CrossMark

A Monte Carlo Algorithm for Immiscible Two-Phase Flow in Porous Media

Isha Savani[1] · **Santanu Sinha**[2] · **Alex Hansen**[2] ·
Dick Bedeaux[3] · **Signe Kjelstrup**[3] · **Morten Vassvik**[1]

Abstract We present a Markov Chain Monte Carlo algorithm based on the Metropolis algorithm for simulation of the flow of two immiscible fluids in a porous medium under macroscopic steady-state conditions using a dynamical pore network model that tracks the motion of the fluid interfaces. The Monte Carlo algorithm is based on the configuration probability, where a configuration is defined by the positions of all fluid interfaces. We show that the configuration probability is proportional to the inverse of the flow rate. Using a two-dimensional network, advancing the interfaces using time integration, the computational time scales as the linear system size to the fourth power, whereas the Monte Carlo computational time scales as the linear size to the second power. We discuss the strengths and the weaknesses of the algorithm.

✉ Alex Hansen
Alex.Hansen@ntnu.no

Isha Savani
Isha.Savani@gmail.com

Santanu Sinha
santanu@csrc.ac.cn

Dick Bedeaux
Dick.Bedeaux@chem.ntnu.no

Signe Kjelstrup
Signe.Kjelstrup@ntnu.no

Morten Vassvik
Morten.Vassvik@ntnu.no

[1] Department of Physics, Norwegian University of Science and Technology, NTNU, 7491 Trondheim, Norway

[2] Beijing Computational Science Research Center, 10 West Dongbeiwang Road, Haidan, Beijing 100193, China

[3] Department of Chemistry, Norwegian University of Science and Technology, NTNU, 7491 Trondheim, Norway

1 Introduction

The characterization of porous media at the pore level is undergoing a revolution (Blunt
et al. 2013). Through the use of new scanning techniques, we are capable of reconstructing
the pore space completely, including the tracking of motion of immiscible fluids. A gap is
now appearing between the geometrical characterization of porous media and our ability to
predict their flow properties based on this knowledge.

The pore scale may be of the order of microns, whereas the largest scales—e.g., the
reservoir scale—may be measured in kilometers. Hence, there are some eight orders of
magnitude between the smallest and the largest scales. At some intermediate scale, that
of the representative elementary volume (REV), the porous medium may be regarded as a
continuum and the equations governing the flow properties are differential equations. The
crucial problem is to construct these effective differential equations from the physics at the
pore scale. This is the upscaling problem. A possible path toward this goal is to use brute
computational power to link the pore-scale physics to pore networks large enough so that a
continuum description makes sense. Alas, this is still beyond what can be done numerically.
However, computational hardware and algorithms are steadily being improved, and we are
moving toward this goal.

It is the aim of this paper to introduce a new algorithm that improves significantly the
efficiency of network models (Joekar-Niasar and Hassanizadeh 2012). These are models that
are based on the skeletonization of the spaces in such a way that a network of links and nodes
emerge. Each link and node is associated with parameters that reflect the geometry of the
pore space they represent. The fluids are then advanced by time stepping some simplified
version of equations of motion of fluid. The bottleneck in this approach is the necessity to
solve the Kirchhoff equations to determine the pressure field whose gradient drive the fluids
in competitions with the capillary forces.

A different and at present popular computational approach, among several, is the lattice
Boltzmann method (Ramstad et al. 2010, 2012). This method, based on simultaneously solv-
ing the Boltzmann equations for different species of lattice gases, is very efficient compared
to the network approach necessitating solving the Kirchhoff equations. However, the draw-
back of the lattice Boltzmann approach is that one needs to resolve the pore space. Hence,
one needs to use a grid with a finer mask than the network used in the network approach.
This makes the lattice Boltzmann approach very efficient at the scale where the actual shape
of the pores matters, but not at the larger scale where the large-scale topology of the pore
network is more important. Further methods that resolve the flow at the pore level are, e.g.,
smoothed particle hydrodynamics (Tartakovsky and Meakin 2005; Ovaysi and Piri 2010; Liu
and Liu 2010) and density functional hydrodynamics (Armstrong et al. 2016). When network
models are so heavy numerically that the networks that can be studied are not much larger
than those studied with the pore-scale methods, the latter win as they can give a more detailed
description of the flow. However, if the computational limitations inherent to network models
could be overcome, they would form an important tool in resolving the scale-up problem:
At small scales network, models would be calibrated against the methods that are capable of
resolving the flow at the pore level. On large scales, their results may be extrapolated to scales
large enough for homogenization, i.e., replacing the original pore network by a continuum.

As pointed out above, the bottleneck in the network models is the necessity to determine the pressure field at each time step. When the time steps are determined by the motion of the fluid interfaces, these will be small as they typically are set by the time lapse before the next interface reaches a node in the network. Time stepping allows detailed questions concerning how flow patterns develop in time to be answered. That is, the time stepping provides a detailed sequence of configurations where each member of the sequence is the child of the one before and the parent of the one after. If the quantities that are calculated are averages over configurations, time stepping will provide too much information; for averages, the *order* in which the configurations occur is of no consequence. If the order in which the fluid configurations occur is scrambled, the averages remain unchanged. This is where the Monte Carlo method enters. It provides a way to produce configurations that will result in the same averages as those obtained through time stepping. The order in which the configurations occur will be different from those obtained by time stepping. The time-stepping procedure necessitates that there are tiny differences between each configuration in the sequence, since the time steps have to be small. This limitation is overcome in the Monte Carlo method, which we will describe here. This makes the Monte Carlo method much more efficient than time stepping as we will see.

In Sect. 2, we describe the network model we use to compare the Monte Carlo method with time stepping, see Aker et al. (1998), Knudsen et al. (2002). In the next Sect. 3, we start by explaining the statistical mechanics approach to immiscible two-phase flow in porous media that lies behind the Monte Carlo algorithm we propose (Hansen et al. 2016; Savani et al. 2016). In particular, we derive the configuration probability—the probability that a given distribution of fluid interfaces in the model will appear. This is also known as the *ensemble distribution* in the statistical physics community. Based on this knowledge, we then go on to describe the Monte Carlo algorithm itself. This section is followed by Sect. 4 where we compare the Monte Carlo method with time stepping using the same network model described in Sect. 2. We then go on to compare the efficiency in terms of computational cost of the two methods. We end this section by discussing the limitations of the Monte Carlo algorithm as it now stands and point toward how these may be overcome. We end by Sect. 5 where we summarize the work and draw our conclusions.

2 Network Model

In order to have a concrete system to work with, we describe here the details of the network model we use. The model is essentially the one first developed in references Aker et al. (1998), Knudsen et al. (2002). For simplicity, we do not consider a reconstructed pore network based on a real porous medium (Tørå et al. 2011; Ramstad et al. 2009). Rather, we simply use a two-dimensional square network, with disorder in the pore radii, oriented at 45° with respect to the average flow direction as shown in Fig. 1. As described in Knudsen et al. (2002), we use *bi-periodic boundary conditions*. Hence, the network takes a form of the surface of a torus. In this way, the two-phase flow enters a steady state after an initial transient period. This steady state does *not* mean that the fluid interfaces are static. Rather, we use capillary numbers high enough so that fluid clusters incessantly form and break up. By *steady state*, we mean that the macroscopic averages—averages over the entire network—are well defined and do not drift.

