

The Application of Flow Diagnostics for Reservoir Management

Olav Møyner
SINTEF ICT

Stein Krogstad
SINTEF ICT

Knut-Andreas Lie
SINTEF ICT

June 19, 2014

Flow-diagnostics, as referred to herein, are computational tools based on controlled numerical flow experiments that yield quantitative information regarding the flow behavior of a reservoir model in settings much simpler than would be encountered in the actual field. In contrast to output from traditional reservoir simulators, flow diagnostic measures can be obtained within seconds. The methodology can be used to evaluate, rank and/or compare realizations or strategies, and the computational speed makes it ideal for interactive visualization output. We also consider application of flow diagnostics as proxies in optimization of reservoir management workflows. In particular, based on finite volume discretizations for pressure, time-of-flight (TOF) and stationary tracer, we efficiently compute general Lorenz coefficients (and variants) which are shown to correlate well with simulated recovery. For efficient optimization, we develop an adjoint code for gradient computations of the considered flow diagnostic measures. We present several numerical examples including optimization of rates, well-placements and drilling sequences for two and three phase synthetic and real field models. Overall, optimizing the diagnostic measures imply substantial improvement in simulation-based objectives.

1 Introduction

Computational tools for reservoir modeling play a critical role in the development of strategies for optimal recovery of hydrocarbon resources. These tools can be simply viewed as a means of forecasting recovery given a set of data, assumptions, and operating constraints, e.g., to validate alternative hypotheses about the reservoir or systematically explore different strategies for optimal recovery.

By nature, reservoir modeling is an inter-disciplinary exercise and reservoir modeling tools must be well integrated to promote collaboration between scientists and engineers with different backgrounds. New workflows are emerging based on recent advances in static reservoir characterization and dynamic flow simulation. Modern numerical flow simulators have evolved to include more general grids, complex fluid descriptions, flow physics, well controls, and couplings to surface facilities. These generalizations have helped to more realistically describe fluid flow in the reservoir on the time scales associated with reservoir management. Meanwhile, modern reservoir characterization techniques have shifted away from traditional variogram-based models towards object- and feature-based models that more accurately describe real geologic structures. To quantify uncertainty in the characterization, it is necessary to generate an ensemble of reservoir models that may include a thousand or more individual realizations. Reservoir simulation is computationally demanding and a

single simulation on a full reservoir model may require from a few tens of minutes to hours or even days. Direct evaluation of multiple production scenarios on large ensembles of earth models is therefore impractical with full-featured flow simulators, and the computational gap has traditionally been overcome by upscaling methods. Another alternative is to cluster the models based on fast proxies for flow behavior and then select an appropriate subset of models for flow simulation [33, 14].

In many geomodelling workflows, there is no time to wait for a reservoir simulation, or there is no call for the complexity of a full simulation to produce the desired information. In this paper, we will discuss flow-diagnostic tools that are designed for this purpose. Flow-diagnostic tools are based on controlled numerical flow experiments that yield quantitative information regarding the flow behavior of a model in settings much simpler than would be encountered in the actual field. While full-featured simulators are capable of making these predictions, they generally cannot do so in a computationally efficient manner unless an unacceptably large degree of upscaling is applied. Industrial applications require fast tools that can be applied directly to high resolution reservoir models. Flow-diagnostic tools should offer interactive rates when applied to a single model and the ability to process large ensembles in the computational time currently associated with a single full-featured flow simulation. There is a specific need for fast and reliable measures of flow behavior that are suitable for comparing, ranking, and clustering models. In addition, effective flow-diagnostics tools should assist in the challenging task of integrating data into reservoir models through fast screening of multiple scenarios, identifying regions associated with mismatches, and suggesting appropriate model updates. While these challenges are not entirely new, the demand for new technology is driven by a need for industry to adopt more rigorous uncertainty quantification workflows and calibrate models with increasing volumes of data from production logs and well tests.

Another area where flow diagnostics may prove to be useful is within the field-development planning and execution of the production plans. Here, the ongoing digitalization opens up for decision-support tools that use mathematical models to structure expert knowledge with increasing amounts of data to facilitate faster and better decisions. That is, models with embedded uncertainty can be combined with rigorous optimization techniques to improve decisions like well location, drilling sequence, well completion including multiple inflow control devices, and choice of time-varying rate/pressure targets. Even though computing power is steadily increasing, the high computational cost of a full forward reservoir simulation still limits the number of trials/iterations one can afford in search for an optimal strategy. In most optimization loops (manual or automatic), the model input only changes slightly from one simulation to the next and this can be utilized to develop efficient upscaling and multiscale methods or reduced-order modeling techniques such as proper orthogonal decomposition. Herein, we will present an alternative approach that makes it possible to optimize well rates and placements based on objective functions defined by time-of-flight, pressure and tracers. In particular, we will present an adjoint formulation based on time-of-flight equations. This approach has many advantages: It can be used for long- or short-term optimization using different types of objective functions while incorporating complex physics effects by using a pre-existing pressure solver and it only requires a modest number of pressure evaluations compared to fully fledged simulator runs.

Previous attempts to develop flow-diagnostic tools have focused on streamline-based simulation techniques [9], and resulting diagnostic tools have been applied e.g., in ranking and upscaling of geostatistical models [16, 5], to optimize well rates in waterflooding [37, 31, 17], for flood surveillance on a pattern-by-pattern basis

[7], and to optimize fracture stages and horizontal well completions in tight gas reservoirs [34]. While streamline-based methods can be very effective for these and similar purposes, they have limitations both in terms of computational complexity and extensibility. Streamline tracing is highly efficient for standard reservoir models composed of hexahedral cells, but special treatment is required to handle mismatched cells and proposed extensions to unstructured grid models are far less efficient, see e.g., [15, 32]. Hence, use of streamlines may be precluded by efficiency considerations for complex earth models that contain unstructured grid cells, or difficult to apply for multi-continuum models. Moreover, from a visual point of view, volumetric information about reservoir partitioning and communication, such as swept and drainage zones, may be easier to interpret and distinguish when presented using grid cells instead of complex bundles of intertwined streamlines.

Ideally, flow-diagnostic tools should require input parameters that are available within existing workflows and be compatible with existing simulators. The majority of simulators used in commercial applications are based on finite-volume methods and are capable of simulating fluid flow on stratigraphic grids with complex geometry and structured and unstructured topography. The flow-diagnostic tools should also be applicable to fully unstructured grids with general polyhedral cells, multicontinuum models, and models containing lower-dimensional objects (discrete-fracture models), even though these are less common. Herein, we present an approach to flow diagnostics that is based on solving time-of-flight and stationary tracer equations using standard finite-volume discretizations [27, 3, 12, 28, 1, 13]; or in other words, reusing general data structures, discretization schemes, and solvers that are robust and already have widespread applicability within industry. From the time-of-flight and tracer distribution, one can easily find time-lines in the reservoir, extract information along these lines, and partition the reservoir into volumes associated with injector/producer pairs. As recently demonstrated by Shahvali et al. [35], time-of-flight and tracer distributions can also be used to compute well-allocation factors or assess the heterogeneity of the reservoir in terms of a flow-capacity/storage-capacity diagram, which can further be used to estimate sweep efficiency and compute various heterogeneity measures such as the Lorenz coefficient. While flow-diagnostics computations are not simulations in the traditional sense, they can, within the order of seconds, provide fast and reliable measures of flow behavior that are suitable for interactive visualization and for comparing, ranking, and clustering models. In addition, the robustness of flow-diagnostic tools may benefit significantly from the mass conservation inherent in finite-volume methods.

The methods presented in the following have all been implemented using the MATLAB Reservoir Simulation Toolbox [25, 22] and are available as free open source software, partially embedded as part of an interactive visualization tool.

2 Flow Diagnostics

Flow diagnostics, as discussed herein, are obtained by analyzing the properties of the flow field. In its simplest form, a representative flow field can be computed by solving an incompressible pressure equation,

$$\nabla \cdot \vec{v} = q, \quad \vec{v} = -K\lambda\nabla p.$$

Here, the primary unknowns are the pressure p and the fluid velocities \vec{v} , whereas K denotes the absolute permeability of the medium. If information is available about the fluid distribution in the reservoir, the flow equations are solved using the multiphase mobilities $\lambda = \lambda(S_1, \dots, S_n)$, and if not, one assumes a single-phase

model for which $\lambda \equiv 1$. Alternatively, the fluxes (and pressures) can be exported from a dynamic (multiphase) reservoir simulation that models more advanced flow physics than the simple incompressible equation above.

In this section, we will outline basic applications of flow diagnostics. Several ideas are similar to those presented in [35], but are here illustrated using models of two real reservoirs: the Gullfaks Field from the North Sea with an artificial well configuration, and a simulation model of the Norne Field from the Norwegian Sea [29, Package 2].

2.1 Time-of-flight

To study how heterogeneity affects flow patterns and define natural time-lines in the reservoir, it is common to study the so-called time-of-flight (TOF), i.e., the time it takes an imaginary particle released at an inflow boundary or at a perforation of an injector to reach a given point in the reservoir. Time-of-flight is usually associated with streamline methods, but can also be computed from linear steady-state transport equations on the form,

$$\begin{aligned} \vec{v} \cdot \nabla \tau_f &= \phi, & \tau_f|_{\text{inflow}} &= 0, \\ -\vec{v} \cdot \nabla \tau_b &= \phi, & \tau_b|_{\text{outflow}} &= 0, \end{aligned} \tag{1}$$

where ϕ is the porosity of the medium. The forward TOF τ_f is the time required for a tracer injected at an injector/inflow boundary to arrive at specific position in the reservoir, whereas the backward TOF τ_b is the time it takes a tracer released at a given location within the reservoir to reach a producer/outflow boundary; see Figure 3. Studying iso-contours of time-of-flight will give an indication of how more complex multiphase displacements may evolve under fixed well and boundary conditions, and will reveal more information about the flow field than pressure and velocities alone. However, beyond providing approximate time-lines for fluid transport, time-of-flight can be used to identify non-targeted regions: regions with high TOF are likely to remain unswept and are obvious targets to investigate for placement of new wells.

2.2 Tracer partitions

Communication patterns within the reservoir can be determined by simulating the evolution of artificial, neutral tracer with concentration c into injection wells or fluid sources. A simple tracer test is to set the tracer concentration equal one in only one injection well, a well completion, or (parts of) an inflow boundary and compute the steady-state that the solution approaches at late times. The steady state cannot generally be achieved in field experiments, but is easy to compute numerically. Hence, one can easily partition a model into swept volumes by repeating the tracer test for each well, well completion, or part of the inflow boundary. Moreover, one can equally well reverse the flow field and compute similar tracer distributions associated with producers and outflow boundaries. Altogether, this gives two types of transport equations,

$$\begin{aligned} \nabla \cdot (\vec{v}c_i) &= 0, & c_i|_{\text{inflow}} &= 1 \\ -\nabla \cdot (\vec{v}c_p) &= 0, & c_p|_{\text{outflow}} &= 1. \end{aligned} \tag{2}$$

The resulting tracer partitions give the portion of volume that *eventually* will come from a given injector or arrive at a given producer if the driving forces used to compute the snapshot of the flux field prevails; see Figures 1 and 2.

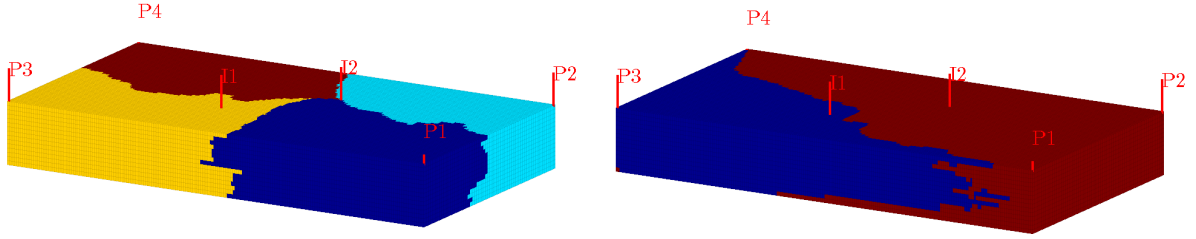


Figure 1: Stationary tracer distributions for producers (left) and injectors (right) for a slightly modified version of the SPE10 model.

2.3 Computational methods

Under the assumption of incompressible flow, the time-of-flight and steady-state tracer equations (1) and (2) are linear transport equations on the form $\nabla \cdot (\vec{v}w) = b$, where \vec{v} is the velocity field, w is the unknown quantity, and b equals zero when computing tracer concentration or b equals the porosity when solving for time-of-flight. Introducing a standard single-point upwind (SPU) finite-volume discretization results in reducible, block-structured linear systems $Ax = b$ that can be solved very efficiently. To illustrate this, we consider Model 2 from the Tenth SPE Comparative Solution Project [8]. This Cartesian box model consists of approximately 1.1 million grid cells and is used widely throughout the literature as a standard benchmark for new simulation methods. In our setup we replace the central injector of the original five-spot well configuration by two injectors that are moved a short distance from the model center, see Figure 1. Altogether, the flow-diagnostics computation will require one pressure solution, two time-of-flight solutions, one tracer solution with two different right-hand sides for the injectors, and one tracer solution with four right-hand sides for the producers. In MATLAB, computing the flow field took 7.3 seconds on a standard workstation PC using a highly efficient algebraic multigrid solver [30], whereas computing time-of-flight and tracers took 4.9 seconds with MATLAB’s standard direct solver, which utilizes the special structure of the equations and solves for multiple right-hand sides in one pass.

To understand the structure of the linear system, we can view the fluxes from the flow problem as edges in a directed graph, and then use a depth-first search of the flux graphs (topological sort) to permute the discretization matrix A , which is identical for the time-of-flight and tracer equations, to a lower block-triangular form $L = PAP^T$, see [10, 28]. Solving the linear systems thus reduces to an efficient block-wise back-substitution algorithm¹ that inverts a sequence of smaller linear problems corresponding to each irreducible diagonal block of L . The size of each diagonal block is given by the number of degrees of freedom in the corresponding grid cell (or group of mutually dependent cells if \vec{v} contains cycles). The resulting computational complexity is close to optimal in the linear case: if LU-factorization is used for each subsystem, the asymptotic complexity is linear in the number of irreducible blocks and roughly cubed in the number of unknowns per cell (for a higher-order scheme). Hence, if one uses a first-order discretization, the time-of-flight in a model with N cells can be computed in $\mathcal{O}(N)$ operations. Likewise, solves involving a single well can be performed in $\mathcal{O}(N_c^w)$ operations, where N_c^w is the number of cells downstream of the well.

