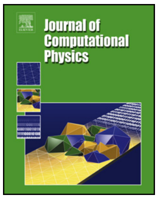
Journal of Computational Physics 428 (2021) 109965

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



A locally conservative multiphase level set method for capillary-controlled displacements in porous media

Espen Jettestuen [a,*], Helmer André Friis [b], Johan Olav Helland [b]

[a] NORCE Norwegian Research Centre, Essendrops gate 3, N-0368 Oslo, Norway
[b] NORCE Norwegian Research Centre, Prof. Olav Hanssensvei 15, N-4021 Stavanger, Norway

ARTICLE INFO

Article history:
Received 28 January 2020
Received in revised form 27 October 2020
Accepted 28 October 2020
Available online 31 October 2020

Keywords:
Level set method
Volume conservation
Three-phase displacements
Ganglia splitting and coalescence
Pore scale
Capillary pressure

ABSTRACT

We present a multiphase level set method with local volume conservation for capillary-controlled displacement in porous structures. Standard numerical formulations of the level set method for capillary-controlled (or, curvature-driven) motions assume phase pressures and interface properties are spatially uniform and disregard the fact that separate phase ganglia typically have distinct pressures. This is a major problem for the suitability of such methods to simulate capillary trapping in porous rocks as it will lead to severe mass loss. The method presented here preserves volumes of individual phase ganglia, while it predicts capillary pressures between ganglia and surrounding phases. A conservative volume redistribution algorithm handles ganglia breakup and coalescence. The method distinguishes between three-phase systems, where separate level set functions describe the different phases, and two-phase systems, where one level set function represents interfaces. We present sequential and parallel algorithms for the new method and emphasize important aspects specific to the patch-based parallel implementation.

We validate the method numerically by applying local volume conservation to simulations of two and three phase systems in both two and three spatial dimensions. The model is tested for both saturation and pressure controlled systems and handles both merging and splitting of phase ganglia.

1. Introduction

Understanding the mechanisms leading to fluid phase trapping and mobilization in three-phase flow in porous media are essential to strategies for improved oil recovery and carbon dioxide (CO_2) storage in mature oil reservoirs. This requires knowledge of how fluid ganglia behave, with coalescence and splitting, in the presence of other phases in the reservoir, as well as their impact on capillary pressure and relative permeability relations used in reservoir simulation. In most of the reservoir, away from wells and fractures, capillary forces typically dominate over viscous and gravity forces during multiphase displacements at the pore scale [1]. Therefore, ganglia of the non-wetting phase (e.g., gas or oil), surrounded by a continuous wetting phase (e.g., water), likely get trapped by capillary forces [2]. This is a desired mechanism for safe CO_2 storage [3]. However, in a three-phase system, the action of capillary pressure can mobilize ganglia by double displacement, in which a continuous phase (e.g., gas) displaces an isolated cluster of the second phase (e.g., oil) that in turn displaces the continuous third phase (e.g., water) [4,5]. A complexity with such three-phase displacements is their different behaviours

[*] Corresponding author.
E-mail address: esje@norceresearch.no (E. Jettestuen).

https://doi.org/10.1016/j.jcp.2020.109965

in the presence of triple lines, where all three phases meet, and in the presence of spreading oil layers separating the other two phases. Pore- and core-scale experiments have shown that spreading oil layers, which make the oil phase more continuous, and double displacements, can mobilize oil and lead to low residual oil saturation after gas invasion [6–8]. An extension of the double-displacement mechanism refers to the situation where the invading phase displaces a chain of several neighbouring fluid clusters that displaces another continuous phase. This can also happen in two-phase systems under weakly-wet states when the fluid phase present on opposite sides of a ganglion are disconnected from each other and associated with different pressures. Multiple displacements play an important role as mechanisms for oil recovery over multiple water-alternate-gas injection cycles [9–11,8,12].

X-ray microtomography imaging of fluid distributions during multiphase flow experiments in porous media has allowed the analysis of shape and topology of fluid ganglia [7,12], capillary pressure states of ganglia compared with continuous phases [13,14], wetting state and displacement mechanisms [15,16,12]. The use of digital rock physics, specifically direct numerical simulation on segmented rock images, is potentially a tool for investigating three-phase processes. Fundamental to simulating multiphase flow with isolated ganglia on realistic pore structures is the numerical method's ability to conserve mass (or volume, for incompressible fluid flow) in a correct manner.

There are many methods for conservation of volume in multiphase flow. Lattice Boltzmann [17] and phase field [18,19] methods add interaction between phases and use the emerging phase separation as the basis for multiphase solvers, where the phase interaction potential set surface tensions and contact angles. In these schemes fluid masses are explicitly conserved but undesired numerical effects such as mutual pollution of non miscible phases, large diffuse interface regions, and spurious currents, can occur.

For capillary-dominated multiphase flow regimes, a reasonable assumption is that capillary forces alone control the interface displacement. Examples of such quasi-static, capillary-controlled methods include pore-network modelling that typically uses an idealization of the pore geometry [e.g., 20] and pore-morphology methods [21]. Intrinsically, these methods require additional rules or techniques for volume conservation.

Implicit interface-tracking techniques are direct numerical simulation methods suitable to capillary-controlled (or, curvature-driven) displacements. Within this category, the two most widely used approaches are the level set (LS) method [22–25] and the volume of fluid (VOF) method [26]. VOF methods have the advantage that they conserve mass but calculation of interfacial properties like normal vectors and curvatures are cumbersome. For the level set method it is the other way around. Calculating interfacial properties is straightforward, but the method has known issues with mass conservation. However, there are methods for improving mass conservation in LS methods. For instance, Sussman and Puckett [27] developed a hybrid method using both VOF and LS approaches, Enright et al. [28] used a hybrid particle-LS method to preserve volume, and Luo et al. devised a method based on spectrally refined interface approaches [29]. Saye and Sethian [30] preserved volumes of each phase within their LS-based Voronoi implicit interface method, by modifying the phase pressures to control interface motion in normal direction.

In this work, we develop a multiphase level set method with local volume conservation (MLS-LVC) to study capillary-controlled three-phase displacements at the pore scale. We have previously implemented global volume conservation in our LS [31] and MLS [32] methods using the approach of Saye and Sethian [30], which allows simulations with prediction of capillary pressure for specified phase saturations [33,32,34]. Here, we extend this approach further to conserve the individual volumes of all isolated ganglia of one or more phases, while treating ganglia coalescence and splitting through a new conservative volume redistribution algorithm. The resulting MLS-LVC method predicts surface area, volume, and capillary pressure for individual fluid ganglia at equilibrium states, thus providing complementary insights to pore-scale experiments addressing ganglion behaviour [e.g., 14,7].

The paper is organized as follows: First, Section 2 presents the main features of the MLS method. Then, we describe the LVC method in the MLS setting (Section 3) and explain the modifications needed to implement it for two-phase systems using a standard LS approach (Section 4). Section 5 presents important aspects of our parallel implementation using LVC. Section 6 presents three-phase MLS-LVC simulations of double and multiple displacements through a 2-D sinusoidal pore channel where several ganglia coalesce and split. We investigate both the effects of applying LVC to one and two phases as well as the behaviour of volume conservation and capillary pressure convergence of disconnected phase ganglia as grid resolution increases. Thereafter, we apply the LVC algorithm to the LS and MLS methods to carry out respective two- and three-phase simulations of primary drainage and subsequent gas invasions in a synthetic 2-D porous medium and on a small subvolume of 3-D sandstone. Section 7 summarizes our work with main conclusions. Future work will target simulation of three-phase residual saturations after water-alternate-gas invasion cycles for comparison with experiments and empirical correlations.

2. Multiphase level set (MLS) method

The level set method [22–24] is a numerical scheme for propagation of curves and surfaces in accordance with a given velocity function, $\vec{F}$. This is accomplished by defining a curve or surface implicitly as the zero contour of a scalar function $\phi(\vec{x})$, where $\phi(\vec{x})$ is the position vector, that evolves according to the equation:

$$\phi_t + \vec{F} \cdot \nabla\phi = 0. \tag{1}$$

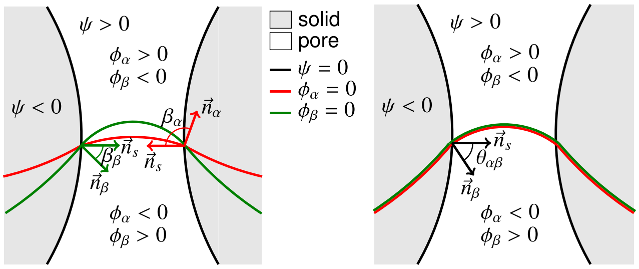


Fig. 1. MLS representation of fluid/fluid/solid contact lines in a pore throat containing fluid phase α and β before (left) and after (right) enforcing the projection step of Losasso et al. [35]. (For interpretation of the colours in the figures, the reader is referred to the web version of this article.)

It is customary to split the velocity term into normal and advective components, F_N and $\vec{F}_A$, respectively, which leads to the following equation:

$$\phi_t + F_N|\nabla\phi| + \vec{F}_A \cdot \nabla\phi = 0. \tag{2}$$

A strength of the level set method is that normal vector $\vec{n}$ of an interface and its curvature κ is calculated from derivatives of the level set function so that $\vec{n} = \nabla\phi/|\nabla\phi|$ and $\kappa = \nabla \cdot \vec{n}$.

For systems containing more than two fluid phases, we will use a multiphase level set (MLS) approach [34,32] that assigns one level set function and one evolution equation per fluid phase α. This yields

$$(\phi_\alpha)_t + F_N^\alpha|\nabla\phi| + \vec{F}_A^\alpha \cdot \nabla\phi_\alpha = 0. \tag{3}$$

We identify domains with $\phi_\alpha < 0$ as part of fluid phase α. The static solid phase is also defined by a scalar field ψ, where $\psi < 0$ is identified as part of the solid and $\psi > 0$ is pore space. One challenge with using multiple level set functions is the presence of overlap regions, where two or more phases are present, and vacuum regions, where no phases are present. To avoid these issues we use a projection step method described by Losasso et al. [35], which amounts to identifying the two smallest level set functions at each point of the domain and then subtract the average of these two from all level set functions.

We will investigate quasi-static, capillary-controlled multiphase systems [36,34,32], with well defined contact angles between fluid/fluid interfaces and pore walls [31,34]. The bulk part of the system is governed by the Young-Laplace equation for the fluid/fluid interfaces,

$$P_{\alpha\beta} = p_\alpha - p_\beta = \sigma_{\alpha\beta} C_{\alpha\beta}, \tag{4}$$

where p_α and p_β are phase pressures, $P_{\alpha\beta}$ is the capillary pressure, $\sigma_{\alpha\beta}$ is the interfacial tension, and $C_{\alpha\beta}$ is the interface curvature, for interfaces between phases α and β.

The contact angle $\theta_{\alpha\beta}$, measured through phase β, is the angle between the $\alpha\beta$-interface and the solid pore wall. It is given by the fluid/fluid interfacial tension, $\sigma_{\alpha\beta}$, and the solid/fluid interfacial tensions $\sigma_{s\alpha}$ and $\sigma_{s\beta}$ through Young's equation:

$$\sigma_{\alpha\beta}\cos(\theta_{\alpha\beta}) = \sigma_{s\alpha} - \sigma_{s\beta}. \tag{5}$$

The MLS method assigns a surface tension contribution γ_α and a solid/fluid intersection angle β_α to each fluid phase [34,32]. The γ_α's are related to the fluid-fluid interfacial tensions through the identity [36]

$$\sigma_{\alpha\beta} = \gamma_\alpha + \gamma_\beta. \tag{6}$$

We define the solid-fluid interfacial tensions as $\sigma_{s\alpha} = \gamma_s - \gamma_\alpha \cos\beta_\alpha$, where γ_s is chosen so that all $\sigma_{s\alpha} > 0$. Further, β_α is a specified angle between the pore wall and the boundary of phase α, defined by $\cos\beta_\alpha = \vec{n}_\alpha \cdot \vec{n}_s$, where $\vec{n}_\alpha$ and $\vec{n}_s$ are the normal vectors derived from ϕ_α and ψ, respectively, see Fig. 1. Consistent with Eq. (5), the contact angle $\theta_{\alpha\beta}$ is given by

$$\cos\theta_{\alpha\beta} = \vec{n}_\beta \cdot \vec{n}_s = \frac{\gamma_\beta\cos\beta_\beta - \gamma_\alpha\cos\beta_\alpha}{\sigma_{\alpha\beta}}. \tag{7}$$

Note that this expression is independent of γ_s. For given sets of $\sigma_{\alpha\beta}$ and $\theta_{\alpha\beta}$ we use γ- and β-parameters that satisfy Eqs. (6) and (7) as input to the MLS simulations. In two-phase simulations, using two level set functions, a possible choice for the intersection angles is $\beta_\beta = \theta_{\alpha\beta}$ and $\beta_\alpha = 180° - \theta_{\alpha\beta}$.

The algorithm used to find the local equilibrium of a multiphase system, where each phase is given a uniform pressure, is based on assigning an energy cost γ_α for each phase boundary so that the total energy of the system becomes the sum of bulk and surface contributions for each phase separately. We have explored this approach previously [32,34]. Here, we also want to consider that the pressure in each isolated region of each phase can be different, so that we need to keep track of individual regions that can change shape and topology in time. However, we shall assume that all regions of the same phase still have the same surface tension and contact angle. The total energy of the system [32] can now be written as

$$E = \sum_{\alpha} \left(\int\limits_{\Omega} p_\alpha H(-\phi_\alpha) H(\psi)\, \mathrm{d}V + \gamma_\alpha \int\limits_{\Omega} \delta(\phi_\alpha) |\nabla \phi_\alpha| H(\psi)\, \mathrm{d}V + \sigma_{s\alpha} \int\limits_{\Omega} \delta(\psi) |\nabla \psi| H(-\phi_\alpha)\, \mathrm{d}V \right), \tag{8}$$

where Ω is the computational domain, $H(\cdots)$ is the Heaviside step function and $\delta(\cdots)$ is the Dirac delta function. The velocity functions that are used to minimize this energy are [32]

$$F_N^\alpha = H(\psi)(p_\alpha - \gamma_\alpha \kappa_\alpha) - \frac{H(-\psi)}{\Delta x} S(\psi) \gamma_\alpha |\nabla \psi| \cos \beta_\alpha, \tag{9}$$

$$\vec{F}_A^\alpha = \frac{H(-\psi)}{\Delta x} S(\psi) \gamma_\alpha \nabla \psi, \tag{10}$$

where $\kappa_\alpha = \nabla \cdot (\nabla \phi_\alpha / |\nabla \phi_\alpha|)$ is the curvature field calculated from phase α's level set function, and $S(\cdots)$ is the sign function. We note that the pressure p_α, and consequently κ_α, is not only dependent on phase but also, possibly, on domain as will be explained in the next section.

We use an explicit numerical scheme based on finite differences to solve Eqs. (3), (9) and (10), as described elsewhere [32,24]. With the projection method of Losasso et al. [35] enforced at the end of each iteration-time step, the equilibrium solutions satisfy [32]

$$(p_\alpha - \gamma_\alpha \kappa_\alpha) |\nabla \phi_\alpha| = \left(p_\beta - \gamma_\beta \kappa_\beta\right) |\nabla \phi_\beta| \tag{11}$$

in pore space, and

$$\gamma_\alpha \left(\nabla \phi_\alpha \cdot \nabla \psi - \cos \beta_\alpha |\nabla \phi_\alpha| |\nabla \psi|\right) = \gamma_\beta \left(\nabla \phi_\beta \cdot \nabla \psi - \cos \beta_\beta |\nabla \phi_\beta| |\nabla \psi|\right) \tag{12}$$

at the pore walls. Around $\alpha\beta$-interfaces, we have that $\phi_\alpha = -\phi_\beta$, which implies $\nabla \phi_\alpha = -\nabla \phi_\beta$, $|\nabla \phi_\alpha| = |\nabla \phi_\beta|$ and $\kappa_\alpha = -\kappa_\beta$. Thus, Eqs. (11) and (12) are consistent with Eqs. (4) and (7).

3. Local volume conservation (LVC) method

Our goal is to develop a simple and lightweight algorithm that, to a good approximation, conserves the volume of a phase and each isolated region of that phase. When evolving this system, regions can coalesce and split, so we need to devise an algorithm that can robustly redistribute volumes between regions that change both shape and topology.

We introduce the following notation: For a given fluid phase α, let $I_\alpha(t) \subset \mathbb{N}$ be the enumeration of all isolated regions of that phase at iteration time t, similar to an indicator function. Then, $\Omega_a(t) \subset \Omega$, where $a \in I_\alpha(t)$, is an isolated region in space. To implement volume conservation of the isolated regions we use the method introduced by Saye and Sethian [30] where the pressure $p_a(t)$ of region Ω_a, $a \in I_\alpha(t)$, at iteration-time t, is given as

$$p_a(t) = \frac{V_a^{(0)}(t) - V_a(t)}{\Delta t A_a(t)}. \tag{13}$$

Here, $V_a^{(0)}(t)$ is the target volume of region Ω_a, $V_a(t)$ is the region volume at iteration-time t, $A_a(t)$ is the region surface area, and Δt is the iteration-time step. The pressure p_a only equals the pressure in region a when the evolution of the LS fields has reached its equilibrium state. In this stationary state, V_a will deviate from $V_a^{(0)}$ by a small amount governed by Δt. Hence, this method works by assigning an artificial numerical compressibility to every isolated region. We note that Δt is not a physical time so that its dimensions are chosen according to Eq. (13), also in agreement with the iterative time unit in Eq. (3).

