Noname manuscript No.
(will be inserted by the editor)

Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow

Drosos Kourounis · Louis J. Durlofsky · Jan Dirk Jansen · Khalid Aziz

Received: date / Accepted: date

Abstract An adjoint formulation for the gradient-based optimization of oil-gas compositional reservoir simulation problems is presented. The method is implemented within an automatic differentiation-based compositional flow simulator (Stanford's AD-GPRS). The development of adjoint procedures for general compositional problems is much more challenging than for oil-water problems due to the increased complexity of the code and the underlying physics. The treatment of nonlinear constraints, an example of which is a maximum gas rate specification in the production wells or injector wells, when the control variables are well bottom-hole pressures, poses a particular challenge. Two approaches for handling these constraints are presented – a formal treatment within the optimizer, and a simpler heuristic treatment in the forward model. The relationship between discrete and continuous adjoint formulations is also elucidated. Results for three example cases are presented. Improvements in the objective function (cumulative oil produced) relative to reference solutions range from 4.2% to 8.8%. The heuristic treatment of nonlinear constraints is shown to offer a cost-effective means for obtaining feasible solutions, which are in many cases better than those obtained using the formal constraint handling procedure.

Keywords Adjoint formulation · gradient-based optimization · production optimization · compositional reservoir simulation · discrete adjoint · continuous adjoint · automatic differentiation · nonlinear constraints · general constraints

1 Introduction

The determination of optimal time-varying well settings, such as injection and production rates or bottom-hole pressure, is an important aspect of production optimization. Both gradient-based and derivative-free methods have been considered for this problem, and both are applicable in different settings. When the simulator source code is accessible, a gradient-based optimization method, in which the gradient is computed using an adjoint formulation, is often the method of choice since it is generally the most efficient.

In this paper we implement an adjoint formulation for compositional reservoir simulation problems. Procedures of this type entail the application of optimal control theory and have their roots in the calculus of variations [6, 32]. Adjoint-based optimization techniques have been used in a reservoir simulation setting both for history matching (see, e.g., [8, 9, 20, 28, 31]) and for production optimization. Much of the early work on their use for production optimization was performed by Ramirez and coworkers, who considered the optimization of several different enhanced oil recovery (EOR) processes [22, 23, 29]. In subsequent work, the focus was on gradient-based optimization (and in some cases on the

D. Kourounis
Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305-2220, USA
Current address: Institute of Computational Science, Faculty of Informatics, Università della Svizzera italiana, CH-6904 Lugano, Switzerland
E-mail: drosos.kourounis@usi.ch

L. J. Durlofsky
Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305-2220, USA
E-mail: lou@stanford.edu

J. D. Jansen
Department of Geoscience and Engineering, Delft University of Technology, Delft, Netherlands
E-mail: j.d.jansen@tudelft.nl

K. Aziz
Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305-2220, USA
E-mail: aziz@stanford.edu

optimization of 'smart wells') for water flooding [1, 5, 31, 33, 34]. Recent studies have addressed the implementation of adjoint-based procedures into general purpose simulators, the treatment of general constraints, and regularization and other numerical issues [12, 18, 21, 30]. Refer to [19] for a more complete overview of adjoint-based optimization methods. We note additionally that, although not considered here, derivative-free methods can also be applied for production optimization problems – see [13] for discussion and examples.

Although much of the early (1980s) work noted above focused on the application of adjoint procedures for EOR problems, there has not been much work on the use of adjoint techniques for large-scale (practical) compositional reservoir simulation problems. This is likely due to the complexity entailed in implementing adjoint procedures into a general purpose compositional reservoir simulator and to the challenging computational problems that must be solved to perform the optimizations. Compositional simulation is inherently more challenging than black-oil simulation because of the need to perform phase-equilibrium (flash) calculations for all grid blocks at every time step, and because multiple hydrocarbon components, rather than just oil and gas, are frequently tracked. Adjoint formulations are challenging to code because they require analytical derivatives of many variables, and the increased complexity of compositional simulators renders these derivatives much more cumbersome to calculate than in the case of a black-oil simulator.

In this work, we implement an adjoint treatment for multicomponent oil-gas compositional systems through use of a recently developed automatic differentiation capability [40]. The application of automatic differentiation in the context of Stanford's General Purpose Research Simulator (AD-GPRS) [7], a modular simulator with many advanced features, enables us to construct a gradient-based optimization framework suitable for use in compositional problems. Our formulation includes the treatment of bound, linear and nonlinear constraints. Along these lines, we consider two different treatments for the nonlinear constraints: a formal treatment within the optimizer, and a heuristic approach, where bound constraints are treated in the optimization and nonlinear constraints are satisfied in the forward model.

Although the results we actually present are for a discrete adjoint formulation, we have also developed a continuous adjoint formulation. In a discrete implementation, the governing equations for the so-called adjoint system are constructed based on the discretized-in-time forward model equations. In continuous formulations, by contrast, the adjoint equations are formed from the continuous forward model. Recent formulations for production optimization have generally been based on discrete formulations, and Brouwer and Jansen [5] reviewed previous work and concluded that the discrete adjoint method was preferable. In other application areas, however, both methods have been used and comparisons have been reported (e.g., [2, 25]). In particular, Nadarajah and Jameson [25], who studied a shape optimization problem in supersonic flow, concluded that the continuous adjoint formulation provides more accurate gradients in the presence of large discretization errors, which are often present in the vicinity of shock waves. Because shocks also occur in multicomponent reservoir simulation, we formulate and code a continuous treatment to enable a comparison.

This paper proceeds as follows. In Section 2, we present the equations governing oil-gas compositional flow and briefly describe the solution of the forward problem. Next, in Section 3, we develop both the discrete and continuous formulations for the adjoint problem. Some details of the numerical treatment are also discussed. The handling of nonlinear constraints in the optimizer is discussed in Section ??, and in Section 5 where we also introduce the heuristic approach within the forward model. Numerical results demonstrating the capabilities of our optimization procedure, for a series of heterogeneous two and three-dimensional problems involving different numbers of hydrocarbon components and wells, are presented in Section 6. Conclusions and suggestions for future work are provided in Section 7.

2 Oil-gas compositional simulation equations

The mass conservation equation for component i, which can exist in any phase j (here $j = o,\ g$, where o indicates oil and g gas), is given by [7, 35, 36]:

$$\frac{\partial}{\partial t}\left(\phi \sum_j x_{ij}\rho_j S_j\right) - \nabla \cdot \left(\sum_j x_{ij}\rho_j \mathsf{K}\frac{k_{rj}}{\mu_j}\nabla \Phi_j\right) + \sum_w \sum_j x_{ij}\rho_j q_j^w = 0, \quad i = 1,\ldots,n_c. \tag{2.1}$$

In the first (accumulation) term, t is time, ϕ is porosity, x_{ij} designates the mole fraction of component i in phase j, S_j is saturation, and ρ_j is molar density. In the second (flow) term, K is the permeability tensor, k_{rj} is the relative permeability to phase j, μ_j the phase viscosity, and the phase potential Φ_j is given by $\Phi_j = p_j - \rho_j g(D - D^0)$, where p_j is phase pressure, D is depth, D^0 is a reference depth, and g is gravitational acceleration. In the third (source/sink) term, q_j^w indicates the phase flow rate for well w. The treatment of this term will be discussed in Section 5. Note, equation (2.1) is written for each of the n_c components present in the system.

For a mixture of n_c components in two fluid phases (oil and gas), thermodynamic equilibrium can be expressed as:

$$f_{io}(p_o,x_{io}) - f_{ig}(p_g,x_{ig}) = 0, \qquad (2.2)$$

where $f_{io}(p_o,x_{io})$ is the fugacity of component i in the oil phase and $f_{ig}(p_g,x_{ig})$ is the fugacity of component i in the gas phase (temperature does not appear because the system is assumed to be isothermal). We additionally must satisfy the saturation constraint ($S_o + S_g = 1$) and the component mole fraction constraints:

$$\sum_{i=1}^{n_c} x_{i0} - 1 = 0, \qquad \sum_{i=1}^{n_c} x_{ig} - 1 = 0. \qquad (2.3)$$

A capillary pressure relationship also appears in cases with nonzero capillary pressure, though here we neglect capillary pressure so $p_o = p_g$.

As discussed by many authors (see, e.g., [7, 11, 35, 38]), the system described above contains a total of only n_c primary equations and primary variables per grid block. These equations and variables are coupled (from block to block), and in a fully-implicit method are all computed simultaneously at each Newton iteration. The remaining (secondary) variables can be computed locally (block by block), and thus very efficiently, once the primary variables are determined. Various options exist for the choice of primary variables (see [35] for discussion). Here we use the so-called natural variable set, which includes, for each grid block, one pressure unknown, $n_p - 1$ saturation unknowns (where n_p is the number of phases; here $n_p = 2$), and $n_c - n_p$ component mole fraction unknowns.

In our formulation, the governing equations (2.1) are solved fully-implicitly, using a backward-Euler time discretization, two-point flux approximation, and single-point upwinding [3]. These treatments are standard in practical reservoir simulation. For the solution of the set of nonlinear equations, we use Newton's method with the solution at the previous time step as the initial guess. A limit on the change of the grid-block saturation and mole fractions over a Newton iteration is applied [40]. The Newton iterations terminate when the maximum relative norm of the residual is less than 10^{-6} (tight convergence criteria are required for the adjoint solution, discussed below). For the solution of the linear system at each Newton iteration we use GMRES preconditioned by the constrained pressure residual method, as described in [18]. Iteration is terminated when the Euclidean norm of the initial residual has decreased by five orders of magnitude.

We employ a simple time stepping strategy. The time step size at step $n+1$ is a multiple of that at n, provided nonlinear convergence was achieved at step n. In this way the time step can increase until it reaches the maximum allowable. If the nonlinear solver fails to converge within a prescribed number of Newton iterations, we divide the time step by a fixed constant. This process is repeated until the nonlinear system converges.

3 Adjoint equations for the compositional system

We now present the discrete and continuous adjoint equations. Some code and numerical issues are also discussed.

3.1 Automatic differentiation

It is quite common for comprehensive computational platforms, in reservoir simulation and other application areas, to undergo frequent modification and enhancement. This poses a problem for adjoint formulations because, when an existing feature is modified the corresponding adjoint code may also be impacted, and when a new feature is added, the associated adjoint code must (in many cases) be created. The maintenance and development of adjoint code poses challenges because the necessary derivatives are frequently complicated. This is particularly the case in compositional simulation where variables couple in many ways, including through the nonlinear equation of state.

Automatic differentiation, or AD, is a recent development in the field of scientific computing that is gaining popularity as a means of facilitating the development and enhancement of large code bases. AD enables, for example, the fast (analytical) determination of Jacobian matrix elements from the code defining the residual vector. The use of AD has allowed the fast construction and assessment of different compositional formulations within the same code [36]. In this work, we take advantage of AD to automate the construction of many of the derivatives required for the adjoint formulation.

The AD implementation used in our compositional simulator is the 'automatic differentiation expression templates library' (ADETL), developed originally by Younis et al. [39]. This library generates efficient computer code for the evaluation of the Jacobian matrix and the corresponding partial derivatives from discrete algebraic expressions of the governing conservation equations, associated constraint relations, and equations of state. We refer to [39] for a detailed description of the underlying theory.

3.2 Discrete adjoint formulation

Following the fully-implicit discretization of the governing equations (using the usual finite volume method, with treatments as noted above), we can express the nonlinear system as:

$$\mathbf{p}_n(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{u}_n) = \mathbf{0}, \tag{3.1}$$

where $\mathbf{p}_n$ denotes the fully discretized, both in space and time, set of equations. Here $\mathbf{x}_n = \mathbf{x}(t_n)$ and $\mathbf{u}_n = \mathbf{u}(t_n)$ are the states and controls (well settings), respectively, at time step n. The corresponding time step size is designated Δt_n.

We are interested in either maximizing or minimizing an objective function f that is in general a nonlinear function of the states $\mathbf{x}_n$ and the controls $\mathbf{u}_n$ of the forward problem. We assume that f has the following form:

$$f(\mathbf{x},\mathbf{u}) = \int_{t_0}^{t_N} F\left(\mathbf{x}(t), \mathbf{u}(t)\right) dt + M(\mathbf{x}(t_N)), \tag{3.2}$$

where $F(\mathbf{x}(t), \mathbf{u}(t))$ is a nonlinear function and $M(\mathbf{x}(t_N))$ is a function of only the last state $\mathbf{x}_N$. After the solution of the forward problem has been obtained, f may be approximated by

$$f \approx \sum_{n=1}^{N} \Delta t_n F_n\left(\mathbf{x}_n, \mathbf{u}_n\right) + M(\mathbf{x}_N). \tag{3.3}$$

Using (3.3) we can state the optimal control problem as:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & f = \sum_{n=1}^{N} \Delta t_n F_n\left(\mathbf{x}_n, \mathbf{u}_n\right) + M(\mathbf{x}_N) \\ \text{subject to} \quad & \mathbf{p}_n(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{u}_n) = \mathbf{0}, \\ & \mathbf{x}_0 = \mathbf{x}(t_0) \end{aligned} .$$

In general, a number of linear and nonlinear constraints may need to be included in the optimal control problem. We postpone the discussion of their treatment until Section 5. Now, since $\mathbf{p}_n = \mathbf{0}$, we can introduce the augmented objective function f_A by 'adjoining' the governing equations to the original objective function f. The new objective f_A shares the same extrema as f and is defined as:

$$f_A = \sum_{n=1}^{N} \left(\Delta t_n F_n(\mathbf{x}_n, \mathbf{u}_n) + \boldsymbol{\lambda}_n^T \mathbf{p}_n(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{u}_n) \right) + M(\mathbf{x}_N). \tag{3.4}$$

In (3.4), the vectors $\boldsymbol{\lambda}_n$ are the Lagrange multipliers.