¹The same idea can be applied to a general family of transport equations—if one first introduces an implicit temporal discretization—resulting in a very effective nonlinear Gauss–Seidel method with local control of the nonlinear iterations [26, 23].

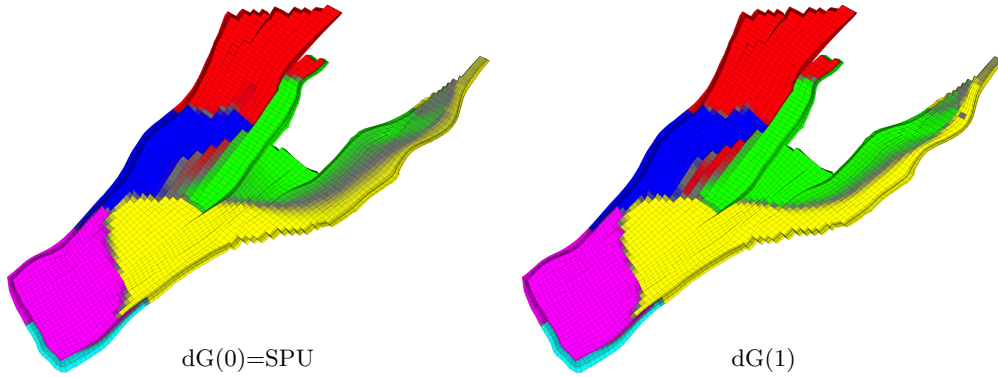


Figure 2: Computation of tracer partitions for the Norne Field using a discontinuous Galerkin method with zeroth and first order polynomials. Colors represent unique tracers, whereas gray colors represent cells in which more than one tracer component is nonzero. The well positions are for illustration purposes only and have nothing to do with the real field.

The SPU method can be applied to compute simple flow diagnostics on general grids and gives acceptable results on smooth heterogeneities, but may introduce too much smearing for highly heterogeneous cases and inaccurate results for grids with large contrasts in cell sizes. Shahvali et al. [35] demonstrated how a truly multi-dimensional upstream-weighting scheme can be applied to reduce transverse diffusion on Cartesian grids. Similarly, higher-order discontinuous Galerkin discretizations have been developed and applied successfully on Cartesian grids [28] and on simplexes [13]. These methods have the advantage that they preserve the ordering of the SPU method because all degrees of freedom are localized to the cell and can hence be combined with the ordering method discussed above to give a computational complexity of $\mathcal{O}(NM^\alpha)$, given a total cost of $\mathcal{O}(M^\alpha)$ for solving the local systems with M degrees of freedom inside each cell. More details about how to extend the Galerkin methods to stratigraphic grids will be given in a forthcoming paper. Here, we include a figure that demonstrates the increased resolution one can expect by going to higher order, see Figure 2.

2.4 Visualization of flow diagnostics

The most obvious use of tracer partitions and time-of-flight values is to use these cell-based values as a basis to provide improved visualization of flow patterns. Tracer partitions have utility in identifying the region swept by an injector or the fraction of fluid production attributed to a particular injector or completion. Likewise, drained regions and injection allocation can be determined by reversing the flow field. In Figure 3, drainage and swept volumes are defined using a majority vote over tracer concentrations that assigns each cell to a specific injector or producer. Communication between wells is obtained by intersecting drainage and swept volumes, and the resulting volume partition can be used to identify the pore volume associated with each well pair, or to compute well allocation factors, i.e., the fraction of the producer's inflow that can be attributed to a given indicator. In the bottom row of Figure 3, the pore volumes shared by producer P5 and injectors I1 to I8 are shown as a pie chart, whereas the flux allocation in P5 is shown as a function of depth. The figure also demonstrates how a very intuitive visualization of the evolution of injected fluids is obtained by combining swept volumes with the forward time-of-flight.

All the different examples of flow visualization discussed above have been im-

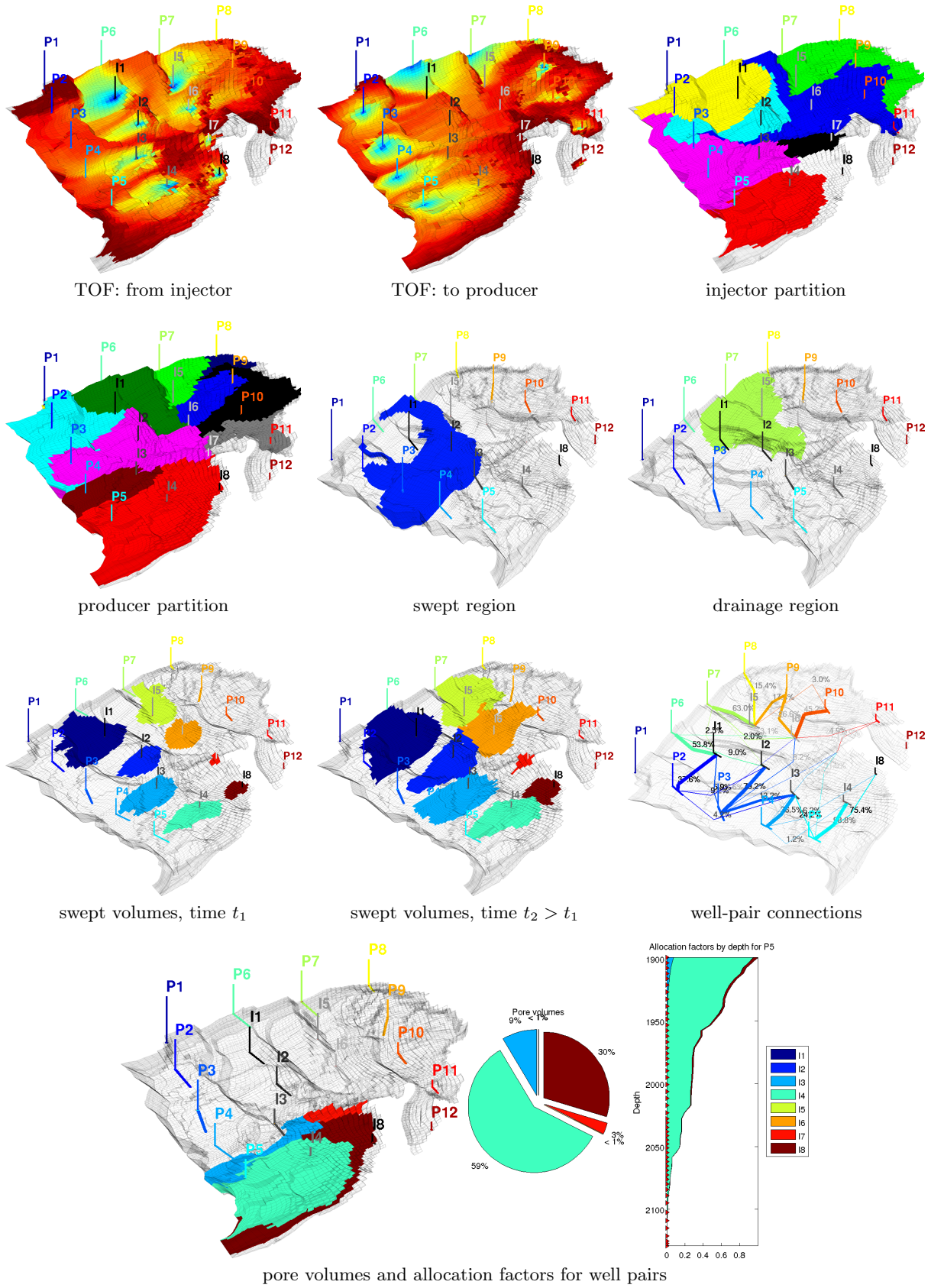


Figure 3: Flow diagnostics based on time-of-flight and tracer partitions for the Gullfaks field. The well positions are for illustration purposes only and have nothing to do with the real field.

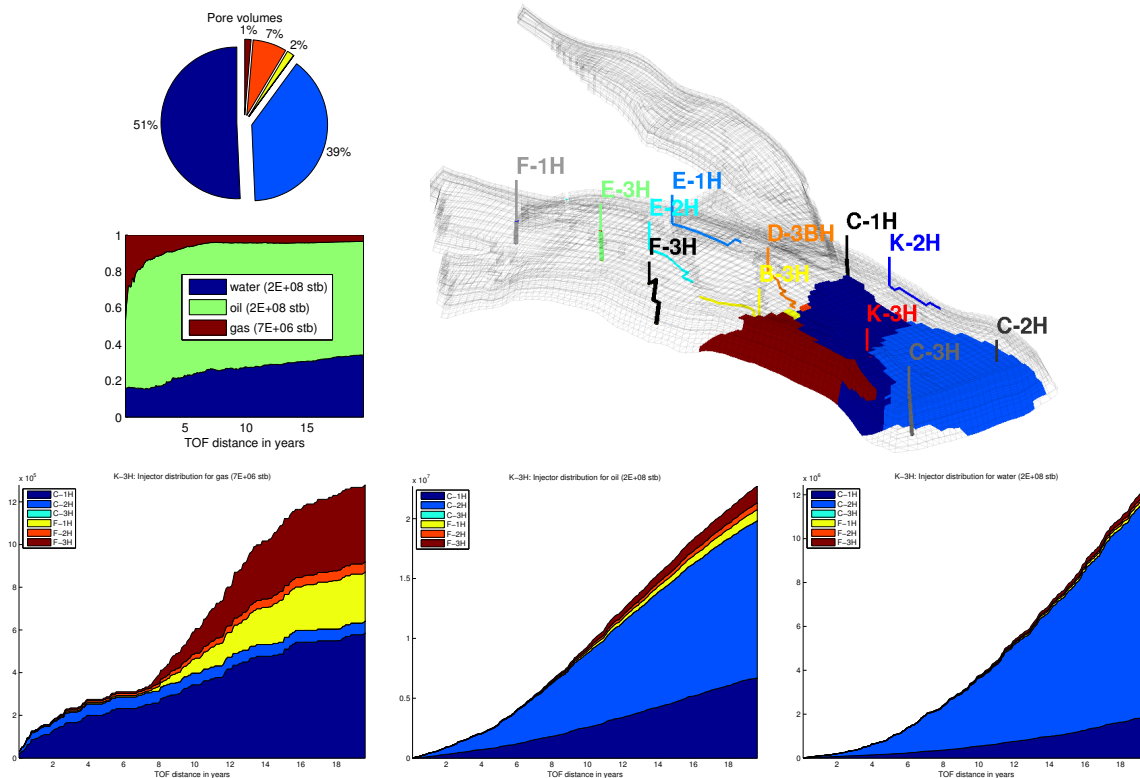


Figure 4: Use of flow diagnostics for interactive post-processing of simulation results for the Norne Field model. Well configuration is for illustration purpose only and does not portray that of the real field.

plemented in the diagnostics module of MRST [25] as part of a simple interactive visualization tool. In our simple prototype tool, all flow diagnostics necessary is computed up front and can be recalculated interactively if the user chooses to modify the well configuration using the simple editor that is part of the tool. Whereas the fast simulation responses allow for great interactivity in MATLAB for models with tens of wells or individual perforations and model sizes up to a million cells, it is generally not possible for more complex models with hundreds of wells or individual perforations or multimillion cells. Here, flow diagnostics should be computed on demand, utilizing the reordering methods to effectively localize the computation of time-of-flight and tracers to regions of interest, e.g., when inspecting individual perforations. Likewise, pressure solutions can be recomputed efficiently using a multiscale method [11, 2, 4], or possibly using model-reduction techniques tuned to previous flow simulations [19].

Example: the Norne field. In the following, we will briefly demonstrate how grid-based time-of-flight and tracer partitions can be used for post-processing simulation results (fluxes and saturation distribution). To this end, we will consider a somewhat simplified simulation model of the Norne field from the Norwegian Sea [29, 1]. Figure 4 focuses on producer 'K-3H' and its drainage region shown in the upper-right plot (delineated using a majority vote). The pie chart to the upper left shows that this well is mainly connected to injectors 'C-1H' and 'C-2H' which influence 51% and 39% of the K-3H's drainage volume. Looking at fluid saturation in the drainage region as function of τ_b (left, middle row), we see that the oil production is likely to increase significantly within the next years and that there will be a large drop

in the gas-oil ratio (GOR) in the well. Further insight can be obtained by looking at how the gas, oil, and water are distributed in different injector-producer regions (lower row). These plots indicate that almost all the current oil production can be attributed to injector 'C-2H', but that injector most likely 'C-1H' will contribute with approximately one third of the oil production within a short period. Likewise, the distribution of gas indicates that approximately 80% of the current gas production can be attributed to 'C-1H'. We also see that there is significant gas and oil in the region supported by injectors 'F-1H' and 'F-3H'. From the plots we cannot generally tell to what extent these hydrocarbons will reach the producer, and when they possibly will do so, since the plots are based on saturations and not mobilities and use the time-of-flight of a neutral particle as a time-line and does not account for the relative wave-speed of fluid fronts. More accurate predictions of fluid volumes and movement are, of course, possible if our simple setup is extended to include proper fluid models that provide formation-volume factors, pressures, and mobilities.

