Two-Phase Flow Simulation by AMMoC, an Adaptive Meshfree Method of Characteristics

Armin Iske and Martin Käser

Abstract. Petroleum reservoir modelling requires effective multiscale methods for the numerical simulation of two-phase flow in porous media. This paper proposes the application of a novel meshfree particle method to the Buckley-Leverett model. The utilized meshfree advection scheme, called **AMMoC**, is essentially a method of characteristics, which combines an adaptive semi-Lagrangian method with local meshfree interpolation by polyharmonic splines. The method **AMMoC** is applied to the five-spot problem, a well-established model problem in petroleum reservoir simulation. The numerical results and subsequent numerical comparisons with two leading commercial reservoir simulators, ECLIPSE and FrontSim, show the good performance of our meshfree advection scheme **AMMoC**.

1 Introduction

Petroleum reservoir simulators help oil companies to make effective use of expensive data collected through field measurements, data processing and interpretation. In fact, reservoir simulators are among the very few tools which are available for modelling various physical multiscale phenomena within hydrocarbon reservoirs. Hydrocarbon exploration and production, in particular, requires computational methods for the numerical simulation of two-phase flow in porous media.

Two-phase flow modelling is concerning the displacement of one fluid, say oil, by another, say water, within a reservoir. This model problem may be characterized by the injection of a wetting fluid (water) into the reservoir at a particular location, displacing the non-wetting fluid (oil), which is being withdrawn at another location.

Due to the physical interaction between the two phases, water and oil, this production process leads to a moving shock front at the interface between the two phases. The evolution of the shock front is of primary importance for the production, where the goal is to withdraw as much oil as possible before the *breakthrough*, when water arrives at the production site.

However, the geometry of the moving shock front may be very complicated, mainly due to varying flow velocities. In order to model the propagation of the shock front effectively, numerical simulation of two-phase flow essentially requires customized multiscale algorithms, which manage to capture and resolve important local features of the flow. To this end, *adaptive* numerical algorithms are necessary in order to combine small computational costs with high accuracy.

In our previous work [8], a novel concept for the *meshfree* numerical simulation of multiscale phenomena in transport processes is proposed. The resulting particle-based advection scheme, called **AMMoC**, is essentially a method of characteristics, which combines an adaptive semi-Lagrangian method with local meshfree interpolation by polyharmonic splines. This paper further supports the utility of this meshfree concept. To this end, we propose the application of the method **AMMoC** to numerical simulation of two-phase flow in porous media.

The outline of this paper is as follows. Further discussion on two-phase flow in petroleum reservoirs is provided in the following Section 2, where in particular the mathematical formulation of the flow problem is given. Basic ingredients of our meshfree advection scheme **AMMoC** are then explained in Section 3, where also some details concerning its implementation are discussed and the required computational costs are analyzed. Numerical results are finally provided in Section 4, where **AMMoC** is applied to the *five-spot problem*, a popular test case scenario in oil reservoir simulation. The good performance of our method **AMMoC** is shown by numerical comparisons with two leading commercial reservoir simulators, ECLIPSE and FrontSim.

2 Two-Phase Flow in Petroleum Reservoirs

Petroleum reservoirs contain hydrocarbons and other chemicals trapped in the pores of a rock. Waterflooding is one effective technique for the exploration and production of hydrocarbons from petroleum reservoirs. This technique involves drilling wells into the rocks, *injectors* and *producers*. By the injection of water through the injectors, hydrocarbons are then, due to the resulting pressure, pushed into the rocks and forced to flow towards the producers.

A somewhat simplified, but fairly realistic model problem for petroleum reservoir simulation is the *Buckley-Leverett* model [10]. This standard model is concerning two-phase flow of two immiscible and incompressible fluids, say water and oil, within a porous medium, where diffusive effects, such as capillary pressure, are ignored and gravitational forces are neglected.

In the following of this section, the governing equations of this particular two-phase flow model are introduced, and some of their fundamental properties are recalled. A more comprehensive discussion concerning geophysical aspects, especially from the viewpoint of petroleum reservoir simulation, can be found in the textbooks [5, 19, 20].

Let us first turn to the governing equation for the fluid flow. Due to mass conservation, the flow for each of the two individual phases is described by a time-dependent hyperbolic conservation law. The two resulting *mass balance*

equations are the *mass conservation of water*,

$$\phi(x)\frac{\partial}{\partial t}u_w(t,x) + \nabla \cdot \mathbf{v}_w(t,x) = 0, \tag{1}$$

and the *mass conservation of oil*,

$$\phi(x)\frac{\partial}{\partial t}u_o(t,x) + \nabla \cdot \mathbf{v}_o(t,x) = 0. \tag{2}$$

In both (1) and (2), the scalar field $\phi(x)$ denotes the *porosity* of the medium, which determines the volume accessible to fluid flow. Hence, for any *homogeneous* medium, its porosity ϕ is constant.

Moreover, the time-dependent vector fields $\mathbf{v}_w(t,x)$ in (1) and $\mathbf{v}_o(t,x)$ in (2) are the phase *velocities*, and $u_w(t,x)$ in (1) and $u_o(t,x)$ in (2) are the *saturations* of water and oil, respectively. The saturations, u_w and u_o, are the corresponding fractions of available volume, of water and oil, in the pores of the medium. Therefore, we have $0 \leq u_w \leq 1$ and $0 \leq u_o \leq 1$, and moreover,

$$u_w(t,x) + u_o(t,x) = 1, \tag{3}$$

since the medium is assumed to contain only two phases, water and oil.

The two phase velocities, $\mathbf{v}_w(t,x)$ in (1) and $\mathbf{v}_o(t,x)$ in (2), are determined by *Darcy's law*

$$\mathbf{v}_w(t,x) = -\mathbf{K}(x)\,\frac{k_w(u_w)}{\mu_w}\,\nabla p(t,x), \tag{4}$$

$$\mathbf{v}_o(t,x) = -\mathbf{K}(x)\,\frac{k_o(u_o)}{\mu_o}\,\nabla p(t,x), \tag{5}$$

where $p(t,x)$ is the reservoir *pressure*, $\mathbf{K}(x)$ is the *permeability tensor* of the porous medium, and $k_w(u_w)$ in (4) and $k_o(u_o)$ in (5) are the *relative permeabilities* of the corresponding phases. Moreover, the scalar parameters μ_w in (4) and μ_o in (5) denote the fluids' *viscosities*, so that the ratios

$$M_w(u_w) = \frac{k_w(u_w)}{\mu_w} \quad \text{and} \quad M_o(u_o) = \frac{k_o(u_o)}{\mu_o}$$

yield the phase *mobilities*. The *total mobility* is thus given by $M = M_w + M_o$.

