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 injection or production 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 four example cases of increasing complexity are presented. Improvements in the objective function (cumulative oil produced) relative to reference solutions range from 4.2% to 11.6%. The heuristic treatment of nonlinear constraints is shown to offer a cost-effective means for obtaining feasible solutions, which are in some cases better than those obtained using the formal constraint handling procedure.

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

1 Introduction

The optimization of time-varying well settings, such as injection and production rates or bottom-hole pressure, is an important aspect of optimal reservoir management. Both gradient-based and derivative-free methods have been considered for this problem, and both are applicable in different situations. 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, 34]. Adjoint-based optimization techniques have been used in a reservoir simulation setting both for history matching (see, e.g., [8, 9, 22, 30, 33]) and for production optimization. Much of the early work on their use for optimization of oil recovery was performed by Ramirez and coworkers, who considered the optimization of several different enhanced oil recovery (EOR) processes [24, 25, 31]. In subsequent work, the focus was

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

on gradient-based optimization (and in some cases on the optimization of 'smart wells') for water flooding [1, 5, 33, 35, 38]. 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 [14, 19, 23, 32]. Refer to [20] 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 [15] 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 iteration of every time step. 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 [45]. 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 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. Consistent with this, 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, 27]). In particular, Nadarajah and Jameson [27], 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 on the numerical treatments are also presented. The gradient-based optimization software is discussed in Section 4. In Section 5 we describe our nonlinear constraint handling procedures. Numerical results demonstrating the capabilities of our optimization procedure, for a series of 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, 39, 40]:

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

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, $\mathbf{K}$ 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 5.2. 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:

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

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:

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

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, 12, 39, 43]), 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 [39] 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 [45]. 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 [19]. 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 value. 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 numerical and coding 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 written. The maintenance and development of adjoint code poses challenges because the necessary derivatives are generally 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 gaining popularity in the field of scientific computing 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 [40]. 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 and Aziz [44]. 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 [44] 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:

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

where $\mathbf{g}_n$ denotes the fully discretized, both in space and time, set of partial differential 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 will use throughout the notation $\partial \mathbf{g}^T/\partial \mathbf{x}$ to denote the matrix $(\partial \mathbf{g}/\partial \mathbf{x})^T$.

We are interested in either maximizing or minimizing an objective function J 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 J has the following form:

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

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

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

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

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

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 $\mathbf{g}_n = \mathbf{0}$, we can introduce the augmented objective function J_A by 'adjoining' the governing equations to the original objective function J. The new objective J_A shares the same extrema as J and is defined as:

$$J_A = \sum_{n=1}^{N} \Big(\Delta t_n f_n(\mathbf{x}_n,\mathbf{u}_n) + \boldsymbol{\lambda}_n^T \mathbf{g}_n(\mathbf{x}_n,\mathbf{x}_{n-1},\mathbf{u}_n)\Big) + \varphi(\mathbf{x}_N). \tag{3.4}$$

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

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

$$\begin{aligned} \delta J_A = & \left(\frac{\partial \varphi_N}{\partial \mathbf{x}_N} + \Delta t_N \frac{\partial f_N}{\partial \mathbf{x}_N} + \boldsymbol{\lambda}_N^T \frac{\partial \mathbf{g}_N}{\partial \mathbf{x}_N}\right) \delta \mathbf{x}_N \\ & + \sum_{n=1}^{N-1} \left(\Delta t_n \frac{\partial f_n}{\partial \mathbf{x}_n} + \boldsymbol{\lambda}_{n+1}^T \frac{\partial \mathbf{g}_{n+1}}{\partial \mathbf{x}_n} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{x}_n}\right) \delta \mathbf{x}_n \\ & + \sum_{n=1}^{N} \left(\Delta t_n \frac{\partial f_n}{\partial \mathbf{u}_n} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{g}_n}{\partial \mathbf{u}_n}\right) \delta \mathbf{u}_n. \end{aligned} \tag{3.5}$$

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

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

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

With this choice of the Lagrange multipliers the total variation becomes

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

and the gradient of the objective function with respect to the controls is

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

The individual entries of $\delta J_A/\delta \mathbf{u}$ are given by

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

By driving $\delta J_A/\delta \mathbf{u}$ to zero, we achieve the minimum or maximum of J_A (and thus J). In practice, $\delta J_A/\delta \mathbf{u}$, 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 optimization problems, the well control variables do not typically change at each time step in the flow simulation. Rather, they are defined over longer time periods that are referred to as control steps. Time steps are usually small in order to capture flow dynamics, reduce time-discretization error, and facilitate convergence of the Newton iterations. The gradient at the control period m, $\delta f_n/\delta \mathbf{u}_m$, is simply the sum of the gradients $\delta f_n/\delta \mathbf{u}_n$ for all time steps that belong to control period m.

3.3 Continuous adjoint formulation

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

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

In this case we express the governing set of partial differential equations, for a specified dynamic well-control strategy

$\mathbf{u}(t)$, as $\mathbf{g}(\dot{\mathbf{x}}(t),\mathbf{x}(t),\mathbf{u}(t)) = \mathbf{0}$. We introduce the Lagrange multipliers $\boldsymbol{\lambda}(t)$ and define the Lagrangian L as:

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

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

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

The first variation of J_A is given by

$$\delta J_A = \int_{t_0}^{t_N} \left(\frac{\partial L}{\partial \dot{\mathbf{x}}} \delta\dot{\mathbf{x}} + \frac{\partial L}{\partial \mathbf{x}} \delta\mathbf{x} + \frac{\partial L}{\partial \mathbf{u}} \delta\mathbf{u} + \frac{\partial L}{\partial \boldsymbol{\lambda}} \delta\boldsymbol{\lambda} \right) dt + \frac{\partial \varphi(\mathbf{x}_N)}{\partial \mathbf{x}_N} \delta\mathbf{x}_N. \tag{3.12}$$

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 $\partial L^T/\partial\boldsymbol{\lambda} = \mathbf{g}(\dot{\mathbf{x}},\mathbf{x},\mathbf{u}) = \mathbf{0}$, the first variation of J_A can be written as:

$$\delta J_A = \int_{t_0}^{t_N} \left(\frac{\partial L}{\partial \mathbf{x}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{x}}} \right) \delta\mathbf{x}\,dt + \left(\frac{\partial L(\mathbf{x}_N)}{\partial \dot{\mathbf{x}}_N} + \frac{\partial \varphi(\mathbf{x}_N)}{\partial \mathbf{x}_N} \right) \delta\mathbf{x}_N + \int_{t_0}^{t_N} \frac{\partial L}{\partial \mathbf{u}} \delta\mathbf{u}\,dt. \tag{3.13}$$

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

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

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

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:

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

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:

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

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 using equations (3.8) and (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 $\varphi(\mathbf{x}_N) = 0$. 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 $\delta\mathbf{x}_N$:

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

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 $\delta J_A/\delta\mathbf{u}$. 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 [18] and [41] for general Runge-Kutta time stepping methods, in [26], where the discrete and continuous adjoint approaches were applied to automatic aerodynamic optimization, and in [28] 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 for a shape optimization problem in [17].

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 time step n, $\partial \mathbf{g}_n^T/\partial \mathbf{x}_n$, is the transpose of the Jacobian matrix for the converged forward problem, $\partial \mathbf{g}_n/\partial \mathbf{x}_n$. 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 $\partial \mathbf{g}_n/\partial \mathbf{x}_n$ 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 [19]. In these linear solutions, we require very high accuracy to guarantee that residual errors accumulated over hundreds of time steps do 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, and in this section we will provide a concise overview of the underlying theory. This discussion loosely follows [16], which provides a much more in-depth description. We note that the adjoint formulation described in this paper may also be used in conjunction with other optimization packages. 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.