2.5 Heterogeneity measures and flow proxies

Based on the flow-diagnostic values introduced above, one can derive many other quantities that can be used to assess how heterogeneity affects the volumetric sweep efficiency of a reservoir model. Shook and Mitchell [36] suggested to use time-of-flight to extend the derivation of classical measures of heterogeneity to three-dimensional models. That is, they proposed to use the following three quantities as efficient flow proxies to rank earth models:

- Flow-capacity/storage-capacity F - Φ diagrams. Making an analogue to 1D displacement theory, the F - Φ curve is the equivalent to a plot of the fractional flow versus saturation. For a homogeneous displacement, this curve will be a straight line $F = \Phi$. The storage capacity Φ is defined as the cumulative pore volume as function of the total travel time $\hat{\tau} = \tau_f + \tau_b$, i.e., $\Phi(\tau) = \int_0^\tau \phi(\vec{x}(\hat{\tau})) d\hat{\tau}$. The flow capacity is defined as the cumulative flux for increasing travel time. For an incompressible flow, the pore volume equals the product of the flux and the total travel volume, and hence $F(\tau) = \int_0^\tau \phi(\vec{x}(\hat{\tau}))/\hat{\tau} d\hat{\tau}$. In practice, the values are normalized so that both F and Φ are within the unit interval.
- Sweep efficiency versus dimensionless time in pore volumes using the F - Φ curve as flux function. Dimensionless time is defined as $t_D = d\Phi/dF$ and the sweep efficiency is defined as $E_v = \Phi + (1 - F(\Phi))t_D$.
- The Lorenz coefficient is a popular measure of heterogeneity, and is defined as the difference in flow capacity from that of an ideal piston-like displacement:

$$L_c = 2 \int_0^1 (F(\Phi) - \Phi) d\Phi, \quad (3)$$

In other words, the Lorenz coefficient is equal to twice the area under the $F(\Phi)$ curve and above the line $F = \Phi$, and has values between zero for homogeneous displacement and unity for infinitely heterogeneous displacement.

We refer to [36] for a more in-depth investigation of the Lorenz coefficient in the context of streamlines and to [35] for the finite-volume interpretation. Although not discussed herein, one could also imagine using other measures like the Dykstra-Parsons coefficient or the Koval factor, see [20, §6.3].

To illustrate how the Lorenz coefficient can be used as a flow proxy for computing recovery factors, we will use two-dimensional, horizontal layers of the 3D SPE10 model [8]. This model is highly heterogeneous and will (for unfavorable displacements) give flow patterns with strong fingering effects. For each of the 85 layers

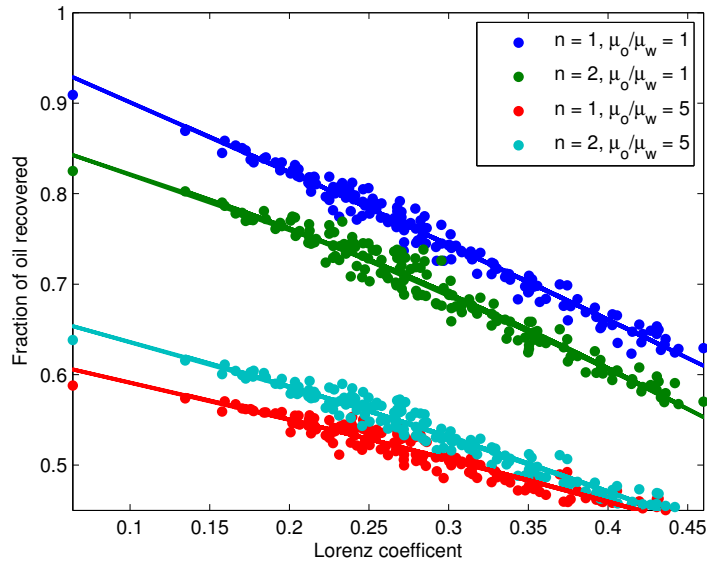


Figure 5: Correlation between the Lorenz coefficient and oil recovery after 0.75 PVI for five-spot patterns on horizontal layers of the SPE10 models using linear or quadratic relative permeabilities with unit or non-unit viscosity ratios.

of the model, we perform four two-phase simulations with linear or quadratic relative permeabilities and viscosity ratios equal one or five. Figure 5 shows that there is a strong correlation between the Lorenz coefficient computed from a single one-phase solution and the oil recovery after 0.75 PVI of a full two-phase simulation. This indicates that the Lorenz coefficient may be an effective flow proxy for (simple) water-flooding scenarios. In the next section, we will investigate in more detail how the Lorenz coefficient can be used in waterflood optimization.

3 Waterflood Optimization

Mathematical methods for waterflood optimization typically require a large number of forward simulations to converge to an optimal well configuration, and since full flow simulations generally are costly, rigorous optimization runs tend to be prohibitively expensive. In many cases, the purpose of a flow simulation is not to resolve reservoir dynamics in detail, but rather perturb the optimization process in the correct direction in parameter space. For this purpose, a complex multiphase flow simulation may not be needed. In this section, we will instead demonstrate how flow diagnostics can be used to develop a computational framework for waterflood optimization that has a very low computational cost compared with methods that rely on standard multiphase flow simulations. The purpose is not to develop an automated method that rigorously converges to the true optimum. Instead we seek to develop a set of simple lightweight tools that can be applied in interactive workflows, to iteratively improve preexisting configurations, or to rapidly search through large portions of parameter space and possibly serve as screening tools for methods based on more comprehensive simulations.

3.1 Reservoir model

As forward model in our optimization framework, we will use a reservoir model describing the flow field and the time-of-flight. To prepare for the derivation of a fully implicit discretization, all the model equations are written in residual form as follows

$$\mathcal{P}(q, \vec{v}) = \nabla \cdot \vec{v} - q = 0, \quad \mathcal{V}(p, \vec{v}) = \vec{v} + \mathbf{K} \lambda \nabla p = 0, \quad \mathcal{T}(\tau, \vec{v}) = \vec{v} \cdot \nabla \tau - \phi = 0. \quad (4)$$

Here, the primary unknowns are the pressure p , the time-of-flight τ , and the Darcy velocity \vec{v} . Notice that a full flow-diagnostic setup may contain both forward and backward time-of-flight equations, as well as any number of tracer equations depending on the problem. Without loss of generality, we will herein take the forward time-of-flight equation as representative and not examine tracer computation, but simply note that its treatment is analogous to time-of-flight.

To drive the flow in the reservoir, we will assume that there are n_{bh} wells controlled by bottom-hole pressure and n_r wells controlled by rate, and that these $n_w = n_{bh} + n_r$ wells altogether have n_{pf} well perforations. All wells will be modeled using a standard Peaceman well model per perforated cell and in addition to the reservoir unknowns, there will be unknowns corresponding to the well-perforation fluxes q_{pf} as well as the bottom-hole pressures p_{bh} . To define the well models, we let $N_w(j)$ denote the index of the well that perforation number j belongs to, $N_{pf}(k)$ be the indices of the perforations belonging to well k , and let W_{pf}^j denote the Peaceman well index for perforation j . Then, for each perforated cell, we have

$$\mathcal{Q}^j(q_{pf}, p_{bh}, p) = q_{pf}^j - W_{pf}^j \lambda(p_{bh}^{N_w(j)} - p) = 0, \quad j = 1, \dots, n_{pf}. \quad (5)$$

In addition, we need to specify a set of controls in the form of closure equations for each pressure-controlled well,

$$\mathcal{C}_{bh} = u_{bh}^k - p_{bh}^k = 0, \quad k = 1, \dots, n_{bh} \quad (6)$$

and each rate-controlled well,

$$\mathcal{C}_r = u_r^k - \sum_{j \in N_{pf}(k)} q_{pf}^j = 0, \quad k = n_{bh} + 1, \dots, n_w. \quad (7)$$

A simulation model is obtained by discretizing (4) using e.g., a standard two-point flux-approximation method, so that the continuous pressure p and time-of-flight τ fields are replaced by vectors \mathbf{p} and $\boldsymbol{\tau}$ of values defined at the cell centers, and \vec{v} is replaced by a vector of fluxes \mathbf{v} across the cell faces. This means that $\mathcal{P}(q, \vec{v}) = 0$ is replaced by $\mathcal{P}_h(\mathbf{q}, \mathbf{v}) = 0$, where \mathcal{P}_h denotes the *discretized* system, and so on. Henceforth, we will only work with the discrete flow equations, and for simplicity we therefore drop the h subscript. Moreover, to ease the implementation of adjoints to be introduced below, we write the discrete model equations as an implicit system in which the linearized system for the Newton updates are given as

$$\begin{bmatrix} 0 & \partial_{\mathbf{v}} \mathcal{P} & \partial_{\mathbf{q}} \mathcal{P} & 0 & 0 \\ \partial_{\mathbf{p}} \mathcal{V} & \partial_{\mathbf{v}} \mathcal{V} & 0 & 0 & 0 \\ \partial_{\mathbf{p}} \mathcal{Q} & 0 & \partial_{\mathbf{q}} \mathcal{Q} & \partial_{\mathbf{p}_{bh}} \mathcal{Q} & 0 \\ 0 & 0 & \partial_{\mathbf{q}} \mathcal{C} & \partial_{\mathbf{p}_{bh}} \mathcal{C} & 0 \\ 0 & \partial_{\mathbf{v}} \mathcal{T} & 0 & 0 & \partial_{\boldsymbol{\tau}} \mathcal{T} \end{bmatrix} \begin{bmatrix} \delta \mathbf{p} \\ \delta \mathbf{v} \\ \delta \mathbf{q} \\ \delta \mathbf{p}_{bh} \\ \delta \boldsymbol{\tau} \end{bmatrix} = - \begin{bmatrix} \mathcal{P}(\mathbf{q}, \mathbf{v}) \\ \mathcal{V}(\mathbf{p}, \mathbf{v}) \\ \mathcal{Q}(\mathbf{p}, \mathbf{q}, \mathbf{p}_{bh}) \\ \mathcal{C}(\mathbf{q}, \mathbf{p}_{bh}) \\ \mathcal{T}(\mathbf{v}, \boldsymbol{\tau}) \end{bmatrix}. \quad (8)$$

We note that the above system is not formed explicitly. Since the pressure equation is independent of $\boldsymbol{\tau}$, the pressure equation and the time-of-flight equations are solved in sequel (see Section 3.3).

3.2 Objective functions

Our hypothesis is that optimal waterfloods should be as close to an ideal piston-like displacement as possible. The Lorenz coefficient measures how much the (single-phase) flow-capacity/storage-capacity diagram for a given well pattern deviates from a piston displacement and in the previous section we saw that the coefficient correlated very well with the recovery predicted by a full multiphase simulation. In the following we will therefore use the Lorenz coefficient to measure the optimality of a waterflood to demonstrate the concept of simplified waterflood optimization based on flow diagnostics. Using Lorenz coefficient as objective function means that the optimization will seek to improve the sweep by equalizing the total travel time in all cells. In simplified conceptual examples, the reservoir is typically assumed to be filled by oil initially, and optimizing the Lorenz coefficient in all cells will lead to improved oil recovery. During injection in a realistic reservoir, however, there can be significant amounts of water present in the reservoir, below the water contact or because of earlier injection operations. Optimizing the Lorenz coefficient will then not differentiate between recovery of oil and recovery of preexisting water. To overcome this issue, we will in some examples modify the Lorenz coefficient by using an oil-weighted pore volume (see also [17]),

$$L_{c,o} = 2 \int_0^1 (F(\Phi) - \Phi) S_o d\Phi, \quad (9)$$

This ensures that the improved sweep applies to any remaining regions of oil and not to regions that are already flooded. The weights of the pore volume may not be only based on oil concentration: It can easily be expanded to gas-oil systems with cost functions depending on the case. We also emphasize that the Lorenz coefficient is just one example of a simplified objective function that suitable for a time-of-flight type optimization framework.

Although the Lorenz coefficient correlates well with recovery, it does not necessarily correlate with economic measures of a field. In particular, this is true for mature fields with limited remaining life. Here, we present a flow-diagnostics *proxy* for the net-present-value (NPV) which we will utilize on a three-phase model in Section 4.5. The NPV is the discounted accumulated net cash-flow. A simplified version for three-phase flow (not taking into account installation costs, shut down, etc.) can be expressed as

$$\text{NPV}(T) = \int_{t=0}^T \sum_{c=o,w,g} (r_c q_c + r_{ci} q_{ci}) (1+d)^{-t} dt, \quad (10)$$

where T is the time-horizon length, q_c and q_{ci} are the (surface volume) field production and injection rates respectively of component c , and r_c and r_{ci} are the revenues/negative costs for production/injection of component c . Finally, d is the discount rate. From flow-diagnostics calculations, we get no direct access to dynamic component production rates, so we define an *NPV-proxy* as follows:

$$\text{NPV-proxy}(T) = \sum_{\ell=1}^N \text{NPV}_\ell(T) + \int_{t=0}^T \sum_{c=o,w,g} r_{ci} q_{ci} (1+d)^{-t} dt, \quad (11)$$

where NPV_ℓ is an estimate of the NPV-contribution from grid cell ℓ . In particular, we let

$$\text{NPV}_\ell = \begin{cases} \sum_{c=o,w,g} r_c V_{c,\ell}^s (1+d)^{-\tau_{b,\ell}} & \text{if } \tau_{b,\ell} < T, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

Here, $\tau_{b,\ell}$ is the computed backward TOF of grid-cell ℓ and $V_{c,\ell}^s$ are the surface component volumes corresponding to the fractional-flow-weighted pore-volumes of cell ℓ . For a black-oil model, the component volumes $V_{c,\ell}^s$ can be computed as (omitting the ℓ -subscript)

$$\begin{bmatrix} V_w^s \\ V_o^s \\ V_g^s \end{bmatrix} = \text{PV} \begin{bmatrix} b_w & 0 & 0 \\ 0 & b_o & b_g r_{og} \\ 0 & b_o r_{go} & b_g \end{bmatrix} \begin{bmatrix} f_w \\ f_o \\ f_g \end{bmatrix}. \quad (13)$$

In the above expression, PV is the pore-volume (of cell ℓ), b_β is the inverse volume formation factors, $r_{\alpha\beta}$ is the ratio of component α in phase β , and finally $f_\beta = \lambda_\beta/\lambda$ is the fractional flow of phase β .

3.3 Solution strategy

To produce a framework for general (waterflood) optimization based on time-of-flight and tracers, we need to solve (8) for the primary variables, but also compute the gradients of an objective function for a given set of controls. Because all equations are linear, no iterations are needed and we can assume an initial guess of zero pressure and fluxes. Our solution strategy is designed to minimize the computational workload by making it possible to expose sub-problems to different types of solvers. This means that (8) will be solved in substeps. First, a pressure equation is assembled and solved. Assuming that the input total mobility λ in (4) is a constant, this results in a linear system of equations that can be written

$$\begin{bmatrix} 0 & \partial_{\mathbf{v}}\mathcal{P} & \partial_{\mathbf{q}}\mathcal{P} & 0 \\ \partial_{\mathbf{p}}\mathcal{V} & \partial_{\mathbf{v}}\mathcal{V} & 0 & 0 \\ \partial_{\mathbf{p}}\mathcal{Q} & 0 & \partial_{\mathbf{q}}\mathcal{Q} & \partial_{\mathbf{p}_{bh}}\mathcal{Q} \\ 0 & 0 & \partial_{\mathbf{q}}\mathcal{C} & \partial_{\mathbf{p}_{bh}}\mathcal{C} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \\ \mathbf{p}_{bh} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (14)$$

Herein, we have utilized a two-point-flux approximation (TPFA) for the spatial discretization (see e.g., [6]). In particular, this means that (14) can be manipulated to an elliptic system for the pressure-unknowns only, and can be solved by an efficient algebraic multigrid solver; in MATLAB we use AGMG [30].

