Chapter 2
Derivative-Free Optimization for Oil Field Operations

David Echeverría Ciaurri, Tapan Mukerji, and Louis J. Durlofsky

Abstract. A variety of optimization problems associated with oil production involve cost functions and constraints that require calls to a subsurface flow simulator. In many situations gradient information cannot be obtained efficiently, or a global search is required. This motivates the use of derivative-free (non-invasive, black-box) optimization methods. This chapter describes the use of several derivative-free techniques, including generalized pattern search, Hooke-Jeeves direct search, a genetic algorithm, and particle swarm optimization, for three key problems that arise in oil field management. These problems are the optimization of settings (pressure or flow rate) in existing wells, optimization of the locations of new wells, and data assimilation or history matching. The performance of the derivative-free algorithms is shown to be quite acceptable, especially when they are implemented within a distributed computing environment.

2.1 Introduction

Oil and natural gas account for around 60% of the current worldwide primary energy supply, and the demand for these key resources is expected to increase for several decades. Because the development of new fields is often very expensive and technically challenging, it is essential that these operations are performed as efficiently as possible. In addition, the high expense of discovering and developing new fields provides a substantial economic incentive to maximize production from existing fields. Both of these trends provide strong motivation for the development and application of robust methodologies for the computational optimization of oil field operations.

David Echeverría Ciaurri · Tapan Mukerji · Louis J. Durlofsky
Department of Energy Resources Engineering, Stanford University,
Stanford, CA 94305-2220, USA
e-mail: echeverr@stanford.edu, mukerji@stanford.edu, lou@stanford.edu

X.-S. Yang & S. Koziel (Eds.): Computational Optimization & Applications, SCI 359, pp. 19–55.
springerlink.com

The closed-loop reservoir management paradigm [1] provides a framework for efficiently operating an oil field. This approach relies on the continuous acquisition of field data, which are then used to calibrate the computational reservoir model. This represents a data assimilation or history-matching step. The resulting (history-matched) model is then used for optimizing future production. This can be accomplished by either determining optimal settings/controls (e.g., flow rates, well pressures) for existing wells or by finding the best locations for new wells. Given the fact that many different types of wells can be drilled, such as deviated, horizontal or multi-branched wells, the determination of the appropriate well type can also be viewed as an optimization problem.

In this chapter, we address three of the key optimization problems that arise in reservoir engineering – optimization of well settings, optimization of the placement of new wells, and data assimilation. Although there are inter-relationships between these various problems, they have important differences and are typically addressed in a decoupled manner. Well control optimization usually has real-valued decision variables, and a nonlinear, simulation-based cost function and constraints. The well location (often referred to as field development) problem entails, in general, finding the number, type, location and drilling sequence of new wells. In practice, because wells are associated to cell centers in the underlying simulation grid, the optimization variables are typically integers. The well type is described by categorical variables. Model calibration (data assimilation) can be formulated as an inverse problem where we seek to minimize the discrepancy between measured data and model output. The requisite optimization usually involves a very large number of variables (normally at least one per simulation grid block, and in practical problems there are $O(10^4 - 10^6)$ blocks), so parameter reduction and regularization techniques are commonly applied. The subsurface flow simulations required for all of the aforementioned optimizations entail numerical solutions of sets of discretized partial differential equations. These function evaluations can be very costly, and this is a key consideration when designing the optimization framework.

Although our emphasis in this paper is on the use of derivative-free optimization methods, it is important to recognize that gradient-based approaches are appropriate in many settings. In particular, when gradients are available through an adjoint procedure [2], these techniques can be highly efficient. Successful applications of gradient-based methods to oil field problems have been presented in many papers; see, e.g., [3,4,5,6].

Gradient-based approaches do, however, have some drawbacks. As a result of the nonconvex nature of the optimizations considered here, these problems generally contain multiple optima, and hence, a purely local search, which can get trapped in local solutions, might not be the best approach. In addition, for some problems (particularly well placement), the optimization surface can be very rough, which results in discontinuous gradients. It is also important to recognize that derivative information is often not readily available. Adjoint-based techniques, which are a popular way for computing derivatives efficiently, are invasive with respect to the flow simulator, and are therefore only feasible with full access to, and detailed knowledge of, the simulator source code. Numerical gradients are straightforward to calculate, though this computation is expensive and may be subject to practical difficulties (for example, in finite differencing, the selection of the perturbation size and/or simulation tolerances can be problematic). Thus there is clearly a need for other, derivative-free, techniques for oil reservoir optimization problems.

The derivative-free techniques considered in this work are noninvasive with respect to the flow simulator. They treat the simulator as a black-box – only cost function values are required and no explicit gradient calculations are involved. These methods are therefore much easier to implement than, for example, adjoint-based techniques, though this advantage is counterbalanced by a significant deterioration in computational efficiency compared to adjoint approaches. The computational cost associated with derivative-free methods depends strongly on the number of optimization variables considered (in adjoint-based schemes this dependence is much weaker). However, most of these algorithms parallelize naturally and easily, and therefore their efficiency, measured in terms of elapsed time, is usually satisfactory.

Derivative-free optimization approaches can be divided into deterministic (e.g., generalized pattern search) and stochastic (e.g., particle swarm optimization) techniques. Stochastic approaches can be useful for dealing with rough functions or functions that contain multiple local optima. Based on the computational resources typically available in current practice (e.g., $O(100)$ cores), derivative-free optimization methods are appropriate when the number of optimization variables is at most a few hundred [7, 8].

Although gradient-free methodologies have been in existence for many years, they have become widely used in only the last 20 years or so [9]. This relatively recent uptake can be attributed to several factors, including the wide availability of large numbers of cores (combined with algorithms that parallelize easily), the significant theoretical results achieved in this period, and the successful application of derivative-free techniques in a number of areas. Examples can be found in molecular geometry [10], aircraft design [11, 12], hydrodynamics [13, 14] and medicine [15, 16].

Many derivative-free stochastic schemes have also been applied within the oil industry. The field development problem has often been addressed by means of global stochastic-search techniques; see, e.g., [17, 18, 19, 20, 21]. These stochastic schemes have also been hybridized with deterministic search techniques, as presented in [22, 23, 24]. Both global [25, 26] and local (deterministic) [27, 28] derivative-free search techniques have been applied for well control optimization. The history matching problem has also been approached from both a stochastic point of view [29, 18, 30, 31] and using local methodologies combined with regularization and initial guess selection [32, 33].

Our goal in this chapter is to illustrate the applicability of derivative-free optimization methods for three types of problems arising in oil field operations. The examples presented are taken from [28] (well control optimization), [21] (field development optimization), and [32] (history matching). This chapter is structured as follows. In Section 2.2 we briefly describe the simulation modeling procedures and basic optimizers considered. Examples demonstrating the use of derivative-free techniques for well control optimization, field development optimization, and history matching are presented in Sections 2.3, 2.4 and 2.5, respectively. Enhancements to the basic optimization algorithms required for the target problem are discussed in these three sections. We end the chapter with a summary and recommendations.

2.2 Basic Methodologies

We now discuss the simulation techniques used in the optimizations, and describe the basic optimizers considered in this work.

2.2.1 Simulation Techniques

The optimization problems studied here rely on simulations of fluid flow in subsurface formations. Additionally, in Section 2.5, equations describing wave diffraction tomography must also be solved as part of the inverse modeling process. These simulations require the numerical solution of systems of partial differential equations (PDEs).

In this work we consider oil-water systems. These two components exist in separate phases, both of which reside within the pore space of porous rock. Within the context of oil production, the subsurface formation containing oil (and associated water) is referred to as a reservoir. The flow of oil and water in a reservoir is described by statements of mass conservation combined with constitutive (Darcy's law) relationships that relate phase flow rates to pressure gradient. For single-phase flow, Darcy's law is given by $\mathbf{u} = -(k/\mu)\nabla p$, where $\mathbf{u}$ is the Darcy velocity (volumetric flow rate divided by total area), k is the absolute permeability, which is a key property of the rock, μ is fluid viscosity and p is fluid pressure. For two or three-phase flow, this relationship is modified by the inclusion of the so-called relative permeability function, which is a scalar function of local phase volume fraction. Another key quantity is porosity ϕ, which specifies the fraction of the bulk rock volume that is pore space.

In most reservoir simulators, the governing equations are discretized using a finite volume numerical procedure. The detailed equations and discretizations can be found in, e.g., [34, 35]. 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 nonlinear and is solved using a Newton-Raphson procedure. Thus the evaluation of reservoir performance is computationally demanding. In this work we apply Stanford's general purpose research simulator (GPRS; [36, 37]) for two of the cases considered and the commercial streamline simulator 3DSL [38] for the other cases. The streamline simulator shares many similarities with GPRS, though it uses the streamlines from the total velocity field (total velocity is equal to the sum of the water and oil Darcy velocities) to define a coordinate system that is used to solve the water transport equation. This introduces some approximations but it provides a more computationally efficient solution than would typically be achieved using a standard simulator. We note finally that, in the examples presented here, some secondary effects (such as capillary pressure in all cases, compressibility in the streamline simulations) are neglected. These effects could be included if necessary though they would not be expected to impact our basic findings.

Seismic measurements involve first a number of sources, such as dynamite, air guns, or piezoelectric transducers, which send out elastic waves through the reservoir. The transmitted and reflected waves are then recorded on geophones that respond to ground displacement or stresses. The recorded wavefields are processed and analyzed, and by means of a data assimilation process, such as that described in Section 2.5, can be used to infer the rock properties needed in the calculation of oil production forecasts.

In this work diffraction tomography (see e.g., [39, 40, 41]) simulations are used as seismic measurements. The simulations for diffraction tomography require the numerical solution of the elastic wave equation, which describes the propagation of mechanical waves in elastic media. This equation is a statement of conservation of momentum, combined with the constitutive relation for an elastic material relating stresses to strains (Hooke's law). The velocity of the traveling waves depends on the elastic properties of the rock (Young's modulus and Poisson's ratio) and the density, which in turn depend on the rock type, porosity, and the saturations of the pore fluids. Rock physics models relate these rock and fluid properties to the seismic velocities.

The wave equation is solved using the Born approximation [42, 43], which is a perturbation method applied to the scattering of waves in inhomogeneous media. In that approximation, the spatial heterogeneities in elastic properties are divided into a smooth background medium with fluctuations around the background. The wavefield is also divided into an incident wavefield traveling in the background medium along with a scattered wavefield from the heterogeneities. The contributions from the scattered field are expressed in terms of an integral which is computed numerically.

2.2.2 Optimization Problem Statement

A general single-objective optimization problem, as is addressed in this chapter, can be stated as:

$$\min_{\mathbf{x}\in\Omega\subset\mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{g}(\mathbf{x}) \leq 0, \tag{2.1}$$

where $f(\mathbf{x})$ is the objective function (e.g., negative of net present value $(-\mathrm{NPV})$ or norm of discrepancy between measurements and model output), $\mathbf{x}\in\mathbb{R}^n$ is the vector of control variables (e.g., sequence of well pressures, locations for each well, or calibration parameters), and $\mathbf{g}:\mathbb{R}^n\rightarrow\mathbb{R}^m$ represents the nonlinear constraints in the problem. Bound and linear constraints are included in the set $\Omega\subset\mathbb{R}^n$. As indicated above, the objective function (and constraints, in some cases) are computed using the output from a simulator.

Though the optimization problems considered in this work share some commonalities, there are important distinctions between them. Well control optimization is in most cases formulated in terms of continuous variables and includes nonlinear, simulation-based constraints. Previous studies demonstrate that this problem often displays multiple solutions with comparable cost function values [44, 28]. For that reason, this optimization is usually addressed using local search optimization techniques. By contrast, the optimization landscapes found in field development problems can be very rough [20], and this motivates the use of global search approaches.