We identify isolated regions with the grassfire algorithm described by Prodanović et al. [37]. To each region we assign a volume $V_a^{(0)}$ so that the total target volume of phase α is given as

$$V_\alpha^{(0)} = \sum_{a \in I_\alpha} V_a^{(0)}. \tag{14}$$

Our method requires a strategy to assign volumes to regions at the next time based on the volume of regions at the previous time step. For this purpose, we define *extended regions* $\Omega_a^{\text{ext}}(t) \supset \Omega_a(t), a \in I_\alpha(t)$, that completely partition the computational domain, i.e., $\Omega = \cup_{a \in I_\alpha(t)} \Omega_a^{\text{ext}}(t)$. We construct these extended regions $\Omega_a^{\text{ext}}(t)$ by including all points nearest to a given region $\Omega_a(t)$ using a distance transform. In this work we have used the method described by Rosenfeld and Pfaltz [38]. Using this algorithm, we can calculate the integer distance $d(\vec{x}, \Omega_a)$ from a voxel (or, grid cell) at position $\vec{x}$ to the boundary of each region $\Omega_a, a \in I_\alpha$. For each voxel $\vec{x} \notin \Omega_a, a \in I_\alpha$, we mark it as part of the extended region Ω_a^{ext}, given

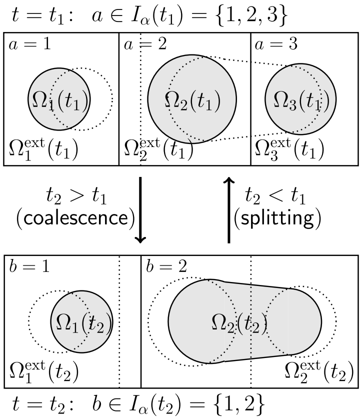

$t = t_1: \ a \in I_\alpha(t_1) = \{1,2,3\}$
$a = 1$
$a = 2$
$a = 3$
$\Omega_1(t_1)$
$\Omega_2(t_1)$
$\Omega_3(t_1)$
$\Omega_1^{\mathrm{ext}}(t_1)$
$\Omega_2^{\mathrm{ext}}(t_1)$
$\Omega_3^{\mathrm{ext}}(t_1)$
$t_2 > t_1$ (coalescence)
$t_2 < t_1$ (splitting)
$b = 1$
$b = 2$
$\Omega_1(t_2)$
$\Omega_2(t_2)$
$\Omega_1^{\mathrm{ext}}(t_2)$
$\Omega_2^{\mathrm{ext}}(t_2)$
$t = t_2: \ b \in I_\alpha(t_2) = \{1,2\}$

Fig. 2. Illustration of the volume redistribution algorithm with an example of domain coalescence ($t_2 > t_1$) and splitting ($t_1 > t_2$). Calculation of target volumes follows Eq. (16). The vertical lines indicate the borders between extended regions, while the grey areas show the regions of conserved phase. The stippled lines show the position of interfaces in the previous/next time step. For the case with domain coalescence ($t_2 > t_1$), we obtain the following target volumes at t_2: $V_1^{(0)}(t_2) = V_1^{(0)}(t_1)$ and $V_2^{(0)}(t_2) = V_2^{(0)}(t_1) + V_3^{(0)}(t_1)$. For the case with domain splitting ($t_1 > t_2$), we obtain the following target volumes at t_1 (by swapping a and b, and t_1 and t_2, in Eq. (16)): $V_1^{(0)}(t_1) = V_1^{(0)}(t_2)$, $V_2^{(0)}(t_1) = f_{22} V_2^{(0)}(t_2)$ and $V_3^{(0)}(t_1) = f_{23} V_2^{(0)}(t_2)$. Here, $f_{22} = |\Delta\Omega_{22}(t_2, t_1)|/|\Omega_2(t_2)|$ and $f_{23} = |\Delta\Omega_{23}(t_2, t_1)|/|\Omega_2(t_2)|$.

that $d(\vec{x}, \Omega_a) \leq d(\vec{x}, \Omega_b)$ for all $b \in I_\alpha$ (where we have that $d(\vec{x}, \Omega) \geq 0$). In cases where the smallest distance to several other domains is equal, we choose $\vec{x}$ to be part of the extended domain with smallest domain number.

Now assume we have identified the sets $\{\Omega_a(t_1) : a \in I_\alpha(t_1)\}$ and $\{\Omega_a^{\mathrm{ext}}(t_1) : a \in I_\alpha(t_1)\}$ at time t_1, and the corresponding sets $\{\Omega_b(t_2) : b \in I_\alpha(t_2)\}$ and $\{\Omega_b^{\mathrm{ext}}(t_2) : b \in I_\alpha(t_2)\}$ at time t_2. We define the intersection between region Ω_a at time t_1 and the extended region Ω_b^{ext} at time $t_2 > t_1$ as

$$\Delta\Omega_{ab}(t_1, t_2) = \Omega_a(t_1) \cap \Omega_b^{\mathrm{ext}}(t_2). \tag{15}$$

Then, we use the relative volume of overlap between the t_1 and t_2 regions as a weight for the redistribution of volume. The target volume assigned to region $\Omega_b(t_2)$ is thus given as

$$V_b^{(0)}(t_2) = \sum_{a \in I_\alpha(t_1)} \frac{|\Delta\Omega_{ab}(t_1, t_2)|}{|\Omega_a(t_1)|} V_a^{(0)}(t_1), \tag{16}$$

where $|\cdots|$ denotes the volume of a region, measured in the number of voxels it contains. In our scheme we know that the Ω^{ext} regions cover the whole computational domain so that no volume is lost in the volume reassignment. Note that when coupled with MLS iteration scheme, restrictions on time step lead to relatively small changes in the fluid configurations within an iteration step. This ensures that target domain volumes are redistributed correctly with LVC. Fig. 2 illustrates the volume redistribution algorithm for simple examples with domain coalescence and splitting.

The pressure term given in Eq. (13) for region $\Omega_b(t_2)$, $b \in I_\alpha(t_2)$, is obtained by using $V_b^{(0)}(t_2)$ as the target volume and by using the following volume integrals to evaluate the volume, $V_b(t_2)$, and area, $A_b(t_2)$:

$$V_b(t_2) = \int_{\Omega_b^{\mathrm{ext}}(t_2)} H(\psi) H(-\phi_\alpha) \mathrm{d}V, \tag{17}$$

$$A_b(t_2) = \int_{\Omega_b^{\mathrm{ext}}(t_2)} H(\psi) \delta(\phi_\alpha) |\nabla\phi_\alpha| \mathrm{d}V. \tag{18}$$

Although the accuracy of these volume and area calculations can decrease when domains are sufficiently close to each other, typically on the length scale of the neighbourhood used by the numerical stencil, the sum of target volumes, $V_\alpha^{(0)}$, will still be conserved. The region phase pressures p_b are assigned to the corresponding extended regions Ω_b^{ext}, $b \in I_\alpha$, for use in the MLS iteration scheme. This ensures that the numerical stencils use the correct pressures in the vicinity of fluid/fluid interfaces.

To allow for mass entering and/or leaving the system, we tag all regions of the same phase in contact with inlet/outlet boundaries of the computational domain. We assume all such regions are connected to the same fluid reservoir and belong to the continuous phase with the same pressure. This phase pressure is either assigned or calculated by controlling the phase saturation [32] (see also Section 6). Regions tagged as boundaries are excluded from the local conservation scheme as the pressure is given by the continuous phase. Once new isolated regions detach from the continuous phase, we calculate their target volumes with Eq. (17). Then, Eq. (13) implies that the phase pressures for such regions are zero in the first iteration after region detachment. However, in subsequent iterations these phase pressures eventually converge to correct values as the system relaxes toward equilibrium.

For simulations on realistic pore geometries it is inevitable that tiny phase domains containing only a few voxels arise. This can lead to high and unrealistic region phase pressures calculated from Eq. (13), which in turn can make the MLS iteration scheme unstable. In this work, we resolve this issue by setting the phase pressure of regions with target volumes smaller than a few grid-cell volumes equal to the pressure of the corresponding continuous phase. Specifically, we use $3(\Delta x)^2$ (in 2-D) and $20(\Delta x)^3$ (in 3-D) as volume limits. Although this implies that tiny regions can vanish during the MLS iteration steps, our simulations (see Section 6) indicate that the effect on the volume redistribution is negligible. We note that an alternative approach, that we have not investigated here, would be to exclude tiny regions completely from the LVC algorithm.

3.1. *Formulation of the algorithm*

We now give the main steps involved in the implementation of our LVC method in the MLS setting.
AlgorithmSeq

1. Initialize the system: (i) Construct the set $\{\Omega_a : a \in I_\alpha(t_0)\}$ (using grassfire algorithm) with unique tags $a \in I_\alpha(t_0)$ assigned to each isolated fluid region of the conserved phase α, and (ii) generate the accompanying set of extended regions $\{\Omega_a^{\text{ext}} : a \in I_\alpha(t_0)\}$. Set $t_n = t_0$.
   Loop over iteration time (from t_n to t_{n+1}):
2. Set pressures of conserved phase α equal to the continuous-phase pressure in regions connected to inlet or outlet boundaries. For the other isolated phase α regions, set pressures to p_a, $a \in I_\alpha(t_n)$, using Eq. (13), enforcing volume conservation.
3. Iterate the multiphase system from t_n to t_{n+1} with Eqs. (3), (9) and (10), using the phase pressures from step 2, and then enforce the projection step [35].
4. Repartition the system at time step t_{n+1}, that is, construct the set $\{\Omega_b : b \in I_\alpha(t_{n+1})\}$. Tag regions as boundaries if they belong to the phase connected to inlet or outlet boundaries.
5. Extend the partitioning $\{\Omega_b(t_{n+1})\}$ to obtain the set $\{\Omega_b^{\text{ext}}(t_{n+1})\}$.
6. Redistribute volumes according to Eq. (16), using t_n and t_{n+1} as time step t_1 and t_2, respectively. Regions tagged as boundary belong to the continuous phase and are excluded from the volume redistribution.
7. Identify new regions that detached from the continuous phase and calculate their region target volumes based on Eq. (17).
8. Go to step 2 with $n \leftarrow n+1$.

After repeating this scheme a specified number of times, we reinitialize the ϕ_α-functions to signed distance functions by solving [39]

$$(\phi_\alpha)_t + S(\phi_\alpha)(|\nabla\phi_\alpha| - 1) = 0. \tag{19}$$

The iteration-loop ends when the system is in a stationary state given by [32]

$$\max_\alpha \left\{ \frac{\sum_\Omega H(\psi+\epsilon)H(\phi_\alpha^n+\epsilon)H(-\phi_\alpha^n+\epsilon)|\phi_\alpha^n - \phi_\alpha^m|}{\sum_\Omega H(\psi+\epsilon)H(\phi_\alpha^n+\epsilon)H(-\phi_\alpha^n+\epsilon)} \right\} < c\Delta x, \tag{20}$$

where we use $c = 0.001$ and $\epsilon = 1.5\Delta x$, and n and m are the two last iteration steps where reinitialization of ϕ_α occurred. This end state is treated as the quasi-static equilibrium of the system. By incremental changes in boundary conditions and/or constraints the system approximates the quasi-static evolution of a multiphase system.

4. Two-phase system as a special case

A two-phase system requires only a single level set function ϕ that evolves according to Eq. (2). In this case we can rescale the velocities from Eqs. (9) and (10) by means of interfacial tension to obtain [31]

$$\begin{aligned} F_N &= H(\psi)(C-\kappa) - \frac{H(-\psi)}{\Delta x} S(\psi)|\nabla\psi|\cos\beta, \\ \vec{F}_A &= \frac{H(-\psi)}{\Delta x} S(\psi)\nabla\psi, \end{aligned} \tag{21}$$

where $\beta = 180° - \theta$. In Eq. (21), C is a prescribed value for capillary pressure divided by interfacial tension, representing interface curvature by Eq. (4). With interfacial tension eliminated from the evolution equation, capillary equilibrium in the pore space occurs when the computed curvature field κ has become equal to the prescribed value C around the interfaces (i.e., $C = \kappa$). Note that in the evolution equations of the MLS approach we have chosen to use phase pressures rather than interface curvatures C because we cannot eliminate interfacial tensions from simulations of systems containing more than two fluid phases.

Compared to the MLS approach, applying LVC algorithm in a single level set framework requires only a few changes related to the sign of ϕ. Assuming $\phi < 0$ in phase α and $\phi > 0$ in phase β, we now update interface curvature C_a of isolated domains as follows:

$$C_a(t) = s\frac{V_a^{(0)}(t) - V_a(t)}{\Delta t A_a(t)}, \tag{22}$$

where it is crucial that $s = 1$ for $a \in I_\alpha$ and $s = -1$ for $a \in I_\beta$. Domain volume and area calculations follow Eqs. (17) and (18), except for $a \in I_\beta$ where $\phi > 0$ so that the volume calculation is $\int H(\psi)H(\phi)dV$.

5. Parallel aspects of the LVC algorithm

We have implemented the level set methods described in the previous sections within the Structured Adaptive Mesh Refinement Application Infrastructure (SAMRAI) software system [40]. SAMRAI is an object-oriented MPI and C++-based programming framework that provides an extensible and scalable infrastructure for parallel adaptive mesh refinement (AMR) applications [41–43].

The implementation strategy for structured AMR employed in SAMRAI is the so-called patch-based approach, as opposed to the quad/octree approach which is the other typical alternative. In SAMRAI a given processor can be associated with one or several patches. We have implemented AMR in our previous level set methods for two- and three-phase capillary-controlled displacements in porous media as demonstrated in our recent papers [44,32].

While the patch based parallel approach in SAMRAI can be easily exploited for the level set methods in our previous works, it is more challenging when including the above LVC algorithm. The purpose of this section is thus to explain the implementation of certain important aspects of the parallel LVC algorithm. In the present paper we will only consider uniform grids. The inclusion of AMR introduces additional challenges and is currently work in progress.

5.1. *Parallel algorithm*

We now give the main steps involved in the implementation of our LVC method in the parallel MLS setting. This will closely follow the steps from AlgorithmSeq (Section 3.1), but with the inclusion of additional substeps and comments to explain important extensions needed for parallelism. We will discuss some of these steps in more detail in Section 5.2 as well as in Appendix A.

AlgorithmPar

1. Initialize the system:
   (a) On each patch: Use the grassfire algorithm to identify fluid regions of the conserved phase α with local tags unique to each processor.
   (b) On each processor: Gather the information in (a) from the corresponding patches and communicate information to the master processor.
   (c) On the master processor: Construct unique global tags $a \in I_\alpha(t_0)$ assigned to each isolated fluid region using the algorithm described in Appendix A. Construct the set $\{\Omega_a : a \in I_\alpha(t_0)\}$ with the help of the information from step (a) above. Communicate relevant information back to all processors and patches.
   (d) On each patch: Generate the set of extended regions $\{\Omega_a^{\text{ext}} : a \in I_\alpha(t_0)\}$.
   (e) On each patch: Modify the set of extended regions $\{\Omega_a^{\text{ext}} : a \in I_\alpha(t_0)\}$ by using the distance function $d(\vec{x}, \Omega)$ together with patch neighbourhood information.
   (f) Set $t_n = t_0$

   Loop over iteration time (from t_n to t_{n+1}):
2. Set pressures of conserved phase α equal to the continuous-phase pressure in regions connected to inlet or outlet boundaries. For the other isolated phase α regions, set pressures to p_a, $a \in I_\alpha(t_n)$, using Eq. (13) (enforcing volume conservation) as follows.
   (a) On each patch: Perform computations according to Eq. (17) and Eq. (18).
   (b) On each processor: Similar approach as in step 1 (b).
   (c) On the master processor: Perform Eq. (13) and communicate relevant information back to all processors and patches.
3. Iterate the multiphase system from t_n to t_{n+1} with Eqs. (3), (9) and (10), using the phase pressures from step 2, and then enforce the projection step [35]. This is done in parallel using the patch-based parallelism in SAMRAI.
4. Repartition the system at time step t_{n+1}, that is, construct the set $\{\Omega_b : b \in I_\alpha(t_{n+1})\}$. Tag regions as boundaries if they belong to the phase connected to inlet or outlet boundaries. In parallel this is done in a similar manner as in step 1 (a)-(c) above.
5. Extend the partitioning $\{\Omega_b(t_{n+1})\}$ to obtain the set $\{\Omega_b^{\text{ext}}(t_{n+1})\}$. In parallel this is done in a similar manner as in step 1 (d)-(e) above.

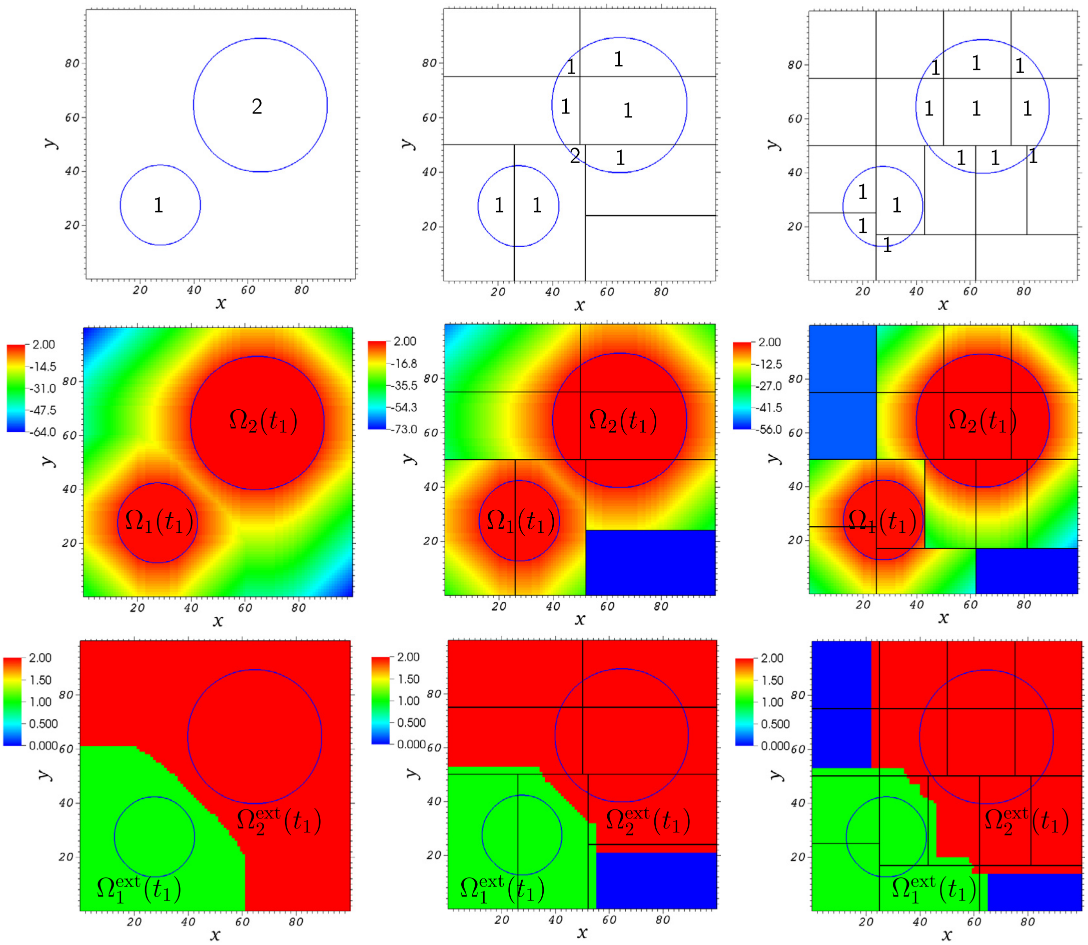


Fig. 3. An example of the decomposition of the computational domain where the left, middle and right columns show the results for 1, 8 and 16 processors, respectively. Each processor contains only one patch in these examples. The top row shows each processor's local tag numbers for the domains. The middle row shows the distance map calculated for individual patches. The distance map consists of the global tags $a \in I_\alpha$ inside the regions Ω_a and in the exterior it contains the negative distance to the various regions. The bottom row shows the extended global map, where green is the extension of domain 1, red is the extension of domain 2 and blue shows patches without domains.