After the fluxes have been computed, the time-of-flight equations also become linear. With initial guess $\tau_0 = 0$ we have from (8) that $\tau = \delta\tau$, and it follows that the system simply becomes

$$\partial_{\tau}\mathcal{T}\tau = \phi. \quad (15)$$

If tracers are required, they have the same linear systems as (15) but with a different right-hand side, making it possible to solve for tracers simultaneously by using a direct linear solver that can invert for multiple right-hand sides with little extra cost or by using the topological ordering methods discussed in Section 2.3. Once all the forward equations have been solved, we can find the gradient of an objective function with regards to the time-of-flight and tracers and the sensitivity with respect to the well controls. To this end, one can either use numerical differentiation or an adjoint method. Numerical differentiation is non-intrusive and simple to implement, but may be very computationally costly since one forward equation needs to be solved for each perturbation. Adjoint methods are significantly more efficient, but require alteration of the model equations, as described next.

3.4 Adjoint formulation

To derive the adjoint equations, we introduce the stacked vector of solution quantities, $\mathbf{x}^T = [\mathbf{p}^T, \mathbf{v}^T, \tau^T, \mathbf{q}^T, \mathbf{p}_{bh}^T]$, whose values depend implicitly on the controls \mathbf{u}

that consist of target values for rate or bottom-hole pressure for each well as given in (6)–(7). The objective function will be denoted as $G(\mathbf{x}(\mathbf{u}))$ and should be regarded as a function of the solution quantities. Finally, we must define a set of constraints $\mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{u}) = 0$ that correspond to some representation of our reservoir model (4)–(7).

We set up the Lagrange function for our problem, which has the same values as our original objective function in the feasible set,

$$G_\lambda = G(\mathbf{x}(\mathbf{u}), \mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{u}). \quad (16)$$

We are interested in the derivatives of the objective function with regards to the control values, so we differentiate to get

$$\frac{dG_\lambda}{d\mathbf{u}} = \frac{\partial G}{\partial \mathbf{u}} + \left(\frac{\partial G}{\partial \mathbf{x}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right) \frac{d\mathbf{x}}{d\mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{g}}{\partial \mathbf{u}} + \mathbf{g}^T \frac{d\boldsymbol{\lambda}}{d\mathbf{u}}. \quad (17)$$

This equation can be greatly simplified if we eliminate all terms involving $d\mathbf{x}/d\mathbf{u}$ by allowing $\boldsymbol{\lambda}$ to be defined (and solved for) by the adjoint equations

$$\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)^T \boldsymbol{\lambda} = \mathbf{J}^T \boldsymbol{\lambda} = - \left(\frac{\partial G}{\partial \mathbf{x}} \right)^T. \quad (18)$$

For a feasible state \mathbf{x} , i.e., $\mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{u}) = 0$, and the gradient becomes

$$\frac{dG_\lambda}{d\mathbf{u}} = \frac{\partial G}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T \frac{\partial \mathbf{g}}{\partial \mathbf{u}}. \quad (19)$$

Furthermore, our objective functions will only depend on the state variables \mathbf{x} and not on the controls \mathbf{u} directly, and accordingly $\partial G/\partial \mathbf{u}$ also vanishes.

The result is a very simple implementation when all Jacobians are known or easily available. First, we solve the forward problem (8) using the block-wise approach outlined earlier to obtain state variables and upwind directions. Then, the backward problem (18) is solved using the same approach to obtain the sensitivities of the objective function.

3.5 Rate optimization

We will be using a simple steepest descent algorithm for finding optimal controls. In the numerical experiments we will mainly be considering rate optimization, i.e., the control variables \mathbf{u} will represent well rates. Well rates are typically constrained from above by a maximal rate, and below by zero (meaning that producers are not allowed to become injectors and vice versa). In addition, the total (field) injection rate might be fixed and/or a voidage replacement constraint needs to be imposed (volume in equals volume out). Fulfilling these (linear) constraints amounts to an update of the form

$$\mathbf{u} \leftarrow \mathbf{u} - P \left(\alpha \frac{dG_\lambda}{d\mathbf{u}} \right)^T, \quad (20)$$

where α is the step size and P is a projection onto the constraints. The goal is to find a stationary point, i.e., such that the projected gradient is zero. An iteration in the steepest-descent algorithm we use herein simply evaluates the objective for the updated controls, and in case of non-improvement, the step-size α is reduced until the objective value is improved. The algorithm stops when the projected gradient is sufficiently small, or the improvements in the objective function between two iterations is less than a prescribed tolerance. This approach was chosen mainly for its simplicity of implementation; there are many algorithms and solvers for continuous constrained optimization that may be more efficient and robust.

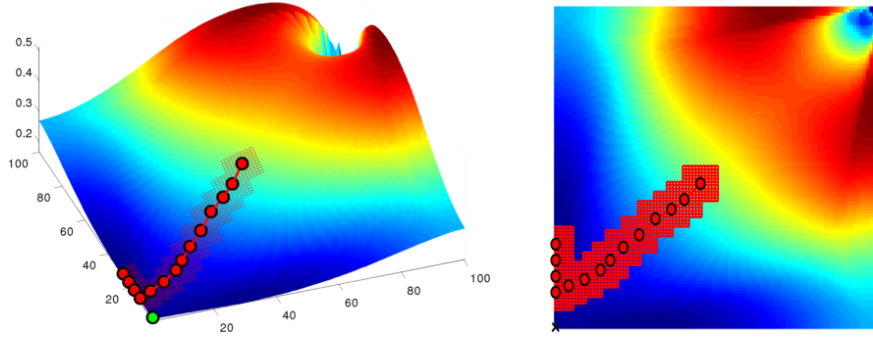


Figure 6: Optimal placement of a single in-fill well in a homogeneous quarter five-spot problem: starting near the middle of the domain, the well-placement region moves the new well toward the boundary where the Lorenz function attains its lowest value.

3.6 Well placement

Another possible application of time-of-flight based optimization is well placement. If several injectors are to be placed during initiation of secondary production, each possible injector configuration will have an objective value. While an exhaustive search of all possible options is possible for small cases, it is not feasible for larger cases. Instead, the well placement can be improved via an optimization process that starts with an initial well placement, which typically will incorporate some expert knowledge, but not necessarily be optimal:

1. Add pseudo-wells (see e.g., [38]) with zero rate in the region around each injector. This does not alter the solution, but the gradients of the pseudo-wells can easily be computed.
2. The pseudo-well with the largest gradient value then replaces the original well.
3. Repeat until all wells are stationary or a cycle is achieved.

Depending on the problem, the well rates can be optimized in each substep. If the rates are optimized in each substep, the number of function evaluations will increase, but the computational cost is still small compared to a fully fledged simulator run. In the examples, we only move one well at the time. This increases the cost slightly as more evaluations will be required for problems in which several wells should be moved, but it will also ensure that we do not miss optimal configurations in cases where some wells should not be moved.

Figure 6 illustrates the response surface obtained by using the Lorenz coefficient as objective function for placing an additional injector in a homogeneous quarter five-spot example on a 21×21 grid. After the new injector is added, the two injectors are set to operate at half the rate of the original injector. Notice that there are more than one optimal well position in this problem because of symmetry and because the Lorenz coefficient is not convex. The secondary minimum is relatively uninteresting, as it will place the second injector on top of the producer so that the three-well problem degenerates to the original two-well problem. (The Lorenz coefficient does not discriminate between cases with different overall field rate). For more complex cases, the distinction between local minimums may not be trivial and one generally needs to utilize a global algorithm to prevent trapping in local minimums and be careful how the objective function is formulated to avoid introducing artifacts like shown here.

4 Numerical examples

In this section, we will present several examples of waterflood optimization. All numerical results are obtained with solvers that are implemented in the open-source MATLAB Reservoir Simulation Toolbox [25, 21], and the scripts and data required to reproduce most of the examples will be published on the MRST webpage.

In the examples, we use a simple steepest descent method to minimize the objective function. The Lorenz coefficient is in our experience not necessarily convex and will generally give multiple local minimums. To automatically find the *true global minimum*, the optimization algorithm must either be augmented with a global algorithm that efficiently subdivides parameter space into sets of local search regions, or be augmented by some kind of meta-heuristic to enable the descent algorithm to escape from local minimums. In practice, however, we foresee that the diagnostic tool will be used to optimize an existing or planned configuration, it will be sufficient to suggest local changes that are feasible and increase recovery.

4.1 Verification of optimization algorithms

We start by verifying the optimization algorithms on two simple problems that have analytical solutions. To verify the rate-optimization method, we consider a grid with 101 cells having a uniform permeability and a porosity that is equal 0.5 in cells 1 to 50, equal 0.75 in cell 51, and equal 1.0 in cells 52 to 101. A bottom-hole controlled producer is placed in cell 51, and rate controlled injectors with equal rates are placed at the left and right endpoint of the domain, see Figure 7. The optimal displacement is trivial to compute: because the second injector sweeps twice the pore volume it should have twice the rate to give synchronized arrival times of the water fronts.

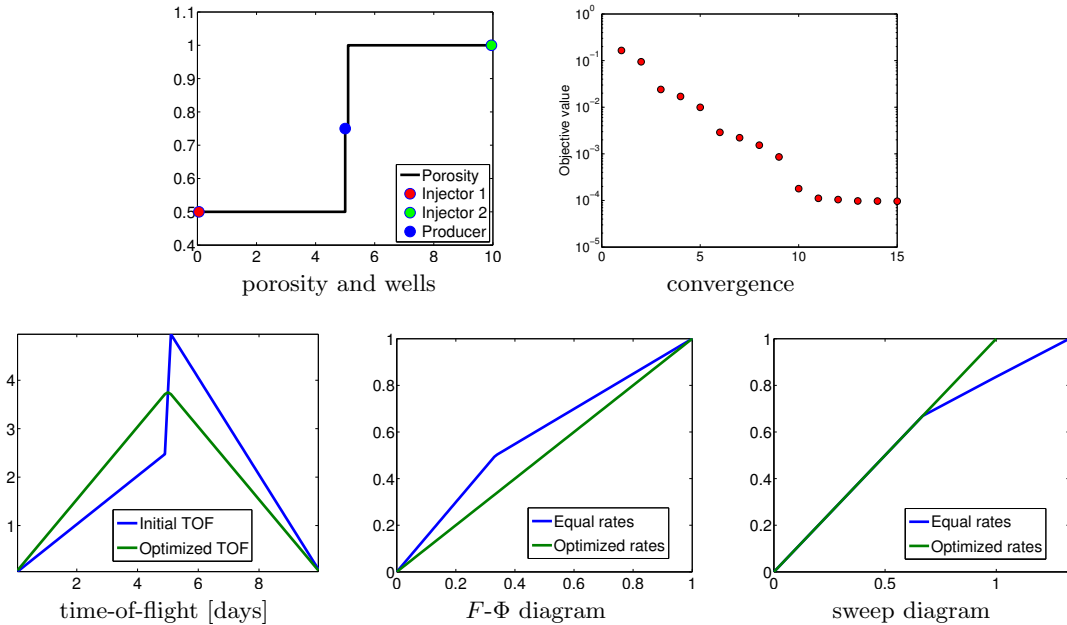


Figure 7: Verification of rate optimization for a simple 1D displacement.

Figure 7 shows time-of-flight and $F-\Phi$ and sweep diagrams for the initial setup. The latter two have a kink where the front from Injector 1 reaches the producer and stops contributing to further increases in time-of-flight. This means that after 0.75 pore volumes of water have been injected, water injected from Injector 1 will no longer contribute to oil production and hence the sweep decays after $\Phi = 0.75$

PVI. After optimization, the ratio between the two injection rates is as expected, the Lorenz coefficient is zero to the given tolerance, and both the F - Φ and the sweep diagrams are straight lines.

To verify the well-placement algorithm, we consider a setup with four injectors and one fixed producer to be placed in a $600 \times 600 \text{ m}^2$, homogeneous domain. The optimal solution is the classical five-spot pattern with injectors in the corners and the producer in the middle. (Notice that the notions of producer and injector are not important for the time-of-flight problem as it only corresponds to a change in sign for the fluxes.) The initial well configuration has a producer along with four injectors placed in the center of the reservoir with equal rates. The setup has a Lorenz coefficient of 0.915 and is an intentionally unsuitable configuration for maximizing oil recovery. The injector positions are then dynamically adjusted by the algorithm from Section 3.6 with rates kept constant. The left plot in Figure 8 shows how the wells correctly move towards the optimal five-spot configuration, at which the F - Φ diagram is almost a straight curve and the Lorenz coefficient has reduced to 0.132. The figure also shows that optimizing the Lorenz coefficient effectively corresponds to equilibrating travel time (sum of forward and backward time-of-flight). When the four injectors are placed in random positions, the problem may sometimes get caught in a local (degenerate) optimum as shown in Figure 6 if one of the injectors is placed too close to the producer. Likewise, the wells may not reach the exact corners if the pseudo-well search radius is too big.