Fig. 1 Geometry of the pore network we use. The *shaded area* constitutes a link between two nodes
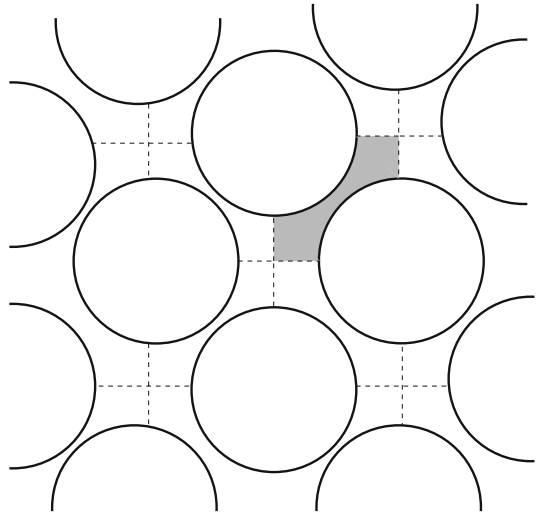

Fig. 2 This is one of the links in the network. The wetting and non-wetting fluids, colored by *white* and *gray*, respectively, are separated by interfaces. Each interface provides a capillary pressure $p_c(x)$ that points in the direction from the non-wetting toward the wetting fluid. Through the link, a flow q passes. We indicate the two node pressures p_1 and p_2 at the end of the link
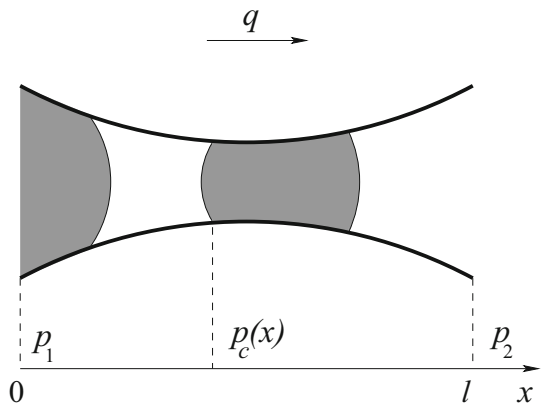
The network contains $L \times L$ links. All links have equal length l, but their radii have been drawn from a uniform distribution of random numbers in the interval $[0.1l, 0.4l]$. We set $l = 1$ mm. We neglect gravitational effects.

Fluid flow through each link in the network is modeled using the Washburn equation (Washburn 1921), see Fig. 2. There is a volume flow q passing through it driven by the two pressures p_1 and p_2. Each fluid interface contributes a capillary pressure $p_c(x)$ where $x \in [0, l]$ is the position of the interface. The capillary pressure is given by the Young–Laplace equation

$$|p_c(x)| = \frac{2\gamma \cos \theta}{r_0} \left[1 - \cos \left(2\pi \frac{x}{l} \right) \right], \tag{1}$$

where γ is the surface tension, θ is the contact angle between the interface and the pore wall. We set $\gamma \cos \theta = 30$ dyn/cm. r_0 is the average link radius. We assume that the link has a shape so that p_c attains the given x dependence. It has been chosen so that $p_c(0) = p_c(l) = 0$ and $\max_x |p_c(x)| = |p_c(l/2)|$. The Washburn equation then becomes

$$q = -\frac{\pi r_0^4}{8\mu_{\text{av}}} \left[p_2 - p_1 - \sum_i p_{\text{c}}(x_i) \right], \tag{2}$$

where $\mu_{\text{av}} = s_{nw}\mu_{nw} + s_w\mu_w$ is the effective viscosity. $s_{nw} = l_{nw}/l$ and $s_w = l_w/l$ are the fractions of the link length that cover the non-wetting and wetting fluids, respectively, so that $s_{nw} + s_w = 1$. We set $\mu_{nw} = \mu_w = 1$ poise.

We define the capillary number Ca as

$$\text{Ca} = \frac{\langle |q| \rangle \langle \mu_{\text{av}} \rangle}{\gamma \pi \langle r_0 \rangle^2}, \tag{3}$$

where $\langle \cdots \rangle$ is an average over all links.

A pressure difference ΔP is applied across the network. This is done in spite of the network being periodic in the direction of the pressure difference, see Knudsen et al. (2002). By demanding balance of flow at each node using the Washburn equation (2), we determine the pressures (p_i) at the nodes. This is done by solving the corresponding matrix inversion problem by using the conjugate gradient algorithm (Batrouni and Hansen 1988).

When the pressures at nodes are known, the flow q_{ij}—here between neighboring nodes i and j connected by a link—is calculated using Eq. (2). Knowing the velocity of the interfaces in each link, we then determine the time step such that any meniscus can move a maximum distance, say, one-tenth of the length of corresponding link in that time. All the interfaces are then moved accordingly, and the pressure at the nodes are determined again by the conjugate gradient algorithm. This is equivalent to event-driven molecular dynamics. When an interface reaches the node, the interface will spread into the links that are connected to the node and which have fluid entering them from the node. The rules for how this is done are described in detail in Knudsen et al. (2002).

3 Metropolis Monte Carlo

We first describe the theory that lies behind the Monte Carlo algorithm that we present. We need to introduce the concepts of *configuration*, and *configuration probability,* also known as the ensemble distribution in the statistical mechanics community. We then go on to derive the configurational probability. Armed with this, we construct the Monte Carlo algorithm (Press et al. 2007) after having presented a short review of the Metropolis version of Monte Carlo (Press et al. 2007; Metropolis et al. 1953).

3.1 Statistical Mechanics of Immiscible Two-Phase Flow

Sinha et al. (2013) studied the motion of bubbles in a single capillary tube with varying radius. Suppose that the capillary tube has a length L and a radius that varies as $r = r_0/[1 - a\cos 2\pi x/l]$ where $l \ll L$, a is an amplitude and r_0 is the average radius of the tube. Suppose furthermore that the tube is filled with wetting fluid except for a bubble of length Δx_b and a center position x_b. By using Eq. (1), one derives the net capillary force from the two interfaces that limit the bubble as,

$$p_b(x_b) = -\sigma \, \sin\left(\frac{2\pi x_b}{l}\right), \tag{4}$$

where $\sigma = 4a\gamma \cos\theta \sin(\pi \Delta x_b/l)/r_0$. By combining this equation with the Washburn equation (2), one finds

$$\dot{x}_b = -\frac{r_0}{8l\mu_{av}} \left[\Delta p + \sigma \sin\left(\frac{2\pi x_b}{l}\right) \right], \tag{5}$$

where $\pi r_0^2 \dot{x}_b = q$, and $\Delta p = (l/L)\Delta P$ where ΔP is the pressure difference across the tube.

Suppose there is a quantity $f = f(x_b)$ that depends on the position of the bubble in the capillary tube. For example, f might be the flow q. Let us now assume that ΔP does not vary in time. The time average of f is then

$$\overline{f} = \frac{1}{T_b} \int_0^{T_b} f(x_b(t)) \, dt, \tag{6}$$

where $x_b(t)$ is the time integration of the Washburn equation (2) and the time period $T_b = (2\pi\sigma)/\sqrt{\Delta p - \sigma^2}$. We note, *and this is the crucial observation,* that we may change integration variable from time t to bubble position x_b,

$$\overline{f} = \frac{1}{T_b} \int_0^l f(x_b) \frac{dx_b}{dx_b/dt} = \int_0^l f(x_b)\Pi(x_b)dx_b, \tag{7}$$

where

$$\Pi(x_b) = \frac{1}{T_b(dx_b/dt)} = \frac{\pi r_0^2}{T_b} \frac{1}{q} \tag{8}$$

is the *configuration probability.* That is, the *configuration* of the tube is given by the position of x_b of the bubble. Equation (8) gives the probability density to find the bubble at position x_b—and hence in that configuration.

The Washburn equation (5) gives the motion of the bubble that is used in Eqs. (7) and (8). The Washburn equation assumes that we control the pressure drop ΔP. If we on the other hand control the flow q, the equation of motion becomes

$$\dot{x}_b = \frac{q}{\pi r_0^2}. \tag{9}$$

The time period now becomes

$$T_b = \frac{\pi r_0^2 L}{q}, \tag{10}$$

and hence the configurational probability is

$$\Pi(x_b) = \frac{\pi r_0^2}{T_b} \frac{1}{q} = \frac{1}{L}, \tag{11}$$

which states that all positions of the bubble are equally probable.

