A Hierarchical Multigrid Method for Oil Production Optimization

Hørsholt, Steen; Nick, Hamid; Jørgensen, John Bagterp

Published in:
IFAC-PapersOnLine

Link to article, DOI:
10.1016/j.ifacol.2019.06.110

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
A Hierarchical Multigrid Method for Oil Production Optimization

Steen Hørscholt * Hamid Nick ** John Bagterp Jørgensen*

* Department of Applied Mathematics and Computer Science & Center for Energy Resources Engineering (CERE), Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark.
** Danish Hydrocarbon Research and Technology Centre, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

Abstract: The large-scale optimization problems that arise from oil production optimization under geological uncertainty of industry-scale reservoir models poses a challenge even for modern computer architecture. In combination with ensemble-based methods for production optimization under uncertainty, gradient-based optimization algorithms provides a powerful approach that ensures a high convergence rate. However, the spatial resolution and complexity of typical industry-scale models has a significant computational impact that renders the optimization problem intractable. To reduce the computational burden model reduction is essential. In this paper, we introduce a grid coarsening method that maintains the overall dynamics of the flow, by preserving the geological features of the model. Furthermore, we present a hierarchical multigrid method for oil production optimization. The method utilizes a hierarchy of coarse level models based on the high-fidelity model. We present the workflow of the hierarchical multigrid optimization procedure and a numerical example that demonstrates the application of oil production optimization on a synthetic reservoir.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Production optimization, Model reduction, Reservoir management, Work-flow, Gradient based optimization

1. INTRODUCTION

The increasing level of detail and complexity of reservoir models is a result of the modern reservoir characterization methods and geological modeling techniques. Together with a growing amount of data, this allows geologist to build high-resolution reservoir models able to reflect the geological heterogeneity. Geological reservoir models comprise detailed heterogeneous geological information at multiple length scales. Geological models is often at the resolution of $10^7$ to $10^8$ grid cells. Even with the computational capabilities of modern computers, the computational burden of such models is in many cases intractable. Therefore, model reduction (grid coarsening and property upscaling) is required to build reservoir models at a scale manageable for simulation. Upscaling techniques that are able to capture the heterogeneity effects of a high-resolution model has been a long-time focus of the research community (Christie, 1996; King and Mansfield, 1997; Durlofsky, 2005). The methods includes statistical averaging techniques and flow-based methods. The typical resolution of upscaled models is roughly, $10^2$ to $10^3$ cells, with simulation runtimes on the scale of hours. Production optimization workflows require a potentially very high number of reservoir simulations. In many cases, the direct use of high-fidelity models (upscaled geological model), will render the optimization problem computationally intractable. The computational load of the optimal control problem in a deterministic production optimization procedure depends on the size of the simulation model, the number of controlled wells and the number of control intervals. Different reduced-order modeling methods that provides for approximate but nonetheless fairly accurate simulation of models, has been proposed by van Doren et al. (2005) and by Cardoso and Durlofsky (2010). Oliveira and Reynolds (2015) propose the use of a hierarchical multiscale method for coarsening the control interval parametrization, based on control value and gradient criteria. Aliyev and Durlofsky (2015) proposes a multilevel procedure for optimizing well-placement and well-controllers, which utilizes a gradient-free particle-swarm optimization algorithm. Krogstad et al. (2016) use a global upscaling procedure to compute a coarse model tuned to specific flow scenarios and applies an adjoint gradient-based optimization procedure.

In this paper, we introduce a model reduction procedure that reduces the computational impact of a high-fidelity model. The model reduction procedure consists of a coarsening procedure and an upscaling procedure and aims to reduce the turnover time compared to typical model reduction workflows. The coarsening method preserves geological features important for capturing the flow dynamics of the high-resolution model. To ensure flexibility in the choice of reservoir simulators we only consider logically indexed grids, i.e. corner-point grids. For the permeability upscaling, we apply a local single-phase flow-
based procedure. We apply the approximate coarse-scale well model, proposed by Chen and Wu (2008) to upscale the well-index of well-bore completions. By applying these methods, we can significantly reduce the simulation time, while capturing the overall flow dynamics of the high-fidelity model. We introduce a hierarchical multigrid optimization procedure that optimizes an objective function over a multigrid hierarchy of coarsened-upscaled levels of a high-fidelity model. The procedure is presented from an oil industry perspective in Hørsholt et al. (2018b). The hierarchical multigrid optimization procedure integrates with our in-house gradient-based constrained optimization software tool RESOPT (Hørsholt et al., 2018a).

