Integrated framework for modelling of thermal-compositional multiphase flow in porous media

Khait, Mark; Voskov, Denis

DOI
10.2118/193932-MS

Publication date
2019

Document Version
Final published version

Published in
Society of Petroleum Engineers - SPE Reservoir Simulation Conference 2019, RSC 2019

SPE-193932-MS

Integrated Framework for Modelling of Thermal-Compositional Multiphase Flow in Porous Media

Mark Khait and Denis Voskov, TU Delft

This paper was prepared for presentation at the SPE Reservoir Simulation Conference held in Galveston, Texas, USA, 10–11 April 2019.

Abstract

Various novel computing architectures, like massively parallel and multi-core, as well as computing accelerators, like GPUs or TPUs, keep regularly expanding. In order to exploit the benefits of these architectures to the full extent and speed up reservoir simulation, the source code has to be inevitably rewritten, sometimes almost completely. We demonstrate how to extract complex physics-related computations from the main simulation loop, leaving only an algebraic multilinear interpolation kernel instead. In combination with linear solvers, which usually have made available soon once the new architecture is introduced, the approach accommodates execution of the entire nonlinear loop on the latest hardware and computational architectures. We describe the integrated simulation framework built on top of this technique and show the applicability of the approach to various challenging physical and chemical problems. All simulation engines along with linear solvers, well controls, interpolation engines, and state operator evaluators are implemented in C++11 and exposed into Python coupling the flexibility of the script language with the performance of C++.

Introduction

Following the desire to cut production costs and reduce risks, the petroleum industry is constantly pushing reservoir simulation to the leading edge. High-fidelity multimillion-cell models are required for better understanding and prediction of fluid behavior in the subsurface. Fast simulation times are essential for history-matching, optimization, and uncertainty quantification. Therefore, the performance of reservoir simulations in terms of both speed and scalability becomes a major issue.

Over the last decades, reservoir simulation evolved together with scientific computing hardware. First, simulation performance benefited from the evolutionary growth of processors clock speed by running on newer hardware. Then, simulators were endowed with the capability to distribute the computational workload among several computers using message passing libraries (e.g., Dogru et al., 2002; Intersect, 2014). This parallelism not only increased simulation performance but also enhanced scalability. Later, despite the halt of the clock speed growth caused by physical limitations, hardware performance continued to increase utilizing multicore architectures. In order to benefit from them, simulators were redesigned

once again to share computations between several threads, keeping, however, coarse-grained parallelism approach (Zhou et al., 2013).

Many-core computing architectures (GPUs) continue to evolve and evidently keep an order of magnitude advantage over CPUs in both processing speed, which is important for computing-bound problems, and memory bandwidth, which is important for memory-bound problems. The vast majority of GPU-related studies cover acceleration of only linear solution stage (Klie et al., 2011; Bayat and Killough, 2013; Fung et al., 2014; Zhou and Tchelepi, 2013; Yu et al., 2012; Chen et al., 2014). In order to exploit the benefits of the accelerator architecture to the full extent for challenging reservoir simulation problems, that is not enough Appleyard et al. (2011). Depending on the complexity of a physical problem, Jacobian matrix assembly (linearization) could take up to 50% of simulation time. Taking into account a limited memory bandwidth between CPU and GPU and an impossibility to overlap linear matrix transfer with its solution, it is evident that the entire simulation loop should be executed on GPU for the best performance (Esler et al., 2014; Mukundakrishnan et al., 2015).

The offloading of the linearization stage to GPU is a challenging problem due to a large size, specificity and complexity of its source code for general purpose applications (Gandham et al., 2016). Recently, a novel Operator- Based Linearization (OBL) approach was proposed Voskov (2017). It allows extracting complex physics-related computations from the main simulation loop, leaving only an algebraic multilinear interpolation kernel instead. It has been shown that the approach is applicable to various challenging physics, including thermal-compositional problems Khait and Voskov (2017b), buoyancy-dominated flow Khait and Voskov (2018a) and chemical multiphase flow and transport Kala and Voskov (2018).

The OBL approach helped to developed a Delft Advanced Research Terra Simulator (DARTS, 2019). The OBL technique separates physics-based rock and fluid property calculations from linearization procedure. In addition to conventional space and time discretizations, a physical parameter space discretization is introduced. The state-based terms of governing equations are parametrized in the physical space of the problem forming dynamically expanding lookup tables. These tables are used in linearization to approximate operators along with derivatives with respect to governing unknowns using piecewise-multilinear interpolation in physical space. The method not only significantly reduces the amount of property-related computations, but detaches physics and simulation kernels. They are built and launched separately and are able to efficiently run on hybrid platforms including the combination of CPU and GPU.

Framework design

Currently, we consider the conservation of mass and energy in a system with n_p phases and n_c components:

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

$$\frac{\partial}{\partial t}\left(\phi \sum_{p=1}^{n_p} \rho_p s_p U_p + (1-\phi)U_r\right) + \mathrm{div}\sum_{p=1}^{n_p} h_p\rho_p \vec{u_p} + \mathrm{div}(\kappa\nabla T) + \sum_{p=1}^{n_p} h_p\rho_p \tilde{q_p} = 0. \tag{2}$$

All terms of the system (1)—(2), can be characterized as functions of physical state $\boldsymbol{\omega}$, spatially variable properties $\boldsymbol{\xi}$, and well control variables $\mathbf{u}$ as follows:

- $\phi(\boldsymbol{\xi}, \boldsymbol{\omega})$ - effective rock porosity,
- $x_{cp}(\boldsymbol{\omega})$ - component mole fraction in a phase,
- $\rho_p(\boldsymbol{\omega})$ - phase molar density,
- $s_p(\boldsymbol{\omega})$ - phase saturation,

- $\vec{u_p}(\boldsymbol{\xi}, \boldsymbol{\omega})$ - phase velocity,
- $\tilde{q_p}(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u})$ - phase in/outflux,
- U_p $(\boldsymbol{\omega})$ - phase internal energy,
- U_r $(\boldsymbol{\xi}, \boldsymbol{\omega})$ - rock internal energy,
- $h_p(\boldsymbol{\omega})$ - phase enthalpy,
- $\kappa(\boldsymbol{\xi}, \boldsymbol{\omega})$ - thermal conduction.