We now discuss the main features of the SQP method used to solve our nonlinear program (NP). This discussion follows [16]. As will be discussed in detail in Section 5.1, all realistic recovery optimization problems involve bound constraints on the inputs, and often several other linear or nonlinear equality or inequality constraints, which together constitute a so-called general nonlinear program (GNP). All features described in this section are readily specialized to the GNP discussed in Section 5.1. Here 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 $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^m$, while the objective function $f(\mathbf{u})$ and the constraints $c_i(\mathbf{u})$, $i = 1, 2, \ldots, m$ have continuous second derivatives. The gradient of f is denoted by the vector $\nabla f(\mathbf{u})$, 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 $\mathbf{u}$ and dual variables $\boldsymbol{\pi}$) from an iterative sequence of QP subproblems. Each QP subproblem, in turn, iteratively 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. In this formulation k is the SQP (major) iteration counter, and $\boldsymbol{\pi}$ are Lagrangian multipliers to adjoin $\mathbf{d}_L$ to f. The first and second derivatives of the modified Lagrangian with respect to $\mathbf{u}$ are

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

Observe that $\nabla^2 \mathscr{L}$ is independent of $\mathbf{u}_k$ and is the same as the Hessian of the conventional Lagrangian. At $\mathbf{u} = \mathbf{u}_k$, i.e., after convergence of the inner iterations, the modified Lagrangian has the same function and gradient values as the objective: $\mathscr{L}(\mathbf{u}_k, \mathbf{u}_k, \boldsymbol{\pi}_k) = f_k$, $\nabla \mathscr{L}(\mathbf{u}_k, \mathbf{u}_k, \boldsymbol{\pi}_k) = \nabla f_k$. 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

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

where ρ and $\mathbf{s}$ are vectors of penalty parameters and slack variables, respectively, is reduced along each search direction to ensure convergence from any starting point.

In summary, 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 major iterate a QP subproblem is solved 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, and the minor iterations of an SQP method are 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

Several authors have discussed the incorporation of constraints in the optimal control problem for oil recovery optimization. This includes partial and sometimes heuristic approaches, valid for particular types of constraints [5, 37, 38, 42], and more systematic approaches, valid for a broader range of constraint equations [10, 13, 21, 32, 36]. An important feature in simulations involving highly compressible fluids, which we have in the systems considered here since we inject gas, is the occurrence of transient peaks in the rate in response to changes in well bottom-hole pressure (bottom-hole pressure, or BHP, is the wellbore pressure at a specified depth within the reservoir). These transient effects impact simulation results in our models because they occur over time scales that are larger than a time step (this is in contrast to nearly incompressible systems, where this type of transient decays very quickly and is not resolved by the simulator). These transient effects lead to challenges in constrained optimization of compositional systems, because rate constraints can be easily violated. Here we will present and assess both a formal constraint handling approach and a simpler heuristic procedure for satisfying rate constraints, which can be used in compressible systems.

5.1 Constraint handling in the optimizer

In the presence of nonlinear output inequality constraints, it can be difficult to determine an initial guess for the control parameters that does not violate these constraints. In such cases the optimizer must apply a process to find a feasible set of control parameters. Some optimizers may also intentionally exit the feasible region, in an attempt to (eventually) find better optima. For these reasons, constraint and infeasibility handling are key architectural components of any general purpose optimizer. Here we described how they are handled within SNOPT.

To simplify notation, we first write the constrained optimization problem in the more general form (GNP) assuming lower and upper bounds on the controls and the nonlinear constraints

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

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 Achieving feasibility

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 reverts to a technique known as 'elastic programming' to solve the problem

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

where $\mathbf{e}$ is a vector containing all ones and $\mathbf{v}$ and $\mathbf{w}$ are 'elastic variables' which allow the constraints to stretch in directions that increase the size of the feasible region. The resistance to stretching is obtained by minimizing the composite objective function $f(\mathbf{u}) + \gamma \mathbf{e}^T(\mathbf{v}+\mathbf{w})$, where 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 'reasonable' 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 require many major iterations until a feasible initial set of control parameters is obtained. For this reason, optimization with simple bound constraints on the inputs usually converges much faster than optimization with additional bounds on output quantities.

5.1.2 Constraint lumping

Many constraints appear as simple bound constraints (e.g., 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. Examples of this are the specification of maximum gas injection or production rates (either for an individual well or for a group of wells) in a general compositional problem where the control variables are BHPs. A variety of methods have been proposed to incorporate nonlinear constraints in the adjoint formulation; see, e.g., [10, 20, 33, 36] for detailed discussion.

As indicated above, operational constraints in reservoir simulation are often inequality constraints specified for input and/or output rates at every time step. In some cases these constraints must be satisfied by many, or by all, of the wells in the problem. This means that the vector $\mathbf{c}(\mathbf{u})$ in (GNP) can be of dimension $O(N_w N)$, where N is the number of time steps and N_w is the number of wells. Computing the gradients for each of the entries of $\mathbf{c}(\mathbf{u})$, which is required by SNOPT, will require N adjoint simulations and thus, in total, the solution of $O(N_w N^2)$ linear systems of equations in addition to each forward simulation. This may be computationally intractable for realistically-sized problems.

A more efficient way of approximating these gradients is to 'lump' the constraints over the full simulation time frame [32]. This lumping can be performed on a well-by-well basis, in which case $O(N_w N)$ linear systems must be solved for the evaluation of the gradient, or over the entire model, in which case only $O(N)$ linear systems must be solved. The approximate gradients of the lumped nonlinear constraints are then obtained in the same manner as the gradient of the objective function; i.e., in terms of time-step contributions. It is important to recognize that the approach used for constraint lumping can impact the performance of the optimization procedure. 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 the rate of the well j at time step n and q_{max} is a specified maximum rate. To satisfy this constraint we must guarantee that

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

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

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

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 given by:

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

The Lagrange multipliers for the constraints $\boldsymbol{\lambda}_n$ are computed from the solution of the following adjoint problem:

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

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

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. This is often readily accomplished because, 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, which (subject to some simplifications) is of the general form

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

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. Here $p_{w,l}$ is related to the BHP through an additional well equation. When 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 at reservoir conditions (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 honored during the optimization. Then, after this initial optimization has converged, we run the forward problem once more 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, as just one additional simulation run is performed.

In our implementation of the heuristic constraint handling procedure, we also allow wells to switch back to BHP control (after they have switched from BHP control to rate control). The switch in control is accomplished, in either case, by checking (at each Newton iteration) to see if the current active control is violated. This assessment can be performed efficiently through use of the well equation. If the well does switch back to BHP control, it remains on BHP control unless (and until) the rate constraint is again violated. This process generally increases the number of Newton iterations required for convergence, though in most of the cases considered the total simulation time is within a factor of two of that for a typical forward simulation run (and, as noted above, the run using heuristic constraint handling is only performed once).

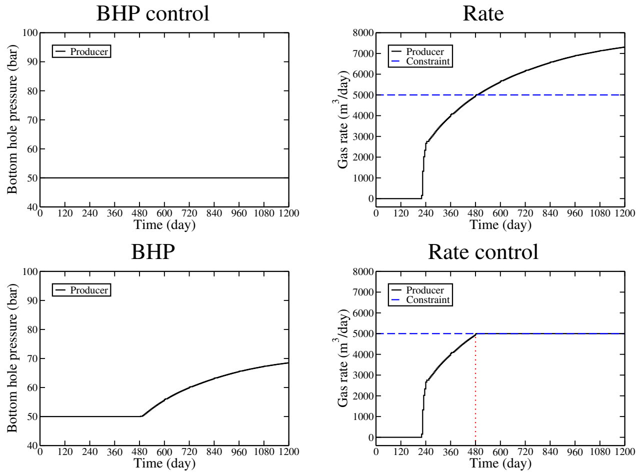

Fig. 5.1 Schematic illustrating heuristic constraint handling. Top: Constant BHP and resulting gas rate. Bottom: BHP and gas rate satisfying constraint.