The maximum or minimum of f_A (and thus f) is achieved when the first variation of f_A is zero ($\delta f_A = 0$). After performing some index-shifting, and grouping terms multiplied by the same variation ($\delta\mathbf{x}_n, \delta\mathbf{x}_N, \delta\mathbf{u}_n$), δf_A can be written as:

$$\begin{aligned} \delta f_A = & \left(\frac{\partial M_N}{\partial \mathbf{x}_N} + \Delta t_N \frac{\partial F_N}{\partial \mathbf{x}_N} + \boldsymbol{\lambda}_N^T \frac{\partial \mathbf{p}_N}{\partial \mathbf{x}_N} \right) \delta \mathbf{x}_N \\ & + \sum_{n=1}^{N-1} \left(\Delta t_n \frac{\partial F_n}{\partial \mathbf{x}_n} + \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{p}_{n+1}}{\partial \mathbf{x}_n} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{p}_n}{\partial \mathbf{x}_n} \right) \delta \mathbf{x}_n \\ & + \sum_{n=1}^{N} \left(\Delta t_n \frac{\partial F_n}{\partial \mathbf{u}_n} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{p}_n}{\partial \mathbf{u}_n} \right) \delta \mathbf{u}_n. \end{aligned} \tag{3.5}$$

In order to achieve $\delta f_A = 0$, we require $\delta f_A / \delta \mathbf{x}_n = \mathbf{0}$ (for $n = 1, 2, \ldots, N$) and $\delta f_A / \delta \mathbf{u}_n = \mathbf{0}$. To satisfy $\delta f_A / \delta \mathbf{x}_n = \mathbf{0}$ for $n = 1, 2, \ldots, N$, we require that the Lagrange multipliers satisfy the following equations:

$$\frac{\partial \mathbf{p}_n^T}{\partial \mathbf{x}_n} \boldsymbol{\lambda}_n = -\left(\frac{\partial \mathbf{p}_{n+1}^T}{\partial \mathbf{x}_n} \boldsymbol{\lambda}_{n+1} + \Delta t_n \frac{\partial F_n^T}{\partial \mathbf{x}_n} \right), \tag{3.6}$$

$$\frac{\partial \mathbf{p}_N^T}{\partial \mathbf{x}_N} \boldsymbol{\lambda}_N = -\left(\Delta t_N \frac{\partial F_N^T}{\partial \mathbf{x}_N} + \frac{\partial M_N^T}{\partial \mathbf{x}_N} \right). \tag{3.7}$$

With this choice of the Lagrange multipliers the total variation becomes

$$\delta f_A = \sum_{n=1}^{N} \left(\Delta t_n \frac{\partial F_n}{\partial \mathbf{u}_n} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{p}_n}{\partial \mathbf{u}_n} \right) \delta \mathbf{u}_n,$$

and the gradient of the objective function with respect to the controls that should be forwarded to the optimizer is

$$\frac{\delta f_A}{\delta \mathbf{u}} = \left[\frac{\delta f_1}{\delta \mathbf{u}_1}, \frac{\delta f_2}{\delta \mathbf{u}_2}, \ldots, \frac{\delta f_N}{\delta \mathbf{u}_N} \right], \tag{3.8}$$

where the individual entries of $\delta f_A / \delta \mathbf{u}$ are given by

$$\frac{\delta f_n}{\delta \mathbf{u}_n} = \Delta t_n \frac{\partial F_n}{\partial \mathbf{u}_n} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{p}_n}{\partial \mathbf{u}_n}, \quad n = 1, 2, \ldots, N. \tag{3.9}$$

By driving $\delta f_A / \delta \mathbf{u}$ to zero, we achieve the minimum or maximum of f_A (and thus f). In practice, $\delta f_A / \delta \mathbf{u}$, along with other quantities related to constraints, are provided to a gradient-based optimization algorithm to determine the next estimate for the controls $\mathbf{u}$.

In reservoir simulation, the control variables of the wells are not defined per time step but for larger periods of time the so called control steps. Time steps sizes can be usually much smaller than control step sizes not only because of the inability of Newton to converge to the solution of the nonlinear system for large time step sizes, but mostly because one would like to capture the dynamics and reduce the discretisation error associated with the time variable. The gradient at the mth control period $\delta f_n / \delta \mathbf{u}_m$ is the sum of the gradients $\delta f_n / \delta \mathbf{u}_n$ for all time step indices n that belong to the mth control period.

3.3 Continuous adjoint formulation

The continuous adjoint formulation employs the continuous representation of the objective function along with the spatially discretized reservoir equations. The optimal control problem can then be stated as:

$$\begin{aligned} &\underset{\mathbf{u}}{\text{minimize}} \quad f(\mathbf{x},\mathbf{u}) = \int_{t_0}^{t_N} F(\mathbf{x}(t),\mathbf{u}(t))\,dt + M(\mathbf{x}(t_N)) \\ &\text{subject to} \quad \mathbf{p}(\dot{\mathbf{x}}(t),\mathbf{x}(t),\mathbf{u}(t)) = \mathbf{0}. \end{aligned}$$

In this case we express the governing set of partial differential equations, for a specified dynamic well-control strategy $\mathbf{u}(t)$, as $\mathbf{p}(\dot{\mathbf{x}}(t),\mathbf{x}(t),\mathbf{u}(t)) = \mathbf{0}$. We introduce the Lagrange multipliers $\boldsymbol{\lambda}(t)$ and define the Lagrangian L as:

$$L(\dot{\mathbf{x}},\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = F(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^T\mathbf{p}(\dot{\mathbf{x}},\mathbf{x},\mathbf{u}), \tag{3.10}$$

The variables $\mathbf{u}(t)$, $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$ and $\boldsymbol{\lambda}(t)$ were denoted as $\mathbf{u}$, $\mathbf{x}$, $\dot{\mathbf{x}}$ and $\boldsymbol{\lambda}$ to simplify notation. The augmented objective function, f_A, can be expressed as:

$$f_A(\dot{\mathbf{x}},\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = \int_{t_0}^{t_N} L(\dot{\mathbf{x}},\mathbf{x},\mathbf{u},\boldsymbol{\lambda})\,dt + M(\mathbf{x}_N). \tag{3.11}$$

The first variation of f_A is given by

$$\begin{aligned} \delta f_A = &\int_{t_0}^{t_N} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\delta\dot{\mathbf{x}} + \frac{\partial L}{\partial \mathbf{x}}\delta\mathbf{x} + \frac{\partial L}{\partial \mathbf{u}}\delta\mathbf{u} + \frac{\partial L}{\partial \boldsymbol{\lambda}}\delta\boldsymbol{\lambda} \right) dt \\ &+ \frac{\partial M(\mathbf{x}_N)}{\partial \mathbf{x}_N}\delta\mathbf{x}_N. \end{aligned} \tag{3.12}$$

Note that $\delta\dot{\mathbf{x}} = d(\delta\mathbf{x})/dt$, so any variation in the state vector $\mathbf{x}$ will introduce a variation in its time derivative $\dot{\mathbf{x}}$.

After integration by parts, using the fact that the variation of the initial conditions $\delta\mathbf{x}_0 = \mathbf{0}$, and taking into account that $\partial L^T/\partial\boldsymbol{\lambda} = \mathbf{p}(\dot{\mathbf{x}},\mathbf{x},\mathbf{u}) = \mathbf{0}$, the first variation of f_A can be written as:

$$\begin{aligned} \delta f_A = &\int_{t_0}^{t_N} \left(\frac{\partial L}{\partial \mathbf{x}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) \delta\mathbf{x}\,dt \\ &+ \left(\frac{\partial L(\mathbf{x}_N)}{\partial \dot{\mathbf{x}}_N} + \frac{\partial M(\mathbf{x}_N)}{\partial \mathbf{x}_N} \right) \delta\mathbf{x}_N \\ &+ \int_{t_0}^{t_N} \frac{\partial L}{\partial \mathbf{u}}\delta\mathbf{u}\,dt. \end{aligned} \tag{3.13}$$

To achieve $\delta f_A/\delta\mathbf{x} = \mathbf{0}$, $\boldsymbol{\lambda}$ must be chosen to satisfy the following:

$$\frac{d}{dt}\left(\frac{\partial \mathbf{p}^T}{\partial \dot{\mathbf{x}}}\boldsymbol{\lambda} \right) - \frac{\partial \mathbf{p}^T}{\partial \mathbf{x}}\boldsymbol{\lambda} - \frac{\partial F^T}{\partial \mathbf{x}} = \mathbf{0} \tag{3.14}$$

$$\frac{\partial \mathbf{p}_N^T}{\partial \dot{\mathbf{x}}_N}\boldsymbol{\lambda}_N = -\frac{\partial M^T(\mathbf{x}_N)}{\partial \mathbf{x}_N}. \tag{3.15}$$

The ordinary differential equation in (3.14) is integrated backwards in time starting from the final time condition (3.15). With the resulting $\boldsymbol{\lambda}$, the first variation of the objective function becomes:

$$\delta f_A = \int_{t_0}^{t_N} \left(\frac{\partial F}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{p}}{\partial \mathbf{u}} \right) \delta\mathbf{u}\,dt. \tag{3.16}$$

In order to allow a direct comparison of the discrete and continuous adjoint formulations, we integrate the continuous adjoint backwards in time fully implicitly, using the same scheme as is applied for the forward problem. The discrete form of (3.14) is:

$$\frac{\partial \mathbf{p}_n^T}{\partial \mathbf{x}_n}\boldsymbol{\lambda}_n = -\frac{\partial \mathbf{p}_{n+1}^T}{\partial \mathbf{x}_n}\boldsymbol{\lambda}_{n+1} - \Delta t_n\frac{\partial F_n^T}{\partial \mathbf{x}_n}. \tag{3.17}$$

This equation is solved backwards in time, starting from the boundary condition (3.15). Once the Lagrange multipliers have been obtained, the gradient is computed by equations (3.8), (3.9).

3.4 Continuous versus discrete adjoint formulation

The gradients obtained by the discrete adjoint formulation are, as would be expected, fully consistent with the discrete forward problem. Indeed, if we compute the gradients using numerical perturbation of the controls, we find that they coincide with those from the discrete adjoint solution in the first 5-8 significant digits (to achieve this level of agreement, tight tolerances must be used for linear and nonlinear convergence of the forward simulation). There are differences, however, between these gradients and those provided by the continuous adjoint formulation.

To illustrate this, consider the simplified case where $M(\mathbf{x}_N) = 0$. The solution of (3.15) in this case will give $\boldsymbol{\lambda}_N = \mathbf{0}$, and as a result, the first term in (3.5) will not vanish. There will be a nonzero term left multiplying the variation of $\delta\mathbf{x}_N$:

$$\delta f_A = \Delta t_N\frac{\partial F_N}{\partial \mathbf{x}_N}\delta\mathbf{x}_N + \sum_{n=1}^{N}\left(\Delta t_n\frac{\partial F_n}{\partial \mathbf{u}_n} + \boldsymbol{\lambda}_n^T\frac{\partial \mathbf{p}_n}{\partial \mathbf{u}_n} \right)\delta\mathbf{u}_n. \tag{3.18}$$

It is evident that, as the time step size $\Delta t_N \to 0$, the term multiplying $\delta\mathbf{x}_N$ will vanish, and the gradient provided by the continuous formulation will become consistent with that from the discrete problem. However, as long as Δt_N is significant, the two gradients will not coincide, especially at the last time step.

We implemented both the continuous and discrete adjoint formulations into our optimization framework. Using a small Δt_N, we observed that the computed gradients were very similar, consistent with (3.18). Even using small Δt_N, however, we did not observe any advantage of the continuous formulation over the discrete formulation. In cases where Δt_N was not small, the continuous formulation required more iterations of the optimizer, presumably because of errors in $\delta f_A/\delta\mathbf{u}$. In light of these observations, we do not present any detailed results using the continuous adjoint formulation since we do not see any advantages to this approach for our problem. We note that these finding are consistent with those reported in [17] and [37] for general Runge-Kutta time stepping methods, in [24], where the discrete and continuous adjoint approaches were applied to automatic aerodynamic optimization and in [26] for general variational inverse problems governed by partial differential equations. A similar gradient discrepancy between discretization/optimization versus optimization/discretization can also occur with respect to the spatial discretization – see the discussion on a shape optimization problem in [16].