As is the case with most inverse problems, history matching typically involves more unknowns than informative measurements, which leads to an undetermined optimization problem. Additionally, noise in the measurements can introduce roughness into the cost function. In our application, many of the multiple optima that can result from history matching are not consistent with prior geological information, and should therefore be discarded. Strategies for finding geologically realistic optima include regularization methodologies, performing a global exploration of the search space, and/or selecting a proper initial guess in local optimization schemes. Since the number of optimization parameters in history matching can be comparable to the number of grid blocks in the simulation model, parameter reduction techniques, which can be interpreted in regularization terms, are extremely helpful. These techniques can be used to assure consistency with prior geological information, as described in [45, 46].

Discrete-valued variables are common in optimization problems in the oil and gas industry. Such problems cannot in general be addressed by gradient-based optimizers. In some cases, however, these variables can be treated as real-valued in order to establish a more amenable optimization problem (in this case we say that the discrete-valued variable is relaxed to a real-valued variable).

2.2.3 Derivative-Free Optimization Methods

In this section we describe, within an unconstrained real-valued optimization framework, the derivative-free local and global methods applied in this chapter. Most of these procedures can be extended to cases with discrete-valued variables, bound and/or linear constraints and, with slightly more effort, to problems with computationally inexpensive nonlinear constraints (in Section 2.3.1 we provide mathematically sound procedures for handling simulation-based nonlinear constraints). Additional enhancements of these basic methodologies are introduced for the case examples when necessary. It is important to note that the variants devised for discrete optimization are generally based on heuristics. In Sections 2.3 and 2.5, a gradient-based method, sequential quadratic programming (SQP; see [47]), with numerical derivatives is also considered to enable additional comparisons between the various approaches. The SQP implementation used in this work is SNOPT [48].

2.2.3.1 Local Search Algorithms

The local search techniques considered here are two different pattern search methods: generalized pattern search and Hooke-Jeeves direct search. Pattern search optimization has recently become popular as a result of the development of a solid mathematical convergence theory [49, 8, 7] and of the increasing availability of parallel computing resources. Pattern search schemes evaluate iteratively the cost function in a stencil-based manner. This stencil is modified as iterations proceed, and convergence theory requires that the stencil size eventually tends toward zero [8, 7]. By using a relatively large stencil size during the first stages in a pattern search technique, some local minima can be avoided. This strategy may endow pattern search with a degree of robustness against noisy cost functions. We note that pattern search schemes (and, in general, most local as well as global optimizers) can be accelerated by means of computationally inexpensive surrogates. The use of surrogates can be quite useful for reservoir engineering problems given the large number of expensive objective function evaluations that are typically required.

Generalized Pattern Search

Generalized pattern search (GPS; [49, 50]) comprises a family of optimization algorithms. By considering different types of stencils and various strategies for evaluating the stencil points (which is known as polling [50]), multiple GPS-based optimizers can be constructed. For unconstrained optimization, the basic GPS iteration, for a given stencil centered at the intermediate solution $\mathbf{x}_0$, is as follows. First, the objective function is evaluated for a number of stencil points. If some of these points yield cost function improvement, the current solution is updated with either the best point (if the full stencil is evaluated) or the first point that improves the solution (if an opportunistic search is used). The stencil can then be modified, but in most implementations it stays unaltered. If none of the stencil points improves on $\mathbf{x}_0$, then the stencil size is decreased. The search progresses until some stopping criterion is satisfied (typically, a minimum stencil size).

The stencil should contain a generating set for $\mathbb{R}^n$ [8]. A generating set of vectors has the property that, if $\nabla f(\mathbf{x}_0) \neq 0$, then at least one element of the set is a descent direction [8]. Though only $n+1$ points are needed to establish a generating set for $\mathbb{R}^n$, stencils containing $2n$ elements are commonly used in GPS. We illustrate these two types of stencils in Figure 2.1(a) and 2.1(b).

If the stencil polling process is opportunistic then, as soon as a point improving on the current solution is found, the stencil is moved to that new point. Therefore, only a subset of stencil points will be polled at a given iteration. We show an example of opportunistic polling for a two-dimensional compass stencil in Figure 2.1(c). The point in the east direction is assumed to yield improvement over $\mathbf{x}_0$. As a consequence, the other three points are not evaluated.

In GPS the set of directions in the stencil remains the same at each iteration, which typically provides a coordinate or compass search, as depicted in Figure 2.1(a). The approach can be further generalized by iteratively varying the set of directions in the stencil. For example, at a given iteration the stencil for a two-dimensional optimization problem could be as shown in Figure 2.1(a). Upon polling success, the new stencil is rotated arbitrarily, as in Figure 2.1(d). If the stencil is randomly selected from an asymptotically dense set of directions, the resulting

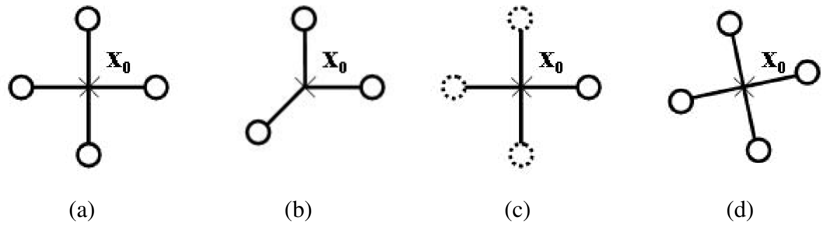


Fig. 2.1 Types of stencil-based search for a two-dimensional space: (a) positive basis with $2n$ directions (compass), (b) positive basis with $n+1$ directions, (c) opportunistic search (the first point tried, the one in the east direction, is assumed to improve on $\mathbf{x}_0$; the other points, for which the cost function is not evaluated, are plotted with dashed lines), and (d) mesh adaptive compass search (the stencil changes randomly at every iteration)

algorithm is the mesh adaptive direct search (MADS; [51]). The MADS approach may be beneficial in situations where the cost function is noisy [51].

If the polling process is not opportunistic (which means the cost function is evaluated for all stencil points), generalized pattern search requires on the order of n function evaluations per iteration. However, the GPS method parallelizes naturally since, at a particular iteration, the objective function evaluations at the polling points are completely independent and can thus be accomplished in a distributed fashion. We note that opportunistic polling is well suited to situations where parallel computing resources are limited or unavailable.

Hooke-Jeeves Direct Search

Hooke-Jeeves direct search (HJDS; [52]) is a compass-based pattern search method. There are two different types of moves in HJDS: exploratory and pattern. In the exploratory move the cost function is evaluated at consecutive perturbations of the stencil center $\mathbf{x}_0$ in the coordinate directions. All directions are polled opportunistically. The exploratory move resembles a numerical gradient estimation (with a perturbation size that may initially be large, but that eventually tends to zero). If no cost function improvement is found in the exploratory step (and this implies $2n$ function evaluations), the stencil size is decreased.

Otherwise, a new point $\mathbf{x}_1$ is obtained, and the next exploratory move is centered at $\mathbf{x}_0 + 2(\mathbf{x}_1 - \mathbf{x}_0)$. This aggressive step in the underlying successful direction is the pattern move, which is somewhat analogous to a line search procedure. The pattern move can be beneficial in situations where an optimum is far from the current solution. If the new exploratory step yields no cost function decrease, another opportunistic compass search is centered at $\mathbf{x}_1$, and if, again, this search yields no improvement, the step size is reduced, keeping the stencil at $\mathbf{x}_1$. Because HJDS is inherently sequential, it is most appropriate for use with serial computing resources.

2.2.3.2 Global Search Algorithms

The global search approaches applied in this work are a genetic algorithm and particle swarm optimization. These techniques share some similarities as they are both based on abstractions of natural processes, have a markedly stochastic nature, and apply sequential updating of a set of solutions (population of individuals in genetic algorithms, swarm of particles in particle swarm optimization).

Genetic Algorithms

Genetic algorithms (GAs) are well known and widely used so our discussion here will be brief (refer to [53] for a detailed description). GAs are inspired by the theory of natural selection. An iteration starts with a population of individuals, which is ranked in terms of cost function (referred to as fitness in the context of GAs). Thereafter, a set of operators, typically selection, crossover and mutation, are applied to generate a new population. The population size, like the swarm size in particle swarm optimization, has a marked impact on the performance of GAs. With a proper population size, a genetic algorithm can be used to explore complex objective function landscapes, and to thus identify promising regions in the search space. A thorough global exploration, even for a moderate number of optimization variables, often requires many function evaluations, and accordingly, a large population size. However, the cost function computation for all of the individuals can be readily performed in a distributed manner.

Particle Swarm Optimization

Particle swarm optimization (PSO; [54,55]) was introduced by Kennedy and Eberhart in the mid 1990s. The algorithm mimics the social behaviors exhibited by swarms of animals. At each PSO iteration, all particles in the swarm move to a new position in the search space. Let $\mathbf{x}_{i,k} \in \mathbb{R}^n$ be the position of particle i at iteration k, $\mathbf{x}^*_{i,k}$ represent the best position (solution) found by particle i up to iteration k, and $\mathbf{y}^*_{i,k}$ be the best position found by any of the particles in the 'neighborhood' of particle i up to iteration k. The neighborhood can include all of the PSO particles, in which case the algorithm is referred to as global-best PSO. Other neighborhood specifications [56] limit particle communication such that particle i interacts with only a subset of the swarm (this has been observed to be useful in avoiding premature convergence). The new position of particle i at iteration $k+1$, $\mathbf{x}_{i,k+1}$, is computed by adding a so-called velocity term, $\mathbf{v}_{i,k} \in \mathbb{R}^n$, to the current position $\mathbf{x}_{i,k}$ [54,55,57]:

$$\mathbf{x}_{i,k+1} = \mathbf{x}_{i,k} + \mathbf{v}_{i,k}. \qquad (2.2)$$

The velocity $\mathbf{v}_{i,k}$ is in turn calculated as follows:

$$\mathbf{v}_{i,k} = \omega \mathbf{v}_{i,k-1} + c_1 \mathbf{r}_1 \circ \left(\mathbf{x}^*_{i,k} - \mathbf{x}_{i,k}\right) + c_2 \mathbf{r}_2 \circ \left(\mathbf{y}^*_{i,k} - \mathbf{x}_{i,k}\right), \qquad (2.3)$$

where ω, c_1, and c_2 are weights, $\mathbf{r}_1$ and $\mathbf{r}_2$ are random vectors in $\mathbb{R}^n$ with components uniformly distributed in the interval $(0,1)$, and $\circ$ denotes the Hadamard (component-wise) product. Thus, we see that each particle moves to a new position based on its existing trajectory, its own memory, and the collective experience of neighboring particles. These three velocity contributions are referred to as the inertia, cognitive, and social components [54, 57].

Some constraints can be handled in PSO through use of the 'absorption' technique [56, 58, 59]. With this approach, particles corresponding to infeasible solutions are moved to the nearest constraint boundary, and the corresponding velocity components are set to zero. We should note that this constraint handling procedure should be accompanied by an efficient scheme for projecting infeasible points back into the feasible domain. When this projection algorithm cannot be applied (e.g., for simulation-based constraints), the penalty function approach is a likely viable alternative (though this approach is not exempt from potential issues; see Section 2.3.1).

2.3 Well Control Optimization with Operational Constraints

The optimization of well settings/controls typically entails maximizing either net present value (NPV) or the cumulative volume of oil produced through time by finding the optimal well flow rates or pressures (these pressures are referred to as bottom-hole pressures or BHPs). In many actual scenarios, and in the cases considered here, water is injected to drive the oil toward production wells and to maintain reservoir pressure. Secondary objectives could include minimizing the total volume of water injected or produced, or maximizing the initial oil production rate. The problem is usually solved subject to operational constraints, such as maximum and minimum BHP, maximum water injection rate, maximum well water cut (fraction of water in the produced fluid), etc. The optimization variables are generally real-valued, and the relationships between these variables and both the objective function and constraints are in general nonlinear. Thus, the problem can be addressed by nonlinear programming techniques [47].

