7,500 psi for all injectors and 3,500 psi for all producers). The initial stencil size for GPS (HJDS) is 600 (750) for the variables related to the injector wells and 400 (500) for the variables related to the producer wells. The associated net present value (NPV) for this starting point is \$31.29 million. The GA has a population size of 300. Because it is a stochastic population-based search scheme, the GA does not require an initial guess.

In the HJDS optimisations here, and in the results reported in Table 3, we have incorporated a heuristic strategy, which seems to be beneficial for these cases. Specifically, at a given exploratory move, the perturbations to the base point are not as described in Section 3.1.2. Rather, the base point is modified in every component, by first considering the perturbation that provided improvement in the cost function at the previous exploratory move.

The performance of the various optimisation algorithms is compared in Figure 5, which presents the evolution of NPV as a function of the number of simulations. In the absence of distributed computing, elapsed time is essentially proportional to the number of simulations. The results are also summarised in Table 2. For the results in the table, we applied the same, practically sensible, stopping criterion for each method (increase in NPV less than \$0.01 million). We note, however, that the methods search differently, so it is not always appropriate to set common stopping criteria. Nevertheless, the results in Table 2, together with those in Figure 5, offer a broad indication of the potential of the algorithms studied. SQP, GPS and HJDS all provide an improvement in NPV of about 25% relative to the starting point, while GA achieves a 23% increase. HJDS requires the least number of function evaluations (simulations) – a factor of 20 fewer than GPS – though as indicated earlier it is a serial algorithm. It is also noteworthy that, in a serial computing environment, HJDS is significantly more efficient than SQP with numerical derivatives.

Figure 5 Performance of the optimisation algorithms for the case in Section 5.1 (see online version for colours)

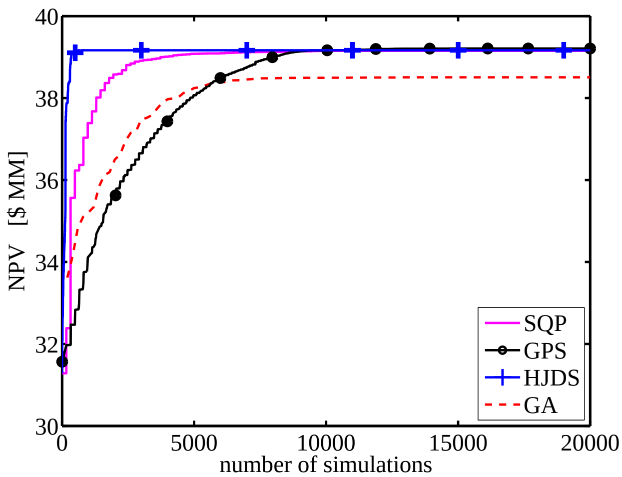


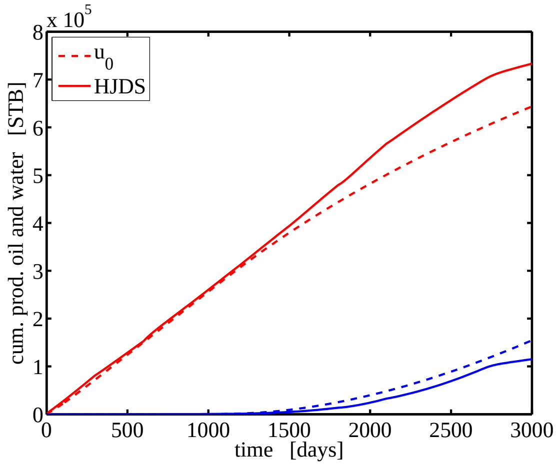
Table 2 Performance summary for the 2D case in Section 5.1

Optimisation approach	*Number of simulations*	*Max. NPV* [\$ *MM*]
SQP	9,450	39.15
GPS	12,570	39.20
HJDS	619	39.16
GA	17,400	38.50

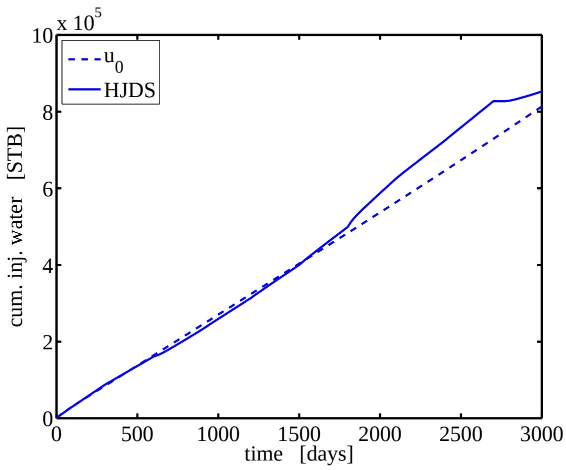
The SQP algorithm with gradients computed through solution of adjoint equations [this implementation is described in Sarma et al. (2006)] was also tested for this problem. This approach provides an optimal NPV of \$39.14 million, essentially the same as

SQP with numerical derivatives, though it requires only 142 function evaluations. We reiterate, however, that simulator-invasive procedures such as this are not the focus of our study. We note also that additional acceleration for all of the methodologies considered could be achieved through use of surrogates.

