Adaptive parameterization for solving of thermal/compositional nonlinear flow and transport with buoyancy

Khait, Mark; Voskov, Denis

DOI
10.2118/182685-PA

Publication date
2018

Document Version
Final published version

Published in
SPE Journal

Adaptive Parameterization for Solving of Thermal/Compositional Nonlinear Flow and Transport With Buoyancy

Mark Khait, Delft University of Technology, and
Denis Voskov, Delft University of Technology and Stanford University

Summary

The nonlinear nature of flow and transport in porous media requires a linearization of the governing numerical-model equations. We propose a new linearization approach and apply it to complex thermal/compositional problems. The key idea of the approach is the transformation of discretized mass- and energy-conservation equations to an operator form with separate space-dependent and state-dependent components. The state-dependent operators are parameterized using a uniformly distributed mesh in parameter space. Multilinear interpolation is used during simulation for a continuous reconstruction of state-dependent operators that are used in the assembly of the Jacobian and residual of the nonlinear problem. This approach approximates exact physics of a simulation problem, which is similar to an approximate representation of space and time discretization performed in conventional simulation. Maintaining control of the error in approximate physics, we perform an adaptive parameterization to improve the performance and flexibility of the method. In addition, we extend the method to compositional problems with buoyancy. We demonstrate the robustness and convergence of the approach using problems of practical interest.

Introduction

Numerical simulations are essential for the modern development of subsurface reservoirs (Aziz and Settari 1979). They are widely used for the evaluation of oil-recovery efficiency, performance analysis, and various optimization problems. Because of the complexity of underlying physical processes and considerable uncertainties in the geological representation of reservoirs, there is a persistent demand for more-accurate models. To increase the accuracy of a model, one can apply a more-refined computational grid in space or time, or use a more-detailed description of the fluids, such as in a thermal/compositional model. However, an improvement in the accuracy of numerical models is usually counterbalanced by a reduction in the overall performance of simulation. In addition, both space and time approximations introduce nonlinearity to the system of governing equations that needs to be solved numerically.

After the discretization stage is complete, a nonlinear system needs to be linearized. Usually, linearization is made using a Newton-based method, which demands an assembly of the Jacobian and the residual for the combined system of equations. Depending on the formulation, different types of nonlinear unknowns and strategies can be used to perform the nonlinear update (Aziz and Wong 1989). The most frequently used approaches for reservoir simulation are natural (Coats 1980) and molar formulations (Acs et al. 1985; Collins et al. 1992). Usually, the natural formulation in combination with different bypassing strategies for a stability test (Rasmussen et al. 2006; Voskov and Tchelepi 2009) performs better in immiscible gas displacement, whereas the molar formulation does better for miscible gas injection (Voskov and Tchelepi 2012; Zaydullin et al. 2013). It was shown recently that some specific treatments on phase appearance or disappearance may help to improve the nonlinear behavior of the natural formulation in the miscible regimes (Cao et al. 2017).

Once the linear system with the Jacobian and residual is constructed, it needs to be solved. Because the dimensionality of a typical reservoir-simulation problem is rather high, iterative linear solvers are usually used with effective preconditioning (e.g., two-stage preconditioning using constrained pressure residual) (Wallis et al. 1985). Once the solution to the linear system with predefined tolerance is found, we need to update the nonlinear unknowns and repeat the nonlinear iteration. The nonlinear solution may require several nonlinear iterations to converge, depending on the nonlinearity of a problem. The number of nonlinear iterations can be sufficiently reduced by using advanced nonlinear solvers. For example, the natural formulation can benefit from the nonlinear solvers that include an inflection-point correction (Jenny et al. 2009), specifically designed for transport problems. The possibility to apply a direct correction to a saturation in the nonlinear loop helps to design more-advanced strategies, such as the trust-region correction (Wang and Tchelepi 2013). In addition, the extension of the natural formulation can help to avoid variable substitution and apply corrections to discontinuous changes in the derivative usually related to phase appearance and disappearance (Voskov 2012).

Although different advanced nonlinear solvers were developed for the natural formulation, there is a lack of advanced strategies for the molar formulation. A version of a trust-region correction was developed for molar formulation (Voskov and Tchelepi 2011), but it still lacks robustness compared with the natural formulation. This can be explained by the more-complicated nonlinear update procedure, which requires performing an exact flash for every block at a two-phase state in each nonlinear iteration. This problem can be avoided by using parameterization in compositional space instead. A strategy derived from the compositional-space-parameterization idea (Voskov and Tchelepi 2009), was designed by Zaydullin et al. (2013). The nonlinear solver dependent on a special-point correction along the fractional-flow curve shows robustness and efficiency (Zaydullin et al. 2013). However, this approach requires reformulation of a nonlinear problem in a tie-line space, and formally cannot be applied to the conventional molar formulation.

Another approach for the molar formulation, called operator-based linearization (OBL), was proposed by Voskov (2017). It could be seen as an extension of the idea to abstract the representation of properties from the governing equations, suggested by Zaydullin et al. (2013) and Haugen and Beckner (2015). In the OBL approach, the parameterization is performed dependent on the conventional molar variables. A similar approach can be designed for the natural formulation, but it requires dealing with interpolation between regions with different numbers of variables. In the OBL approach, all properties involved in the governing equations are lumped in a few operators, which are parameterized in the parameter space of the simulation problem either in advance or adaptively during the

This paper (SPE 182685) was accepted for presentation at the SPE Reservoir Simulation Conference, Montgomery, Texas, USA, 20–22 February 2017, and revised for publication. Original manuscript received for review 27 November 2016. Revised manuscript received for review 27 August 2017. Paper peer approved 31 August 2017.

simulation process. The control on the size of parameterization hypervolume helps to preserve the balance between the accuracy of approximation and the performance of the nonlinear solver (Khait and Voskov 2016). In the current version of the OBL approach, we do not reduce the number of unknowns and only use the fact that the physical description (i.e., fluid properties) is represented using piecewise linear interpolation.

In this paper, we show the applicability of the OBL approach for general-purpose simulation using the thermal/compositional description. We demonstrate the numerical convergence of the physical solution using the OBL technique for problems with a large number of components (up to six). In addition, we extend the method for cases with buoyancy. We describe an approach for the robust treatment of gravity forces and demonstrate its applicability for challenging problems, including a full-field example.

Governing Equations and Nonlinear Formulation

In this section, we briefly consider the governing equations and nonlinear formulation for a general-purpose thermal/compositional simulator. These equations describe the conservation of mass and energy in a system with n_p phases and n_c components:

$$\frac{\partial}{\partial t}\left(\phi\sum_{p=1}^{n_p} x_{cp}\rho_p s_p\right) + \mathrm{div}\sum_{p=1}^{n_p} x_{cp}\rho_p \overrightarrow{u_p} + \sum_{p=1}^{n_p} x_{cp}\rho_p \tilde{q}_p = 0, \quad c = 1, \ldots, n_c, \quad \ldots\ldots (1)$$

$$\frac{\partial}{\partial t}\left[\phi\sum_{p=1}^{n_p} \rho_p s_p U_p + (1-\phi)U_r\right] + \mathrm{div}\sum_{p=1}^{n_p} h_p\rho_p \overrightarrow{u_p} + \mathrm{div}(\kappa\nabla T) + \sum_{p=1}^{n_p} h_p\rho_p\tilde{q}_p = 0. \quad \ldots\ldots (2)$$

All terms of the system in Eqs. 1 and 2, written for an arbitrary control volume, can be characterized as functions of the spatially distributed properties—$\boldsymbol{\xi}$ of control volume, physical state $\boldsymbol{\omega}$ of the control volume (and that of its neighbors), and well-control variable $\mathbf{u}$—as follows: $\phi(\boldsymbol{\xi}, \boldsymbol{\omega})$ is effective rock porosity; $x_{cp}(\boldsymbol{\omega})$ is component mole fraction in a phase; $\rho_p(\boldsymbol{\omega})$ is phase molar density; $s_p(\boldsymbol{\omega})$ is phase saturation; $\overrightarrow{u_p}(\boldsymbol{\xi}, \boldsymbol{\omega})$ is phase velocity; $\tilde{q}_p(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u})$ is phase influx/outflux; $U_p(\boldsymbol{\omega})$ is phase internal energy; $U_r(\boldsymbol{\xi}, \boldsymbol{\omega})$ is rock internal energy; $h_p(\boldsymbol{\omega})$ is phase enthalpy; and $\kappa(\boldsymbol{\xi}, \boldsymbol{\omega})$ is thermal conduction.

Phase-flow velocity is assumed to follow Darcy's law as

$$\overrightarrow{u_j} = -[\mathbf{K}\lambda_j(\nabla p_j - \delta_j \nabla D)], \quad \ldots\ldots (3)$$

where $\mathbf{K}(\boldsymbol{\xi})$ is the effective permeability tensor; $\lambda_j(\boldsymbol{\omega})$ is phase mobility; $p_j(\boldsymbol{\omega})$ is phase pressure; $\delta_j(\boldsymbol{\omega})$ is the vertical pressure gradient; and $D(\boldsymbol{\xi})$ is the vertical-depth vector (up/down-oriented).

Eqs. 1 and 2 are approximated in space using the two-point flux approximation (TPFA) and in time using the fully implicit method. This introduces strong nonlinearity into the system of governing equations. Another effect to the nonlinearity of the problem is related to the closure assumption of instantaneous thermodynamic equilibrium. To satisfy this assumption, one needs to define the number of coexisting phases and compute the phase mole fractions x_{cp} and phase saturations s_p using the masses of components from the multiphase flash (Iranshahr et al. 2013). In our formulation, we use the overall composition of each component $z_c = \sum_p x_{cp}\rho_p s_p \Big/ \sum_p \rho_p s_p$ and pressure p as nonlinear unknowns following the overall molar-formulation approach (Collins et al. 1992; Voskov and Tchelepi 2012).