The structure of the paper is as follows. A short description of the dynamic optimization problem, including the system of differential algebraic equations that governs multiphase flow in porous media, the net present value objective function and the optimal control problem. The next section describes the coarsening procedure, the grid property up-scaling procedure and the well-index upscaling procedure. Next, we introduce the workflow of the hierarchical multigrid optimization procedure. Finally, we present a numerical example that demonstrates the optimization workflow in a deterministic water-flooding scenario. The example is a rate-controlled optimization of the well-known Egg model (Jansen et al., 2013).

2. OPTIMAL CONTROL PROBLEM

In this section, we state the dynamic optimization problem and the underlying system of differential algebraic equations for multiphase flow in subsurface porous media, the net present value objective and the optimal control problem for oil production optimization.

2.1 Dynamic optimization problem

The dynamic optimization problem computes an open loop operating strategy that optimizes some performance measure over a given time horizon. We consider the dynamic optimization problem in the form

$$\begin{align*}
\min_{\{x(t);y(t);u_k\}_{k=0}^{N-1}} & \varphi = \int_{t_0}^{t_f} \phi(y(t),u(t),\theta)dt, \\
\text{subject to} & \\
x(t_0) = x_0, \\
G(x(t),y(t),\theta) = 0, & t \in [t_0,t_f], \\
\dot{x}(t) = F(y(t),u(t),\theta), & t \in [t_0,t_f], \\
u(t) = u_k, & t \in [t_0,t_f], \ k = 0,\ldots,N-1, \\
\{u_k\}_{k=0}^{N-1} & \in U.
\end{align*}$$

(1a)

Temporal discretization by the implicit Euler method gives the system of non-linear residual equations, $R_{k+1} = 0, k = 0,1,\ldots,N-1$. To allow for a more compact notation, we introduce the vectors, $\bar{x} = (x_1;x_2;\ldots;x_N; y_1; y_2;\ldots;y_N)$, $\bar{R} = (R_1; R_2;\ldots; R_N)$, and $\bar{u} = (u_0; u_1;\ldots; u_{N-1})$. In compact notation we write the residual equations as

$$\bar{R}(\bar{x}, \bar{u}, x_0; \theta) = 0.$$  

(2)

2.2 Objective function and net present value

As a performance measure, $\varphi$ in the dynamic optimization problem (1a), we consider the net present value. The net present value of an oil field is defined as a function, $\varphi$, of the states, $x(t)$, the algebraic variables, $y(t)$, the operating profile, $u(t)$, the initial state, $x_0$, and the geological parameters, $\theta$. In continuous time the definition is

$$\varphi = \int_{t_0}^{t_f} \phi(x(t),y(t),u(t),\theta)dt,$$

(3a)

where $\phi$ is the discounted profit rate

$$\phi(t, x(t), y(t), u(t), \theta) = e^{-d_r(t-t_0)/\tau_r} p(x(t), y(t), u(t), \theta),$$

(3b)

and $p$ is the profit rate

$$p = p(x(t), y(t), u(t), \theta) = \left[r_o q_{Os}(t) + r_g q_{Gs}(t) - (r_w q_{Ws}(t) + r_{w, inj} q_{w, inj}(t) + r_{g, inj} q_{g, inj}(t))\right].$$

(3c)

$x(t)$ and $y(t)$ are the solution to (2), $u(t)$ is the operating profile, $d_c$ is the continuous-time discount rate and $\tau_r$ is the discounted time interval (i.e. the time interval at which compound interest are ascribed). $r_o$ and $r_g$ are the sales prices of oil and gas. $r_w$ is the water production cost, $r_{w, inj}$ and $r_{g, inj}$ are the water and gas injection cost. The corresponding surface flow rates for oil, gas, water injection and gas injection are, $q_{Os}$, $q_{Gs}$, $q_{Ws}$, $q_{w, inj}$, $q_{g, inj}$.