By combining the two fluids' mass balance equations, (1) and (2), with the relation (3), we obtain the *incompressibility relation*

$$\nabla \cdot (\mathbf{v}_w(t,x) + \mathbf{v}_o(t,x)) = \nabla \cdot \mathbf{v}(t,x) = 0,$$

which states that the total fluid velocity $\mathbf{v}(t,x)$ is divergence-free.

Now the phase velocity of water can be expressed as

$$\mathbf{v}_w(t,x) = \mathbf{v}(t,x) \cdot f_w(u_w), \tag{6}$$

where $f_w(u_w)$ is the *flux tensor*, given by the ratio $f_w(u_w) = M_w(u_w)/M(u_w)$ between the phase mobility $M_w(u_w)$ of water and the total mobility M.

A corresponding relation for the phase velocity $\mathbf{v}_o$ of oil can be established accordingly. But we wish to further simplify the notation. To this end, we let $u \equiv u_w$, and so $u_o = 1 - u$ by (3), and we also let $f(u) = f_w(u_w)$.

We remark that the fractional flow function f in (6) is monotonically increasing and it satisfies $0 \leq f(u) \leq 1$ for $u \in [0, 1]$. The following considerations rely on the *Corey model* [5], according to which the relative permeabilities, k_w and k_o, are quadratic functions of the form

$$k_w(u) = u^2, \qquad k_o(u_o) = (1-u)^2.$$

This yields

$$M(u) = \frac{u^2}{\mu_w} + \frac{(1-u)^2}{\mu_o},$$

for the total mobility and so in this case the fractional flow function f in (6) is given by

$$f(u) = \frac{u^2}{u^2 + \mu(1-u)^2}, \tag{7}$$

where we let $\mu = \mu_w/\mu_o$ for the ratio of the two fluids' viscosities.

We summarize the discussion of this section as follows.

The *Buckley-Leverett equation*

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla f(u) = 0, \tag{8}$$

together with the *incompressibility relation*

$$\nabla \cdot \mathbf{v}(t, x) = 0, \tag{9}$$

and *Darcy's law*

$$\mathbf{v}(t, x) = -\mathbf{K}(x) M(u) \nabla p(t, x), \tag{10}$$

describes the flow of two immiscible incompressible fluids, water and oil, through a porous *homogeneous* medium, $\phi \equiv 1$, in the absence of capillary pressure and gravitational effects (see also [5, 19, 20] and the related discussion in [21]).

The solution u of (8),(9),(10) is the *saturation* of the wetting fluid (water). Hence, the value $u(t, x)$ is, at a time t and at a point x, the fraction of available volume (in the pores of the medium) filled with water, and so $u = 1$ means pure water, and $u = 0$ means pure oil.

We remark that the incompressibility relation (9) together with Darcy's law (10) forms an elliptic equation. The Buckley-Leverett equation (8) is a hyperbolic equation, which develops discontinuities in the solution u, corresponding to a shock front at the interface between the two phases.

3 Meshfree Flow Simulation

This section introduces a novel concept for *meshfree* flow simulation, where special emphasis is placed on the particularities of the Buckley-Leverett model (8).

Generally speaking, meshfree methods provide very flexible, robust and reliable discretization techniques for multiscale simulation, which have recently gained much attention in many different applications from computational sciences and engineering, as well as in numerical analysis. Among a few others, prominent meshfree discretization techniques include the Meshless Local Petrov-Galerkin (MLPG) Method, see the papers [3, 4, 17] and the textbook [2] of Atluri for an up-to-date account and comprehensive treatment of the method and its rich variety of applications.

The resulting particle-based meshfree advection scheme of this paper, **AMMoC**, is essentially an **A**daptive **M**eshfree **M**ethod **o**f **C**haracteristics, which combines the well-established semi-Lagrangian method with local meshfree interpolation by polyharmonic splines [6, 7, 8] and customized adaption rules [7] required for the effective refinement and coarsening of flow particles. For further background on the *method of characteristics*, we refer to the textbook [13]. Moreover, a comprehensive overview over the semi-Lagrangian method is offered in [18, Section 7], and scattered data interpolation by polyharmonic splines (and other radial basis functions) is explained in the recent tutorial [15].

This section first explains a standard stabilization technique for the Buckley-Leverett equation (8). This is done in Subsection 3.1, where the relevant Cauchy problem for the viscous Buckley-Leverett equation is formulated. In order to give a short introduction to the basic ingredients of our meshfree advection algorithm **AMMoC**, some selected features of the meshfree method of characteristics are then explained in Subsection 3.2. This is followed by a short discussion on polyharmonic splines in Subsection 3.3. Some selected aspects concerning the implementation of **AMMoC** and the required computational costs are finally analyzed in Subsection 3.4.

For deeper insight into the ingredients of **AMMoC**, especially the implementation of the adaption rules and other computational aspects, we refer to our previous papers [6, 7, 8] and the recent research monograph [16].

3.1 Viscous Buckley-Leverett Equation

The Buckley-Leverett equation (8) is a time-dependent hyperbolic equation, whose solution u develops discontinuities, even for smooth initial data. Recall that the discontinuities of the saturation u are corresponding to the shock front at the interface between the two phases.

In order to model the propagation of the shock front, we consider the

viscous Buckley-Leverett equation

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla f(u) = \epsilon \cdot \Delta u, \tag{11}$$

on a computational domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, and for a compact time interval $I = [0, T]$, $T > 0$, where f is the fractional flow function in (7), $\epsilon > 0$ denotes a small *diffusion coefficient*, and Δ denotes the Laplace operator on $\mathbb{R}^d$.

We remark that this modification (11) of the Buckley-Leverett equation (8) relies on a standard stabilization technique, dating back to Burgers [11]. The idea of this stabilization is to transfer the *hyperbolic* equation (8) to a *parabolic* equation (11), which has, unlike (8), at any time $t > 0$, a smooth solution u, even for discontinuous initial data. Moreover, the solution u of the Buckley-Leverett equation (8) is approximated arbitrarily well by the solution $u \equiv u_\epsilon$ of the viscous Buckley-Leverett equation (11), provided that the diffusion coefficient ϵ is sufficiently small.