After the multiphase flash determines all related properties, one needs to linearize the problem, which requires the determination of all partial derivatives with respect to nonlinear unknowns and the assembly of the Jacobian and residuals. After the linearization step, we apply a version of the Newton-Raphson method, where the following linear system of equations should be solved at each nonlinear iteration:

$$\mathbf{J}(\boldsymbol{\omega}^k)(\boldsymbol{\omega}^{k+1} - \boldsymbol{\omega}^k) = -\mathbf{r}(\boldsymbol{\omega}^k), \quad \ldots\ldots (4)$$

where $\mathbf{J}(\boldsymbol{\omega}_k)$ and $\mathbf{r}(\boldsymbol{\omega}_k)$ are the Jacobian and residual defined at the nonlinear iteration k.

OBL Approach

In this section, we briefly describe the application of the OBL approach to the general-purpose thermal/compositional problem.

Transformation of Governing Equations. To apply the OBL approach, the discretized version of the governing equations, Eqs. 1 and 2, should be rewritten to represent each term as a product of state-dependent and space-dependent operators (Khait and Voskov 2016). In addition, we assume that the porosity is a pseudophysical-state variable ($\phi \in \boldsymbol{\omega}$).

The resulting mass-conservation equation, written for a control volume i, is

$$a(\boldsymbol{\xi})[\alpha_c(\boldsymbol{\omega}) - \alpha_c(\boldsymbol{\omega}_n)] + \sum_{j\in L(i)} b(\boldsymbol{\xi}, \boldsymbol{\omega})\beta_c(\boldsymbol{\omega}) + \theta_c(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u}) = 0, \quad c = 1, \ldots, n_c, \quad \ldots\ldots (5)$$

where

$$a(\boldsymbol{\xi}) = \phi_{0,i} V_i, \quad \ldots\ldots (6)$$

$$\alpha_c(\boldsymbol{\omega}) = [1 + c_r(p_i - p_{\mathrm{ref}})]\sum_{p=1}^{n_p} x_{cp,i}\rho_{p,i}s_{p,i}, \quad \ldots\ldots (7)$$

$$b(\boldsymbol{\xi}, \boldsymbol{\omega}) = \Delta t \Gamma_{ij}(p_j - p_i), \quad \ldots\ldots (8)$$

$$\beta_c(\boldsymbol{\omega}) = \sum_{p=1}^{n_p} x_{cp,ij}\rho_{p,ij}\lambda_{p,ij}, \quad \ldots\ldots (9)$$

where $x_{cp,ij}\rho_{p,ij}\lambda_{p,ij} = \begin{cases} x_{cp,i}\rho_{p,i}\lambda_{p,i} & \text{if } p_i > p_j \\ x_{cp,j}\rho_{p,j}\lambda_{p,j} & \text{otherwise} \end{cases}$.

In Eqs. 5 through 9, $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_n$ are nonlinear unknowns on the current and previous timestep, respectively; $L(i)$ is the set of neighbors of the control volume I; and $\theta_c(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u})$ is the influx/outflux term. In addition, $\phi_{0,i}$, V_i, and p_i, are initial porosity, volume, and pressure in the control volume i, respectively, while $x_{cp,i}$, $\rho_{p,i}$, and $\lambda_{p,i}$ are the component c mole fraction, density, and mobility of a phase p in the control volume i. In addition, c_r is rock compressibility, p_{ref} is the reference pressure, and Γ_{ij} is a constant geometrical part of transmissibility, combining permeabilities and geometries of neighboring control volumes i and j. For simplicity, we neglect both gravity and capillary forces here, which makes the phase potential differences equal across all phases at any interface. We will describe the extended OBL approach with buoyancy later in this paper.

The modified energy conservation becomes

$$a_e(\boldsymbol{\xi})[\alpha_e(\boldsymbol{\omega}) - \alpha_e(\boldsymbol{\omega}_n)] + \sum_{j \in L(i)} b_e(\boldsymbol{\xi}, \boldsymbol{\omega})\beta_e(\boldsymbol{\omega}) + \sum_{j \in L(i)} c_e(\boldsymbol{\xi}, \boldsymbol{\omega})\gamma_e(\boldsymbol{\omega}) + \theta_e(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u}) = 0, \quad (10)$$

where

$$a_e(\boldsymbol{\xi}) = V_i, \quad (11)$$

$$\alpha_e(\boldsymbol{\omega}) = \phi_i\left(\sum_{p=1}^{n_p} \rho_{p,i} s_{p,i} U_{p,i} - U_r\right) + U_r, \quad (12)$$

$$b_e(\boldsymbol{\xi}, \boldsymbol{\omega}) = b(\boldsymbol{\xi}, \boldsymbol{\omega}), \quad (13)$$

$$\beta_e(\boldsymbol{\omega}) = \sum_{p=1}^{n_p} h_{p,ij}\rho_{p,ij}\lambda_{p,ij}, \quad \text{where } h_{p,ij}\rho_{p,ij}\lambda_{p,ij} = \begin{cases} h_{p,i}\rho_{p,i}\lambda_{p,i} & \text{if } p_i > p_j \\ h_{p,j}\rho_{p,j}\lambda_{p,j} & \text{otherwise,} \end{cases} \quad (14)$$

$$c_e(\boldsymbol{\xi}) = \Delta t \Gamma_{ij}(T_j - T_i), \quad (15)$$

$$\gamma_e(\boldsymbol{\omega}) = \phi\left(\sum_{p=1}^{n_p} s_{p,ij}\kappa_{p,ij} - \kappa_r\right) + \kappa_r, \quad \text{where } s_{p,ij}\kappa_{p,ij} = \begin{cases} s_{p,i}\kappa_{p,i} & \text{if } T_i > T_j \\ s_{p,j}\kappa_{p,j} & \text{otherwise.} \end{cases} \quad (16)$$

In Eqs. 10 through 16, $\theta_e(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u})$ is energy influx/outflux; T_i is the temperature in the control volume i; and $U_{p,i}$, $h_{p,i}$, and $\kappa_{p,i}$ are internal energy, enthalpy, and thermal conduction of phase p in the control volume i, respectively. We also assume that rock internal energy U_r and rock thermal conduction κ_r are defined homogeneously for the whole reservoir.

This representation allows us to identify and distinguish the physical-state dependent operators—α_c, β_c, α_e, β_e, and γ_e—in both the mass-conservation (Eq. 1) and energy-conservation (Eq. 2) equations. The source/sink terms can also be processed in a similar manner, but in this work, we apply a conventional treatment for these operators.

Adaptive Parameterization and Coarsening. The proposed approach simplifies the description of fluid and rock properties by building approximation interpolants for the operators α_c, β_c, α_e, β_e, and γ_e within the parameter space of a simulation problem. For a general thermal/compositional problem with n_c components, the method requires $(2n_c + 3)$ operators, which depend on $N = (n_c + 1)$ independent variables $\{p, T, z_1, \ldots, z_{n_c-1}\}$. The values of the operators are fully determined by this set. The pressure and temperature ranges in the compositional parameter space can usually be determined by the conditions, assigned to wells, whereas the overall mole fraction is naturally bounded by the interval (0,1). As mentioned previously, we add the porosity as a pseudophysical-state variable with the corresponding range, as proposed by Khait and Voskov (2016).

Next, we parameterize the interval of each state variable using, for simplicity, the same number $n = n_1 = \ldots = n_N$ of uniformly distributed supporting points on the intervals of parameters. This results in a set of supporting points in N-dimensional space $\boldsymbol{\omega}^n = (p^{i_1}, T^{i_2}, z_1^{i_3}, \ldots, z_{n_c-1}^{i_{n_c+1}}, \phi^{i_{n_c+2}}) : i_1, \ldots, i_{n_c+2} = 1, \ldots, n$, which can be interpreted as a discrete representation of the parameter space in the simulation. At the preprocessing stage, or adaptively in the course of simulation, the operators $f = \{\alpha_c, \beta_c, \alpha_e, \beta_e, \gamma_e\}$ are evaluated at every supporting point in the discrete parameter space and stored in multidimensional tables $F^n = \{A_c^n, B_c^n, A_e^n, B_e^n, \Gamma_e^n\}$. During the simulation, we interpolate the values and evaluate the partial derivatives of all state-dependent operators f for each gridblock, using supporting points in created tables.

$$f(\boldsymbol{\omega}) = I(\boldsymbol{\omega}, F^n). \quad (17)$$

The details of the partial-derivative evaluation can be found in Khait and Voskov (2016).

This provides a continuous description taken from the interpolation operator, the accuracy of which is controlled by the resolution of the discretization in parameter space. For an isothermal compositional problem, the parameter space is defined by $\{p, z_1, \ldots, z_{n_c-1}\}$, while in thermal simulation, an extended set $\{p, T, \phi, z_1, \ldots, z_{n_c-1}\}$ is used. This representation significantly simplifies the implementation of general-purpose simulation frameworks. Instead of keeping track of each property and its derivatives with respect to nonlinear unknowns, we can construct an algebraic system of equations with abstract algebraic operators representing the complex physics. The performance of this formulation benefits from the fact that all expensive evaluations can be performed using a limited number of supporting points. Finally, the performance of the nonlinear solver can be improved because the Jacobian is constructed using a combination of piecewise linear operators, directly dependent on the nonlinear unknowns, as shown in Voskov (2017). Note that this approach is different from the numerical derivatives often used in reservoir simulation (CMG 2009; Xu et al. 2011) because the nonlinear physics are fully defined by interpolated properties $f = \{\alpha_c, \beta_c, \alpha_e, \beta_e, \gamma_e\}$ and are consistent with their derivatives.