The Darcy law describes phase flow velocity

$$\vec{u_j} = -\left(\mathbf{K}\lambda_j(\nabla p_j - \delta_j \nabla D)\right), \tag{3}$$

where

- $K(\boldsymbol{\xi})$ - effective permeability tensor,
- λ_j $(\boldsymbol{\omega})$ - phase mobility,
- $p_j^{(\omega)}$ - phase pressure,
- δ_j $(\boldsymbol{\omega})$ - vertical pressure gradient,
- $D(\boldsymbol{\xi})$ - vertical depth vector (up-down oriented).

Time discretization

Equations (1-2) are discretized in time using Backward-Euler scheme and approximated in time using the Fully Implicit Method (FIM). It provides unconditional stability and is preferred for general purpose reservoir simulation. This brings strong nonlinearity into the system to be solved, which is further increased by the closure assumption of instantaneous thermodynamic equilibrium.

Space discretization

In order to keep the framework general and flexible, space discretization procedure is left out of simulation engines. They are initialized by a connection list, which represents the connectivity between control volumes in the reservoir and can be built in the same format for both structured and unstructured grids. The connection list for two-point flux approximation (TPFA) is defined by the total amount of grid blocks and a list of connections. Each connection is defined by the set *(i, j,* Γ,Γ_d), where *i* and *j* are indexes of neighboring control volumes, Γ is transmissibility of fluxes and Γ_d is diffusion transmissibility. The sparsity pattern of the Jacobian matrix is computed directly based on the connection list and remains fixed during the simulation.

Linearization

Following the main idea of OBL, the DARTS framework distinguishes operators from governing equations and treats them in a special way. Operators are functions of the state in a single control volume. Typically, they represent a combination of fluid and rock properties and correspond to the most complex and nonlinear part of the governing equations. Sometimes, the dependency of operators on the state is determined through indirect procedures like phase-split, and therefore it is hard to linearize them in a general way. Suggesting an alternative to an automatic or direct hand-differentiation solution to this problem, OBL replaces the operators with their piecewise multilinear approximations. For those, it is possible to express their derivatives with respect to state variables in a general way.

In DARTS, approximated operators values (along with partial derivatives) are computed via multilinear interpolation, where the amount ofdimensions matches the number of nonlinear variables (i.e. the length

ofthe state, or the number of degrees of freedom) in a single control volume. The true operator values, which are used in interpolations, are called supporting points or base points. They are computed in an adaptive manner during simulation (Khait and Voskov, 2018a) once for given state and saved in a special two-level sparse storage for further re-use (see below for details). This approach has proven to be especially effective when property calculations, involved in the evaluation of operator values, are computationally expensive (e.g., complex phase behavior). Since supporting points are values of functions of the state, they do not depend on spatial location and can be applied either across the whole reservoir or at least within the sub-regions where fluid and rock properties remain constant (e.g. PVT regions). Thereby in DARTS, property calculations occur relatively rare and their amount depends not on spatial discretization, but rather on a discretization of parameter space used for operators approximation and development of the simulation in that space.

From the perspective of simulation nonlinear loop, the operators' interpolation replaces properties calculation in the Jacobian assembly step. In addition, it also 'shadows' physical phenomena behind the operators, leaving out only the values of supporting points, which are rarely computed but utilized all the time during interpolation. This allows to detach fluid and rock properties calculation (operators) from the main nonlinear loop, as well as to relax the performance requirements for such calculations. The Jacobian assembly now depends on the choice of nonlinear variables and governing physical mechanisms which are taken into account. The former determines the dimensionality of parameter space, while the latter defines the operators required for the assembly. Once the choice is made, the Jacobian assembly is simply the right combination of approximated operator values with spatial properties and states, which we refer to as the engine.

Detaching operators and engines create unique opportunities in terms of both flexibility and performance. From the perspective of the engine, the exact implementation of evaluation of operators supporting points is not relevant, because operators values along with their derivatives are computed by interpolation. Receiving those, the Jacobian assembly is then done using straightforward computation of derivatives, which is feasible because of the simplicity of the governing equations written in operator form even for a complex physical applications.

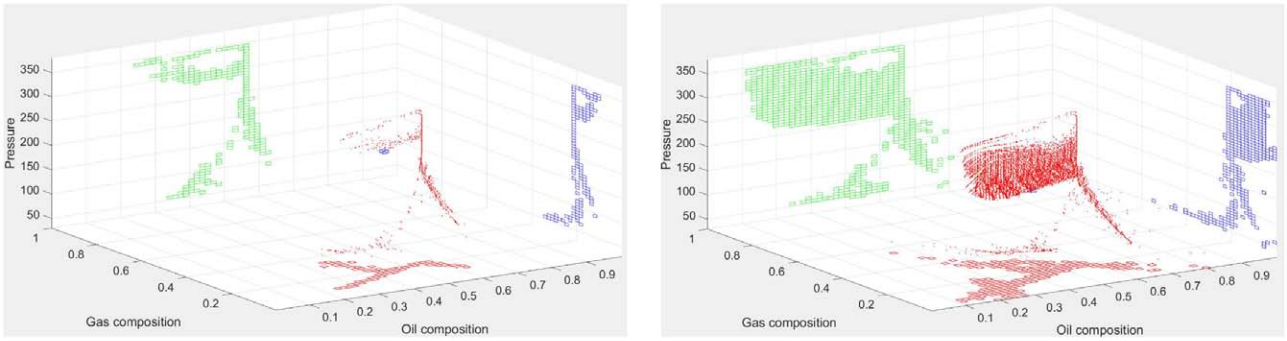
Parameter space discretization

The description of fluid and rock properties in DARTS is simplified via approximation interpolants for the operators within the parameter space of a simulation problem. The dimensionality N and axes of that space are defined by the choice of nonlinear unknowns $\boldsymbol{\omega}$. The bounds of axes correspondent to pressure, temperature or enthalpy can be derived from initial and boundary conditions, while the overall mole fraction is naturally bounded by the interval [0,1].