We apply the implicit Euler method for temporal discretization. Hence the net present value function in discrete-time is

$$\Phi(\bar{x}, \bar{u}, x_0; \theta) = \sum_{k=0}^{N-1} J_k \left(x_{k+1}, u_k; \theta\right),$$

(4)

where the discounted net present value, $J_k$, for the $k^\text{th}$ time interval is given by

$$J_k = \Delta t_k \left[ r_o q_{Os,k+1} + \frac{r_g q_{Gs,k+1} - (r_w q_{Ws,k+1} + r_{w, inj} q_{w, inj,k+1})}{(1 + d)^{t_{k+1}/\tau_r}} \right].$$

(5)

Note that the relation between the discrete-time discount rate, $d$, and the continuous-time discount rate, $d_c$, is $d_c = \ln(1 + d)$.

2.3 Optimal control problem

The discrete-time constrained optimal control problem for oil production optimization is given by (Capolei et al., 2012)

$$\max_{\bar{u} \in U} \psi = \psi(\bar{u}; x_0, \theta),$$

(6)
where the objective function is
\[ \psi(\bar{u}; x_0, \theta) = \left\{ F(\bar{x}, \bar{u}, x_0, \theta) : \bar{R}(\bar{x}, \bar{u}, x_0, \theta) = 0 \right\}. \] (7)

In this work, we only consider linear input constraints. The constraints are bounds on controls, rate of movement constraints on controls to prevent large jumps, and bounds on total injection rates in each time step. We write the constraints as
\[ U = \{ \bar{u} : \bar{u}_{\text{min}} \leq \bar{u} \leq \bar{u}_{\text{max}}, \Delta \bar{u}_{\text{min}} \leq \Delta \bar{u} \leq \Delta \bar{u}_{\text{max}}, b_l \leq \bar{A} \bar{u} \leq b_u \}. \] (8)

2.4 Black-box simulator

The optimization workflow presented in this paper does not depend on any specific formulation and/or solution method of the governing equations (1b)-(1e). Thus we treat the reservoir simulator as a black-box function, \( S(\bar{u}; x_0, \theta) \). For a simulator to be an eligible candidate for gradient-based production optimization purposes, it must fulfill the following requirements. Given an initial state, \( x_0 \), an operating profile, \( u \), and a geological realization vector, \( \theta \), the simulator must at every time-step return the states, \( \bar{x} \), the net present value, \( \psi \), and the gradient of the net present value with respect to the controls, \( \nabla_{\bar{u}} \psi \).

\[ [\bar{x}, \psi, \nabla_{\bar{u}} \psi] = S(\bar{u}; x_0, \theta). \] (9)

3. MODEL REDUCTION PROCEDURE

In this section, we outline the workflow of the model reduction procedure. We describe a practical grid-coarsening method, which preserve the geological features of a high-resolution model to a satisfactory degree for optimization purposes. We apply simple averaging of additive properties, e.g. porosity and net to gross and local flow-based upscaling for the absolute permeability. To capture the effect of high-resolution heterogeneity in the near well region, we upscale the well-index by the "near-well arithmetic averaging" method proposed by Chen and Wu (2008).

3.1 Coarsening

In this work, we only consider logical grids with \( ijk \)-indexing, i.e. Cartesian grids and corner-point grids. This means that all grid cell has exactly 6-faces. The grid indexing is defined by the "normal reading order", i.e. the \( i \)-index runs first, followed by the \( j \)-index and last by the \( k \)-index. The product of grid cells in each direction is the total number of grid cells, \( N_c \). Formally a grid consists of a set of \( ijk \)-indexed grid cells, \( \{c_n\}_{n=1}^{N_c} \), \( N_c = N_x N_y N_z \).