We consider solving the viscous Buckley-Leverett equation (11), in combination with given initial conditions

$$u(0, x) = u_0(x), \quad \text{for } x \in \Omega. \tag{12}$$

To this end, we work with our above mentioned meshfree advection scheme, **AMMoC**, whose basic ingredients are discussed in the remainder of this section.

3.2 Meshfree Method of Characteristics

The discretization of the given Cauchy problem (11), (12), suggested in [8], works with a finite set $\Xi \subset \Omega$ of nodes, each of which corresponds at a time $t \in I$ to one fluid particle. According to the basic concept of semi-Lagrangian advection, the equation (11) is integrated along the trajectories of the particles' streamlines. Moreover, the node set Ξ is adaptively modified during the simulation, where the adaption rules rely on a customized a posteriori error estimator, which is introduced in [7].

Starting point of our meshfree advection scheme is the Lagrangian form

$$\frac{du}{dt} = \epsilon \cdot \Delta u,$$

of (11), where $\frac{du}{dt} = \frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla f(u)$ is the *material derivative*. This leads us to the discretization

$$\frac{u(t + \tau, \xi) - u(t, \mathbf{x})}{\tau} = \epsilon \cdot \Delta u(t, \mathbf{x}), \tag{13}$$

where $\mathbf{x} \equiv \mathbf{x}(\xi)$ is the *upstream point*, corresponding to the node $\xi \in \Xi$. The upstream point $\mathbf{x}$ of ξ can be viewed as the position of a particle at time

t, which by traversing along its trajectory, arrives at ξ at time $t+\tau$, where $\tau > 0$ denotes the time step size. Adopting some standard notation from dynamical systems [12], we express the upstream point $\mathbf{x}$ of ξ as

$$\mathbf{x} = \Phi^{t,t+\tau}\xi, \tag{14}$$

where $\Phi^{t,t+\tau} : \Omega \to \Omega$ denotes the *continuous evolution* of the (backward) flow of the *ordinary differential equation* (ODE)

$$\dot{x} = \frac{dx}{dt} = \mathbf{a}(t, x), \tag{15}$$

with $\mathbf{a} = \frac{\partial f(u)}{\partial u}$ being the advection velocity.

Note that the exact location of $\mathbf{x}$ is usually unknown. Therefore, in order to compute an approximation $\tilde{\mathbf{x}} \approx \mathbf{x}$ numerically, we work with a specific *discrete evolution* $\Psi^{t,t+\tau}$ of the flow, corresponding to the continuous evolution $\Phi^{t,t+\tau}$ in (14). The operator $\Psi^{t,t+\tau}$ is given by any suitable numerical method for solving the above ODE (15), which allows us to express the resulting approximation $\tilde{\mathbf{x}}$ of $\mathbf{x}$ as

$$\tilde{\mathbf{x}} = \Psi^{t,t+\tau}\xi.$$

For the sake of brevity, we refrain from expanding details concerning the employed ODE solver of our preference, but rather refer to our previous paper [8].

Having computed $\tilde{\mathbf{x}} = \Psi^{t,t+\tau}\xi$, the desired approximation of $u(t+\tau, \xi)$ in (13) would thus be given by

$$u(t+\tau, \xi) = u(t, \tilde{\mathbf{x}}) + \tau \cdot \epsilon \Delta u(t, \tilde{\mathbf{x}}), \quad \text{for } \xi \in \Xi. \tag{16}$$

In order to determine the *unknown* function values $u(t, \tilde{\mathbf{x}})$, $\Delta u(t, \tilde{\mathbf{x}})$ in the right hand side of (16), we work with local interpolation by using polyharmonic splines. Some selected details concerning the relevant background of this particular interpolation scheme are briefly discussed in the following Subsection 3.3. For the moment be it sufficient to say that, on any given upstream point approximation $\tilde{\mathbf{x}}$, we determine a neighbouring set $\mathcal{N} \subset \Xi$ of current nodes around $\tilde{\mathbf{x}}$, all of whose values $u(t, \nu)$, $\nu \in \mathcal{N}$, are known. Then, we compute a polyharmonic spline interpolant s satisfying

$$s(\nu) = u(t, \nu), \quad \text{for all } \nu \in \mathcal{N}, \tag{17}$$

before we replace (16) by

$$u(t+\tau, \xi) = s(\tilde{\mathbf{x}}) + \tau \cdot \epsilon \Delta s(\tilde{\mathbf{x}}), \quad \text{for } \xi \in \Xi.$$

Altogether, the advection step $t \to t+\tau$ is accomplished as follows.

Algorithm 1 (Method of Characteristics).

INPUT: *Time step* τ*, nodes* Ξ*, values* $\{u(t,\xi)\}_{\xi\in\Xi}$.

FOR *each* $\xi \in \Xi$ **DO**

(a) *Compute the upstream point approximation* $\tilde{\mathbf{x}} = \Psi^{t,t+\tau}\xi$*;*

(b) *Determine* $s(\tilde{\mathbf{x}}) \approx u(t,\tilde{\mathbf{x}})$ *by local interpolation, i.e., solve (17);*

(c) *Advect by letting* $u(t+\tau,\xi) = s(\tilde{\mathbf{x}}) + \tau \cdot \epsilon\Delta s(\tilde{\mathbf{x}})$.

OUTPUT: *The values* $u(t+\tau,\xi)$*, for all* $\xi \in \Xi$*, at time* $t+\tau$.

3.3 Polyharmonic Spline Interpolation

In order to solve the *local* interpolation problem (17), we prefer to work with *polyharmonic splines*, which are popular tools for multivariate interpolation from scattered data. In this particular interpolation scheme, the interpolant s in (17) has the form

$$s = \sum_{\nu\in\mathcal{N}} c_\nu \cdot \phi_{d,k}(\|\cdot - \nu\|) + p, \qquad p \in \mathcal{P}_k^d, \tag{18}$$

where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^d$, and the polyharmonic spline $\phi_{d,k}$ is given by

$$\phi_{d,k}(r) = \begin{cases} r^{2k-d}\log(r), & \text{for } d \text{ even}, \\ r^{2k-d}, & \text{for } d \text{ odd}, \end{cases}$$

with $2k > d$. Moreover, $\mathcal{P}_k^d$ denotes the linear space of all d-variate polynomials of order at most k. We remark that the interpolation problem (17) has under constraints

$$\sum_{\nu\in\mathcal{N}} c_\nu p(\nu) = 0, \qquad \text{for all } p \in \mathcal{P}_k^d,$$

a unique solution, provided that the points in $\mathcal{N}$ are $\mathcal{P}_k^d$*-unisolvent*, i.e., for $p \in \mathcal{P}_k^d$,

$$p(\nu) = 0 \quad \text{for all } \nu \in \mathcal{N} \qquad \Longrightarrow \qquad p \equiv 0.$$