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 large increase in the number of control variables would result in a more difficult optimization problem, which could negatively impact the performance of the optimizer. In the examples below, we will compare the performance of these two approaches for handling nonlinear constraints.

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.

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	$[p_I^u, p_P^u]$

6 Numerical results

We will present results for four 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.

6.1 Example 1 - Π obstacle

In the first example we maximize cumulative oil recovery under CO_2 injection. The two-dimensional geological model is depicted in Fig. 6.1. A Π-shaped region is located at the center of a homogeneous reservoir. The model is discretized on a grid of dimensions 80×80. The permeability in most of the domain (red cells) is set to 4000 mD, while the permeability for the blue cells that comprise the Π-shaped region is set to 10^{-4} mD. In all of our examples we describe the permeability with a diagonal tensor: $\mathsf{K} = diag(\mathsf{k}_x, \mathsf{k}_y, \mathsf{k}_z)$; here, in addition, the permeability is isotropic and uniform within each of the regions. Four injection wells are placed at the corners of the model, and the single production well is located inside the Π-shaped region. The model includes a total of four components (three hydrocarbon components plus CO_2), as specified in Table 6.2. Further details on the reservoir model are provided in Table 6.3.

Table 6.2 Fluid description for Example 1

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

Table 6.3 Model parameters for Example 1

Grid size	80 × 80 × 1	
Parameter	Value	Units
Δx	6	m
Δy	6	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	80
Injector 3	80	1
Injector 4	80	80
Producer 1	40	48

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 a maximum (per-well) gas injection rate 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 (maximum gas injection rate of 500 m^3/d). Next, we apply the heuristic constraint handling approach described above, with the maximum gas injection 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 a cumulative oil production of

190,200 m^3, obtained in Run 7. This clearly exceeds the unconstrained reference result of 163,900 m^3. Results using heuristic constraint handling are shown in the third column. Here the best result is a cumulative oil production of 163,500 m^3 (Run 8), which exceeds the feasible reference solution (152,200 m^3) by 7.4%. In the next set of runs we apply the formal constraint handling treatment. For these runs, the best optimum is 160,600 m^3 of oil (Run 4). This value exceeds the feasible reference solution by 5.5%, but it is about 2% less than that achieved using heuristic constraint handling. We will show below that results using the formal procedure can be improved through use of more control steps.

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 general similarity. For example, they both show less variation in oil rate over the course of the simulation than the reference case.

It is important to note that the heuristic constraint handling approach is more efficient computationally than the formal treatment. In terms of computational requirements, for this case the formal approach required 49 forward simulations (on average) to converge to the optimal solution, while the heuristic procedure needed only an average of 27 forward simulations. This significant difference results from the need to enforce feasibility within the optimizer in the formal constraint handling approach.

We now assess the impact of using additional control variables. Theoretically, if the optimization problem remains sufficiently 'tractable', as we increase the number of control variables the formal approach should (eventually) outperform the heuristic approach. However, if the optimization problem becomes significantly more difficult with increasing numbers of control variables (which may be related to the constraint lumping procedure), or if a large number of local optima associated with relatively poor objective function values appear, then the formal approach will not necessarily outperform the heuristic treatment.

To test the performance of our procedures, we now consider optimizations with 64 control steps, which corresponds to 320 control variables (the results above used eight control steps and 40 control variables). Results for this case are presented in Table 6.5. The best result using heuristic constraint handling provides cumulative oil production of 159,400 m^3 (Run 7). This is slightly lower than that achieved using 40 controls, which presumably reflects the fact that this is a more difficult optimization problem. Using the formal constraint handling approach, however, we achieve cumulative oil production of 170,200 m^3 (Run 1). This exceeds the feasible reference solution (152,500 m^3) by 11.6%, which represents a substantial improvement. It also exceeds the best solution found using heuristic constraint handling (163,500 m^3 in Run 8, using 40 controls) by 4.1%. In fact, three of the nine local optima achieved in this case using formal constraint handling exceed the best result obtained using heuristic constraint handling. These findings suggest that, for this problem, the formal approach does indeed outperform the heuristic approach given a sufficient number of control variables.

Finally, it is worth noting that the spread in the results from run to run for (optimized) cumulative oil is larger with 320 control variables than it is with 40 control variables. In fact, with 320 control variables, optimizations using both constraint handling procedures lead to some local optima that are below the corresponding lowest optima achieved using 40 control variables (Runs 3 and 5 for the heuristic approach; Runs 2, 3 and 4 for the formal approach). Again, we believe this is indicative of the challenges associated with performing constrained optimization with increasing numbers of control variables. These results suggest that it may be useful to explore the use of a sequence of optimizations, with increasing numbers of control periods, for production optimization problems.

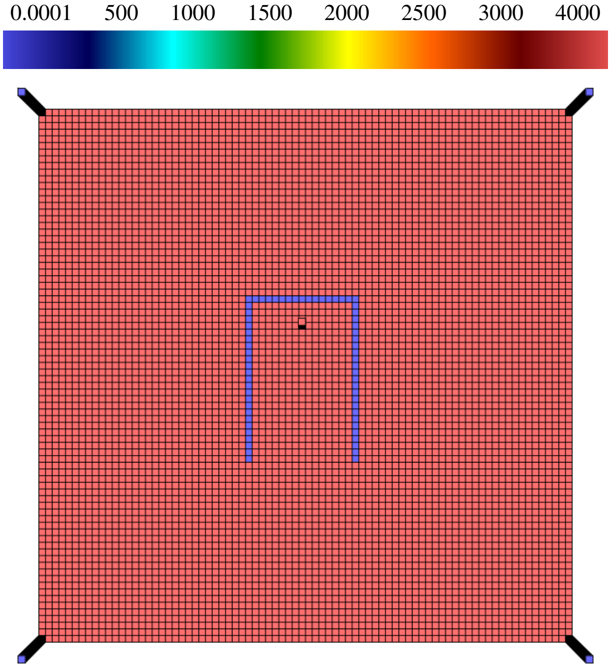

Fig. 6.1 Injection wells (blue) and production well (red) for Example 1. Background shows k_x ($k_x = k_y$).

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.

Run	Unconstr.	Heuristic	Formal
Reference	163.9	152.2	
1	187.5	156.6	158.2
2	189.1	162.0	146.2
3	177.3	149.0	149.2
4	183.4	150.2	**160.6**
5	186.1	152.2	152.9
6	185.0	158.9	160.2
7	190.2	162.3	142.5
8	190.1	**163.5**	158.5
9	190.1	162.0	158.0

Table 6.5 Oil production in 10^3 m^3 (Example 1, **320 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.

Run	Unconstr.	Heuristic	Formal
Reference	150.1	152.5	
1	188.1	149.9	**170.2**
2	192.4	155.6	131.4
3	186.5	136.4	117.8
4	186.8	149.9	133.7
5	186.5	142.2	156.1
6	192.5	157.8	168.2
7	192.5	**159.4**	158.8
8	192.4	156.9	161.1
9	192.2	157.0	165.8

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 again maximize cumulative oil recovery under CO_2 injection. The two-dimensional geological model is the top layer of the model defined in the SPE comparative solution project [11], referred to as SPE 10. The model includes a total of four components, as specified in Table 6.6. Details on the reservoir model 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 again well BHPs, constrained to lie between 150 bar and 50 bar. A maximum (per well) gas production rate of 200 m^3/d at reservoir conditions is also specified. The total simulation period is 1000 days. The well controls are determined at initial time and then at every 100-day interval. There are a total of ten control steps and 80 control parameters in this example.