3.5 Solution of adjoint equations

The solution of the linear system of equations that arises when solving (3.6) constitutes the largest computational demand in the adjoint problem. The matrix appearing in this equation at the time step n, $\partial\mathbf{p}_n^T/\partial\mathbf{x}_n$, is the transpose of the Jacobian matrix for the converged forward problem, $\partial\mathbf{p}_n/\partial\mathbf{x}_n$. In our implementation, the converged states are written to disk during the solution of the forward problem. These converged states are then read back, during the solution of the adjoint problem, and $\partial\mathbf{p}_n/\partial\mathbf{x}_n$ is reconstructed, along with all other derivatives appearing in equations (3.6), (3.7) and (3.9). This enables the evaluation of the Lagrange multipliers $\boldsymbol{\lambda}_n$ and the gradients $\partial f_n/\partial\mathbf{u}_n$.

For the solution of the linear system in (3.6), we use GMRES preconditioned by the transpose of the CPR (constrained pressure residual) preconditioner, as described in [18]. In these linear solutions, we require very high accuracy to guarantee that residual errors accumulated over hundreds of time steps will not pollute the gradients (which would influence the computed optimum). For this reason, we continue iterating the linear solver until the Euclidean norm of the initial residual has decreased by 10 orders of magnitude. This is significantly higher accuracy than is required for the forward problem.

4 Gradient-based optimization and related software

The SNOPT optimizer is used in this work for solving the nonlinear constrained optimization problem [15]. SNOPT uses a sparse sequential quadratic programming (SQP) algorithm that exploits sparsity in the constraint Jacobian and maintains a limited-memory quasi-Newton approximation to the Hessian of the Lagrangian. The QP subproblems are solved using an inertia-controlling reduced-Hessian active-set method (SQOPT) that allows for variables appearing linearly in the objective and constraint functions.

4.1 The SQP approach

Here we discuss the main features of an SQP method for solving a nonlinear program (NP). All features described here are readily specialized to general inequality constraints appearing in the definition of the general nonlinear program (GNP) discussed in section 5.1. In this section we take the problem to be

$$\text{(NP)}\qquad \begin{array}{ll} \underset{\mathbf{u}}{\text{minimize}} & f(\mathbf{u}) \\ \text{subject to} & \mathbf{c}(\mathbf{u}) \geq 0, \end{array}$$

where $\mathbf{u}\in\mathbb{R}^n$, $\mathbf{c}\in\mathbb{R}^m$, and the functions $f(\mathbf{u})$ and $\mathbf{c}(\mathbf{u})$ have continuous second derivatives. The gradient of f is denoted by the vector $\mathbf{g}(\mathbf{u})$, and the gradients of each element of $\mathbf{c}$ form the rows of the Jacobian matrix $\mathbf{J}(\mathbf{u})$.

An SQP method obtains search directions (for the primal and dual variables) from a sequence of QP subproblems. Each QP subproblem minimizes a convex quadratic model of a certain Lagrangian function subject to linearized constraints associated with (NP), namely,

$$\mathscr{L}(\mathbf{u},\mathbf{u}_k,\boldsymbol{\pi}_k) = f(\mathbf{u}) - \boldsymbol{\pi}_k^T\mathbf{d}_L(\mathbf{u},\mathbf{u}_k), \tag{4.1}$$

defined in terms of the constraint linearization $\mathbf{c}_L(\mathbf{u},\mathbf{u}_k)$ and the departure from linearity $\mathbf{d}_L(\mathbf{u},\mathbf{u}_k)$:

$$\begin{aligned} \mathbf{c}_L(\mathbf{u},\mathbf{u}_k) &= \mathbf{c}_k + \mathbf{J}_k(\mathbf{u}-\mathbf{u}_k), \\ \mathbf{d}_L(\mathbf{u},\mathbf{u}_k) &= \mathbf{c}(\mathbf{u}) - \mathbf{c}_L(\mathbf{u},\mathbf{u}_k), \end{aligned}$$

subject to linearized constraints. The first and second derivatives of the modified Lagrangian with respect to $\mathbf{u}$ are

$$\begin{aligned} \nabla\mathscr{L}(\mathbf{u},\mathbf{u}_k,\boldsymbol{\pi}_k) &= \mathbf{g}(\mathbf{u}) - (\mathbf{J}(\mathbf{u})-\mathbf{J}_k)^T\boldsymbol{\pi}_k, \\ \nabla^2\mathscr{L}(\mathbf{u},\mathbf{u}_k,\boldsymbol{\pi}_k) &= \nabla^2 f(\mathbf{u}) - \sum_i(\boldsymbol{\pi}_k)_i\nabla^2 c_i(\mathbf{u}). \end{aligned}$$

Observe that $\nabla^2\mathscr{L}$ is independent of $\mathbf{u}_k$ and is the same as the Hessian of the conventional Lagrangian. At $\mathbf{u}=\mathbf{u}_k$, the modified Lagrangian has the same function and gradient values as the objective: $\mathscr{L}(\mathbf{u}_k,\mathbf{u}_k,\boldsymbol{\pi}_k)=f_k$, $\nabla\mathscr{L}(\mathbf{u}_k,\mathbf{u}_k,\boldsymbol{\pi}_k)=\mathbf{g}_k$. The modified augmented Lagrangian is "less" nonlinear than the augmented Lagrangian itself because linear terms in the constraints disappear, especially in the quadratic penalty term. The "number of nonlinear variables" in the modified augmented Lagrangian is the same as in the original problem.

The merit function

$$\mathscr{M}_\rho(\mathbf{u},\boldsymbol{\pi},\mathbf{s}) = f(\mathbf{u}) - \boldsymbol{\pi}^T\big(\mathbf{c}(\mathbf{u})-\mathbf{s}\big) + \tfrac{1}{2}\sum_{i=1}^{m} \rho_i \big(c_i(\mathbf{u})-s_i\big)^2, \tag{4.2}$$

where ρ is a vector of penalty parameters, is reduced along each search direction to ensure convergence from any starting point.

The basic structure of an SQP method involves major and minor iterations. The major iterations generate a sequence of iterates $(\mathbf{u}_k,\boldsymbol{\pi}_k)$ that converge to the optimal solution $(\mathbf{u}^*,\boldsymbol{\pi}^*)$. At each iterate a QP subproblem is used to generate a search direction towards the next iterate $(\mathbf{u}_{k+1},\boldsymbol{\pi}_{k+1})$. Solving such a subproblem is itself an iterative procedure, with the minor iterations of an SQP method being the iterations of the QP method. SNOPT requires first-order derivatives of the nonlinear objective and constraint functions with respect to the control variables, which are provided by our adjoint procedure.

5 Nonlinear constraints

Many constraints appear as simple bound constraints (e.g., bottom hole pressure (BHP) limits in a problem where BHPs are the control variables), but in other cases the constraints are nonlinear since a (nonlinear) simulation is required to evaluate them. An example of this is the specification of maximum gas production rate (either for an individual well or for a group of wells) in a general compositional problem where the control variables are BHPs. In this section we discuss the handling of nonlinear constraints. We describe a detailed treatment within the optimizer, and an approximate heuristic treatment in the forward model.

5.1 Constraint handling in the optimizer

In the presence of nonlinear output inequality constraints, it is extremely hard to predict a-priori an initial guess for the control parameters which will not violate the desired bounds on the nonlinear output constraints. In such cases the optimizer needs to initiate a process (elastic mode) for finding a set of control parameters for which the values of the nonlinear constraints lie within the bounds. Several optimizers may also intentionally exit the feasible region, in an attempt to find better extrema of the objective function being extremized. For these reasons constraint and infeasibility handling are both indispensable architectural components of every general purpose optimizer. Here we described how these are implemented within SNOPT.

For simplifying notation lets us first write our constrained optimization problem in the more general form (GNP) assuming lower and upper bounds on the controls and the nonlinear constraints

$$\text{(GNP)} \qquad \underset{\mathbf{u}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{u})$$
$$\text{subject to} \quad \mathbf{b}_l \le \begin{pmatrix}\mathbf{u}\\ \mathbf{c}(\mathbf{u})\end{pmatrix} \le \mathbf{b}_u,$$

where $f(\mathbf{u})$ is a linear or nonlinear objective function, $\mathbf{c}(\mathbf{u})$ is a vector of nonlinear constraint functions $c_i(\mathbf{u})$ with sparse derivatives and $\mathbf{b}_l$ and $\mathbf{b}_u$ are vectors of lower and upper bounds. We assume that the nonlinear functions are smooth and that their first derivatives are available.

5.1.1 Infeasible constraints

SNOPT deals with infeasibility using ℓ_1 penalty functions. It solves (GNP) as given, using QP subproblems based on linearizations of the nonlinear constraints. If a QP subproblem proves to be infeasible or unbounded (or if the Lagrange multiplier estimates for the nonlinear constraints become large), SNOPT solves the problem

$$\text{(GNP}(\gamma)) \qquad \underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{minimize}} \quad f(\mathbf{u}) + \gamma\mathbf{e}^T(\mathbf{v}+\mathbf{w})$$
$$\text{subject to} \quad \mathbf{b}_l \le \begin{pmatrix}\mathbf{u}\\ \mathbf{c}(\mathbf{u})-\mathbf{v}+\mathbf{w}\end{pmatrix} \le \mathbf{b}_u,$$
$$\mathbf{v}\ge 0, \quad \mathbf{w}\ge 0,$$

where $f(\mathbf{u}) + \gamma\mathbf{e}^T(\mathbf{v}+\mathbf{w})$ is called a composite objective, and the penalty parameter γ ($\gamma \ge 0$) may take a finite sequence of increasing values. If (GNP) has a feasible solution and γ is sufficiently large, the solutions to (GNP) and (GNP(γ)) are identical. If (GNP) has no feasible solution, (GNP(γ)) will tend to determine a "good" infeasible point if γ is again sufficiently large. If γ were infinite, the nonlinear constraint violations would be minimized subject to the linear constraints and bounds.

This process may take many major iterations until a feasible initial set of control parameters is obtained. For this reason, optimization with simple bound constraints on the inputs, is usually expected to converge much faster than optimization with additional lower or upper bounds on output quantities.

5.1.2 Constraint lumping

A variety of methods have been proposed to incorporate nonlinear constraints in the adjoint formulation; see, e.g., [5, 19, 30, 31] for detailed discussion. Operational constraints in reservoir simulation are usually inequality constraints set on input and/or output rates at every time step and well. This means that the vector $\mathbf{c}(\mathbf{u})$ in (GNP) has size equal to the number of constraints per well times the number of time steps. Computing the gradients for each one of the entries of $\mathbf{c}(\mathbf{u})$ as it is required by SNOPT may be computationally infeasible. For instance, assuming that N time steps were performed during the forward simulation, for a constraint defined at time step $n, n \leq N$, n per well linear systems need to be solved for evaluating the gradient of that constraint. So in total if constrains are defined on N_w wells, $N_w N(N+1)/2$ linear systems needs to be solved for the evaluation of the gradient of these constraints. A more efficient way for approximating the same gradients can be obtained by lumping the constraints over the full simulation time frame [30]. This lumping can be performed on a well-by-well basis, in which case $N_w N$ linear systems need to be solved, or for the entire model where only N linear systems need to be solved for the evaluation of the gradient. In this way the approximate gradients of the lumped nonlinear constraints are obtained in the same manner as the gradient of the objective function, i.e., in terms of time-step contributions. Bound constraints on the controls do not require any special treatment as they are readily handled by the optimizer.

Constraints that are described by nondifferentiable functions can be challenging to incorporate. A constraint of this type that appears frequently in production optimization problems is the maximum (or minimum) well flow rate constraint, e.g.,

$$q_{jn} \leq q_{max}, \tag{5.1}$$

where q_{jn} is a rate of the jth well at time step n and q_{max} is a specified maximum rate. To satisfy this constraint one should guarantee that

$$\max_{j,n} (q_{jn}) \leq q_{max}. \tag{5.2}$$

However, the max function is a nondifferentiable funcion and it cannot be used to provide gradient information. We thus approximate it by a smooth function, specifically the one suggested by [4]:

$$\begin{aligned} c &= \max_{j,n}(q_{jn}) \approx \alpha \log Q, \\ Q &= \sum_{j=1}^{N_w} \sum_{n=1}^{N} e^{q_{jn}/\alpha}, \qquad (5.3) \\ \alpha &= 0.05 q_{max}, \qquad (5.4) \end{aligned}$$

where N_w is the number of wells and N the number of total time steps. With this definition of the constraint function, the gradient of the constraint with respect to the control variables, which is required by the optimizer, is obtained by:

$$\begin{aligned} \frac{\delta c}{\delta \mathbf{u}_n} &= \frac{1}{Q} \sum_{j=1}^{N_w} e^{q_{jn}/\alpha} \frac{\partial q_{jn}}{\partial \mathbf{u}_n} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{p}_n}{\partial \mathbf{u}_n}, \\ \frac{\partial q_{jn}}{\partial \mathbf{u}_n} &= \left[0, 0, \ldots, \frac{\partial q_{jn}}{\partial u_{jn}}, \ldots, 0\right], \end{aligned} \tag{5.5}$$

as soon as the Lagrange multipliers for the constraints $\boldsymbol{\lambda}_n$ have been computed from the solution of the following adjoint problem:

$$\frac{\partial \mathbf{p}_n^T}{\partial \mathbf{x}_n} \boldsymbol{\lambda}_n = -\frac{\partial \mathbf{p}_{n+1}^T}{\partial \mathbf{x}_n} \boldsymbol{\lambda}_{n+1} + \frac{1}{Q} \sum_{j=1}^{N_w} e^{q_{jn}/\alpha} \frac{\partial q_{jn}^T}{\partial \mathbf{x}_n}, \tag{5.6}$$

$$\frac{\partial \mathbf{p}_N^T}{\partial \mathbf{x}_N} \boldsymbol{\lambda}_N = -\frac{1}{Q} \sum_{j=1}^{N_w} e^{q_{jN}/\alpha} \frac{\partial q_{jN}^T}{\partial \mathbf{x}_N}. \tag{5.7}$$

The smaller the coefficient multiplying q_{max} in (5.3), the more accurate the approximation of max becomes. The numerical value used here, 0.05, ensures that no overflow occurs in any of the exponential terms in the summation. The approximation of max in (5.3) is always greater than the maximum of the component well rates, so if this maximum is honored, the true constraint is guaranteed to be satisfied.