To construct a coarse grid based on a high-resolution grid, \( \{c_n\}_{n=1}^{N_{fine}} \), we group the fine grid into a number of subsets and amalgamate them into coarse grid cells, \( \{c_n\}_{n=1}^{N_{coarse}} \), \( N_{fine} = N_{x,coarse} N_{y,coarse} N_{z,coarse} \). An amalgamated coarse grid is represented by a partitioning vector, \( p \in N_{fine} \), that maps all fine grid cells into a coarse grid cell, i.e. \( p_i = j \), if the fine grid cell, \( c_i \), belongs to coarse grid cell, \( c_j \). Although this allows for grouping arbitrary selections of fine grid cells into a subset, we only allow amalgamation of neighboring cells in the fine grid. The restriction that all cells has 6-faces means the corner nodes consists of the outermost fine-grid cell corner nodes. We note that this procedure does not preserve the reservoir volume. However, for the purpose of production optimization, this is of less importance. A simple way of computing the partitioning is to divide the dimensions of a fine grid by a coarsening factor in each direction, \( (c_{f_x}, c_{f_y}, c_{f_z}) \). The coarse grid dimensions are then computed by
\[ (N_{x,coarse}, N_{y,coarse}, N_{z,coarse}) = ([N_{x,fine}/c_{f_x}], [N_{y,fine}/c_{f_y}], [N_{z,fine}/c_{f_z}]). \] (10)

When there is no important geological features to account for, e.g. faults, this is a very robust method. In the grid-processing for a faulted reservoir the geologist/engineer typically tries to align the grid with the fault directions. In this case we identify the direction and the, \( ijk \)-index of the faults and amalgamate cells on each side of the faults. By applying these techniques, we construct coarse corner-point grids that preserve the geological features of a high-fidelity model to a degree that is satisfactory for optimization purposes.

3.2 Flow-based upscaling

After the construction of a coarse-grid, the problem of populating the grid cell with properties, e.g. net to gross, porosity and permeability remains. There exist numerous property upscaling methods. The methods also depends on the type of the upscaled property. These methods includes simple averaging of properties over the fine grid to more advanced methods, such as flow-based permeability upscaling. For high-level coarsening of multi-phase flow systems, upscaling the relative permeability functions is often required. However, we do not consider relative permeability upscaling in this paper. For a more detailed discussion of these methods, we refer the reader to Barker and Thibeau (1997), and Renard and De Marsily (1997).

In this paper, we only consider upscaling of the following grid properties: net to gross, porosity and absolute permeability. We apply arithmetic average upscaling of the porosity and net to gross properties. To upscale the absolute permeability we apply local single-phase flow-based method. To upscale the absolute permeability we apply local single-phase flow-based method. By emulating a laboratory setup, we compute each component in the permeability tensor separately. The setup applies a pressure drop in each axial direction and keeps all other boundaries sealed. In the \( x \)-direction we compute the upscaled permeability, \( K_x^\text{eff} \) in a coarse-grid cell as
\[ K_x^\text{eff} = \frac{q_{x,\text{out}/x}}{L_x L_y (P_{\text{out}} - P_{\text{in}})}, \] (11)
where, \( q_{x,\text{out}} \), is the flow through the boundary, \( L_x, L_y, L_z \), are the cell dimensions, and \( P_{\text{in}}, P_{\text{out}} \) are the pressures at the boundaries. The absolute permeability in the, \( y \) and \( z \)-direction is upscaled by the same method.

3.3 Well-index upscaling

We upscale the well-index, WI, to capture the effects of the high resolution heterogeneous permeability in the near-well regions. The upscaling is computed by applying the method "near-well arithmetic averaging" proposed by Chen and Wu (2008). The method applies arithmetic averaging of the effective permeabilities, \( K_{\text{eff}} \), along the
well trajectory within a coarse cell. We denote the fine scale permeability and the fine scale volume along the well trajectory in a coarse cell as, \(K_{\text{fine}}\) and \(V_{\text{fine}}\), respectively. The effective permeability averaging in a given direction is then computed as

\[
K_{\text{eff}} = \frac{\sum K_{\text{fine}} V_{\text{fine}}}{\sum V_{\text{fine}}}. \tag{12}
\]

The coarse scale well-index for a vertical well is computed using the Peaceman model by inserting the effective permeabilities:

\[
W_{\text{coarse}} = \frac{2\pi \sqrt{K_{z,\text{eff}} K_{y,\text{eff}}} \Delta z}{\ln \left( \frac{r_o}{r_w} \right)}. \tag{13}
\]