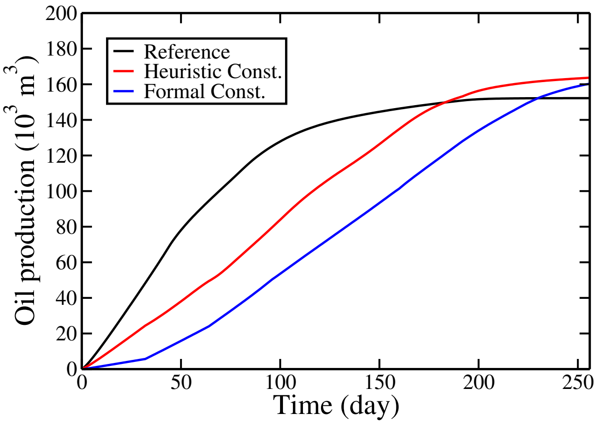

Fig. 6.2 Oil production versus time (Example 1, 40 control variables). Results are for feasible reference case (black curve), best heuristically constrained solution (Run 8, red curve) and best formally constrained solution (Run 4, blue curve).

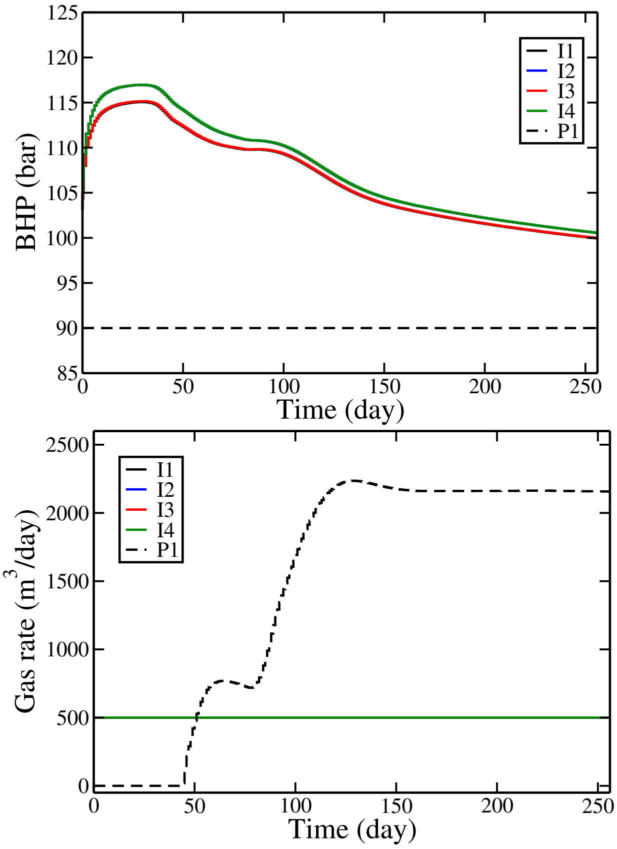

Fig. 6.3 BHPs (top) and gas rates (bottom) for the feasible reference solution (Example 1). Gas injection rates are all equal to 500 m^3/d during the entire simulation period.

We first generate two reference solutions as in the previous example. The cumulative oil production for these two cases, given in the first row ('Reference') of Table 6.8, are nearly identical because the nonlinear constraint violation in the

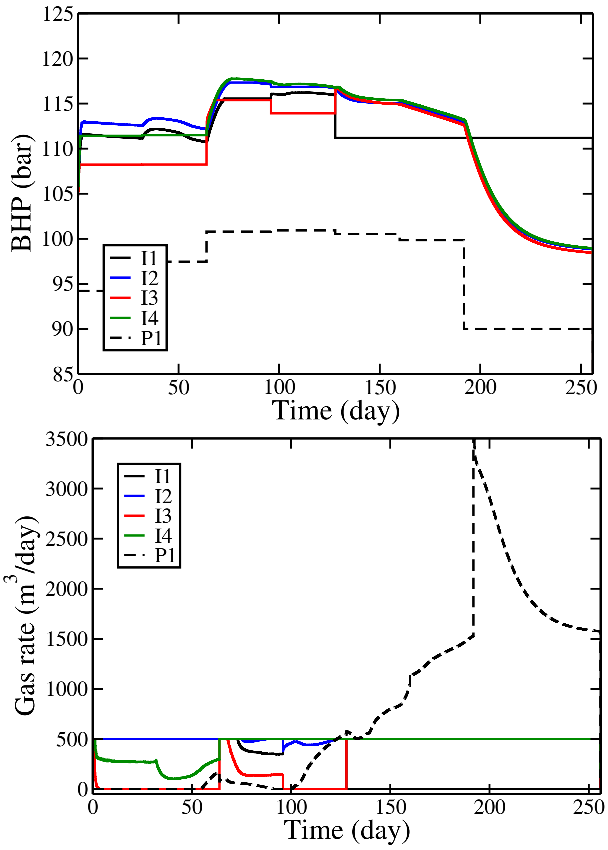

Fig. 6.4 BHPs (top) and gas rates (bottom) for the best heuristically constrained solution using 40 controls (Example 1, Run 8).

Table 6.7 Model parameters for Example 2

Grid size	$60 \times 220 \times 1$	
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

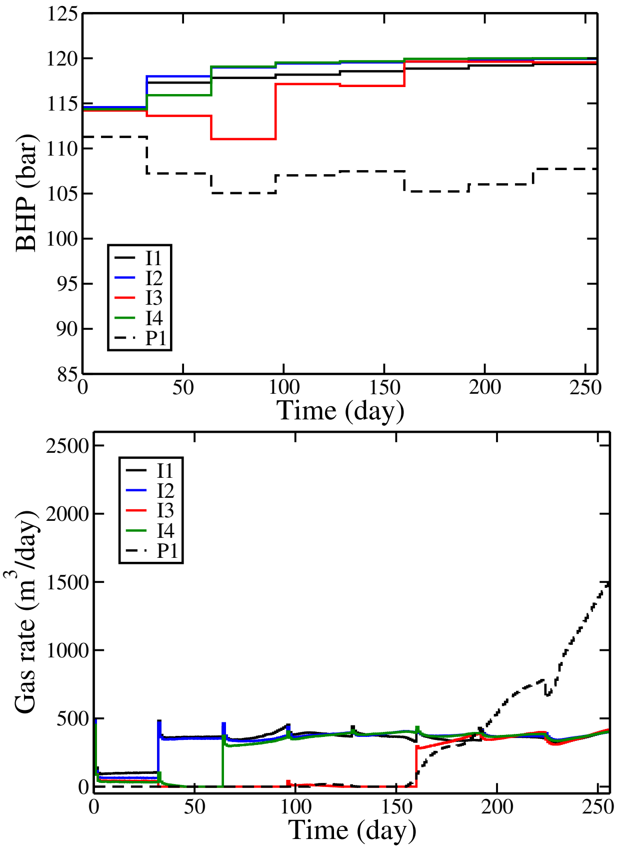

Fig. 6.5 BHPs (top) and gas rates (bottom) for the best formally constrained solution using 40 controls (Example 1, Run 4).