6. Redistribute volumes
   (a) On each patch: Perform computations according to Eq. (15) and Eq. (16) (denominator).
   (b) On each processor: Similar approach as in step 1 (b).
   (c) On the master processor: Perform Eq. (16), using t_n and t_{n+1} as time step t_1 and t_2, respectively. Regions tagged as boundary belong to the continuous phase and are excluded from the volume redistribution. Finally, communicate relevant information back to all processors and patches.
7. Rest of algorithm resembles steps 7 and 8 in AlgorithmSeq.

5.2. *Simple illustration of some aspects of the parallel algorithm*

Here, we elaborate on essential parts of the parallel algorithm given above using the following simple two-dimensional example containing two isolated regions (see Fig. 3). Two domains need to be identified in this simple example, and in the top row of the figure we show local labels (tags) as well as the various patches for one, eight and sixteen processors, respectively. Obviously, local and global labels coincide in the case of a single processor. Note that in parallel computing local tags unique to each processor are identified by performing the grassfire algorithm on each patch. After communicating the various results to the master processor global tags have to be associated with each isolated region, since a given region can cover several patches (and processors). This is done by using the algorithm described in Appendix A.

In the middle row of Fig. 3 the regions are denoted by Ω_1 and Ω_2, with corresponding global labels 1 and 2, respectively. Outside of the two domains in these three subfigures we show the distance map, which is used for computing the extended regions. This is just the negative integer distance computed with respect to the nearest domain (i.e. $-d(\vec{x}, \Omega)$). In patches

Table 1
Weak scaling results for various cases of conserved phases and number of isolated domains.

Absolute timing (s) for weak scalability tests					
# cores	Without LVC	LVC 1 phase (32 domains)	LVC 2 phases (64 domains)	LVC 1 phase (108 domains)	LVC 2 phases (216 domains)
1	11.97	13.77	15.61	14.36	15.85
8	12.78	14.48	16.15	16.33	19.8
64	16.41	19.72	23.07	22.48	28.07
512	17.33	21.14	24.83	27.02	36.19

without domains this negative integer distance is simply put to the negative sum of the number of grid points in each coordinate direction.

The corresponding extended regions are shown in Fig. 3, bottom row. In the parallel computing process we iterate through the patches connected to a processor and perform the “extension algorithm” (described in Section 3) for each patch. However, when this is completed we must yet again iterate through the patches and possibly correct the results from the extension based on patch neighbourhood information. In essence we compare the “negative distance map” of neighbouring patches and give preference to the shortest negative distance when updating the “extension map”.

As seen in Section 3, volume and area computations of the different regions are central to the LVC algorithm. We perform such computations on each patch in parallel. Different contributions are then gathered on each processor and communicated to the master processor where they can be put together with the help of the global domain identification tags discussed above.

The volume redistribution algorithm described in the previous section is also at first done on each patch in parallel in order to find the various domain sizes and corresponding intersections (“hits”), both represented as integers, between old and new domains. Communications are again needed to gather the various data structures on the master processor. After a somewhat involved process involving matrix-like structures the new domain volumes can be computed as in the sequential algorithm (see Section 3). Finally, the pressures in the different regions can be computed, and then communicated to the various processors for use in the level set algorithm.

5.3. Parallel scalability of the LVC algorithm

The parallel version of the LVC algorithm introduces additional MPI communication in the LS/MLS codes. In this section we thus investigate the parallel performance of the LVC algorithm using the MLS method. We present tests for both weak and strong scalability of the parallel code based on simulations of curvature-driven evolution of isolated domains toward a steady state where the domains attain spherical shapes. In most realistic applications the size of the data structures involved in the MPI communication will be relatively modest, but it will grow with the number of isolated domains that needs to be identified. Since the parallel scalability thus clearly will depend on this number, we show results where the number of isolated domains ranges from 0 to 216, representing typical simulations in realistic pore geometries.

In the scaling tests the distributions of the phases subjected to LVC (that is, the oil and gas phases) constitute fixed numbers of isolated cubic domains, whereas the remaining phase (water) is continuous. We consider two such cases, as shown in Fig. 4(a). The first case contains 64 ($=4^3$) domains (32 of each phase) and the second case contains 216 ($=6^3$) domains (108 of each phase). We explore scaling behaviour in each of these cases for the settings without LVC and with LVC applied to one (i.e., oil) and two (i.e., oil and gas) phases. To maintain control of the number of isolated domains present in the system the computational domain consists entirely of pore space. In the weak scaling tests we first simulate a computational domain with 40^3 grid cells using a single processor. We then refine this configuration three times (keeping the number of isolated domains constant) so that the problem size and number of processors used in the simulations both increase with a factor of 8 for each refinement. We thus end up with a problem containing 320^3 grid cells in the case of 512 processors. In the strong scaling tests we perform simulations on the largest problem size (320^3 grid cells) using 128, 256, 512 and 1024 processors. For 512 processors, the simulations are identical to the corresponding simulations from the weak scalability test.

In the weak and strong scaling tests each simulation executes 20 iteration steps with each of the three fluid LS evolution equations, during which three reinitialization solves (each consisting of 10 iterations) occur after the 10th and 20th LS iteration steps, respectively. In addition, at the end of each LS iteration-time step, the simulations also include the projection step [35] and, if applicable, the LVC algorithm. Overall, the three-phase simulations carry out 120 iterations, of which 60 are main LS iterations and 60 are reinitialization iterations. These fractions are representative for a typical usage of our code. All computations were performed on the supercomputer *Fram* provided by UNINETT Sigma2 – the National Infrastructure for High Performance Computing and Data Storage in Norway.

Results for the weak scaling are presented in Table 1. With a perfect (ideal) parallel speedup the absolute timing for each of these simulations should equal the timing obtained for the corresponding simulation (i.e., with an equal number of conserved phases and isolated domains) on a single processor. In the case without LVC we observe a very good parallel performance. However, in our opinion the results are also satisfactory when we include LVC on one or two phases, even for

Table 2
Strong scaling results for various cases of conserved phases and number of isolated domains.

Absolute timing (s) for strong scalability tests

# cores	Without LVC	LVC 1 phase (32 domains)	LVC 2 phases (64 domains)	LVC 1 phase (108 domains)	LVC 2 phases (216 domains)
128	61.26	71.28	80.99	74.94	87.81
256	31.88	38.4	44.43	43.21	54.14
512	17.33	21.14	24.83	27.02	36.19
1024	11.04	14.0	17.2	19.77	27.85

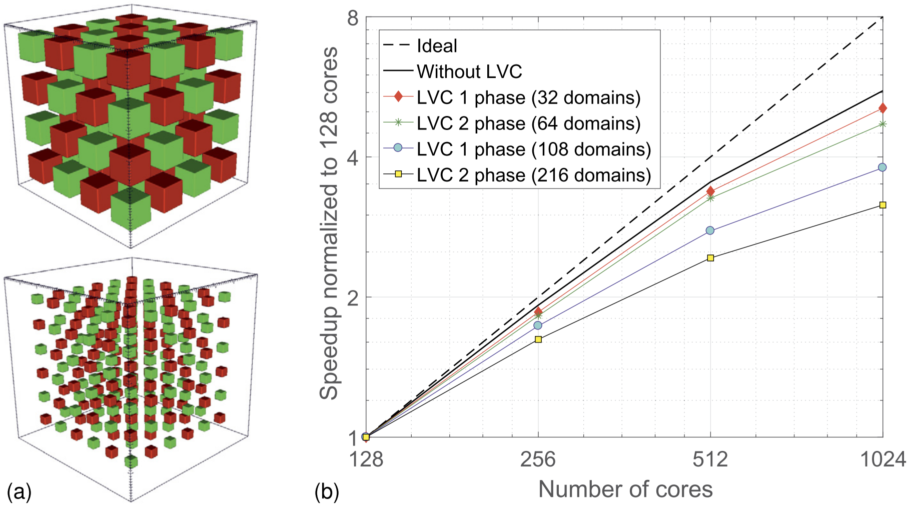


Fig. 4. (a) Configurations of the disconnected phases (oil in red and gas in green) used in the scaling tests. Each phase contains 32 domains (top) and 108 domains (bottom). The intermediate continuous phase is transparent. Simulations with LVC applied to one phase conserve the red domains, while simulations with LVC applied to two phases conserve both red and green domains. (b) Speedup normalized to 128 computing cores (processors) as a function of number of cores from the strong scaling tests.

a high number of isolated domains. Even though the timings obtained with 216 domains may seem a bit far from ideal for 64 and 512 processors, they still represent efficient and useful high performance computations.

The results from the strong scaling tests are presented in Table 2 and Fig. 4(b). We observe that the parallel speedup is very good for the case without LVC. For the cases with LVC the parallel speedup decreases with an increasing number of isolated domains present in the system. This is due to parallel communication introduced by the LVC algorithm. However, the parallel speedup obtained is still absolutely acceptable and definitely demonstrates that our codes run efficiently on modern supercomputers.

6. Simulations

In the numerical model given by Eqs. (3), (9) and (10), we introduce the dimensionless variables $\hat{\gamma}_\alpha = \gamma_\alpha/\gamma^*$, $\hat{p}_\alpha = p_\alpha/p^*$, $\hat{\phi}_\alpha = \phi_\alpha/L^*$, and $\hat{x} = x/L^*$, where L^* and γ^* are the characteristic length and surface tension, respectively, and $p^* = \gamma^*/L^*$ is the characteristic pressure following Young-Laplace's Eq. (4). It then follows that the dimensionless form of Eq. (3) is unchanged with the iterative time unit chosen as $(L^*)^2/\gamma^*$. Clearly, this will not affect the capillary equilibrium solutions. Thus, hereafter we treat γ_α, p_α, ϕ_α and x as dimensionless variables.

We demonstrate the LVC algorithm on different pore geometries using a fluid system of gas (g), oil (o) and water (w) with physically realistic interfacial tensions $\sigma_{gw} = 0.03, \sigma_{ow} = 0.02$ and $\sigma_{go} = 0.011$ N/m. The characteristic surface tension γ^* is 0.01 N/m, which together with Eq. (6) implies that $\gamma_w = 1.95$, $\gamma_o = 0.05$ and $\gamma_g = 1.05$. We will also ensure that all nonzero γ_α are sufficiently larger than Δx, to obtain correct behaviour in the limit $\gamma_\alpha \to 0$. For a discussion of vanishing viscosity solutions with level set methods we refer to [36].

For a three-phase fluid system in thermodynamic equilibrium with a solid the three contact angles $\theta_{\alpha\beta}$ cannot be specified independently. Instead, Eq. (5) (or, equivalently Eq. (7)) implies that the interfacial tensions and contact angles satisfy

$$\sigma_{gw} \cos\theta_{gw} = \sigma_{go} \cos\theta_{go} + \sigma_{ow} \cos\theta_{ow}. \tag{23}$$

In this work, we specify θ_{ow} and calculate θ_{go} and θ_{gw} from linear $\cos\theta_{go}(\cos\theta_{ow})$- and $\cos\theta_{gw}(\cos\theta_{ow})$-relationships [45] that satisfy Eq. (23). Our model uses a corresponding set of intersection angles β_α that are consistent with these relationships [32]:

$$\beta_w = \theta_{ow}, \qquad \beta_o = 180^\circ - \theta_{ow}, \quad \text{and} \quad \beta_g = 180^\circ. \tag{24}$$

6.1. 2-D sinusoidal pore

We perform simulations on an idealized 2-D pore throat geometry to investigate effects of grid resolution on conservation of volume (area in 2-D) and pressure convergence of isolated domains. The characteristic quantities are $L^* = 1 \times 10^{-4}$ m and $p^* = \gamma^*/L^* = 100$ N/m^2. In dimensionless variables, we consider the sinusoidal pore throat

$$\Omega_P = \{(x, y) : 0 \le x \le 2,\, y_2 \le y \le y_1\}, \tag{25}$$

where

$$\begin{aligned} y_1(x) &= \frac{1}{100}\left(52 + 5\sin\left(\frac{25}{9}\pi x\right) + 8\sin\left(\frac{50}{9}\pi x\right)\right), \\ y_2(x) &= \frac{1}{100}\left(17 - 5\sin\left(\frac{25}{9}\pi x\right) - 8\sin\left(\frac{50}{9}\pi x\right)\right). \end{aligned} \tag{26}$$

An analytic signed distance function for ψ describes the idealized pore geometry in the simulations. The initial ϕ_α-configuration is given by five circular oil drops surrounded by water with different radii and centres at different positions on the axis of symmetry along the x-direction of the pore throat. Some of these circles overlap with solid, allowing for possible contact angle formation at the pore walls. Initially, gas is present in a layer with thickness $3\Delta x$ at the left boundary ($x = 0$).

A saturation-controlled approach simulates quasi-static gas invasion from $x = 0$ while the pressure of the continuous oil and water phases, p_o and p_w, are kept constant during the simulation. This amounts to solving Eq. (3) for a series of gas saturations $S_{g,k} = S_{g,1} + (k-1)\Delta S_g$, $k = 1, \ldots, m$, that corresponds to target gas volumes $V_{g,k}^{\text{target}} = S_{g,k} V_P$, where V_P is pore volume. For each saturation $S_{g,k}$ we solve Eq. (3) while updating p_g in each level set iteration step based on

$$p_g(t) = \frac{V_{g,k}^{\text{target}} - V_g(t)}{\Delta t A_g(t)}. \tag{27}$$

We use initial gas saturation $S_{g,1} = 0.05$ and saturation step $\Delta S_g = 0.02$. The converged fluid configuration from the last saturation step is the initial configuration for the next. Note that the saturation-controlled approach mimics quasi-static invasion at constant low flow rate, while it predicts capillary pressure at each stationary state [32,33]. Pressure-controlled displacement, which is the other alternative, predicts saturations for each stationary state given a series of fixed capillary pressures. Fluids can enter and leave the system at the inlet and outlet boundaries (that is, $x = 0$ or $x = 2$).

We perform simulations with $\Delta x = 0.02$, 0.01, 0.005, and 0.0025. The corresponding reinitialization frequencies of ϕ_α are 5, 10, 25 and 50 time-iteration steps. For $\Delta x = 0.01$, we use γ_α as stated above and continuous phase pressures $p_w = 1$, $p_o = 10$, and p_g obtained from Eq. (27). For the other grid resolutions, we multiply Eq. (3) by $100\Delta x$ so that the smallest γ-value (γ_o) remains equal to $5\Delta x$. Note that we still solve the dimensionless equation, this time with iterative time unit $100\Delta x(L^*)^2/\gamma^*$. While this will impact the speed of gradient descent (i.e., number of iterations), the capillary equilibrium solutions remain unaffected.

At stationary states we estimate the relative error $E[V_a]$ of simulated volumes V_a with respect to target volumes $V_a^{(0)}$ for all individual domains $a \in I_\alpha$, as well as the relative error $E[V_\alpha]$ of simulated total volume $V_\alpha = \sum_{a \in I_\alpha} V_a$ with respect to total target volume $V_\alpha^{(0)}$ given by Eq. (14). For the relative errors of the total volume we also estimate the average absolute value over the number of equilibrium states m, $E_{\text{av}}[V_\alpha] = \sum_{i=1}^{m} |E[V_\alpha]_i|/m$. We evaluate domain pressures, $p_a, a \in I_\alpha$, by means of their pressure difference to the other two continuous phases (i.e., local capillary pressure), which represents interface curvature when scaled by interfacial tension, see Eq. (4).

6.1.1. Double displacement with domain fragmentation and coalescence

First we investigate LVC of oil only, while water pressure is constant. Hence, we assume sub-resolution wetting films along the pore walls connect the water to inlet/outlet boundaries. For this purpose we use $\theta_{ow} = 20^\circ$. Calculation of the other contact angles based on Eqs. (7) and (24) yields $\theta_{go} = 24.2^\circ$ and $\theta_{gw} = 16.1^\circ$. We wish to investigate double displacements where a continuous invading phase (e.g., gas) displaces a disconnected phase cluster (e.g., oil) that in turn displaces a continuous third phase (e.g., water). Pore-scale experiments have shown that this mechanism can mobilize oil in capillary-dominated three-phase flow [4,5,15,16].