### 4. HIERARCHICAL MULTIGRID OPTIMIZATION WORKFLOW

The optimization workflow presented in this paper utilizes a hierarchy of coarse level models, based on the high-fidelity model. In the following, we refer to the high-fidelity model as the basecase and the increasingly coarse models as levels. The levels are numbered in reverse order, i.e. in a set of coarse models consisting of, \(n\), levels the coarsest level is labeled, level-\(n\). By applying the coarsening-upscaling procedure presented in the previous section, we compute a hierarchy of coarse level models consisting of, \(n\)-levels. The optimization procedure starts at level-\(n\), which is optimized with an initial guess of the operating profile. The computed set of optimal controls is then used as the staring guess on level-(\(n-1\)). This procedure is repeated for each subsequent level. The optimization procedure can be stopped at any level depending on the aim of the optimization, time limitations and the computational resources available. Figure 1 illustrates the concept of the hierarchical multigrid optimization workflow.

### 5. CASE STUDY

As a case study we consider the well-known Egg-model proposed by (Jansen et al., 2013). The Egg-model is a three-dimensional synthetic two-phase flow reservoir model designed as a benchmark test case for production optimization. The model consist of 101 realizations of the permeability field. In this paper, we consider the first realization only.

#### 5.1 Egg-model

The physical reservoir dimensions of the reservoir are, \((480, 480, 28)\)m. It is discretized on a Cartesian grid with \((N_x, N_y, N_z) = (60, 60, 7)\) cells of uniform size, \((\Delta x, \Delta y, \Delta z) = (8, 8, 4)\) m. The total of number of cells are \(25200\), where \(18553\) are active. The porosity is uniform, \(\phi = 0.2\), and the permeability field is highly channelized. The reservoir has eight water injection wells and four producer wells. All wells are vertical and are completed throughout all layers in the \(z\)-direction. For a detailed description of the model see (Jansen et al., 2013).

#### 5.2 Optimization parameters

The reservoir production is simulated for 3600 days with 120 control time-steps of length, \(\Delta t = 30\) days. The water injection rate in all injector wells are subject to control, giving 960 controls, \(u\). The bottom-hole pressures in the producers are constant at 395 bar in all controls steps. All injection rates are bounded in the interval \([0, 80] m^3/\text{day}\) and constrained by a rate of movement of \(5 m^3/\text{day}\) between control intervals. The discount rate, \(d\) in (5) is set to 10%. The optimization is initialized with a water injection rate of, \(79.0 m^3/\text{day}\).

#### 5.3 Model hierarchy

The multigrid hierarchy set consists of four levels of the Egg-model. For the model reduction procedure we keep the vertical resolution of the high-resolution model intact and apply three levels of coarsening factors, \((2, 2, 1), (4, 4, 1), \) and \((8, 8, 1)\). As mentioned earlier this does not preserve the grid volume, but this is not essential for the purpose of the optimization. Table 2 shows the details of the model reduction procedure. The base-case model properties are upscaled as described in section 3. The uniform porosity requires no upsampling. The absolute permeability is upscaled by the local flow-based method. The well-index is upscaled by the near well arithmetic averaging method. Figure 2 shows the Egg-model multigrid hierarchy with the upscaled horizontal permeability.

#### 5.4 Optimization procedure

The hierarchical multigrid optimization procedure starts at level-3 (the coarsest level), where the optimization is initiated with, \(u_{\text{init}}\). After convergence the optimal controls, \(u_{\text{opt,level-1}}\), is passed on to the next level where it is used as the initial guess on the operating profile. This procedure continues through the hierarchy to the base-level. This gives us four sets of optimal controls denoted, \(u_{\text{opt,level-3}}, u_{\text{opt,level-2}}, u_{\text{opt,level-1}}, u_{\text{opt,base-level}}\). At all levels in the hierarchy, we optimize by first applying a SQP-algorithm for a limited number of iterations and then