Then, we apply uniform discretization of parameter space by splitting the bounds for each axis by the same number of points evenly distributed. This results in a set of supporting points in N-dimensional space $\boldsymbol{\omega}^n$, which can be interpreted as a discrete representation of the parameter space in the model. Values of operators are evaluated only for those supporting points in the discrete parameter space which are requested by the interpolation engine during simulation for the first time. Evaluated values are then stored for subsequent requests of interpolation engine. By the end ofthe simulation, the resulting sparse multi-dimensional table of stored supporting points represents an actual subspace of physical parameters being used in the process. For example, Fig. 7 shows a representation of such subspace for a black-oil problem in the beginning (on the left) and by the end (on the right) of simulation.

The evaluated values ofoperators are stored in separate two-level storage, where the first level is optimized for fast data access by the interpolation engine, while the second maximizes the re-use of already computed values. Both levels are associative containers of key-value(s) pairs with unique hash-based keys. This choice was made to ensure fast data access for high-dimensional cases. The first level stores all operator values which belong to a given hyper-rectangular element in the parameter space - all the data required for

interpolation at any point inside that element. The second level stores all operator values that belong to a given vertex (node) ofa hyper-rectangle. The vertex storage helps to avoid re-computations and provides fast data access when a new element is requested by the simulation process. If the new hyper-rectangle shares some vertices with already visited hyper-rectangles, then that vertices will be simply copied to the first level storage. Missing supporting points will be calculated and added to both storages. This approach is crucial in high-dimensional cases when each vertex is shared among many hyper-rectangular elements.



After 10 days of simulation After 1900 days of simulation

Figure 1—Adaptive parametrization

A simple example of this algorithm is displayed in Fig. 2. On the left, an example of two-dimensional parameter space is shown at the moment, when simulation requests rectangle 2, while rectangle 1 was already used. Each rectangle has 4 vertices (for n-dimensional space there will be hyper-rectangles, 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. Since rectangles share vertices, and simulation process is likely to spread continuously over parameter space, in the most cases operator values are re-used. On the right, black arrows represent the process of copying already computed data from vertex storage to hyper-rectangle storage. Green arrows correspond to the supporting points which have never been computed before.

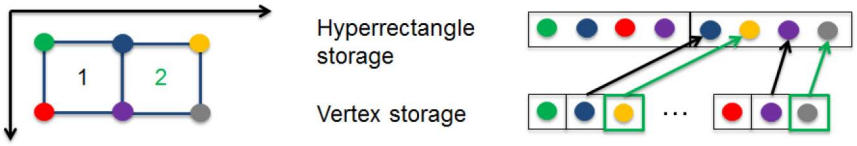


Figure 2—Representation of adaptive OBL storage

DARTS engines

To support the best performance for simulation, there are multiple engines implemented in DARTS. In this section, the main engines are briefly described.

Multiphase multi-component mass transport

For the isothermal engine, the physical state is characterized by $\boldsymbol{\omega} = \{p, z_1, \ldots, z_{n_c-1}\}$. The discretized versions of equations (1) written for block i

$$a(\boldsymbol{\xi})\left(\alpha_c(\boldsymbol{\omega}) - \alpha_c(\boldsymbol{\omega}_n)\right) + \sum_{j \in L(i)} \sum_{p} b_p(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j)\beta_{cp}(\boldsymbol{\omega}_u) + \theta_c(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u}) = 0, \quad c = 1, \ldots, n_c, \tag{4}$$

where

$$a(\boldsymbol{\xi}) = \phi_0 V, \tag{5}$$

$$b_p(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j) = \Delta t \Gamma_{ij} \Phi_{p,ij}, \tag{6}$$

$$\Phi_{p,ij} = \left(p_j + \gamma_p(\boldsymbol{\omega}_j) - p_i - \gamma_p(\boldsymbol{\omega}) - \frac{\delta_p(\boldsymbol{\omega}) + \delta_p(\boldsymbol{\omega}_j)}{2}(D_j - D_i) \right), \tag{7}$$

while

- $\boldsymbol{\omega}$ - the physical state of block i,
- $\boldsymbol{\omega}_n$ - the physical state of block i on the previous time step,
- $\boldsymbol{\omega}_j$ - the physical state of block j,
- $\boldsymbol{\omega}_u$ - the physical state of upwind block determined by $\Phi_{p,ij}$,
- $\alpha_c(\boldsymbol{\omega})$ - the mass accumulation operator for component c with physical meaning of the molar mass of component c per unit pore volume of uncompressed rock under physical state $\boldsymbol{\omega}$,
- $\beta_{cp}(\boldsymbol{\omega})$ - the mass flux operator for component c in phase p with physical meaning of the movable molar mass of that component in that phase in the mixture under physical state $\boldsymbol{\omega}$ per unit time, pressure gradient and transmissibility,
- $\gamma_p(\boldsymbol{\omega})$ - the capillary pressure operator for phase p with physical meaning ofthe difference between phase p pressure and p_i in the mixture under physical state $\boldsymbol{\omega}$,
- $\delta_p(\boldsymbol{\omega})$ - the density operator for phase p with physical meaning of the density of phase p in the mixture under physical state $\boldsymbol{\omega}$.

Multiphase multi-component mass and energy transport

The physical state of the thermal system is characterized by $\boldsymbol{\omega} = \{p, T, z_1, \ldots, z_{n_c-1}\}$. In addition to mass balance equation (4), which form is preserved and the only change is extended physical state vector, the energy balance equation is introduced:

$$\begin{aligned} a_{ef}(\boldsymbol{\xi})(\alpha_{ef}(\boldsymbol{\omega}) - \alpha_{ef}(\boldsymbol{\omega}_n)) + a_{er}(\boldsymbol{\xi})(\alpha_{er}(\boldsymbol{\omega}) - \alpha_{er}(\boldsymbol{\omega}_n)) + \sum_{j \in L(i)} \sum_{p} b_{ep}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j)\beta_{ep}(\boldsymbol{\omega}_u) + \\ \sum_{j \in L(i)} \left(c_{ef}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j)\varepsilon_{ef}(\boldsymbol{\omega}_{eu}) + c_{er}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j)\varepsilon_{er}(\boldsymbol{\omega}_{eu}) \right) + \theta_e(\boldsymbol{\xi}, \boldsymbol{\omega}, \mathbf{u}) = 0, \end{aligned} \tag{8}$$