The total size of the interpolation tables is defined by the number of dimensions N and the number of interpolation points n as n^N. Although the dimensionality depends on the number of components and thermal assumptions in a problem of interest, the number of interpolation points corresponds to the desired accuracy of the physical representation. Therefore, parameterization at the preprocessing stage would require a substantial amount of memory for the multicomponent systems modeled at a high interpolation precision. Furthermore, the necessity of searching supporting points (i.e., operator values) for every interpolation in a huge array of data affects the performance of the simulation. Notice that because of the hyperbolic nature of some variables (e.g., overall compositions), the vast majority of parameter space remains unused (Voskov and Tchelepi 2009; Zaydullin et al. 2013).

The adaptive parameterization approach avoids these disadvantages by removing the need for the entire preprocessing stage (Zaydullin et al. 2013). In this approach, supporting points are computed only when they are required by the current physical state of a control volume. The obtained operator values are then used in the interpolation process and stored for future use. Consequently, the method adds a new supporting point and computes appropriate operators, if the supporting point was not evaluated before, as shown in **Fig. 1.** On the left, an example of 2D parameter space is shown at the moment, when simulation occupies Rectangle 2, whereas Rectangle 1 was already used. Each rectangle has four vertices (for n-dimensional space, there will be hyperrectangles, or n-orthotopes, with 2^n vertices each), depicted as colored circles. They represent supporting points (i.e., operator values) required to perform interpolation within the rectangle. Because rectangles share vertices, and the simulation process is likely to spread continuously over parameter space, in the most cases some operator values can be reused.

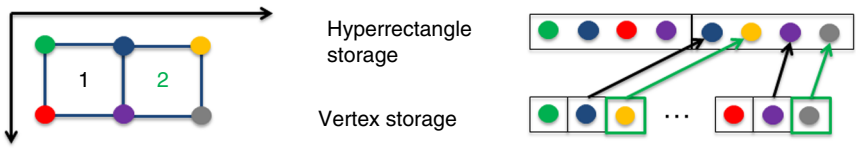


Fig. 1—Representation of adaptive OBL storage.

The current implementation of adaptive parameterization includes two storages—hyperrectangle and vertex—which are associative containers of key-value(s) pairs with unique hash-based keys. This choice was made to ensure rapid data access for high-dimensional cases. In the hyperrectangle storage, all vertices of each occupied hyperrectangle are kept together to maximize interpolation performance. The vertex storage is used when a new hyperrectangle is requested by simulation process. If the new hyperrectangle shares some vertices with already-visited hyperrectangles, then those vertices will be simply copied to the first storage, as shown by black arrows on the right in Fig. 1. Missing supporting points will be calculated and added to both storages (shown with green arrows). This approach is crucial in high-dimensional cases when each vertex is shared among many hyperrectangles.

In the end of the simulation, the resulting sparse multidimensional table of stored operators represents an actual subspace of physical parameters being used in the process. For example, **Fig. 2** shows an adaptive parameterization in the parameter space for a black-oil simulation at two different timesteps (at the beginning and the end of simulation). The adaptive approach reproduces the exact numerical results of the preprocessing method used in Khait and Voskov (2016) with greatly improved overall performance, especially for multicomponent systems.

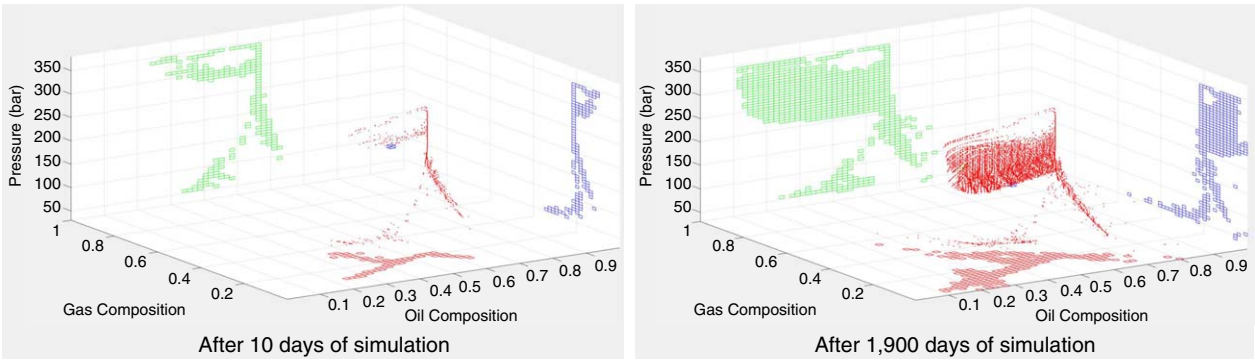


Fig. 2—Adaptive parameterization in the OBL approach for black-oil physical kernel: after 10 days of simulation; after 1,900 days of simulation.

The Performance of the OBL Approach

In this section, we present the results of modeling with the OBL approach, implemented in the Automatic Differentiation General Purpose Research Simulator (ADGPRS) (Voskov and Tchelepi 2012; Zaydullin et al. 2014). A performance study and an error analysis are provided for different resolutions of the physical parameter space, using the results of the conventional approach as a reference solution. The improvement in the performance of OBL-based simulations is achieved by a smaller number of nonlinear iterations, the absence of iterative phase-behavior computations in OBL method, and avoidance of derivative computations in the Automatic Differentiation Expression Templates Library (ADETL) (Younis 2011), which is the underlying automatic differentiation library used by ADGPRS for construction and assembly of the Jacobian. However, the necessity of artificial-data injection (values and derivatives obtained by interpolation) back to automatic-differentiation-based data structures negatively affects the performance. That can be avoided if a standalone simulator is implemented entirely from the OBL perspective, as shown by Khait et al. (2017)

For all simulations, we used the same discretization operators (Eqs. 6 and 8) derived from the seventh layer of the SPE10 model (Christie and Blunt 2001) shown in **Fig. 3.** An injection well is placed in the middle of the reservoir, with four producers set at the corners. We applied TPFA discretization and coupled this model with different physical kernels to demonstrate the applicability of the OBL method for a general-purpose simulation.

Isothermal Black-Oil Kernel. Here, we used a standard black-oil model, where only the gas component can dissolve in the oil phase and most of the properties are described as table-based correlations. The water-injection well operated with bottomhole-pressure (BHP) control at a pressure $P_i = 350$ bar, and the producer well operated at $P_p = 270$ bar for the first 2,000 days and then switched to $P_p = 170$ bar for the rest of simulation. The reservoir was initialized uniformly with pressure $P_0 = 300$ bar, water saturation $S_w = 0.2$, gas saturation $S_g = 0$, and bubblepoint pressure $P_{\text{bub}} = 270$ bar. All simulations were run for 6,000 days with a maximum timestep of $\Delta t = 10$

days, which corresponds to average Courant–Friedrichs–Lewy (CFL) condition = 5.3. The pressure/volume/temperature properties and relative permeabilities were used from the SPE9 test case (Killough 1995).

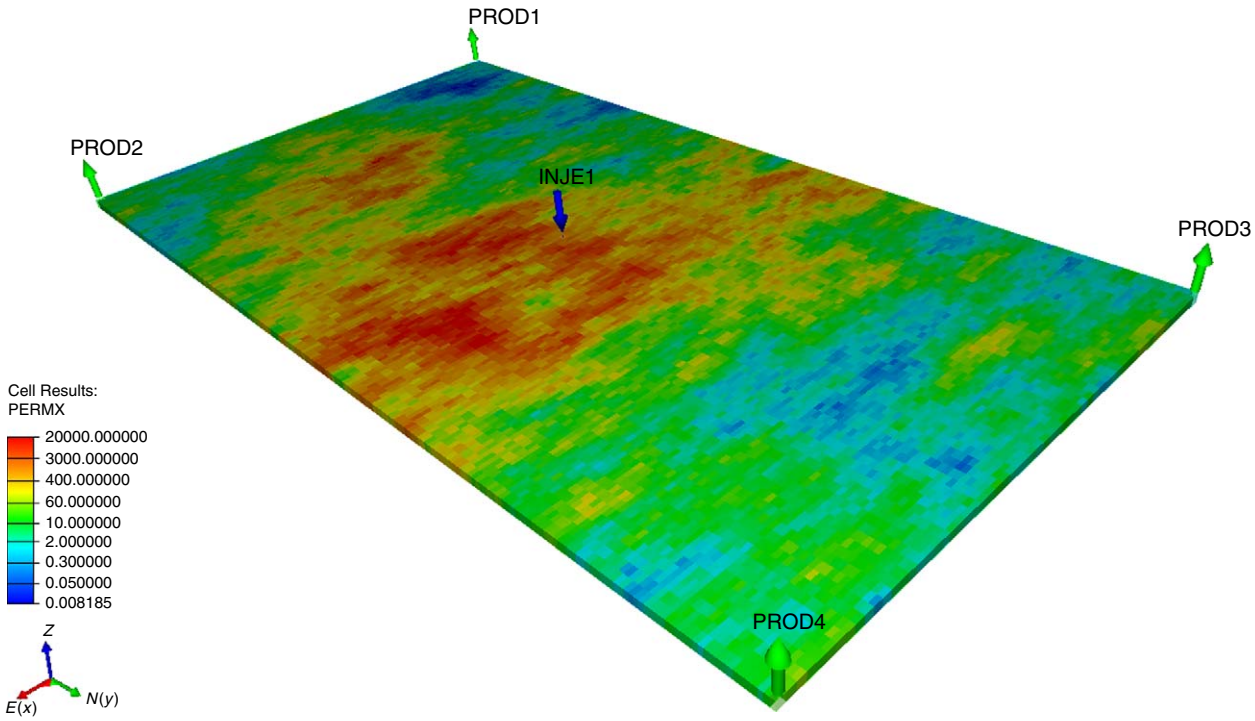


Fig. 3—Reservoir-permeability map used for all simulations.