5.2 Constraint handling in the simulator

A simpler way to render an infeasible solution feasible, when the constraints are upper and lower bounds on output quantities (e.g., rates when BHPs are specified), is to satisfy these constraints in the forward model. Indeed, during the forward simulation, wells can be controlled by specifying either the bottom-hole pressure or the rate. In the former case, the well rate q_j^w for phase j in block l is calculated from the well model

$$\left(q_j^w\right)_l = \left(T_w \frac{k_{rj}}{\mu_j}\right)_l (p_{w,l} - p_l), \tag{5.8}$$

where T_w is the well index (or well transmissibility), $p_{w,l}$ is the wellbore pressure for the well in block l, and p_l is the well-block pressure. When the total well rate is specified, equations of the form of (5.8) can be used to compute $p_{w,l}$. For more details on well models, and on the relationship between $p_{w,l}$ and BHP, see [7].

This flexibility in defining the controls of the wells allows some constraints to be easily satisfied during the forward simulation. For example, if a maximum gas production rate is specified and a well operating under BHP control violates this maximum, the well can be switched from BHP control to rate control and operated at the maximum rate. This treatment is illustrated in Fig. 5.1. The BHP (top left) is prescribed to be 50 bar throughout the simulation, but we also have a maximum gas rate of 5,000 m^3/d (indicated by the blue line in the upper right figure). The well operates at 50 bar for the first 480 days, at which point the maximum rate constraint is reached. The well then switches to operate at a gas rate of 5,000 m^3/d for the rest of the simulation. The resulting BHP and gas production rate profiles actually used in the simulation (and which honor the constraints) are shown in the bottom two figures.

In the results below, we apply this treatment and compare it to the formal approach described in Section 5.1. Using this procedure, which we refer to as 'heuristic constraint handling', we first perform the optimization without including the nonlinear constraints. Bound (linear) constraints are ignored during the optimization. Then, once this initial optimization has converged, we run the forward problem one more time using the 'optimized' BHPs, but this time the simulator is allowed to switch to rate control when required to satisfy the nonlinear constraints. Thus the computational effort for this approach is little more than that required for optimizing the bound (and linearly) constrained problem.

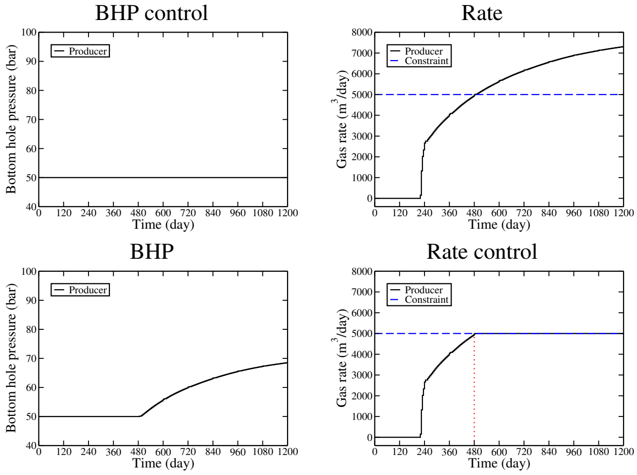

Fig. 5.1 Schematic illustrating heuristic constraint handling. Top: Constant BHP and resulting gas rate. Bottom: BHP and gas rate satisfying constraint.

Although it is clearly approximate, this heuristic constraint handling approach has some potential advantages over the formal method described in Section 5.1. For example, the heuristic treatment allows the simulator to switch controls at any time step in the simulation, while the formal approach only allows controls to switch at a relatively small number of control steps (by way of comparison, in a typical problem we may have $O(10^2-10^3)$ time steps but only $O(10)$ control steps). The heuristic approach thus enables, in some sense, a more 'fine-grained' response, and it can be viewed as having many more 'control' variables (though these variables are not optimized formally). Increasing the number of control steps to provide the same granularity to the optimizer as in the forward problem (i.e., setting the control step size to equal the time step size) should theoretically result in better performance by the formal approach, though in practice the significant increase in the number of control variables would likely make the control problem ill-posed and impact the performance of the optimizer. In the examples below, we will compare the performance of these two approaches for handling nonlinear constraints.

Table 6.1 Initial guesses for the optimizations for all cases considered

Run	Initial guess
1	$[p_I^l, p_P^l]$
2	$[p_I^l, p_P^a]$
3	$[p_I^l, p_P^u]$
4	$[p_I^a, p_P^l]$
5	$[p_I^a, p_P^a]$
6	$[p_I^a, p_P^u]$
7	$[p_I^u, p_P^l]$
8	$[p_I^u, p_P^a]$
9	$[p_I^u, p_P^u]$

We note finally that if rates are used as the control variables, then the rate constraints enter the optimization problem as simple bound constraints, which are easy to satisfy. In this case, however, the BHPs become nonlinear constraints. Our heuristic treatment would then entail the switch from rate control to BHP control if the BHP constraint would otherwise be violated. We did not test the performance of our procedure using rates as the control variables, but this should be considered in future work.

6 Numerical results

We will present results for three different cases. All involve bound and nonlinear constraints, and we will compare the performance of the two approaches described above for treating the nonlinear constraints. Because our gradient-based optimization will find only a local optimum, we run each case nine times, using a different initial guess for the well controls. Each initial guess corresponds to a combination of BHPs from the set $\{p_I^u, p_I^l, p_I^a\}$ for the injectors and from the set $\{p_P^u, p_P^l, p_P^a\}$ for the producers, where p^u, p^l and p^a designate the upper and lower limits on the initial BHPs, and the average between these limits, respectively. We set $p^l = p_{init} + 1$ bar for the injectors and $p^u = p_{init} - 1$ bar for the producers, where p_{init} is the initial reservoir pressure. Note that these 'limits' are simply used to prescribe initial guesses for the optimization – they are not related to the actual BHP bound constraints. For clarity, we will refer to each case by the number of the corresponding run, as listed in Table 6.1.

Table 6.2 Model parameters for Example 1

Grid size	20 × 20 × 1	
Parameter	Value	Units
Δx	24	m
Δy	24	m
Δz	4	m
Depth	4000	m
Initial pressure	100	bar
Temperature	100	°C
Rock compressibility	7.2×10^{-5}	1 / bar
Simulation time	256	d
Pressure upper bound	120	bar
Pressure lower bound	90	bar
Residual gas saturation	0	-
Residual oil saturation	0	-
End point rel perm gas	1	-
End point rel perm oil	1	-
Corey exponent gas	2	-
Corey exponent oil	2	-
Well locations [grid block no.]	i	j
Injector 1	1	1
Injector 2	1	20
Injector 3	20	1
Injector 4	20	20
Producer 1	10	12

Table 6.3 Fluid description for Example 1

Component	CO_2	C_1	C_4	C_{10}
Initial composition (%)	1	20	29	50
Injection composition (%)	100	-	-	-

6.1 Example 1 - Π obstacle

In the first example we maximize cumulative oil recovery under immiscible CO_2 injection. The two-dimensional geological model is the one depicted in Fig. 6.1. A Π-shaped impermeable layer is located at the center of the 20 × 20 grid-cell homogeneous reservoir. The permeability of the red grid-cells is set to 4000 mD, while the permeability for the blue grid-cells is set to 10^{-4} mD. Four injectors wells are placed at the four corners (coloured blue) and the only producer well is located inside the Π-shaped impermeable layer (coloured red). The model includes a total of four components (three hydrocarbon components plus CO_2), as specified in Table 6.3. Details about the reservoir model, the well locations and model parameters are provided in Table 6.2.

The control parameters in the optimization problem are the well BHPs. These are constrained to lie between an upper bound of 120 bar and a lower bound of 90 bar. We additionally specify maximum gas flow rate in the producers of 500 m^3/d at reservoir conditions. The total simulation period is 256 days, and the well controls are determined at initial time and for every subsequent 32-day interval. There are thus a total of eight control steps and 40 control parameters.

Two reference solutions are generated. First, we run the simulation with the production wells operating at the minimum BHP and the injection wells at the maximum BHP. This solution is infeasible because it violates the nonlinear output constraints. Next, we apply the heuristic constraint handling approach described above, with the maximum gas production rate set to 500 m^3/d. The cumulative oil production for these two cases is given in the first row ('Reference') of Table 6.4. The table headings refer to the treatment of the nonlinear constraints – bound constraints are satisfied in all cases.

We next perform optimizations that honor the bound constraints but not the nonlinear constraints. The results for the nine runs, starting from different initial guesses, are presented in Table 6.4 in the column labeled 'Unconstr.' The best optimum achieved is 179.8 m^3 of oil, obtained from Run 4. This clearly exceeds the reference result of 150.1 m^3. Results using the heuristic constraint handling are shown in the third column. Here the best result is 151.3 m^3 of oil (Run 5), which is by 0.78% worse than the reference. It is important to note that, for both the (nonlinearly) unconstrained and heuristic approaches, some of the runs give results that are below those from the reference cases (e.g., Run 1in both cases). This is presumably due to convergence to poor local minima. In the next set of runs we apply the formal constraint handling treatment. For these runs, the best optimum is 154.0 m^3 of oil (Run 3). This value is approximately 1% better than the reference.

In an attempt to increase oil production, we allow for more flexibility to the BHPs by letting them change every single day. This introduces 256 control intervals and in total 1280 control variables. Table 6.5 summarises our results. Although the solutions at the second column labeled 'Unconstr.' changed only slightly this time, on the contrary results using the heuristic constraint handling approach show an objective of 155.9 m^3 of oil (Run 8), which exceeds the reference by 2.2% unlike the case where we used only 40 control parameters and the heuristic treatment failed to produce an optimal result exceeding the reference solution. The highest optimal however is obtained here from the formal constraint treatment and it is 162.5 m^3 of oil (Run 4) exceeding the reference by 6.6%.

The oil production profiles for the best runs, along with the reference (heuristic) case, are shown in Fig. 6.2. Recall that we are maximizing cumulative oil, so the fact that early time production in the reference case exceeds that of the optimized cases is not of concern. The detailed BHP and gas rate profiles for each case are shown in Figs. 6.3, 6.4 and 6.5. The oil rates for all three cases are depicted in Fig. 6.6. Although the BHPs for the two optimized cases are clearly different, the oil rate profiles do show some similarities. For

Table 6.4 Oil production in 10^3 m^3 (Example 1, 40 control variables) for the optimized objective function without satisfying the nonlinear constraints ('Unconstr.'), satisfying the nonlinear constraints using the heuristic treatment ('Heuristic'), and satisfying the nonlinear constraints using the formal approach ('Formal'). Best feasible results shown in bold.

Controls 40	Unconstraint	Heuristic	Formal
Reference	150.1	152.5	
1	178.9	145.2	146.9
2	177.8	146.3	150.4
3	177.5	143.8	**154.0**
4	179.8	150.4	151.8
5	177.8	**151.3**	150.6
6	178.4	142.4	148.0
7	178.9	150.0	149.5
8	179.0	149.1	152.1
9	178.9	149.8	153.4

example, unlike the reference case where most of the oil production takes place at the first 120 days, the optimized cases maintain a constant (almost the same) average production rate up to the first 200 days.

It is important to note that the heuristic constraint handling approach is more efficient computationally than the formal treatment. On average, the formal approach required about three times the number of forward simulations used in the corresponding heuristically constrained cases. This large discrepancy results from the need to enforce feasibility within the optimizer in the formal constraint handling approach.