---

**Table 1. Egg-model; optimization parameters.**

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Value</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimization strategy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of control steps</td>
<td></td>
<td>120</td>
<td>-</td>
</tr>
<tr>
<td>length of control steps</td>
<td></td>
<td>30 [day]</td>
<td>-</td>
</tr>
<tr>
<td>controls, water inj rates</td>
<td>(q_{\text{wi}})</td>
<td>- [m^3/day]</td>
<td>-</td>
</tr>
<tr>
<td>bound constraints</td>
<td>(q_{\text{wi, min}})</td>
<td>0.2 [m^3/day]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(q_{\text{wi, max}})</td>
<td>80.0 [m^3/day]</td>
<td>-</td>
</tr>
<tr>
<td>rate of movement</td>
<td>(r_{\text{om}})</td>
<td>±5 [m^3/day]</td>
<td>-</td>
</tr>
<tr>
<td>initial controls</td>
<td>(u_{\text{init}})</td>
<td>79.0 [m^3/day]</td>
<td>-</td>
</tr>
<tr>
<td>prices</td>
<td>(r_{\text{a}})</td>
<td>- [US$/m^3]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(r_{\text{w}})</td>
<td>283.04 [US$/m^3]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(r_{\text{wi,min}})</td>
<td>37.74 [US$/m^3]</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>(r_{\text{wi,max}})</td>
<td>12.58 [US$/m^3]</td>
<td>-</td>
</tr>
<tr>
<td>discount factor</td>
<td>(d)</td>
<td>10%</td>
<td>-</td>
</tr>
<tr>
<td>optimization algorithm</td>
<td></td>
<td>SQP, interior-point</td>
<td>-</td>
</tr>
<tr>
<td>optimality tolerance</td>
<td>(\text{tol}_{\text{opt}})</td>
<td>1e-6</td>
<td>-</td>
</tr>
<tr>
<td>step-size tolerance</td>
<td>(\text{tol}_{\text{x}})</td>
<td>1e-6</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 2. Egg-model grid coarsening levels.**

<table>
<thead>
<tr>
<th>Level</th>
<th>coarsening factor</th>
<th>grid dimensions</th>
<th># cells active cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>base-case</td>
<td>-</td>
<td>(100, 100, 7)</td>
<td>25200</td>
</tr>
<tr>
<td>level-1</td>
<td>(2, 2, 1)</td>
<td>(30, 30, 7)</td>
<td>6300</td>
</tr>
<tr>
<td>level-2</td>
<td>(4, 4, 1)</td>
<td>(15, 15, 7)</td>
<td>1575</td>
</tr>
<tr>
<td>level-3</td>
<td>(8, 8, 1)</td>
<td>(8, 8, 7)</td>
<td>448</td>
</tr>
</tbody>
</table>

---
2019 IFAC DYCOPS

We define the limit of the number of iterations and simulation time of the two algorithms depending on the hierarchy level. When the cost function becomes uneconomical, i.e. the water-cut rises above 0.88, we switch to an interior-point algorithm until convergence.

Table 3. Combined SQP and interior-point strategy tolerances.

<table>
<thead>
<tr>
<th>level</th>
<th>SQP max it</th>
<th>SQP max sim</th>
<th>interior-point max it</th>
<th>interior-point max sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>level-3*</td>
<td>50</td>
<td>300</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>level-2</td>
<td>25</td>
<td>300</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>level-1</td>
<td>10</td>
<td>300</td>
<td>50</td>
<td>300</td>
</tr>
<tr>
<td>base-level</td>
<td>10</td>
<td>300</td>
<td>25</td>
<td>300</td>
</tr>
<tr>
<td>level-2*</td>
<td>50</td>
<td>300</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>level-1*</td>
<td>50</td>
<td>300</td>
<td>50</td>
<td>300</td>
</tr>
<tr>
<td>base-level*</td>
<td>50</td>
<td>300</td>
<td>50</td>
<td>300</td>
</tr>
</tbody>
</table>

The result shows a significant reduction of both the simulation time and adjoint simulation time for all levels in the hierarchy. However, as the reduction is directly dependent on the reduction in the number of grid cells, the difference between levels decreases the lower the levels are. This means that one needs to consider the number of levels needed in the hierarchy. Table 5 lists the results for all levels including the individual optimizations with initial controls, \( \tilde{u}_{\text{init}} \).

Evaluation of the obtained optimal controls at the E100-base level, shows that the net present value increases at every level. This is also the case for the individual level optimizations using, \( \tilde{u}_{\text{init}} \), as initial controls. Only optimization at the coarsest level, level-3*, is not able to provide a net present value higher than that obtained by the reactive-strategy case.