where

$$a_{ef}(\boldsymbol{\xi}) = a(\boldsymbol{\xi}), \tag{9}$$

$$a_{er}(\boldsymbol{\xi}) = (1 - \phi_0) V U_r, \tag{10}$$

$$b_{ep}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j) = b_p(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j) \tag{11}$$

$$c_{ef}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j) = \Delta t \Gamma_{d,ij}(T_j - T_i)\phi_{0,eu}, \tag{12}$$

$$c_{er}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j) = \Delta t \Gamma_{d,ij}(T_j - T_i)(1 - \phi_{0,eu})\kappa_{r,eu}, \tag{13}$$

while

- $\boldsymbol{\omega}_{eu}$ - the physical state of energy upwind block determined by $T_j - T_i$,
- $\kappa_{r,eu}$ - the rock thermal conduction of energy upwind block determined by T_j — T_i,

- $\alpha_{ef}(\boldsymbol{\omega})$ - the energy accumulation operator for the fluid with physical meaning of the internal energy of fluid per unit pore volume of uncompressed rock under physical state $\boldsymbol{\omega}$,
- $\alpha_{er}(\boldsymbol{\omega})$ - the energy accumulation operator for rock with physical meaning of the internal energy of rock per unit uncompressed volume ofrock under physical state $\boldsymbol{\omega}$,
- $\beta_{ep}(\boldsymbol{\omega})$ - the energy flux operator for phase p with physical meaning of the movable energy of that phase in the mixture under physical state $\boldsymbol{\omega}$ per unit time, pressure gradient and transmissibility,
- $\varepsilon_{ef}(\boldsymbol{\omega})$ - the energy conduction (diffusion) operator for the fluid with physical meaning of the conducted energy of fluid under physical state $\boldsymbol{\omega}$ per unit time, temperature gradient, diffusion transmissibility and uncompressed pore volume,
- $\varepsilon_{er}(\boldsymbol{\omega})$ - the energy conduction (diffusion) operator for rock with physical meaning of the conducted energy of rock under physical state $\boldsymbol{\omega}$ per unit time, temperature gradient, diffusion transmissibility and uncompressed rock volume.

Enthapy-based thermal formulation

Forthermalapplications, there are specific cases, when the physical state $\boldsymbol{\omega} = \{p, T, z_1, \ldots, z_{n_c-1}\}$ cannot describe the phase composition ofthe system. For example, for high enthalpy geothermal formulation $n_p = 2$ and $n_c = 1$. According to the Gibbs phaserule, in 2-phaseregion the systemhas only 1 degree offreedom, and steam saturation cannot be derived from the state. For such cases, an alternative is to use the fluid enthalpy for a thermal description ofthe system: the physical state $\boldsymbol{\omega} = \{p, T, z_1, \ldots, z_{n_c-1}\}$. In this case, the form of equations (4) and (8) remains the same with only the difference that temperature now becomes an operator:

$$c_{ef}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j) = \Delta t \Gamma_{d,ij}(T(\boldsymbol{\omega}_j) - T(\boldsymbol{\omega}))\phi_{0,eu}, \tag{14}$$

$$c_{er}(\boldsymbol{\xi}, \boldsymbol{\omega}, \boldsymbol{\omega}_j) = \Delta t \Gamma_{d,ij}((T(\boldsymbol{\omega}_j) - T(\boldsymbol{\omega}))(1 - \phi_{0,eu})\kappa_{r,eu}. \tag{15}$$

Physical operators

In this section, we briefly describe all physics-based operators used in governing equations.

Isothermal compositional

In the conventional compositional problem, only conservation of mass is involved in governing equations. All physics-based operators can be present in the generic form for each component as

$$\alpha_c(\boldsymbol{\omega}) = (1 + c_r(p - p_{ref})) \sum_{p=1}^{n_p} x_{cp}\rho_p s_p, \tag{16}$$

$$\beta_{cp}(\boldsymbol{\omega}) = x_{cp}\rho_p\lambda_p, \tag{17}$$

$$\gamma_p(\boldsymbol{\omega}) = p_p - p, \tag{18}$$

$$\delta_p(\boldsymbol{\omega}) = \rho_p. \tag{19}$$

Here, all fluid-phase equilibrium calculations and property evaluation performed based on Peng-Robinson Equation of State (EoS) and different EoS-based correlations as described in Iranshahr et al. (2013) or mixed treatment of EoS and PVT correlations as used in Ganapathy and Voskov (2018).

Thermal-compositional operators

The last formulation is thermal-compositional operators involved in the energy balance equation. All thermal properties are calculated based on correlations described in Zaydullin et al. (2015). The physical operators can be present in the following form

$$\alpha_{ef}(\boldsymbol{\omega}) = (1 + c_r(p - p_{ref})) \sum_{p=1}^{n_p} \rho_p s_p U_p \tag{20}$$

$$\alpha_{er}(\boldsymbol{\omega}) = \frac{1}{(1 + c_r(p - p_{ref}))}, \tag{21}$$

$$\beta_e(\boldsymbol{\omega}) = \sum_{p=1}^{n_p} h_p \rho_p \lambda_p, \tag{22}$$

$$\varepsilon_{ef}(\boldsymbol{\omega}) = (1 + c_r(p - p_{ref})) \sum_{p=1}^{n_p} s_p \kappa_p, \tag{23}$$

$$\varepsilon_{er}(\boldsymbol{\omega}) = \frac{1}{(1 + c_r(p - p_{ref}))}. \tag{24}$$

$$\tag{25}$$

The operators for geothermal formulation are computed using the open-source implementation of IAPWS-97 formulation (Cooper and Dooley, 2007). The performance of the simulation does not deteriorate significantly despite the library is implemented in Python.