Figure 6 Cumulative fieldwide production and injection profiles for the initial guess $\mathbf{u}_0$ and HJDS solution for the case in Section 5.1: (a) Cumulative oil (red) and water (blue) production, (b) cumulative water injection (see online version for colours)



(a)

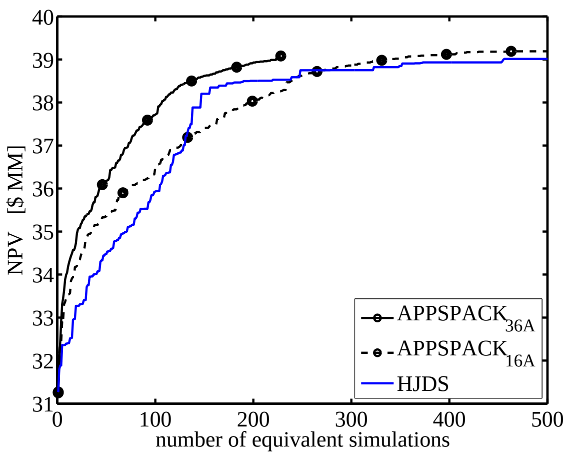


(b)

The cumulative production and injection profiles for the initial guess $\mathbf{u}_0$ and the HJDS optimised case are compared in Figure 6. It is evident that the increase in NPV over the initial guess provided by the optimiser is due to an increase in cumulative oil production and a decrease in cumulative water production. Slightly more water is injected in the optimised solution, though this cost is more than offset by the improvements in production.

For this example we also evaluated the performance of APPSPACK (Griffin et al., 2008), a distributed computing implementation of GPS. We applied an asynchronous distributed implementation of APPSPACK that aims at balancing the computational load for each node used in the cluster. The ideal number of nodes is related to the number of function evaluations required at each iteration (though quite often, this number is determined by the resources available). Results for two APPSPACK optimisation runs, as well as an HJDS run (this optimisation uses a standard implementation of HJDS), are shown in Figure 7, where we plot NPV against the number of equivalent simulations. In this context, the number of equivalent simulations is the total number of simulations divided by the number of nodes used for simulations (one node is not used for simulation because it serves as the master node). Because HJDS is inherently serial, in that case the number of equivalent simulations coincides with the total number of simulations. We observe in Figure 7 that HJDS performs nearly as well as asynchronous APPSPACK using 16 nodes (APPSPACK_{16A}). We also see that APPSPACK using 36 nodes (APPSPACK_{36A}) provides a higher NPV, for a given number of equivalent simulations, than either APPSPACK_{16A} or HJDS. It is of course to be expected that, with enough nodes, APPSPACK will clearly outperform HJDS (when performance is measured on the basis of equivalent simulations).

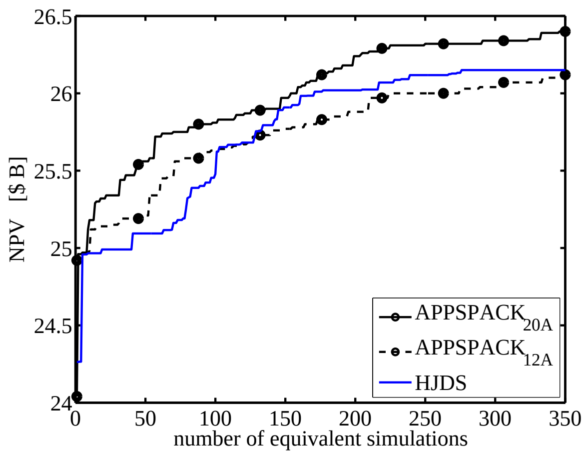
Figure 7 Comparison between GPS within a distributed computing environment (APPSPACK) and HJDS for the case in Section 5.1 (see online version for colours)



We next consider the application of APPSPACK and HJDS for a more involved bound-constrained production optimisation problem. The reservoir model in this case is three-dimensional (the model contains $30 \times 40 \times 10$ grid blocks) and includes 31 production wells and 15 injection wells. There are a total of 93 optimisation variables. A commercial reservoir simulator with 20 licenses was used for this case, and this limited the maximum number of concurrent simulations that could be performed in APPSPACK. Optimisation results are presented in Figure 8. We observe that APPSPACK using 20 nodes (APPSPACK_{20A}) provides the best solution after 350 equivalent simulations.

However, HJDS provides a slightly better result than APPSPACK using 12 nodes ($\text{APPSPACK}_{12\text{A}}$). This demonstrates that HJDS is useful not only in the absence of distributed computing resources, but also when the number of nodes is limited (which may result from commercial simulator licensing issues).

Figure 8 Comparison between GPS within a distributed computing environment (APPSPACK) and HJDS for a bound-constrained production optimisation for a model with 46 wells (see online version for colours)



5.2 Production optimisation with general constraints: 2D example case

The reservoir model used in this case is the same as that considered in Section 5.1, though here the grid block dimensions are different and only four wells are included [two injectors and two producers; see Figure 4(b)]. The reservoir production time frame is 3,650 days, which is divided into five control periods of 730 days each. Thus, the total number of optimisation variables in this problem is 20. There are five non-linear constraints included in this problem, namely maximum total water injection and fluid production rate, minimum total oil production rate, and maximum water cut at each of the two production wells. The production optimisation parameters for this case, including the bounds for the general constraints, are given in Table 1. More details can be found in Isebor (2009).

The specifications for the various optimisation procedures are as follows. The initial stencil sizes for GPS and HJDS are 800 and $1,000$, respectively. The GA has a population size of 200. In all cases, the penalty functions were used as in Nocedal and Wright (2006), with a penalty parameter ρ iteratively increased by one order of magnitude from 10^5 to 10^9 for GPS, and to 10^8 for HJDS and GA. The mechanisms used in this section for modifying ρ are specific to each of the three derivative-free methods considered, and rely on heuristics (minimum stencil size for pattern search techniques, and minimum increment in the cost function averaged over the entire population for GA). The initial guess $\mathbf{u}_0$ for all methods, except the GA, is again the centre of the optimisation domain (i.e., constant BHP of 8,000 psi for all injectors, and

2,500 psi for all producers). This base case has an associated NPV of \$72.90 million. SQP deals with non-linear constraints via an active set method (Nocedal and Wright, 2006) and approximates the required derivatives using second-order finite differencing. Here we use a perturbation size of 0.01 psi.