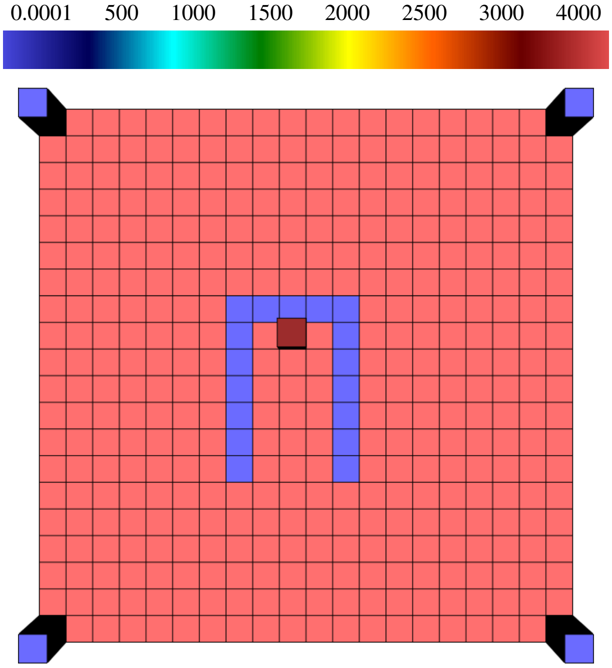

Fig. 6.1 Injection wells (blue) and production wells (red) for Example 1. Background shows K_x ($K_x = K_y$).

Table 6.5 Oil production in 10^3 m^3 (Example 1, 1280 control variables) for the optimized objective function without satisfying the nonlinear constraints ('Unconstr.'), satisfying the nonlinear constraints using the heuristic treatment ('Heuristic'), and satisfying the nonlinear constraints using the formal approach ('Formal'). Best feasible results shown in bold.

Controls 1280	Unconstr.	Heuristic	Formal
Reference	150.1	152.5	
1	174.9	134.7	149.7
2	178.2	107.7	134
3	172.1	107.5	94.9
4	178.7	145.3	**162.5**
5	179.0	147.3	152.1
6	178.8	148	151.1
7	179.3	153.1	159.7
8	180.0	**155.9**	150.9
9	179.7	139.3	152.9

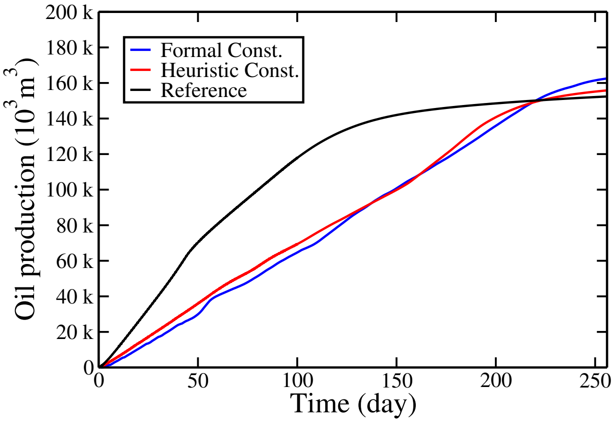

Fig. 6.2 Oil production versus time for Example 1. Results are for feasible reference case (black curve), best heuristically constrained solution (Run 8, red curve) and best formally constrained solution (Run 4, blue curve).

Table 6.6 Fluid description for Example 2

Component	CO_2	C_1	C_4	C_{10}
Initial composition (%)	1	20	29	50
Injection composition (%)	90	10	-	-

6.2 Example 2 - Top layer of SPE 10 model

In the second example we maximize cumulative oil recovery under immiscible CO_2 injection. The two-dimensional geological model is the top layer of the model defined in the SPE comparative solution project [10], referred to as SPE 10. The model includes a total of four components (three hydrocarbon components plus CO_2), as specified in Table 6.6. Details about the reservoir model, the well locations and model parameters are provided in Table 6.7.

The well locations, along with a map of the permeability field, are depicted in Fig. 6.7. The control parameters in the optimization problem are the well BHPs. These are constrained to lie between an upper bound of 150 bar and a

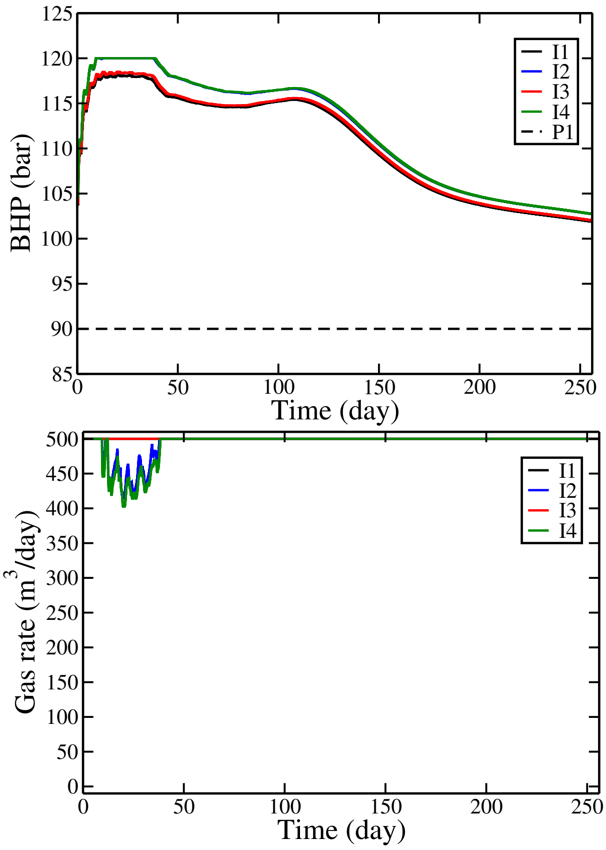

Fig. 6.3 BHPs (top), gas rates (middle) and oil rates (bottom) for the feasible reference solution (Example 1).

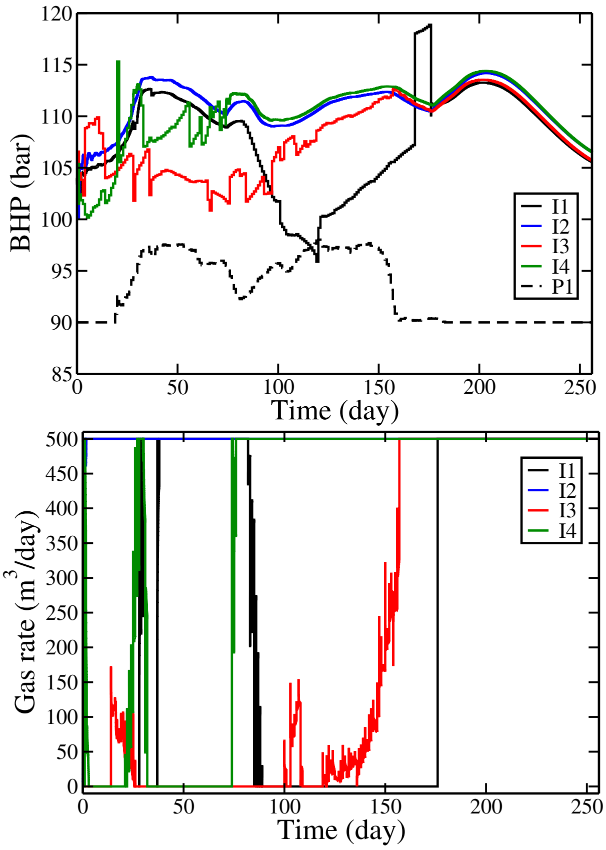

Fig. 6.4 BHPs (top), gas rates (middle) and oil rates (bottom) for the best heuristically constrained solution (Example 1, Run 9).

Table 6.7 Model parameters for Example 2

Grid size	60 × 220 × 1	
Parameter	Value	Units
Δx	6.096	m
Δy	3.048	m
Δz	0.6096	m
Depth	2574	m
Initial pressure	75	bar
Temperature	100	°C
Rock compressibility	7.2×10^{-5}	1 / bar
Simulation time	1000	d
Pressure upper bound	150	bar
Pressure lower bound	50	bar
Residual gas saturation	0	-
Residual oil saturation	0	-
End point rel perm gas	1	-
End point rel perm oil	1	-
Corey exponent gas	2	-
Corey exponent oil	2	-
Well locations [grid block no.]	i	j
Injector 1	58	9
Injector 2	58	126
Injector 3	2	67
Injector 4	2	211
Producer 1	2	3
Producer 2	58	67
Producer 3	2	143
Producer 4	58	210

Table 6.8 Oil production in 10^3 m^3 (Example 2) for the optimized objective function without satisfying the nonlinear constraints ('Unconstr.'), satisfying the nonlinear constraints using the heuristic treatment ('Heuristic'), and satisfying the nonlinear constraints using the formal approach ('Formal'). Best feasible results shown in bold.

Run	Unconstr.	Heuristic	Formal
Reference	20.29	20.28	
1	21.77	21.77	19.74
2	22.01	22.01	19.99
3	18.16	18.16	18.15
4	21.68	21.68	20.68
5	22.03	**22.03**	20.79
6	22.13	20.67	21.76
7	21.68	21.68	21.73
8	22.07	21.48	**22.07**
9	22.03	**22.03**	21.95

lower bound of 50 bar. We additionally specify maximum gas flow rate in the producers of 200 m^3/d at reservoir conditions. The total simulation period is 1000 days, and the well controls are determined at initial time and for every subsequent 100-day interval. There are thus a total of ten control steps and 80 control parameters.

For this example we generate two reference solutions. First, we run the simulation with the production wells operating at the minimum BHP and the injection wells at the max-

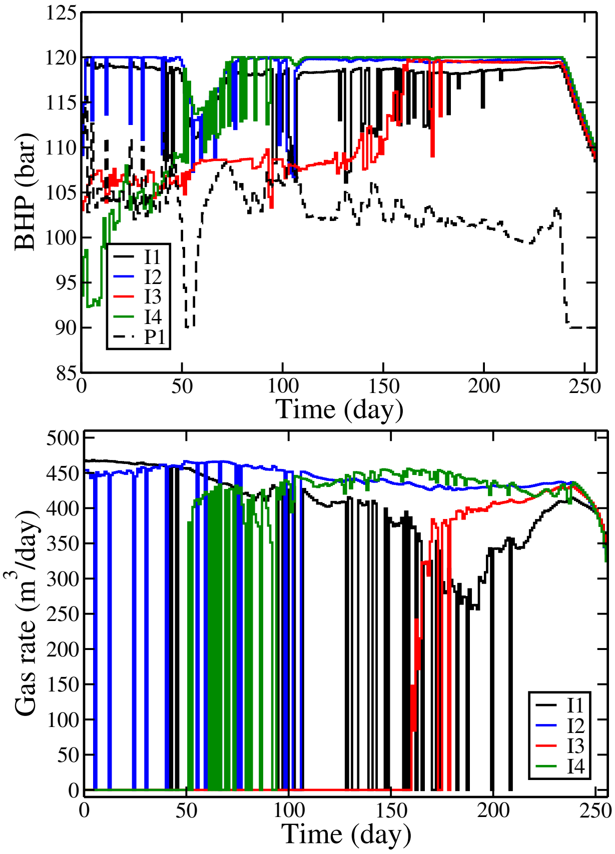

Fig. 6.5 BHPs (top), gas rates (middle) and oil rates (bottom) for the best formally constrained solution (Example 1, Run 8).

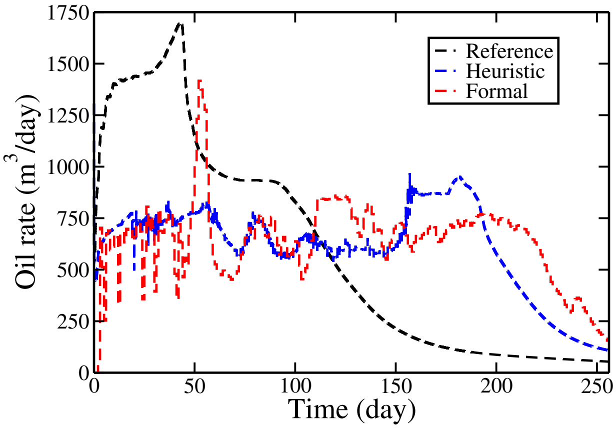

Fig. 6.6 Oil rates BHPs (top), gas rates (middle) and oil rates (bottom) for the best formally constrained solution (Example 1, Run 8).

imum BHP. This solution is infeasible because it violates the nonlinear output constraints. Next, we apply the heuristic constraint handling approach described above, with the maximum gas production rate set to 200 m^3/d. The cumulative oil production for these two cases is given in the first row ('Reference') of Table 6.8. Note that the values are very similar because the nonlinear constraint violation in the unconstrained case is small. The table headings refer to the treatment of the nonlinear constraints – bound constraints are satisfied in all cases.

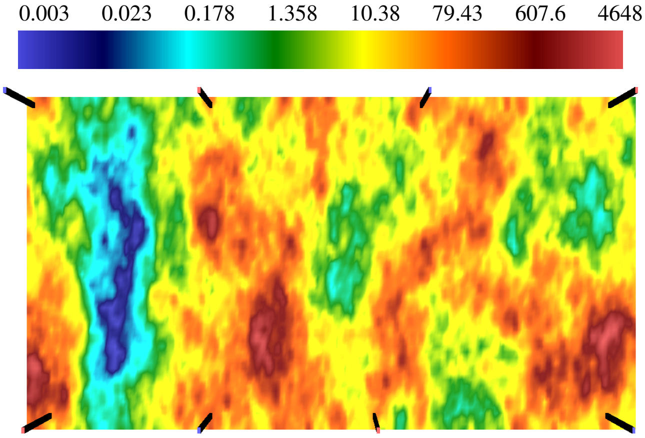

Fig. 6.7 Injection wells (blue) and production wells (red) for Example 2. Background shows $\log \mathsf{K}_x$ ($\mathsf{K}_x = \mathsf{K}_y$).