Modeling of wells

Following the general unstructured grid framework, a well is discretized by a set of control volumes of well segments, chained together by connections. The current implementation only includes the homogeneous flow model in segments. Any well segment can be connected with an arbitrary number of reservoir control volumes, representing well perforations. For well discretization, we use a connection-based approach, suggested by Lim et al. (1995). Each perforation is characterized by geometrical transmissibility representing the connectivity of corresponding well segments to the reservoir, also referred to as well index. Similar to the connections between reservoir grid blocks, well indexes are computed outside simulation engine taking into account geometry and orientation of wellbore and perforated grid block in general unstructured grid. In addition, the top well segment is also connected to a ghost control volume, which has exactly one connection and is used as a placeholder for well control equations, see details in Jiang (2007).

Two examples are shown in Fig. 3: one-segment well configuration (similar to regular well) is on the left and multi-segment well configuration is on the right. Reservoir control volumes are shown in gray; well control volumes including the top segment w_1 - in blue; well ghost control volume w_0 - in red. The interface between w_0 and w_1 is denoted as w. Black arrows represent connections between reservoir control volumes; blue arrows - perforation connections; red arrows - intra-well connections. Even though the examples show structured grid case with a vertically oriented wellbore, the well configuration in DARTS can be arbitrary owing to connection-based approach to describe well perforations. All well control volumes are considered as the extensions of a reservoir and treated exactly the same way during Jacobian assembly, except for w_0. Each well segment is defined by a volume dependent on a wellbore diameter and a segment length, while other properties are neglected. The flow in the multi-segment well is following the homogeneous multiphase flow in an idealized tube without roughness and slip (Jiang, 2007).

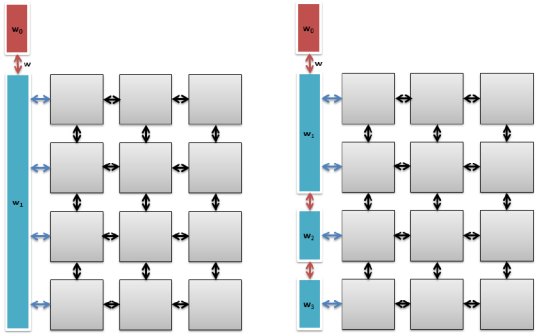


Figure 3—Example of multi-segment well discretization for structured reservoir

BHP well control

One of the two most common controls for wells in reservoir simulation is fixed bottom hole pressure. The following system of equations is applied to w0 control volume instead of Eq. 4 in order to maintain target pressure p^{target}:

$$p - p^{target} = 0, \tag{26}$$

$$z_c - z_c^{up} = 0, \quad c = 1, \ldots, n_c - 1, \quad z_c^{up} = \begin{cases} z_c^{inj} & \text{for injector} \\ z_c^{w_1} & \text{for producer} \end{cases} \tag{27}$$

Rate well control

Another common way to define well regime is to specify volumetric phase rate at surface conditions. In order to parametrize this rate, we first define the state at separator (or surface) conditions using the overall composition of the flux β_c^w over interface w, which is evaluated according to 18:

$$\boldsymbol{\omega}^{SC} = [p^{SC}, T^{SC}, \frac{\beta_1^w}{\sum_c \beta_c^w}, \ldots, \frac{\beta_{n_c-1}^w}{\sum_c \beta_c^w}] \tag{28}$$

Now, we can obtain target rate introducing rate operator $\zeta_j(\boldsymbol{\omega})$:

$$Q_j = \frac{b^w \sum_c \beta_c^w}{dt} \frac{S_j(\boldsymbol{\omega}^{SC})}{\rho_t(\boldsymbol{\omega}^{SC})} = \frac{b^w}{dt} \zeta^w(\boldsymbol{\omega}), \quad \zeta_j^w(\boldsymbol{\omega}) = \begin{cases} \zeta_j(\boldsymbol{\omega}) & \text{for injector} \\ \zeta_j(\boldsymbol{\omega}^w) & \text{for producer} \end{cases} \tag{29}$$

Next, we write down equations for control volume w_0 to maintain target rate Q_j^{target}

$$\frac{b^w}{dt} \zeta_j^w(\boldsymbol{\omega}) - Q_j = 0, \tag{30}$$

$$z_c - z_c^{up} = 0, \quad c = 1, \ldots, n_c - 1, \quad z_c^{up} = \begin{cases} z_c^{inj} & \text{for injector} \\ z_c^{w_1} & \text{for producer} \end{cases} \tag{31}$$

Due to more nonlinear relations involved in operator evaluation for well controls, parametrization tables with higher than in reservoir resolution are often needed to control an error.

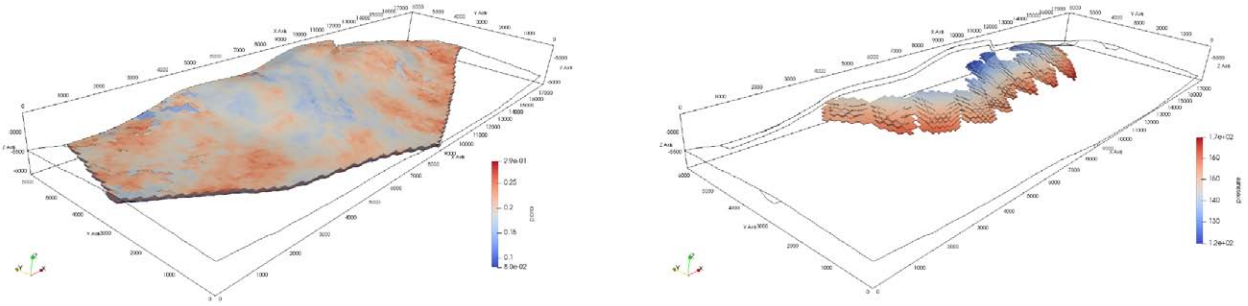
Numerical models

This section presents different test cases utilized for performance comparisons. Models introduced by ascending order in the number of control volumes, the number of unknowns per control volume and the

complexity of physics. All test cases used in this study can be downloaded from the project page (DARTS, 2019).

Brugge field model

The Brugge test case is often used as the optimization benchmark problem in reservoir simulation (Peters et al., 2010). In our study, we used a particular permeability realization and production scenario described in Bukshtynov et al. (2015). The simulation time spans 10 years with BHP controls changing every 3 months for both injection and production wells. In this study, we only use this test case for performance comparisons. The detailed convergence analysis and the comparison with the reference physics can be found in Khait and Voskov (2018a). The number of control volumes in this model is equal to 43,846 and 124,370 connections with 2 unknowns per control volume and dead-oil reference physics.