In our simulation example, gas displaces an oil drop that coalesces with static oil drops ahead of the frontal oil/water interface during the motion, while oil-drop fragmentation forms trapped oil layers in pore corners behind the trailing gas/oil interface. Fig. 5 shows only small grid resolution effects on the stationary fluid configurations from these gas-invasion simulations, except for oil drop coalescence events that in some cases occur at slightly different gas saturations. As expected, the capillary pressures $P_{aw}, a \in I_o$, between static oil drops and surrounding water are constant and display excellent convergence behaviour with decreasing Δx, see Fig. 6(a). For the oil drop displaced by gas, the local capillary pressures P_{aw} and

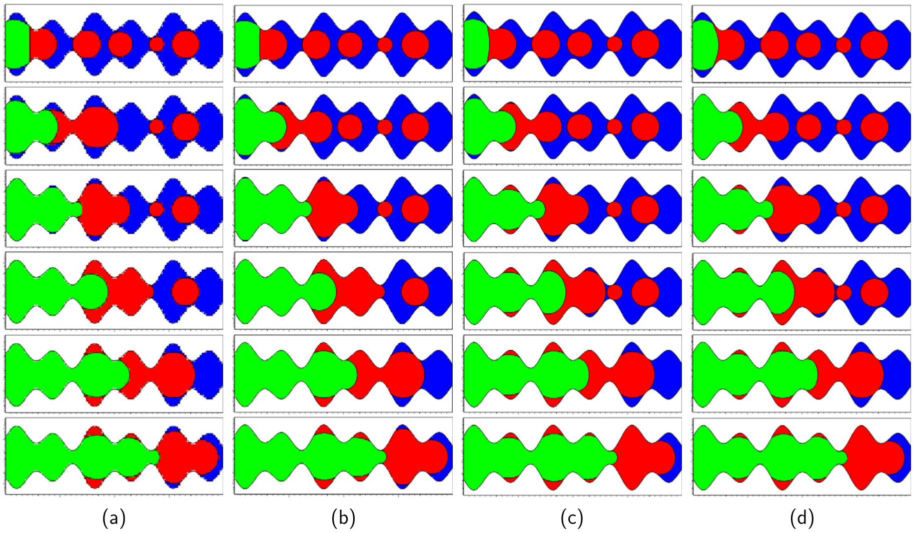


Fig. 5. Effect of grid resolution on equilibrium configurations of gas (green), oil (red) and water (blue) during simulations of gas invasion in a 2-D sinusoidal pore throat, using local oil-volume conservation and $\theta_{ow} = 20°$. Grid-cell lengths Δx are (a) 0.02, (b) = 0.01, (c) 0.005 and (d) 0.0025. Gas saturations S_g are 0.13 (row 1), 0.23 (row 2), 0.33 (row 3), 0.43 (row 4), 0.53 (row 5), and 0.63 (row 6). Note that the isolated water pockets disappear as no LVC is enforced on the water phase.

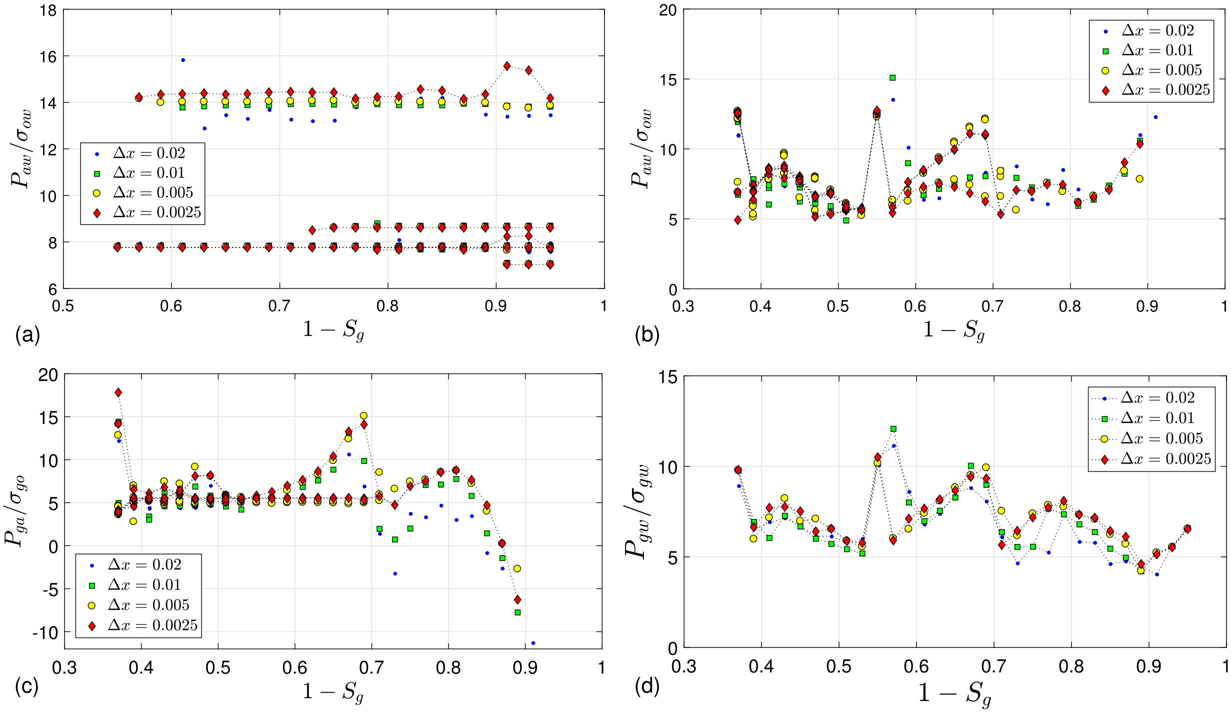


Fig. 6. Effect of grid resolution on relationships between capillary pressure (scaled by interfacial tension) and total liquid saturation from simulations of gas invasion with local oil-volume conservation in a 2-D sinusoidal pore throat, using $\theta_{ow} = 20°$. (a, b) Capillary pressure between disconnected oil drops ($a \in I_o$) and continuous water (w) for (a) all oil drops not in contact with gas, and (b) all oil drops contacting gas. (c) Capillary pressure between continuous gas (g) and disconnected oil ($a \in I_o$) for the set of oil drops contacting gas. (d) Capillary pressure between continuous gas and water. The dotted lines in (a), (b) and (c) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 5 (d).

P_{ga}, $a \in I_o$, vary during displacement, as the frontal oil/water and trailing gas/oil interfaces move through wide and narrow pore sections, see Figs. 6(b, c). We obtain excellent capillary pressure convergence in this case as well, despite coalescence and splitting events occur during the displacement. Further, Fig. 6(c), shows that the capillary pressures P_{ga}, $a \in I_o$ for the oil layers trapped by gas in the pore corners behind the front stabilize at constant values. The capillary pressure between the continuous gas and water phases P_{gw} obtains local maxima when either the frontal oil/water interface or trailing gas/oil

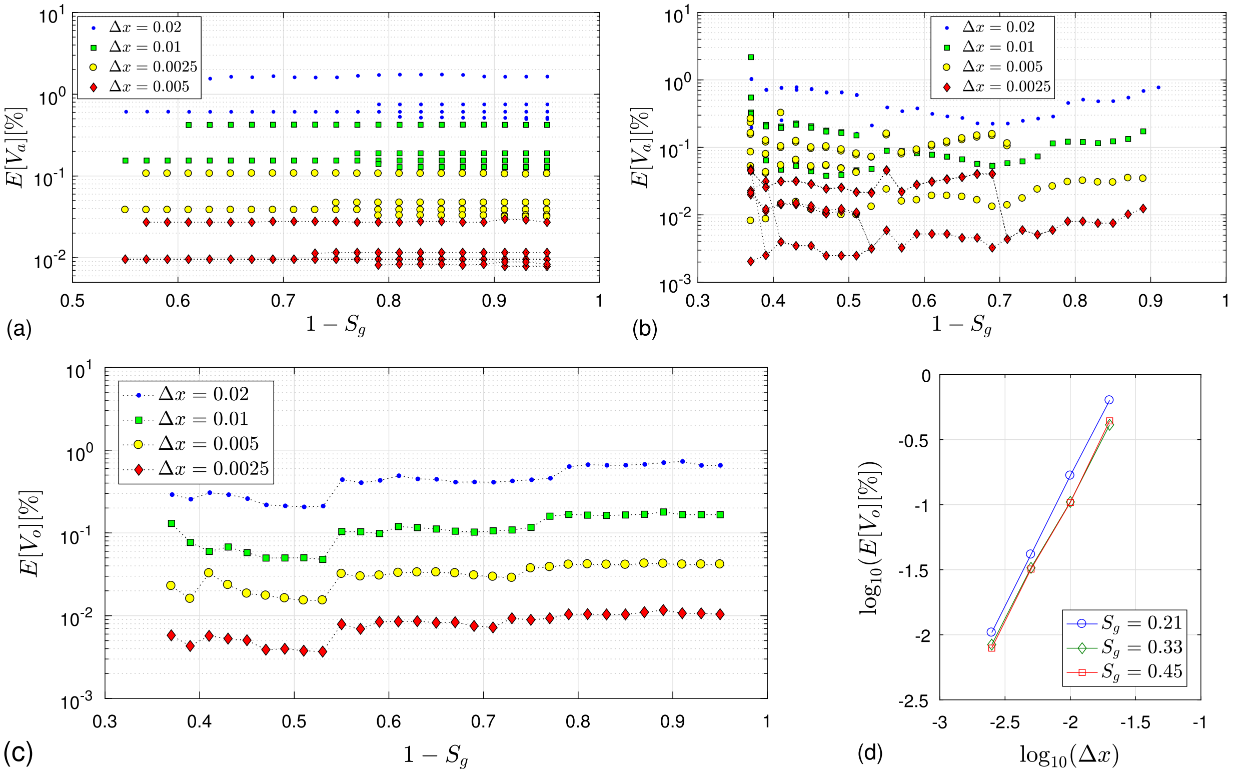


Fig. 7. Effect of grid resolution on oil-volume relative error as a function of total liquid saturation from simulations of gas invasion with local oil-volume conservation in a 2-D sinusoidal pore throat, using $\theta_{ow} = 20°$. (a, b) Evolution of volume relative errors for (a) oil drops not in contact with gas, and (b) oil drops contacting gas. (c) Evolution of relative error for the total disconnected oil volume. (d) Relative error for the total disconnected oil volume as a function of grid-cell length for equilibrium configurations at three gas saturations. The dotted lines in (a) and (b) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 5 (d).

interface invades the narrowest pore sections. Although P_{gw} varies with gas saturation, Fig. 6(d) shows that the effects of Δx on P_{gw} are small.

Figs. 7(a, b) show that the domain volume relative errors $E[V_a]$, $a \in I_o$, decrease consistently with decreasing Δx. For the static oil drops surrounded by water $E[V_a]$ remains constant during gas invasion, see Fig. 7(a), while for the oil drops contacting gas $E[V_a]$ varies more with gas saturation due to displacement, coalescence and splitting events (Fig. 7(b)). Fig. 7(c) shows that the relative error for the total disconnected oil volume $E[V_o]$ behaves similarly. Examining the behaviour at three gas saturations shows that $E[V_o]$ decreases quadratically with decreasing Δx, see Fig. 7(d).

We now investigate the above example with LVC applied to both oil and water in the simulations. The sequence of stable fluid configurations presented in Fig. 8 shows that the oil drop displaced by gas still merges with the static oil drops surrounded by continuous water during its motion. However, this time isolated water domains break up from the continuous water and trap in pore corners surrounded by gas or oil behind the frontal oil/water interface. For the lowest grid resolution ($\Delta x = 0.02$) two oil drops completely isolate some water from its continuous phase, leading to a different displacement pattern than for $\Delta x \leq 0.01$. Formation of oil layers between bulk gas and corner water is unlikely to occur in this example because the fluid system represents a non-spreading oil ($\sigma_{gw} - \sigma_{go} - \sigma_{ow} < 0$) with unfavourable contact angles ($\theta_{go} > \theta_{ow}$) and because the amount of water trapped in pore corners is large. Instead, triple junctions form as the trailing gas/oil interface passes water domains trapped in the corners, see Fig. 8.

Fig. 9 investigates accuracy of capillary pressure between isolated oil drops $a \in I_o$ and continuous gas and water phases as Δx decreases. The set of oil drops not in contact with gas converge to constant capillary pressure levels as before, see Fig. 9(a). However, Figs. 9(b, c) show that the capillary pressures for oil drops contacting gas vary slightly more with Δx than in the previous case where LVC was applied to oil only. This is also the case for the capillary pressure between continuous gas and water, P_{gw}, as shown in Fig. 9(d). We anticipate this is an effect of preserving the volumes of coarsely resolved water domains trapped in the pore corners, which in turn will affect the pressure calculations of both the water and oil domains as well as the continuous gas. However, the relative errors for the individual oil-drop volumes, $E[V_a]$, $a \in I_o$, and the relative error for the total disconnected oil volume, $E[V_o]$, show the same consistent trends as before, see Fig. 10.

Fig. 11 presents capillary pressure and volume preservation properties for the disconnected water. The capillary pressures between continuous gas and water domains $a \in I_w$ exhibit some variations with Δx due to grid-resolution effects of the preserved water layers in the pore corners, see Fig. 11(a). Further, the relative error of individual water volumes

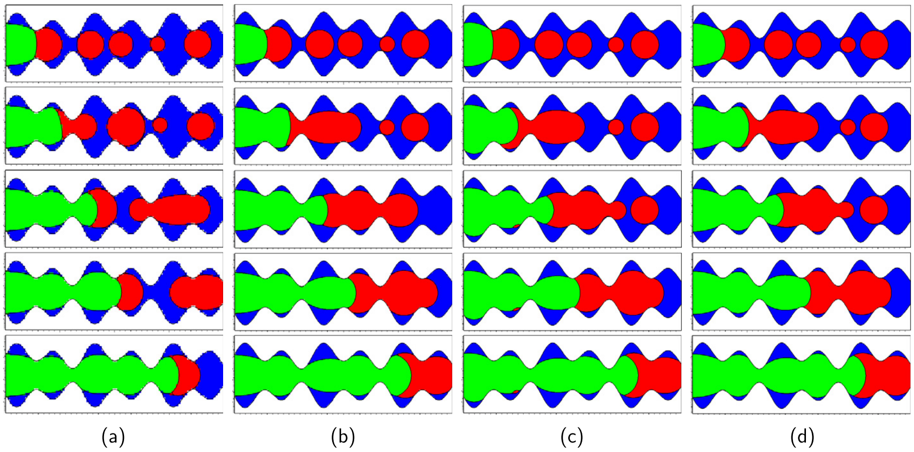


Fig. 8. Effect of grid resolution on equilibrium configurations of gas (green), oil (red) and water (blue) during simulations of gas invasion in a 2-D sinusoidal pore throat, using LVC of oil and water, and $\theta_{ow} = 20°$. Grid-cell lengths Δx are (a) 0.02, (b) = 0.01, (c) 0.005 and (d) 0.0025. Gas saturations S_g are 0.13 (row 1), 0.23 (row 2), 0.33 (row 3), 0.43 (row 4), and 0.63 (row 5).

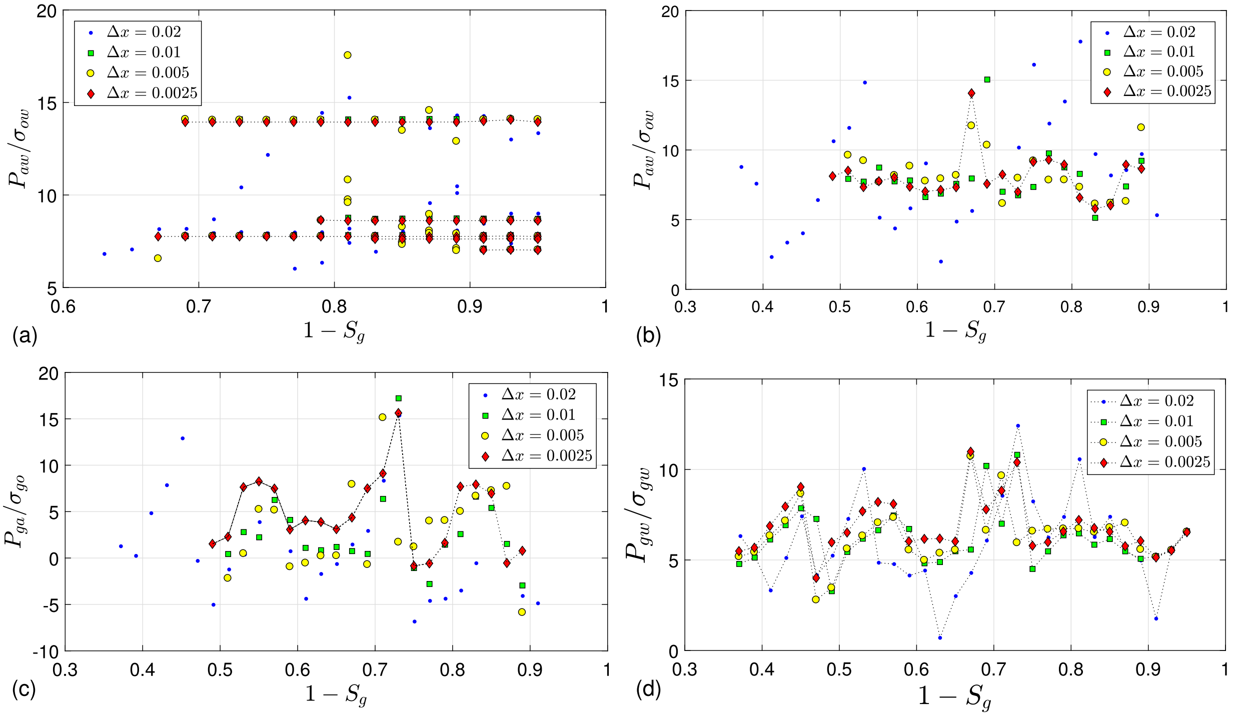


Fig. 9. Effect of grid resolution on relationships between capillary pressure (scaled by interfacial tension) and total liquid saturation from simulations of gas invasion with LVC of oil and water in a 2-D sinusoidal pore throat, using $\theta_{ow} = 20°$. (a, b) Capillary pressure between disconnected oil drops ($a \in I_o$) and continuous water (w) for (a) oil drops not in contact with gas, and (b) oil drops contacting gas. (c) Capillary pressure between continuous gas (g) and disconnected oil ($a \in I_o$) for the set of oil drops contacting gas. (d) Capillary pressure between continuous gas and water. The dotted lines in (a), (b) and (c) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 8 (d).

$E[V_a]$, $a \in I_w$, decreases consistently with Δx, see Fig. 11(b). However, Fig. 11(c) shows that the relative errors for the total disconnected water volume, $E[V_w]$, are similar for $\Delta x = 0.02$ and 0.01. This is because the configurations of water differ significantly; for $\Delta x = 0.02$ a large water domain spans the pore cross-section during a significant part of the gas invasion, while for $\Delta x \leq 0.01$ all isolated water domains exist as layers in the pore corners. Thus, for $\Delta x \leq 0.01$, $E[V_w]$ decreases quadratically with decreasing Δx, as shown in Fig. 11(d).