To ramp up the complexity of the problem, we assume that there are N bubbles in the one-dimensional tube. The centers of mass of bubble number $j \in [1, N]$ are x_j, and it has a width of Δx_j. Since the system is one dimensional, all bubbles move with the same speed $\dot{x}_j = \dot{x}_1$. The Washburn equation is then

$$\dot{x}_j = \dot{x}_1 = -\frac{r_0}{8L\mu_{av}} \left[\Delta p + \sum_{j=1}^N \gamma_j \sin\left(\frac{2\pi}{l}(x_1 + \delta x_j)\right) \right], \tag{12}$$

where $\delta x_j = x_j - x_1$ and

$$\gamma_j = \frac{4\sigma a}{r_0} \sin\left(\frac{\pi \Delta x_j}{l}\right). \tag{13}$$

Solving the equations of motion (12) gives $x_j = x_j(t)$. We may invert $x_1 = x_1(t)$ to get $t = t(x_1)$. Hence, we then have $x_j(x_1) = x_j(t(x_1))$ for all j. Suppose now we have a function $f = f(x_1, \ldots, x_N)$, analogous to the one introduced in Eq. (7). Its time average is

$$
\begin{aligned}
\overline{f} &= \frac{1}{T_b} \int_0^{T_b} f(x_1(t), \ldots, x_N(t)) \ dt \\
&= \frac{1}{T_b} \int_0^L f(x_1, \ldots, x_N(x_1)) \ \frac{dx_1}{dx_1/dt} \\
&= \int_0^L f(x_1, \ldots, x_N(x_1)) \ \Pi(x_1) dx_1,
\end{aligned}
\tag{14}
$$

where

$$\Pi(x_1) = \frac{1}{T_b} \frac{1}{(dx_1/dt)} = \frac{\pi r_0^2}{T_b} \frac{1}{q}, \tag{15}$$

where $q = \pi r_0^2 \dot{x}_1$. This is precisely the same expression as in (8).

We now turn to complex network topologies. For concreteness, we may imagine a two-dimensional square network. However, the arguments presented in the following are general. A configuration is given by the position of all interfaces. Let us denote that $\mathbf{x} = (x_1, x_1, x_2, \ldots, x_N)$, where x_i is the position of the ith interface. Hence, x_i contains information both on which link the interface sits in and where it sits in the link. A flow Q passes through the network. The flow equations for the network consist of a Washburn constitutive equation for each link combined with the Kirchhoff equations distributing the flow between the links. The motion of the interfaces is highly nonlinear, but of the form $\dot{x}_i = g_i(\mathbf{x})$. Solving these equations gives $x_j = x_j(t)$.

Again we consider a function $f = f(\mathbf{x})$ of the position of the interfaces. Its time average is

$$
\begin{aligned}
\overline{f} &= \frac{1}{T_b} \int_0^{T_b} f(\mathbf{x}(t)) \ dt = \frac{1}{T_b} \int_0^L f(\mathbf{x}(x_i)) \ \frac{dx_i}{dx_i/dt} \\
&= \int_0^L f(\mathbf{x}(x_i)) \ \Pi(x_i) dx_i.
\end{aligned}
\tag{16}
$$

Here, we have inverted $x_i = x_i(t)$ so that we have $t = t(x_i)$ and then substituted $\mathbf{x}(t) = \mathbf{x}(t(x_i)) = \mathbf{x}(x_i)$. The configurational probability is defined as before,

$$\Pi(\mathbf{x}) = \frac{1}{T_b} \frac{1}{dx_i/dt}. \tag{17}$$

Let us now choose $x_i = x_1$ to be an interface moving in a link that carries all the flow in the network. Such a link is a capillary tube connected in series with the rest of the network. In this case, we have $\dot{x}_1 = Q/\pi r_0^2$, where Q is the total flow. Hence, we have

$$\Pi(x_1) = \frac{\pi r_0^2}{T_b} \frac{1}{Q}. \tag{18}$$

We have in the discussion so far compared the time evolution of a given sample defined by an initial configuration of interfaces. We now imagine *an ensemble* of initial configurations

of interfaces. Each sample evolves in time, and there will be a configurational probability (18) for each. This will have the same value for each configuration **x** that corresponds to the same flow Q. Hence, we have the configurational probability

$$\Pi(\mathbf{x}) \propto \frac{1}{Q}. \tag{19}$$

This equation is the major theoretical result of this paper: all configurations corresponding to the same Q are equally probable. Intuitively, Eq. (19) makes sense: The slower the flow, proportionally the more the system stays in—or close to—a given configuration (Savani et al. 2016).

Is the system ergodic? Equations (7), (14) and (16) answer this question positively. Time averages give, by construction, the same results as configurational averages.

3.2 Implementation of the Metropolis Algorithm

In order to present the details of the Metropolis Monte Carlo algorithm that we propose, we first review the general formulation of the Metropolis algorithm (Krauth 2006; Landau and Binder 2015).

3.2.1 General Considerations

We have a set of configurations characterized by the variable **x**, the positions of the interfaces. We now wish to construct a *biased random walk* through these configurations so that the number of times each configuration is visited—i.e., the random walk comes within d**x** of the configuration—is proportional to $\Pi(\mathbf{x})$, proportional to the probability for that configuration. The Metropolis algorithm accomplishes this goal. In order to do so, a transitional probability density from state **x** to state $\mathbf{x}'$ is constructed as

$$\Pi(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x}, \mathbf{x}') \ \min\left(1, \frac{\Pi(\mathbf{x}')}{\Pi(\mathbf{x})}\right). \tag{20}$$

where $\pi(\mathbf{x}, \mathbf{x}')$ is the probability density to pick trial configuration $\mathbf{x}'$, given that the system is in configuration **x**. It is crucial that $\pi(\mathbf{x}', \mathbf{x})$ is symmetric,

$$\pi(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x}', \mathbf{x}). \tag{21}$$

Equations (20) and (21) ensure detailed balance,

$$\Pi(\mathbf{x})\Pi(\mathbf{x}, \mathbf{x}') = \Pi(\mathbf{x}')\Pi(\mathbf{x}', \mathbf{x}). \tag{22}$$

Detailed balance guarantees that the biased random walk visits the configurations **x** with a frequency proportional to $\Pi(\mathbf{x})$. The generated configurations follow the ensemble distribution.

When we combine Eqs. (19) and (20), we have

$$\Pi(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x}, \mathbf{x}') \ \min\left(1, \frac{Q(\mathbf{x})}{Q(\mathbf{x}')}\right). \tag{23}$$

3.2.2 The Implementation

The Metropolis Monte Carlo algorithm based on Eq. (23) consists of two crucial steps. The first step consists in generating a *trial configuration*, and the second step consists in deciding whether to keep the old configuration or replacing it with the trial configuration.

Fig. 3 We show here a typical network of the kind we use for comparing the time stepping and Monte Carlo methods. The network is bi-periodic, and the flow is from the *bottom toward the top*. The *dark red* constitutes the non-wetting fluid and the *gray* constitutes the wetting fluid. When using the Monte Carlo method, a random sub-network is chosen as shown in the *box*, taken out of the network, integrated forward in time after having been made bi-periodic, and then re-entered into the network. This is the heart of the Metropolis Monte Carlo algorithm
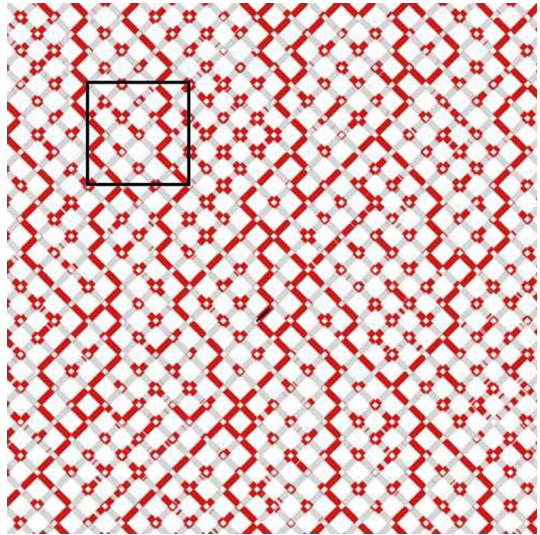
The first step, generating the trial configuration, is governed by the trial configuration probability $\pi(\mathbf{x}, \mathbf{x}')$, which must obey the symmetry (21). That is, if the system is in configuration $\mathbf{x}$, the probability to pick a trial configuration $\mathbf{x}'$ must be equal to the probability to pick as trial configuration $\mathbf{x}$ if the system is in configuration $\mathbf{x}'$.