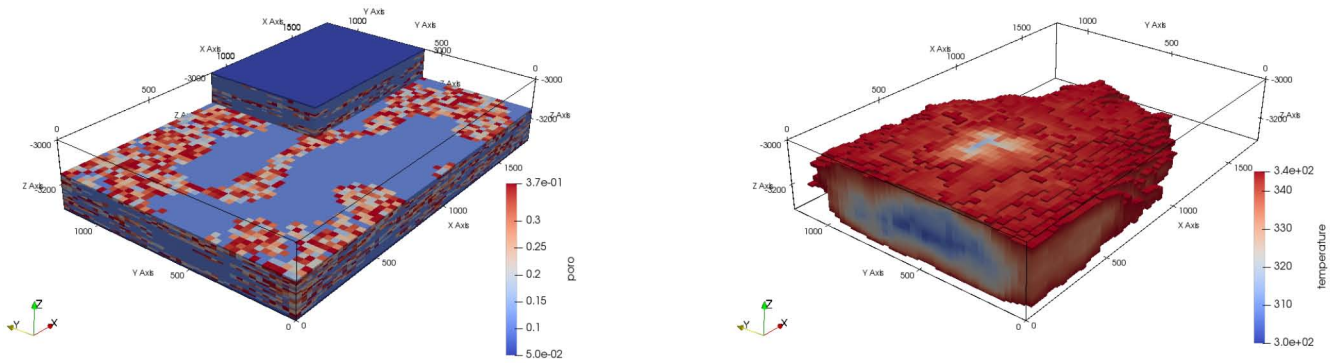
Porosity scaled 3 times along Z axis

Pressure along the water distribution

Figure 4—Brugge field

Geothermal model

To test thermal-compositional formulation in the DARTS framework, we use a geothermal model from Khait and Voskov (2018b). This model is one of the realizations of process-based geological simulation for the Nieuwerkerk sedimentary formation in the West Netherlands Basin used to investigate the performance of a geothermal doublet in low-enthalpy geothermal systems in Shetty et al. (2018). Here we use a modification of the original model where brine and the dissolved methane is present at reservoir conditions. Details on validation of the geothermal model and the convergence analysis for OBL resolution accuracy can be found in Khait and Voskov (2018b). This model has 100,800 control volumes with 295,882 connections, 3 independent unknowns (pressure, enthalpy and composition) and mixed EoS-based and thermal-compositional reference physics (Zaydullin et al., 2015).



Porosity scaled 3 times along Z axis

Temperature solution with threshold below 345 K

Figure 5—Geothermal model

SPE10 model with gas injection

The results of isothermal compositional simulation for gas injection processes is demonstrated using a four- component model described in Khait and Voskov (2018a). In this model, the original distribution of permeability and porosity was taken from SPE10 problem Christie and Blunt (2001). The compositional properties were processed using the Peng-Robinson Equation of State (Peng and Robinson, 1976) with original oil composition from Orr Jr. et al. (1995). The details of the model, comparison with the reference physics and convergence analysis for numerical results can be found in Khait and Voskov (2018a). This model has 1,122,000 control volumes with 3,329,020 connections, 4 nonlinear unknowns per control volume and compositional reference physics.