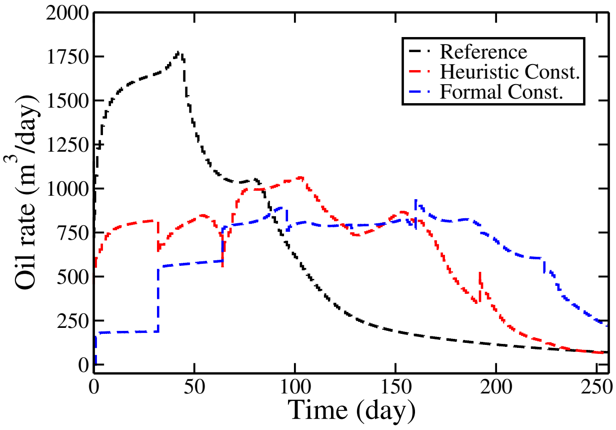

Fig. 6.6 Oil rates for Example 1 (40 controls), reference (black dashed line), heuristically constrained (red dashed line) and formally constrained (blue dashed line).

unconstrained case is small. We next perform (nine) optimizations that honor the bound constraints but not the nonlinear constraints. The best optimum achieved in this case provides a cumulative oil production of 22,130 m^3 (Run 6). Using the heuristic constraint handling procedure (third column), the best result is 22,030 m^3 of oil (Runs 5 and 9), which exceeds the reference heuristic result by 8.8%. The best optimum achieved using the formal constraint handling

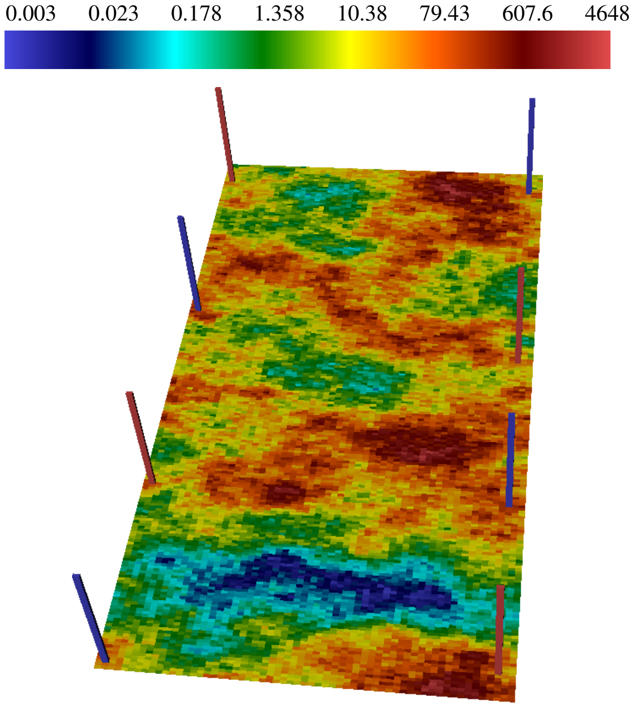

Fig. 6.7 Injection wells (blue) and production wells (red) for Example 2. Background shows $\log \mathsf{k}_x$ ($\mathsf{k}_x = \mathsf{k}_y$).

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	21.77
2	22.01	22.01	22.01
3	18.16	18.16	18.17
4	21.68	21.68	21.68
5	22.03	**22.03**	22.04
6	22.13	20.67	**22.20**
7	21.68	21.68	21.99
8	22.07	21.48	22.07
9	22.03	**22.03**	22.03

treatment is 22,200 m^3 of oil (Run 6). This value exceeds the reference solution by 9.5% and the best result using the heuristic treatment by 0.8%. It also exceeds the best unconstrained result (again, unconstrained here refers to the nonlinear constraints) of 22,130 m^3, which is presumably because the nonlinear constraints are not very important in this example.

Oil production profiles are shown in Fig. 6.8. These profiles are very similar for the runs using the heuristic and formal constraint handling procedures. The BHP, gas rate, and oil rate profiles for the three cases are shown in Figs. 6.9, 6.10 and 6.11. We see that the gas production rates satisfy the nonlinear constraints (200 m^3/d) at all times for both constraint handling procedures. Consistent with Fig. 6.8, the oil production rates in Figs. 6.10 and 6.11 resemble one another, and they are clearly different than the reference solution in Fig. 6.9. For this example, the formal approach required an average of 56 forward simulations, while the heuristic approach required an average of only 24.

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

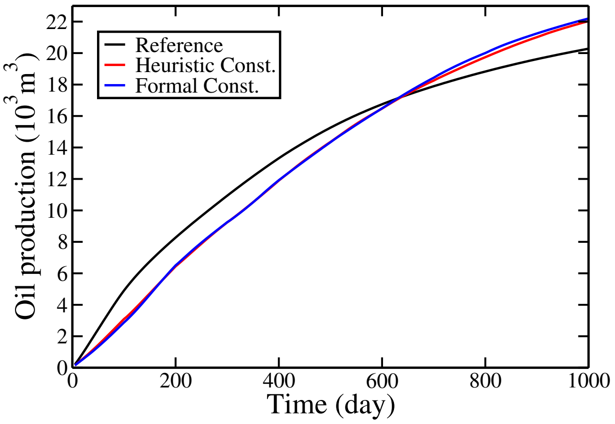

Fig. 6.8 Oil production versus time for Example 2. Results are for feasible reference case (black curve), best heuristically constrained solution (Run 9, red curve) and best formally constrained solution (Run 6, blue curve).

6.3 Example 3: Twelve-well channelized system

Our third example uses the three-dimensional geological model introduced in [37]. We again consider CO_2 injection, though this model contains a total of six components, defined in Table 6.9. Further details are given 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 producers and 40,000 m^3/d for the injectors (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 param-

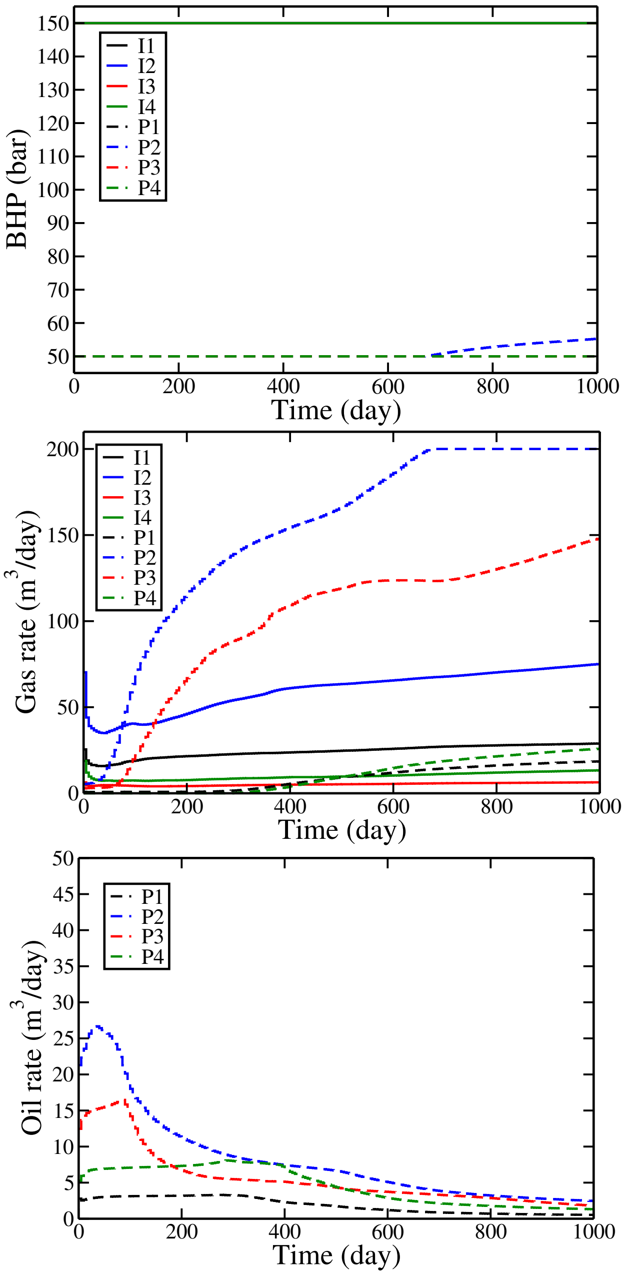

Fig. 6.9 BHPs (top), gas rates (middle) and oil rates (bottom) for the feasible reference solution (Example 2).