The stability and the approximation order of *local* polyharmonic spline interpolation has recently been analyzed in [14]. One of the key observation in [14] is that the Lagrange basis $(\lambda_\nu(x))_{\nu\in\mathcal{N}}$, and thus the *Lebesgue constant*

$$\Lambda(U,\mathcal{N}) = \max_{x\in U} \sum_{\nu\in\mathcal{N}} |\lambda_\nu(x)|, \qquad \text{for } \mathcal{N} \subset U \subset \Omega,$$

of the interpolation scheme is invariant under uniform scalings. This in turn leads to a *stable* algorithm for solving (17). Moreover, it shows that the *approximation order* of local polyharmonic spline interpolation around any $\tilde{\mathbf{x}}$ is k, i.e., for any point $\tilde{\mathbf{x}} + h(x - \tilde{\mathbf{x}}) \in U$, $h > 0$, and a fixed local neighbourhood U of $\tilde{\mathbf{x}}$ we have

$$|s^h(\tilde{\mathbf{x}} + h(x - \tilde{\mathbf{x}})) - u(t, \tilde{\mathbf{x}} + h(x - \tilde{\mathbf{x}}))| = \mathcal{O}(h^k),\ h \to 0, \quad \text{for } u(t, \cdot) \in C^k,$$

where s^h denotes the unique polyharmonic spline interpolant satisfying

$$s^h(\tilde{\mathbf{x}} + h(\nu - \tilde{\mathbf{x}})) = u(t, \tilde{\mathbf{x}} + h(\nu - \tilde{\mathbf{x}})), \qquad \text{for all } \nu \in \mathcal{N}.$$

For further details on local polyharmonic spline interpolation, we refer to [14].

Polyharmonic spline interpolation is also used in order to adaptively modify the current node set $\Xi \equiv \Xi(t)$ after each advection step of Algorithm 1, yielding a modified node set $\Xi \equiv \Xi(t + \tau)$. To this end, we work with an *error indicator*, which assigns to each current node $\xi \in \Xi(t)$ a *significance value*

$$\eta(\xi) = |s_{\mathcal{N} \setminus \xi} - u(t, \xi)|, \qquad \text{for } \xi \in \Xi, \tag{19}$$

where $s_{\mathcal{N} \setminus \xi}$ denotes the polyharmonic spline interpolant which interpolates the values $u(t, \nu)$, $\nu \in \mathcal{N}$, at a set $\mathcal{N} \subset \Xi \setminus \xi$ of current nodes in the neighbourhood of ξ. The error indicator $\eta : \Xi \to \mathbb{R}$ thus evaluates the local approximation quality around the nodes in $\Xi(t)$. The modification of $\Xi(t)$ is then accomplished by the removal of nodes with small significances, *coarsening*, whereas in the neighbourhood of nodes with large significances new nodes are inserted, *refinement*. For further details concerning the implementation of these adaption rules, see [7].

3.4 Implementation and Computational Complexity

In this subsection, we discuss some selected aspects concerning the implementation of the proposed advection scheme **AMMoC**, and we consider analyzing its computational complexity. To this end, let us first determine the computational costs required for the performance of *one* advection step, Algorithm 1, where the current node set Ξ, of size $N = |\Xi|$, is fixed. In the following, we analyze the computational costs required for each of the steps **(a)**–**(c)** in Algorithm 1 per node $\xi \in \Xi$.

First, note that the assignment in step **(c)** costs $\mathcal{O}(1)$ operations. As regards the performance of step **(a)**, recall that the computation of the upstream point approximation $\tilde{\mathbf{x}}$ relies on a specific ODE solver. In our implementation of **AMMoC**, we employ a recursion of the form

$$\beta_{k+1} = \tau \cdot \mathbf{a}(t + \tau/2, \xi - \beta_k/2)$$

in order to obtain after merely a few iterations a sufficiently accurate linear approximation $\beta \in \mathbb{R}^d$ to the trajectory arriving at ξ. This complies with the recommendations in [18, equation (7.66a)] and yields by $\tilde{\mathbf{x}} = \xi - \beta$ a sufficiently accurate upstream point approximation in constant time, i.e., at $\mathcal{O}(1)$ operations.

The performance of step **(b)** requires solving a local interpolation problem of the form (17) by a polyharmonic spline (18). This is accomplished by solving a linear equation system of *small* size, $(n+q)$-by-$(n+q)$, where $n = |\mathcal{N}|$ is the number of neighbouring nodes and $q = \binom{k-1+d}{d}$ is the dimension of the polynomial space $\mathcal{P}_k^d$, see [15] for details. We use a direct method for solving this *small* linear system, which requires $\mathcal{O}(n^3)$ operations, where $n \ll N$. In our numerical example in Subsection 4.2, for instance, we have $n \leq 15$ during the entire simulation, whereas the number N of current nodes is in the range $250 \leq N \leq 6000$, see Figure 1.

Now let us turn to the computational costs required for the adaptive modification of the nodes. Recall that after each advection step, by Algorithm 1, the current node set Ξ is modified. This is done by using the adaption rules proposed in our previous paper [7]. To be more precise, the modification of Ξ relies on the a posteriori error estimator $\eta : \Xi \to [0, \infty)$, introduced in the previous Subsection 3.3, which first assigns a significance value $\eta(\xi)$ in (19) to each node $\xi \in \Xi$. This basically requires computing the polyharmonic spline interpolant $s_{\mathcal{N}\setminus\xi}$ in (19). Like in step **(b)** of Algorithm 1, this is accomplished by solving a linear equation system of *small* size $(n+q)$-by-$(n+q)$, which costs only $\mathcal{O}(n^3)$ operations.

Moreover, according to the adaption rules in [7], either operation, the coarsening or the refinement of a node, can be accomplished in constant time, i.e., at $\mathcal{O}(1)$ operations. Therefore, the performance of both *one* advection step by Algorithm 1 and the subsequent node adaption requires at most $\mathcal{O}(N \cdot n^3)$ operations, where $N = |\Xi|$ and $n \ll N$ denotes the (maximum) size of neighbouring nodes.