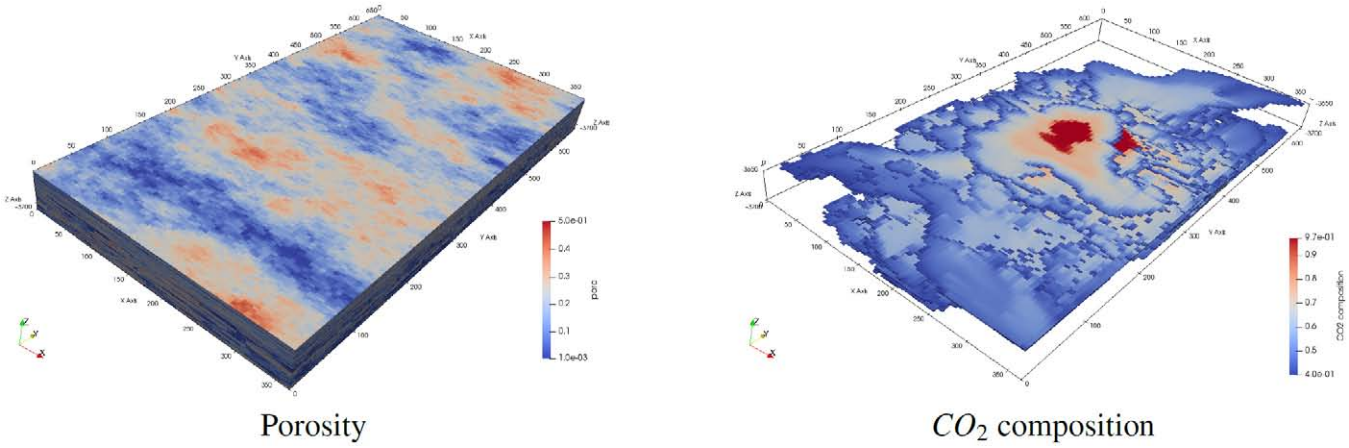


Figure 6—SPE 10 model

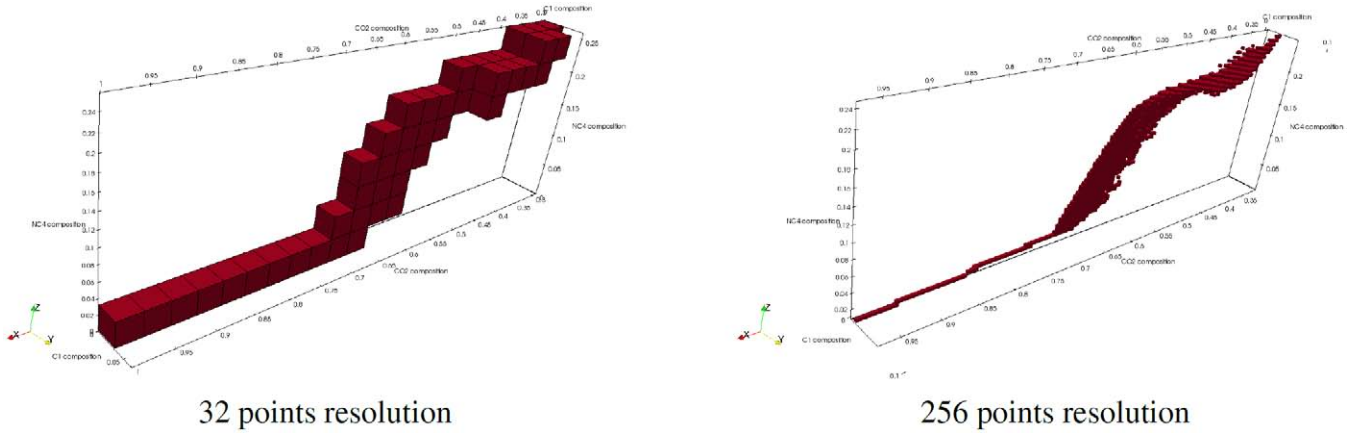


Figure 7—SPE 10 model compositional parameter space for pressure range 53-85 bar

Sensitivity to OBL resolution

Here, we present the results of numerical simulation for the models described above. The models order in the ascending order of complexity where the first model has 87,728 degrees of freedom and simplest physics, the second model has 302,400 degrees of freedom and more complicated physics and the last model has 4,488,000 degrees of freedom with the most nonlinear physics.

The table below presents inclusive simulation time for each model where the first column ‘Sim’ corresponds to the total simulation time, the second column ‘Jac’ represents the linearization time (Jacobian assemble) and the last column ‘Gen’ corresponds to the time spent on generation of supporting points in OBL parametrization. It is clear that for the simplest (Dead-Oil) physics in the first model, the generation time is almost negligible since the property evaluation is extremely cheap (table-based). Even at the most

expensive OBL resolution, the total cost of linearization is below 15% of total simulation time. For the larger model with more complex binary thermal- compositional physics, the cost of generation is growing much faster and soon enough (with the resolution above 256 points) becomes dominant in the simulation. For bigger and more involved four-component compositional model, the linearization cost only becomes noticeable at extremely high OBL resolutions.

Notice that according to our previous investigations (see Khait and Voskov, 2017a, 2018a), the resolution above 64 points already guarantee an error in simulation results below 1%. It is also worth to mention that our multiphase flash solver is not optimized for the performance and only tuned for the accuracy of the phase behavior prediction especially in the near-miscible gas injection regime (close to the critical point). In addition, the parameterized points in OBL can be effectively reused for repeated simulations since the solution in compositional space is mostly controlled by the thermodynamics of the problem (Zaydullin et al., 2013). Therefore, for subsequent launches of Brugge field and Geothermal models, the effective simulation time remains nearly constant for any resolution of parameter space discretization.

Conclusions

The Delft Advanced Research Terra Simulator (DARTS) was introduced and described in this study. We demonstrate how OBL approach predetermined the composition of the computational framework by detaching physics- based rock and fluid property calculations from linearization procedure. The engines and operators concepts were introduced. The major computational load of the simulation process is delegated to a relatively simple engine object. The simplicity of the governing equations written in operator form allows computing all derivatives of Jacobian in a straightforward manner. For operators, DARTS introduces a physical parameter space discretization in addition to conventional space and time discretizations. The actual implementation of operator evaluation becomes almost irrelevant for simulation engine in terms of performance and complexity.

Table 1—Performance results

Resolution	Brugge			Geothermal			SPE10		
	Sim, s	Jac, s	Gen, s	Sim, s	Jac, s	Gen, s	Sim, s	Jac, s	Gen, s
16384	84.6	17.2	5.23	1441.1	1316.1	1300.56	-	-	-
8192	83.5	15.5	4.02	1214.6	1082.7	1066.76	-	-	-
4096	81.5	13.4	2.19	1089.6	959.7	944.00	-	-	-
2048	80.0	11.7	0.88	809.1	683.4	668.40	14731.3	2639.2	2029.9
1024	79.6	11.0	0.30	491.5	368.5	354.22	13359.9	1020.1	419.6
512	79.1	10.4	0.09	266.3	139.3	125.19	10947.5	583.3	71.8
256	80.3	10.3	0.03	170.3	45.1	32.10	9627.5	476.8	12.3
128	78.0	9.8	0.01	137.0	17.6	6.35	7360.8	366.1	2.3
64	82.7	10.2	0.00	133.4	11.3	1.14	6323.8	327.3	0.5
32	84.8	10.3	0.00	130.2	9.7	0.21	5425.8	290.7	0.1
16	81.5	10.0	0.00	129.0	9.4	0.03	5432.4	307.2	0.1

The architecture of DARTS creates unique opportunities in terms of both flexibility and performance. Various formulations were implemented in DARTS, including multiphase thermal-compositional and geothermal models. For the latter, an external library was adopted and utilized without sacrificing in simulation performance. Thanks to the relative simplicity of engine implementation, the whole simulation

is now able to run effectively on a different computing architecture such as GPU, while the calculations of operator values can still be performed on CPU. Adaptive parametrization of operators and localized behavior of simulation development in the parameter space significantly reduce the number of such calculations and prevent them to become a bottleneck. We demonstrate the robustness of the developed framework for models with growing complexity in the number of control volumes, nonlinear unknowns per control volume and complexity of the physics for different table resolutions.

Acknowledgments

We acknowledge the Qatar National Research Fund for partial funding of this research. We also acknowledge Arthur Yuldashev and Ratmir Gubaidullin from GPU Research Center at Ufa State Aviation Technical University, and Oleg Borschuk for their help with linear solvers and valuable discussions.

References

Appleyard, J. R. et al. (2011). Accelerating reservoir simulators using gpu technology. *In SPE Reservoir Simulation Symposium*.

Bayat, M. and Killough, J. (2013). An experimental study of gpu acceleration for reservoir simulation. In Society of Petroleum Engineers - *SPE Reservoir Simulation Symposium 2013*, volume **1**, pages 707–717.