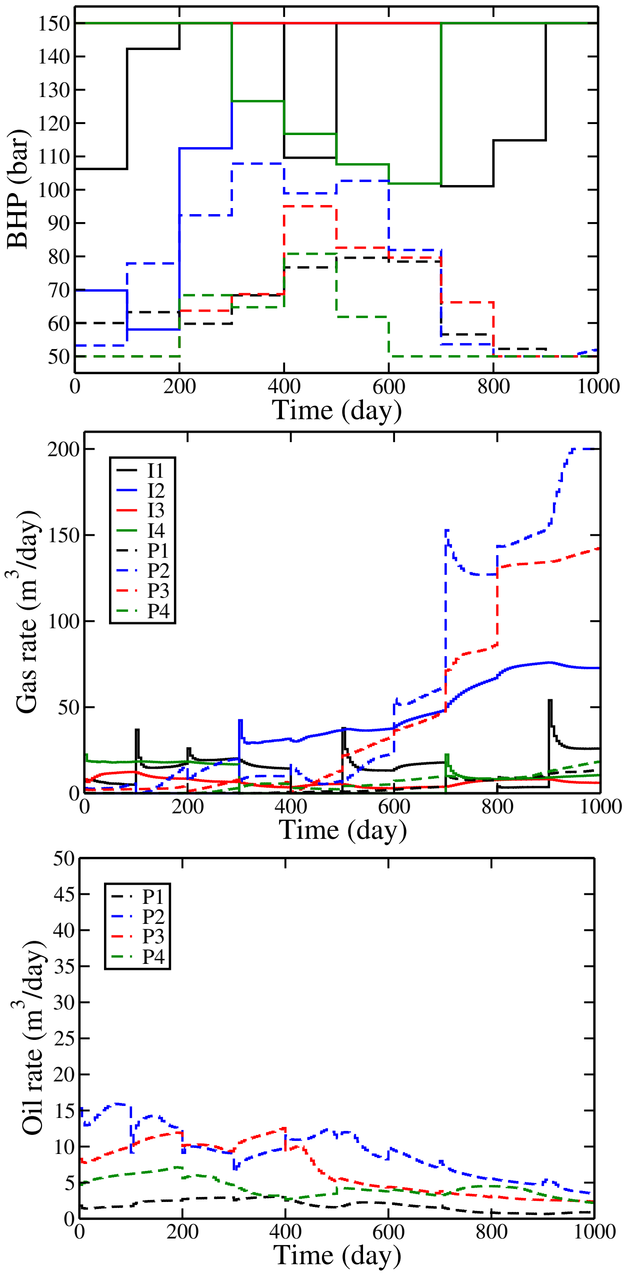

Fig. 6.10 BHPs (top), gas rates (middle) and oil rates (bottom) for the best heuristically constrained solution (Example 2, Run 9).

eters in this problem, and our objective is again to maximize cumulative oil production.

We simulate this model using the same procedures as in the previous examples. 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 case exceeds that of the optimized cases, though the cumulative oil produced in the optimized cases is of course higher.

In this example, convergence of the optimizations using the formal constraint handling approach typically required about 48 forward simulations, while the heuristic treatment required only about 26. Our findings for this example clearly illustrate the potential advantages of the heuristic treatment for complex optimization problems involving multiple wells operating under nonlinear constraints.

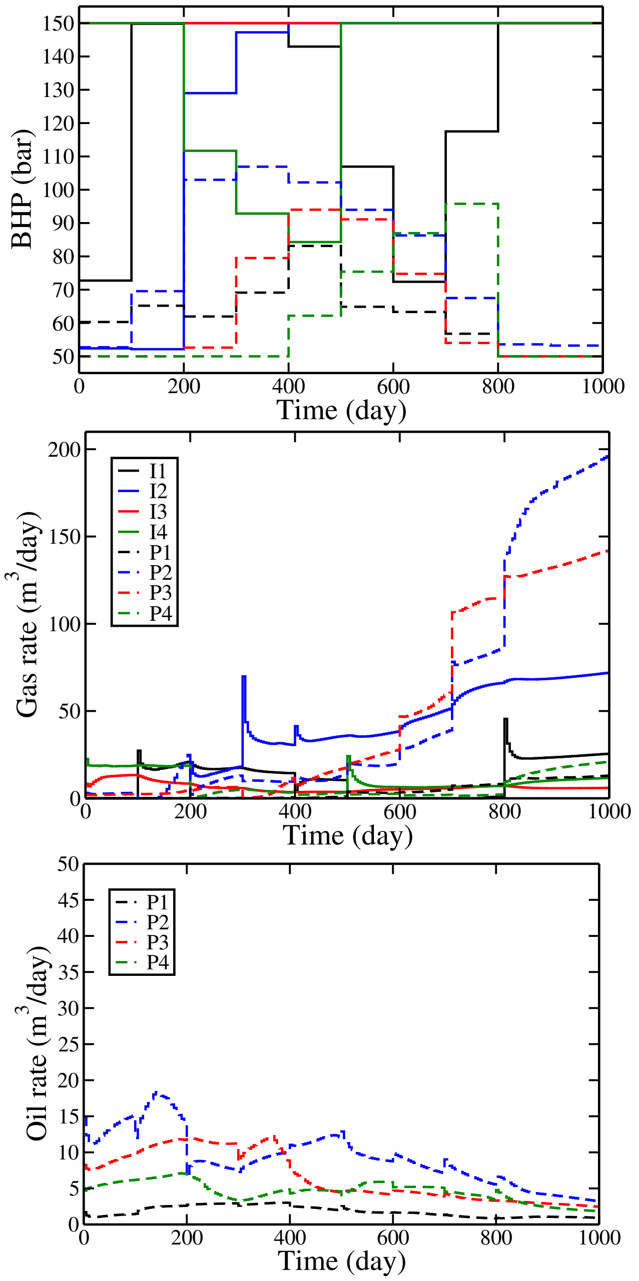

Fig. 6.11 BHPs (top), gas rates (middle) and oil rates (bottom) for the best formally constrained solution (Example 2, Run 6).

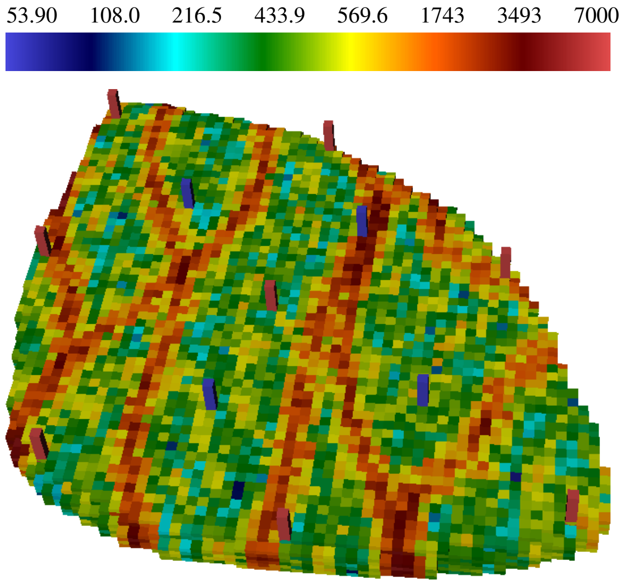

Fig. 6.12 Reservoir model and wells for Example 3 (from [37]). Background shows $\log k_x$.