The production optimization cases presented here involve the maximization of undiscounted NPV by adjusting the BHPs of water injection and production wells (well flow rates could also have been the optimization variables). The objective function we seek to minimize is

$$f(\mathbf{x}) = -\mathrm{NPV}(\mathbf{x}) = -r_o Q_o(\mathbf{x}) + c_{wp} Q_{wp}(\mathbf{x}) + c_{wi} Q_{wi}(\mathbf{x}), \tag{2.4}$$

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

2.3.1 Constraint Handling Techniques

The nonlinear programming methods applied here are generalized pattern search (GPS), Hooke-Jeeves direct search (HJDS), and a genetic algorithm (GA), with enhancements introduced to deal with general constraints. Consistent with the derivative-free spirit of this work, the constraint handling techniques considered, namely penalty functions and filter methods, allow us to continue treating the simulator as a black-box. These methodologies are not exclusive to gradient-free optimizers, so they could be implemented with a wide variety of optimization approaches. The description below of constraint handling techniques follows the discussion presented in [28].

Penalty Functions

The penalty function method (see, e.g., [47]) for general optimization constraints entails modification of the objective function with a penalty term that depends on some measure of the constraint violation $h : \mathbb{R}^n \to \mathbb{R}$. The modified optimization problem

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x}) + \rho\, h(\mathbf{x}) \,, \tag{2.5}$$

where $\rho > 0$ is a penalty parameter, may still have constraints, but they should be straightforward to handle (for example, bound constraints). In this work we apply $h(\mathbf{x}) = ||\mathbf{g}^+(\mathbf{x})||_2^2$, with $\mathbf{g}^+ : \mathbb{R}^n \to \mathbb{R}^m$ defined as $g_i^+(\mathbf{x}) = \max\{0, g_i(\mathbf{x})\}$ (normalizing the constraints can be beneficial since they are all weighted equally in the penalty term). If the penalty parameter is iteratively increased (tending to infinity), the solution of the modified optimization problem (2.5) converges to that of the original nonlinearly constrained problem. However, the sequence of values to use for ρ may require some numerical experimentation and the overall procedure can lead to significant additional computation. In certain cases, a finite (and fixed) value of the penalty parameter also yields the correct solution (this is the so-called *exact* penalty; see [47]). However, for exact penalties, the modified cost function is not smooth around the solution [47], and thus the corresponding optimization problem can be challenging to solve.

Filter Method

The penalty function approach is straightforward to implement but, as discussed above, can introduce some potential difficulties and complications. Filter methods [60, 47] provide an alternate and systematic approach for handling general constraints. A filter is a set of pairs $(h(\mathbf{x}), f(\mathbf{x}))$, such that no pair dominates another pair. The concept of dominance, borrowed from multi-objective optimization, is defined as follows: the point $\mathbf{x}_1 \in \mathbb{R}^n$ dominates $\mathbf{x}_2 \in \mathbb{R}^n$ if and only if either $f(\mathbf{x}_1) \le f(\mathbf{x}_2)$ and $h(\mathbf{x}_1) < h(\mathbf{x}_2)$, or $f(\mathbf{x}_1) < f(\mathbf{x}_2)$ and $h(\mathbf{x}_1) \le h(\mathbf{x}_2)$. In this work, the constraint violation h associated to the filter method is computed the same way as described above for the penalty method.

Filters have been combined with a variety of basic optimization algorithms including sequential quadratic programming [60], interior point methods [61], and pattern search techniques [62, 63]. They can be understood as essentially an add-on for a basic optimization procedure. Within the context of a pattern search method, a filter acts to modify the standard acceptance criterion which, as discussed in Section 2.2.3, is based only on cost function improvement. At a given iteration, the basic optimization algorithm proposes a number of intermediate solutions. These solutions are accepted if they are not dominated by any point in the filter. Prior to continuing with the next iteration, the filter is updated based on all the points evaluated by the optimizer. Using filters, the original problem (2.1) is thus viewed as a bi-objective optimization: besides minimizing the cost function $f(\mathbf{x})$, we also minimize the constraint violation $h(\mathbf{x})$. Using this multi-objective perspective, the optimization search is enriched by considering infeasible points. We reiterate that the ultimate solution is intended to be feasible (it may however show a very small constraint violation).

2.3.2 Production Optimization Example

This example is taken from [28]. The reservoir is a portion of the synthetic SPE 10 model [64]. It is represented on a three-dimensional grid containing $60 \times 60 \times 5$ blocks. The reservoir contains oil and water. The 25 wells (16 water injectors and nine producers) are distributed following a five-spot pattern (see Figure 2.2). This model is similar to models used in practice except it contains fewer grid blocks. The variation in permeability, evident in Figure 2.2, strongly impacts the flow field.

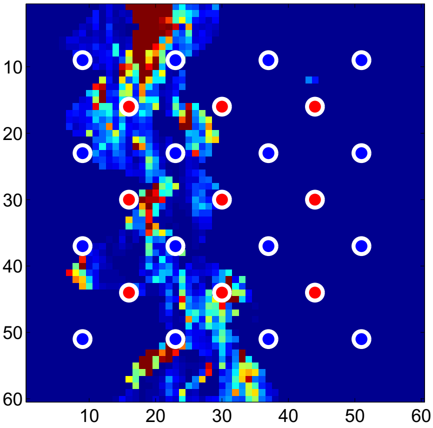


Fig. 2.2 Well configurations and top layer of the geological model considered in the production optimization case in Section 2.3. Grid blocks are colored to indicate value of permeability (red is high permeability, blue is low permeability). Injection and production wells are represented as blue and red circles, respectively (from [28]); see online version for colors.

By optimizing the well settings, we can achieve a more uniform distribution of the injected water, thus increasing the amount of oil produced and maximizing NPV.

Reservoir production proceeds for a total of 1460 days. The BHP of each well is updated every 365 days. There are thus a total of four control intervals. Since there are 25 wells, the number of optimization variables is 100. During each control interval, the BHPs are held constant. Injection well BHPs are specified to be in the range 6500 – 12000 psi and production wells are constrained to the range 500 – 5500 psi.

The additional constraints, which are nonlinear, specify that (1) the maximum field-wide water injection rate not exceed 15000 STB/day, (2) the maximum field-wide liquid (oil+water) production rate not exceed 10000 STB/day, (3) the minimum field-wide oil production rate not fall below 3000 STB/day, and (4) the fraction of water in the produced fluid (water cut) not exceed 0.7 in any of the nine production wells. The oil price considered is $\$50/\mathrm{STB}$, and the costs of produced and injected water are $\$10/\mathrm{STB}$ and $\$5/\mathrm{STB}$, respectively. Additional details of the problem specification are provided in [65].

Based on results for another nonlinearly constrained production optimization problem presented in [28], we apply the following four approaches for this case: sequential quadratic programming (SQP) with numerical derivatives and an active set constraint handling method [47], generalized pattern search (GPS) with penalty function, GPS with filter, and Hooke-Jeeves direct search (HJDS) with filter. The gradients required by SQP were computed using second-order finite differencing, with a perturbation size of 0.1 psi (this perturbation size was established through numerical experimentation – we reiterate that this can be an issue when estimating gradients numerically). In all cases, the initial stencil size for GPS and HJDS was 1375 psi. The penalty method relies on some heuristics for increasing the penalty parameter and terminating each corresponding intermediate optimization. Details on the strategy used here can be found in [28]. The two approaches considered with the filter method, GPS and HJDS, do not rely nearly as directly on heuristics.

The initial guess $\mathbf{x}_0$ for all methods was the center of the orthotope given by the bound constraints (i.e., BHP of 9250 psi for all injectors at all times, BHP of 3000 psi for all producers at all times). This reference case has an associated NPV of \$193.43 million and a constraint violation value of 0.3731. The optimization results are summarized in Table 2.1. Consistent with the underdetermined nature of

Table 2.1 Performance summary for the production optimization case (from [28])

Optimization approach	Number of simulations	Max. NPV [\$ MM]	*h*
SQP + active set	41004	341.32	0.0031
GPS + penalty function	60001	342.95	0.0000
GPS + filter	39201	342.61	0.0001
HJDS + filter	1618	336.28	0.0001

the optimization problem, the solutions computed by the four approaches differ. The NPVs for the first three methods are within 0.5% of 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 allowed 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.

All algorithms other than HJDS were implemented within a distributed computing environment (67 cores were used, which provided a speedup factor of around 50). We therefore observe that, although SQP and GPS with filter required a factor of about 24 times more function evaluations than HJDS, in terms of elapsed time, these two methods required only about half the time as HJDS. The procedure that required the highest number of function evaluations, GPS with penalty function, needed about 3/4 of the time of HJDS. This highlights the impact of the availability of multiple cores on algorithm selection. We note finally that, although the results in Table 2.1 for GPS with penalty function and GPS with filter are similar, the filter method is less heuristic and may, therefore, be preferable for many problems.

We now illustrate the degree of nonlinear constraint satisfaction provided by the various optimization algorithms. Figures 2.3 and 2.4 present the field-wide fluid production rates and the maximum of the water cut in any producer well. The red horizontal lines in these figures indicate the constraint value. It is evident that, at late time, the initial guess settings lead to constraint violations. The constraints are essentially satisfied by the other algorithms, with the exception of SQP. This occurs

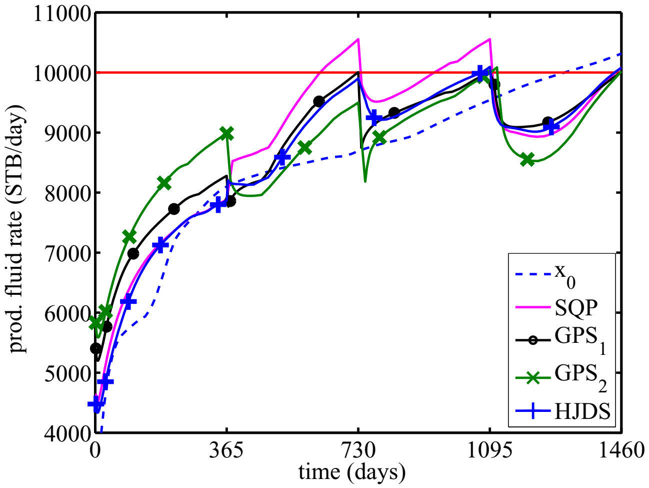


Fig. 2.3 Total field-wide fluid production rate for the initial guess $\mathbf{x}_0$ and the four solutions found for the production optimization case. The red line indicates the maximum total fluid rate allowed. GPS_1 and GPS_2 denote GPS with the penalty function and the filter method, respectively (from [28]); see online version for colors.

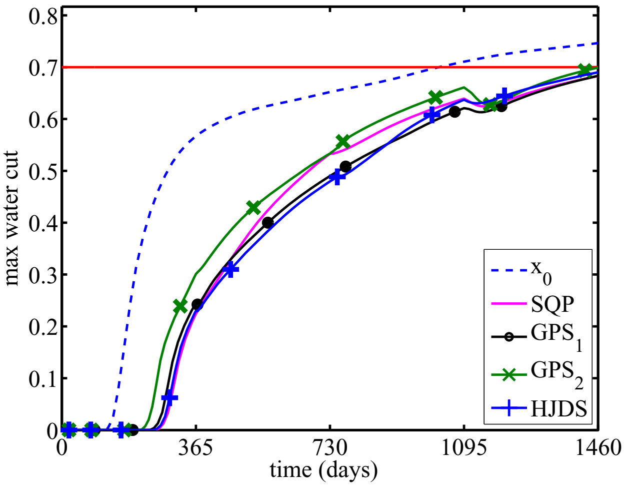


Fig. 2.4 Maximum well water cut for the initial guess $\mathbf{x}_0$ and the four solutions found for the production optimization case. The maximum water cut at a given time is the maximum of the water cut values for all producer wells at that time. The red line indicates the maximum water cut allowed for any producer well. GPS_1 and GPS_2 denote GPS with the penalty function and the filter method, respectively (from [28]); see online version for colors.

because our SQP stopping criterion does not enforce strict feasibility. SQP does, however, encounter solutions during the course of the optimization with lower constraint violations but also with lower NPVs. Thus, it is clear that the SQP results could be improved if it was used with a filter.