Recall that in our particular application the adaptive modification of the nodes is done in order to effectively capture the evolution of the shock front and other local features of the flow. In all of our numerical experiments we found that the current number $N = |\Xi|$ of nodes in Ξ is about proportional to the length of the current shock front. Since the geometry of the moving shock front may be very complicated, it is very hard, if not impossible, to predict the required size $N \equiv N(t)$ of the node set $\Xi \equiv \Xi(t)$, $t \in I$, a priori.

Nevertheless, under the assumption that the number of nodes is bounded above by $N_{\max}$, i.e., $|\Xi(t)| \leq N_{\max}$ at any time $t \in I$, the entire simulation requires at most $\mathcal{O}(N_{\max} \cdot S \cdot n^3)$ operations in total, where S denotes the number of time steps. In the numerical example of Subsection 4.2, we found $N_{\max} = 6000$ (a posteriori), and we let $S = 2100$, and $n = 15$ (a priori).

4 Numerical Results and Comparisons

In order to illustrate the good performance of our adaptive meshfree advection scheme, **AMMoC**, we consider using one popular test case scenario from hydrocarbon reservoir modelling, termed the *five-spot problem*, where **AMMoC** is shown to be competitive with two leading commercial reservoir simulators, ECLIPSE and FrontSim of Schlumberger.

4.1 The Five-Spot Problem

The following variant of the five-spot problem in two dimensions, $d = 2$, may be summarized as follows. The computational domain $\Omega = [-0.5, 0.5]^2$ is corresponding to a bounded reservoir, where we normalize, for the sake of simplicity, the permeability $\mathbf{K}$ of the homogeneous porous medium, so that $\mathbf{K}(x) \equiv 1$.

Initially, the pores of the reservoir are saturated with non-wetting fluid (oil, $u \equiv 0$), before wetting fluid (water, $u \equiv 1$) is injected through one injection well, placed at the center $\mathbf{o} = (0, 0)$ of Ω. During the simulation, the non-wetting fluid (oil) is displaced by the wetting fluid (water) towards the four corner points

$$\mathcal{C} = \{(-0.5, -0.5), (-0.5, 0.5), (0.5, -0.5), (0.5, 0.5)\}$$

of the square domain Ω.

The five-spot problem requires solving the equations (8),(9),(10) on Ω, in combination with the initial condition

$$u_0(x) = \begin{cases} 1 & \text{for } \|x - \mathbf{o}\| \leq R, \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

where we let $R = 0.02$ for the radius of the injection well at the center $\mathbf{o} \in \Omega$.

Our aim, however, is to merely solve the Cauchy problem (8),(20) for the Buckley-Leverett equation. This is because we wish to evaluate the performance of our meshfree advection method **AMMoC** as an adaptive saturation solver on unstructured particle sets. Therefore, we decided to work with the following simplifications of the five-spot model problem.

Firstly, following along the lines of Albright [1], we assume unit mobility, $M \equiv 1$. Secondly, we work with a *stationary* pressure field, $p(x) \equiv p(\cdot, x)$, given by

$$p(x) = \sum_{\mathbf{c} \in \mathcal{C}} \log(\|x - \mathbf{c}\|) - \log(\|x - \mathbf{o}\|), \quad \text{for all } x \in \Omega, t \in I, \tag{21}$$

which yields the *stationary* velocity field

$$\mathbf{v} = -\nabla \cdot p, \tag{22}$$

due to Darcy's law (10), and with the assumption $M \equiv 1$. It is easy to see that the velocity field $\mathbf{v}$ is in this case divergence-free, i.e., $\mathbf{v}$ in (22) satisfies the incompressibility relation (9). Figure 3 shows the contour lines of the pressure field p together with the streamlines of the velocity field $\mathbf{v}$, resulting from Darcy's law (10).

Note that by these two simplifications, the elliptic equations (9),(10) uncouple from the Buckley-Leverett equation (8). This allows us to neglect the pressure equation (10), so that we restrict ourselves to solving the flow equation (8).

Let us make a few comments in order to support the two simplifications taken above. First note that the pressure field p in (21) has singularities at the corners and at the center of the domain Ω, so that the pressure in the vicinity of the injection well at $\mathbf{o}$ is arbitrarily high, whereas the pressure around the production wells, placed at the four corner points in $\mathcal{C}$, is arbitrarily small.

Therefore, the resulting particle flow exhibits high velocities near the five wells but small velocities between the wells. This complies with the actual physical behaviour corresponding to the five-spot model problem, where moreover the variation of pressure has rather small effects on the saturation.

Let us moreover remark that even in sophisticated, full reservoir simulators, such as in ECLIPSE and FrontSim, the pressure (and thus the velocity) are, unlike the saturation, updated rather infrequently. In other words, long time steps are made between updates of the pressure, whereas many smaller time steps update the saturation between the pressure updates.

That the above taken simplifications for the five-spot problem are quite reasonable is further supported by numerical comparisons (in Subsection 4.3) with two commercial reservoir simulators, ECLIPSE and FrontSim, each of which solves the coupled set of equations (8),(9),(10).

4.2 Meshfree Simulation by AMMoC

According to the discussion in Section 3, we apply our meshfree advection scheme **AMMoC** to the Cauchy problem (11),(20) for the viscous Buckley-Leverett equation, rather than for the hyperbolic flow equation (8). Recall that this is in order to model the propagation of the shock front, which is of primary importance in the relevant application. Therefore, the accurate approximation of the shock front requires particular care. This is in **AMMoC** mainly accomplished by the adaptive modification of the nodes during the simulation.

Now let us turn straight to our numerical results, provided by our meshfree advection scheme **AMMoC**. In our simulation, we decided to select a constant time step size $\tau = 5 \cdot 10^{-5}$, and the simulation comprises 2100 time steps, so that $I = [0, 2100\tau]$. Moreover, we let $\epsilon = 0.015$ for the diffusion

coefficient in (11), and we selected the value $\mu = 0.5$ for the viscosity ratio of water and oil, appearing in the fractional flow function (7).

Figure 4 shows the water saturation u during the simulation at six different times, $t = t_0$, $t = t_{420}$, $t = t_{840}$, $t = t_{1260}$, $t = t_{1680}$, and $t = t_{2100}$, where u is evaluated at a fixed cartesian mesh comprising 100×100 rectangular cells. The corresponding color code for the water saturation is shown at the right margin of Figure 4, respectively.