The obtained performance results are shown in **Table 1.** The resolution of parameter space, defined by the number of interpolation points n, is shown in the first column. The total number of nonlinear iterations for each test case is presented in the second column. The next two columns show the error in pressure and compositions (average for all components). The next column shows the percentage of points used for the adaptive parameterization of parameter space by the OBL approach. The next column reflects the central-processing-unit (CPU) time required for a serial run on an Intel Xeon E5-1620 and 3.5-GHz processor. Finally, the last two columns show the percentage of CPU time spent on generation and interpolation of all operators, respectively.

Resolution	Iterations	E_p (%)	E_z (%)	Space (%)	CPU (seconds)	Generation (%)	Interpolation (%)
Standard	6,404	–	–	–	1,217.2	–	–
64	4,206	1.12	2.60	1.7447	659.4	<0.1	19.7
32	3,544	1.60	3.18	3.8681	555.9	<0.1	19.7
16	3,303	1.69	4.09	9.3506	542.2	<0.1	18.8
8	2,916	2.26	7.53	22.5586	482.2	<0.1	18.8

Table 1—Results of black-oil simulation.

Table 1 demonstrates that a smaller number of interpolation points results in a simpler nonlinear system because it requires fewer Newton iterations to be solved. Note that the number of Newton iterations performed in the OBL method is significantly lower than that for the standard simulation. Based on this and other CPU improvements provided by the OBL approach (e.g., simplified Jacobian assemble), the corresponding CPU time is significantly reduced compared with the conventional approach implemented in ADGPRS. For a black-oil kernel, the generation stage is quite inexpensive and requires almost no extra time. The time spent on interpolation of operators is almost independent of the resolution of parameterization space, and represents the time spent for complete Jacobian and residual assembly, including property and derivative evaluation.

In this test, the overall composition of the water component introduces the largest error with respect to the remaining unknowns. Maps of water overall composition and distribution of errors (in %) after 6,000 days of the simulation are shown in **Fig. 4.** It is clear that the error is distributed near the displacement front and for the finest resolution ($n = 64$) is significantly lower than typical time- and space-truncation errors.

Isothermal Compositional Kernel. Next, we demonstrate an applicability of the OBL technique for an isothermal process of carbon dioxide (CO_2) and methane (CH_4) injection into an oil with composition from Orr et al. (1995). The initial oil was made of four components, CO_2, C_1, C_4, and C_{10}, at corresponding compositions: 1% of CO_2, 11% of CH_4, 38% of n-butane, and 50% of decane. We injected a mixture of 80% of CO_2 and 20% of C_1 at a BHP $P_i = 120$ bar. The production wells operated at BHP $P_p = 60$ bar. The initial pressure was $P_0 = 90$ bar and initial temperature was $T_0 = 80°C$. The simulation period was 4,000 days with a maximum timestep $\Delta t = 50$ days, which corresponds to the average CFL = 110. The description of phase behavior and properties dependent on the Peng and Robinson (1976) equation of state and the Lohrenz-Bray-Clark correlations for viscosity (Lohrenz et al. 1964) was used in this kernel.

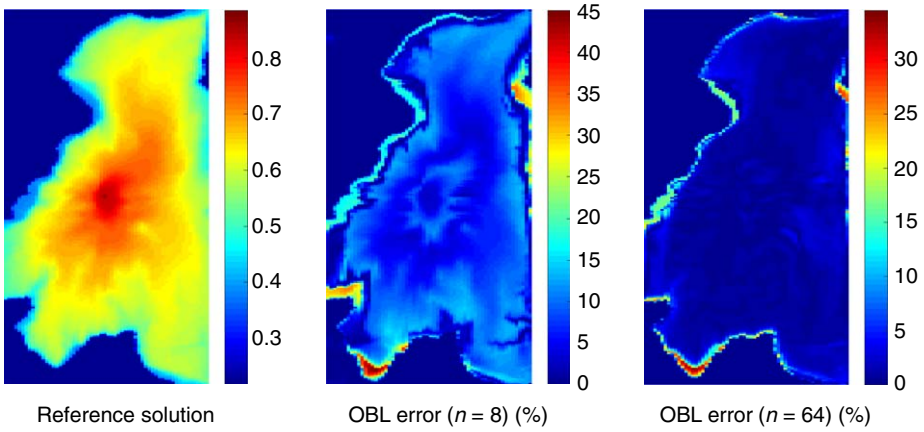


Fig. 4—Composition of water and error distribution for two OBL resolutions after 6,000 days of simulation: reference solution; OBL error ($n = 8$) (%); OBL error ($n = 64$) (%).

Table 2 shows the main results of the isothermal/compositional simulation. The difference in the number of Newton iterations between the standard and OBL simulations is fewer than in the previous case, but the trend is similar, with an exception for eight points where the number of nonlinear iterations is slightly larger than for the previous resolution. This reflects the fact that the location of interpolation-supporting points in the current version of the approach was blindly determined by uniform distribution without any analysis of nonlinearity.

Resolution	Iterations	E_p (%)	E_z (%)	Space (%)	CPU (seconds)	Generation (%)	Interpolation (%)
Standard	626	–	–	–	562.2	–	–
64	561	0.33	0.71	0.1262	152.6	11.6	20.3
32	531	0.34	1.00	0.4392	129.8	3.0	21.7
16	498	0.34	1.73	1.7006	119.5	0.8	22.0
8	509	0.54	3.96	7.4463	123.7	0.2	21.5

Table 2—Results of compositional (four-component) simulation.

At the same time, the performance of simulation with the OBL approach was improved even more significantly compared with the conventional simulation than it was for the black-oil kernel. It can be explained by more-expensive phase behavior, usually required for conventional compositional simulation, compared with black oil. Notice that these phase-behavior calculations are nearly completely absent in the OBL approach, which explains an additional CPU gain. On the other hand, the interpolation kernel still performs effectively (see the Generation and Interpolation columns of Table 2) because the dimension of parameter space is relatively small. In this test, the overall composition of the CO_2 component generates the largest error. The distribution of CO_2 composition and error maps (in %) after 2,000 days of the simulation are shown in **Fig. 5.** It is clear that the error is distributed near the displacement front and is significantly lower than typical time- and space-truncation error.

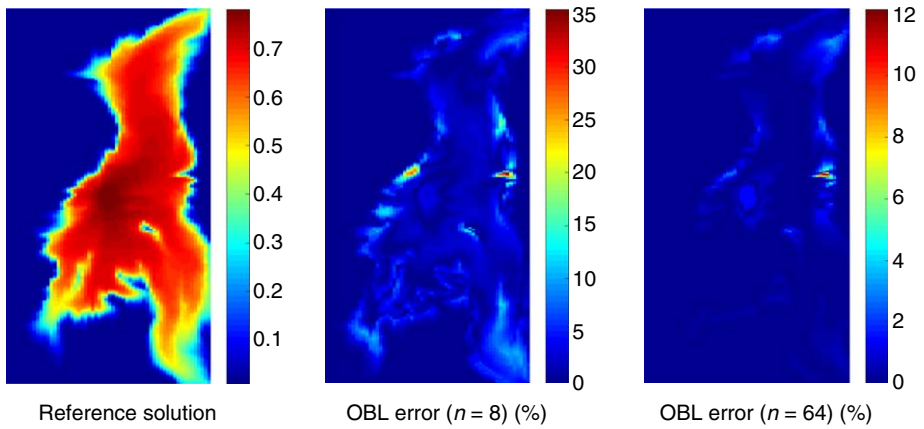


Fig. 5—Composition of CO_2 and error distribution for two OBL resolutions after 2,000 days of simulation: reference solution; OBL error ($n = 8$) (%); OBL error ($n = 64$) (%).

To estimate the performance of the OBL approach for a system with a larger number of components, we ran a similar simulation with six-component oil made of CO_2 (1%), C_1 (10%), C_2 (9%), C_3 (10%), C_4 (15%), and C_{10} (55%). The same mixture of CO_2 (80%) and C_1 (20%) was used as an injection stream and the same timestep $\Delta T = 50$ days, which corresponds to average CFL = 139, was used

in this case. The results of the simulation are presented in **Table 3.** Here, the performance of the OBL approach still improves compared with the conventional technique, but the speedup is lower. This is because the performance of the OBL approach is directly dependent on the performance of a piecewise multilinear interpolation, which becomes more expensive in the case of high-dimension parameter space, as discussed in Weiser and Zarantonello (1988).

Resolution	Iterations	E_p (%)	E_z (%)	Space (%)	CPU (seconds)	Generation (%)	Interpolation (%)
Standard	577	–	–	–	829.5	–	–
64	466	0.31	0.91	0.0001	393.6	44.0	18.5
32	448	0.31	1.51	0.0017	249.2	15.0	27.4
16	431	0.35	3.02	0.0280	213.8	4.4	30.5
8	416	0.70	7.53	0.4761	202.6	1.0	31.2

Table 3—Results of compositional (six-component) simulation.

In Table 3, both the generation and interpolation times significantly increase compared with the previous (four-component) simulations. Here, the generation of operator tables becomes the slowest procedure for a high-resolution case because of the larger dimensionality of parameterization space. In this case, it is more convenient to switch to the barycentric interpolation, which requires fewer supporting points and was fully used in Zaydullin et al. (2013) for compositional simulation using tie-line space parameterization. Another possibility is to improve the generation stage by optimizing flash calculations (Haugen and Beckner 2013). Still, the most expensive high-resolution OBL case performs more than two times faster than the conventional compositional approach implemented in ADGPRS. The error distribution in this case is similar to that of a four-component-test case.