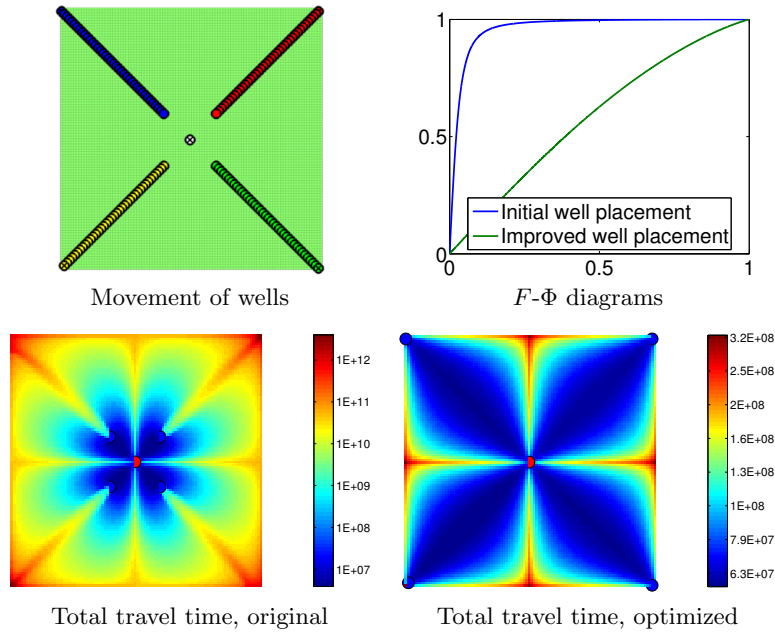


Figure 8: Verification of well placement for a square, homogeneous reservoir.

4.2 Optimizing five-spot patterns for layers of SPE10

In onshore fields, the configuration of injection and production wells are often set according to well-established patterns such as line-drive, five-spot, seven-spot, or nine-spot patterns. These patterns will in most cases not be optimal for a strongly heterogeneous reservoir. In this example, we will therefore apply our time-of-flight algorithms to optimize the original five-spot pattern for layers of the SPE10 benchmark [8]. That is, we will optimize the well placement, and for each placement, optimize to ensure that the rates are optimal under the constraints of a total rate

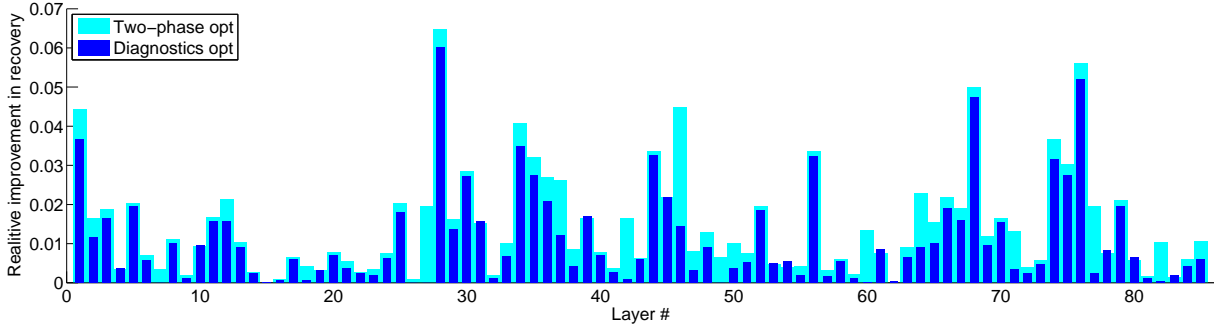


Figure 9: Relative improvement in recovery for initial well-placements using diagnostics-based optimization and two-phase-based optimization.

and a minimum injection per well set to 1/16 of the field injection rate.

In this simple setup, the optimization algorithm does not account for multiphase effects. However, once the optimal results have been found, see Figure 10, multiphase simulations are run to validate the proposed well configurations. The first fluid model has quadratic relative permeabilities and unit viscosity and density ratios. In the second fluid model, the oil has a viscosity that is ten times higher than the injected water, so that the latter will tend to form long viscous fingers and create an unfavorable displacement that will lead to a significantly lower recovery for the same amount of injected water. Densities are set to 1000 and 700 kg/m³, respectively. To investigate the influence of multiphase effects for the optimization algorithm, we compared the optimal rates obtained for initial well-placements, with those obtained using a much more time-consuming optimization based on two-phase forward and adjoint simulations. Using an oil-to-water viscosity ration of ten, we performed diagnostics-based and two-phase-based optimization of all 85 layers of the SPE10 benchmark, see Figure 9. As expected, the optimization based on simulations results overall in higher recovery than that obtained with the diagnostics optimization. However, for the majority of the cases, the obtained recovery for the two optimization techniques are fairly similar. We note that for this case, the optimization using two-phase simulation required the computation time of roughly 100 diagnostics-based optimizations.

Figure 11 shows initial and optimized waterfloods for the bottom layer of the Tarbert formation, which is a marginal marine deposit in which the horizontal permeability follows a log-normal distribution. In the initial setup, water injected at the SW injector will flow towards the sweep area of the SE injector and together lead to early breakthrough. This will leave the large low-permeable area south-west of the producer unswept. Likewise, there is a large unswept area north of the producer. The optimization manages to equilibrate the total travel times in the model by moving the SW injector into the low-permeable region south-west of the producer and moving the NE and SE injectors west to increase the overall sweep. This decreases the Lorenz coefficient from 0.233 to 0.073, which is lower than for the corresponding homogeneous model and corresponds to an almost ideal piston-like displacement. The overall recovery after 1 PVI is 81.5% for fluids with equal properties and 46.0% for fluids with an unfavorable viscosity ratio of 1:10. This is the same recovery as for a homogeneous model, and represents and 12% and 7% improvement, respectively, compared to the initial well configuration.

Figure 12 shows initial and optimized waterfloods for Layer 66, which is in the middle of Upper Ness. This fluvial formation is thought to represent a delta-plain or coastal-plain deposition and consists of high-permeable channels on a low-permeable

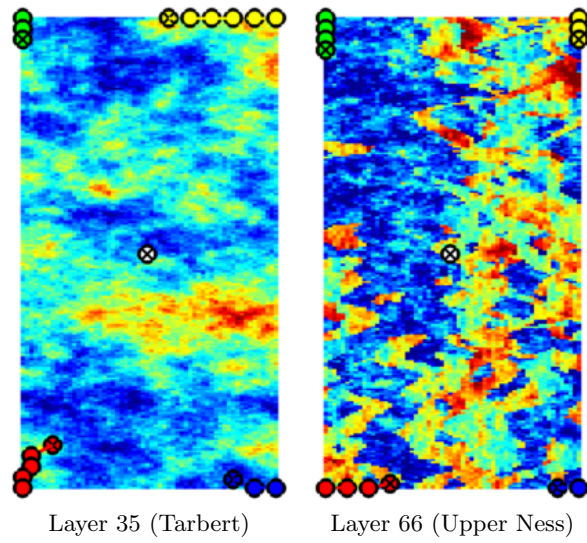


Figure 10: Permeabilities and movement of wells for two layers of the SPE10 data set.

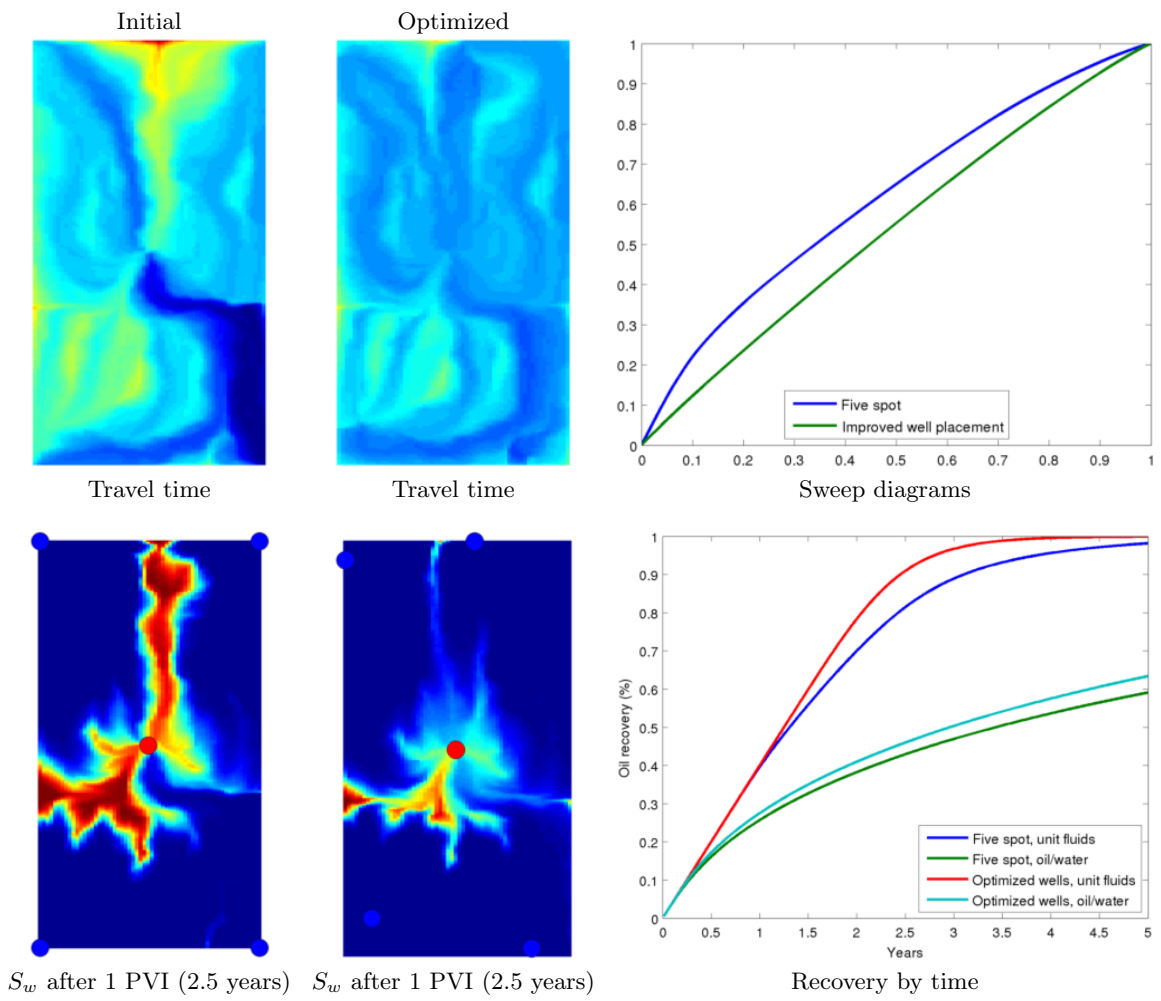


Figure 11: Optimization of well placement and well rates for Layer 35 with a two-phase fluid model with equal viscosities.

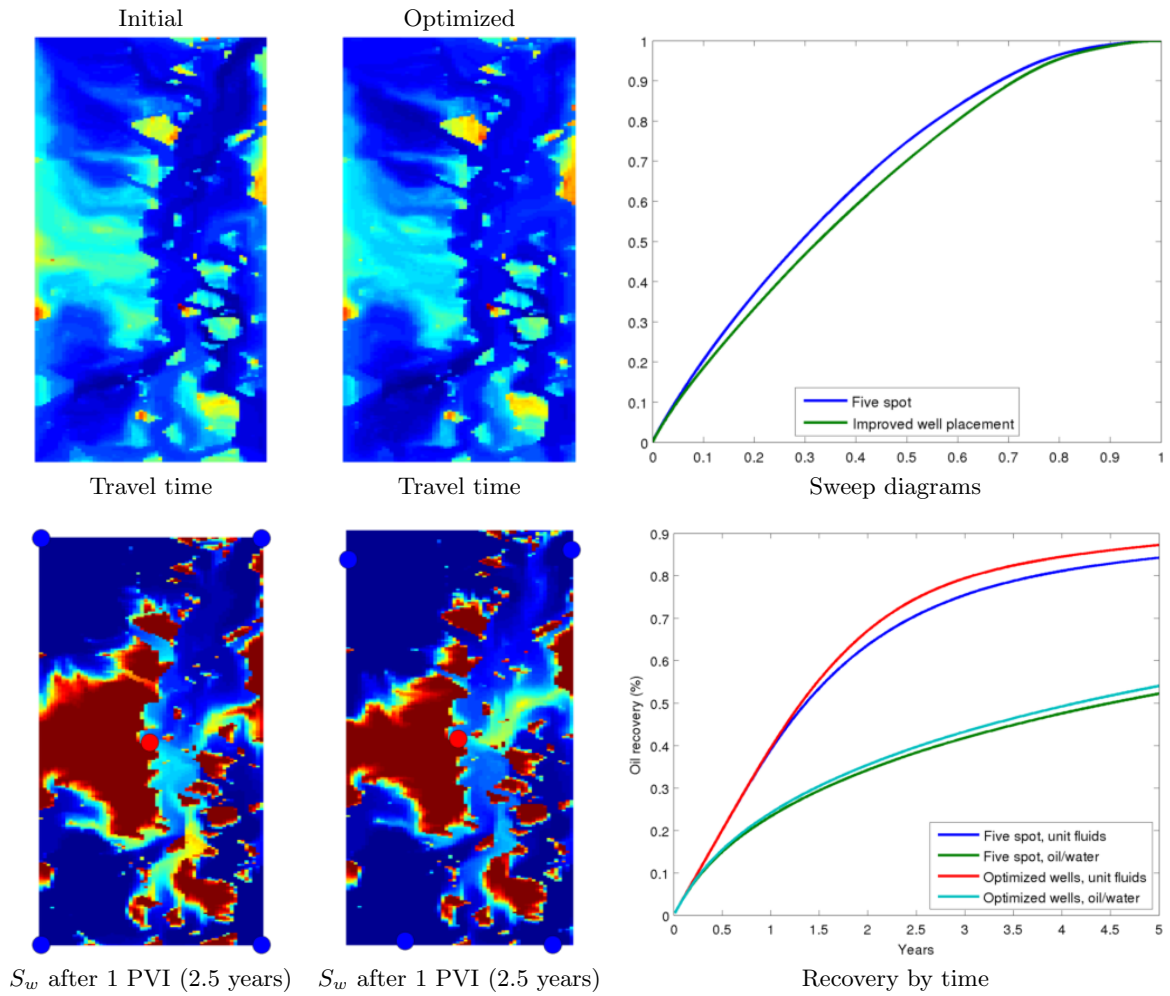


Figure 12: Optimization of well placement and well rates for Layer 66 with a two-phase fluid model with equal viscosities.