We next perform optimizations that honor the bound constraints but not the nonlinear constraints. The results for the nine runs, starting from different initial guesses, are presented in Table 6.8 in the column labeled 'Unconstr.' The best optimum achieved is 22,130 m^3 of oil, obtained from Run 6. This clearly exceeds the reference result of 20,290 m^3. Results using the heuristic constraint handling are shown in the third column. Here the best result is 22,030 m^3 of oil (Runs 5 and 9), which exceeds the reference heuristic result by 8.8%. It is important to note that, for both the (nonlinearly) unconstrained and heuristic approaches, some of the runs give results that are below those from the reference cases (e.g., Run 3 in both cases). This is presumably due to convergence to poor local minima. In the next set of runs we apply the formal constraint handling treatment. For these runs, the best optimum is 22,070 m^3 of oil (Run 8). This value is slightly better than the best result using the heuristic treatment.

The oil production profiles for the best runs, along with the reference (heuristic) case, are shown in Fig. 6.8. Recall that we are maximizing cumulative oil, so the fact that early time production in the reference case exceeds that of the optimized cases is not of concern. The detailed BHP, gas rate, and oil rate profiles for the three cases are shown in Figs. 6.9, 6.10 and 6.11. Although the BHPs for the two optimized cases are clearly different, the oil rate profiles do show some similarities. For example, P1 and P3 produce the most oil, and their rates at early times are much lower than those in the reference solution.

It is important to note that the heuristic constraint handling approach is more efficient computationally than the formal treatment. On average, the formal approach required about three times the number of forward simulations used in the

Table 6.9 Fluid description for Example 3

Component	CO_2	C_1	C_2	C_3	C_4	C_{10}
Initial comp. (%)	1	20	30	19	10	20
Injection comp. (%)	95	1	1	1	1	1

corresponding heuristically constrained cases. This large discrepancy results from the need to enforce feasibility within the optimizer in the formal constraint handling approach.

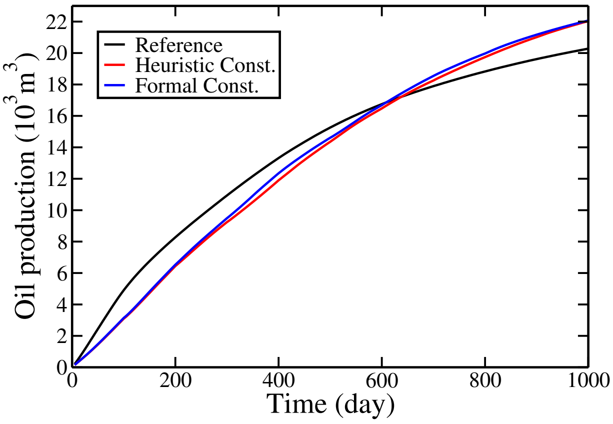

Fig. 6.8 Oil production versus time for Example 2. Results are for feasible reference case (black curve), best heuristically constrained solution (Run 9, red curve) and best formally constrained solution (Run 8, blue curve).

6.3 Example 3: Twelve-well channelized system

Our third example uses the three-dimensional geological model introduced by [14]. As in Example 1 we consider CO_2 injection, though this model contains a total of six components, defined in Table 6.9. Details about the reservoir model, the well locations, and other model parameters are provided in Table 6.10. A map of the x-component of permeability (here $\mathsf{K}_x = \mathsf{K}_y = 10\mathsf{K}_z$), along with the locations of the wells, is shown in Fig. 6.12.

The control parameters of our optimization problem are again the well BHPs. The wells are constrained to operate between an upper bound of 120 bar and a lower bound of 90 bar. We also specify nonlinear constraints on both injection and production in the form of maximum gas flow rates of 200,000 m^3/d for the injectors and 40,000 m^3/d for the producers (both at reservoir conditions). This model is run for a total of 100 days, and we control the BHPs at initial time and then every ten days (the simulation time frame is short in this case because the problem specification is such that oil is produced quickly). There are a total of 120 control parameters in this problem, and our objective is again to maximize cumulative oil production.

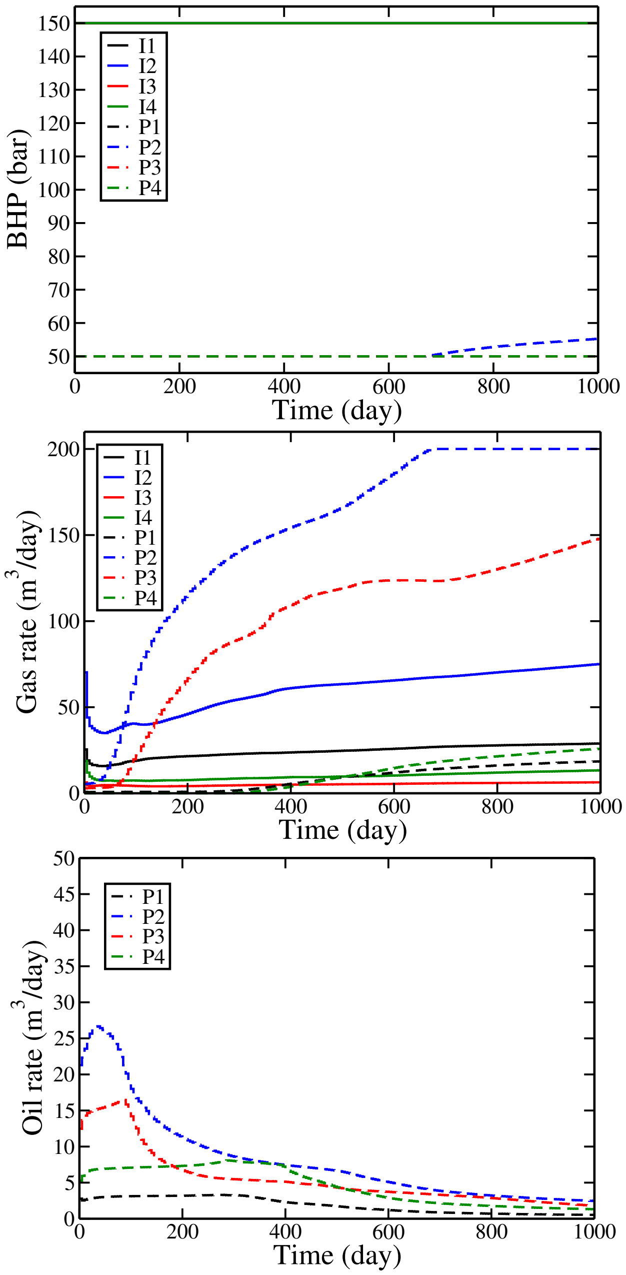

Fig. 6.9 BHPs (top), gas rates (middle) and oil rates (bottom) for the feasible reference solution (Example 2).

We simulate this model using the same procedures as in the previous example. Results for the nine runs for each case are presented in Table 6.11. The feasible reference case yields 5.030×10^6 m^3 of oil, while the best heuristically constrained case (Run 2) provides 5.457×10^6 m^3 of oil, an improvement of 8.5%. The best formally constrained case (Run 3) achieves an optimum of 5.306×10^6 m^3 of oil, which exceeds the reference case by 5.5% but is less than the best heuristic case. The oil production profiles for the best runs, along with the feasible reference case, are shown in Fig. 6.13. We again see that the early time production in the reference

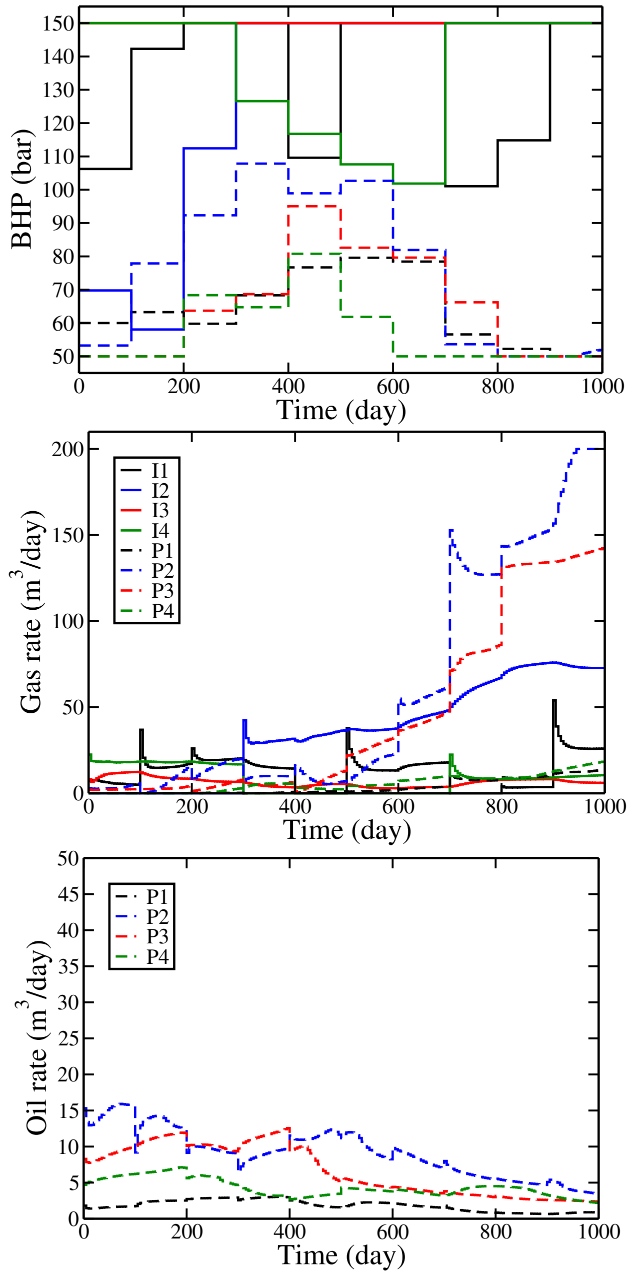

Fig. 6.10 BHPs (top), gas rates (middle) and oil rates (bottom) for the best heuristically constrained solution (Example 2, Run 9).

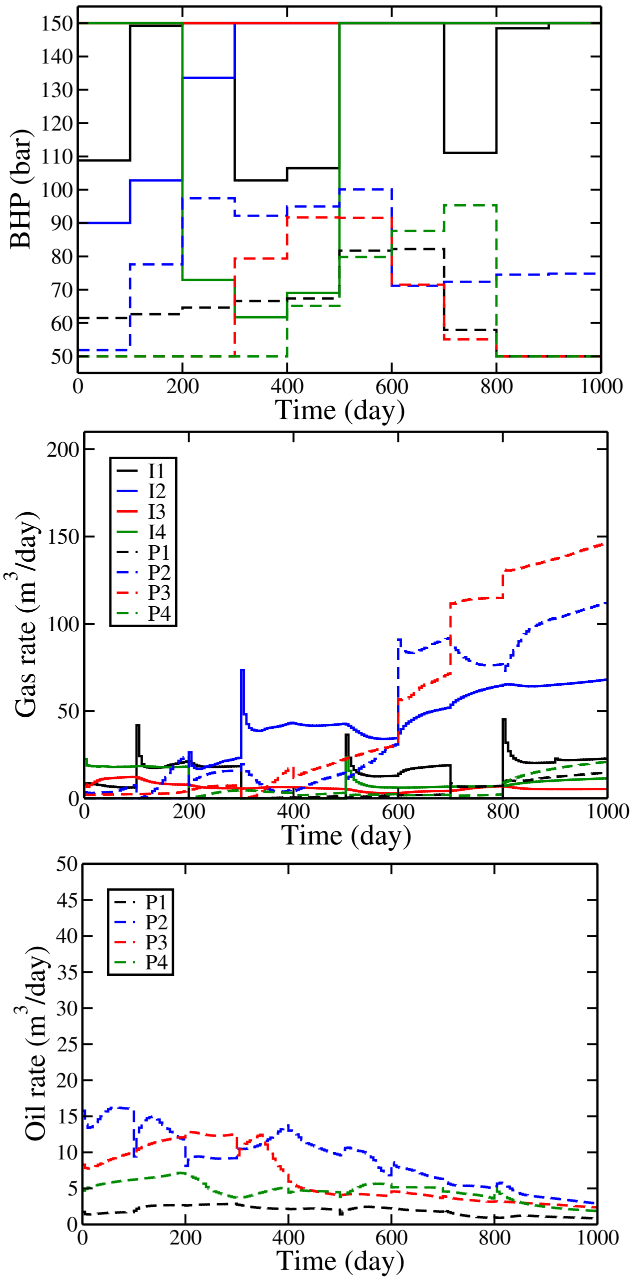

Fig. 6.11 BHPs (top), gas rates (middle) and oil rates (bottom) for the best formally constrained solution (Example 2, Run 8).

case exceeds that of the optimized cases, though the cumulative oil produced in the optimized cases is of course higher.