The quantities that directly impact NPV are displayed in Figure 2.5, where we show the production and injection profiles for $\mathbf{x}_0$ and for the solution computed by GPS with filter. The peaks in the rates in the optimized solution, evident every 365 days, result from the changes in the well BHPs, which occur at those times. It is evident that, relative to the initial guess, the optimized controls lead to a significant increase in cumulative oil production along with a significant decrease in cumulative water production (note that cumulative oil production corresponds to the integral of the curve shown in Figure 2.5, and similarly for other quantities). The cumulative water injection does not vary significantly between the two cases. This example illustrates the substantial gains that can potentially be achieved in oil field operations through the use of computational optimization.

2.4 Optimal Well Placement with Particle Swarm Optimization

The general problem of field development optimization involves the determination of how many new wells to drill, what type of wells these should be (i.e., injection well or production well; vertical, horizontal or multi-branched well; type of

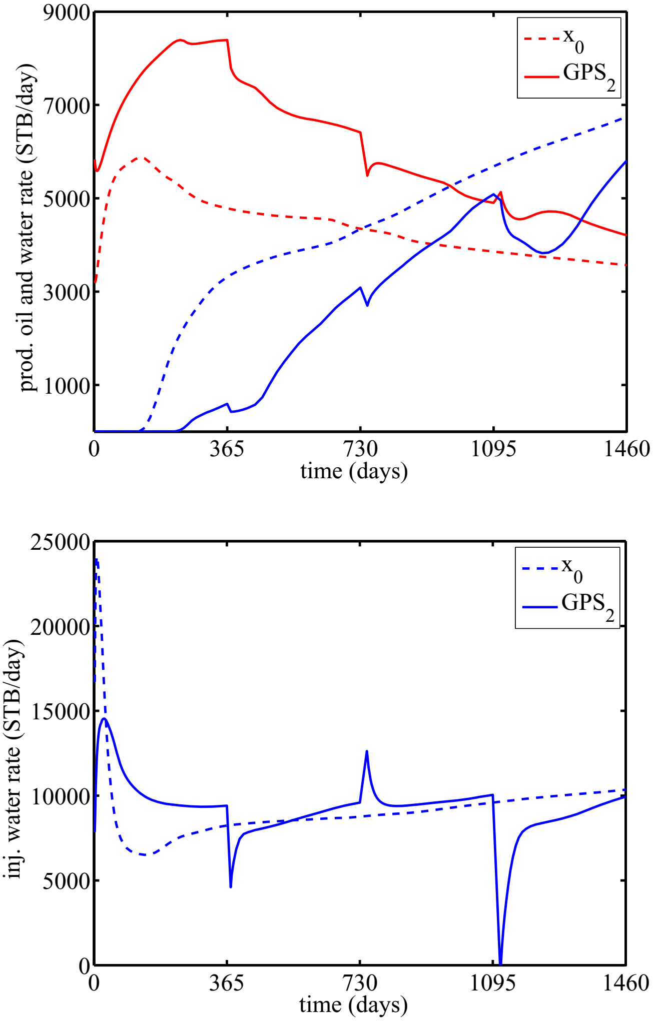


Fig. 2.5 Total field-wide production and injection rates for the initial guess $\mathbf{x}_0$ and solution computed by GPS with filter for the production optimization case. Top: Oil (red) and water (blue) production rates. Bottom: Water injection rate (from [28]); see online version for colors.

downhole instrumentation), and the drilling schedule, in order to maximize a prescribed objective function. In previous work, a number of gradient-based and derivative-free procedures have been developed and applied for this problem (see [20] for a full discussion). Of the stochastic search approaches employed, many researchers have applied genetic algorithms (e.g., [23, 66, 24, 67, 68, 19, 69, 70]), though simultaneous perturbation stochastic approximation algorithms [71], as well as other approaches, have also been explored [68]. Mattot et al. [72] evaluated several optimization algorithms for a groundwater remediation problem and achieved the best results using particle swarm optimization (PSO). This motivated the use of PSO for optimization of oil field development in [20]. Consistent with [72], in [20] PSO was found to outperform GA for several example cases. All of these examples involved relatively few wells (20 or less).

The development of large-scale oil fields, however, often involves drilling many wells. If we restrict ourselves for now to vertical wells (which can be either production or injection wells) that penetrate the entire thickness of the formation, the optimization variables include the areal (x,y) location of each well and a binary variable b defining the well type. Thus there are a total of $n = 3N_w$ optimization variables, where N_w is the number of wells. Even given the restriction of fully-penetrating vertical wells, the optimization problem is challenging. For large-scale problems, N_w can be several hundred, so the number of optimization variables can be large. In addition, for large N_w the imposition of well-to-well distance constraints (which are commonly used in field applications) can lead to a large number of infeasible solutions, and this can negatively impact the performance of a population-based algorithm such as PSO. Another key concern is that the number of wells N_w should itself be an optimization variable. Direct inclusion of N_w as an integer variable in the set of parameters will further complicate the optimization and will lead to much larger computational requirements.

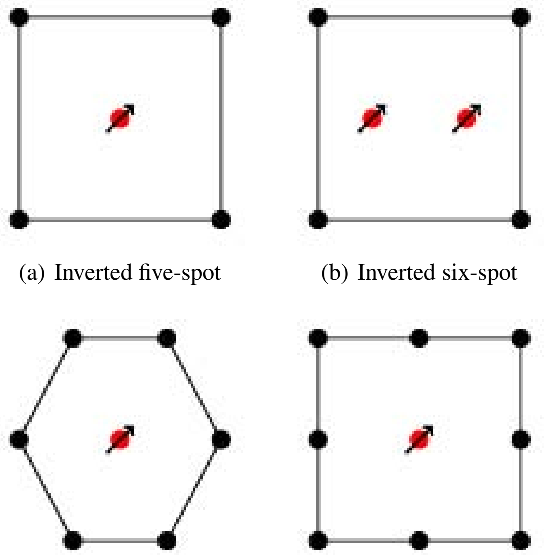
2.4.1 Optimization Methodology

In recent work, a field development optimization procedure that addresses some of the issues raised above was presented [21]. In this implementation, rather than prescribe N_w and optimize $3N_w$ parameters, the wells were constrained to be arranged in repeated patterns (such patterns are commonly used for onshore oil field development). By optimizing the parameters that define the well patterns, a close-to-optimal N_w and the locations and types of all wells can be determined. This method would theoretically be expected to lead to suboptimal results relative to those that could be achieved by optimizing the number of wells and the associated $3N_w$ parameters, but it is much more tractable computationally than the more exhaustive approach.

In this section we describe and then apply this new well pattern optimization procedure and a second-stage optimization that perturbs well locations within the patterns. The core optimizer used is PSO, but the method could be implemented with other derivative-free optimization algorithms including GA.

2.4.1.1 Well Pattern Description

The basic PSO procedure was described in Section 2.2.3.2. In the well pattern description (WPD), the optimization parameters define the target pattern. This pattern is then replicated over the entire domain, with wells that fall outside of the reservoir eliminated. The algorithm considers four different well pattern types, as shown



(a) Inverted five-spot (b) Inverted six-spot

(c) Inverted seven-spot (d) Inverted nine-spot

Fig. 2.6 Illustration of the well patterns considered. The solid black circles represent production wells and the circles with arrows represent injection wells. The patterns are referred to as 'inverted' because the injection wells are at the centers of the patterns (from [21]).

in Figure 2.6. Optimization variables include the pattern type (categorical variable I_i^{wp}), the location of one of the wells in the pattern (ξ_i^0, η_i^0), pattern dimensions (a_i, b_i), and parameters associated with a number of pattern operators, which we now describe.

The patterns determined using the representation above will be quite regular and oriented with the $x-y$ coordinate system. It may be advantageous, however, to adjust the orientation of the pattern to better accommodate the reservoir shape or the spatial variation/correlation of rock properties such as permeability. To accomplish this, several different pattern operators were introduced in [21]. These include a rotation operator, a shear operator and a scale operator. Well locations for the target pattern, after application of these operators, can be expressed as:

$$\mathbf{W}_{out}^T = \mathbf{M}\,\mathbf{W}_{in}^T, \tag{2.6}$$

where $\mathbf{W}_{out}$ and $\mathbf{W}_{in}$ are $N_{wp} \times 2$ (relative) well location matrices, where N_{wp} is the number of wells in the pattern, and $\mathbf{M}$ is a 2×2 transformation matrix, defined for each operator. For example, for the rotation operator, we have:

$$\mathbf{M}_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \tag{2.7}$$

where θ designates the angle of rotation. **M** matrices are also defined for shear and scale operators; see [21] for details. A fourth operator, referred to as 'switch,' which acts to convert all injection wells to production wells and vice versa, was also introduced. This operator changes the target pattern from the so-called 'normal' form to the 'inverted' form (or back).

The full set of optimization variables for the well pattern description, for PSO particle i, is given by (with the iteration index k omitted for clarity):

$$\mathbf{x}_i = [\underbrace{\{I_i^{wp}, [\xi_i^0, \eta_i^0, a_i, b_i]\}}_{\text{pattern parameters}} \underbrace{\{S_{i,1}, S_{i,2}, \ldots, S_{i,\mathscr{N}_o}\}}_{\text{operator sequence}} \underbrace{\{\mathscr{O}_{i,1}, \mathscr{O}_{i,2}, \ldots, \mathscr{O}_{i,\mathscr{N}_o}\}}_{\text{pattern operators}}]. \tag{2.8}$$

Here $\{I_i^{wp}, [\xi_i^0, \eta_i^0, a_i, b_i]\}$ are the basic pattern parameters for particle i, $\mathscr{N}_o$ is the number of pattern operators, $\{\mathscr{O}_{i,1}, \mathscr{O}_{i,2}, \ldots, \mathscr{O}_{i,\mathscr{N}_o}\}$ are the parameters associated with the pattern operators, and $\{S_{i,1}, S_{i,2}, \ldots, S_{i,\mathscr{N}_o}\}$ defines the sequence in which the operators are applied. The total number of optimization variables depends on the number and type of operators included, but it is only around 25 when all of the operators noted above are used. All components of $\mathbf{x}_i$ are treated as real numbers in the optimization. Some of these parameters (e.g., I_i^{wp} and $S_{i,j}$) are, however, integers. Where necessary, integer values are determined from real values by simply rounding to the nearest integer.

2.4.1.2 Second-Stage Optimization

Following the determination of the optimum repeated pattern using the well pattern description (WPD) approach described above, a second-stage optimization can be applied to further improve the solution. This procedure is based on a well-by-well perturbation (WWP) and involves the local shifting of wells within patterns. Optimization variables (PSO particles) for WWP optimization are:

$$\mathbf{x}_i = \{\underbrace{\Delta\xi_1,\ \Delta\eta_1}_{\text{well 1}},\ \underbrace{\Delta\xi_2,\ \Delta\eta_2}_{\text{well 2}},\ \ldots,\ \underbrace{\Delta\xi_j,\ \Delta\eta_j}_{\text{well } j},\ \ldots,\ \underbrace{\Delta\xi_{N_w},\ \Delta\eta_{N_w}}_{\text{well } N_w}\}, \tag{2.9}$$

where N_w is the number of wells determined in the first-stage (WPD) optimization and $\Delta\xi_j$ and $\Delta\eta_j$ are the perturbations of the spatial locations of well j. The minimum and maximum values of $\Delta\xi_j$ and $\Delta\eta_j$ are constrained to keep wells essentially within their original patterns. The dimension of this optimization problem can be high for large N_w, but the size of the search space is greatly limited by bound constraints on $\Delta\xi_j$ and $\Delta\eta_j$. We note finally that this second-stage optimization could be extended to determine completion intervals (i.e., vertical locations where the well is open to flow), to eliminate particular wells, or to modify individual well types.