6.1.2. Displacement chains with multiple domains

Here we consider gas invasion in the sinusoidal pore with LVC of oil and water under a more weakly water-wet state, represented by the consistent contact angle set $\theta_{ow} = 60°$, $\theta_{go} = 21.3°$ and $\theta_{gw} = 47.5°$. A large oil/water contact angle

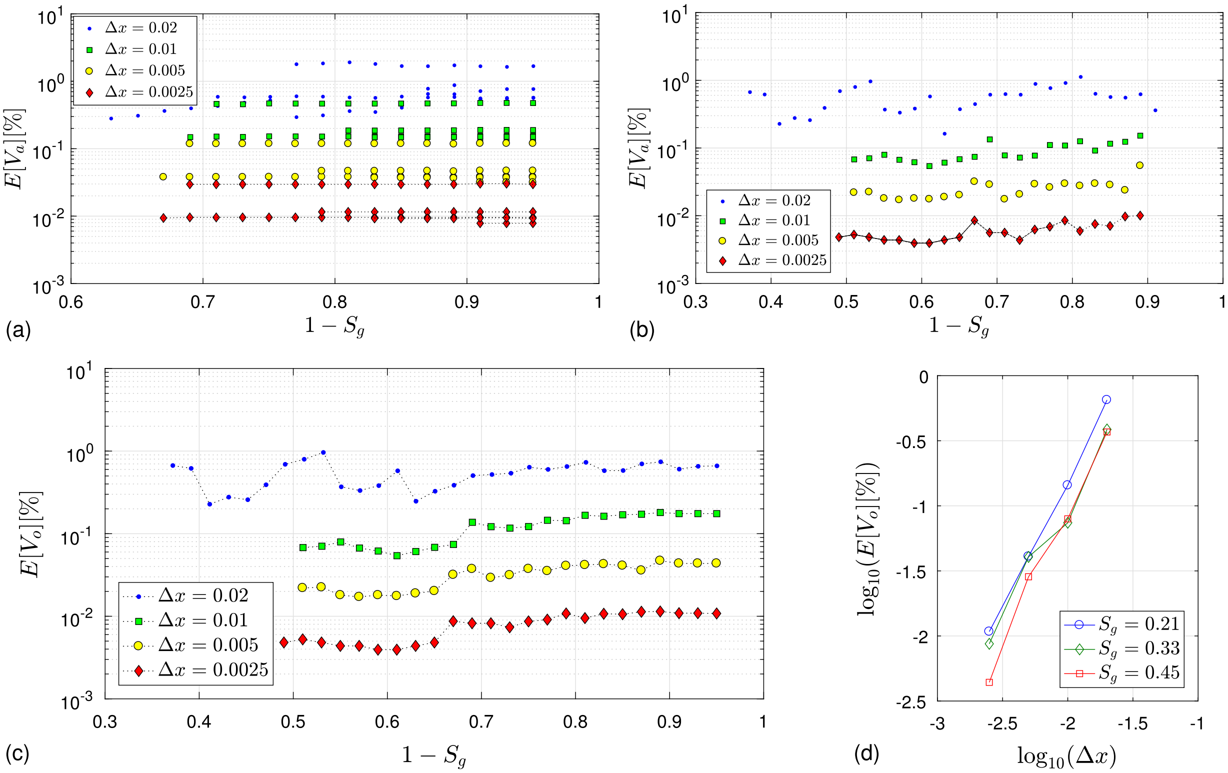


Fig. 10. Effect of grid resolution on oil-volume relative error as a function of total liquid saturation from simulations of gas invasion with local oil- and water-volume conservation in a 2-D sinusoidal pore throat, using $\theta_{ow} = 20°$. (a, b) Evolution of volume relative errors for (a) oil drops not in contact with gas, and (b) oil drops contacting gas. (c) Evolution of relative error for the total disconnected oil volume. (d) Relative error for the total disconnected oil volume as a function of grid-cell length for equilibrium configurations at three gas saturations. The dotted lines in (a) and (b) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 8 (d).

($\theta_{ow} > \theta_{go}$) favours formation of oil layers between bulk gas and water in pore corners and allows the oil drops to contact pore walls and isolate intermediate water domains. Then, invasion of continuous gas can displace a chain of several isolated water and oil domains that in turn displaces the water phase connected with outlet, while oil and water layers deposit in pore corners behind the trailing gas/oil interface. Such displacement chains can mobilize disconnected oil ganglia over multiple water-alternate-gas invasion cycles, as observed in micromodel pore-scale experiments [9–11].

In our example (for $\Delta x \leq 0.01$), Fig. 12 shows that the two leftmost oil drops contact the pore walls and isolate a water domain from the continuous water phase, resulting in the displacement chain $g \to o \to w \to o \to w$. The three rightmost oil drops are static as they are completely surrounded by continuous water. However, during gas invasion, these oil drops coalesce with the nearest oil drop approaching from the left, while the leftmost oil drop and the water domain split to form oil and water layers in pore corners behind the trailing gas/oil interface. Note that the mobile oil drop surrounded by isolated water to the left and connected water to the right has different interface curvature (that vary during displacement) on the left and right sides due to significantly different pressures in the water domain and the continuous water phase. Further, Fig. 12 shows that simulations with $\Delta x \leq 0.005$ capture configurations with oil layers sandwiched between corner water and bulk gas, whereas the coarse-grid simulation ($\Delta x = 0.02$) exhibits significant deviations to the other results.

Fig. 13 explores accuracy of the capillary pressures between disconnected oil drops $a \in I_o$ and continuous gas and water phases with decreasing Δx. Fig. 13(a) shows that oil/water capillary pressures for the subset of oil drops surrounded by water converge as Δx decreases for both the static oil drops where capillary pressures are constant and the mobile oil drops where capillary pressures vary with displacement. Figs. 13(b) and (c) demonstrate convergence of capillary pressures with decreasing Δx for the oil drops contacting gas. However, a few scattered data points deviate from this behaviour due to coarse grid resolution of oil layers in pore corners. For $\Delta x = 0.005$ oil layers break into two oil drops on the gas/water interfaces as the trailing gas/oil interface moves into narrower pore cross-sections accompanied by higher gas pressure. These oil drops attain a high and negative gas/oil capillary pressure and a corresponding high and positive oil/water capillary pressure. The oil layers repair themselves with normalized domain capillary pressures when the trailing gas/oil interface moves into wider pore regions accompanied by lower gas pressure. For $\Delta x = 0.0025$ oil layers are unaffected by these gas pressure variations that depend on the position of the trailing gas/oil interface. Fig. 13(d) shows that the capillary pressure between continuous gas and water phases, P_{gw}, converges consistently with decreasing Δx.

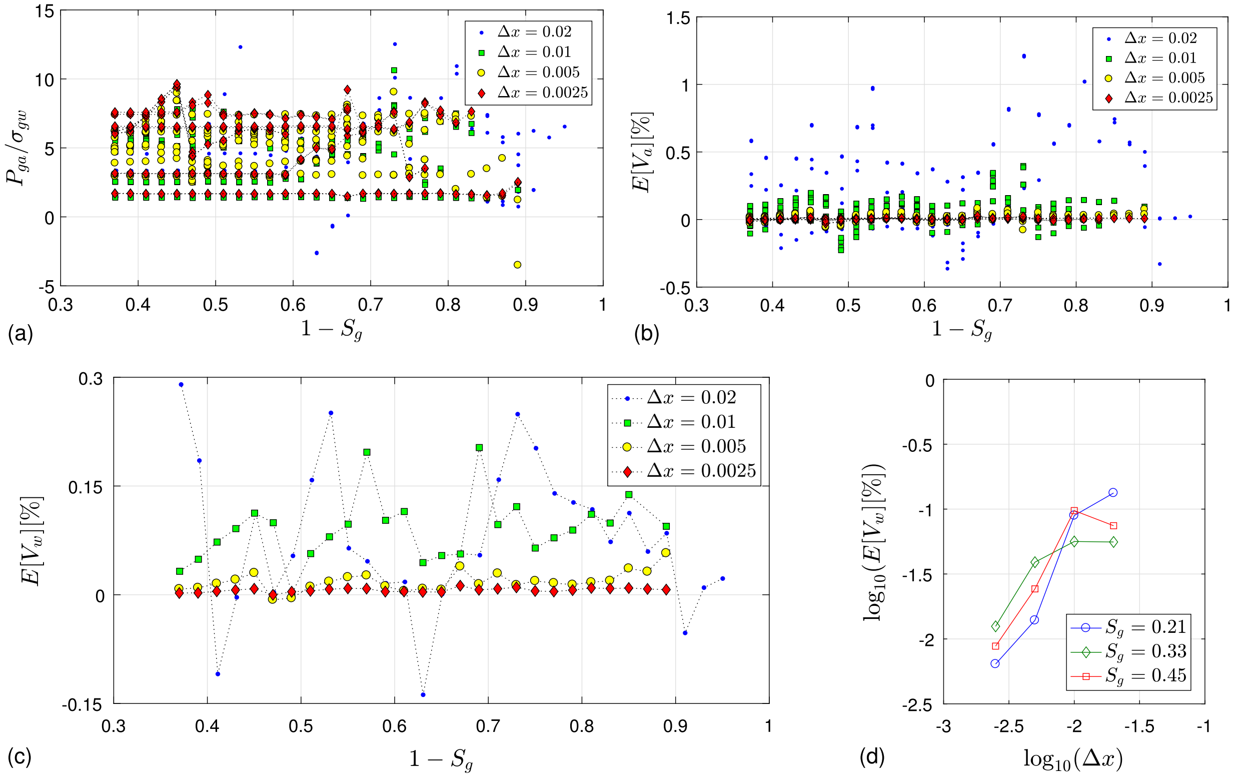


Fig. 11. Effect of grid resolution on capillary pressure and water-volume relative error as a function of total liquid saturation from simulations of gas invasion with local oil- and water-volume conservation in a 2-D sinusoidal pore throat, using $\theta_{ow}=20°$. (a) Evolution of capillary pressure (scaled by interfacial tension) between continuous gas (g) and disconnected water drops ($a \in I_w$). (b) Evolution of volume relative errors for the individual water drops. (c) Evolution of relative errors for the total disconnected water volume. (d) Relative error for the total disconnected water volume as a function of grid-cell length for equilibrium configurations at three gas saturations. The dotted lines in (a) and (b) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 8 (d).

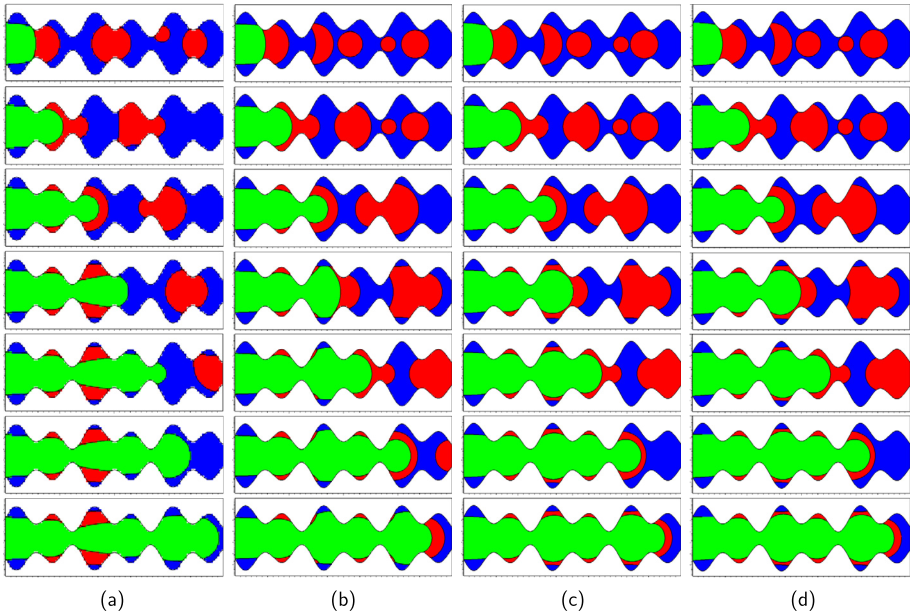


Fig. 12. Effect of grid resolution on configurations of gas (green), oil (red) and water (blue) during simulations of gas invasion in a 2-D sinusoidal pore throat, using local conservation of oil and water volumes, and $\theta_{ow}=60°$. Grid-cell lengths Δx are (a) 0.02, (b) = 0.01, (c) 0.005 and (d) 0.0025. Gas saturations S_g are 0.13 (row 1), 0.23 (row 2), 0.33 (row 3), 0.43 (row 4), 0.55 (row 5), 0.67 (row 6), and 0.79 (row 7).

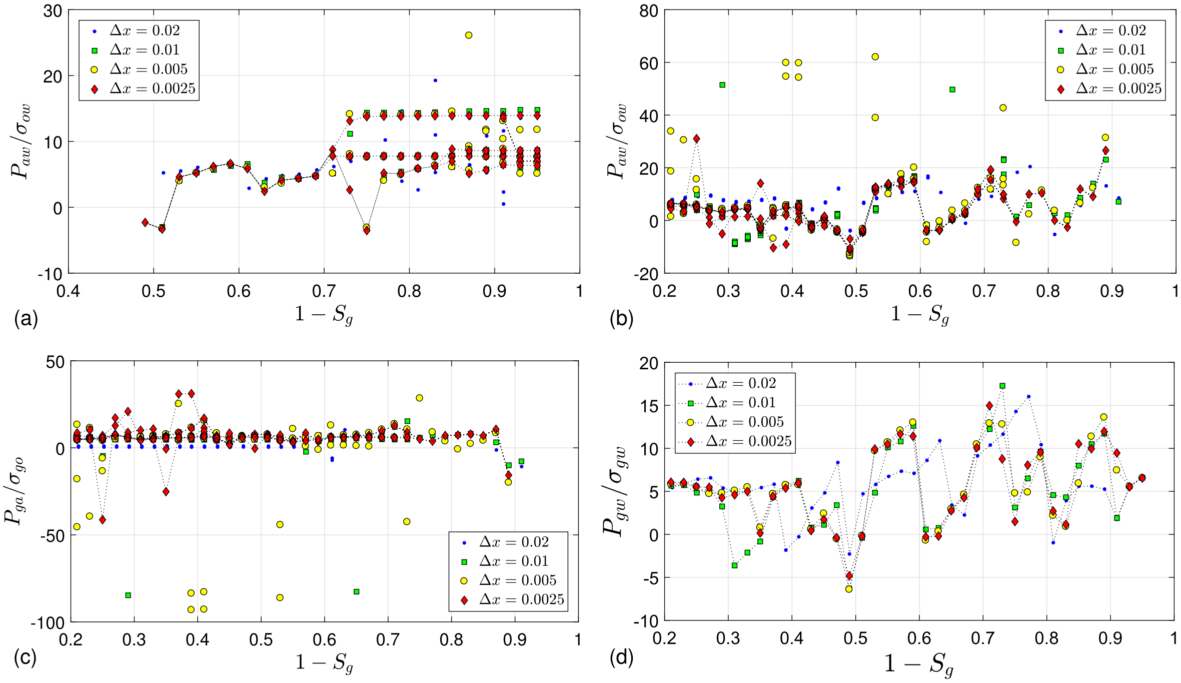


Fig. 13. Effect of grid resolution on relationships between capillary pressure (scaled by interfacial tension) and total liquid saturation from simulations of gas invasion with LVC of oil and water in a 2-D sinusoidal pore throat, using $\theta_{ow} = 60°$. (a, b) Capillary pressure between disconnected oil drops ($a \in I_o$) and continuous water (w) for (a) oil drops not in contact with gas, and (b) oil drops contacting gas. (c) Capillary pressure between continuous gas (g) and disconnected oil ($a \in I_o$) for the set of oil drops contacting gas. (d) Capillary pressure between continuous gas and water. The dotted lines in (a), (b) and (c) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 12 (d).

The relative errors for the individual oil drop volumes show the same consistent behaviour with decreasing Δx as the previous examples, see Figs. 14(a) and (b). However, a few high domain-volume errors occur for $\Delta x = 0.005$ due to the coarse resolution of oil layers contacting gas, see Fig. 14(b). These volumes are small and do not contribute much to the relative error of the total disconnected oil volume, $E[V_o]$, as shown in Fig. 14(c). Fig. 14(d) and (e) shows that $E[V_o]$ for three selected equilibrium states as well as the average relative error over all equilibrium states, $E_{\text{av}}[V_o]$, decreases quadratically with decreasing Δx.

Fig. 15 presents the results for the water domains. The capillary pressures between continuous gas and the water domains converge with decreasing Δx. However, a few small water layers with low grid resolution give a couple of deviating data points. The water domains exhibit a similar volume-conservation behaviour with decreasing Δx as observed for the oil phase.

6.2. 2-D synthetic porous medium

Here, we test the LVC algorithm on a 2-D synthetic porous medium with characteristic length $L^* = 1 \times 10^{-4}$ m, consisting of 110 circles (representing solid phase) with dimensionless radii in the range $[0.04, 0.07]$. The circle centres are placed at uniformly distributed random positions within the computational domain $\Omega \in [0, 1] \times [0, 2]$ without eliminating possible circle overlaps. The space outside the circles represents pore space. First we use the two-phase LS method with LVC of water and simulate oil invasion from $y = 0$ into a fully water-filled pore space (primary drainage), for both $\theta_{ow} = 20°$ and 60°. From three different oil saturations established during primary drainage we simulate gas invasions at constant P_{ow}, also from $y = 0$, using MLS method with LVC of both oil and water. For primary drainage, we simulate saturation-controlled oil invasion with $\Delta S_o = 0.02$ and update oil/water interface curvature continuously according to

$$C_{ow}(t) = \frac{V_{o,k}^{\text{target}} - V_o(t)}{\Delta t A_o(t)}, \tag{28}$$

given a set of target oil volumes $V_{o,k}^{\text{target}}$ that correspond to target oil saturations $S_{o,k}$. The following gas invasions occur in a similar manner, using $\Delta S_g = 0.01$ and Eq. (27). For the LS functions, we use reflective boundary conditions except at the inlet ($y = 0$) where we add a pore-space layer with thickness $2\Delta x$ occupied by invading phase and use linear extrapolation. For the phases subjected to LVC, we assign the continuous phase pressure to isolated domains connected to boundaries $y = 0$ (inlet) and $y = 2$ (outlet), while we preserve domains contacting the other boundaries ($x = 0$ and $x = 2$). The LS functions are reinitialized, according to Eq. (19), for each 10th iteration steps.