The performance of the methods for this problem is summarised in Table 3. All of the solutions reported in the table are feasible (i.e., the constraint violation h is zero). The best solution, in terms of maximum NPV for a feasible solution, was found by GPS with penalty function (NPV of \$93.40 million) followed by the SQP method and GA with penalty function. The filter combined with HJDS yields a relatively high NPV using many fewer function evaluations, and it does not require any penalty parameter. The optimised NPVs from the filter-based procedures are not as high, however, as those from penalty-function-based methods. It is difficult to draw firm conclusions regarding the relative performance of different methods or constraint handling techniques in cases where there are only slight differences in NPV, as these discrepancies may be impacted by the choice of stopping criterion (increase in NPV below \$0.01 million).

Table 3 Performance summary for the 2D case in Section 5.2

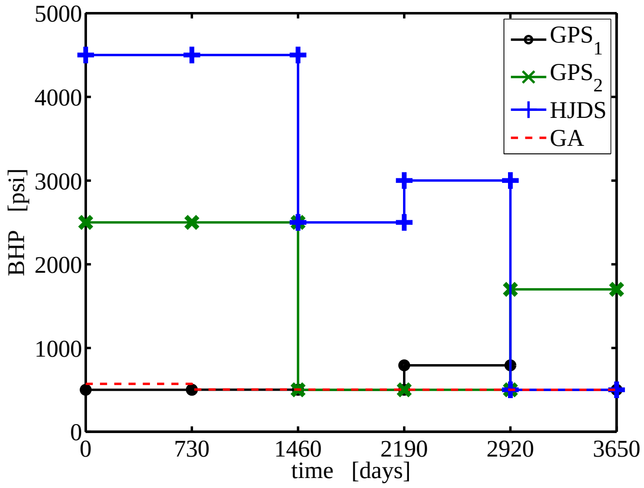
Optimisation approach	*Number of simulations*	*Max. NPV* [\$ *MM*]
SQP + active set	8,059	92.88
GPS + penalty function	6,951	93.40
GPS + filter	1,361	87.63
HJDS + penalty function	4,132	91.65
HJDS + filter	134	90.34
GA + penalty function	16,800	92.76
GA + parameterless penalty	5,400	87.23

GA with parameterless penalty function does, however, appear to underperform relative to the other methods. This algorithm can be readily improved if hybridised with a local optimiser. When combined with GPS and a penalty function, an NPV of \$92.79 million is obtained after $6,417$ function evaluations. This hybrid approach relies, however, on some heuristics (e.g., when to switch from GA to GPS). Further investigation of hybrid procedures will be very useful for delineating practical approaches. We also note that, within a distributed computing environment, the results in Table 3 (except for those using HJDS) can be accelerated according to the number of nodes available.

The optimised BHP values (for one of the injectors) corresponding to four of the methods studied are shown in Figure 9. These BHPs (optimisation variables) are determined for each 730 day period, which gives a total of five control variables per well. As discussed previously, these controls prescribe the source terms in (1) and (2), which determine the performance of the production scenario. It is apparent that the BHP sequences provided by the various methods are quite different (this is also the case for other wells). This phenomenon, which has been observed previously (see e.g., van Essen et al., 2009), appears to result from the fact that the production optimisation problem is underspecified.

In this example, the improvement in NPV over the base case is mainly explained by increased oil production. This is evident in Figure 10 where we present optimal cumulative production and injection profiles determined by GPS with the penalty function and GPS with the filter. The specification of maximum field water injection and maximum water cut leads to solutions that use water more efficiently than in an optimisation scenario that includes only bound constraints.

Figure 9 Four different control sequences extracted from the solutions in Table 3 for one injector well (see online version for colours)

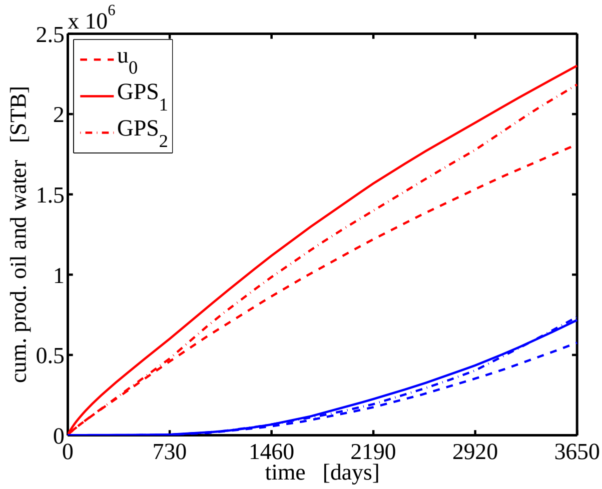


Note: GPS_1, GPS_2, HJDS and GA denote GPS with the penalty function and the filter method, HJDS with the filter method, and GA with the penalty function, respectively.

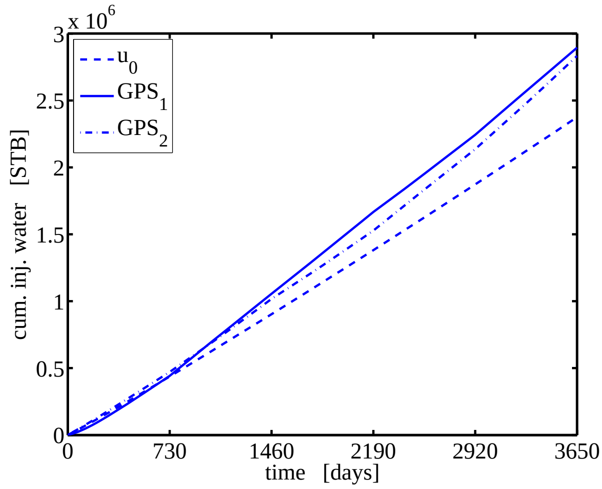
The initial guess $\mathbf{u}_0$ is infeasible and has an associated constraint violation $h(\mathbf{u}_0)$ of 0.0330. This violation is due only to insufficient total oil production. That constraint, together with the maximum total water injection rate constraint, seem to be more important than the other constraints in this problem. In Figures 11 and 12 we illustrate the constraint violation/satisfaction for the initial guess and four of the solutions presented in Table 3. It is apparent that all of the optimised solutions satisfy the constraints, though the various methods accomplish this in quite different ways. Note that, unlike some optimisation procedures, the techniques considered here do not require a feasible initial guess.