2.4.2 Field Development Optimization Example

We now apply the procedures described above to a two-dimensional reservoir model. This example is taken from [21]; refer to that paper for full details. The reservoir domain is irregular, as shown in Figure 2.7, where the dark regions along the boundaries designate non-reservoir zones. Wells that fall outside of the reservoir region are eliminated from the set. The model contains a total of 80 $\times$ 132 grid blocks. The production and injection wells are prescribed to operate at fixed bottom-hole pressures of 1200 psi and 2900 psi, respectively. The total production time is 1825 days. Flow simulations for this case were performed using the streamline simulator 3DSL [38]. Streamline simulators are not as broadly applicable as standard finite-volume based simulators, but when appropriate, as they are in many waterflood simulations, streamline approaches can be considerably more efficient than standard procedures.

The well pattern optimization runs used 40 PSO particles and proceeded for 40 iterations. The optimization was run five times. Following these five runs, the best optimization solution (run 3 in Table 2.2) was used for five subsequent WWP optimizations. Results for NPV for the well pattern optimizations are shown in Table 2.2, while those from the subsequent use of WWP are presented in Table 2.3. It is evident from Table 2.2 that the inverted five-spot was the best pattern in all runs. We see from Table 2.3 that WWP consistently led to improvements of around 20% over the unperturbed patterns. The progress of the overall optimization is displayed in Figure 2.8, where the improvement in NPV during both stages is evident.

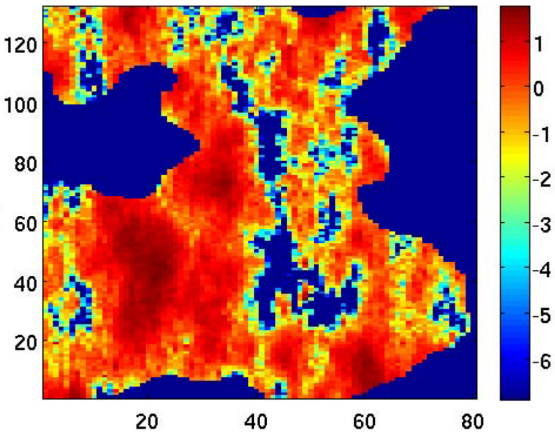


Fig. 2.7 Logarithm of permeability field for field development optimization example (from [21]).

Table 2.2 Optimization results using well pattern description (from [21])

Run	Best pattern	NPV ($MM)	Well count	
			Producers	Injectors
1	inv. 5-spot	1377	16	15
2	inv. 5-spot	1459	15	15
3	inv. 5-spot	1460	15	15
4	inv. 5-spot	1372	15	15
5	inv. 5-spot	1342	13	15
Average		**1402**		

Table 2.3 Optimization results using the second-stage procedure relative to run 3 (from [21])

Run	NPV ($MM)	Increase over well pattern description	
		($MM)	%
1	1777	317	21.7
2	1787	327	22.4
3	1776	316	21.6
4	1801	341	23.4
5	1771	311	21.3
Average	**1782**	**322**	**22.1**

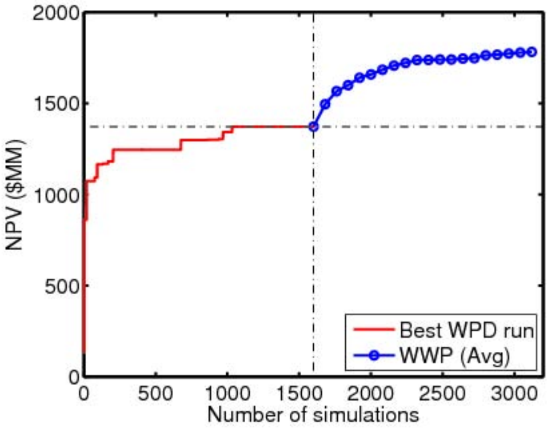
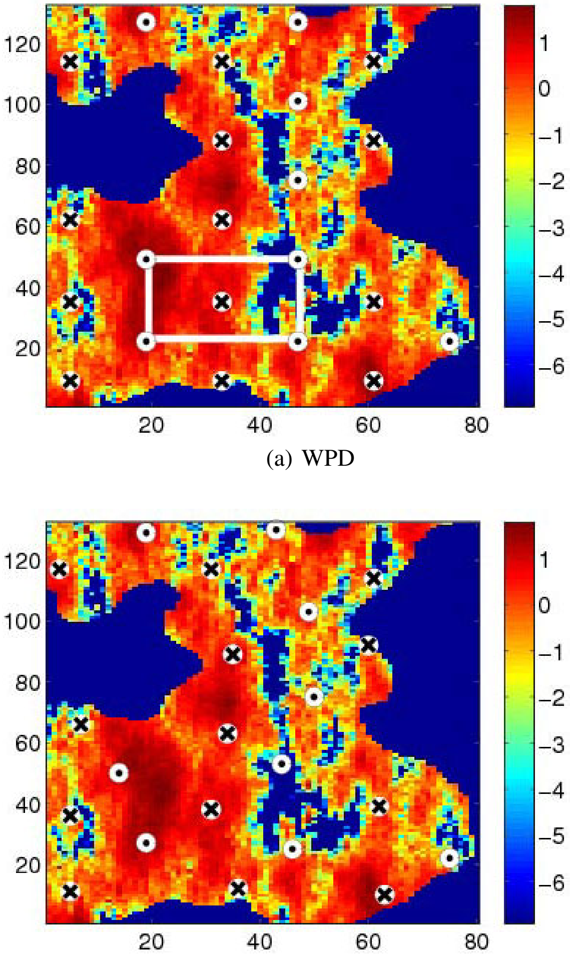


Fig. 2.8 NPV of best result from well pattern description (WPD), and average NPV of the best second-stage well-by-well perturbation (WWP) solutions, versus number of simulations (from [21]).

Figures 2.9(a) and (b) show the optimal well locations from both stages of the optimization. Repeated five-spot patterns are evident in both figures. It is interesting to observe that, although the differences in well locations between the two figures are relatively slight, these perturbations result in an improvement in NPV of 23%.



(a) WPD

(b) WWP

Fig. 2.9 Well locations for the best well pattern description (WPD) and well-by-well perturbation (WWP) solutions (circles indicate production wells, crosses indicate injection wells). Logarithm of permeability field is shown as background (from [21]).

We note finally that several other examples demonstrating the use of PSO for well placement optimization were presented in [20, 21]. In the examples in [20] the number of wells was always specified, though in some cases the well type was also optimized (e.g., deviated and branched wells were considered in some cases). Comparisons to optimizations using a genetic algorithm (GA) were presented and, as noted above, PSO was shown to consistently outperform the GA considered. In one of the examples in [21], the well pattern optimization followed by well-by-well perturbation was compared to an unconstrained optimization that used $3N_w$ decision variables (the latter is referred to as the 'concatenation' approach). For this case the two-stage optimization consistently outperformed the concatenation approach. Taken in total, the results in [20,21] display the applicability of PSO for well placement optimization problems, as well as the potential advantages of the well pattern description and the two-stage optimization procedure.

It will clearly be useful to combine the well control optimization described in Section 2.3 with the field development optimization considered here. This coupled optimization problem will be computationally demanding, but the solutions provided can be expected to outperform those determined through the sequential application of the two procedures. Work along these lines is currently underway.

2.5 Assimilation of Reservoir Data (Inverse Modeling)

The reliability of oil production forecasts, and the 'optimal' strategy that is determined based on these predictions, depend strongly on the proper calibration of the reservoir simulation model. In essence, this calibration aims at finding appropriate model parameters given a number of observations. The two model parameters that (in many cases) most directly impact reservoir flow are permeability and porosity. Both of these parameters vary spatially. For a given rock type, which is denoted as facies in this context, porosity and permeability are often correlated, and one can be estimated from the other. In this work, the calibration parameter is taken to be the facies in each grid block, and we assume that each facies corresponds to a particular permeability and porosity.

Historic flow production represents one set of observed data. Such data are crucial because it is precisely the prediction of the reservoir flow response that is the ultimate purpose of the modeling. However, production data provides direct information only at well locations (though of course the flow rates and pressures observed at wells are impacted by reservoir properties outside the well region). In contrast to production data, seismic measurements (such as diffraction tomography) provide more global information and thus can be used to improve estimates of the spatial distribution of rock properties. Here we consider as observable data both flow and seismic measurements.

The use of observational data to infer reservoir properties is an inverse problem. As such, we anticipate that the solution will be non-unique. This is typically the case because there are more parameters to estimate than there are independent measurements, so many combinations of parameters yield similar model responses. In addition to the underspecified nature of the problem, additional complications arise from the approximations used in the forward modeling and from the presence of noise in the data. Uncertainty quantification/assessment involves finding multiple solutions of the inverse problem in order to generate a collection of production forecasts. For more information on data assimilation under uncertainty in this context, refer to, e.g., [73,45,74].

2.5.1 Problem Statement

The solutions of a geophysical inverse problem are the set of geological models that, when forward-modeled to provide simulation data, match the observations to within some tolerance. Since the approach here, as shown below, involves formulating the data assimilation process in optimization terms, any model configuration (set of inversion parameters) will be denoted by $\mathbf{x} \in \Omega \subset \mathbb{R}^n$, and Ω is the set of admissible models. The admissibility criteria can be formulated with respect to geological consistency. Geological consistency typically implies a particular spatial correlation of parameters (e.g., a given spatial covariance). The model $\mathbf{x}$ in this work represents the facies type associated with every grid block. Thus, the number of optimization variables n is on the order of the number of grid blocks in the discretized reservoir model (which can be very large in practical models).

From an optimization perspective, the inverse problem can be stated as follows

$$\min_{\mathbf{x} \in \Omega} \|\mathbf{O}(\mathbf{x}) - \mathbf{y}\|^2, \qquad (2.10)$$

where $\mathbf{y} \in \mathbb{R}^m$ are the observations and $\mathbf{O}(\mathbf{x}) \in \mathbb{R}^m$ represent the numerically-simulated observations. All the observable data considered are concatenated in $\mathbf{y}$ and $\mathbf{O}(\mathbf{x})$. Thus, if $\mathbf{O}_1(\mathbf{x}) \in \mathbb{R}^{m_1}$ and $\mathbf{O}_2(\mathbf{x}) \in \mathbb{R}^{m_2}$ are the two sets of observable data considered, then $\mathbf{O}(\mathbf{x}) = [\mathbf{O}_1(\mathbf{x}), \mathbf{O}_2(\mathbf{x})]$, with $m_1 + m_2 = m$. In the norm (Euclidean in this work), we can account for data uncertainty and include weights for the different sets of data. Since the observable data in this work are normalized, weights are taken to be unity. We reiterate that there are typically a much larger number of inversion parameters than there are independent measurements ($n \gg m$), and therefore the optimization problem in (2.10) is frequently ill-conditioned.

2.5.2 Methodologies for Data Assimilation

The optimization problem in (2.10) presents a number of challenges in addition to ill-conditioning. The cost function requires costly simulations, and in many cases derivative information is expensive to obtain or not available. The number of optimization variables is often large and the objective function can be non-smooth due to, for example, the presence of noise in the observations. These difficulties can be addressed by means of the following strategies.

The integration of disparate data in reservoir modeling has been suggested in a number of publications (e.g., [75,76,77]) as a means to alleviate the ill-conditioned character of (2.10). In essence, the use of different data types provides a degree of regularization for the inverse problem. Here, as in [32], we use as observable data oil and water production rates and diffraction tomography data. These data sets are complementary since they measure system responses on different spatial and temporal scales.

We can also expect a better conditioned optimization problem if the number of parameters is decreased. Instead of searching in n dimensions, we consider a subspace of dimension n_R. This subspace selection is not arbitrary and essentially aims at reducing the correlation between inversion parameters. The parameter reduction used here is based on principal component analysis (PCA), or the Karhunen-Loève transform, and can also be interpreted from a data compression perspective. The statistical information needed is generally obtained from a prior (rough) knowledge of the reservoir properties, and provides the inversion with geological consistency. We essentially follow the PCA-based parameter reduction technique used in [6], though that approach considers only flow production data and is invasive with respect to the flow simulator (thus it is very efficient but requires source-code access to implement).