Note that the shock front, at the interface between the non-wetting fluid (oil, $u \equiv 0$) and the wetting fluid (water, $u \equiv 1$), is moving from the center towards the four corner points of the computational domain Ω. This way, the non-wetting fluid (oil) is effectively displaced by the wetting fluid (water) into the four production wells, as expected.

Just before the breakthrough, when the shock front arrives at the production wells, an increased velocity can be observed around the four production wells, see the transition between the time step $t = t_{1680}$, Figure 4 **(e)** and $t = t_{2100}$, Figure 4 **(f)**. This *sucking effect* is due to the singularities of the pressure field p at the corners in $\mathcal{C}$.

The distribution of the nodes, corresponding to the six different times, $t = t_0$, $t = t_{420}$, $t = t_{840}$, $t = t_{1260}$, $t = t_{1680}$, and $t = t_{2100}$, is shown in Figure 5. Due to the adaptive distribution of the nodes, the shock front propagation of the solution u is captured very well. This helps to reduce the required computational costs while maintaining the accuracy, due to a higher resolution around the shock front. The effective distribution of the nodes around the shock supports the utility of the adaption rules, proposed in our previous paper [7], yet once more.

We found that the number of nodes in Ξ is roughly proportional to the length of the shock front. This is confirmed by the graph in Figure 1, where the number of nodes is plotted as a function of time.

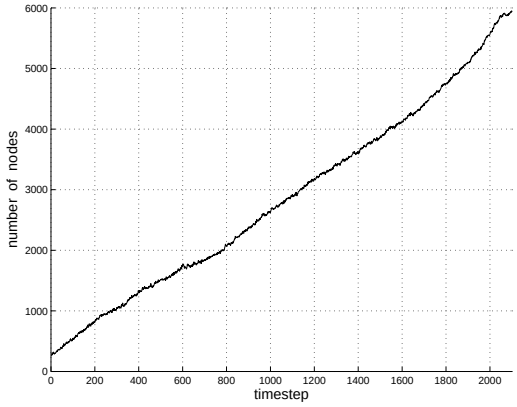

Figure 1: Five-spot problem. Number of nodes during the simulation by our meshfree advection scheme **AMMoC**.

4.3 Comparison with ECLIPSE and FrontSim

Let us compare the results of our simulation by **AMMoC** with two different commercial reservoir simulators, ECLIPSE and FrontSim. Both ECLIPSE and FrontSim are products of Schlumberger Oil Field Services. We remark that ECLIPSE and FrontSim are regarded as the industry standard in reservoir simulation.

ECLIPSE is reservoir simulation software, which works with *first order* finite differences. In contrast, the multi-phase simulator FrontSim is based on a *streamline method* [9], which solves the Buckley-Leverett equation along pre-calculated streamlines of the flow particles. Each of these two simulators solves the coupled system of equations (8),(9),(10). In particular, unlike in our model simplification, the pressure field p is updated during the simulation.

The latter requires, due to Darcy's law (10), the maintenance of the total velocity $\mathbf{v}$, which also appears in the flow equation (8). However, our simplifications taken in the previous subsection, are quite reasonable for the special case of the five-sport problem. In particular, the variation of the pressure field can be neglected. This is supported by the following numerical results, where our advection scheme **AMMoC** is compared with ECLIPSE and FrontSim.

Figure 6 shows the water saturation obtained from the simulator ECLIPSE at six different times, whereas the corresponding results obtained by the simulator FrontSim are displayed in Figure 7. The evolution of the saturation u, obtained by either of these two simulators, especially the location and the propagation of the shock front, is comparable with that obtained by our method **AMMoC**, whose results are shown in Figure 4.

For the purpose of further comparison, let us regard the water saturation u at time $t = t_{1260}$, for each of the three different simulation methods, **AMMoC**, ECLIPSE, and FrontSim, see Figures 4**(d)**, 6**(d)**, and 7**(d)**. Figure 2 shows the three different profiles of the saturation $u(t_{1260}, \cdot)$ across the half diagonal of Ω, drawn from the center $\mathbf{o} = (0, 0)$ to the corner point $(0.5, 0.5)$. For better orientation, the dotted line in Figure 2 shows the expected height of the shock front, which can be computed analytically by Welge's tangent method [22].

Note that the three different methods lead to similar saturation profiles. Moreover, each method captures the expected height of the shock front very well. When it comes to accurately resolving the shock front, the method FrontSim is the best, followed by our meshfree scheme **AMMoC** and lastly ECLIPSE. This is not very surprising insofar as FrontSim relies on *front tracking*, a technique which is well-known for its small numerical diffusion.

Since the method ECLIPSE is only of first order, ECLIPSE is inferior to both **AMMoC** and FrontSim, due to enhanced numerical diffusion around the shock front. Our meshfree advection scheme **AMMoC**, of second order,

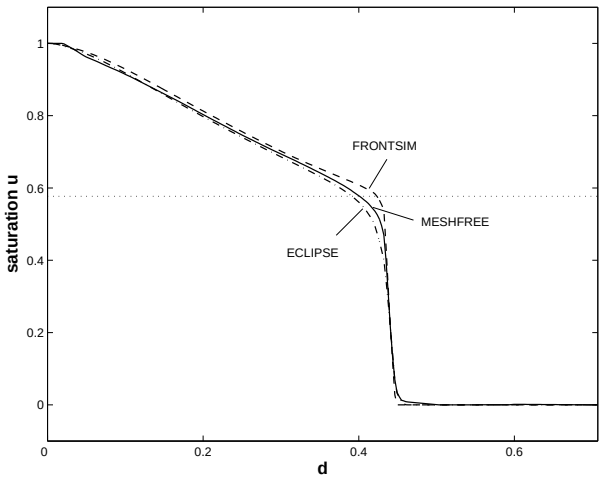

Figure 2: Five-spot problem. Comparison between ECLIPSE, FrontSim, and our meshfree advection scheme **AMMoC**. The saturation profiles are, at time $t = t_{1260}$, compared along the half diagonal from the center $\mathbf{o} = (0, 0)$ to the corner point $(0.5, 0.5)$. The values on the axis of abscissae correspond to the distance $\mathbf{d}$ from $\mathbf{o}$.

reduces (compared with ECLIPSE) the numerical diffusion, mainly due to the effective adaptive node distribution. Moreover, the saturation profile obtained by our meshfree method **AMMoC** is fairly close that of FrontSim, see Figure 2.