The solutions in Table 3 were obtained with a stopping criterion based only on improvement in the cost function. Because in filter methodologies there is a trade-off between the objective function and the constraint violation, that stopping criterion might not be the most appropriate. We have observed that, by performing additional function evaluations and allowing a very small constraint violation, GPS and HJDS with filter yield solutions with higher NPVs. Specifically, setting the maximum normalised constraint violation $h_{\max}$ to 1, and using the same initial guess as above and an initial stencil size of $1,000$, NPVs of $\$94.09$ and $\$92.93$ million, with associated constraint violations of 0.000093 and 0.000096, are obtained after $2,761$ and $1,551$ function evaluations for GPS and HJDS with filter, respectively. Note that these runs demand more function evaluations than were used for the results in Table 3. These values of h correspond to violations in the constraints of less than 1%. The final filters for GPS and HJDS are shown in Figure 13 (the dots indicate the points that constitute the filter). These filters illustrate quantitatively the trade-off between objective function and constraint violation. This trade-off is more conspicuous for HJDS than for GPS. Further research on proper stopping criteria for filter-based production optimisation is needed in order to fully exploit the potential of these approaches.

Figure 10 Cumulative fieldwide production and injection profiles for the initial guess $\mathbf{u}_0$ and solutions computed by GPS for the 2D case in Section 5.2 (see online version for colours)



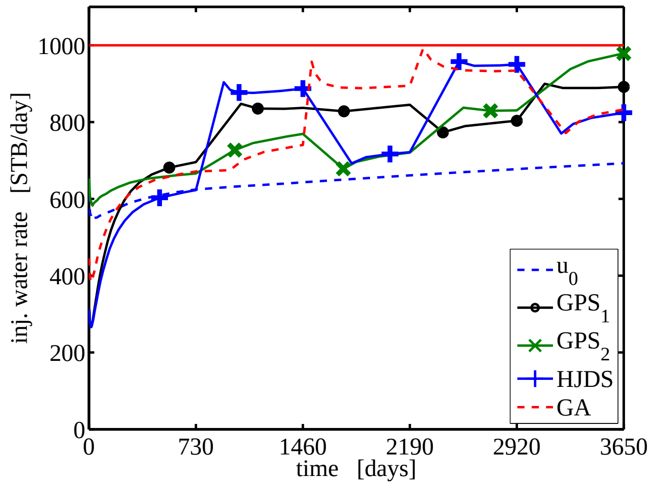
(a)



(b)

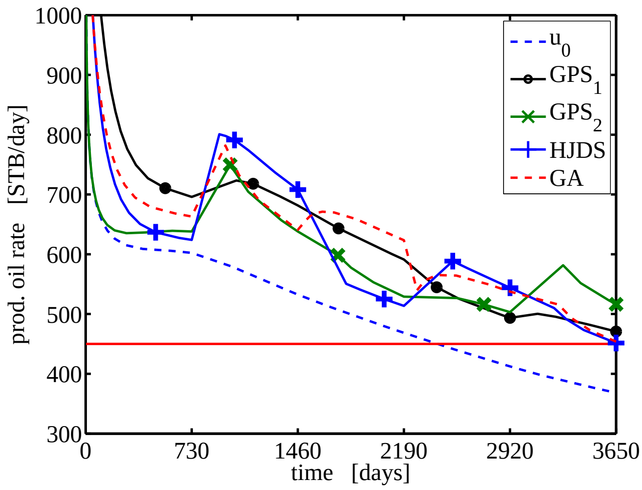
Notes: GPS_1 and GPS_2 denote GPS with the penalty function and the filter method, respectively. (a) Cumulative oil (red) and water (blue) production, (b) cumulative water injection.

Figure 11 Total fieldwide water injection rate for the initial guess $\mathbf{u}_0$ and four solutions found for the 2D case in Section 5.2 (see online version for colours)



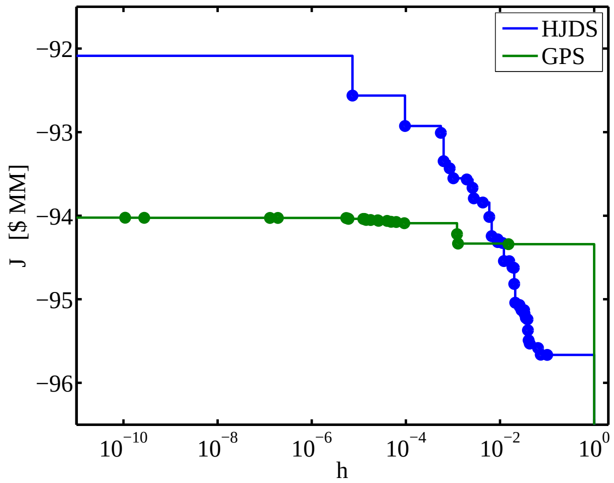
Notes: The red line indicates the maximum total water rate allowed. GPS_1, GPS_2, HJDS and GA denote GPS with the penalty function and the filter method, HJDS with the filter method, and GA with the penalty function, respectively.

Figure 12 Total fieldwide oil production rate for the initial guess $\mathbf{u}_0$ and four solutions found for the 2D case in Section 5.2 (see online version for colours)



Notes: The red line indicates the minimum total oil rate required. GPS_1, GPS_2, HJDS and GA denote GPS with the penalty function and the filter method, HJDS with the filter method, and GA with the penalty function, respectively.

Figure 13 GPS and HJDS final filters for the 2D case in Section 5.2 (see online version for colours)



Note: The dots indicate the points that constitute the filter.