Suppose the system is in configuration $\mathbf{x}$. One needs to define a *neighborhood* of configurations among which the trial configuration is chosen. If the neighborhood is too restricted, the Monte Carlo random walk will take steps that are too small and hence would be inefficient. If, on the other hand, the neighborhood is too large, the random walk ends up doing huge steps that will miss the details.

We propose generating the trial configurations as follows. Our system is shown in Fig. 3 and consists of $L \times L$ links as described in Sect. 2. There is a flow q_{ij} through link ij connecting the neighboring nodes i and j. There is a total flow rate Q in the network given by

$$Q = \frac{1}{L} \sum_{\text{all } ij} q_{ij}, \tag{24}$$

and a corresponding pressure drop ΔP.

We choose a randomly positioned sub-network as shown in Fig. 3. The network consists of $\Lambda \times \Lambda$ links. We "lift" the sub-network out of the complete network and fold it into a torus, i.e., implementing bi-periodic boundary conditions. The configurations of fluid interfaces in the sub-network remain unchanged at this point.

We calculate the flow rate in the sub-network

$$\Theta = \frac{1}{\Lambda} \sum_{ij \text{ in sub network}} q_{ij}. \tag{25}$$

By solving the Kirchhoff equations on the sub-network, we *time step* the configuration forwards in time while keeping the flow rate Θ constant. We end the time integration when 4—arbitrarily chosen—sub-network pore volumes have passed through it.
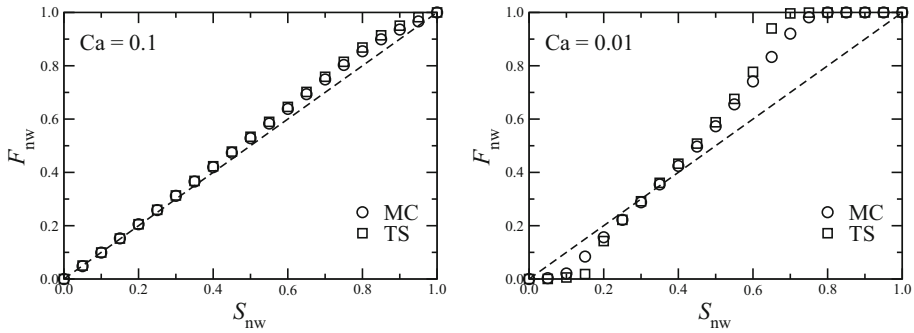
Fig. 4 Non-wetting fractional flow (F_{nw}) as a function of non-wetting saturation (S_{nw}) in the steady state obtained via Monte Carlo simulations (MC) with constant flow rate (Q) at capillary numbers Ca = 0.1 and 0.01. Results are compared with that obtained via time-stepping simulations (TS). The *diagonal dashed lines* in the plots imply $F_{nw} = S_{nw}$, a system of miscible fluids would follow that line. The data are averaged over 10 samples

The bi-periodic boundaries of the sub-network are then opened up, and the sub-network with the new configuration of fluid interfaces is placed back into the full network. This is then the trial configuration $\mathbf{x}'$.

Part of the probabilistic choice of the trial configuration that defines $\pi(\mathbf{x}, \mathbf{x}')$ rests on the choice of the sub-network: Its position is picked at random. Hence, if the system is in state $\mathbf{x}$ or in trial state $\mathbf{x}'$, the probability to pick a particular sub-network is the same. This makes this part of the choice of trial configuration symmetric. When the sub-network is time stepped for 4 sub-system pore volumes, this is done at constant flow rate Θ. Hence, all sub-network configurations are equally probable, see Eq. (19). Hence, also this part of the choice of trial configuration is symmetric. The full probability $\pi(\mathbf{x}, \mathbf{x}')$ is the probability of picking a given sub-network times the probability that a given configuration will occur. Combining the two leads to the necessary symmetry (21).

We point out here that whereas the configurational probability $\Pi(\mathbf{x})$ in (19) is valid for all configurations, through the way we generate our samples, we are restricting ourselves to physically realistic samples in that they are generated through time-stepping parts of the system. We cannot at this stage prove that this does not bias our sampling.

Once the trial configuration $\mathbf{x}$ has been generated, it is necessary to calculate the total flow rate $Q = Q(\mathbf{x}')$ in the network. We then decide to accept the trial configuration $\mathbf{x}'$ by using (23). This defines a Monte Carlo *update*.

We repeat this procedure until each link in the network has been part of at least one sub-network. This defines a Monte Carlo *sweep*.

4 Results

We now present numerical results of the Monte Carlo simulation considering the model described in Sect. 2, and we will compare them with the results by time-stepping simulations. Simulations are performed for two different ensembles, one is when the total flow rate Q is kept constant (CQ ensemble) and the other when the total pressure drop ΔP is kept constant (CP ensemble). A network of 40×40 links ($L = 40$) is considered for both Monte Carlo and time-stepping procedure. The sub-network size is 20×20 links ($\Lambda = 20$). To identify whether

the system has reached the steady state, we measured the quantities as a function of time steps in time stepping and as a function of sweeps in the case of Monte Carlo. We then identified the steady states when the averages of measured quantities (e.g., F_{nw} and ΔP or Q) did not drift with time or with sweeps. We then take average over time (time stepping) or sweeps (Monte Carlo), which give us the time average and the ensemble average, respectively. We average 10 different networks, but with the same sequence of networks for both Monte Carlo and time stepping. First we present the results for CQ ensemble. Two capillary numbers, Ca = 0.1 and 0.01, are used, and for each Ca, simulations are performed for different values of non-wetting saturations in intervals of 0.05 from 0.05 to 0.95.

4.1 Constant Q Ensemble

With Q constant, the Metropolis Monte Carlo algorithm becomes very simple. Equation (23) simply becomes

$$\Pi(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x}, \mathbf{x}'). \tag{26}$$

In other words, all trial configurations are accepted.

In Fig. 4, we plot F_{nw}—the non-wetting fractional flow—as a function of S_{nw}—the non-wetting saturation—where the circles and the squares denote the results from Monte Carlo and time stepping, respectively. The plots, as expected, show an S-shape. This is because the two immiscible fluids do not flow equally, and the one with higher saturation dominates. Hence, the curve does not follow the diagonal dashed line, which corresponds to $F_{nw} = S_{nw}$, shown in the figure. Rather, F_{nw} is less than S_{nw} for low values of S_{nw} and higher than S_{nw} for higher value of S_{nw}. It therefore crosses the $F_{nw} = S_{nw}$ line at some point, which is not at $S_{nw} = 0.5$. This is due to the asymmetry between the two fluids, as one is more wetting than the other with respect to the pore walls. This behavior is more prominent for the lower value of Ca, as capillary forces play a more dominant role. The curves from the Monte Carlo and time-stepping calculations fall on top of each other for most of the lower to intermediate range of the saturation values, and we only see some difference at very high or low S_{nw}. We will present a more quantitative comparison between the results of Monte Carlo and time stepping later in Sect. 4.4. The variation of total pressure drop ΔP for the two capillary numbers as a function of S_{nw} is shown in Fig. 5. Similar to the fractional flow plots, we see that the results are same for Monte Carlo and time stepping for a wide range of S_{nw}. We
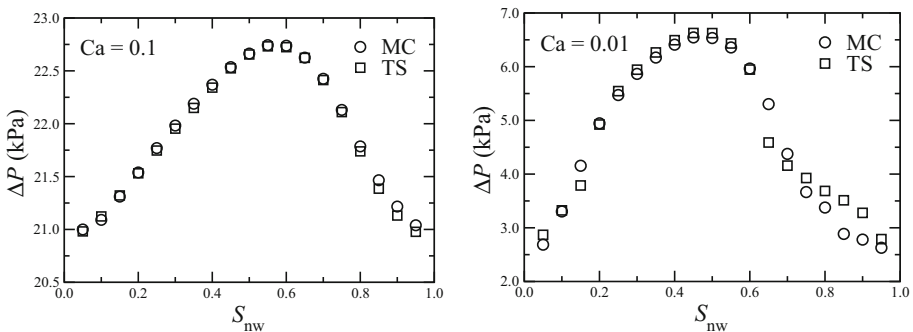

Fig. 5 Values of pressure difference (ΔP) for constant flow rate (Q) in the steady state as a function of non-wetting saturation (S_{nw}) for the capillary numbers Ca = 0.1 and 0.01 obtained via Monte Carlo (MC) simulations and time stepping (TS). The data are averaged over 10 samples

only see differences at high values of S_{nw}. ΔP increases with S_{nw}, reaching a maximum at some intermediate saturation and then decreases again. When S_{nw} increases from zero, more and more interfaces appear in the system causing an increase in capillary barriers associated with interfaces. As the total flow rate Q is constant, a higher pressure is needed to overcome the capillary barriers. The decrease of ΔP after the maximum is due to the decrease in the number of interfaces blocking the fluids.