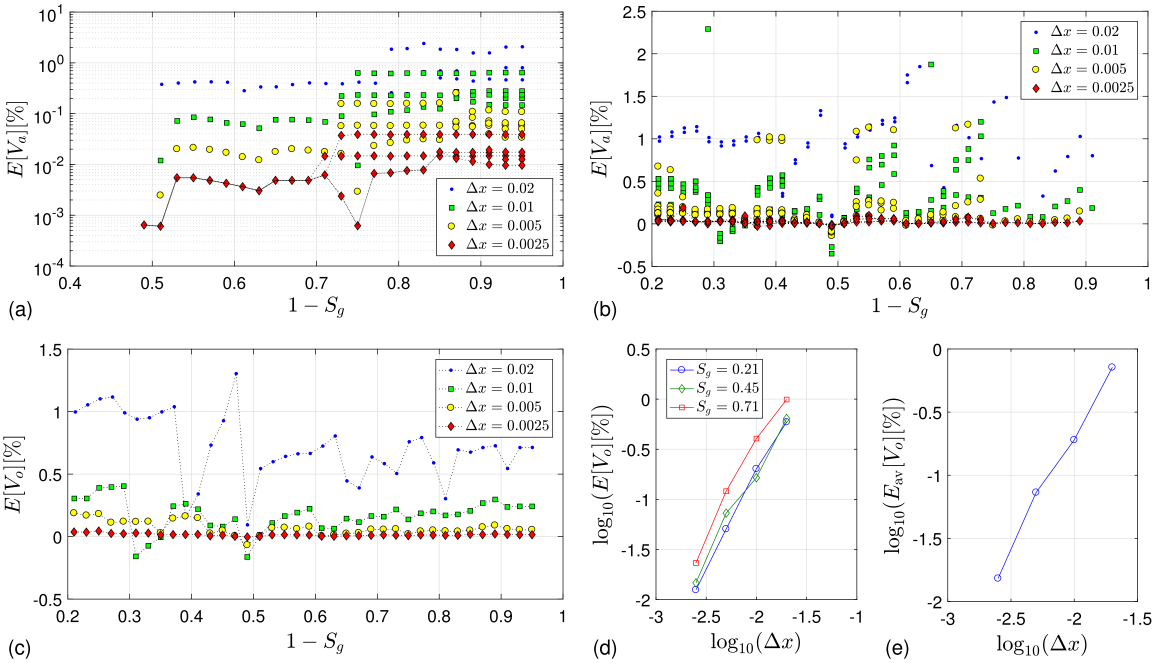


Fig. 14. Effect of grid resolution on oil-volume relative error as a function of total liquid saturation from simulations of gas invasion with local oil- and water-volume conservation in a 2-D sinusoidal pore throat, using $\theta_{ow} = 60°$. (a, b) Evolution of volume relative errors for (a) oil drops not in contact with gas, and (b) oil drops contacting gas. (c) Evolution of relative error for the total disconnected oil volume. (d) Relative error for the total disconnected oil volume as a function of grid-cell length for equilibrium states at three gas saturations. (e) Average relative error over all equilibrium states for the total disconnected oil volume as a function of grid-cell length. The dotted lines in (a) and (b) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 12 (d).

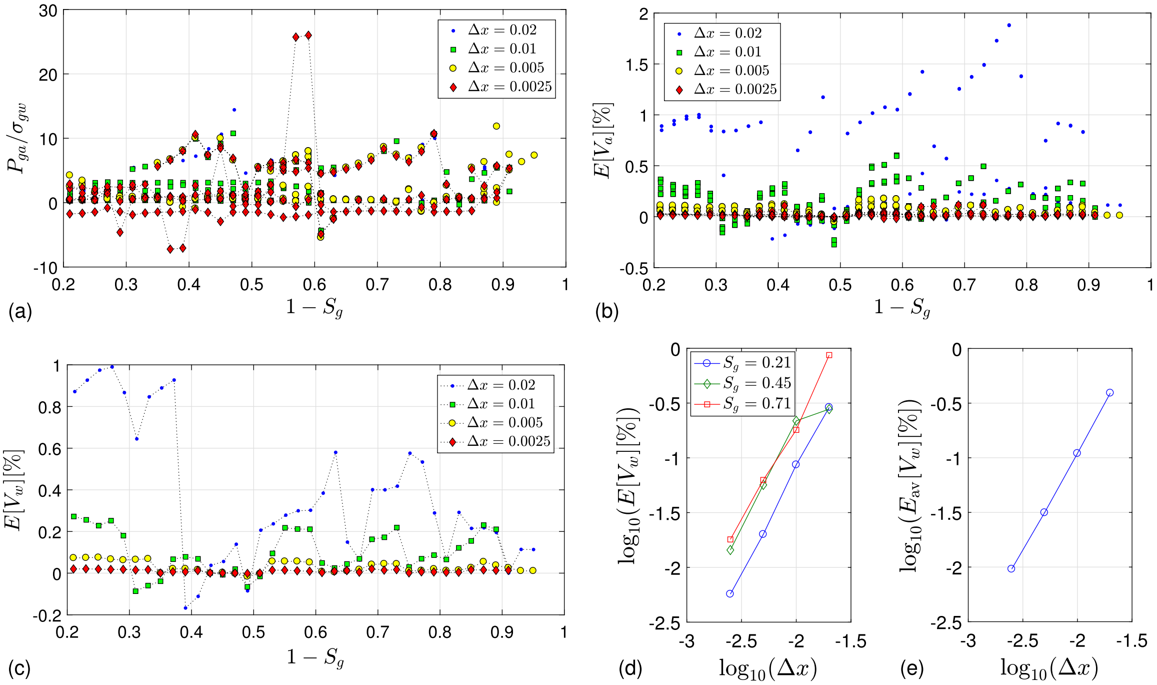


Fig. 15. Effect of grid resolution on capillary pressure and water-volume relative error as a function of total liquid saturation from simulations of gas invasion with local oil- and water-volume conservation in a 2-D sinusoidal pore throat, using $\theta_{ow} = 60°$. (a) Evolution of capillary pressure (scaled by interfacial tension) between continuous gas (g) and disconnected water drops ($a \in I_w$). (b) Evolution of volume relative errors for the individual water drops. (c) Evolution of relative errors for the total disconnected water volume. (d) Relative error for the total disconnected water volume as a function of grid-cell length for equilibrium configurations at three gas saturations. (e) Average relative error over all equilibrium states for the total disconnected water volume as a function of grid-cell length. The dotted lines in (a) and (b) show the evolution of individual drops in the simulation with $\Delta x = 0.0025$, shown in Fig. 12 (d).

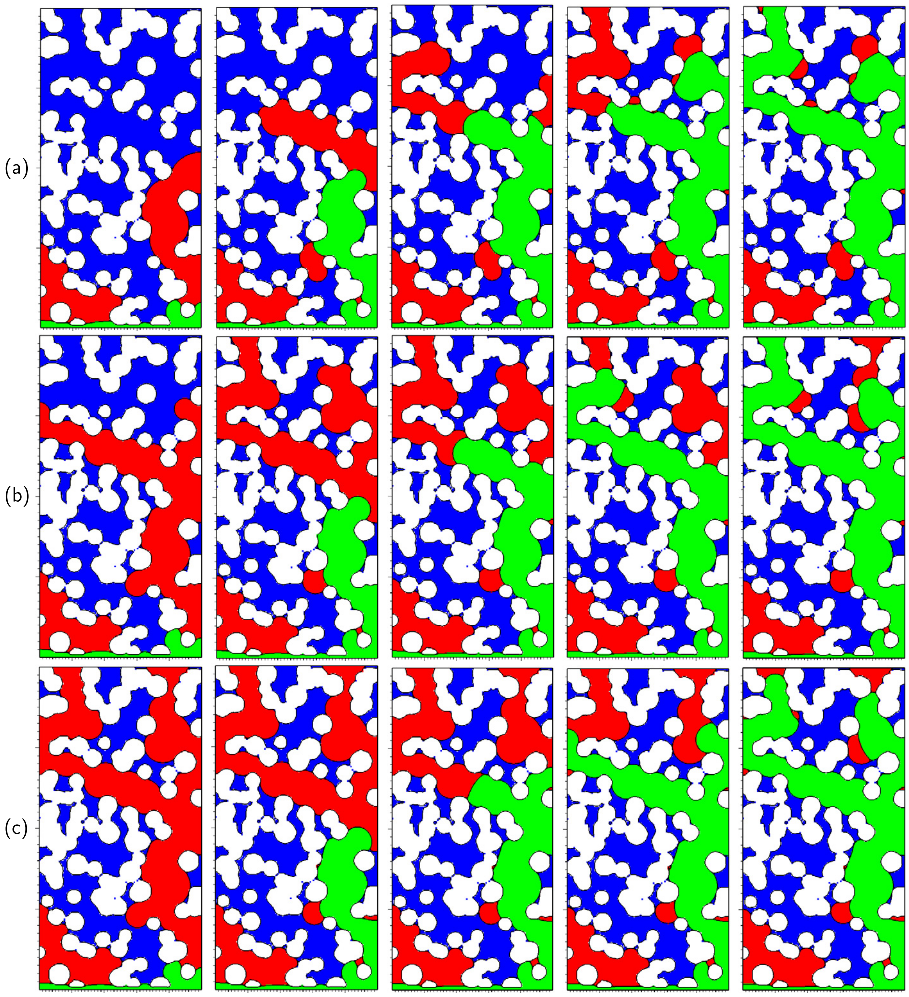


Fig. 16. Stable configurations of gas (green), oil (red) and water (blue) in the 2-D synthetic porous medium during gas invasion from initial oil saturations 0.25 (a), 0.45 (b), and 0.63 (c). Oil/water contact angle is 20°. The gas saturations are 0.02 (first column), 0.17 (second column), 0.27 (third column), 0.37 (fourth column), and 0.47 (fifth column).

The left columns of Figs. 16 and 17 show the initial oil and water configurations prior to gas invasion for $\theta_{ow} = 20°$ and $60°$, respectively. As we expect, the amount of water left behind the front during primary drainage is higher for $\theta_{ow} = 20°$ than for $\theta_{ow} = 60°$ although both simulations trap a large water domain in the lower left part of the pore geometry. These configurations also show that the oil phase has a different interface curvature to the continuous water phase than to the various disconnected water regions. Fig. 18 shows the distribution of domain capillary pressures compared with the continuous-phase capillary pressure curve (in physical units). The capillary pressures for isolated water domains are significantly lower for $\theta_{ow} = 60°$ than for $\theta_{ow} = 20°$. Several water domains stabilize at negative capillary pressures due to the higher contact angle.

Gas invasion from the two lowest oil saturations occurs by double displacement of a large disconnected oil cluster, while smaller oil domains and layers get trapped between disconnected water and continuous gas, see Figs. 16(a, b) and 17(a, b). Branching of the gas phase triggers another double displacement of oil toward outlet boundary at the upper right part of the pore geometry. The number of trapped oil layers left behind the invading gas is higher for $\theta_{ow} = 60°$ than for $\theta_{ow} = 20°$ because oil invades narrower pore constrictions during primary drainage under weak water-wet states. Gas invasion at high initial oil saturation occurs by direct gas/oil displacement, while fragmentation of continuous oil traps oil domains behind the front, see Figs. 16(c) and 17(c). The extent of displacement chains with multiple domains is limited in these simulations. The most conspicuous example is displacement $g \to o \to w \to o$, as highlighted by an arrow in the fourth image of Fig. 17(c). This is consistent with micromodel studies where such multiple displacements occur more frequently for subsequent water- and gas-invasion cycles [9–11].

The three-phase configurations in Figs. 16 and 17 show high variation of interface curvatures between continuous gas and disconnected water and oil domains. Figs. 19(a) and (b) quantify gas/water capillary pressures for continuous water and disconnected water domains from gas invasion simulations with initial oil saturation 0.25. Because the case with $\theta_{ow} = 60°$

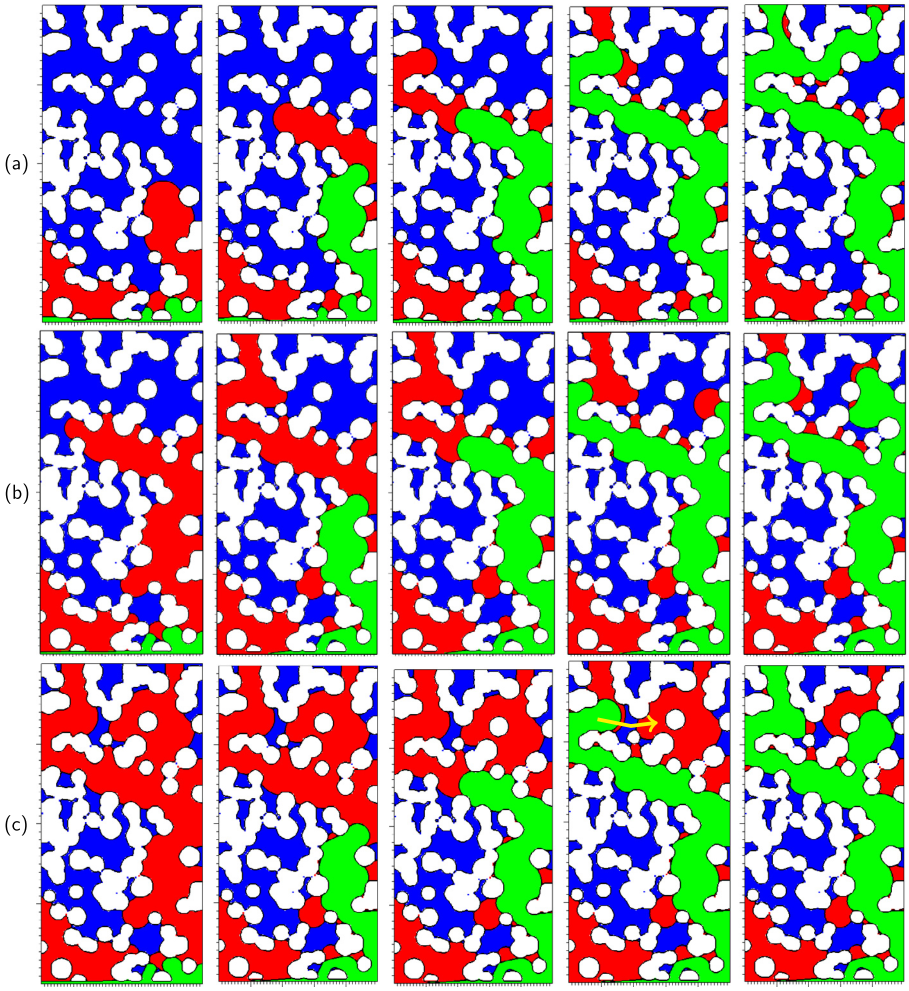


Fig. 17. Stable configurations of gas (green), oil (red) and water (blue) in the 2-D synthetic porous medium during gas invasion from initial oil saturations 0.25 (a), 0.45 (b), and 0.72 (c). Oil/water contact angle is 60°. The gas saturations are 0.02 (first column), 0.17 (second column), 0.27 (third column), 0.37 (fourth column), and 0.47 (fifth column). (c) The yellow arrow in fourth image from left highlights a displacement chain with two disconnected domains ($g \to o \to w \to o$).

has high gas/water contact angle ($\theta_{gw} = 47.5°$), more water domains have lower (and even negative) capillary pressure than the case with $\theta_{ow} = 20°$ where $\theta_{gw} = 16.1°$. Figs. 19(c) and (d) present gas/oil capillary pressure with respect to continuous oil and disconnected oil domains for the gas invasion simulations from endpoint oil saturations 0.63 ($\theta_{ow} = 20°$) and 0.72 ($\theta_{ow} = 60°$) after primary drainage. Note that several domain capillary pressures are negative despite gas/oil contact angles are small ($\theta_{go} = 21.3°$ and 24.2°). This is possible when gas/oil/water triple lines with water cusps, rather than gas/oil/solid contact lines, form on the pore walls. For such instances, configurations of isolated oil domains resemble oil drops (confined by pore geometry) resting on gas/water interfaces with negative gas/oil capillary pressures and positive oil/water capillary pressures. The ranges of gas/oil capillary pressures for oil domains are similar for both $\theta_{ow} = 20°$ and $\theta_{ow} = 60°$ because the gas/oil contact angles in the two cases are about the same.

6.3. 3-D porous medium

We use a pore structure extracted from a segmented microtomography data set of Castlegate sandstone [46] to explore effects of LVC in simulations of capillary pressure curves on realistic 3-D porous media. The subset contains 128^3 voxels, where physical voxel length is 5.6 µm. Thus, $L^* = 7.168 \times 10^{-4}$ m. The simulations use contact angles $\theta_{ow} = 40°$, $\theta_{go} = 23.1°$, and $\theta_{gw} = 32°$. We use reflective boundary conditions for the LS functions at all faces of the computational domain, except at inlet ($z = 0$), where we add a pore-space layer (thickness $2\Delta x$) occupied by invading phase and use linear extrapolation. For phases with LVC we assign the continuous-phase pressure to domains connected with top or bottom boundaries of the sample, while we preserve domains contacting side walls. We simulate pressure-controlled displacement

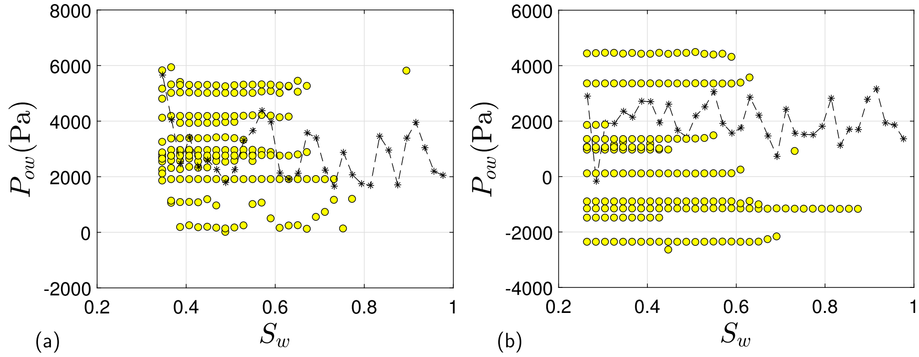


Fig. 18. Oil/water capillary pressure (in physical units) for continuous water (asterisks) and disconnected water domains (circles) as a function of water saturation from simulations of primary drainage in a 2-D synthetic porous medium, using (a) $\theta_{ow} = 20°$ and (b) $\theta_{ow} = 60°$.