Thermal/Compositional Kernel. The next simulation model is related to a problem of a CH_4 coproduction from geothermal reservoirs. This problem has a thermal/compositional kernel with only two components, but it corresponds to the most nonlinear physics among all examples. We used the same reservoir for this model and consider an initial composition of water (80 mol%) and CH_4 (20 mol%) at saturated conditions ($p = 60$ bar, $T = 210°C$) close to the superheated regime. The well in the middle injects cold water ($T = 35°C$), triggering the propagation of a cold condensation front to the production wells at the corners. The simulation period was 600 days with a maximum timestep of $\Delta t = 5$ days, which corresponds to an average CFL = 22.2.

The convergence results of the thermal/compositional simulation are presented in **Table 4.** Here, the improvement in the nonlinear behavior (i.e., the number of iterations) is less significant than in previous simulations. It can be understood as a consequence of a more-nonlinear nature of the physical kernel. A similar observation is valid in relation to the errors between approximate and reference solutions. However, the improvements in the CPU performance are still significant compared with the default ADGPRS performance because of the absence of phase-behavior computations and the improved Jacobian assembled in the OBL method. The water distribution and error map are presented in **Fig. 6.**

Resolution	Iterations	E_p (%)	E_T (%)	E_z (%)	Space (%)	CPU (seconds)	Generation (%)	Interpolation (%)
Standard	808	–	–	–	–	429.2	–	–
64	774	0.06	0.14	0.41	2.5978	279.9	0.8	12.9
32	762	0.32	0.64	1.33	4.6994	268.9	0.1	12.6
16	741	0.71	1.34	4.49	9.2095	263.5	<0.1	12.4
8	753	1.10	2.19	7.90	21.8750	265.6	<0.1	12.3

Table 4—Results of thermal/compositional simulation.

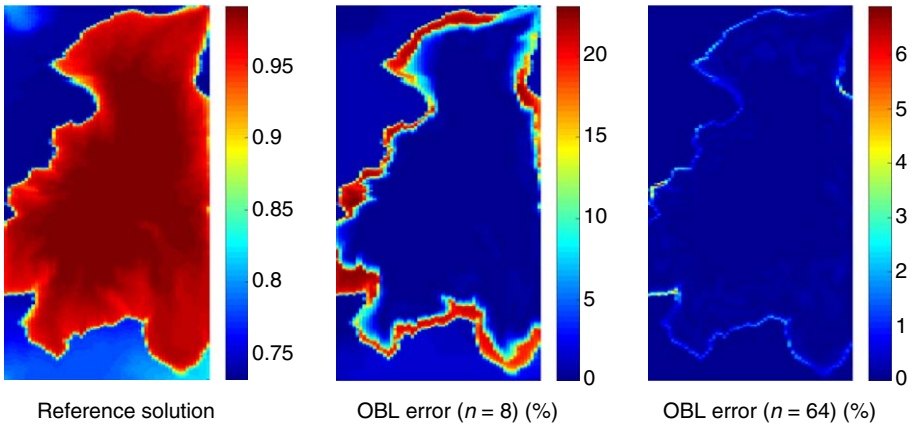


Fig. 6—Composition of water and error distribution for two OBL resolutions after 200 days of simulation: reference solution; OBL error ($n = 8$) (%); OBL error ($n = 64$) (%).

OBL With Buoyancy

In the conventional modeling approach, the introduction of buoyancy in multiphase flux calculations assumes that each phase has its own phase potential difference at a given interface *ij*, allowing countercurrent flow:

$$\Phi_{p,ij} = [p_j - p_i - \delta_{p,ij}(D_j - D_i)], \quad \ldots (18)$$

$$\delta_{p,ij} = \begin{cases} \dfrac{\delta_{p,i} + \delta_{p,j}}{2}, & \text{if phase } p \text{ appears in both cell } i \text{ and } j \\ \delta_{p,i}, & \text{if phase } p \text{ appears only in cell } i \\ \delta_{p,j}, & \text{if phase } p \text{ appears only in cell } j \\ 0, & \text{if phase } p \text{ does not exist in either cell } i \text{ or } j, \end{cases} \quad \ldots (19)$$

where $\delta_{p,i} = \delta_p(\boldsymbol{\omega}_i) = g\rho^m_{p,i}$, g is gravitational acceleration, and $\rho^m_{p,i}$ is the mass density of a phase p in a control volume i. Numerical fluxes are usually computed using the phase-potential-upwinding (PPU) strategy, in which phase mobilities are selected depending on a sign of the corresponding phase potential difference separately for each phase.

Straightforward implementation of PPU within the OBL approach implies the increase in the number of flux operators from n_c to $n_c n_p$, because phases should be treated separately. In addition, a mass-density operator δ_p has to be introduced for each phase. Striving to reduce the amount of required interpolations, we evaluate a single mass-density value per each phase for united control volume of adjacent blocks, instead of averaging of two values obtained for each of the blocks. Hence, Eq. 19 becomes

$$\delta_{p,ij} = \delta_p(\boldsymbol{\omega}_{ij}), \quad \ldots (20)$$

$$\boldsymbol{\omega}_{ij} = \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_j}{2}. \quad \ldots (21)$$

Considering buoyancy, Eq. 5 is transformed into

$$a(\boldsymbol{\xi})[\alpha_c(\boldsymbol{\omega}) - \alpha_c(\boldsymbol{\omega}_n)] + \sum_{j \in L(i)} \sum_p b_p(\boldsymbol{\xi}, \boldsymbol{\omega}) \beta_{cp}(\boldsymbol{\omega}) + \theta_c(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u}) = 0, \ \ c = 1, \ldots, n_c, \quad \ldots (22)$$

where

$$b_p(\boldsymbol{\xi}, \boldsymbol{\omega}) = \Delta t \Gamma_{ij} \Phi_{p,ij}, \quad \ldots (23)$$

$$\beta_{cp}(\boldsymbol{\omega}) = x_{cp,ij} \rho_{p,ij} \lambda_{p,ij}, \quad \ldots (24)$$

$$\text{where } x_{cp,ij}\rho_{p,ij}\lambda_{p,ij} = \begin{cases} x_{cp,i}\rho_{p,i}\lambda_{p,i} & \text{if } \Phi_{p,ij} < 0 \\ x_{cp,j}\rho_{p,j}\lambda_{p,j} & \text{otherwise.} \end{cases} \quad \ldots (25)$$

Similarly, the energy-balance equation (Eq. 10) becomes

$$a_e(\boldsymbol{\xi})[\alpha_e(\boldsymbol{\omega}) - \alpha_e(\boldsymbol{\omega}_n)] + \sum_{j \in L(i)} \sum_p b_{ep}(\boldsymbol{\xi}, \boldsymbol{\omega}) \beta_{ep}(\boldsymbol{\omega}) + \sum_l c_e(\boldsymbol{\xi}, \boldsymbol{\omega}) \gamma_e(\boldsymbol{\omega}) + \theta_e(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u}) = 0, \quad \ldots (26)$$

where

$$b_{ep}(\boldsymbol{\xi}, \boldsymbol{\omega}) = b_p(\boldsymbol{\xi}, \boldsymbol{\omega}), \quad \ldots (27)$$

$$\beta_{ep}(\boldsymbol{\omega}) = h_{p,ij} \rho_{p,ij} \lambda_{p,ij}, \quad \ldots (28)$$

$$\text{where } h_{p,ij}\rho_{p,ij}\lambda_{p,ij} = \begin{cases} h_{p,i}\rho_{p,i}\lambda_{p,i} & \text{if } \Phi_{p,ij} < 0 \\ h_{p,j}\rho_{p,j}\lambda_{p,j} & \text{otherwise.} \end{cases} \quad \ldots (29)$$

Therefore, OBL with PPU requires $n_c + n_c n_p + n_p$ operators for the isothermal problem with n_c components and n_p phases, and $n_c + n_c n_p + 2n_p + 2$ for the nonisothermal problem.

Results of OBL With Buoyancy

1D Dead-Oil Model With Gravity Segregation. We started with a simple 1D domain for a vertical segregation model with buoyancy-driven flow. It consists of five grid cells that are $10 \times 10 \times 10$ m each, extending over a depth of 50 m overall. The rock permeability is equal to 100 md, and the porosity is equal to 0.2. We started with dead-oil kernel, filling the top three grid cells with water (with a constant density $\rho^m_w = 1000$ kg/m^3), whereas the bottom two cells were filled by oil (with a constant density $\rho^m_o = 800$ kg/m^3). The reservoir was initialized with $P_0 = 100$ bar. We ran the model for 10,000 days until the system reached an equilibrium. The dynamic distribution of fluids is shown in **Fig. 7.** It can be seen that the heavier-water phase, placed on top, exchanges position with the oil phase by the end of the simulation time.

In **Fig. 8a,** the error between the reference and OBL simulations is shown depending on parameterization resolution. It is clear that the error is converging to zero when at high OBL resolution. Next, Fig. 8b demonstrates the dynamic cumulative number of nonlinear iterations vs. time for two OBL resolutions of 16 and 100 points. The plot covers only the first 1,800 days, because later the system is close to equilibrium and requires a single Newton iteration to converge for all simulations. The lower resolution demonstrates better convergence in the beginning of the simulation and requires more iterations by its end. The finer-resolution model behaves similarly to the model with the reference physics.