4.2 Constant ΔP Ensemble

We now turn to the constant pressure ensemble. Here we keep ΔP constant throughout the calculations. In this case, the Metropolis Monte Carlo algorithm, Eq. (23), becomes

$$\Pi(\mathbf{x}, \mathbf{x}') = \pi(\mathbf{x}, \mathbf{x}') \ \min\left(1, \frac{Q(\mathbf{x}, \Delta P)}{Q(\mathbf{x}', \Delta P)}\right). \tag{27}$$

Results for the simulations with constant ΔP are shown in Figs. 6 and 7. Simulations are performed for two different values of ΔP, 15 and 6.5 kPa. The steady-state values of F_{nw} show similar variation with S_{nw} as in the constant Q ensemble, and we see good agreement between the results for Monte Carlo and time stepping for a wide range of S_{nw}. Here Q varies
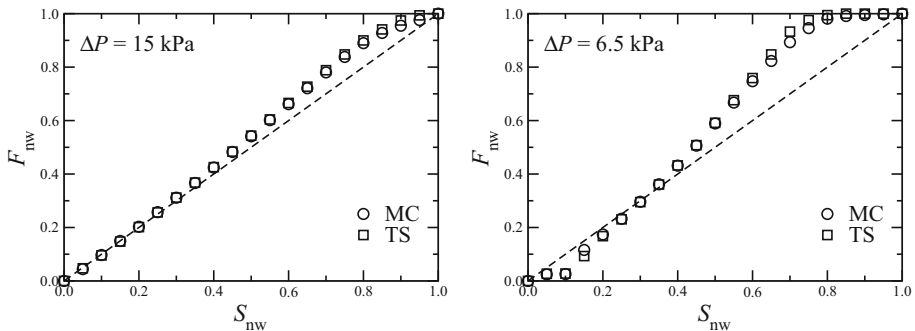

Fig. 6 Non-wetting fractional flow (F_{nw}) as a function of non-wetting saturation in the steady state for constant ΔP ensemble. Results are presented for Monte Carlo (MC) and time stepping (TS) for two different overall pressure drops $\Delta P = 15$ and 6.5 kPa. As Q varies with saturation for constant ΔP, Ca is not constant here, which is demonstrated in the next Fig.7. The data are averaged over 10 samples
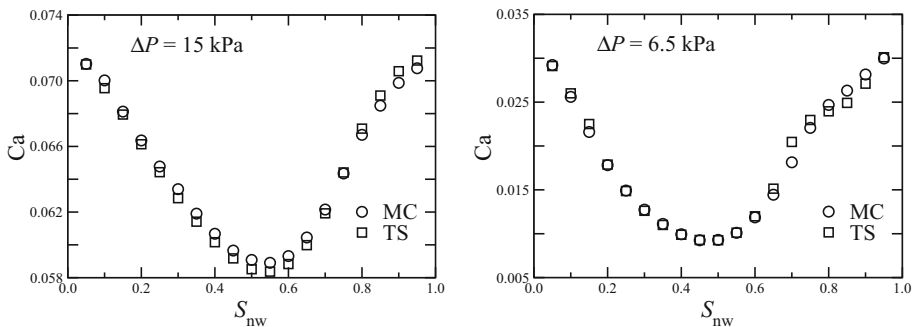

Fig. 7 Capillary numbers, calculated from the total flow rates (Q), in the steady state as a function of the non-wetting saturation S_{nw} for constant ΔP ensemble. Results are compared between Monte Carlo (MC) and time stepping (TS). The data are averaged over 10 samples

Table 1 Percentage of rejected configurations in the constant ΔP ensemble

Δp	S_{nw}	Rejections (%)
15 kPa	0.3	2.1
	0.5	2.3
	0.7	1.5
6.5 kPa	0.3	8.8
	0.5	11.6
	0.7	4.2

with the saturation, and the corresponding capillary numbers are plotted in Fig. 7 for Monte Carlo and time stepping. As discussed before, the number of interfaces first increases with the increase in saturation from zero, reaches a maximum value, and then decreases again as S_{nw} approaches 1. The pressure is constant here, so the total flow rate decreases with increasing capillary barriers at the interfaces and correspondingly Ca varies as in Fig. 7. Here again, good match between the results Monte Carlo and time stepping can be observed.

We show in Table 1 the percentage of rejections for the data shown in Fig. 7. The number of rejections is in all cases quite small. This can be understood as follows. Set $Q(\mathbf{x}, \Delta P) = Q$ and $Q(\mathbf{x}', \Delta P) = Q + \delta$ where δ may be positive or negative. Hence, the probability to accept the new configuration is

$$\min\left(1, \frac{Q(\mathbf{x}, \Delta P)}{Q(\mathbf{x}', \Delta P)}\right) = \min\left(1, 1 - \frac{\delta}{Q}\right), \tag{28}$$

where we have assumed $\delta \ll Q$. With a small δ, the probability to reject the trial configuration is small. This is reflected in Table 1.

4.3 Computational Cost

Here, we present a detailed comparative analysis of the computational cost of the two algorithms. We do this by measuring the computational time (T_{MC} for the Monte Carlo method and T_{TS} for the time-stepping method, respectively) for different system sizes L.

We use the conjugate gradient method to solve the Kirchhoff equations. This is an iterative solver. When the network contains $L \times L$ links ($L^2/2$ nodes), each iteration demands $L^2/2$ operations. The number of iterations necessary to solve the equations *exactly* scales as L^2, making the total cost scale as L^β, where $\beta = 2 + 2 = 4$. However, in practice, the number of iterations necessary to reach the solution of the Kirchhoff equations to within machine precision is much lower than that needed for the theoretically exact solution. As we shall see, β is much smaller than four.

The number of time steps needed to push one pore volume through the network is n_t. We expect it to depend on L as $n_t = aL^\tau$, where a is a prefactor essentially measuring the number of time steps on the average it takes for an interface to cross a link. In our calculations, this is of the order of 10. Intuitively, this number should be proportional to the length of the network, L, making $\tau = 1$. In practice, as we shall see, it is slightly larger.