Bukshtynov, V., Volkov, O., Durlofsky, L., and Aziz, K. (2015). Comprehensive framework for gradient-based optimization in closed-loop reservoir management. *Computational Geosciences*, **19**(4):877–897.

Chen, Z., Liu, H., Yu, S., Hsieh, B., and Shao, L. (2014). Gpu-based parallel reservoir simulators. *Lecture Notes in Computational Science and Engineering*, **98**:199–206.

Christie, M. and Blunt, M. (2001). Tenth spe comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Evaluation and Engineering*, **4**(4):308–316.

Cooper, J. and Dooley, R. (2007). Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. *International Association for the Properties of Water and Steam*.

DARTS (2019). Delft Advanced Research Terra Simulator. In https://darts-web.github.io/darts-web/.

Dogru, A. et al. (2002). A parallel reservoir simulator for large-scale reservoir simulation. *SPE Reservoir Evaluation & Engineering*, **5**:11–23.

Esler, K., Mukundakrishnan, K., Natoli, V., Shumway, J., Zhang, Y., and Gilman, J. (2014). Realizing the potential of gpus for reservoir simulation. In *14th European Conference on the Mathematics of Oil Recovery 2014*, ECMOR 2014.

Fung, L., Sindi, M., and Dogru, A. (2014). Multiparadigm parallel acceleration for reservoir simulation. *SPE Journal*, **19**(4):716–725.

Ganapathy, C. and Voskov, D. (2018). Multiscale reconstruction in physics for compositional simulation. *Journal of Computational Physics*, **375**:747–762.

Gandham, R. et al. (2016). Gpu acceleration of equation of state calculations in compositional reservoir simulation. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*.

Intersect, S. (2014). *Intersect Technical Description*.

Iranshahr, A., Voskov, D., and Tchelepi, H. (2013). A negative-flash tie-simplex approach for multiphase reservoir simulation. *SPEJournal*, **18**(6):1140–1149.

Jiang, Y. (2007). *Techniques for Modeling Complex Reservoirs and Advanced Wells*. PhD Thesis, Stanford University.

Kala, K. and Voskov, D. (2018). Parameterization of element balance formulation in reactive compositional flow and transport. In *16th European Conference on the Mathematics of Oil Recovery*, ECMOR 2018.

Khait, M. and Voskov, D. (2017a). Gpu-offloaded general purpose simulator for multiphase flow in porous media. In Society of Petroleum Engineers - *SPE Reservoir Simulation Conference 2017*, pages 1293–1302.

Khait, M. and Voskov, D. (2018a). Adaptive parameterization for solving of thermal/compositional nonlinear flow and transport with buoyancy. *SPE Journal*, **33**(02):522 -534. SPE-182685-PA.

Khait, M. and Voskov, D. (2018b). Operator-based linearization for efficient modeling of geothermal processes. *Geothermics*, **74**:7–18.

Khait, M. and Voskov, D. V. (2017b). Operator-based linearization for general purpose reservoir simulation. *Journal of Petroleum Science and Engineering*, **157**:990 -998.

Klie, H., Sudan, H., Li, R., and Saad, Y. (2011). Exploiting capabilities of many core platforms in reservoir simulation. In Society of Petroleum Engineers - *SPE Reservoir Simulation Symposium 2011*, volume **1**, pages 264–275.

Lim, K.-T. et al. (1995). A new approach for residual and jacobian arrays construction in reservoir simulators. *SPE Computer Applications*, **7**(04):93–96.

Mukundakrishnan, K., Esler, K., Dembeck, D., Natoli, V., Shumway, J., Zhang, Y., Gilman, J., and Meng, H. (2015). Accelerating tight reservoir workflows with gpus. In Society of Petroleum Engineers - *SPE Reservoir Simulation Symposium 2015*, volume **2**, pages 957–973.

Orr, F.Jr., Dindoruk, B., and Johns, R. (1995). Theory of multicomponent gas/oil displacements. *Industrial and Engineering Chemistry Research*, **34**(8):2661–2669.

Peng, D.-Y. and Robinson, D. B. (1976). New two-constant equation of state. *Ind Eng Chem Fundam*, **15**(1):59–64. cited By5991.

Peters, E., Arts, R., Brouwer, G., Geel, C., Cullick, S., Lorentzen, R., Chen, Y., Dunlop, K., Vossepoel, F., Xu, R., Sarma, P., Alhutali, A., and Reynolds, A. (2010). Results ofthe brugge benchmark study for flooding optimization and history matching. *SPE Reservoir Evaluation and Engineering*, **13**(3):391–405.

Shetty, S., Voskov, D., and Bruhn, D. (2018). Numerical strategy for uncertainty quantification in low enthalpy geothermal projects. In *43rd Workshop on Geothermal Reservoir Engineering at Stanford University*.

Voskov, D. V. (2017). Operator based linearization approach for modeling of multiphase multicomponent flow in porous media. *Journal of Computational Physics*, **337**:275 - 288.

Yu, S., Chen, H., Chen, Z., Deng, H., Hsieh, B., Liu, H., and Shao, L. (2012). Gpu-based parallel reservoir simulation for large-scale simulation problems (spe 152271). In *74th European Association of Geoscientists and Engineers Conference and Exhibition 2012 Incorporating SPE EUROPEC 2012: Responsibly Securing Natural Resources*, pages 3168–3173.

Zaydullin, R., Voskov, D., and Tchelepi, H. (2013). Nonlinear formulation based on an equation-of-state free method for compositional flow simulation. *SPE Journal*, **18**(2):264–273.

Zaydullin, R., Voskov, D., and Tchelepi, H. (2015). Phase-state identification bypass method for three-phase thermal compositional simulation. *Computational Geosciences*, pages 1-14.

Zhou, Y., Jiang, Y., and Tchelepi, H. A. (2013). A scalable multistage linear solver for reservoir models with multisegment wells. *Computational Geosciences*, **17**(2):197–216.

Zhou, Y. and Tchelepi, H. (2013). Multi-gpu parallelization of nested factorization for solving large linear systems. In Society of Petroleum Engineers - *SPE Reservoir Simulation Symposium 2013*, volume **1**, pages 163–181.