In the example in Section 2.5.3, both production and seismic measurements provide the observable data. We reduce the number of optimization variables and introduce geological consistency through principal component analysis, and we approach (2.10) by derivative-free local optimization with an initial guess selected by a heuristic procedure that is based on information obtained by PCA. The use of numerical derivatives and a global procedure (GA) are also considered for comparison. All of these black-box approaches are more demanding computationally than an invasive adjoint-based gradient procedure but, as mentioned above, can be significantly accelerated through distributed computing. We briefly present below the fundamentals of PCA, since that transformation is a key component of our methodology.

2.5.2.1 Parameter Reduction Using Principal Component Analysis

Principal component analysis (PCA) optimally selects a subspace of dimension n_R from a larger space of dimension n. Given N possible models sampled from Ω, the region of the search space where plausible optimal solutions are expected, $\{\mathbf{x}_k\}_{k=1}^{N} \subset \Omega \subset \mathbb{R}^n$, PCA seeks an affine transformation

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{n_R} \left(\mathbf{s}_i^T \left(\mathbf{x}_k - \mu\right)\right) \mathbf{s}_i + \mu\,,$$

with $\mu \in \mathbb{R}^n$ and the set $\{\mathbf{s}_i\}_{i=1}^{n_R} \subset \mathbb{R}^n$ orthonormal. We note that this transformation is essentially an orthogonal projection. PCA is optimal in the sense that the Euclidean reconstruction error $||\hat{\mathbf{x}}_k - \mathbf{x}_k||_2$, averaged over $\{\mathbf{x}_k\}_{k=1}^{N}$, is minimized (or, equivalently, that the average reconstruction energy is maximized).

The optimal solution [78] implies that μ is the average of the N models sampled $\{\mathbf{x}_k\}_{k=1}^{N}$, and that each $\mathbf{s}_i$ is an eigenvector for the covariance matrix associated with these models. Additionally, it can be seen that the covariance matrix for the n_R PCA coefficients for $\{\mathbf{x}_k\}_{k=1}^{N}$ is a diagonal matrix, and that the contribution to the average reconstruction error from each of the PCA basis components $\mathbf{s}_i$ is equal to the corresponding eigenvalue.

The selection of the N models $\{\mathbf{x}_k\}_{k=1}^{N}$ is crucial and is done based on prior information. If these models provide an acceptable representation of Ω, a large part of the n_R-dimensional search space will provide solutions that are (in this case, geologically) consistent. Therefore, PCA not only reduces the search space, but also helps to ensure that the solutions obtained are practically acceptable. The value n_R is typically much smaller than n. Low values of n_R yield low-dimensional search spaces that are easier to explore, but the reconstruction error can be unacceptably large. In other words, the optimal search would take place only in a small part of Ω, and thus the solutions obtained in that reduced space may be clearly suboptimal. The determination of the appropriate value for n_R is application specific and is typically done through numerical experimentation.

A ranking for the PCA components can be established based on their respective eigenvalues – the higher the eigenvalue, the higher the rank (and thus the importance) of the associated PCA basis vector. This, together with the fact that the covariance matrix for the PCA coefficients is diagonal, suggests that a sequence of one-dimensional optimizations aimed at computing coefficients for the highest-rank PCA basis vectors may be beneficial in the overall optimization. Based on this observation, a heuristic PCA-based procedure for computing the initial guess in (2.10) can be obtained (please consult [32] for details).

2.5.3 Data Assimilation Example

The case study is taken from [32] and is based on a ten-layer synthetic model (with $20 \times 20 \times 10 = 4000$ cells) extracted from the Stanford VI reservoir model [79]. This approach provides a good framework for comparing inversion methodologies since the true model is known. We simulate a five-spot well pattern (four injectors in the corners, and one producer in the center of the domain; see Figure 10(a)). The optimization variable $\mathbf{x}$ is a binary facies indicator in every grid block (designating the block as either sand or shale). Though this variable is binary valued, it can be relaxed to a continuous variable. Thus, a value of 0.5 indicates that in the corresponding grid block, sand and shale are distributed equally.

The observable production data consists of the total field cumulative oil production and water injection, obtained at intervals of ten days up to 90 days (therefore, $m_1 = 10 + 10 = 20$). The production data are computed by solving the (discretized) reservoir flow equations. Here we use Stanford's general purpose research simulator (GPRS; [36,37]). The permeability and porosity fields are functions of the facies parameter $\mathbf{x}$. Given a (real-valued) facies parameter for grid block i, designated x_i, we compute the associated porosity ϕ_i by the following expression

$$\phi_i(x_i) = \phi_0 \exp(x_i \ln(\phi_1/\phi_0)),$$

where the coefficients ϕ_0 and ϕ_1 are the porosity values associated with the shale and sand facies (in practice, these values can be determined through measurements on rock cores or regression). We relate the block permeability k_i to the porosity ϕ_i using the Kozeny-Carman equation (see, e.g., [80])

$$k_i(\phi_i) = \alpha \frac{\phi_i^3}{(1-\phi_i)^2},$$

with the parameter α calculated from measurements or regression.

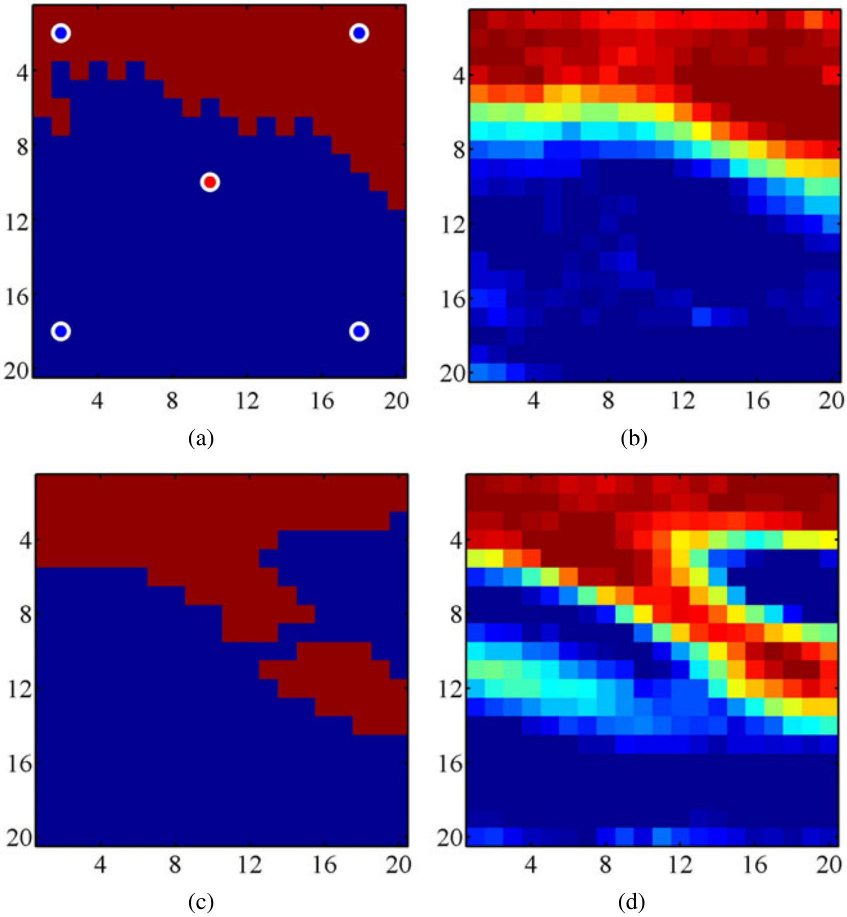


Fig. 2.10 Layer 4 from (a) the true model studied in Section 2.5 (injection and production wells are indicated as blue and red circles, respectively), (b) corresponding reconstruction after PCA with $N = 1000$ realizations and $n_R = 30$, (c) model selected randomly from the set of $N = 1000$ realizations, and (d) corresponding reconstruction after PCA with $n_R = 30$. Red and blue represent sand and shale facies, respectively. The original facies model is binary-valued, but after PCA it becomes continuous (from [32]); see online version for colors.

The second set of observable data is derived from crosswell diffraction tomography. In crosswell tomography, sound wave sources are placed in one (usually vertical) well and recorded and placed in another well (typically some hundred meters away). By recording the waves propagating from one well to another, it is possible to reconstruct approximately the structure of the earth in between the wells. The estimated earth image is sometimes called a crosswell section. In this example we have two crosswell sections obtained by associating diagonally the injectors in the five-spot pattern in Figure 10(a). Each section involves the ten layers in the model and is discretized by a 20×20 matrix of velocities (hence, $m_2 = 400 + 400 = 800$). The tomographic data along these two perpendicular crosswell sections are computed only once, after 90 days. The seismic observable data depends on certain rock properties (elastic bulk modulus and density) which in turn are functions of the fluid saturations at each grid block [80]. The input for the seismic tomography simulator thus includes the model $\mathbf{x}$, which provides the porosity for each grid block, and fluid saturations. These quantities, together with rock physics models, are used to compute the elastic velocities [80]. In all tomography calculations, both for the observations and during optimization, a simplified geometry for the top of the reservoir is considered, and the associated corrections are not included.

A priori knowledge of the reservoir geology, in the form of a so-called training image [81], together with facies data obtained at the well locations, allow the generation of $N = 1000$ geologically consistent model realizations, all conditioned to the prior information. These models are generated using a multipoint geostatistical algorithm [81], which can represent complex spatial structures. Through application of PCA to these 1000 realizations we reduce the number of inversion parameters from $n = 4000$ to $n_R = 30$. In Figure 2.10 we show two of these models (one of the ten model layers is shown) and their corresponding reconstructions. For our application, these PCA reconstructions are acceptable.

2.5.3.1 Inversion Results and Prediction

We compare here sequential quadratic programming (SQP) using numerical gradients with generalized pattern search (GPS), Hooke-Jeeves direct search (HJDS), and a genetic algorithm (GA). The initial guess for the local optimizers is computed as outlined above (see [32] for details). The GA population is 60 individuals and the algorithm is run for 100 generations. The initial population in the GA does not contain the initial guess taken for the local optimizers. In this way, we can test if GA can be beneficial in cases when useful initial guesses are not available. The distributed computing environment consists of a cluster with 48 nodes, and it is used for the SQP, GPS and GA optimizations. Each observable data value is assigned random noise with an amplitude of 5% of the standard deviation of the corresponding data type.

The models determined through inversion are shown in Figure 2.11. These results are for the same layer as shown in Figure 2.10, though they are generally representative for all ten layers in the model. As noted earlier, after PCA the original binary facies model is continuous (it could be transformed back to binary values using thresholding if necessary). It is evident that all of the methods provide reasonable models. A carefully selected initial guess is crucial for obtaining acceptable inversion results with the SQP, GPS and HJDS methods. Our process for determining the initial guess relies on some heuristics and therefore is not fully general, though it appears adequate for this case. The GA result appears slightly less accurate than the others, though the main model features are captured.