5.3 Production optimisation with general constraints: 3D example case

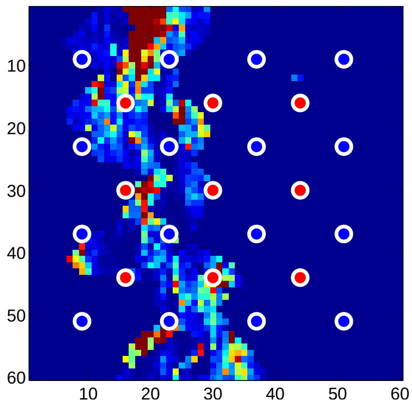
In this problem, the reservoir description is a portion of the SPE 10 model (Christie and Blunt, 2001). This model is three-dimensional and contains $60 \times 60 \times 5$ grid blocks. The 25 wells (16 injectors and nine producers) are distributed following a five-spot pattern (see Figure 14). The production time frame is $1,460$ days, which is divided into four control intervals. Hence, the number of optimisation variables is now 100.

The non-linear constraints refer again to maximum total water injection and fluid production rates, a minimum total oil production rate, and a maximum water cut at each of the nine production wells. The values and ranges for the optimisation parameters are presented in Table 1. Additional details are provided in Isebor (2009).

Based on the results obtained in the previous example, for this case we apply the following four approaches: SQP with numerical derivatives and an active set method, GPS with penalty function, GPS with filter, and HJDS with filter. The gradients within SQP were estimated using second-order finite differencing, with a perturbation size of 0.1 psi. It is important to note that, using the original settings (tolerances) in the GPRS reservoir simulator, the numerical derivatives were not sufficiently accurate for SQP. The numerical gradients eventually used (for which results are shown here) were validated by a brief study. The use of SQP with numerical derivatives thus entails additional computational costs that are not included in the number of simulations reported below.

The initial penalty parameter and stencil size for GPS were 10^5 and $1,375$, respectively. Here we use the following (heuristic) strategy for the penalty parameters and corresponding optimisation stopping criterion. Since the optimisation associated with the first penalty function does not need to be solved very accurately (as the penalty parameter will normally be increased several times thereafter), the stopping criterion at this first stage was a maximum number of simulations of $20,000$. Subsequently, the penalty parameter was set to 10^6, the new optimisation was started using the solution and stencil size from the end of the first stage, and the stopping criterion was updated to a maximum number of simulations of $10,000$. This procedure continued likewise with penalty parameters 10^7, 10^8, 10^9, 10^{10} and 10^{11}, and associated maximum number of simulations of $10,000$, $5,000$, $5,000$, $5,000$, and $5,000$, respectively. The two approaches considered with the filter method, GPS and HJDS, do not rely nearly as directly on heuristics. For both strategies the initial stencil size was $1,375$ and the maximum normalised constraint violation $h_{\max}$ was 1.

Figure 14 Well configurations and top layer of geological model considered in Section 5.3 (see online version for colours)



Notes: Injection and production wells are represented as blue and red circles, respectively. Grid blocks are coloured to indicate value of permeability (red is high permeability, blue is low permeability).

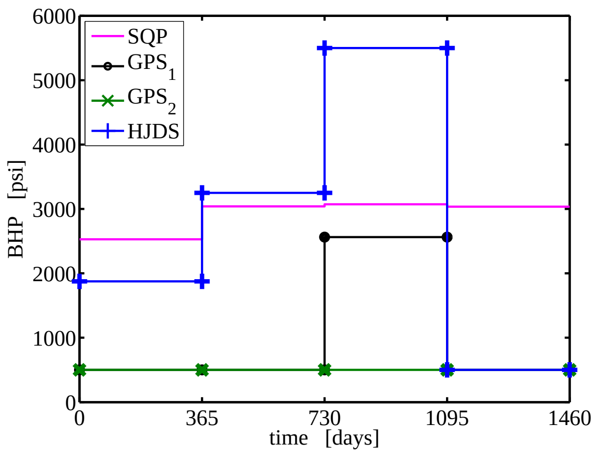
The initial guess $\mathbf{u}_0$ in all cases was the centre of the optimisation domain (i.e., constant BHP of $9,250$ psi for all injectors and $3,000$ psi for all producers). This reference case has an associated NPV of \$193.43 million and a constraint violation value of 0.3731. Most of this constraint violation derives from an excessive total water injection rate, though it is worth noting that the only constraint that is satisfied using the initial controls is the minimum total oil production rate.

The results for this case are summarised in Table 4. As in the case in Section 5.2, the solutions (BHP values) computed by the four approaches are noticeably different (the BHP sequences from the four algorithms for one of the producers are shown in Figure 15). Again, this is consistent with an underspecified optimisation problem. The NPVs for the first three methods are very close to one another, though the NPV for the last method (HJDS with filter) is about 1.5% less. Note that all algorithms except GPS with penalty function have nonzero constraint violations. For the filter-based methods, we allow a constraint violation of 0.0001. Were we to require zero constraint violation, GPS with filter would provide an NPV of \$341.12 million and HJDS with filter would provide an NPV of \$332.93 million. The final filters for GPS and HJDS are shown in Figure 16 (the dots indicate the points that constitute the filter). We again observe a more noticeable trade-off for HJDS than for GPS.