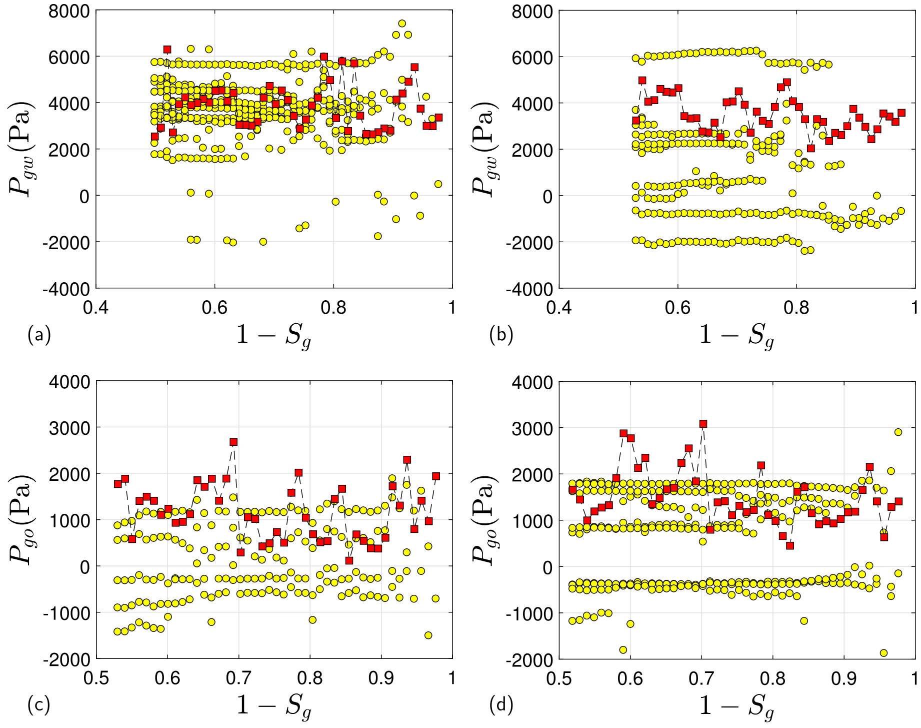


Fig. 19. Three-phase capillary pressure (in physical units) as a function of total liquid saturation from simulations of gas invasion in a 2-D synthetic porous medium with initial oil saturation (a, b) 0.25, (c) 0.63, and (d) 0.72, established after primary drainage. (a, b) Gas/water capillary pressure for continuous water (red squares) and disconnected water domains (yellow circles), using (a) $\theta_{ow} = 20°$ and (b) $\theta_{ow} = 60°$. (c, d) Gas/oil capillary pressure for continuous oil (red squares) and disconnected oil domains (yellow circles), using (c) $\theta_{ow} = 20°$ and (d) $\theta_{ow} = 60°$.

by calculating stable configurations for each stepwise increment of invading-phase pressure while the pressures of the continuous defending phases are constant. Reinitialization occurs every 5th LS iteration steps.

First we investigate effects of LVC applied to the wetting phase (water) during primary drainage, using the two-phase LS method (Section 4). Fig. 20(a) shows that LVC has little effect on capillary pressure in the high water saturation regime, but substantial differences occur for $S_w < 0.5$. This leads to significantly different initial water saturations S_{wi} (that is, endpoint saturations) in the two simulations. We obtain $S_{wi} = 0.23$ with LVC in contrast to $S_{wi} = 0.02$ without LVC. Figs. 20(b) and (c) show that the difference occurs because large water domains form midway in the z-direction in the case with LVC. Previously, we performed LS simulations without LVC on the same sample to investigate the impact of using one level with adaptive mesh refinement (AMR) around fluid interfaces [44]. AMR raised the P_{ow}-level in the low S_w-range, but S_{wi} remained about the same. Simulations with θ_{ow} = 0°, 20°, and 40°, gave $S_{wi} = 0.05$, 0.04, and 0.02, respectively [44]. In comparison, core-scale measurements of S_{wi} on homogeneous sandstone typically vary between 0.02 and 0.13 [47,48]. Specifically, Shikhov et al. [48] measured $S_{wi} = 0.11$ on Castlegate sandstone.

Apart from the difference in rock volumes, capillary pressure curves from pore-scale simulations and core-scale measurements typically deviate at low wetting-phase saturations because of a lacking representation of microporosity on the rock

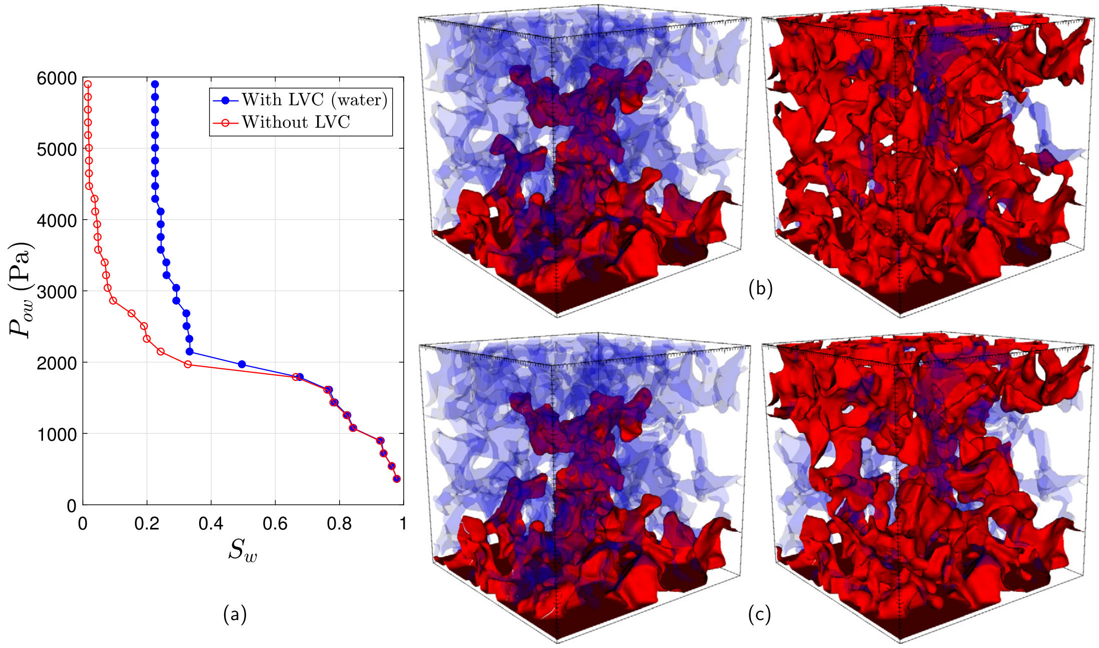


Fig. 20. Effect of using LVC on water during LS simulations of primary drainage on a subset of Castlegate sandstone with 128^3 grid cells. (a) $P_{ow}(S_w)$-curves for $\theta_{ow} = 40°$ and $\sigma_{ow} = 0.020$ N/m. (b) Stable configurations of oil (red) and water (blue) at $S_w = 0.67$ (left) and 0.05 (right) from simulation without LVC. (c) Stable configurations of oil (red) and water (blue) at $S_w = 0.67$ (left) and 0.25 (right) from simulation with LVC applied to water.

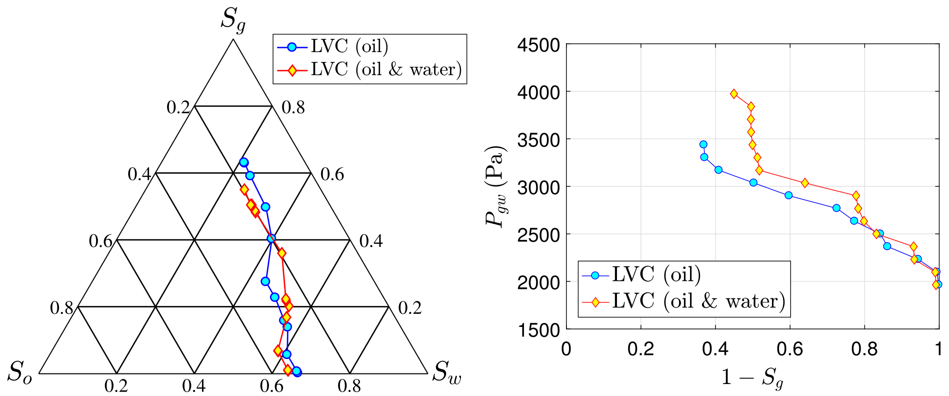


Fig. 21. Effect of using LVC on water in addition to oil during simulations of gas invasion from $S_w = 0.67$ after primary drainage in Castlegate sandstone. (a) Saturation paths for gas invasion. (b) Gas/water capillary pressure as a function of total liquid saturation.

image and because simulation models do not capture thin sub-grid-scale wetting-phase films along the pore walls [49–54]. In our simulations, the case without LVC assumes all water is continuous through thin films, whereas the case with LVC traps all water domains that appear isolated. We anticipate AMR simulations combined with higher image resolution can allow for drainage to a lower S_{wi} with LVC because the existence of more water filaments or layers in pore corners connects water domains and makes water more continuous. However, incorporating AMR within the current LVC algorithm requires further development. Another aspect is that calculations of S_{wi} might depend on whether phase domains contacting side walls shall preserve their volumes or not. We expect that these LVC boundary conditions become less sensitive to S_{wi} with larger computational domain due to a higher volume/area ratio.

Finally, we compare simulations with LVC applied to oil against simulations with LVC applied to both oil and water for gas invasions from $S_w = 0.67$ after primary drainage. The simulations end when the oil saturation no longer decreases. Fig. 21 displays minor differences between the saturation paths, whereas the capillary pressure between continuous gas and water is consistently higher for the case with LVC applied to both oil and water. The case with LVC of oil reaches a lower oil saturation, suggesting that volume conservation of water makes it harder to displace disconnected oil drops. Fig. 22 shows three-phase fluid configurations during the gas invasions. In both simulations, gas displaces continuous oil directly and disconnected oil by double displacement. These mechanisms also lead to oil fragmentation with trapping of smaller oil drops in narrower pore spaces.

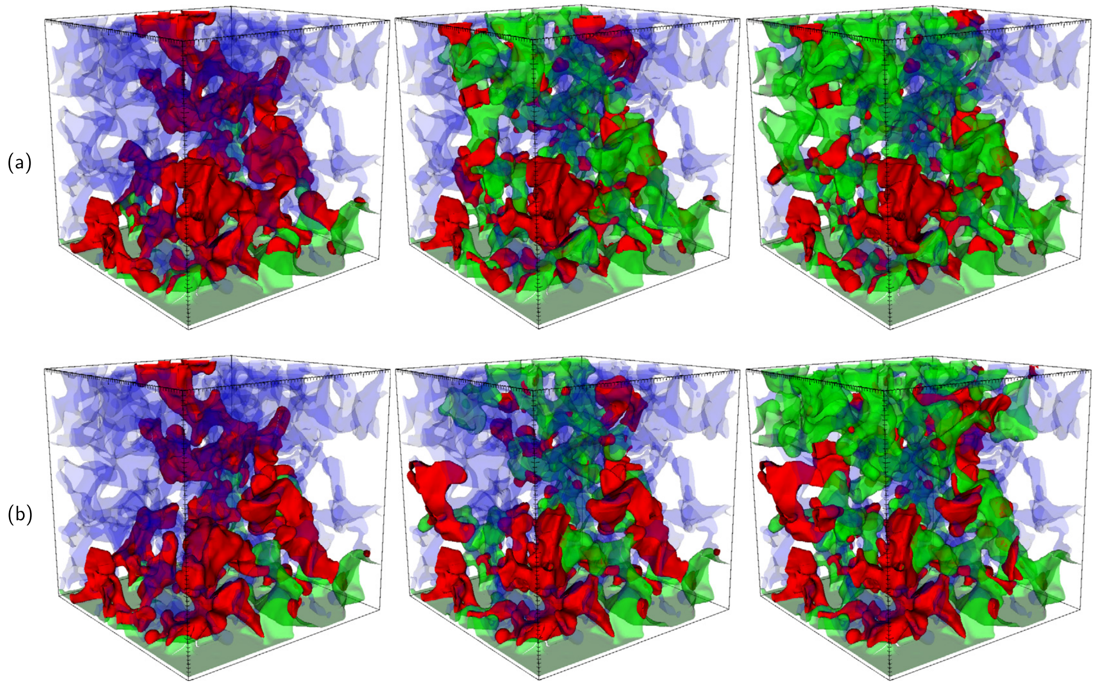


Fig. 22. Stable configurations of water (blue), oil (red), and gas (green) during gas invasion from $S_w = 0.67$ after primary drainage in Castlegate sandstone, as simulated with MLS-LVC method for (a) LVC applied to oil, and (b) LVC applied to both oil and water. (a) $S_g = 0.16$ (left), 0.50 (middle), and 0.63 (right). (b) $S_g = 0.17$ (left), 0.36 (middle), and 0.55 (right).

7. Summary and conclusions

This work presents a multiphase level set method with LVC for capillary-controlled pore-scale displacements in porous media. The method conserves volumes of disconnected phase ganglia and computes capillary pressures between ganglia and surrounding phases. A conservative volume redistribution technique handles ganglia splitting and coalescence. The algorithm is extended to handle domains that spanned multiple patches for simulation using parallel codes.

Although the method allows for an arbitrary number of phases, here we perform simulations of two- and three-phase displacements with LVC of one and two phases. Specifically, we provide simulation examples of important three-phase displacement mechanisms in porous media, like double displacements and displacement chains of multiple domains with splitting and coalescence.

Simulations on idealized 2-D pore geometries with increasing grid resolution demonstrate convergence of ganglia capillary pressures as well as consistently decreasing relative error of individual ganglia volumes. Relative error for the global disconnected phase volume decreases quadratically with decreasing grid-cell length. Simulations of gas invasion after primary drainage in 2-D water-wet porous media show that double displacement with oil fragmentation is a dominant mechanism. Disconnected oil and water ganglia develop large capillary pressure ranges that encase the simulated capillary pressure between corresponding continuous phases. Static oil drops surrounded by continuous water have constant capillary pressures, while the capillary pressures of oil drops contacting gas and water fluctuate during double displacement through wide and narrow pore regions. Finally, we demonstrate that the main effects of using LVC in level set simulations of primary drainage and gas invasion on 3-D porous rock is higher capillary pressure level for the continuous phases and higher endpoint saturations.

Future work will use the developed method to investigate residual phase trapping as well as pressure states and topology of isolated oil and gas ganglia during gas and water invasion cycles for recovery and storage processes in porous rock. We also anticipate that using the local conservative level set approach for interface tracking as an alternative to standard level set methods has potential to improve mass conservation significantly in multiphase flow simulations based on the Navier-Stokes equations.

CRediT authorship contribution statement

Espen Jettestuen: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Helmer André Friis:** Conceptualization, Methodology, Software, Writing - review & editing. **Johan Olav Helland:** Conceptualization, Funding acquisition, Methodology, Software, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Financial support was provided by the Research Council of Norway through Petromaks2 projects “Three-phase capillary pressure, hysteresis, and trapping in mixed-wet rock (grant no. 234131)” and “Foam dynamics in the presence of oil during multiphase flow in porous rock (grant no. 294886)”, and ConocoPhillips Norway through the research centre COREC. All computations were performed on resources provided by UNINETT Sigma2 – the National Infrastructure for High Performance Computing and Data Storage in Norway.

Appendix A

In the LVC algorithm we compute global labels to identify the various domains, see algorithm in Section 5. If we were to perform the grassfire algorithm in one go it would demand a lot of communication between processors, and this could be a potential bottleneck for the computational efficiency. Instead we perform the grassfire algorithm on each individual patch connected to a given processor. We make sure that each processor will a unique labelling of its distinct regions. We call these regions for subdomains as it is possible that they can be part of domains that span multiple processors.

The purpose of the algorithm we describe here is to identify which subdomains that are part of the same domain. To do this we need unique tags for the subdomains, and that each processor makes a “list of neighbours” that contains which subdomains are in contact through the boundaries between the processor and its neighbours. Each processor sends its “list of neighbours” to a master processor where the algorithm for identifying connected subdomains is preformed.

We use the ordered pair, $\mathbf{a} = (a_r, a_l)$, of the processor rank, a_r and the local label, a_l, as a subdomain's unique tag. If subdomains with tags $\mathbf{a}$ and $\mathbf{b}$ are connected then we will add the pair $[\mathbf{a}, \mathbf{b}]$ to a “list of neighbours”. As an example, the “list of neighbours” for processor with rank 3 in Fig. A.23 is $[(3,1),(2,1)]$, $[(3,2),(1,1)]$, $[(3,2),(5,1)]$, $[(3,2),(6,1)]$, where pair of connected subdomains are in square parenthesis. In the algorithm we will also need an ordering of the tags, which we define so that $\mathbf{a} < \mathbf{b}$ if $a_r < b_r$, or if $a_r = b_r$ and $a_l < b_l$ further, $\mathbf{a} = \mathbf{b}$ if $a_r = b_r$ and $a_l = b_l$.

Subdomains can be connected through many processors that span longer than nearest neighbour processors, and the algorithm on the master processor needs to take this into account. To accomplish this we make a list, L, on the master processor, of ordered pairs of subdomain tags. The 1st tag of the pair keeps its value while the 2nd tag can be changed by the algorithm, that is, for the pair $(\mathbf{a}, \mathbf{b}) \in L$, $\mathbf{a}$ will be unchanged while $\mathbf{b}$ is changed according to the steps in the algorithm. L contains one ordered pair for each subdomain, where the 1st tag is equal to the subdomain's tag. Initially the 1st and 2nd tag in a pair has the same value. The initial L for the domain decomposition in Fig. A.23 will look like this

$$L = \begin{bmatrix} ((0,1),(0,1)) \\ ((1,1),(1,1)) \\ ((2,1),(2,1)) \\ ((3,1),(3,1)) \\ ((3,2),(3,2)) \\ \vdots \end{bmatrix}.$$

Since all subdomain tags are represented as the 1st element in the pairs in L, we can use it to define a function, f_L, so given that $(\mathbf{a}, \mathbf{b}) \in L$ then $f_L(\mathbf{a}) = \mathbf{b}$, e.g. $f_L((2,1)) = (2,1)$ for the initial L in the example above. Note that f_L is redefined each time L is changed.