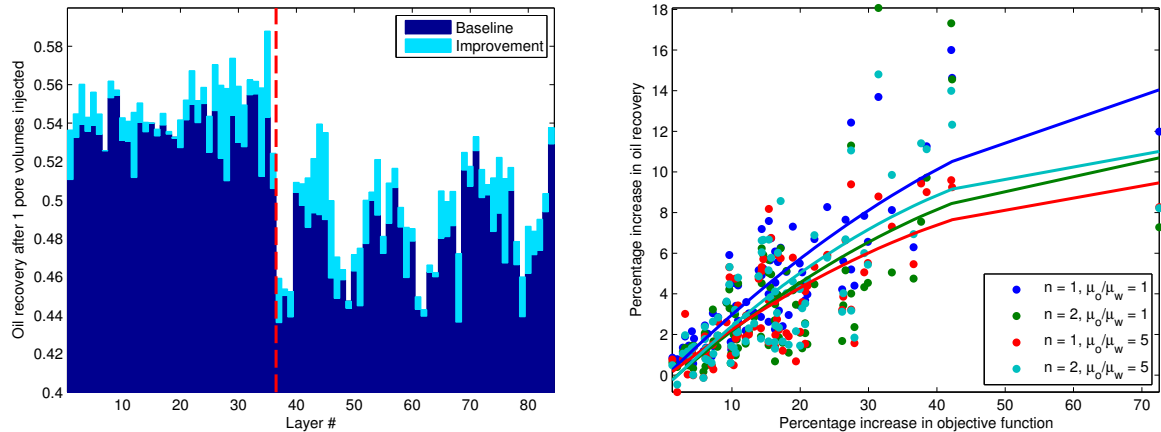


Figure 13: Validation of waterflood optimization based on the Lorenz coefficient for all 85 layers of the SPE10 model using four different fluid models. The left plot shows recovery factor before and after optimization for a viscosity ratio of five. The right plot shows the increase in oil recovery as a function of decrease in Lorenz coefficient for different Corey exponents and viscosity ratios.

background of mudstone and coals. The optimization reduces the Lorenz coefficient from 0.339 to 0.285, mainly by moving the NW producer to the south. This reduces the large unswept, low-permeable region west of the producer slightly, which increases the eventual recovery somewhat but does not mitigate early water breakthrough. As a result, we only observe a 3.4% increase in recovery after 1 PVI for both fluid models. First of all, the presence of high-permeable channels makes it difficult to increase the sweep in the low-permeable regions of the model. Secondly, the Lorenz objective function contains a large number of local optima in which the optimization can easily get stuck. Third, pseudo-well do not necessarily give a good approximation. When defining a search region for a well, it is important to strike a balance between searching a large enough area to avoid local minima and ensuring that the gradients are representative of a movement of the well. Here, the intertwined pattern of high-permeable sands may introduce several local minimums inside almost any search region. For this reason, one should therefore consider using multiple start points.

Layers 35 and 66 were chosen somewhat arbitrarily to illustrate the optimization process. To get a more systematic validation, the same optimization process was applied to all the 85 layers of the model, and then the optimized well positions were validated against multiphase simulations with four different fluid setups: linear or quadratic relative permeabilities with $\mu_o : \mu_w$ equal one or five. The results can be seen in Figure 13.

4.3 Optimizing placement and rates: realistic shallow-marine reservoir

In this example, we consider a realistic and representative model of a progradational shallow-marine reservoir generated by the multidisciplinary SAIGUP modeling project [24], which focused on assessing the influence of geological factors on production in a large suite of synthetic models. The model has anisotropic permeability and contains both channel-like structures as well as layered structures, giving a faulted geological model with 96 778 cells. The initial well configuration is set up for a waterflood, with six injectors in the corners and the middle of the grid, with two producers in between. Figure 14 shows the permeability and the initial well

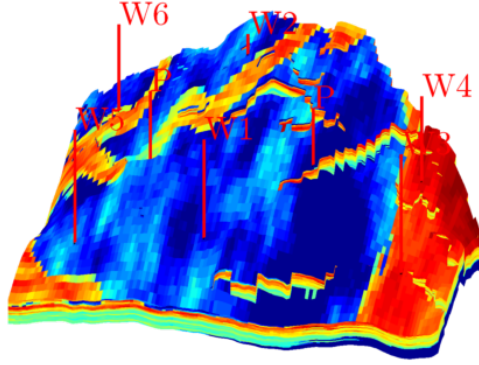


Figure 14: Horizontal permeability and initial well configuration for the SAIGUP model.

configuration of the model.

While the Lorenz coefficient correlates well with oil recovery for a full simulation run, optimizing the well rates at the initial time step will likely not be an optimal strategy at a later time when mobilities and dynamics in the reservoir has been greatly altered by the waterflood. For this reason, we will examine three different strategies for producing the reservoir:

- Default well placement, equal rates
- Optimized well placement, rates optimized at $t = 0$ and kept fixed
- Optimized well placement, rates optimized every fourth month

The third strategy will be noted as just-in-time (JIT) optimization, as it is performed at regular intervals during the full simulation run based on the current fluid saturation profile. By optimizing during the injection phase, the Lorenz coefficient can be used to preview the current injection efficiency: If the coefficient changes drastically between evaluations, the previously optimized rates are likely to no longer be optimal and a re-optimization may be useful. We will assume that each injector has three different uniformly distributed segments that can be controlled separately with a minimum outflow limit equal to $1/5$ of the initial injection rate for that segment to avoid cross flow.

Table 1: Recovery after injection of one pore volume of water using three different strategies to produce the shallow-marine SAIGUP reservoir model.

Viscosity ratio	Initial	Improved	JIT
10	30.1%	39.9%	40.2%
25	24.0%	32.4%	32.7%
50	20.1%	27.3%	27.6%

The total water injected is one pore volume over ten years, distributed over all six injectors. Gravity is present in the simulation, but is not used for the time-of-flight solver. Relative permeability curves are standard quadratic functions of the saturations. The recovery over time can be seen in Figure 15 and Table 1. The optimized well placement gives greatly improved oil recovery compared to the suboptimal initial placement. Although the initial placement pattern is obviously not optimal, the improved well placements shown in Figure 15 seem reasonable: The wells are placed near the boundary and near highly permeable areas. However, the final recovery shows only small differences between the initial optimization and the JIT-recovery. Any changes in the well rates during the simulation are limited to the

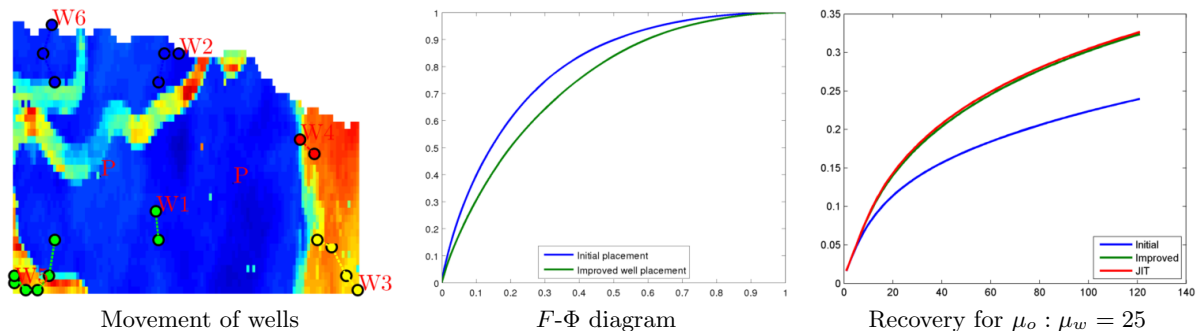


Figure 15: Waterflood optimization for a shallow-marine reservoir (SAIGUP).

initial time steps, indicating that the major changes in the reservoir happen when the initial front starts to propagate.

4.4 Optimizing the drilling sequence

In real operations, all wells are not drilled simultaneously as in the previous example, but may be introduced gradually over the life-cycle of a reservoir to increase oil recovery underway. The placement of additional wells is a more complex problem that must account for changing dynamics of the waterflood at different points in time.

To demonstrate the capability of our flow-diagnostics framework in a drilling workflow, we will consider a simple problem: A fixed number of wells are to be drilled, with a single injector/producer pair introduced at time zero. A fixed amount of water is to be injected per well and new wells will be introduced at specific timesteps during the simulation. Different approaches targeted at getting better life-cycle recovery will be demonstrated, along with a discussion of advantages and disadvantages.

For the well placement, we are considering six injectors and two producers. The baseline case has injectors confined to the boundary of the domain and producers in the middle as shown in Figure 16. The two main choices concern which wells are introduced at which timestep, as well as their placement. Here, we use a computationally inexpensive strategy: First we compute the Lorenz coefficient of all injector-producer pairs and chose the pair with the lowest coefficient to be drilled at time zero. Then, the remaining wells (five injectors and one producer) are added using a greedy strategy that picks the well that will give the lowest Lorenz coefficient when drilling a single additional well with the previously placed wells present.

We apply four different strategies to the drilling problem:

1. Initial ranking and placement, which is the baseline case based on intuition.
2. Same placement as 1), but with the drilling schedule based on L_c ranking at $t=0$.
3. Drilling schedule as in 2), and the placement of new wells optimized.
4. Placement of all wells optimized at $t=0$ with all future wells present, and drilled by ranking.

Each strategy will be evaluated both with and without rate optimization during the simulation with regards to $L_{c,o}$ for both a subset of SPE10 and the shallow-marine SAIGUP reservoir. The schedule in both cases consists of a ten year period, with wells being drilled at the beginning of each year, starting with a producer and injector at $t=0$ and the remaining six wells one at a time during the next six years. Each well is initially assigned a rate corresponding to $1/4$ pore volumes over ten years.

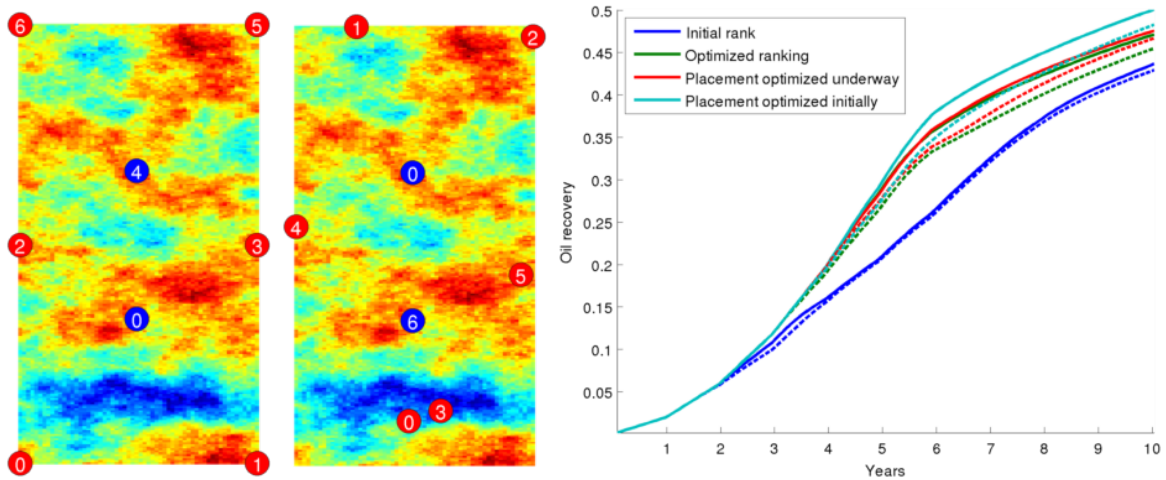


Figure 16: Optimized production strategy for a horizontal layer of the SPE10 benchmark. The initial well placement and drilling sequence (left) can be improved both by ranking the wells and optimizing their placement (middle), resulting in significantly increased oil recovery (right). Solid lines contain rate optimization every period while stapled lines have constant injection rate per well.

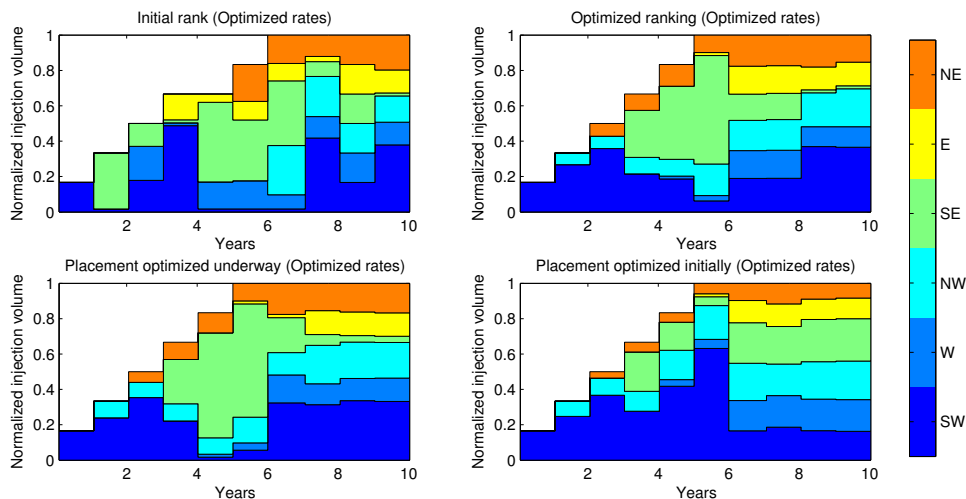


Figure 17: Well controls for the optimized production strategies presented in Figure 16. Well names refer to the positions in the initial configuration.

2D reservoir, SPE10. For the simple 2D Tarbert subset, it can be seen in Figure 16 that the best strategy is to optimize all well placements at time zero. As there is a significant production period after all wells are drilled, exploiting the large picture is unsurprisingly quite advantageous. Dynamical placement of injectors during the simulation comes in at second place, but there is not any significant difference between the dynamical placement and the optimal ranking case, indicating that the order of wells is more important than their local placement – it is only when a global strategy is employed that the placement optimization is effective. When observing the well rates in Figure 17, the global optimization (bottom-left) seems to have a more even distribution of rates as wells are phased in compared to the other strategies, for which one well tends to dominate the time interval, and rates are changed drastically when new wells are drilled. This may be because the well optimization assumes equal rates, giving an optimal solution for constant injection per well.