Table 4 Performance summary for the 3D case in Section 5.3

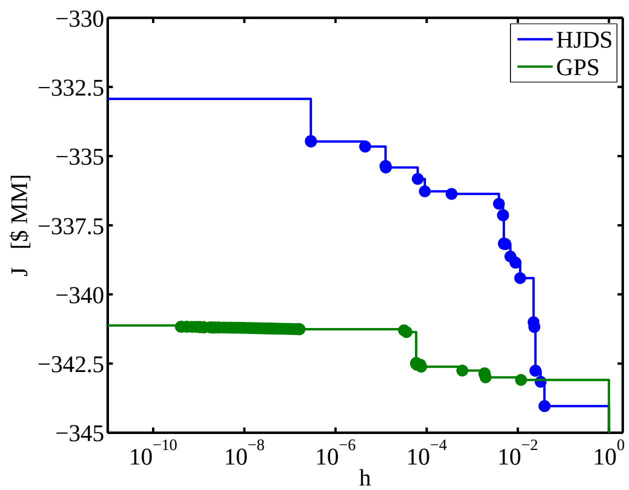
Optimisation approach	*Number of simulations*	*Max. NPV* [*$MM*]	*h*
SQP + active set	41,004	341.32	0.0031
GPS + penalty function	60,001	342.95	0.0000
GPS + filter	39,201	342.61	0.0001
HJDS + filter	1,618	336.28	0.0001

Figure 15 Four different control sequences extracted from the solutions in Table 4 for one producer well (see online version for colours)



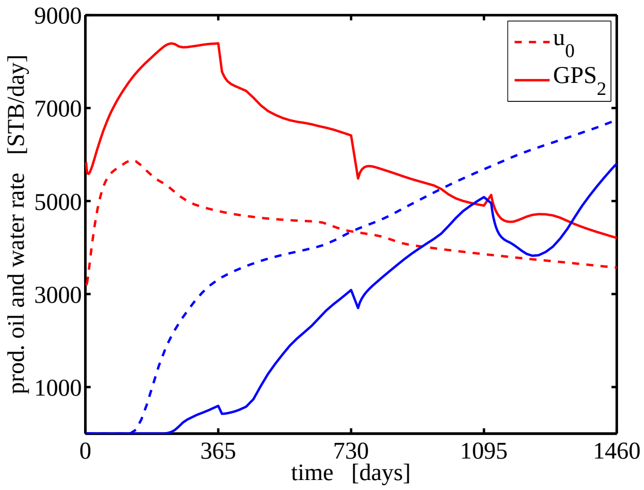
Note: GPS_1 and GPS_2 denote GPS with the penalty function and the filter method, respectively.

Figure 16 GPS and HJDS final filters for the 3D case in Section 5.3 (see online version for colours)

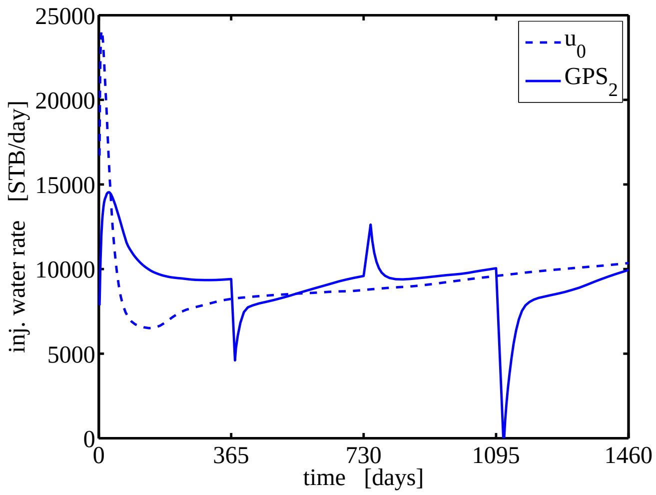


Note: The dots indicate the points that constitute the filter.

Figure 17 Total fieldwide production and injection rates for the initial guess $\mathbf{u}_0$ and solution computed by GPS with filter for the 3D case in Section 5.3: (a) oil (red) and water (blue) production rates, (b) Water injection rate (see online version for colours)



(a)



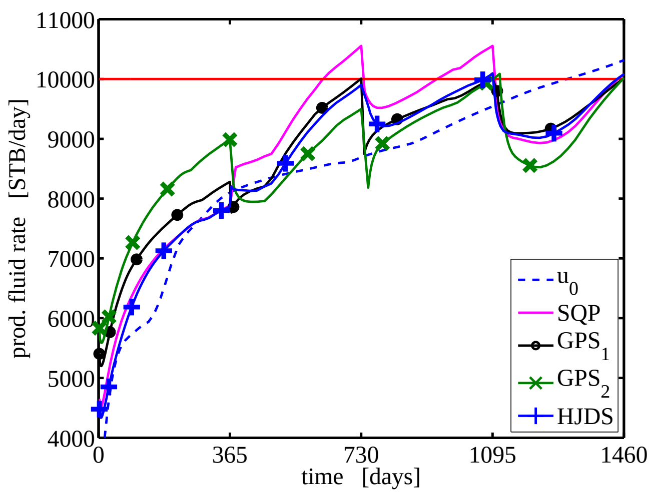
(b)

The SQP algorithm, and the two procedures based on GPS, were implemented within a distributed computing environment (67 cores were used and the speed-up factor achieved was around 50). Hence, although HJDS required many fewer function evaluations, in terms of elapsed time, SQP and GPS with filter took about half the time as HJDS, while GPS with penalty function needed about 3/4 of the time of HJDS. Although the results in Table 4 for GPS with penalty function and GPS with filter are similar, we reiterate that the latter strategy is less heuristic and may, therefore, be preferable in many settings.

The improvement using the optimised controls relative to the initial guess $\mathbf{u}_0$ is due to an increase in the oil production and a decrease in the water production (the cumulative water injection does not vary significantly). This is evident in Figure 17, where we show the production and injection profiles for $\mathbf{u}_0$ and the solution computed by GPS with filter. The peaks in the rates correspond to changes in the BHP controls.