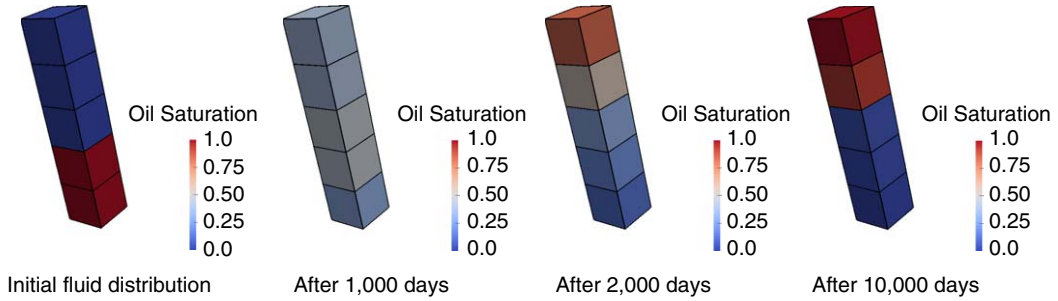
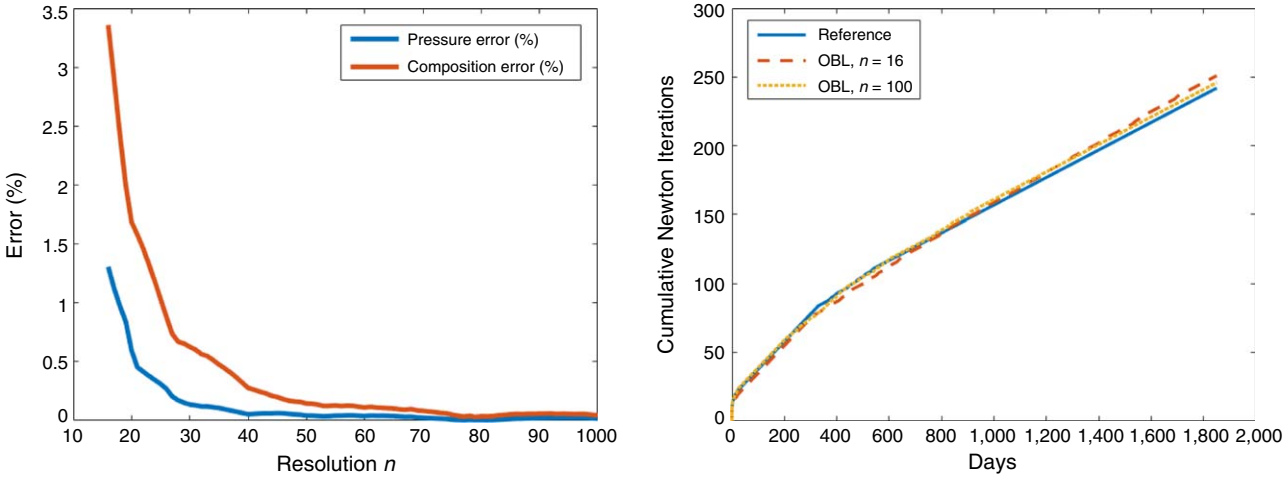


Fig. 7—Dead-oil gravity segregation: initial fluid distribution; after 1,000 days; after 2,000 days; after 10,000 days.



(a) Final solution error of OBL compared with reference model

(b) Comparison of nonlinear solver behavior

Fig. 8—OBL behavior for dead-oil kernel. (a) Final solution error of OBL compared with reference model. (b) Comparison of nonlinear solver behavior.

1D Compositional Model With Gravity Segregation. Next, we ran the gravity-segregation test with the four-component isothermal/compositional model used previously. The initial oil with 1% of CO_2, 11% of C_1, 38% of n-C_4, and 50% of C_{10} was placed in the top three grid cells, while the mixture of gas with 80% of CO_2 and 20% of C_1 was placed in the two bottom cells. The initial reservoir pressure was set to $P_0 = 120$ bar and initial temperature was set to $T_0 = 350$ K, forming a pure-liquid phase in the top and a pure-gas phase in the bottom. The dynamic distribution of phases is shown in **Fig. 9.** Unlike in the dead-oil kernel, the gravity segregation here is combined with extensive mass exchange between liquid and vapor phases, which dramatically changes the composition of fluids.

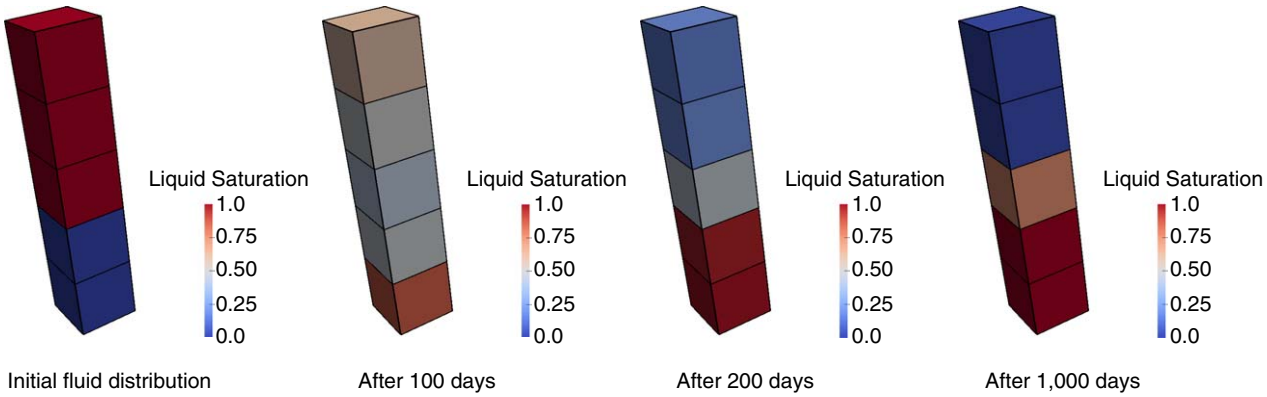
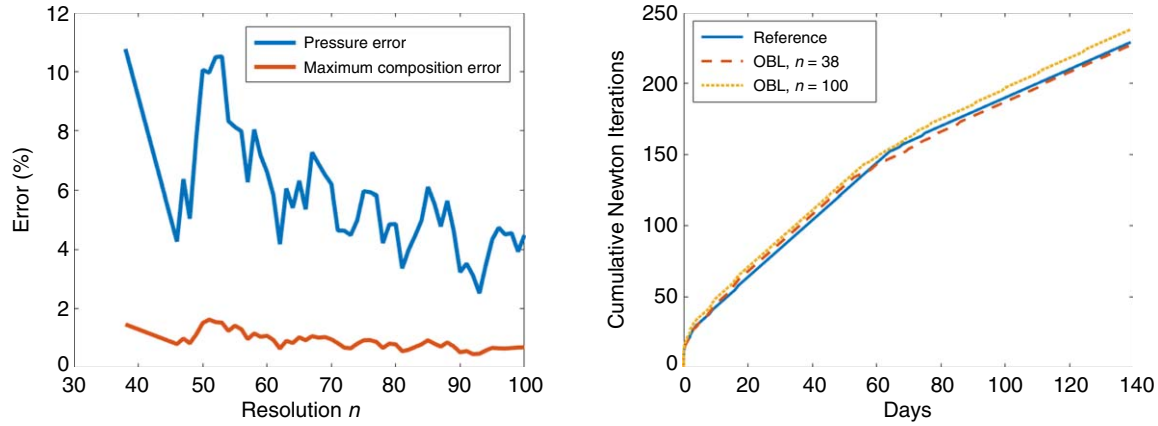


Fig. 9—Compositional gravity segregation: initial fluid distribution; after 100 days; after 200 days; after 1,000 days.

Fig. 10a shows the corresponding error in pressure and composition (maximum over all components) final solutions of the OBL model, compared with reference physics, depending on the resolution of the OBL approach. It is clear that the error is converging slower than in the dead-oil kernel because of a more-complicated process dynamic. At the same time, the nonlinear performance (Fig. 10b) behaves more predictably, being slightly better for the low-resolution and slightly worse for the high-resolution OBL simulation. Again, the plot covers only the most intense first 140 days, because later the nonlinear behavior of all simulations is equal.



(a) Final solution error of OBL compared with reference model

(b) Comparison of nonlinear solver behavior

Fig. 10—OBL error behavior for compositional kernel. (a) Final solution error of OBL compared with reference model. (b) Comparison of nonlinear solver behavior.

Full-Field Brugge Model. To demonstrate the applicability of the OBL approach for a full-field model, we use the Brugge Field, which is often used as the optimization benchmark for reservoir-simulation study (Peters et al. 2010). This model is derived from realistic reservoir structures and properties shown in **Fig. 11.** The simulation time spans 10 years, with BHP controls changing every 3 months for both injection and production wells. For reservoir parameters and well controls, we used the base-case realization described in Bukshtynov et al. (2015).

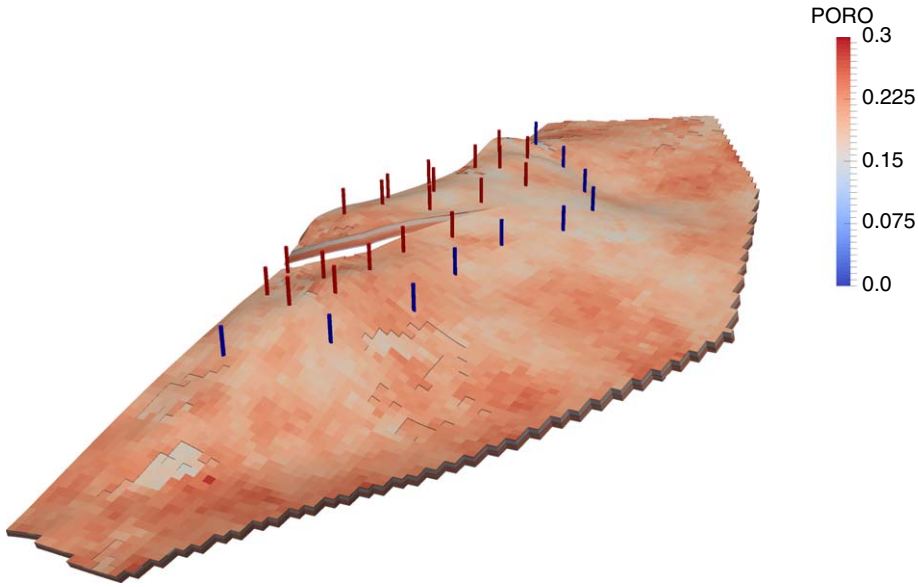


Fig. 11—Porosity distribution of the Brugge Field.