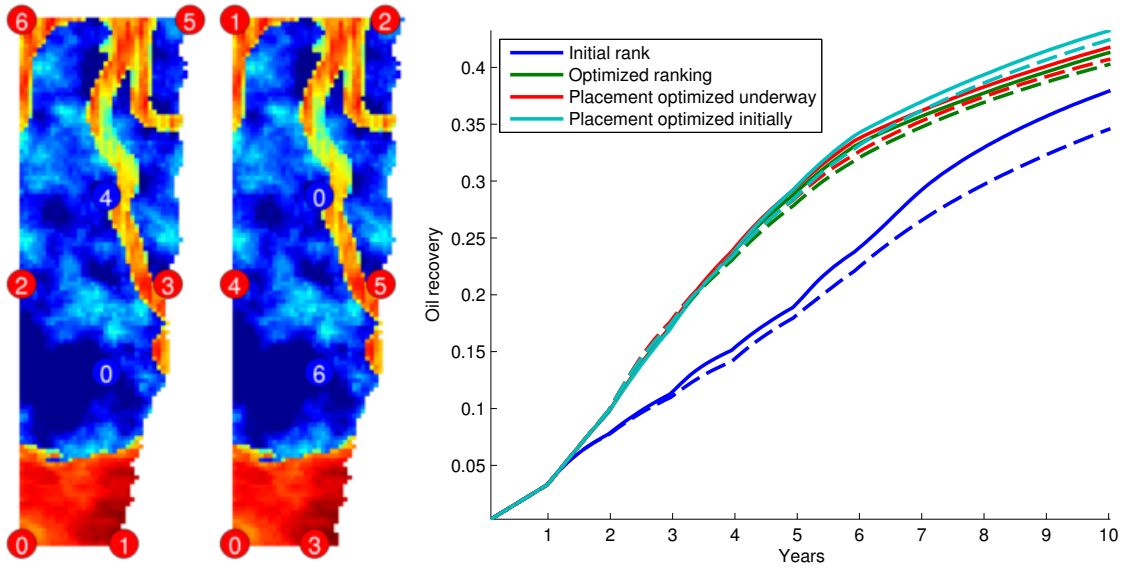


Figure 18: Optimized well placement and drilling sequence for the SAIGUP shallow-marine reservoir model.

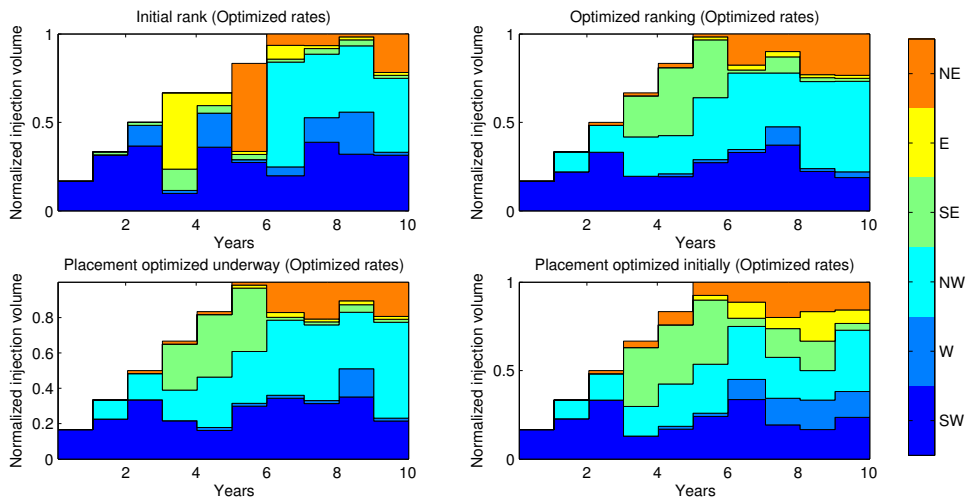


Figure 19: Well controls for the optimized production strategies presented in Figure 18.

Shallow-marine reservoir (SAIGUP) The same strategy is applied to the reservoir described in Section 4.3 to demonstrate the methodology on a grid containing complex geological features. Again, we see that optimizing all well placements with respect to long-term production comes out as the best strategy, and that the other strategies perform similarly to the results for SPE10. It is interesting to note from Figure 19 that most of the schedules have very similar features when optimized. The most obvious example is the south-west injector, which is always injecting at a rate that is higher than the average rate, which should not be surprising considering that this injector is associated with the largest flooding volume at time zero. A larger flooding volume should correspond to a larger rate if the recovery is to approximate a piston-like displacement.

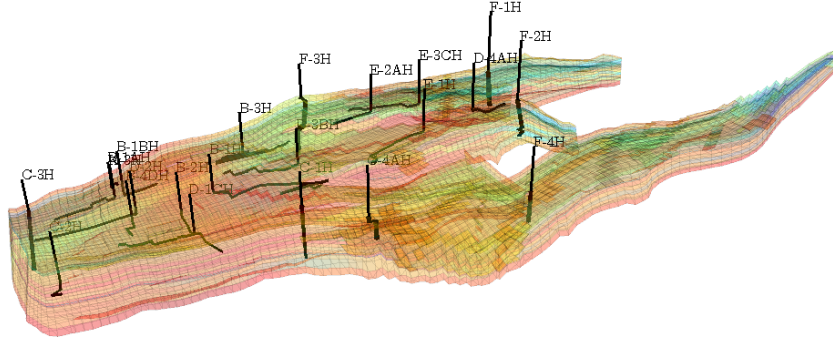


Figure 20: The Norne model with wells.

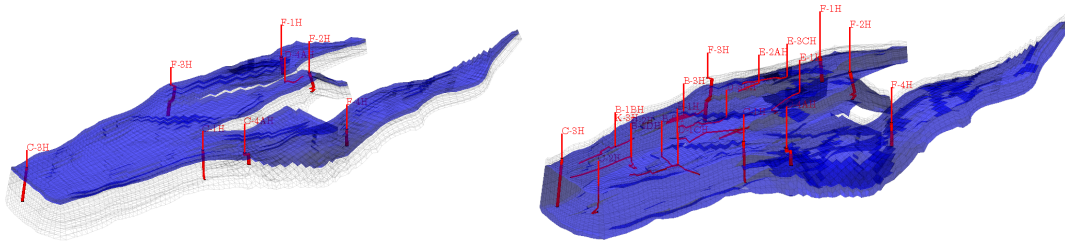


Figure 21: Upper (left) and lower (right) part of the Norne model with perforating wells.

4.5 Optimized strategy for production of the Norne field

In this example, we illustrate the use of flow diagnostics in conjunction with a commercial simulator. The key idea is that we optimize using the flow diagnostics, imposing constraints approximately, and then feed the optimal controls as *targets* to the commercial simulator. In this way, the simulator will follow the suggestions provided by the diagnostics optimization when feasible but switch control if constraints are violated. We consider a model of the Norne field (Figure 20) as provided by *The Second Norne Comparative Case Study* [29]. The case study included a history-matching part plus finding an optimal strategy for the period 2006 to 2016. Here, we consider only the optimal strategy using conventional water/gas injection, and take the provided model states at December 1 2006 as the starting point. Since Norne is a mature field, the potential for oil production is limited. The model of the field consists of two parts that are only in communication through wells as shown in Figure 21. Since the upper part is mainly filled with gas, we only consider the lower part in our diagnostics optimization. However, when we evaluate the optimal settings using the simulator, we utilize the complete model.

We include all wells that had been active during 2006. This amounts to twelve producers and eight injectors. In the comparative case study, the following operational limits were given:

- For each injector: bottom-hole pressure should not exceed 450 bar and water rate should not exceed 12000 Sm³/day.
- For each producer: bottom-hole pressure should be above 150 bar, liquid rate (oil+water) should not exceed 6000 Sm³/day and the well should be shut if water-cut exceeds 0.95.

Under these constraints, NPV (as defined in (10)) should be optimized using an oil price of 75 USD/bbl, water injection/production cost of 6 USD/bbl and a gas injection cost of 1.2 USD/Mscf. The yearly discount rate is 10%.

Since the flow-diagnostics computations are in essence incompressible single-phase simulations, we need to *translate* the limits and constraints from the multiphase model. Limits on bottom-hole pressure are adjusted based on the depth of the wells, and the rate-constraints are converted to reservoir conditions using the initial state of the reservoir. The well controls in the black-oil simulator operate according to reservoir volume rates. Because total injected and produced volumes are constrained to be equal, the reservoir pressure is approximately maintained and average pressure only varies twenty bar during simulation. Moreover, since the bottom-hole pressures in the producers are constrained from below (150 bar), the gas coming out of solution in the reservoir is overall small.

In this example, we use numerical perturbation instead of adjoints for computing gradients. The derivatives were computed using a relative magnitude of 10^{-3} . Numerical perturbation makes the switching from rate-control to pressure-control and vice versa more straight forward, and does not require the use of advanced optimization tools. In general, however, we would recommend using an adjoint formulation, possibly in combination with the generalized reduced gradient method as suggested in [18]. This method allows for control switching between rate and bottom-hole pressure depending on which constraints become active. To prevent switching back and forth repeatedly, an option is to run multiple adjoint simulations to obtain gradients with respect to all potential control types, e.g., obtaining gradients with respect to rates and bottom-hole pressure requires two adjoint simulations. However, even though the numerical perturbation technique is not optimal, the entire optimization for the cases discussed in the following required less time than a single black-oil simulation.

In our first attempt, we let all injectors inject water. Our first base-case operational setting is full-blown injection and production within the limits provided, with shut down of producers that exceeded a water-cut of 0.95. Figure 22 shows that this strategy is profitable for approximately two years, and the maximum NPV is $1.25 \cdot 10^9$ USD. If we assume that there is a balance between injected and produced reservoir volumes, the maximal profitable water cut turns out to be approximately 0.85. We use this water cut as shut-in limit for our second base strategy, which is profitable for approximately 3.6 years, having a peak NPV of $1.53 \cdot 10^9$ USD. In our optimized approach, we use flow-diagnostics calculations with the NPV-proxy (11) as objective function. The resulting strategy suggests to shut down five of the producers, thus resulting in seven active producers and eight injectors. To evaluate the strategy, we feed the optimal rates into the commercial simulator as reservoir-volume rate targets. As seen in Figure 22, the optimized strategy is profitable for approximately 3.6 years with a maximal NPV of $1.79 \cdot 10^9$ USD, which is a large improvement over the base cases. In Figure 22 we have also plotted the field-production rates of oil, water, and gas for the three approaches. Here, the produced free gas comes mainly from previous gas injection. Note also that the base cases are slightly more aggressive than the optimal approach with respect to oil production and this leads to higher NPVs for time-horizons up to a year.

In our second approach, we let wells C-1H, C-3H and C-4AH inject gas rather than water (these are gas injectors prior to our starting point). Again, our base case is full-blown injection and production within limits, and shut down of wells with water cut exceeding 0.95. As seen in Figure 23, the base case is profitable for almost two years and reaches an NPV of $1.42 \cdot 10^9$ USD. The optimized case, however, is profitable for approximately 3.5 years with a maximal NPV of $1.72 \cdot 10^9$ USD. Again, we observe that the base case is more aggressive with respect to oil production, but in this case this does not give higher short-term NPVs because of high gas-injection

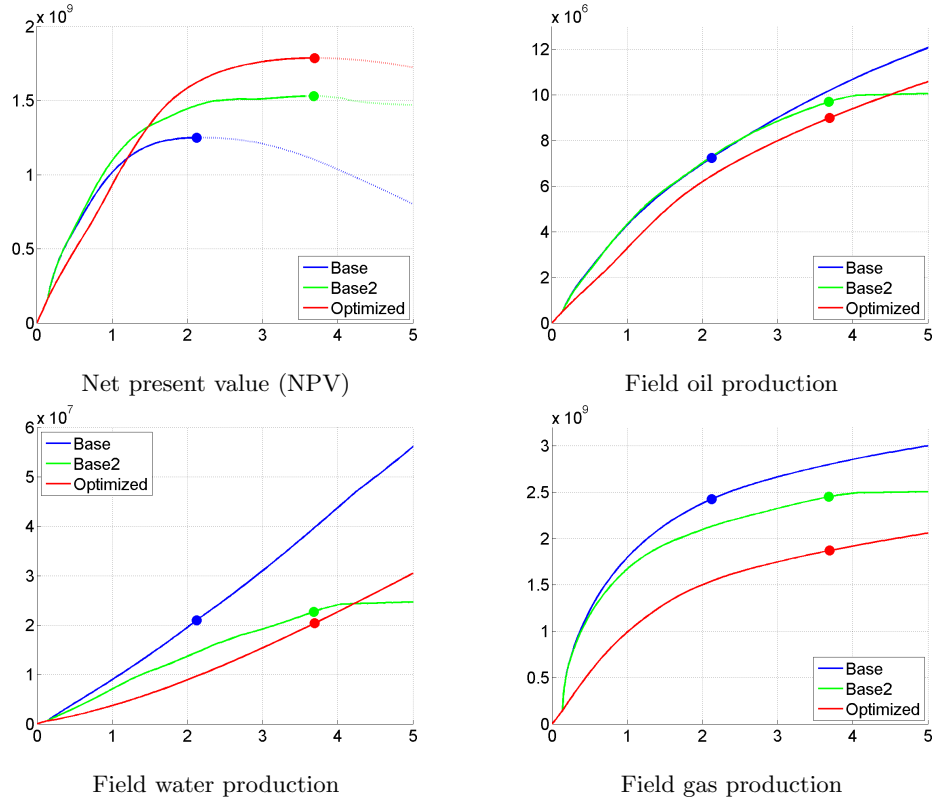


Figure 22: Net present value (NPV) and field production for the water-injection case. The markers indicate the peak NPV; production beyond this point is not profitable.

and water-production rates. In Figure 24 we have plotted the initial and end-time oil saturation for the two cases. We observe that the water/gas injection case displaces more of the oil in the top layer compared to the water-only case. Moreover, the free gas production is much higher than for the water-only case because of breakthrough from gas injectors.

5 Conclusion and summary

In this paper we have presented a computational methodology for optimization workflows within reservoir management based on fast computation of pressure, time-of-flight (TOF), and stationary tracers. Flow diagnostics differ from traditional simulations in that there is no time-stepping, which means that valuable output can be produced in seconds. Fast response times together with visual-friendly output make approached based on flow diagnostics ideal for interactive "what-if" exploration. In effect, more realizations and scenarios can be tested, while the number of traditional simulations can be reduced by eliminating the unfavorable ones at an early stage. This makes the methodology amenable for model ranking, interactive visualization and as proxies for optimization; herein, we have focused on the latter two. Flow-diagnostics as presented herein has a lot in common with streamline methods, but an important advantage over streamline methods is that the equations are discretized directly on the simulation grid. Accordingly, potential problems related to streamline tracing and solution mapping on challenging (polyhedral) grids are avoided.