5.5 Results

Table 4 list the combined average forward simulation time and adjoint simulation time for all levels in the hierarchy. The table shows that there is a significant reduction in the simulation time for all lower levels in the hierarchy. This is expected and not significant as long as we experience an increasing objective value, when evaluated at the high-fidelity level.

Table 4. Combined average simulation time for all levels in the hierarchy.

<table>
<thead>
<tr>
<th>Hierarchy level</th>
<th>base-level</th>
<th>level-1</th>
<th>level-2</th>
<th>level-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time [s]</td>
<td>210.24</td>
<td>31.94</td>
<td>15.69</td>
<td>113.14</td>
</tr>
</tbody>
</table>

Table 5. Hierarchical multigrid optimization results.

<table>
<thead>
<tr>
<th>level</th>
<th># iter</th>
<th># sims</th>
<th>time [hour]</th>
<th>flag</th>
<th>NPV E100</th>
</tr>
</thead>
<tbody>
<tr>
<td>level-3*</td>
<td>46</td>
<td>351</td>
<td>2.079</td>
<td>max sim</td>
<td>89.7</td>
</tr>
<tr>
<td>level-2</td>
<td>27</td>
<td>261</td>
<td>1.313</td>
<td>( \Delta \tilde{u} )</td>
<td>91.1</td>
</tr>
<tr>
<td>level-1</td>
<td>38</td>
<td>138</td>
<td>1.131</td>
<td>max it</td>
<td>88.0</td>
</tr>
<tr>
<td>base-level</td>
<td>19</td>
<td>193</td>
<td>0.592</td>
<td>max it</td>
<td>89.2</td>
</tr>
<tr>
<td>level-2*</td>
<td>66</td>
<td>318</td>
<td>1.644</td>
<td>( \Delta \tilde{u} )</td>
<td>91.2</td>
</tr>
<tr>
<td>level-1*</td>
<td>50</td>
<td>370</td>
<td>3.532</td>
<td>max sim</td>
<td>86.7</td>
</tr>
<tr>
<td>base-level*</td>
<td>38</td>
<td>248</td>
<td>11.527</td>
<td>( \Delta \tilde{u} )</td>
<td>88.7</td>
</tr>
<tr>
<td>Reactive-strategy</td>
<td>46</td>
<td>351</td>
<td>2.079</td>
<td>max sim</td>
<td>89.7</td>
</tr>
</tbody>
</table>

The hierarchical multigrid optimization procedure, we see clear jumps in the objective value when moving from one level to the next. This is partially an effect of the fact that the coarsening/upscaling procedure does not preserve volumes. Furthermore, the coarser a level is, the worse the ability to capture of the flow dynamic of the high-fidelity model becomes. However, this is expected and not significant as long as we experience an increasing objective value, when evaluated at the high-fidelity level.
iterations. The result shows a significant reduction of both the simulation run time and the number of performed optimization procedures workflow on the Egg-model (Jansen et al., 2013).

We have introduced a model reduction method for oil production optimization, which significantly decrease the computational impact of the forward simulations while maintaining the overall flow dynamics of the high-fidelity model. Furthermore, we introduce a hierarchical multigrid optimization procedure based on the model reduction method. The optimization procedure utilizes a hierarchy of coarsened and upscaled model levels based on a high-fidelity model. We demonstrate the model reduction method and the hierarchical multigrid optimization procedure workflow on the Egg-model (Jansen et al., 2013). The result shows a significant reduction of both the simulation run time and the number of performed optimization iterations.

6. CONCLUSION

We have introduced a model reduction method for oil production optimization, which significantly decrease the computational impact of the forward simulations while maintaining the overall flow dynamics of the high-fidelity model. Furthermore, we introduce a hierarchical multigrid optimization procedure based on the model reduction method. The optimization procedure utilizes a hierarchy of coarsened and upscaled model levels based on a high-fidelity model. We demonstrate the model reduction method and the hierarchical multigrid optimization procedure workflow on the Egg-model (Jansen et al., 2013). The result shows a significant reduction of both the simulation run time and the number of performed optimization iterations.

REFERENCES