Our goal is now to identify the set of all subdomains that either directly or indirectly share a boundary. To do this we will define a function g_L so that, when all “list of neighbours” are added to L, all subdomains part of the same domain have the same value of g_L or more precisely $g_L^{-1}(g_L(\mathbf{a}))$, where g_L^{-1} is the inverse of g_L, will be the set of all the tags of subdomains that is part of the same domain as $\mathbf{a}$. To accomplish this we use f_L to define a recurrence relation $\mathbf{a}^{(n+1)} = f_L(\mathbf{a}^{(n)})$, where $\mathbf{a} = \mathbf{a}^{(0)}$, and we will update L so that the sequence f_L generates is decreasing, that is $\mathbf{a}^{(n+1)} \leq \mathbf{a}^{(n)}$. We remark that $n \geq 0$ and limited by the total number of subdomains. For instance, if the sequence only contains one element we have simply identified an isolated subdomain. We note that this is fulfilled by the initial L, since $\mathbf{a} = f_L(\mathbf{a})$ for all $\mathbf{a} \in L$. The sequence $a^{(n)}$ is bounded and will attend its minimum value which is a fixed point, a^*, of f_L. We define the function g_L so that $g_L(\mathbf{a}) = a^*$.

For each entry, $[\mathbf{a}_1, \mathbf{a}_2]$, in the “lists of neighbours”, sent to the master processor, we have the values $\mathbf{a}_1^* = g_L(\mathbf{a}_1)$ and $\mathbf{a}_2^* = g_L(\mathbf{a}_2)$. If $\mathbf{a}_1^* < \mathbf{a}_2^*$, we change the tuple $(\mathbf{a}_2^*, \mathbf{a}_2^*)$ in L to $(\mathbf{a}_2^*, \mathbf{a}_1^*)$, so that $f_L(\mathbf{a}_2^*) = \mathbf{a}_1^*$, if $\mathbf{a}_1^* > \mathbf{a}_2^*$, we change L so that $f_L(\mathbf{a}_1^*) = \mathbf{a}_2^*$, or if they are equal L is unchanged. We note that this redefinition of L is in accordance with the assumption of a decreasing sequence generated by the recurrence relation. When this is done for all “lists of neighbours”, the function g_L partition the subdomains into domains, where subdomains with the same value of g_L is part of the same domain.

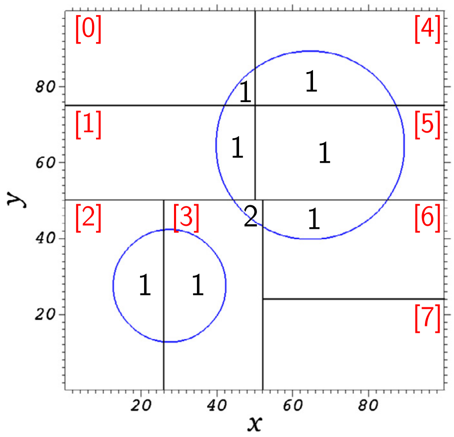


Fig. A.23. The plot from Fig. 3 (top row, middle column) with the local tag numbers (black numbers) of the two domains and here extended with processor numbers (red numbers in square brackets).

We can justify this by noting that when a pair of subdomains from a "list of neighbours" with tags $\mathbf{a}_1$ and $\mathbf{a}_2$ are added to L, L is changed so that $g_L(\mathbf{a}_1) = g_L(\mathbf{a}_2) = \mathbf{a}^*$, and we have the sequences, $\mathbf{a}_1 > \mathbf{a}_1^{(1)} > \ldots, \mathbf{a}_1^{(n_1)} = \mathbf{a}^*$ and $\mathbf{a}_2 > \mathbf{a}_2^{(1)} > \ldots, \mathbf{a}_2^{(n_2)} = \mathbf{a}^*$. When a new pair of subdomains, from a "list of neighbours", is added, we change L to L'. If the pair contains tags equal to elements in any of the sequences $a_1^{(n)}$ and $a_2^{(n)}$ defined above, we know that it is only the value of $f_{L'}(\mathbf{a}^*)$ that can change, per definition, so we still have that $g_{L'}(\mathbf{a}_1) = f_{L'}(g_L(\mathbf{a}_1)) = f_{L'}(g_L(\mathbf{a}_2)) = g_{L'}(\mathbf{a}_2)$, and by iterating this argument we know that they will have the same value when all pair of tags in all "lists of neighbours" are added. So as long as two subdomains are related through a sequence of subdomains they will have the same value of g_L and thus be part of the same domain.

Below we give a short example of the algorithm based on the subdomains on processors 0 to 3 in Fig. A.23. The pairs enclosed in square brackets represent L, and the arrows indicate addition of a pair of subdomain tags in a "list of neighbours" and bold fonts show changes in the list.

$$\begin{bmatrix} ((0,1),(0,1)) \\ ((1,1),(1,1)) \\ ((2,1),(2,1)) \\ ((3,1),(3,1)) \\ ((3,2),(3,2)) \\ \vdots \end{bmatrix} \xrightarrow[\begin{array}{c}(1,1)\\(3,2)\end{array}]{} \begin{bmatrix} ((0,1),(0,1)) \\ ((1,1),(1,1)) \\ ((2,1),(2,1)) \\ ((3,1),(3,1)) \\ ((3,2),(\mathbf{1},\mathbf{1})) \\ \vdots \end{bmatrix} \xrightarrow[\begin{array}{c}(1,1)\\(0,1)\end{array}]{} \begin{bmatrix} ((0,1),(0,1)) \\ ((1,1),(\mathbf{0},\mathbf{1})) \\ ((2,1),(2,1)) \\ ((3,1),(3,1)) \\ ((3,2),(1,1)) \\ \vdots \end{bmatrix} \xrightarrow[\begin{array}{c}(2,1)\\(3,1)\end{array}]{}$$

$$\begin{bmatrix} ((0,1),(0,1)) \\ ((1,1),(0,1)) \\ ((2,1),(2,1)) \\ ((3,1),(\mathbf{2},\mathbf{1})) \\ ((3,2),(1,1)) \\ \vdots \end{bmatrix} \xrightarrow[\begin{array}{c}(3,2)\\(1,1)\end{array}]{} \begin{bmatrix} ((0,1),(0,1)) \\ ((1,1),(0,1)) \\ ((2,1),(2,1)) \\ ((3,1),(2,1)) \\ ((3,2),(\mathbf{0},\mathbf{1})) \\ \vdots \end{bmatrix}.$$

We have not exhausted all possible pairs in "the lists of neighbours", but we note that the domain structure is recognized. For the first four processors we have identified a domain that spans three of processors.

References

[1] R. Hilfer, P.E. Øren, Dimensional analysis of pore scale and field scale immiscible displacement, Transp. Porous Media 22 (1996) 53–72.

[2] I. Chatzis, N.R. Morrow, Correlation of capillary number relationships for sandstone, SPE J. 24 (1984) 555–562, https://doi.org/10.2118/10114-PA.

[3] H. Ni, M. Boon, C. Garing, S.M. Benson, Predicting CO$_2$ residual trapping ability based on experimental petrophysical properties for different sandstone types, Int. J. Greenh. Gas Control 86 (2019) 158–176.

[4] P.E. Øren, W.V. Pinczewski, Fluid distribution and pore-scale displacement mechanisms in drainage dominated three-phase flow, Transp. Porous Media 20 (1995) 105–133, https://doi.org/10.1007/BF00616927.

[5] A.A. Keller, M.J. Blunt, P.V. Roberts, Micromodel observation of the role of oil layers in three-phase flow, Transp. Porous Media 26 (1997) 277–297.

[6] B. Amaechi, S. Iglauer, C.H. Pentland, B. Bijeljic, M.J. Blunt, An experimental study of three-phase trapping in sand packs, Transp. Porous Media 103 (2014) 421–436, https://doi.org/10.1007/s11242-014-0309-4.

[7] S. Iglauer, A. Paluszny, M.J. Blunt, Simultaneous oil recovery and residual gas storage: a pore-level analysis using in-situ X-ray micro-tomography, Fuel 103 (2013) 905–914, https://doi.org/10.1016/j.fuel.2012.06.094.

[8] Z. Qin, M. Arshadi, M. Piri, Micro-scale experimental investigations of multiphase flow in oil-wet carbonates. II. Tertiary gas injection and WAG, Fuel 257 (2019), https://doi.org/10.1016/j.fuel.2019.116012.

[9] M. Sohrabi, D.H. Tehrani, A. Danesh, G.D. Henderson, Visualization of oil recovery by water-alternating-gas injection using high-pressure micromodels, SPE J. 9 (3) (2004) 290–301, https://doi.org/10.2118/89000-PA.

[10] M.I.J. van Dijke, K.S. Sorbie, M. Sohrabi, A. Danesh, Three-phase flow WAG processes in mixed-wet porous media: pore-scale network simulations and comparison with water-wet micromodel experiments, SPE J. 9 (2004) 57–66, https://doi.org/10.2118/87822-PA.

[11] M.I.J. van Dijke, M. Lorentzen, M. Sohrabi, K.S. Sorbie, Pore-scale simulation of WAG floods in mixed-wet micromodels, SPE J. 15 (1) (2010) 238–247, https://doi.org/10.2118/113864-PA.

[12] A. Scanziani, K. Singh, H. Menke, B. Bijeljic, M.J. Blunt, Dynamics of enhanced gas trapping applied to CO_2 storage in the presence of oil using synchrotron X-ray micro tomography, Appl. Energy 259 (2020) 114136, https://doi.org/10.1016/j.apenergy.2019.114136.

[13] A. Herring, J. Middleton, R. Walsh, A. Kingston, A. Sheppard, Flow rate impacts on capillary pressure and interface curvature of connected and disconnected fluid phases during multiphase flow in sandstone, Adv. Water Resour. 107 (2017) 460–469.

[14] T. Li, S. Schlüter, M.I. Dragila, D. Wildenschild, An improved method for estimating capillary pressure from 3D microtomography images and its application to the study of disconnected nonwetting phase, Adv. Water Resour. 114 (2018) 249–260.

[15] A.H. Alizadeh, M. Khishvand, M.A. Ioannidis, M. Piri, Multi-scale experimental study of carbonated water injection: an effective process for mobilization and recovery of trapped oil, Fuel 132 (2014) 219–235, https://doi.org/10.1016/j.fuel.2014.04.080.

[16] M. Khishvand, A.H. Alizadeh, M. Piri, In-situ characterization of wettability and pore-scale displacements during two- and three-phase flow in natural porous media, Adv. Water Resour. 97 (2016) 279–298, https://doi.org/10.1016/j.advwatres.2016.10.009.

[17] X. Shan, H. Chen, Lattice Boltzmann model for simulating flows with multiple phases and components, Phys. Rev. E 47 (3) (1993) 1815–1819.

[18] Y. Sun, C. Beckermann, Sharp interface tracking using the phase-field equation, J. Comput. Phys. 220 (2007) 626–653.

[19] J. Kim, Phase-field models for multi-component fluid flows, Commun. Comput. Phys. 12 (3) (2012) 613–661.

[20] A. Zolfaghari, M. Piri, Pore-scale network modeling of three-phase flow based on thermodynamically consistent threshold capillary pressures. I. Cusp formation and collapse, Transp. Porous Media 116 (3) (2017) 1093–1137, https://doi.org/10.1007/s11242-016-0814-8.

[21] P. Mohammadmoradi, A. Kantzas, Toward direct pore-scale modeling of three-phase displacements, Adv. Water Resour. 110 (2017) 120–135.

[22] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.

[23] J.A. Sethian, Level Set Methods and Fast Marching Methods, 2nd edition, Cambridge University Press, 1999.

[24] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, 2003.

[25] M. Prodanovic, S.L. Bryant, A level set method for determining critical curvatures for drainage and imbibition, J. Colloid Interface Sci. 304 (2006) 442–458.

[26] C.W. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225.

[27] M. Sussman, E. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.

[28] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (1) (2002) 83–116.

[29] K. Luo, C. Shao, Y. Yang, J. Fan, A mass conserving level set method for detailed numerical simulation of liquid atomization, J. Comput. Phys. 298 (1) (2015) 495–519.

[30] R.I. Saye, J.A. Sethian, The Voronoi implicit interface method for computing multiphase physics, Proc. Natl. Acad. Sci. USA 108 (49) (2011) 19498–19503, https://doi.org/10.1073/pnas.1111557108.

[31] E. Jettestuen, J.O. Helland, M. Prodanović, A level set method for simulating capillary-controlled displacements at the pore scale with nonzero contact angles, Water Resour. Res. 49 (8) (2013) 4645–4661, https://doi.org/10.1002/wrcr.20334.

[32] J.O. Helland, J. Pedersen, H.A. Friis, E. Jettestuen, A multiphase level set approach to motion of disconnected fluid ganglia during capillary-dominated three-phase flow in porous media: numerical validation and applications, Chem. Eng. Sci. 203 (2019) 138–162.

[33] J.O. Helland, H.A. Friis, E. Jettestuen, S.M. Skjæveland, Footprints of spontaneous fluid redistribution on capillary pressure in porous rock, Geophys. Res. Lett. 44 (2017) 4933–4943, https://doi.org/10.1002/2017GL073442.

[34] J.O. Helland, E. Jettestuen, Mechanisms for trapping and mobilization of residual fluids during capillary-dominated three-phase flow in porous rock, Water Resour. Res. 52 (2016) 5376–5392, https://doi.org/10.1002/2016WR018912.

[35] F. Losasso, T. Shinar, A. Selle, R. Fedkiw, Multiple interacting liquids, Water Resour. Res. 25 (3) (2006) 812–819.

[36] H.K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1) (1996) 179–195.

[37] M. Prodanović, W.B. Lindquist, R.S. Seright, Porous structure and fluid partitioning in polyethylene cores from 3D x-ray microtomographic imaging, J. Colloid Interface Sci. 298 (2006) 282–297.

[38] A. Rosenfeld, J.L. Pfaltz, Sequential operations in digital picture processing, J. Assoc. Comput. Mach. 13 (4) (1966) 471–494.

[39] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.

[40] R. Anderson, W. Arrighi, N. Elliott, B. Gunney, R. Hornung, SAMRAI concepts and software design, Technical Report LLNL-SM-617092-DRAFT, Center for Applied Scientific Computing (CASC), Lawrence Livermore National Laboratory, Livermore, CA, 2013.

[41] B.T.N. Gunney, A.M. Wissink, D.A. Hysom, Parallel clustering algorithms for structured AMR, J. Parallel Distrib. Comput. 66 (11) (2006) 1419–1430.

[42] R. Hornung, S. Kohn, Managing application complexity in the SAMRAI object-oriented framework, Concurr. Comput., Pract. Exp. 14 (2002) 347–368.

[43] R. Hornung, A. Wissink, S. Kohn, Managing complex data and geometry in parallel structured AMR applications, Eng. Comput. 22 (2006) 181–195.

[44] H.A. Friis, J. Pedersen, E. Jettestuen, J.O. Helland, M. Prodanović, Pore-scale level set simulations of capillary-controlled displacement with adaptive mesh refinement, Transp. Porous Media 128 (1) (2019) 123–151.

[45] M.I.J. van Dijke, K.S. Sorbie, The relation between interfacial tensions and wettability in three-phase systems: consequences for pore occupancy and relative permeability, J. Pet. Sci. Eng. 33 (1–3) (2002) 39–48.

[46] A. Sheppard, Network generation comparison forum, Digital Rocks Portal, University of Texas, Austin, https://doi.org/10.17612/P7059V, 2015.

[47] I. Shikhov, C. Arns, Evaluation of capillary pressure methods via digital rock simulations, Transp. Porous Media 107 (2015) 623–640, https://doi.org/10.1007/s11242-015-0459-z.

[48] I. Shikhov, M. d'Eurydice, J. Arns, C. Arns, An experimental and numerical study of relative permeability estimates using spatially resolved t_1-z NMR, Transp. Porous Media 118 (2017) 225–250, https://doi.org/10.1007/s11242-017-0855-7.

[49] B. Ahrenholz, J. Tölke, P. Lehmann, A. Peters, A. Kaestner, M. Krafczyk, W. Durner, Prediction of capillary hysteresis in a porous material using lattice-Boltzmann methods and comparison to experimental data and a morphological pore network model, Adv. Water Resour. 31 (2008) 1151–1173.

[50] T. Ramstad, P. Øren, S. Bakke, Simulation of two-phase flow in reservoir rocks using a lattice Boltzmann method, SPE J. 15 (2010) 917–927, https://doi.org/10.2118/124617-PA.

[51] E.S. Boek, I. Zacharoudiou, F. Gray, S.M. Shah, J.P. Crawshaw, J. Yang, Multiphase-flow and reactive-transport validation studies at the pore scale by use of lattice Boltzmann computer simulations, SPE J. 22 (2017) 940–949, https://doi.org/10.2118/170941-PA.

[52] Q. Lin, B. Bijeljic, H. Rieke, M.J. Blunt, Visualization and quantification of capillary drainage in the pore space of laminated sandstone by a porous plate method using differential imaging X-ray microtomography, Water Resour. Res. 53 (2017) 7457–7468, https://doi.org/10.1002/2017WR021083.

[53] N. Nair, J.V. Koelman, An Ising-based simulator for capillary action in porous media, Transp. Porous Media 124 (2018) 413–437, https://doi.org/10.1007/s11242-018-1075-5.

[54] T. Bultreys, Q. Lin, Y. Gao, A.Q. Raeini, A. AlRatrout, B. Bijeljic, M.J. Blunt, Validation of model predictions of pore-scale fluid distributions during two-phase flow, Phys. Rev. E 97 (2018) 053104, https://doi.org/10.1103/PhysRevE.97.053104.