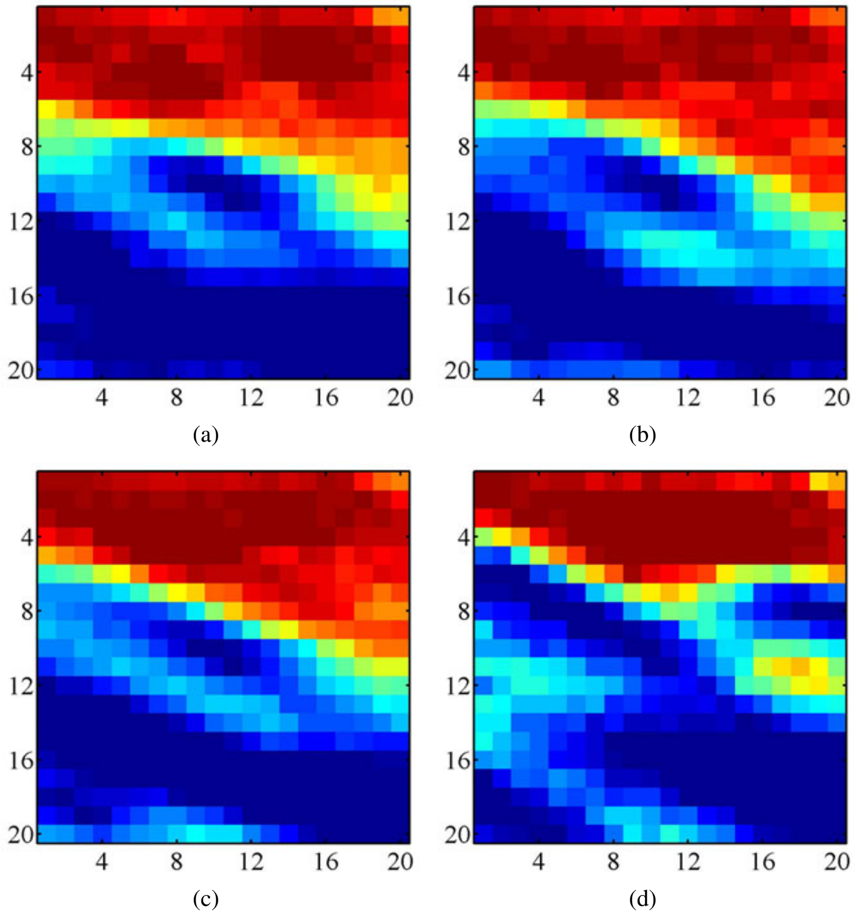


Fig. 2.11 Inverse model results for layer 4 of the reservoir section studied in Section 2.5. Facies distribution obtained by (a) sequential quadratic programming, (b) generalized pattern search, (c) Hooke-Jeeves direct search, and (d) a genetic algorithm. The genetic algorithm, because of its global nature, does not require an initial guess. The true distribution for layer 4 is shown in Figure 10(a). Red and blue represent sand and shale facies, respectively. Though the original facies model is binary-valued, after PCA it becomes continuous (from [32]); see online version for colors.

Figure 2.12 illustrates the performance of the local optimizers used for this problem. In this plot the horizontal axis is the number of equivalent simulations, which is defined as the total number of simulations divided by the speedup obtained by the parallel implementation. The concept of equivalent simulation is used to enable comparisons, in terms of elapsed time (not total computation time), between HJDS and the other (parallel) procedures. Since HJDS is inherently serial, for that algorithm the number of equivalent simulations coincides with the total number of

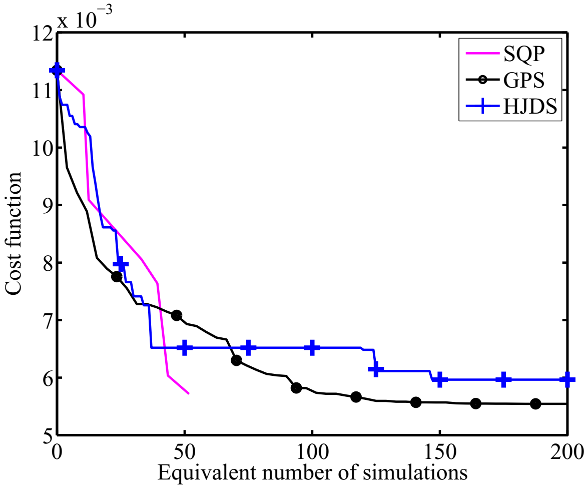
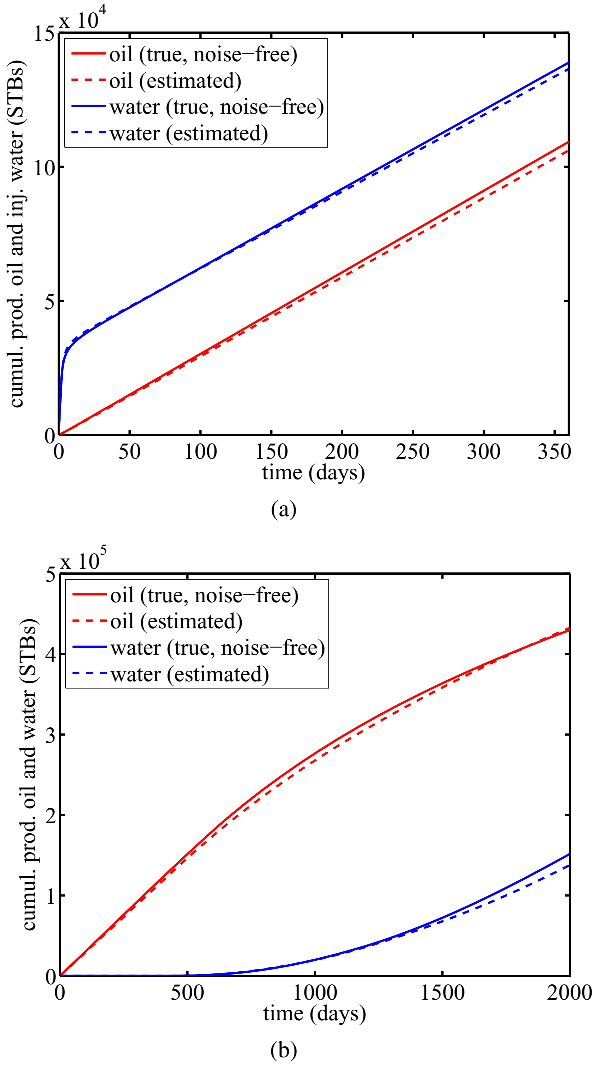


Fig. 2.12 Performance results for the local optimizers studied in the model inversion in Section 2.5 (from [32]); see online version for colors.

simulations. Note that one simulation involves calls to both the flow and seismic tomography simulators and that the initial guess computation for the local optimizers requires roughly five equivalent simulations. It is evident from Figure 2.12 that SQP provides the most efficient performance for this case. However, we expect that SQP performance would degrade if the cost function was less smooth. If the comparison was made in terms of total computation time, HJDS would be the most efficient algorithm for this problem (HJDS would thus be the method of choice in the absence of distributed computing resources).

The best individual in the initial GA population had a cost function of 0.036. After around 200 equivalent function evaluations, the objective function for GA decreased to about 0.006, though more gradually than for the other methods shown in Figure 2.12. This performance is promising since the GA was run without providing any initial guess as input. If a larger population is used, GA can explore the global search space and, as a consequence, potentially identify multiple solutions that are comparable in terms of the cost function. These solutions could then be used for uncertainty assessment.

The oil production and water injection forecasts over 360 days, for the model obtained using SQP, are shown in Figure 13(a) (we note that the inversion involved data over only the first 90 days). Agreement is generally very close, though slight mismatches are evident at later times, and these mismatches grow with time. In order to achieve accuracy over long simulation periods (up to 2000 days), the solution determined by SQP was adjusted as follows. A new data assimilation was performed after the first 1000 days. The observable data considered were the cumulative production of oil and water, together with two new crosswell tomographies at the end



(a)

(b)

Fig. 2.13 (a) Oil production and water injection forecast (360 days) for the solution obtained by SQP. (b) Oil and water production forecast (2000 days) for the solution recalibrated after 1000 days. In both cases the noise in the observations has been removed (from [32]); see online version for colors.

of the interval. The calibration at 1000 days started with the previously determined model (as shown in Figure 2.11) and it involves only one additional parameter (λ). This parameter simply scales globally the facies distribution; i.e., the new model is given by $\lambda\mathbf{x}$, where $\mathbf{x}$ is the (old) model obtained using data for the first 90 days.

The attendant one-dimensional optimization problem in λ required approximately two additional equivalent simulations.

Figure 13(b) shows predictions from the new model for oil and water production over 2000 days. The model provides accurate predictions over the entire period. This type of recalibration can be done in practice whenever the deviation between the prediction and the corresponding data is larger than some acceptable tolerance. Since this new calibration is performed using a solution calculated previously, the number of parameters considered can be relatively small. Alternative approaches, in which more parameters (or the entire model) are computed, could also be applied.

2.6 Concluding Remarks

In this chapter we have applied derivative-free optimization methods to three different problems relevant to oil field operations. The examples considered are representative of a wide range of practical simulation-based optimization problems and involve oil production optimization with general operating constraints, field development using a well pattern description, and data assimilation based on flow and seismic measurements. These problems involved continuous, integer and categorical variables, and the search spaces contained at most 100 dimensions. The successful use of derivative-free methods for these problems clearly demonstrates that these algorithms are viable for a range of oil field applications.

The derivative-free algorithms studied include generalized pattern search, Hooke-Jeeves direct search, a genetic algorithm, and particle swarm optimization. In order to enable additional comparisons, we also tested a gradient-based method, sequential quadratic programming, with derivatives estimated numerically. With the exception of Hooke-Jeeves direct search, all of these procedures can be readily parallelized and as such benefit immensely when implemented in a distributed manner. When parallel computing resources are limited or nonexistent, Hooke-Jeeves direct search represents a promising serial derivative-free optimization strategy.

The performance of derivative-free approaches depends strongly on the dimension of the search space, and for the computational resources typically available, these approaches are applicable when the number of optimization variables is on the order of a few hundred or less. Therefore, it may be necessary in some occasions to combine these approaches with some type of parameter reduction strategy. In this work, in one case we limited the size of the search space by restricting wells to be located within patterns, while in another case we applied principal component analysis to reduce the number of inversion parameters.

There are still a number of challenges related to the problems considered in this chapter. Though categorical (decision) variables were included in the optimal field development example presented in Section 2.4, a comprehensive study on the use and limitations of derivative-free algorithms for this type of mixed-integer nonlinear optimization problem would be of great interest. In addition, further comparisons between local and global methods, and the development of hybrid procedures, will also be useful. It will be beneficial to jointly address field development optimization and well control optimization, as the optimal well locations will in general depend on how the wells are operated. Multi-objective optimization may be of interest for this and other applications.

The efficient treatment of uncertainty in all of the problems considered is also a topic of great importance. Data assimilation methodologies that generate multiple solutions consistent with observed data are required. Optimization techniques that can efficiently handle multiple models are also needed. Finally, because the forward simulations required for our optimization methods are themselves often very time-consuming, the development of fast and reliable surrogate models will be of great use. Research in many of these areas is currently underway.

Acknowledgements. We are grateful to the industry sponsors of the Stanford Smart Fields Consortium and the Stanford Center for Reservoir Forecasting for partial funding of this work, and to the Stanford Center for Computational Earth and Environmental Science for providing distributed computing resources. We also thank Obiajulu J. Isebor (Stanford University), Jérome E. Onwunalu (now at BP) and Eduardo T. F. Santos (now at CEFET-BA) for their contributions to this work.