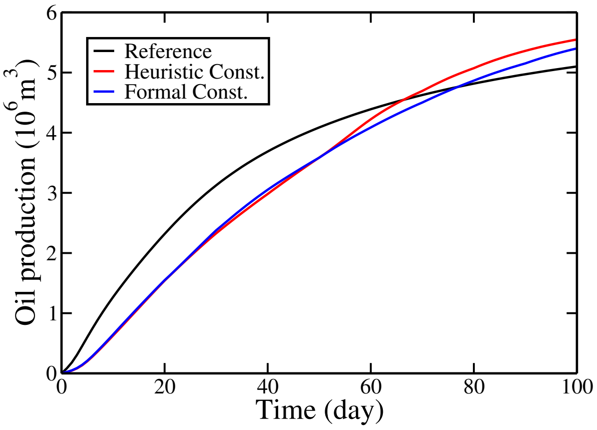

Fig. 6.13 Oil production versus time for Example 3. Results are for feasible reference case (black curve), best heuristically constrained solution (Run 2, red curve) and best formally constrained solution (Run 3, blue curve).

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 [29]. 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, 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). 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 m^3/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	$60 \times 60 \times 7$	
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	100	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

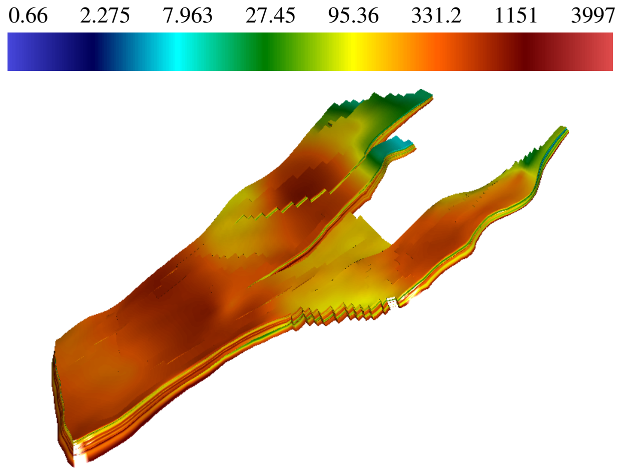

Fig. 6.14 Permeability field in log scale for Example 4 (Norne model).

Table 6.13 Model parameters for Example 4 (Norne model)

Grid size	$46 \times 112 \times 22$	
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	-

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 large effect in this example. Applying these constraints heuristically, the best maximum obtained is 147.8×10^6 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 a result for cumulative oil production (138.5×10^6 m^3 in the best case, Run 1) that is lower than that for the heuristic reference case, which does not involve any optimization. The results using the formal treatment illustrate the potential challenges that can arise in complex problems with large numbers of control parameters and many active nonlinear constraints. It is worth reiterating that the heuristic approach does, even in this challenging case, provide improvement over the feasible reference solution.

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	252.0	142.0	
1	270.2	146.0	**138.5**
2	266.7	143.3	138.3
3	261.6	124.3	129.3
4	270.3	**147.8**	137.5
5	270.7	146.5	138.0
6	268.8	146.8	137.8
7	270.4	146.2	129.7
8	270.8	146.8	129.9
9	270.6	146.2	136.6

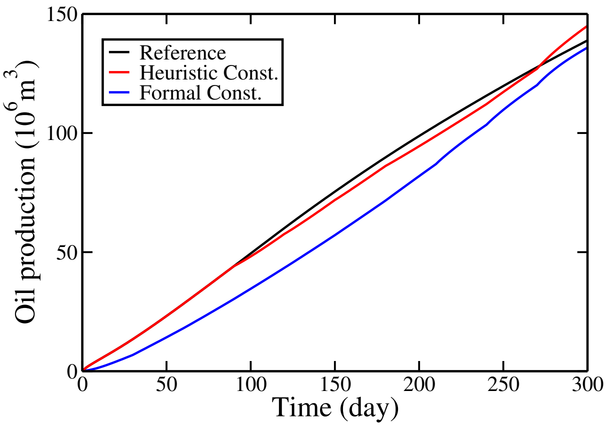

Fig. 6.15 Oil production versus time for Example 4. Results are for feasible reference case (black curve), best heuristically constrained solution (Run 4, red curve) and best formally constrained solution (Run 1, blue curve).

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 must be held constant over the entire 30-day 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 constraint handling required about 178 forward simulations on average. Optimizations using the heuristic constraint handling, by contrast, required an average of only about 36 forward simulations. Thus, for this example we observe a very substantial improvement in computational efficiency using the heuristic approach.

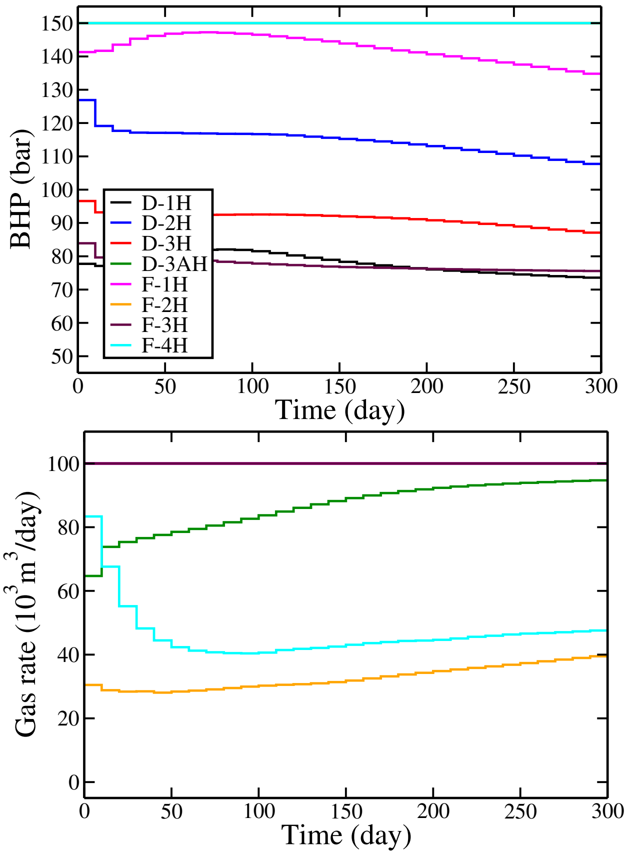

Fig. 6.16 Injector BHPs (top) and gas injection rates (bottom) for the feasible reference solution (Example 4).

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 prob-

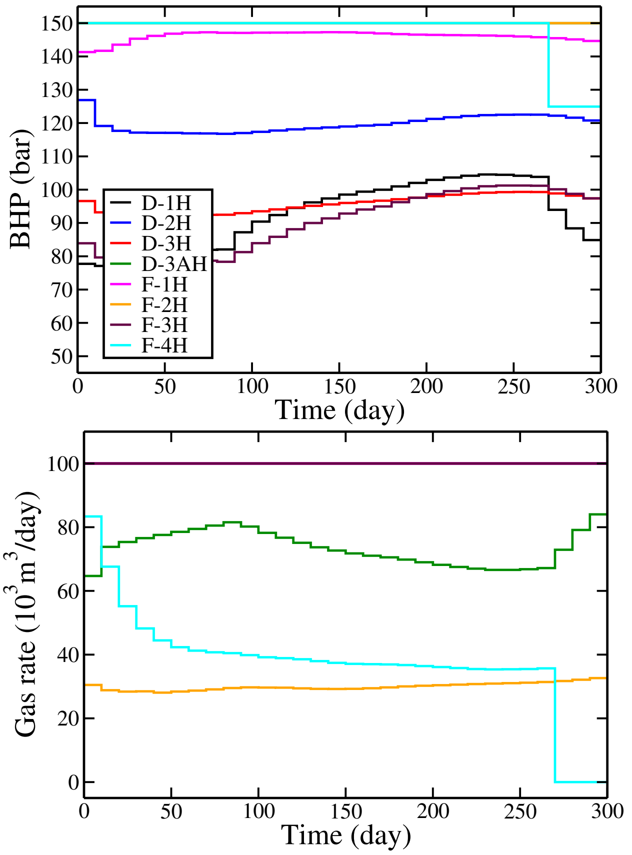

Fig. 6.17 Injector BHPs (top) and gas injection rates (bottom) for the best heuristically constrained solution (Example 4, Run 4).