In conclusion, we feel that our meshfree method **AMMoC** is, as regards its performance concerning the five-spot problem, quite competitive with both ECLIPSE and FrontSim, since it produces only small amount of numerical diffusion and tracks the shock front very well.

Let us finally remark that neither ECLIPSE nor FrontSim is accessible to us. The presented results by ECLIPSE and FrontSim, each based on hardware optimized Fortran codes, were obtained at Schlumberger Stavanger Research. In contrast, the numerical simulation by **AMMoC** was performed by using MATLAB 6 Release 13 on inferior hardware, namely on a PC (model: IBM 236623G) with processor type Intel Pentium(R) 4 1600MHz. Therefore, we refrained from providing the required CPU times for the simulation by the three different methods, since any such comparison can only be unfair on our method **AMMoC**.

Acknowledgment

We wish to thank Brice Vallès from Schlumberger Stavanger Research for providing the reference solutions with ECLIPSE and FrontSim. The authors were partly supported by the European Union within the project NetAGES (Network for Automated Geometry Extraction from Seismic), contract no. IST-1999-29034.

References

[1] N. Albright, P. Concus, and W. Proskurowski (1979) Numerical solution of the multi-dimensional Buckley-Leverett equation by a sampling method. Paper SPE 7681, Soc. Petrol. Eng. Fifth Symp. on Reservoir Simulation, Denver, Feb. 1979.

[2] S.N. Atluri (2004) *The Meshless Method (MLPG) for Domain & BIE Discretizations*, Tech Science Press.

[3] S.N. Atluri, Z.D. Han, and A.M. Rajendran (2004) A new implementation of the meshless finite volume method, through the MLPG "mixed" approach. CMES: Computer Modeling in Engineering & Sciences, vol. 6, no. 6, 491–514.

[4] S.N. Atluri and S. Shen (2002) The meshless local Petrov-Galerkin (MLPG) method: a simple & less-costly alternative to the finite element and boundary element methods. CMES: Computer Modeling in Engineering & Sciences, vol. 3, no. 1, 11–52.

[5] K. Aziz and A. Settari (1979) *Petroleum Reservoir Simulation*. Applied Science, London.

[6] J. Behrens and A. Iske (2002) Grid-free adaptive semi-Lagrangian advection using radial basis functions. Comput. Math. Appl. **43**, 319–327.

[7] J. Behrens, A. Iske, and S. Pöhn (2001) Effective node adaption for grid-free semi-Lagrangian advection. *Discrete Modelling and Discrete Algorithms in Continuum Mechanics*, T. Sonar and I. Thomas (eds.), Logos, Berlin, 110–119.

[8] J. Behrens, A. Iske, and M. Käser (2002) Adaptive meshfree method of backward characteristics for nonlinear transport equations. *Meshfree Methods for Partial Differential Equations*, M. Griebel and M.A. Schweitzer (eds.), Springer-Verlag, Heidelberg, 21–36.

[9] F. Bratvedt, K. Bratvedt, C. Buchholz, H. Holden, L. Holden, and N.H. Risebro (1989) A new front-tracking method for reservoir simu-

lation. SPE 19805. In *Proceedings of the 64th SPE Annual Technical Conference and Exhibition*, San Antonio.

[10] J.M. Buckley and M.C. Leverett (1942) Mechanism of fluid displacement in sands. Trans. AIME **146**, 107–116.

[11] J.M. Burgers (1948) A mathematical model illustrating the theory of turbulence. Adv. Appl. Mech. **1**, 171–199.

[12] P. Deuflhard and F. Bornemann (2002) *Scientific Computing with Ordinary Differential Equations*. Springer, New York.

[13] B. Gustafsson, H.-O. Kreiss, and J. Oliger (1995) *Time Dependent Problems and Difference Methods*. John Wiley and Sons, New York.

[14] A. Iske (2003) On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines. *Modern Developments in Multivariate Approximation*, W. Haussmann, K. Jetter, M. Reimer, J. Stöckler (eds.), ISNM 145, Birkhäuser, Basel, 153–165.

[15] A. Iske (2002) Scattered data modelling using radial basis functions. *Tutorials on Multiresolution in Geometric Modelling*, A. Iske, E. Quak, and M.S. Floater (eds.), Springer-Verlag, Heidelberg, 205–242.

[16] A. Iske (2004) *Multiresolution Methods in Scattered Data Modelling*. Lecture Notes in Computational Science and Engineering, Vol. **37**, Springer-Verlag, Heidelberg, 2004.

[17] H. Lin and S.N. Atluri (2001): The meshless local Petrov-Galerkin (MLPG) method for solving incompressible Navier-Stokes equations. CMES: Computer Modeling in Engineering & Sciences, vol. 2, no. 2, 117–142.

[18] K.W. Morton (1996) *Numerical Solution of Convection-Diffusion Problems*. Chapman & Hall, London.

[19] D.W. Peaceman (1977) *Fundamentals of Numerical Reservoir Simulation*. Elsevier, Amsterdam.

[20] A.E. Scheidegger (1974) *The Physics of Flow in Porous Media*. University of Toronto Press, Toronto.

[21] J.A. Sethian, A.J. Chorin, and P. Concus (1983) Numerical solution of the Buckley-Leverett equations. Paper SPE 12254, Soc. Petrol. Eng. Symp. on Reservoir Simulation, San Francisco, Nov. 1983.

[22] H.J. Welge (1952) A simplified method for computing oil recovery by gas or water drive. *Trans. AIME* **195**, 97–108.