References

1. Jansen, J.D., Brouwer, D.R., Naevdal, G., van Kruijsdijk, C.P.J.W.: Closed-loop reservoir management. First Brea 23, 43–48 (2005)
2. Pironneau, O.: On optimum design in fluid mechanics. J. Fluid Mech 64, 97–110 (1974)
3. Ramirez, W.F.: Application of Optimal Control Theory to Enhanced Oil Recovery. Elsevier, Amsterdam (1987)
4. Brouwer, D.R., Jansen, J.D.: Dynamic optimization of waterflooding with smart wells using optimal control theory. SPE Journal 9(4), 391–402 (2004)
5. Zandvliet, M., Handels, M., van Essen, G., Brouwer, R., Jansen, J.D.: Adjoint-based well-placement optimization under production constraints. SPE Journal 13(4), 392–399 (2008)
6. Sarma, P., Durlofsky, L.J., Aziz, K., Chen, W.H.: Efficient real-time reservoir management using adjoint-based optimal control and model updating. Computational Geosciences 10, 3–36 (2006)
7. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPSSeries on Optimization, MPS-SIAM (2009)
8. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Review 45(3), 385–482 (2003)
9. Wright, M.H.: Direct Search methods: once scorned, now respectable. In: Griffiths, D.F., Watson, G.A. (eds.) Numerical Analysis 1995 (Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis). Pitman Research Notes in Mathematical Series, pp. 191–208. CRC Press, Boca Raton (1995)
10. Meza, J.C., Martinez, M.L.: On the use of direct search methods for the molecular conformation problem. Journal of Computational Chemistry 15, 627–632 (1994)
11. Booker, A.J., Dennis Jr., J.E., Frank, P.D., Moore, D.W., Serafini, D.B.: Optimization using surrogate objectives on a helicopter test example. In: Borggaard, J.T., Burns, J., Cliff, E., Schreck, S. (eds.) Computational Methods for Optimal Design and Control, pp. 49–58. Birkhäuser, Basel (1998)

12. Marsden, A.L., Wang, M., Dennis Jr., J.E., Moin, P.: Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. Journal of Fluid Mechanics 572, 13–36 (2003)
13. Duvigneau, R., Visonneau, M.: Hydrodynamic design using a derivative-free method. Structural and Multidisciplinary Optimization 28, 195–205 (2004)
14. Fowler, K.R., Reese, J.P., Kees, C.E., Dennis Jr., J.E., 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., Kolda, T.G.: Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems. Advances in Water Resources 31(5), 743–757 (2008)
15. Oeuvray, R., Bierlaire, M.: A new derivative-free algorithm for the medical image registration problem. International Journal of Modelling and Simulation 27, 115–124 (2007)
16. Marsden, A.L., Feinstein, J.A., Taylor, C.A.: A computational framework for derivative-free optimization of cardiovascular geometries. Computational Methods in Applied Mechanics and Engineering 197, 1890–1905 (2008)
17. Cullick, A.S., Heath, D., Narayanan, K., April, J., Kelly, J.: 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, October 5–8 (2009)
18. Velez-Langs, O.: Genetic algorithms in oil industry: an overview. Journal of Petroleum Science and Engineering 47, 15–22 (2005)
19. Artus, V., Durlofsky, L.J., Onwunalu, J., Aziz, K.: Optimization of nonconventional wells under uncertainty using statistical proxies. Computational Geosciences 10, 389–404 (2006)
20. Onwunalu, J., Durlofsky, L.J.: Application of a particle swarm optimization algorithm for determining optimum well location and type. Computational Geosciences 14, 183–198 (2010)
21. Onwunalu, J., Durlofsky, L.J.: A new well pattern optimization procedure for large-scale field development. SPE Journal (in press)
22. Bittencourt, A.: Optimizing Hydrocarbon Field Development Using a Genetic Algorithm Based Approach. PhD thesis. Dept. of Petroleum Engineering, Stanford University (1997)
23. Bittencourt, A.C., Horne, R.N.: Reservoir development and design optimization. SPE paper 38895 presented at the 1997 SPE Annual Technical Conference and Exhibition, San Antonio, Texas, October 5–8 (1997)
24. Yeten, B., Durlofsky, L.J., Aziz, K.: Optimization of nonconventional well type, location and trajectory. SPE Journal 8(3), 200–210 (2003)
25. Harding, T.J., Radcliffe, N.J., King, P.R.: Optimization of production strategies using stochastic search methods. SPE paper 35518 presented at the 1996 European 3-D Reservoir Modeling Confe rence, Stavanger, Norway, April 16–17 (1996)
26. Almeida, L.F., Tupac, Y.J., Lazo Lazo, J.G., Pacheco, M.A., Vellasco, M.M.B.R.: Evolutionary optimization of smart-wells control under technical uncertainties. SPE paper 107872 presented at the, Latin American & Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina, April 15–18 (2007)
27. Carroll III, J.A.: Multivariate production systems optimization. Master's thesis, Dept. of Petroleum Engineering, Stanford University (1990)
28. Echeverría Ciaurri, D., Isebor, O.J., Durlofsky, L.J.: Application of derivative-free methodologies for generally constrained oil production optimization problems. International Journal of Mathematical Modelling and Numerical Optimisation (in press)

29. Schulze-Riegert, R.W., Axmann, J.K., Haase, O., Rian, D.T., You, Y.-L.: Evolutionary algorithms applied to history matching of complex reservoirs. SPE Reservoir Evaluation & Engineering 5(2), 163–173 (2002)
30. Ballester, P.J., Carter, J.N.: A parallel real-coded genetic algorithm for history matching and its application to a real petroleum reservoir. Journal of Petroleum Science and Engineering 59, 157–168 (2007)
31. Maschio, C., Campane Vidal, A., Schiozer, D.J.: A framework to integrate history matching and geostatistical modeling using genetic algorithm and direct search methods. Journal of Petroleum Science and Engineering 63, 34–42 (2008)
32. Echeverria, D., Mukerji, T.: A robust scheme for spatio-temporal inverse modeling of oil reservoirs. In: Anderssen, R.S., Braddock, R.D., Newham, L.T.H. (eds.) Proceedings of the 18th World IMACS Congress and MODSIM 2009 International Congress on Modelling and Simulation, pp. 4206–4212 (2009)
33. Dadashpour, M., Echeverria Ciaurri, D., Mukerji, T., Kleppe, J., Landrø, M.: A derivative-free approach for the estimation of porosity and permeability using time-lapse seismic and production data. Journal of Geophysics and Engineering 7, 351–368 (2010)
34. Aziz, K., Settari, A.: Petroleum Reservoir Simulation. Kluwer Academic Publishers, Dordrecht (1979)
35. Gerritsen, M.G., Durlofsky, L.J.: Modeling fluid flow in oil reservoirs. Annual Review of Fluid Mechanics 37, 211–238 (2005)
36. Cao, H.: Development of Techniques for General Purpose Simulators. PhD thesis, Dept. of Petroleum Engineering, Stanford University (2002)
37. Jiang, Y.: Techniques for Modeling Complex Reservoirs and Advanced Wells. PhD thesis, Dept. of Energy Resources Engineering, Stanford University (2007)
38. Streamsim Technologies Inc., 3DSL v2.30 User Manual (2006)
39. Stewart, R.R.: Exploration Seismic Tomography: Fundamentals. Course Notes Series, Society of Exploration Geophysicists (1991)
40. Devaney, A.J.: Geophysical diffraction tomography. IEEE Transactions on Geoscience and Remote Sensing 22(1), 3–13 (1984)
41. Harris, J.M.: Diffraction tomography with arrays of discrete sources and receivers. IEEE Transactions on Geoscience and Remote Sensing 25(4), 448–455 (1987)
42. Chernov, L.A.: Wave Propagation in a Random Medium. McGraw-Hill, New York (1960)
43. Aki, K., Richards, P.: Quantitative Seismology. W.H. Freeman, New York (1980)
44. van Essen, G.M., van den Hof, P.M.J., Jansen, J.D.: Hierarchical long-term and short-term production optimization. SPE Journal (in press)
45. Oliver, D.S.: Multiple realizations of the permeability field from well test data. SPE Journal 1, 145–154 (1996)
46. Sarma, P., Durlofsky, L.J., Aziz, K.: Kernel principal component analysis for efficient, differentiable parameterization of multipoint geostatistics. Mathematical Geosciences 40, 3–32 (2008)
47. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Heidelberg (2006)
48. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Review 47(1), 99–131 (2005)
49. Torczon, V.: On the convergence of pattern search algorithms. SIAM Journal on Optimization 7(1), 1–25 (1997)
50. Audet, C., Dennis Jr., J.E.: Analysis of generalized pattern searches. SIAM Journal on Optimization 13(3), 889–903 (2002)

51. Audet, C., Dennis Jr., J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization 17(1), 188–217 (2006)
52. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. Journal of the ACM 8(2), 212–229 (1961)
53. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
54. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, pp. 39–43 (1995)
55. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Joint Conference on Neural Networks, pp. 1942–1948 (1995)
56. Clerc, M.: Particle Swarm Optimization. ISTE Ltd (2006)
57. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)
58. Helwig, S., Wanka, R.: Theoretical analysis of initial particle swarm behavior. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 889–898. Springer, Heidelberg (2008)
59. Carlisle, A., Dozier, G.: An off-the-shelf PSO. In: Proceedings of the 2001 Workshop on Particle Swarm Optimization, pp. 1–6 (2001)
60. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Mathematical Programming 91, 239–269 (2000)
61. Wächter, A., Biegler, T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106, 25–57 (2006)
62. Audet, C., Dennis Jr., J.E.: A pattern search filter method for nonlinear programming without derivatives. SIAM Journal on Optimization 14(4), 980–1010 (2004)
63. Abramson, M.A.: NOMADm version 4.6 User's Guide. Dept. of Mathematics and Statistics, Air Force Institute of Technology (2007)
64. Christie, M.A., Blunt, M.J.: Tenth SPE comparative solution project: a comparison of upscaling techniques. SPE Reservoir Evaluation & Engineering 4, 308–317 (2001)
65. Isebor, O.J.: Constrained production optimization with an emphasis on derivative-free methods. Master's thesis, Dept. of Energy Resources Engineering, Stanford University (2009)
66. Franstrom, K.L., Litvak, M.L.: Automatic simulation algorithm for appraisal of future infill development potential of Prudhoe Bay. SPE paper 59374 presented at the, SPE/DOE Improved Oil Recovery Symposium, Tulsa, Oklahoma, April 3–5 (2000)
67. Güyagüler, B., Horne, R.N.: Uncertainty assessment of well placement optimization. SPE Reservoir Evaluation & Engineering 7(1), 24–32 (2004)
68. Bangerth, W., Klie, H., Wheeler, M.F., Stoffa, P.L., Sen, M.K.: On optimization algorithms for the reservoir oil well placement problem. Computational Geosciences 10, 303–319 (2006)
69. Litvak, M., Gane, B., Williams, G., Mansfield, M., Angert, P., Macdonald, C., McMurray, L., Skinner, R., Walker, G.J.: Field development optimization technology. SPE paper 106426 presented at the 2007 SPE Reservoir Simulation Symposium, Houston, Texas, February 26–28 (2007)
70. Tupac, Y.J., Faletti, L., Pacheco, M.A.C., Vellasco, M.M.B.R.: Evolutionary optimization of oil field development. SPE paper 107552 presented at the, SPE Digital Energy Conference and Exhibition, Houston, Texas, April 11–12 (2007)
71. Spall, J.C.: An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins APL Technical Digest 19(4), 482–492 (1998)

72. Mattot, L.S., Rabideau, A.J., Craig, J.R.: Pump-and-treat optimization using analytic element method flow models. Advances in Water Resources 29, 760–775 (2006)
73. Tarantola, A.: Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM, Philadelphia (2005)
74. Caers, J., Hoffman, T.: The probability perturbation method: a new look at Bayesian inverse modeling. Mathematical Geology 38(1), 81–100 (2006)
75. Gosselin, O., van den Berg, S., Cominelli, A.: Integrated history matching of production and 4D seismic data. SPE paper 71599 presented at the 2001 SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, September–30 October–3 (2001)
76. Waggoner, J.R., Cominelli, A., Seymour, R.H.: Improved reservoir modeling with time-lapse seismic in a Gulf of Mexico gas condensate reservoir. SPE paper 77514 presented at the, SPE Annual Technical Conference and Exhibition, San Antonio, Texas, September–29 October–2 (2002)
77. Aanonsen, S.I., Aavatsmark, I., Barkve, T., Cominelli, A., Gonard, R., Gosselin, O., Kolasinski, M., Reme, H.: Effect of scale dependent data correlations in an integrated history matching loop combining production data and 4D seismic data. SPE paper 79665 presented at the, SPE Reservoir Simulation Symposium, Houston, Texas, February 3–5 (2003)
78. Miranda, A.A., Le Borgne, Y.A., Bontempi, G.: New routes for minimal approximation error to principal components. Neural Processing Letters 27(3), 197–207 (2008)
79. Castro, S.: A Probabilistic Approach to Jointly Integrate 3D/4D Seismic, Production Data and Geological Information for Building Reservoir Models. PhD thesis, Dept. of Energy Resources Engineering, Stanford University (2007)
80. Mavko, G., Mukerji, T., Dvorkin, J.: The Rock Physics Handbook, 2nd edn. Cambridge University Press, Cambridge (2009)
81. Strebelle, S.: Conditional simulation of complex geological structures using multi-point statistics. Mathematical Geology 34, 1–21 (2002)