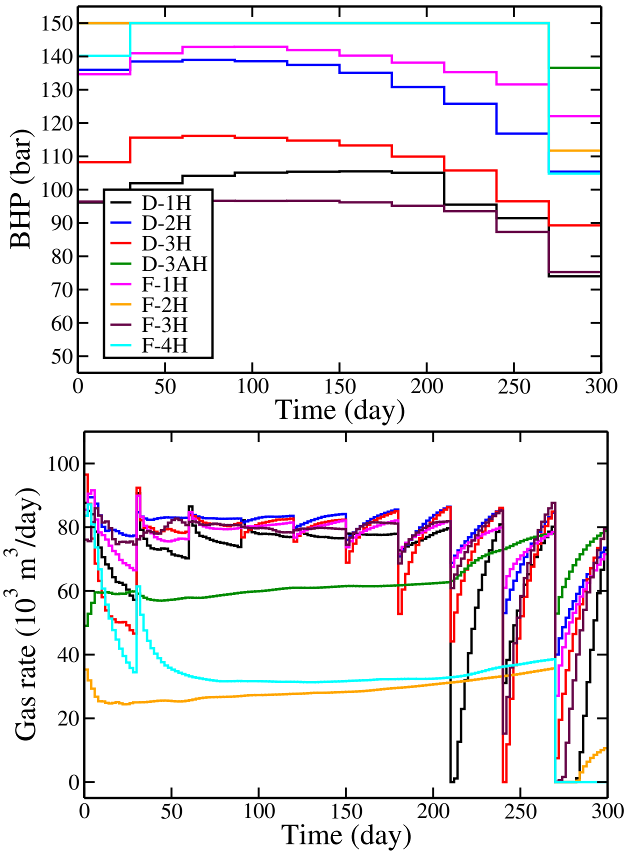

Fig. 6.18 Injector BHPs (top) and gas injection rates (bottom) for the best formally constrained solution (Example 4, Run 1).

lem in which BHPs are the control variables) as required to satisfy the nonlinear rate constraints.

Numerical results were presented for four 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 our examples, the control variables were the time-varying well BHPs, and maximum injection or production rate specifications entered as nonlinear constraints. The total number of control variables ranged from 40 to 320. Improvement in cumulative oil produced (which was the objective function in all cases) using our optimization procedures ranged from 4.2% to 11.6% relative to the reference solutions.

In the simplest case (Example 1), the formal constraint handling approach was shown to outperform the heuristic procedure when we increased the number of control variables from 40 to 320. In the next (somewhat more complicated) case, Example 2, the formal constraint treatment continued to outperform the heuristic procedure, though its advantage was very slight. In the other two cases, which were more challenging in terms of model size and number of wells (and they involved three-dimensional models, while the first two examples were two-dimensional), the heuristic treatment provided better objective function values than the formal approach.

These observations suggest that, although the formal constraint handling approach is theoretically superior, complications related to constraint lumping and the existence of poor local optima (and the additional complexity inherent in problems with large numbers of optimization variables and nonlinear constraints) may render the formal procedure less effective than the heuristic approach in challenging cases. Thus, though we expect (and observe) the formal procedure to be the method of choice in relatively simple cases, the heuristic approach should be considered for use in more complex problems. It may even be beneficial to apply some type of hybrid technique, where the result from the heuristic method is used as the initial guess for the formal procedure. The use of a sequence of optimizations, with increasing numbers of control periods, should also be considered. Finally, it is important to note that the heuristic constraint treatment was found to be significantly 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 conditions and as a result the gradients do not agree in general. However, the two boundary conditions become almost identical as the size of the last time step approaches zero, in which case 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 overall approach can be extended to perform robust control (to account for geological uncertainty) and hierarchical (multi-objective) control to balance long-term and short-term objectives. Finally, our procedures 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 for useful discussions and assistance with AD-GPRS, and Michael Saunders for his support on SNOPT. We are grateful to the industrial affiliates 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.R., Jansen, J.D.: 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. Chen, Y., Oliver, D.: Ensemble-based closed-loop optimization applied to Brugge field. SPE Reservoir Evaluation & Engineering **13**(1), 56–71 (2010)
11. Christie, M., Blunt, M.: Tenth SPE comparative solution project: a comparison of upscaling techniques. SPE Reservoir Evaluation and Engineering **4**(4), 308–317 (2001)
12. Coats, K.: An equation of state compositional model. SPE Journal **20**(5), 363–376 (1980)
13. De Montleau, P., Cominelli, A., Neylon, K., Rowan, D., Pallister, I., Tesaker, O., Nygard, I.: Production optimization under constraints using adjoint gradients. In: 10th European Conference on the Mathematics of Oil Recovery (2006)
14. 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)
15. Echeverría Ciaurri, D., Isebor, O.J., Durlofsky, L.J.: Application of derivative-free methologies to generally constrained oil production optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation **2**, 134–161 (2011)
16. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM Rev. **47**(1), 99–131 (2005)
17. Gunzburger, M.: Perspectives in Flow Control and Optimization. SIAM (2003)
18. 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
19. Han, C., Wallis, J., Sarma, P., Li, G., Schrader, M.L., Chen, W.: Adaptation of the CPR preconditioner for efficient solution of the adjoint equation. SPE Journal **18**(2), 207–213 (2013). DOI 10.2118/141300-PA
20. 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
21. Kraaijevanger, J., Egberts, P., Valstar, J., Buurman, H.: Optimal waterflood design using the adjoint method. In: Paper SPE 105764 presented at the SPE Reservoir Simulation Symposium. Houston, USA (2007). DOI DOI:10.2118/105764-MS
22. 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

23. Lien, M., Brouwer, D., Manseth, T., Jansen, J.D.: Multiscale regularization of flooding optimization for smart field management. SPE Journal **13**(2), 195–204 (2008)
24. Liu, W., Ramirez, W., Qi, Y.: Optimal control of steam flooding. SPE Advanced Technology Series **1**(2), 73–82 (1993)
25. 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)
26. 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)
27. 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
28. 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)
29. NTNU (IO Center): Center of Integrated Operations in Petroleum Industry. Website (2011). `www.ipt.ntnu.no/~norne`
30. Oliver, D., Reynolds, A., Liu, N.: Inverse Theory for Petroleum Reservoir Characterization and History Matching. Cambridge University Press, Cambridge (2008)
31. Ramirez, W.: Application of Optimal Control Theory to Enhanced Oil Recovery. Elsevier Science Ltd, Amsterdam (1987)
32. Sarma, P., Chen, W., Durlofsky, L.J., Aziz, K.: Production optimization with adjoint models under nonlinear control-state path inequality constraints. SPE Reservoir Evaluation and Engineering **11**(2), 326–339 (2008)
33. 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
34. Stengel, R.: Stochastic Optimal Control Theory and Application. Wiley, New York (1986)
35. Sudaryanto, B., Yortsos, Y.: Optimization of fluid front dynamics in porous media using rate control. Physics of Fluids **12**(7), 1656–1670 (2000)
36. Suwartadi, E., Krogstad, S., Foss, B.: Nonlinear output constraints handling for production optimization of oil reservoirs. Computational Geosciences **16**, 499–517 (2012)
37. 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). DOI 10.2118/102913-PA
38. Virnovski, G.: Waterflooding strategy design using optimal control theory. In: 6th European IOR Symposium, pp. 437–446. Stavanger, Norway (1991)
39. 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
40. 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)
41. 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
42. Wang, C., Li, G., Reynolds, A.: Production optimization in closed-loop reservoir management. SPE Journal **14**(3), 506–523 (2009)
43. 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
44. 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
45. 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