For each time step, the conjugate gradient demands $t_{cg} = bL^\beta$ operations where b is another prefactor. The total computational time (T_{TS}) per pore volume is then

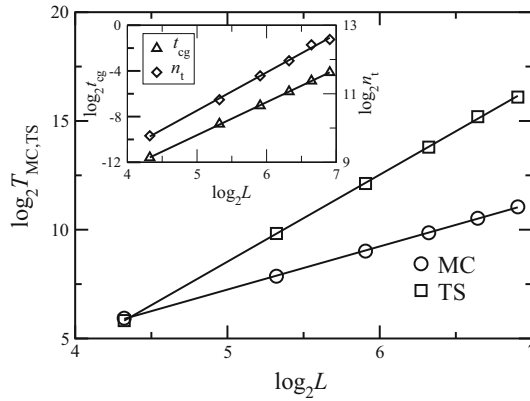$$T_{TS} = n_t \times t_{cg} = abL^{\tau+\beta} = abL^{\alpha_{TS}}, \tag{29}$$

Fig. 8 Plot of total computational time, $T_{MC,TS}$ (in seconds) used by Monte Carlo (MC) and time stepping (TS) for different system sizes (L). Here the time-stepping procedure is run for 100 injected pore volumes and in the Monte Carlo method, we do 25 sweeps. Each update is based on running the sub-system for 4 injected sub-network pore volumes. In this way, when $\Lambda = L = 20$, the timing of the two methods is equal. We use the CQ ensemble with Ca $= 0.1$ and $S_{nw} = 0.4$. Six different system sizes, $L = 20, 40, 60, 80$ and 120, are considered. From the slopes, the exponents α for time stepping and Monte Carlo are found. For Monte Carlo, we find $\alpha_{MC} = 1.98 \pm 0.01$, which is close to the theoretically expected value $T_{MC} \sim L^2$ (*see text*). However, for time stepping, we find $\alpha_{TS} = 3.99 \pm 0.03$, which is much smaller than theoretical expectation—$T_{TS} \sim L^5$. In the inset, we plot the average time, t_{cg}, taken by the conjugate gradient solver to solve one entire pressure field. We find $t_{cg} \sim L^{2.88 \pm 0.02}$. The number of time steps per pore volume, n_t, scales as $n_t \sim L^{1.11 \pm 0.03}$. Combining these two results, we find that the computational time for the time-stepping procedure to scales as $T_{TS} \sim L^{3.99}$

where $\alpha_{TS} = \tau + \beta$. Based on the theoretical considerations above, setting $\beta = 4$ and $\tau = 1$, we have $T_{TS} \sim L^5$. The actual computational time measured using the `clock()` function in C is plotted in Fig. 8 for Ca $= 0.1$ and $S_{nw} = 0.4$. We find that T_{TS} scales with L with an exponent $\alpha_{TS} = 3.99 \pm 0.03$, which is much smaller than 5. Measuring n_t and t_{cg} independently gives $\tau = 1.11 \pm 0.03$ and $\beta = 2.88 \pm 0.02$, see the inset in Fig. 8.

For the Monte Carlo algorithm, each sweep ideally contains $(L/\Lambda)^2$ individual Monte Carlo updates. Each Monte Carlo update consists of time stepping a sub-lattice of size $\Lambda \times \Lambda$. Hence, the cost of a Monte Carlo update is $ab\Lambda^{\alpha_{TS}}$ when using Eq. (29). However, each time stepping of a sub-lattice is followed by solving the Kirchhoff equations for the *entire* lattice in order to determine Q for the trial configuration. The cost of this operation is bL^β. The time per Monte Carlo sweep is then

$$T_{MC} = \left(\frac{L}{\Lambda}\right)^2 \left[4ab\Lambda^{\alpha_{TS}} + bL^\beta\right] = 4ab\Lambda^{\alpha_{TS}-2}L^2 + \frac{b}{\Lambda^2} L^{2+\beta}, \qquad (30)$$

where $\alpha_{TS} = 3.99$ and $\beta = 2.88$. The factor "4" signifies that we time step the sub-lattice for four pore volumes. By setting $a \approx 10$ and $\Lambda = 20$, the first term will dominate compared to the second term on the right-hand side of this equation if $4a\Lambda^{\alpha_{TS}} \approx 6.4 \times 10^6 > L^{2.88}$ or $L > 230$ where the second term, which scales as $L^{4.88}$, starts dominating. It is this behavior we see in Fig. 8: The computational time in the Monte Carlo method scales according to the first term, i.e., as L^2.

Hence, we summarize: The time-stepping procedure scales as $L^{3.99}$, whereas the Monte Carlo algorithm scales as $L^{1.98}$, as shown in Fig. 8.
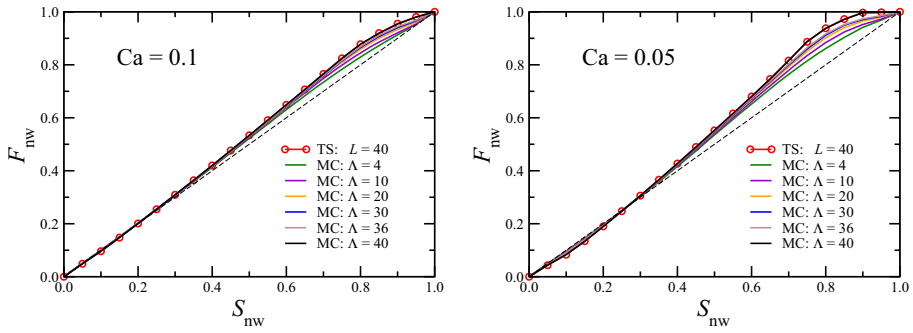
Fig. 9 Non-wetting fractional flow F_{nw} as a function of non-wetting saturation S_{nw} for time stepping compared to Monte Carlo for different sub-network sizes (Λ) in the constant Q ensemble. The size of the network, L, is 40 for both Monte Carlo and time stepping
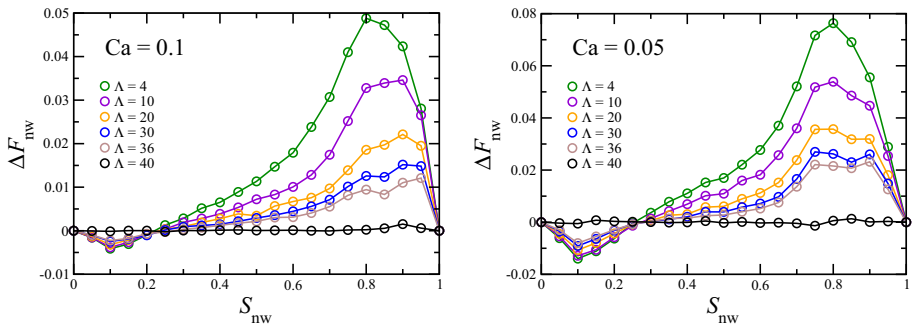

Fig. 10 Difference of the non-wetting fractional flow (ΔF_{nw}) between time stepping and Monte Carlo for different values of Λ is plotted as a function of S_{nw}. ΔF_{nw} fluctuates around zero for $\Lambda = L$, and a systematic increase is observed with the decrease in Λ for the whole range of S_{nw}

4.4 Limitations

A closer inspection of Figs. 4, 5, 6 and 7 shows that the match between the Monte Carlo and the time-stepping procedures is good, but not perfect. In this section, we discuss the discrepancies between the two methods quantitatively.

We show in Fig. 9 the non-wetting fractional flow for a 40×40 network using both time stepping and Monte Carlo with sub-network size Λ ranging from 4 to 40. Notice that we also consider the sub-network size 40, which is equal to L. The calculations here are done in the constant Q ensemble with a capillary number Ca equal to 0.1 or 0.05. As we see, there is a systematic deviation between the time stepping and the Monte Carlo results that increases with increasing non-wetting saturation S_{nw}. This deviation is highlighted in Fig. 10 where the difference between the time stepping and the Monte Carlo results for different Λ is shown. We note that the difference between the Monte Carlo and the time stepping decreases with increasing capillary number Ca. This is, however, to be expected, as for infinite Ca, any curve, Monte Carlo or time stepping, must fall on the diagonal of Fig. 9.