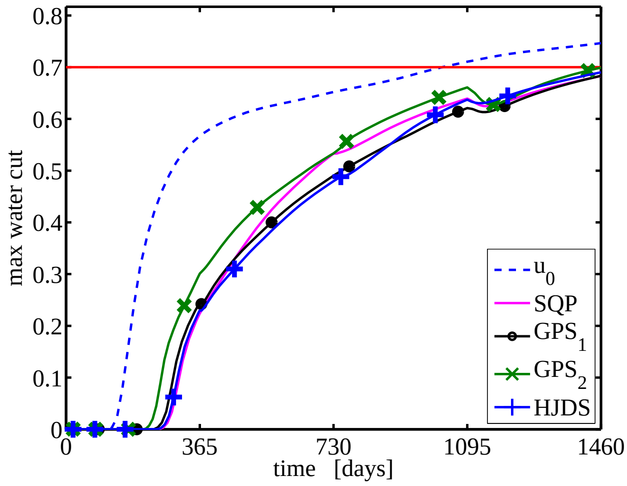
As noted above, at the start of the optimisation the excess in total water injection is responsible for most of the constraint violation. However, for the cases in Table 4, as the optimum solution is approached, the total fluid production rate and water cut are the two most relevant constraints. In Figures 18 and 19 we illustrate the constraint violation/satisfaction for the initial guess and final solutions. Note that the maximum water cut shown in Figure 19 is the maximum of the water cut values for all producer wells. Since the constraints are normalised, when the violation derives mainly from just one constraint, the value $\sqrt{h(\mathbf{u})} \times 100$ quantifies, as a percentage, the violation of that constraint. For example, the constraint violation for SQP, which results from a total fluid production rate that exceeds the maximum allowable, translates into around a 5% violation. Such violation occurs with SQP because our stopping criterion does not enforce strict feasibility. It should be noted that, along the optimisation course, SQP encounters points with a lower constraint violation but also with a lower NPV. This suggests that the results observed for SQP may be improved if this technique is combined with a filter.

Figure 18 Total fieldwide fluid production rate for the initial guess $\mathbf{u}_0$ and the four solutions found for the 3D case in Section 5.3 (see online version for colours)



Notes: The red line indicates the maximum total fluid rate allowed. GPS_1 and GPS_2 denote GPS with the penalty function and the filter method, respectively.

Figure 19 Maximum well water cut for the initial guess $\mathbf{u}_0$ and the four solutions found for the 3D case in Section 5.3 (see online version for colours)



Notes: The maximum water cut at a given time is the maximum of the water cut values for all producer wells at that time. The red line indicates the maximum water cut allowed for any producer well. GPS_1 and GPS_2 denote GPS with the penalty function and the filter method, respectively.

6 Concluding remarks

In this work we have applied non-invasive optimisation methodologies, which can handle general constraints, to simulation-based oil production optimisation problems with up to 100 continuous optimisation variables. The algorithms considered are attractive in complex optimisation scenarios when derivative information is not directly available. We have identified GPS and HJDS as suitable procedures for these optimisation problems. Though a sequential quadratic programming algorithm, with derivatives estimated by finite differencing, performed well in this study, such schemes may lack robustness since the gradient approximation can be sensitive to the perturbation size and the simulator tolerances.

Based on our findings, HJDS is the recommended approach when distributed computing resources are limited or not available at all. When general constraints are present, the penalty function method can provide acceptable results, but a potentially problematic iterative process is required. An alternative approach is to use the filter method, which is an add-on for most optimisation algorithms and does not require a penalty parameter. The filter method combined with pattern search was found to provide promising results in our tests. A parameterless penalty method for genetic algorithms was also studied, and when hybridised with GPS, may provide a robust optimisation strategy. Further research that formalises this hybridisation is needed. All of the techniques discussed can be significantly improved in terms of efficiency if surrogates for the cost function and constraints are available. The development and use of surrogate modelling procedures will be addressed in future work.

Acknowledgements

We are grateful to the industry sponsors of the Stanford Smart Fields Consortium for partial funding of this work. We also thank the Stanford Center for Computational Earth and Environmental Science for providing distributed computing resources.

References

Abramson, M.A. (2007) *NOMADm version 4.6 User's Guide*, Dept. of Mathematics and Statistics, Air Force Institute of Technology.

Almeida, L.F., Tupac, Y.J., Lazo Lazo, J.G., Pacheco, M.A. and Vellasco, M.M.B.R. (2007) 'Evolutionary optimization of smart-wells control under technical uncertainties', SPE paper 107872 presented at the *2007 Latin American & Caribbean Petroleum Engineering Conference*, Buenos Aires, Argentina, 15–18 April.

Artus, V., Durlofsky, L.J., Onwunalu, J. and Aziz, K. (2006) 'Optimization of nonconventional wells under uncertainty using statistical proxies', *Computational Geosciences*, Vol. 10, pp.389–404.

Audet, C. and Dennis, J.E., Jr. (2002) 'Analysis of generalized pattern searches', *SIAM Journal on Optimization*, Vol. 13, No. 3, pp.889–903.

Audet, C. and Dennis, J.E., Jr. (2004) 'A pattern search filter method for nonlinear programming without derivatives', *SIAM Journal on Optimization*, Vol. 14, No. 4, pp.980–1010.

Audet, C. and Dennis, J.E., Jr. (2006) 'Mesh adaptive direct search algorithms for constrained optimization', *SIAM Journal on Optimization*, Vol. 17, No. 1, pp.188–217.

Aziz, K. and Settari, A. (1979) *Petroleum Reservoir Simulation*, Applied Science Publishers.

Bittencourt, A. (1997) *Optimizing Hydrocarbon Field Development Using a Genetic Algorithm Based Approach*, PhD thesis, Dept. of Petroleum Engineering, Stanford University.

Booker, A.J., Dennis, J.E., Jr., Frank, P.D., Moore, D.W. and Serafini, D.B. (1998) 'Optimization using surrogate objectives on a helicopter test example', in J.T. Borggaard, J. Burns, E. Cliff and S. Schreck (Eds.): *Computational Methods for Optimal Design and Control*, Birkhäuser, pp.49–58.

Brouwer, D.R. and Jansen, J.D. (2004) 'Dynamic optimization of waterflooding with smart wells using optimal control theory', *SPE Journal*, Vol. 9, No. 4, pp.391–402.

Cao, H. (2002) *Development of Techniques for General Purpose Simulators*, PhD thesis, Dept. of Petroleum Engineering, Stanford University.