In **Fig. 12,** we compare total well rates for both production and injection wells for the reference dead-oil kernel and the OBL implementation (with the resolution $n = 64$) with and without gravity. It can be seen that the buoyant forces play an important role in this model, and only the simulation using OBL with buoyancy successfully recovers reference-well rates.

The corresponding errors and nonlinear behavior are shown in **Fig. 13.** It can be seen that the overall error is quite insignificant, even for the coarsest resolution ($n = 8$). The error stabilizes at $n = 32$ and remains stable up to the finest resolution $n = 96$. The nonlinear behavior is nearly equivalent between the reference solution and OBL at different resolutions in the first half of the simulation. Afterward, OBL-based simulations require slightly more nonlinear iterations to converge. The CPU cost of simulation with the reference physics is comparable with the OBL approach without buoyancy (592 vs. 582 seconds, respectively, whereas linearization takes approximately 260 seconds in both). The corresponding number of operators in the OBL approach with gravity grows significantly (from two to six for dead-oil kernel) increasing the overhead because of ADETL. That can explain why the simulation time for the OBL approach with buoyancy is larger (793 seconds with linearization cost of 470 seconds). As mentioned previously, the proper implementation of the OBL approach can speed up Jacobian assembly by a factor of 14. Moreover, the migration and optimization of algorithms for emerging architectures (e.g., graphics processing unit or GPU) improve the linearization performance by another order of magnitude (Khait et al. 2017).

Conclusion and Discussion

In this work, we studied the applicability of a new linearization approach for petroleum-related simulation. In our approach, the governing equations of a general-purpose reservoir-simulation problem are represented in the operator form, where each term in the governing equations is a product of two operators. The first type of operator is fully defined by the physical state of the problem, while the second

is characterized by spatial and temporal discretization. To perform the linearization of the governing equations, we introduced a uniform parameterization in the space of physical unknowns, where each state-dependent operator is evaluated and stored at every supporting point of the parameterization. An adaptive parameterization in the discretized parameter space was applied to improve the performance of the parameterization procedure. The operators responsible for spatial and temporal discretization were involved in a conventional manner using TPFA approximation for flux and fully implicit approximation for time.

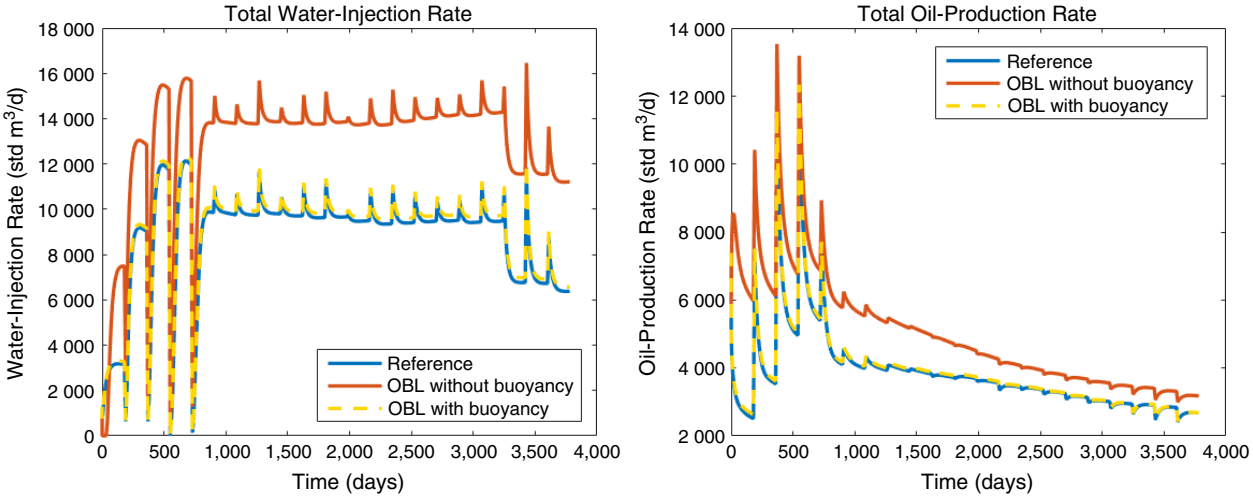


Fig. 12—Well-rate comparison for Brugge Field.

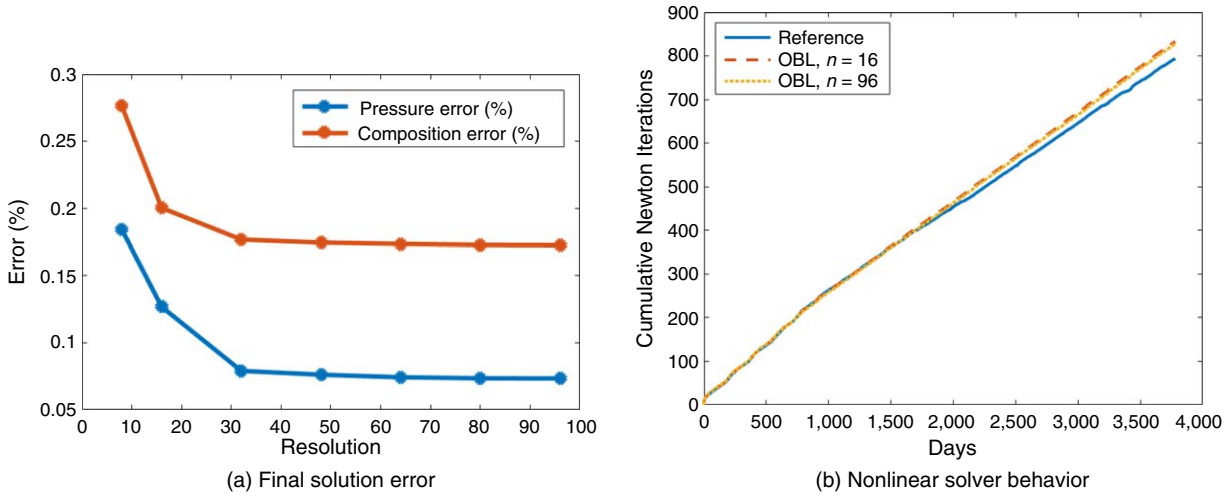


Fig. 13—OBL error behavior for Brugge Field model: final solution error; nonlinear solver behavior.

Next, we demonstrated the applicability of the OBL approach to the general-purpose reservoir-simulation problems. In particular, we applied different physical kernels that include black-oil, isothermal/compositional, and thermal/compositional kernels with a different number of components. We showed that the OBL approach reproduces the results of the reference solution at any reasonable resolution with insignificant errors, localized at displacement fronts. On the other hand, the limited coarsening of parameter space improves the nonlinear solution in most cases. The performance of the OBL approach benefits from the simplified assembly of the Jacobian of the simulation problem, the nearly complete bypass of phase-behavior calculations (except supporting points), and the better resolution of nonlinear physics. The new approach provides almost unlimited flexibility for the extension of the physical model in a simulation framework without the overhead related to the application of the automatic-differentiation technique. The method also constitutes a promising option for implementation of a full nonlinear loop on emerging architectures, for example on GPU, as shown by Khait et al. (2017).

Finally, we described the extension of the OBL approach for multiphase flow and transport with buoyancy. We introduced a phase-based treatment of the buoyancy term and parameterized it in the physics space of a problem. We demonstrated the convergence of this scheme on both simplified segregation problems and a realistic 3D reservoir (Brugge Field). In the current realization of the OBL approach, the nonlinear behavior does not benefit directly from the piecewise linear approximation of the state operators. However, with the specifically designed nonlinear solvers and more-efficient implementation of the numerical framework, we expect that the OBL approach can improve the performance time quite significantly.

Our future work will include a development of the more-robust component-based strategy for an approximation of the gravity flux. This scheme will potentially reduce the number of operators in the OBL approach and improve the nonlinear convergence. In addition, we will develop an extension of the component-based OBL approach to the systems with capillary-dominated flow. These systems usually introduce a significant challenge to the standard Newton-based nonlinear solver and require more-robust nonlinear strategy. That defines another direction of our research, where we will develop advanced nonlinear solvers using the trust region in the parameter space.

Nomenclature

I = interpolation operator
J = Jacobian
n_c = number of components
n_p = number of phases
r = residual
$\mathbf{u}$ = control properties
Γ_{ij} = geometrical part of transmissibility
$\boldsymbol{\xi}$ = spatial properties
$\boldsymbol{\omega}$ = physical properties

Linearization Operators

a = spatial term of mass-accumulation operator
a_e = spatial term of energy-accumulation operator
b = spatial term of mass-convection operator
b_e = spatial term of energy-convection operator
c_e = spatial term of energy-conductive operator
α_c = physical term of mass-accumulation operator
α_e = physical term of energy-accumulation operator
β_c = physical term of mass-convection operator
β_e = physical term of energy-convection operator
γ_e = physical term of energy-conductive operator
δ = mass density operator
θ_c = mass influx/outflux term
θ_e = energy influx/outflux term

Physics Relations

D = vertical-depth vector (up/down-oriented)
h_p = phase enthalpy
p_j = phase pressure
$\tilde{q}_p$ = phase in/outflux
s_p = phase saturation
$\overrightarrow{u_p}$ = phase velocity
U_p = phase internal energy
U_r = rock internal energy
x_{cp} = component mole fraction in a phase
δ_j = vertical pressure gradient
κ = thermal conduction
$\mathbf{K}$ = effective permeability tensor
λ_j = phase mobility
ρ_p = phase molar density
ϕ = effective rock porosity

Acknowledgments

We acknowledge Delft University of Technology for financial support. We also thank the Stanford Reservoir Simulation Research Program, SUPRI-B, for permission to use ADGPRS in this research.