Typical objectives in reservoir management are oil-recovery and net-present-value (NPV). Flow-diagnostics computations do not give direct approximations of such objectives, but for optimization purposes it suffices to compute measures that correlate

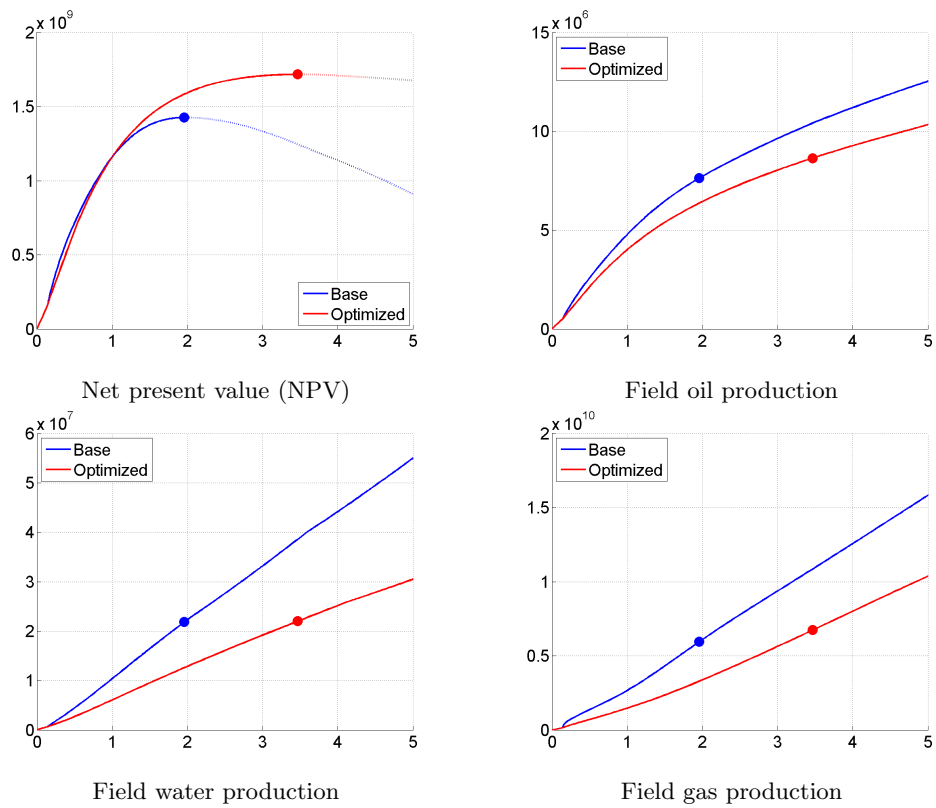


Figure 23: Net present value and field production for the water and gas injection case. The markers indicate the peak NPV; production beyond this point is not profitable.

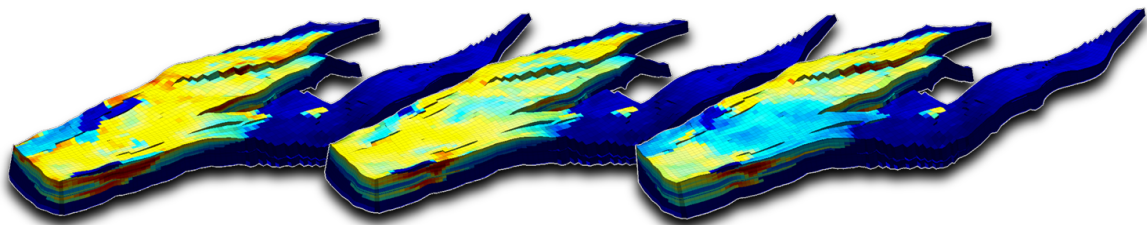


Figure 24: Original oil saturation (left), oil saturation after optimal water injection (middle), and oil saturation after optimal water/gas injection (right).

well with the actual objective. In particular, we have showed that there is a strong correlation between (variants) of the Lorenz coefficient and oil recovery for a large number of cases. The Lorenz coefficient was further used as objective in an efficient adjoint-based optimization of injection rates, well-placement, and well scheduling. Both idealized and realistic models were used and substantial improvement in recovery was achieved in all cases. Finally, we presented a TOF-adapted NPV-proxy which we used for optimizing NPV on a three-phase black-oil model of a real field.

We believe that methodologies as presented here can contribute to speed up and improve decision-support tools used in conjunction with traditional simulators for reservoir management. Unquestionably, optimization based on traditional simulations is more accurate than the flow diagnostics approach. High accuracy, however, is not worth the extra effort unless one has a mathematically precise description of the objective and constraints at hand. Obtaining a precise formulation may very well be an optimization loop in itself, in which an engineer iteratively improves on the mathematical description of the problem based on output from an inner, model-based optimization loop. In general, it is only by converging this outer loop that one can find optimal solutions that incorporate expert knowledge in a satisfactory manner. In this larger picture, interactive tools based on flow diagnostics can both accelerate the optimization of a given problem, and maybe even more important, drastically reduce the time required to obtain a precise mathematical description of the problem. Although traditional simulators can be used for optimization purposes, we believe that a drastically reduced response time (at the cost of somewhat less accurate output) will enable much more interactive exploration of different scenarios. In effect, more realizations and scenarios can be tested, potentially leading higher confidence and better decisions.

Given the simplicity and utility of flow diagnostics, we generally recommend that reservoir simulators, as well as workflow tools for building reservoir models, implement these techniques and use them for (interactive) pre- and postprocessing.

6 Acknowledgments

The research is funded in part by the Center for Integrated Operations in the Petroleum Industry (IO Center) at NTNU, by Chevron, and by the Research Council of Norway under grant no. 215665. The authors thank Statoil (operator of the Norne field) and its license partners ENI and Petoro for the release of the Norne data. Further, the authors acknowledge the IO Center for cooperation and coordination of the Norne cases.

References

- [1] J. E. Aarnes, S. Krogstad, and K.-A. Lie. Application of multiscale modelling concepts on the reservoir model for the Norne field. Technical Report F4102, SINTEF ICT, 2007.
- [2] J. E. Aarnes, S. Krogstad, and K.-A. Lie. Multiscale mixed/mimetic methods on corner-point grids. *Comput. Geosci.*, 12(3):297–315, 2008.
- [3] J. E. Aarnes, S. Krogstad, K.-A. Lie, and J. R. Natvig. Fast sequential implicit porous media flow simulations using multiscale finite elements and reordering of cells for solution of nonlinear transport equations. In *Proceedings of ECMOR X (10th European Conference on the Mathematics of Oil Recovery)*, Amsterdam, The Netherlands, September 2006. ECMOR.
- [4] F. O. Alpak, M. Pal, and K.-A. Lie. A multiscale method for modeling flow in stratigraphically complex reservoirs. *SPE J.*, 17(4):1056–1070, 2012.

- [5] H. Ates, A. Bahar, S. El-Abd, M. Charfeddine, M. Kelkar, and A. Datta-Gupta. Ranking and upscaling of geostatistical reservoir models using streamline simulation: A field case study. *SPE Res. Eval. Eng.*, 8(1):22–32, 2005.
- [6] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, London and New York, 1979.
- [7] R. P. Batycky, M. R. Thieles, R. O. Baker, and S. H. Chugh. Revisiting reservoir flood-surveillance methods using streamlines. *SPE Res. Eval. Eng.*, 11(2):387–394, 2008.
- [8] M. A. Christie and M. J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.*, 4:308–317, 2001. Url: <http://www.spe.org/csp/>.
- [9] A. Datta-Gupta and M. J. King. *Streamline Simulation: Theory and Practice*, volume 11 of *SPE Textbook Series*. Society of Petroleum Engineers, 2007.
- [10] I. S. Duff and J. K. Reid. An implementation of Tarjan’s algorithm for the block triangularization of a matrix. *ACM Trans. Math. Software*, 4(2):137–147, 1978.
- [11] Y. Efendiev and T. Y. Hou. *Multiscale Finite Element Methods*, volume 4 of *Surveys and Tutorials in the Applied Mathematical Sciences*. Springer Verlag, New York, 2009.
- [12] B. Eikemo, I. Berre, H. K. Dahle, K.-A. Lie, and J. R. Natvig. A discontinuous Galerkin method for computing time-of-flight in discrete-fracture models. In P. Binning, P. Engesgaard, H. Dahle, G. Pinder, and W. Gray, editors, *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, Copenhagen, Denmark, 18–22 June 2006.
- [13] B. Eikemo, K.-A. Lie, H. K. Dahle, and G. T. Eigestad. Discontinuous Galerkin methods for transport in advective transport in single-continuum models of fractured media. *Adv. Water Resour.*, 32(4):493–506, 2009.
- [14] H. Gross, M. Honarkhah, and Y. Chen. Uncertainty modeling of a diverse portfolio of history-matched models with distance-based methods: application to an offshore gas-condensate reservoir. In *Proceedings of the SPE Asia Pacific Oil and Gas Conference & Exhibition*, Jakarta, Indonesia, September 20–22, 2011, 2011.
- [15] H. Hægland. *Streamline methods with application to flow and transport in fractured media*. PhD thesis, University of Bergen, 2009.
- [16] E. Idrobo, M. Choudhary, and A. Datta-Gupta. Swept volume calculations and ranking of geostatistical reservoir models using streamline simulation. In *SPE/AAPG Western Regional Meeting*, Long Beach, California, USA, 19–23 June 2000. SPE 62557.
- [17] O. Izgec, M. Sayarpour, and G. M. Shook. Maximizing volumetric sweep efficiency in waterfloods with hydrocarbon f - ϕ curves. *Journal of Petroleum Science and Engineering*, 78(1):54–64, 2011.
- [18] J. F. B. M. Kraaijevanger, P. J. P. Egberts, and J. R. V. H. W. Buurman. Optimal waterflood design using the adjoint method. In *SPE Reservoir Simulation Symposium, 26-28 February, Houston, Texas, U.S.A*, number SPE-105764-MS, 2007.
- [19] S. Krogstad. A sparse basis POD for model reduction of multiphase compressible flow. In *2011 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21-23 February 2011*, 2011.
- [20] L. W. Lake. *Enhanced Oil Recovery*. Prentice-Hall, 1989.
- [21] K.-A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, 16:297–322, 2012.
- [22] K.-A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad. Open source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, 16:297–322, 2012.
- [23] K.-A. Lie, H. M. Nilsen, A. F. Rasmussen, and X. Raynaud. Fast simulation of polymer injection in heavy-oil reservoirs based on topological sorting and sequential splitting. In *2013 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18-20 February 2013*, 2013. SPE 163599-MS.

- [24] T. Manzocchi et al. Sensitivity of the impact of geological uncertainty on production from faulted and unfaulted shallow-marine oil reservoirs: objectives and methods. *Petrol. Geosci.*, 14(1):3–15, 2008.
- [25] The MATLAB Reservoir Simulation Toolbox, version 2013a, Apr. 2013. <http://www.sintef.no/MRST/>.
- [26] J. R. Natvig and K.-A. Lie. Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements. *J. Comput. Phys.*, 227(24):10108–10124, 2008.
- [27] J. R. Natvig, K.-A. Lie, and B. Eikemo. Fast solvers for flow in porous media based on discontinuous Galerkin methods and optimal reordering. In P. Binning, P. Engesgaard, H. Dahle, G. Pinder, and W. Gray, editors, *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, Copenhagen, Denmark, 18–22 June 2006.
- [28] J. R. Natvig, K.-A. Lie, B. Eikemo, and I. Berre. An efficient discontinuous Galerkin method for advective transport in porous media. *Adv. Water Resour.*, 30(12):2424–2438, 2007.
- [29] Norwegian University of Science and Technology. IO Center – Norne Benchmark Case, June 2012. <http://www.ipt.ntnu.no/~norne>.
- [30] Y. Notay. An aggregation-based algebraic multigrid method. *Electron. Trans. Numer. Anal.*, 37:123–140, 2010.
- [31] H.-Y. Park and A. Datta-Gupta. Reservoir management using streamline-based flood efficiency maps and application to rate optimization. In *Proceedings of the SPE Western North American Region Meeting*, 7-11 May 2011, Anchorage, Alaska, USA, 2011.
- [32] A. F. Rasmussen. Streamline tracing on irregular geometries. In *Proceedings of ECMOR XII–12th European Conference on the Mathematics of Oil Recovery*, Oxford, UK, 6–9 September 2010. EAGE.
- [33] C. Scheidt and J. Caers. Uncertainty quantification in reservoir performance using distances and kernel methods – application to a West-Africa deepwater turbidite reservoir. *SPE J.*, 14(4):680–692, 2009.
- [34] B. S. Sehbi, S. Kang, A. Datta-Gupta, and W. J. Lee. Optimizing fracture stages and completions in horizontal wells in tight gas reservoirs using drainage volume calculations. In *Proceedings of the North American Unconventional Gas Conference and Exhibition*, 14-16 June 2011, The Woodlands, Texas, USA, 2011.
- [35] M. Shahvali, B. Mallison, K. Wei, and H. Gross. An alternative to streamlines for flow diagnostics on structured and unstructured grids. *SPE J.*, 17(3):768–778, 2012.
- [36] G. Shook and K. Mitchell. A robust measure of heterogeneity for ranking earth models: The F-Phi curve and dynamic Lorenz coefficient. In *SPE Annual Technical Conference and Exhibition*, 2009.
- [37] M. R. Thiele and R. P. Batycky. Water injection optimization using a streamline-based workflow. In *Proceedings of the SPE Annual Technical Conference and Exhibition*, 5-8 October 2003, Denver, Colorado, 2003.
- [38] M. J. Zandvliet, M. Handels, G. M. van Essen, D. R. Brouwer, and J. D. Jansen. Adjoint-based well-placement optimization under production constraints. *SPE J.*, 13(4):392–399, DEC 2008. 2007 SPE Reservoir Simulation Symposium, Houston, TX, FEB 26-28, 2007.