In Fig. 11, we show the discrepancy between the pressure drop ΔP using time stepping and Monte Carlo for different sub-lattice size Λ. The systematics seen in the fractional flow
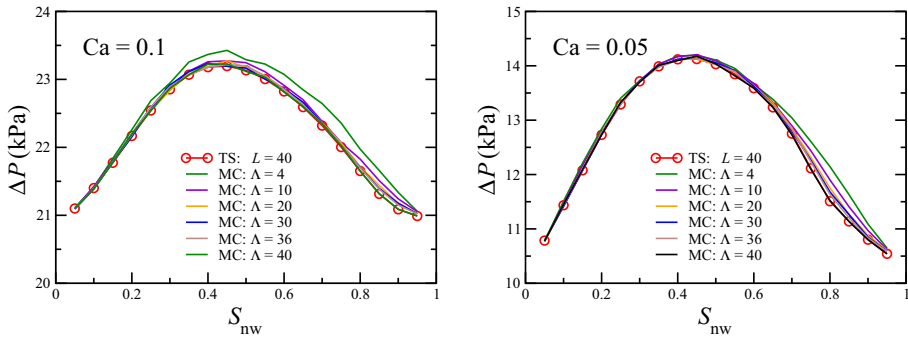
Fig. 11 Pressure difference ΔP as a function of non-wetting saturation S_{nw} for time stepping compared with Monte Carlo for different sub-network sizes (Λ) in the CQ ensemble. The size of the network, L, is 40 for both Monte Carlo and time stepping
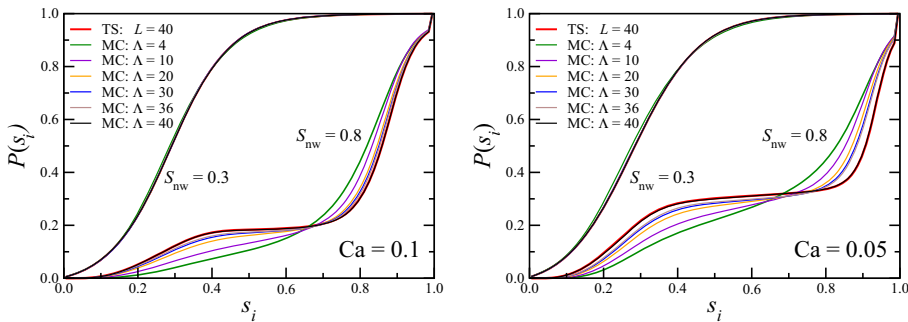

Fig. 12 Comparison of cumulative distribution $P(s_i)$ of link saturation s_i for time stepping and for Monte Carlo with different sub-system sizes. For $S_{nw} = 0.3$, $P(s)$ for Monte Carlo match with time stepping for all the sub-system sizes, whereas for $S_{nw} = 0.8$, a systematic difference in $P(s)$ is observed for $\Lambda < L$

data, Figs. 9 and 10, where the difference grows with increasing non-wetting saturation is much less pronounced in this case.

In Fig. 12, we show histograms over the non-wetting saturation of the links. That is, we measure how much non-wetting fluid each link contains. When the overall non-wetting saturation $S_{nw} = 0.3$, there is essentially no difference between the time stepping and the Monte Carlo result. However, for $S_{nw} = 0.8$, there is a difference that depends on the sub-lattice size Λ. This difference, measured as the area between the time stepping and the Monte Carlo histograms, is shown in Fig. 13 as a function of S_{nw}. The picture seen here resembles that seen for the non-wetting fractional flow (Fig. 9): the difference grows with increasing S_{nw}.

When the non-wetting saturation S_{nw} is small, the non-wetting fluid will form bubbles or small clusters surrounded by the wetting fluid. As S_{nw} is increased, these clusters grow in size until there is a percolation-type transition where the wetting fluid starts forming clusters surrounded by the non-wetting fluid. This scenario has been studied experimentally by Tallakstad et al. (2009a, b). They argued that there is a length scale l^*. Clusters that are larger than this length scale will move, whereas clusters that are smaller will be held in place by the capillary forces. The Monte Carlo algorithm calls for selecting a sub-network, which is then "lifted" out of the system, "folded" into a torus and then time stepped. The boundaries
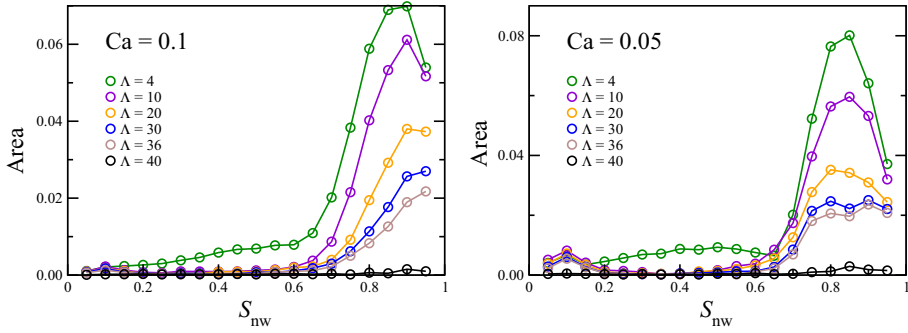
Fig. 13 Area between the $P(s_i)$ curves (Fig. 12) for time stepping and for Monte Carlo with different subsystem sizes as a function of S_{nw}
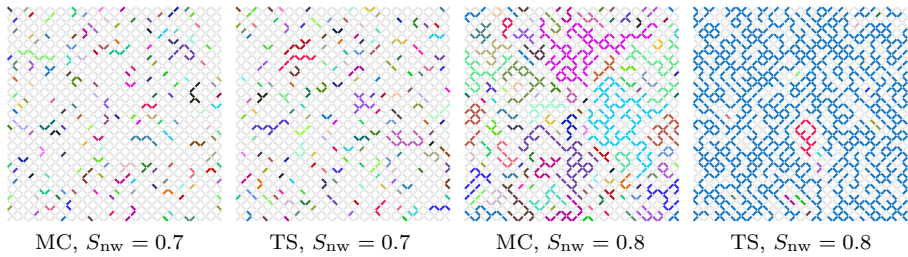

MC, $S_{nw} = 0.7$ TS, $S_{nw} = 0.7$ MC, $S_{nw} = 0.8$ TS, $S_{nw} = 0.8$

Fig. 14 Typical non-wetting clusters for Monte Carlo (MC) and time stepping (TS) at Ca = 0.05. The network is of 40×40 links, and the sub-network size for Monte Carlo is 20×20 links. Each cluster is marked with *different colors* so that the structure is readily visible

of the sub-network will cut through clusters and mobilize these. This changes the cluster structure from that of the time-stepping procedure.

In order to investigate this, we have studied the cluster structure in the model under Monte Carlo and time stepping. In order to do this, we identify the non-wetting clusters. To do this, two nodes are considered to be part of the same cluster if the link between them has a non-wetting saturation more than a threshold value, a clip threshold c_t. Here, we use a clip threshold equal to $c_t = 0.9$ (Ramstad and Hansen 2009). In Fig. 14, we show typical cluster structures for two different non-wetting saturations obtained with Monte Carlo and with time stepping. For $S_{nw} = 0.7$, the non-wetting clusters are still quite small, and there is no discernable difference between the configurations obtained with time stepping and with Monte Carlo. However, for $S_{nw} = 0.8$, there is one dominating cluster in the time-stepping case, whereas the clusters are more broken up in the Monte Carlo case.

We measure this qualitative difference in cluster structure for $S_{nw} = 0.8$ by recording the cluster size distribution for the two types of updating, see Fig. 15. When following the time-stepping procedure, we run the system for 500 pore volumes. During the last 125 pore volumes injected (1/4th of the total), we measure the cluster size distribution after passing each pore volume of fluids. When using Monte Carlo, we run the system for 400 Monte Carlo updates. We record the cluster size distribution for every of the last 100 updates. In both the time stepping and Monte Carlo runs, we average over 10 samples. The number of links belong to a cluster defines the size of that cluster. The total number of clusters is N_{total} and the number of clusters of size k that we record is N_k. We show $P(k) = N_k/N_{total}$ in the figure. For $S_{nw} = 0.6$ and 0.7, there is no discernable difference in the cluster structure

Fig. 15 Cluster size distribution $P(k) = N_k/N_{\text{total}}$ versus cluster size k for time stepping and Monte Carlo. The *blue circles* signify the Monte Carlo data and the *red circles* the time-stepping data. The *red* and *blue* curves with triangles pointing upwards or downwards signify the Monte Carlo and time-stepping data after logarithmic binning. Here $L = 40$ and $\Lambda = 20$. The data are averaged over 10 samples
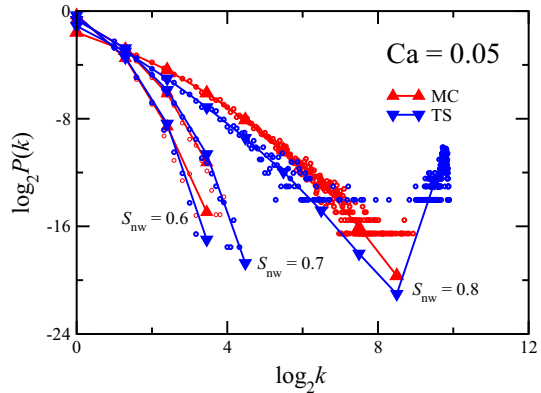
between the Monte Carlo and the time-stepping procedures. However, for $S_{\text{nw}} = 0.8$, there are differences. For every k, the number of clusters during the Monte Carlo updating procedure is larger than for the time-stepping procedure, except for the largest clusters, the percolating cluster seen in Fig. 14. This supports the supposition that the Monte Carlo breaks up the large non-wetting clusters.