References

Acs, G., Doleshall, S., and Farkas, E. 1985. General Purpose Compositional Model. *SPE J.* **25** (4): 543–553. SPE-10515-PA. https://doi.org/10.2118/10515-PA.

Aziz, K. and Settari, T. 1979. *Petroleum Reservoir Simulation*. London: Applied Science Publishers.

Aziz, K. and Wong, T. 1989. Considerations in the Development of Multipurpose Reservoir Simulation Models. Oral presentation given at the 1st and 2nd International Forum on Reservoir Simulation, Alpbach, Austria, 4–8 September.

Bukshtynov, V., Volkov, O., Durlofsky, L. et al. 2015. Comprehensive Framework for Gradient-Based Optimization in Closed-Loop Reservoir Management. *Computat. Geosci.* **19** (4): 877–897. https://doi.org/10.1007/s10596-015-9496-5.

Cao, H., Zaydullin, R., and Obi, E. 2017. Nonlinear Convergence for Near-Miscible Problem: A Mystery Unveiled for Natural Variable Simulator. Presented at the SPE Reservoir Simulation Conference, Montgomery, Texas, 20–22 February. SPE-182633-MS. https://doi.org/10.2118/182633-MS.

Christie, M. and Blunt, M. 2001. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. *SPE Res Eval & Eng* **4** (4): 308–317. SPE-72469-PA. https://doi.org/10.2118/72469-PA.

Coats, K. H. 1980. An Equation of State Compositional Model. *SPE J.* **20** (5): 363–376. SPE-8284-PA. https://doi.org/10.2118/8284-PA.

Collins, D., Nghiem, L. X., Li, Y.-K. et al. 1992. An Efficient Approach to Adaptive- Implicit Compositional Simulation With an Equation of State. *SPE J.* **7** (2): 259–264. SPE-15133-PA. https://doi.org/10.2118/15133-PA.

Computer Modelling Group (CMG). 2009. STARS User Guide. Calgary: CMG.

Haugen, K. and Beckner, B. 2013. Highly Optimized Phase Equilibrium Calculations. Presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 18–20 February. SPE-163583-MS. https://doi.org/10.2118/163583-MS.

Haugen, K. B. and Beckner, B. L. 2015. A General Flow Equation Framework. Presented at the SPE Reservoir Simulation Symposium, Houston, 23–25 February. SPE-173266-MS. https://doi.org/10.2118/173266-MS.

Iranshahr, A., Voskov, D., and Tchelepi, H. 2013. Tie-Simplex Based Compositional Space Parameterization: Continuity and Generalization to Multiphase Systems. *AIChE J.* **59** (5):1684–1701. https://doi.org/10.1002/aic.13919.

Jenny, P., Tchelepi, H. A., and Lee, S. H. 2009. Unconditionally Convergent Nonlinear Solver for Hyperbolic Conservation Laws with S-Shaped Flux Functions. *J. Comput. Phys.* **228** (20): 7497–7512. https://doi.org/10.1016/j.jcp.2009.06.032.

Khait, M. and Voskov, D. 2016. Operator-Based Linearization for Non-Isothermal Multiphase Compositional Flow in Porous Media. Oral presentation given at ECMOR XIV–15th European Conference on the Mathematics of Oil Recovery, Amsterdam, 29 August–1 September.

Khait, M., Voskov, D. et al. 2017. GPU-Offloaded General Purpose Simulator for Multiphase Flow in Porous Media. Presented at the SPE Reservoir Simulation Conference, Montgomery, Texas, 20–22 February. SPE-182663-MS. https://doi.org/10.2118/182663-MS.

Killough, J. E. 1995. Ninth SPE Comparative Solution Project: A Reexamination of Black-Oil Simulation. Presented at the SPE Reservoir Simulation Symposium, San Antonio, Texas, 12–15 February. SPE-29110-MS. https://doi.org/10.2118/29110-MS.

Lohrenz, J., Bray, B., and Clark, C. 1964. Calculating Viscosities of Reservoir Fluids From Their Compositions. *J Pet Technol* **16** (10): 1171–1176. SPE-915-PA. https://doi.org/10.2118/915-PA.

Orr, F. Jr., Dindoruk, B., and Johns, R. 1995. Theory of Multicomponent Gas/Oil Displacements. *Ind. Eng. Chem. Res.* **34** (8): 2661–2669. https://doi.org/10.1021/ie00047a015.

Peng, D.-Y. and Robinson, D. B. 1976. A New Two-Constant Equation of State. *Ind. Eng. Chem. Fundamen.* **15** (1): 59–64. https://doi.org/10.1021/i160057a011.

Peters, E., Arts, R., Brouwer, G. et al. 2010. Results of the Brugge Benchmark Study for Flooding Optimization and History Matching. *SPE Res Eval & Eng* **13** (3): 391–405. SPE-119094-PA. https://doi.org/10.2118/119094-PA.

Rasmussen, C., Krejbjerg, K., Michelsen, M. L. et al. 2006. Increasing Computational Speed of Flash Calculations With Applications for Compositional, Transient Simulations. *SPE Res Eval & Eng* **9** (1): 32–38. SPE-84181-PA. https://doi.org/10.2118/84181-PA.

Voskov, D. V. 2012. An Extended Natural Variable Formulation for Compositional Simulation Based on Tie-Line Parameterization. *Transport Porous Med.* **92** (3): 541–557. https://doi.org/10.1007/s11242-011-9919-2.

Voskov, D. V. 2017. Operator-Based Linearization Approach for Modeling of Multiphase Multi-Component Flow in Porous Media. *J. Comput. Phys.* **337** (15 May): 275–288. https://doi.org/10.1016/j.jcp.2017.02.041.

Voskov, D. V. and Tchelepi, H. A. 2009. Compositional Space Parameterization: Theory and Application for Immiscible Displacements. *SPE J.* **14** (3): 431–440. SPE-106029-PA. https://doi.org/10.2118/106029-PA.

Voskov, D. V. and Tchelepi, H. A. 2011. Compositional Nonlinear Solver Based on Trust Regions of the Flux Function Along Key Tie-Lines. Presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, 21–23 February. SPE-141743-MS. https://doi.org/10.2118/141743-MS.

Voskov, D. V. and Tchelepi, H. A. 2012. Comparison of Nonlinear Formulations for Two-Phase Multi-Component EoS Based Simulation. *J. Pet. Sci. Eng.* **82–83** (February–March): 101–111. https://doi.org/10.1016/j.petrol.2011.10.012.

Wallis, J., Kendall, R., Little, T. et al. 1985. Constrained Residual Acceleration of Conjugate Residual Methods. Presented at the SPE Reservoir Simulation Symposium, Dallas, 10–13 February. SPE-13536-MS. https://doi.org/10.2118/13536-MS.

Wang, X. and Tchelepi, H. 2013. Trust-Region Based Solver for Nonlinear Transport in Heterogeneous Porous Media. *J. Comput. Phys.* **253** (15 November): 114–137. https://doi.org/10.1016/j.jcp.2013.06.041.

Weiser, A. and Zarantonello, S. E. 1988. A Note on Piecewise Linear and Multilinear Table Interpolation in Many Dimensions. *Math. Comput.* **50** (181): 189–196. https://doi.org/10.2307/2007922.

Xu, T., Spycher, N., Sonnenthal, E. et al. 2011. TOUGHREACT Version 2.0: A Simulator for Subsurface Reactive Transport Under Non-Isothermal Multiphase Flow Conditions. *Comput. Geosci.* **37** (6): 763–774. https://doi.org/10.1016/j.cageo.2010.10.007.

Younis, R. 2011. *Modern Advances in Software and Solution Algorithms for Reservoir Simulation.* PhD dissertation, Stanford University, Stanford, California.

Zaydullin, R., Voskov, D., James, S. et al. 2014. Fully Compositional and Thermal Reservoir Simulation. *Comput. Chem. Eng.* **63** (17 April): 51–65. https://doi.org/10.1016/j.compchemeng.2013.12.008.

Zaydullin, R., Voskov, D., and Tchelepi, H. 2013. Nonlinear Formulation Based on an Equation-of-State Free Method for Compositional Flow Simulation. *SPE J.* **18** (2): 264–273. SPE-146989-PA. https://doi.org/10.2118/146989-PA.

Mark Khait is a PhD degree student in petroleum engineering at Delft University of Technology. Previously, he worked for 7 years as a senior researcher at Rosneft Oil Company. Khait's current interests include reservoir simulation, nonlinear solvers, parallel computations, many-core architectures, and neural networks. He holds a master's degree in information security from Ufa State Aviation Technical University, Russia.

Denis Voskov joined Delft University of Technology in 2015 as an associate professor in the Department of Geoscience and Engineering. He is leading a group on modeling of flow and transport in the presence of complex physical processes in the deep subsurface. Voskov previously worked at Stanford University as a senior researcher. Before that, he was a chief engineer at Yukos Company, founder and chief technology officer of Rock Flow Dynamics Company, and a leading specialist at the Institute for Problems in Mechanics, Russian Academy of Sciences. Voskov holds a PhD degree in applied mathematics from Gubkin Russian State University of Oil and Gas.