Carroll, J.A., III (1990) *Multivariate production systems optimization*, Master's thesis, Dept. of Petroleum Engineering, Stanford University.

Christie, M.A. and Blunt, M.J. (2001) 'Tenth SPE comparative solution project: a comparison of upscaling techniques', *SPE Reservoir Evaluation & Engineering*, Vol. 4, pp.308–317.

Conn, A.R., Scheinberg, K. and Vicente, L.N. (2009) *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, MPS-SIAM.

Cullick, A.S., Heath, D., Narayanan, K., April, J. and Kelly, J. (2003) 'Optimizing multiple-field scheduling and production strategy with reduced risk', SPE paper 84239 presented at the *2009 SPE Annual Technical Conference and Exhibition*, Denver, Colorado, 5–8 October.

Deb, K. (2000) 'An efficient constraint handling method for genetic algorithms', *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2–4, pp.311–338.

Duvigneau, R. and Visonneau, M. (2004) 'Hydrodynamic design using a derivative-free method', *Struct. Multidisp. Optim.*, Vol. 28, pp.195–205.

Echeverria, D. and Mukerji, T. (2009) 'A robust scheme for spatio-temporal inverse modeling of oil reservoirs', in R.S. Anderssen, R.D. Braddock and L.T.H. Newham (Eds.): *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, pp.4206–4212.

Fletcher, R., Leyffer, S. and Toint, P. (2006) 'A brief history of filter methods', Tech. Rep. ANL/MCS/JA-58300, Argonne National Laboratory.

Fowler, K.R., Reese, J.P., Kees, C.E., Dennis, J.E. Jr., Kelley, C.T., Miller, C.T., Audet, C., Booker, A.J., Couture, G., Darwin, R.W., Farthing, M.W., Finkel, D.E., Gablonsky, J.M., Gray, G. and Kolda, T.G. (2008) 'Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems', *Advances in Water Resources*, Vol. 31, No. 5, pp.743–757.

Gerritsen, M.G. and Durlofsky, L.J. (2005) 'Modeling fluid flow in oil reservoirs', *Annual Review of Fluid Mechanics*, Vol. 37, pp.211–238.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Professional.

Griffin, J.D. and Kolda, T.G. (2007) 'Nonlinearly-constrained optimization using asynchronous parallel generating set search', Tech. Rep. SAND2007-3257, Sandia National Laboratories.

Griffin, J.D., Kolda, T.G. and Lewis, R.M. (2008) 'Asynchronous parallel generating set search for linearly-constrained optimization', *SIAM Journal on Scientific Computing*, Vol. 30, No. 4, pp.1892–1924.

Harding, T.J., Radcliffe, N.J. and King, P.R. (1996) 'Optimization of production strategies using stochastic search methods', SPE paper 35518 presented at the *1996 European 3-D Reservoir Modeling Conference*, Stavanger, Norway, 16–17 April.

Hooke, R. and Jeeves, T.A. (1961) 'Direct search solution of numerical and statistical problems', *Journal of the ACM*, Vol. 8, No. 2, pp.212–229.

Isebor, O.J. (2009) *Constrained production optimization with an emphasis on derivative-free methods*, Master's thesis, Dept. of Energy Resources Engineering, Stanford University.

Jansen, J.D., Brouwer, D.R., Naevdal, G. and van Kruijsdiik, C.P.J.W. (2005) 'Closed-loop reservoir management', *First Break*, Vol. 23, pp.43–48.

Jiang, Y. (2007) *Techniques for Modeling Complex Reservoirs and Advanced Wells*, PhD thesis, Dept. of Energy Resources Engineering, Stanford University.

Kolda, T.G., Lewis, R.M. and Torczon, V. (2003) 'Optimization by direct search: new perspectives on some classical and modern methods', *SIAM Review*, Vol. 45, No. 3, pp.385–482.

Marsden, A.L., Feinstein, J.A. and Taylor, C.A. (2008) 'A computational framework for derivative-free optimization of cardiovascular geometries', *Computational Methods in Applied Mechanics and Engineering*, Vol. 197, pp.1890–1905.

Marsden, A.L., Wang, M., Dennis, J.E., Jr. and Moin, P. (2007) 'Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation', *J. Fluid Mech.*, Vol. 572, pp.13–36.

Meza, J.C. and Martinez, M.L. (1994) 'On the use of direct search methods for the molecular conformation problem', *J. Comput. Chem.*, Vol. 15, pp.627–632.

Nocedal, J. and Wright, S.J. (2006) *Numerical Optimization*, Springer, 2nd ed.

Oeuvray, R. and Bierlaire, M. (2007) 'A new derivative-free algorithm for the medical image registration problem', *Int. J. Model. Simul.*, Vol. 27, pp.115–124.

Ramirez, W.F. (1987) *Application of Optimal Control Theory to Enhanced Oil Recovery*, Elsevier.

Sarma, P., Durlofsky, L.J., Aziz, K. and Chen, W.H. (2006) 'Efficient real-time reservoir management using adjoint-based optimal control and model updating', *Computational Geosciences*, Vol. 10, pp.3–36.

Schlumberger (2008) *ECLIPSE Reference Manual*.

Surry, P.D., Radcliffe, N.J. and Boyd, I.D. (1995) 'A multi-objective approach to constrained optimisation of gas supply networks: the COMOGA method', in T.C. Fogarty (Ed.): *Evolutionary Computing: AISB Workshop*, *Lecture Notes in Computer Science*, Springer-Verlag LNCS 993, pp.166–180.

Torczon, V. (1997) 'On the convergence of pattern search algorithms', *SIAM Journal on Optimization*, Vol. 7, No. 1, pp.1–25.

van Essen, G.M., van den Hof, P.M.J. and Jansen, J.D. (2009) 'Hierarchical long-term and short-term production optimization', SPE paper 124332 presented at the *2009 SPE Annual Technical Conference and Exhibition*, New Orleans, Louisiana, 4–7 October.