Clearly, for the Monte Carlo algorithm to be perfected, this tendency of chopping up large non-wetting clusters needs to be counteracted. Presumably, this is a problem that decreases with increasing system and sub-lattice size as it is a boundary effect.

5 Conclusion

We have in this work presented a new Monte Carlo algorithm for immiscible two-phase flow in porous media under steady-state conditions using network models. It is based on the Metropolis transition probability (23), which in turn is build upon the configuration probability (19) which we derive here. By steady-state conditions, we mean that the macroscopic parameters that describe the flow such as pressure difference, flow rate, fractional flow rate and saturation all have well-defined means that stay constant. On the pore level, however, clusters flow, merge, break up, and so on. The flow may be anything but stationary. We described the algorithm in Sect. 3.2.2.

Computationally, the Monte Carlo algorithm is very fast compared to time stepping. We find that the time-stepping procedure when implemented on a square lattice demands a computing time that scales as the linear size of the lattice, L, to the fourth power, whereas the Monte Carlo method scales as the linear size to the second power, see Sect. 4.3. However, there is another term that contributes to the computing time in the Monte Carlo procedure which scales as $L^{4.88}$. This term has a prefactor associated with it, which is very small compared to the other term scaling as L^2. For L up to about 230, this term is small compared to the first one.

5.1 Open Questions

There are open questions with respect to the Metropolis Monte Carlo approach that we present here. The most important step in the direction of constructing such an approach is to identify the configuration probability (19). The second most important step is to provide a

way to generate trial configurations that obey the symmetry requirement (21). Section 3.2.2 is concerned with this.

We see three challenges that will need to be overcome before the Monte Carlo algorithm that we propose here is fully capable of replacing time stepping.

– The Monte Carlo algorithm needs to be generalized to irregular networks, e.g., those based on reconstructed porous media (Blunt et al. 2013).
– The necessity to solve the Kirchhoff equations for the entire pore network once for every Monte Carlo update will slow down the algorithm when it is implemented for large systems. Ideally, one should find a way to circumvent this necessity.
– The Monte Carlo algorithm has a tendency to break up large non-wetting clusters as described in Sect. 4.4. This is a problem for large non-wetting saturations. It is most probably a boundary effect that comes from the way the sub-networks are constructed. However, it needs to be overcome if the algorithm is to be useful for the entire range of saturations.

Overcoming these three challenges will allow network models to take advantage to the full of the ongoing revolution in pore space characterization.

We have in this article presented a first attempt at constructing a Markov Chain Monte Carlo algorithm based on the configurational probability (19). There is no reason not to believe that other ways of constructing such Monte Carlo algorithms might be possible that are both faster and do not pose the challenges listed above.

References

Aker, E., Måløy, K.J., Hansen, A., Batrouni, G.G.: A two-dimensional network simulator for two-phase flow in porous media. Transp. Porous Media **32**, 163 (1998)

Armstrong, R.T., Berg, S., Dinariev, O., Evseev, N., Klemin, D., Koroteev, D., Safanov, S.: Modeling of pore-scale phenomena using density functional hydrodynamics. Transp. Porous Media **112**, 577 (2016)

Batrouni, G.G., Hansen, A.: Fourier acceleration of iterative processes in disordered media. J. Stat. Phys. **52**, 747 (1988)

Blunt, M.J., Bijeljic, B., Dong, H., Gharbi, O., Iglauer, S., Mostaghimi, P., Paluszny, A., Pentland, C.: Pore-scale imaging and modeling. Adv. Water Resour. **51**, 197 (2013)

Hansen, A., Sinha, S., Bedeaux, D., Kjelstrup, S., Savani, S., Vassvik, M.: A new set of equations describing immiscible two-phase flow in homogeneous porous media. arXiv:1605.02874 (2016)

Joekar-Niasar, V., Hassanizadeh, S.M.: Analysis of fundamentals of two-phase flow in porous media using dynamic pore-network models: a review. Crit. Rev. Environ. Sci. Technol. **42**, 1895 (2012)

Knudsen, H.A., Aker, E., Hansen, A.: Bulk flow regimes and fractional flow in 2D porous media by numerical simulations. Transp. Porous Media **47**, 99 (2002)

Krauth, W.: Statistical Mechanics: Algorithms and Computations. Oxford University Press, Oxford (2006)

Landau, L.D., Binder, K.: A Guide to Monte Carlo Simulations in Statistical Mechanics, 4th edn. Cambridge University Press, Cambridge (2015)

Liu, M.B., Liu, G.R.: Smoothed particle hydrodynamics (SPH): an overview and recent developments. Arch. Comput. Methods Eng. **17**, 25 (2010)

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Calculation of equations of state by fast computing machines. J. Chem. Phys. **134**, 1087 (1953)

Ovaysi, S., Piri, M.: Direct pore-level modeling of incompressible fluid flow in porous media. J. Comput. Phys. **229**, 7456 (2010)

Press, W.H., Teukolsky, S.A., Wetterling, W.T., Flannery, B.P.: Numerical Recipes 3rd Edition: The Art of Scientific Computing. Cambridge University Press, Cambridge (2007)

Ramstad, T., Hansen, A.: Cluster evolution in steady-state two-phase flow in porous media. Phys. Rev. E **73**, 026306 (2009)

Ramstad, T., Hansen, A., Øren, P.E.: Flux-dependent percolation transition in immiscible two-phase flows in porous media. Phys. Rev. E **79**, 036310 (2009)

Ramstad, T., Øren, P.E., Bakke, S.: Simulation of two-phase flow in reservoir rocks using a lattice Boltzmann method. SPE J. **15**, 917 (2010)

Ramstad, T., Idowu, N., Nardi, C., Øren, P.E.: Relative permeability calculations from two-phase flow simulations directly on digital images of porous rocks. Transp. Porous Media **94**, 487 (2012)

Savani, I., Bedeaux, D., Kjelstrup, S., Sinha, S., Vassvik, M., Hansen, A.: Ensemble distribution for immiscible two-phase flow in two-dimensional networks. arXiv:1606.02569. (2016)

Sinha, S., Hansen, A., Bedeaux, D., Kjelstrup, S.: Effective rheology of bubbles moving in a capillary tube. Phys. Rev. E **87**, 025001 (2013)

Tallakstad, K.T., Knudsen, H.A., Ramstad, T., Løvoll, G., Måløy, K.J., Toussaint, R., Flekkøy, E.G.: Steady-state two-phase flow in porous media: statistics and transport properties. Phys. Rev. Lett. **102**, 074502 (2009a)

Tallakstad, K.T., Løvoll, G., Knudsen, H.A., Ramstad, T., Flekkøy, E.G., Måløy, K.J.: Steady-state two-phase flow in porous media: an experimental study. Phys. Rev. E **80**, 036308 (2009b)

Tartakovsky, A.M., Meakin, P.: A smoothed particle hydrodynamics model for miscible flow in three-dimensional fractures and the two-dimensional Rayleigh–Taylor instability. J. Comput. Phys. **207**, 610 (2005)

Tørå, G., Øren, P.E., Hansen, A.: A dynamic network model for two-phase flow in porous media. Transp. Porous Media **92**, 145 (2011)

Washburn, E.W.: Dynamics of capillary flow. Phys. Rev. **17**, 273 (1921)