In this example, the formal constraint handling approach typically required about 1.5 times the number of forward simulations as were required using the heuristic treatment. Our overall findings for this example clearly illustrate the potential advantages of the heuristic treatment for complex problems with multiple wells operating under nonlinear constraints.

6.4 Example 4: Norne model

In our final example we consider the Norne benchmark problem, which is a model of a real field located offshore Norway [27]. The actual Norne model involves a three-phase black-oil system. Here we use the prescribed Norne geological model and well positions (for wells that were operational in January 2005 in the original model). The Norne model contains 29 wells, as shown in Fig. 6.14, though our model involves only 28 of these wells (we do not include the injector C-4H because it does not operate from 2005-2008).

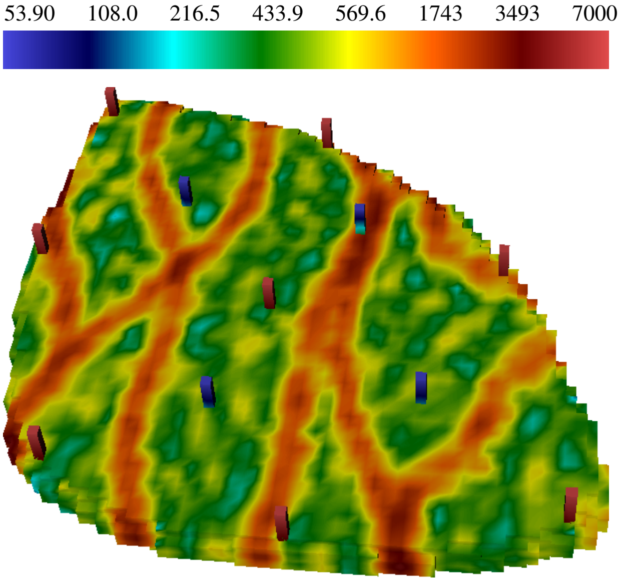

Fig. 6.12 Reservoir model and wells for Example 3 (from [14]). Background shows $\log K_x$.

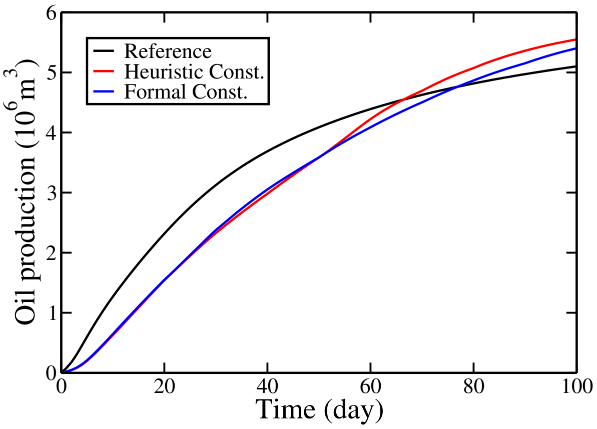

Fig. 6.13 Oil production versus time for Example 3. Results are for feasible reference case (black curve), best heuristically constrained solution (Run 2, red curve) and best formally constrained solution (Run 3, blue curve).

Instead of black-oil, we consider a five-component compositional system with CO_2 injection (see Table 6.12 for the fluid description). The model contains a total of 113,344 grid blocks, though only 44,431 of these blocks are active. Other model parameters are provided in Table 6.13.

The control parameters for the optimization are again the well BHPs, constrained to lie between 50 bar and 150 bar. The nonlinear constraints are maximum gas injection rate of $10^5\text{m}^3/\text{d}$ at reservoir conditions for each injector. The simulation is run for 300 days, and we control the BHPs at initial time and then every 30 days thereafter. Because this problem involves 28 wells, there are 280 control parameters. Our objective is to maximize cumulative oil production.

Table 6.10 Model parameters for Example 3

Grid size	60 × 60 × 7	
Parameter	Value	Units
Δx	24	m
Δy	24	m
Δz	4	m
Depth	2538	m
Initial pressure	100	bar
Temperature	372	°C
Rock compressibility	10^{-5}	1 / bar
Simulation time	300	d
Pressure upper bound	120	bar
Pressure lower bound	90	bar
Residual gas saturation	0	-
Residual oil saturation	0	-
End point rel perm gas	1	-
End point rel perm oil	1	-
Corey exponent gas	2	-
Corey exponent oil	2	-
Well locations [grid block no.]	*i*	*j*
Injector 1	5	57
Injector 2	30	53
Injector 3	2	35
Injector 4	27	29
Injector 5	50	35
Injector 6	8	9
Injector 7	32	2
Injector 8	57	6
Producer 1	16	43
Producer 2	35	40
Producer 3	23	16
Producer 4	43	18

Table 6.11 Oil production in 10^6 m^3 (Example 3) for the optimized objective function without satisfying the nonlinear constraints ('Unconstr.'), satisfying the nonlinear constraints using the heuristic treatment ('Heuristic'), and satisfying the nonlinear constraints using the formal approach ('Formal'). Best feasible results shown in bold.

Run	Unconstr.	Heuristic	Formal
Reference	5.030	5.030	
1	5.450	5.449	5.284
2	5.467	**5.457**	5.294
3	5.171	5.171	**5.306**
4	5.288	5.287	5.132
5	5.424	5.423	5.224
6	5.344	5.348	5.260
7	5.321	5.230	4.994
8	5.207	5.205	5.196
9	5.353	5.349	4.986

Table 6.12 Fluid description for Example 4

Component	CO_2	NC_4	C_8	C_1	C_{15}
Initial composition (%)	1	9	40	10	40
Injection composition (%)	90	7	1	1	1

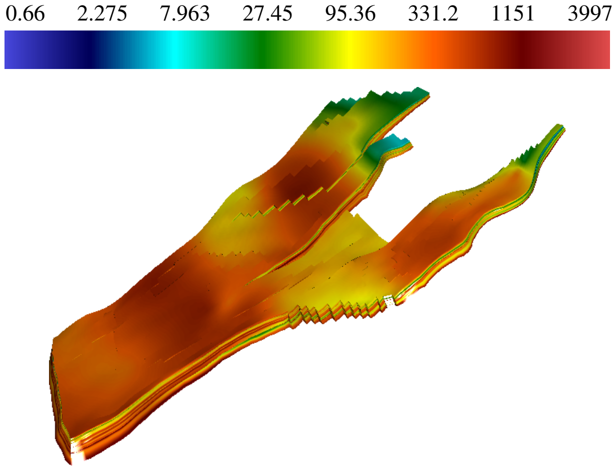

Fig. 6.14 Permeability field in log scale for Example 4 (Norne model).

Table 6.13 Model parameters for Example 4 (Norne model)

Grid size	46 × 112 × 22	

Parameter	Value	Units
Δx	24	m
Δy	24	m
Δz	4	m
Depth	3000	m
Initial pressure	100	bar
Temperature	372	°C
Rock compressibility	8×10^{-5}	1 / bar
Simulation time	300	d
Pressure upper bound	150	bar
Pressure lower bound	50	bar
Residual gas saturation	0	-
Residual oil saturation	0	-
End point rel perm gas	1	-
End point rel perm oil	1	-
Corey exponent gas	2	-
Corey exponent oil	2	-

Table 6.14 Oil production in 10^6 m^3 (Example 4) for the optimized objective function without satisfying the nonlinear constraints ('Unconstr.'), satisfying the nonlinear constraints using the heuristic treatment ('Heuristic'), and satisfying the nonlinear constraints using the formal approach ('Formal'). Best feasible results shown in bold.

Run	Unconstr.	Heuristic	Formal
Reference	252	142	
1	270	146	125
2	266	143	**130**
3	262	120	88
4	270	**148**	121
5	271	147	**130**
6	269	145	119
7	270	146	129
8	271	147	119
9	271	147	109

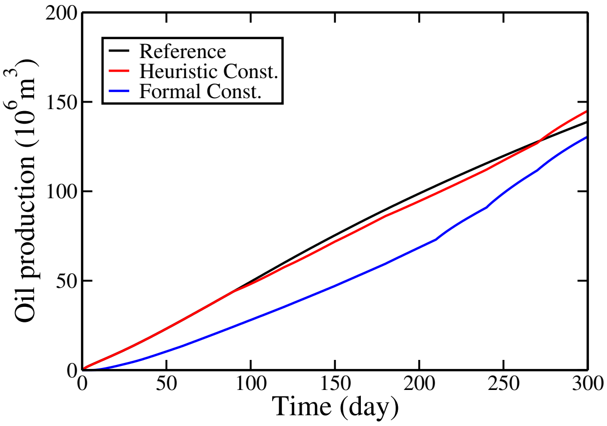

Fig. 6.15 Oil production versus time for Example 4. Results are for feasible reference case (black curve), best heuristically constrained solution (Run 4, red curve) and best formally constrained solution (Run 2, blue curve).

Results for the three sets of runs are reported in Table 6.14. It is evident from the large differences between the nonlinearly unconstrained runs (second column) and the constrained runs (third and fourth columns) that the nonlinear constraints have a much larger effect in this example than in Examples 1 and 2. Applying these constraints heuristically, the best maximum obtained is 148×10^6 m^3 (Run 4), an improvement of 4.2% over the heuristically constrained reference case. This level of improvement is less than that observed for the other examples. We also see that the formal constraint handling approach leads to results for cumulative oil production (130×10^6 m^3 in the best cases, Runs 2 and 5) that are lower than that for the heuristic reference case, which does not involve any optimization. Note also that poor solutions are achieved in Run 3 for both the heuristic and formal constraint treatments. The results using the formal treatment illustrate the potential challenges that can arise in complex problems with large numbers of control parameters and nonlinear constraints. It is worth reiterating that the heuristic approach does, even in this challenging case, provide improvement over the feasible reference solution.

The oil production profiles for the best optimization runs, along with the feasible reference case, are shown in Fig. 6.15. The slight improvement offered by the heuristic procedure over the reference case is evident. The detailed BHP and gas injection rates (for some of the wells) versus time for the three cases are shown in Figs. 6.16, 6.17 and 6.18. The frequent shifts in BHP in the reference case (Fig. 6.16) and in the heuristic case (Fig. 6.17) enable higher oil production in those runs. In the formally constrained case (Fig. 6.18), BHPs are held constant over the entire control period. This results in less gas injection and, as a result, less oil production than in the other cases.

We expect that the use of a sufficiently large number of control periods would provide improvement in the results using the formal constraint handling procedure. This would lead, however, to a more challenging optimization problem that might require many more forward simulations. For the results presented here, the optimizations using formal con-

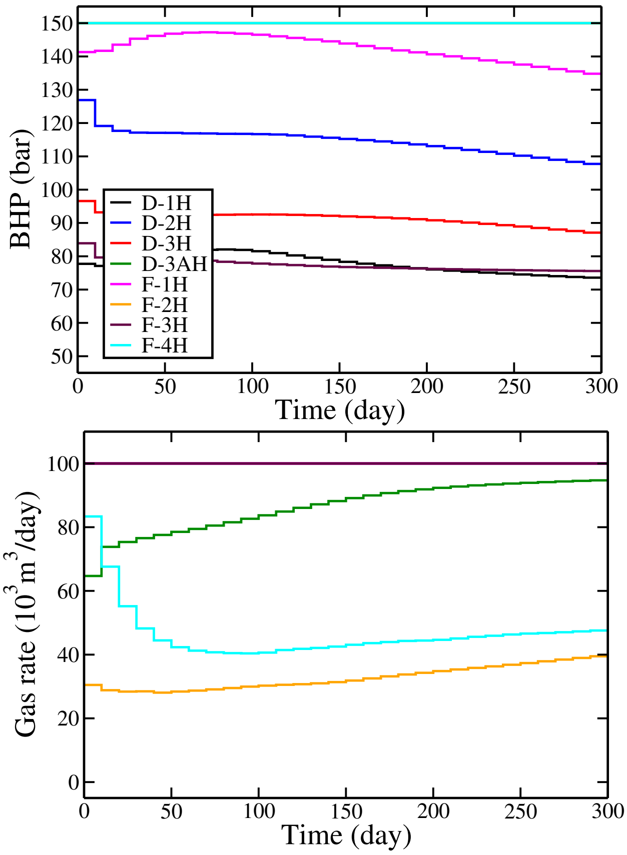

Fig. 6.16 Injector BHPs (top) and gas injection rates (bottom) for the feasible reference solution (Example 4).

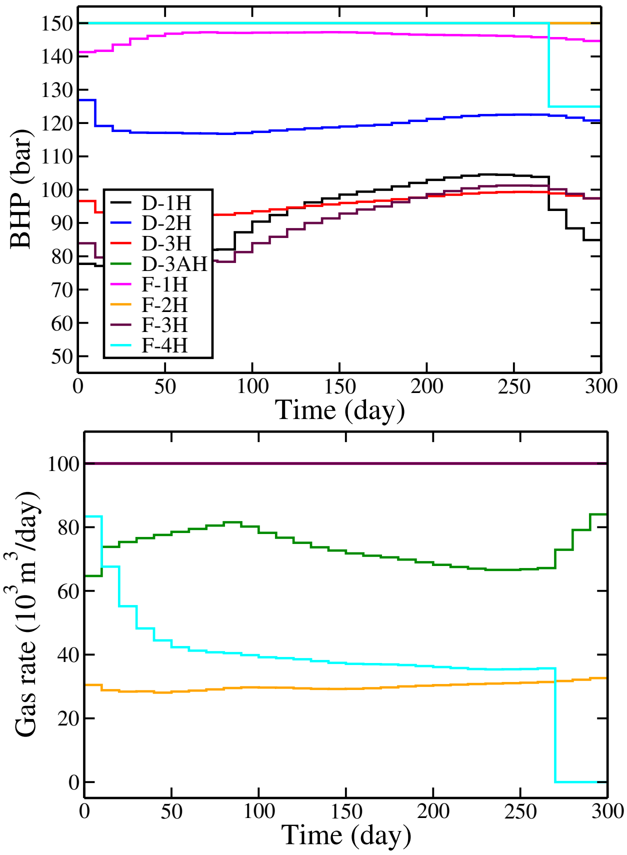

Fig. 6.17 Injector BHPs (top) and gas injection rates (bottom) for the best heuristically constrained solution (Example 4, Run 4).

straint handling required about 150 forward simulations on average. Optimizations using the heuristic constraint handling, by contrast, required about 41 forward simulations on average. Thus we again observe significant improvements in computational efficiency using the heuristic approach.

7 Concluding remarks

In this work we formulated and tested an adjoint-based optimization procedure for compositional reservoir simulation. The method we employed was implemented into Stanford's Automatic Differentiation-based General Purpose Research Simulator (AD-GPRS). The use of automatic differentiation simplifies the adjoint implementation and subsequent code enhancements. Two different treatments for handling nonlinear constraints were presented. In the formal constraint handling procedure, lumped constraints and their gradients are provided to the optimizer, and feasibility is enforced by the optimization algorithm. In the second (heuristic) procedure, an optimization satisfying only the bound and linear constraints is performed first. Then, the forward model is run using the controls from the first stage, but the simulator is allowed to switch from BHP to rate control (for a problem in which BHPs are the control variables) as required to satisfy the nonlinear rate constraints.

Numerical results were presented for three example cases of increasing complexity. Nine runs, starting from different initial conditions, were performed in all cases, for both the heuristic and the formal nonlinear constraint treatments. In the examples, the control variables were the well BHPs at various times, and maximum rate specifications entered as nonlinear constraints. The total number of control variables ranged from 80 to 280. In the simplest case, the formal constraint handling approach slightly outperformed the heuristic procedure, but in the other two cases the heuristic treatment provided the highest objective function value. Improvements in cumulative oil produced (which was the objective function in all cases) relative to the reference solutions ranged from 4.2% to 8.8%. The heuristic constraint treatment was shown to be much more efficient than the formal approach.

In addition to the discrete adjoint procedure, which was used for all of the examples presented in this paper, we also derived and implemented a continuous adjoint formulation. We showed that the two formulations use different final time

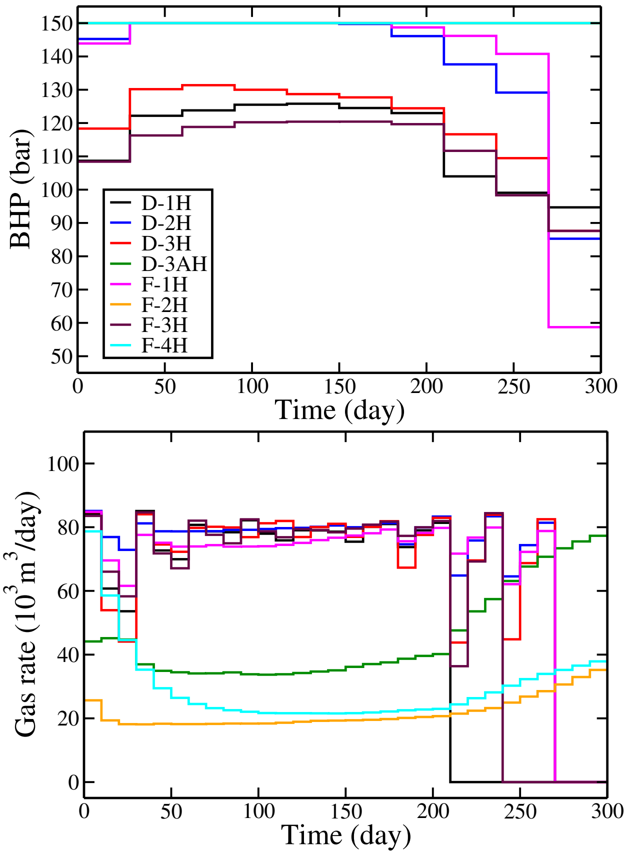

Fig. 6.18 Injector BHPs (top) and gas injection rates (bottom) for the best formally constrained solution (Example 4, Run 2).

conditions and as a result the constructed gradients do not agree in general. However, the two boundary conditions become almost identical as the size of the last time step approaches zero and thus, the gradients obtained by both formulations are essentially the same.

There are a number of areas in which future research should be directed. Other treatments for nonlinear constraints, both formal and heuristic, should be considered, and the relative benefits of controlling rates instead of BHPs should be assessed. It will be of interest to apply the general optimization framework to larger and more realistic simulation models. Other types of wells (horizontal, deviated, multilateral) should also be considered, along with the optimization of downhole inflow control devices. The overal approach can be extended to robust control (to cope with geological uncertainty) and hierarchical (multi-objective) control to balance long-term and short-term objectives. Finally, the overall approach could be applied for the optimization of CO_2 storage or for combined EOR-CO_2 storage operations.

Acknowledgements We thank Denis Voskov and Oleg Volkov at Stanford University for useful discussions and assistance with AD-GPRS. We also thank professor Michael Saunders for his ongoing support on SNOPT. We are also grateful to the sponsors of the Stanford University Smart Fields Consortium for partial funding of this work.

References

1. Asheim, H.: Maximization of water sweep efficiency by controlling production and injection rates. In: SPE Paper 18365 presented at the SPE European Petroleum Conference. London, UK (1988)
2. Asouti, V.G., Zymaris, A.S., Papadimitriou, D.I., Giannakoglou, K.C.: Continuous and discrete adjoint approaches for aerodynamic shape optimization with low Mach number preconditioning. International Journal for Numerical Methods in Fluids **57**(10), 1485–1504 (2008)
3. Aziz, K., Settari, A.: Petroleum Reservoir Simulation. Applied Science Publishers, London (1979)
4. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific (1999)
5. Brouwer, D., Jansen, J.: Dynamic optimization of water flooding with smart wells using optimal control theory. SPE Journal **9**(4), 391–402 (2004)
6. Bryson, A., Ho, Y.: Applied Optimal Control. Taylor and Francis (Hemisphere), Levittown (2001)
7. Cao, H.: Development of techniques for general purpose simulators. Ph.D. thesis, Stanford University (2002)
8. Chavent, G., Dupuy, M., Lemonnier, P.: History matching by use of optimal theory. SPE Journal **15**, 74–86 (1975). DOI 10.2118/4627-PA
9. Chen, W., Gavalas, G., Wasserman, M.: A new algorithm for automatic history matching. SPE Journal **14**(6), 593–608 (1974)
10. Christie, M., Blunt, M.: Tenth SPE comparative solution project: a comparison of upscaling techniques. SPE Reservoir Evaluation and Engineering **4**(4), 308–317 (2001)
11. Coats, K.: An equation of state compositional model. SPE Journal **20**(5), 363–376 (1980)
12. Doublet, D., Aanonsen, S., Tai, X.: An efficient method for smart well production optimisation. Journal of Petroleum Science and Engineering **69**(1–2), 25–39 (2009)
13. Echeverría Ciaurri, D., Isebor, O., Durlofsky, L.: Application of derivative-free methologies to generally constrained oil production optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation **2**, 134–161 (2011)
14. van Essen, G., Zandvliet, M., Van den Hof, P., Bosgra, O., Jansen, J.D.: Robust waterflooding optimization of multiple geological scenarios. SPE Journal **14**(1), 202–210 (2009)

15. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM Rev. **47**(1), 99–131 (2005)
16. Gunzburger, M.: Perspectives in Flow Control and Optimization. SIAM (2003)
17. Hager, W.W.: Runge-kutta methods in optimal control and the transformed adjoint system. Numerische Mathematik **87**(2), 247–282 (2000). DOI 10.1007/s002110000178
18. Han, C., Wallis, J., Sarma, P., Li, G., Schrader, M.L., Chen, W.: Enhancement of the CPR preconditioner for efficient solution of the adjoint problem. In: SPE Paper 141300 presented at the SPE Reservoir Simulation Symposium. The Woodlands, Texas, USA (2011). DOI 10.2118/141300-MS
19. Jansen, J.D.: Adjoint-based optimization of multi-phase flow through porous media - a review. Computers & Fluids **46**(1), 40 – 51 (2011). DOI 10.1016/j.compfluid.2010.09.039
20. Li, R., Reynolds, A., Oliver, D., of Tulsa, U.: History matching of three-phase flow production data. SPE Journal **8**(4), 328–340 (2003). DOI 10.2118/87336-PA
21. Lien, M., Brouwer, D., Manseth, T., Jansen, J.: Multiscale regularization of flooding optimization for smart field management. SPE Journal **13**(2), 195–204 (2008)
22. Liu, W., Ramirez, W., Qi, Y.: Optimal control of steam flooding. SPE Advanced Technology Series **1**(2), 73–82 (1993)
23. Mehos, G., Ramirez, W.: Use of optimal control theory to optimize carbon dioxide miscible-flooding enhanced oil recovery. Journal of Petroleum Science and Engineering **2**(4), 247–260 (1989)
24. Nadarajah, S., Jameson, A.: A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In: AIAA 38th Aerospace Sciences Meeting and Exhibit. Reno, NV (2000)
25. Nadarajah, S.K., Jameson, A.: Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint methods. AIAA Journal **45**(7), 1478–1491 (2007). DOI 10.2514/1.24332
26. Noemi Petra, G.S.: Model variational inverse problems governed by partial differential equations. ICES Report 11-05, The Institute for Computational Engineering and Sciences, The University of Texas at Austin (2011)
27. NTNU (IO Center): Center of Integrated Operations in Petroleum Industry. Website (2011). `www.ipt.ntnu.no/~norne`
28. Oliver, D., Reynolds, A., Liu, N.: Inverse Theory for Petroleum Reservoir Characterization and History Matching. Cambridge University Press, Cambridge (2008)
29. Ramirez, W.: Application of Optimal Control Theory to Enhanced Oil Recovery. Elsevier Science Ltd, Amsterdam (1987)
30. Sarma, P., Chen, W., Durlofsky, L., Aziz, K.: Production optimization with adjoint models under nonlinear control-state path inequality constraints. SPE Reservoir Evaluation and Engineering **11**(2), 326–339 (2008)
31. 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**(1), 3–36 (2006). DOI 10.1007/s10596-005-9009-z
32. Stengel, R.: Stochastic Optimal Control Theory and Application. Wiley, New York (1986)
33. Sudaryanto, B., Yortsos, Y.: Optimization of fluid front dynamics in porous media using rate control. Physics of Fluids **12**(7), 1656–1670 (2000)
34. Virnovski, G.: Waterflooding strategy design using optimal control theory. In: 6th European IOR Symposium, pp. 437–446. Stavanger, Norway (1991)
35. Voskov, D., Tchelepi, H.: Comparison of nonlinear formulations for two-phase multi-component EoS based simulation. Journal of Petroleum Science and Engineering **82–83**, 101–111 (2012). DOI 10.1016/j.petrol.2011.10.012
36. Voskov, D.V., Younis, R., Tchelepi, H.A.: Comparison of nonlinear formulations for isothermal compositional flow simulation. In: SPE paper 118966 presented at the SPE Reservoir Simulation Symposium (2009)
37. Walther, A.: Automatic differentiation of explicit runge-kutta methods for optimal control. Computational Optimization and Applications **36**(1), 83–108 (2007). DOI 10.1007/s10589-006-0397-3
38. Young, L., Stephenson, R.: A generalized compositional approach for reservoir simulation. Journal of Petroleum Science and Engineering **23**(5), 727–742 (1983). DOI 10.2118/10516-PA
39. Younis, R., Aziz, K.: Parallel automatically differentiable data-types for next-generation simulator development. In: SPE Paper 106493 presented at the SPE Reservoir Simulation Symposium. Houston, Texas (2007). DOI 10.2118/106493-MS
40. Younis, R., Tchelepi, H., Aziz, K.: Adaptively localized continuation-Newton method–Nonlinear solvers that converge all the time. SPE Journal **15**(2), 526–544 (2010). DOI 10.2118/119147-PA. SPE-119147-PA