Authors' addresses:

Armin Iske (corresponding author)
University of Leicester
Department of Mathematics
Leicester LE1 7RH, UK
iske@mcs.le.ac.uk

Martin Käser
Civil & Environmental Engineering
University of Trento
I-38050 Trento, ITALY
martin.kaeser@ing.unitn.it

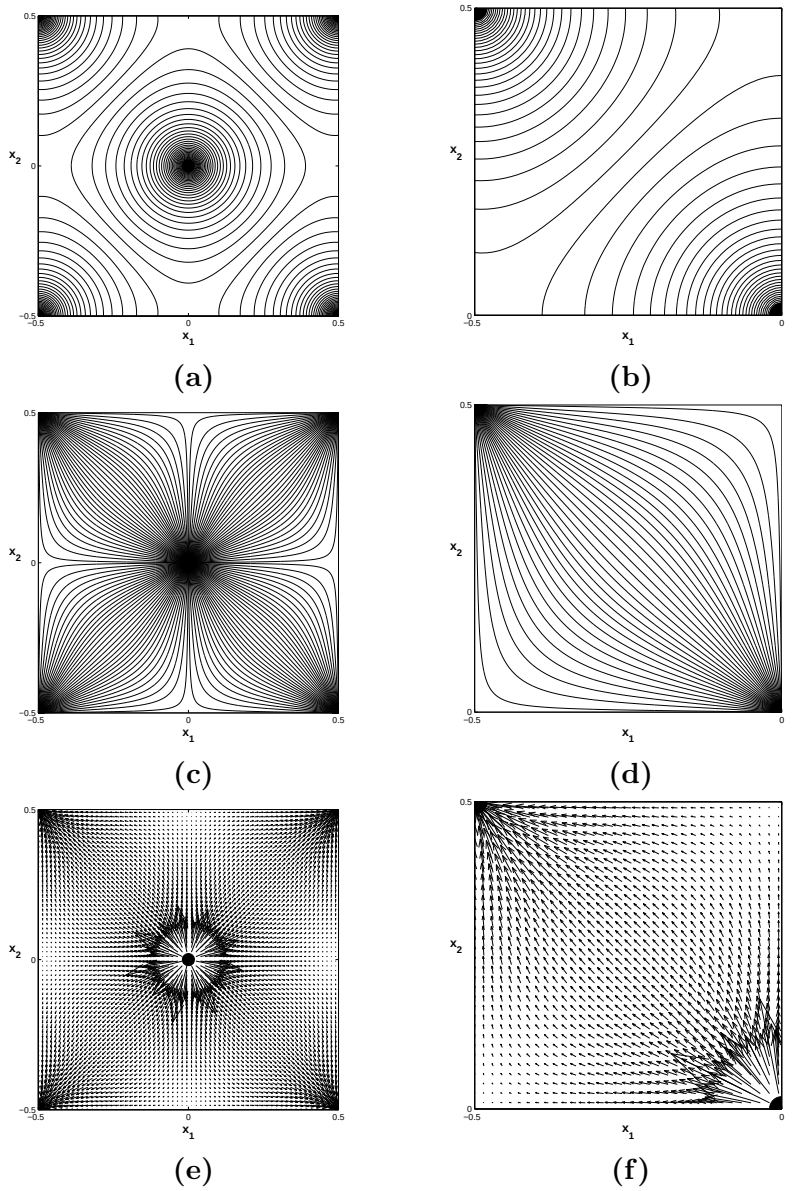

Figure 3: Five-spot problem. **(a)** Contours of the pressure field, **(c)** streamlines of the velocity field, and **(e)** velocity vectors in $\Omega = [-0.5, 0.5]^2$. The corresponding plots of these data in the top left quarter $[-0.5, 0] \times [0, 0.5]$ are shown in **(b)**, **(d)**, and **(f)**.

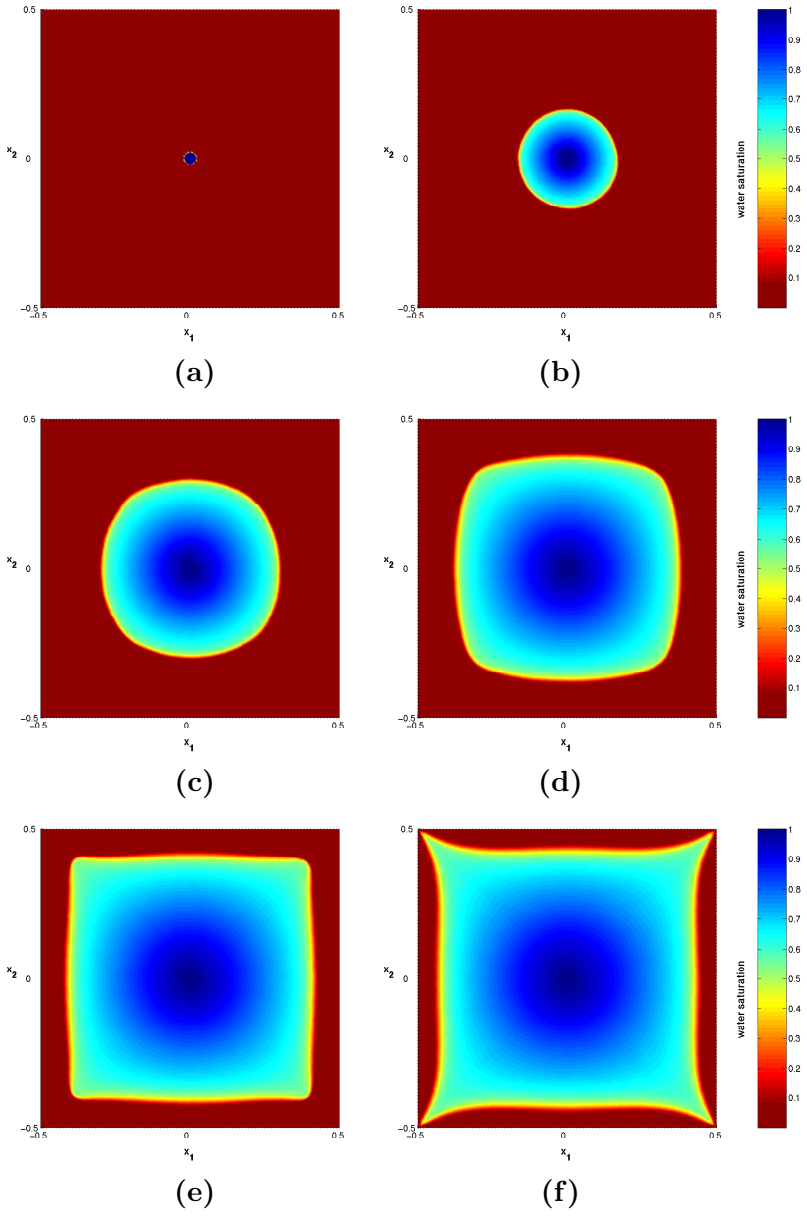

Figure 4: Five-spot problem. Solution obtained by **AMMoC**. The color plots indicate the water saturation u during the simulation at six different times, **(a)** $t = t_0$, **(b)** $t = t_{420}$, **(c)** $t = t_{840}$, **(d)** $t = t_{1260}$, **(e)** $t = t_{1680}$, and **(f)** $t = t_{2100}$.

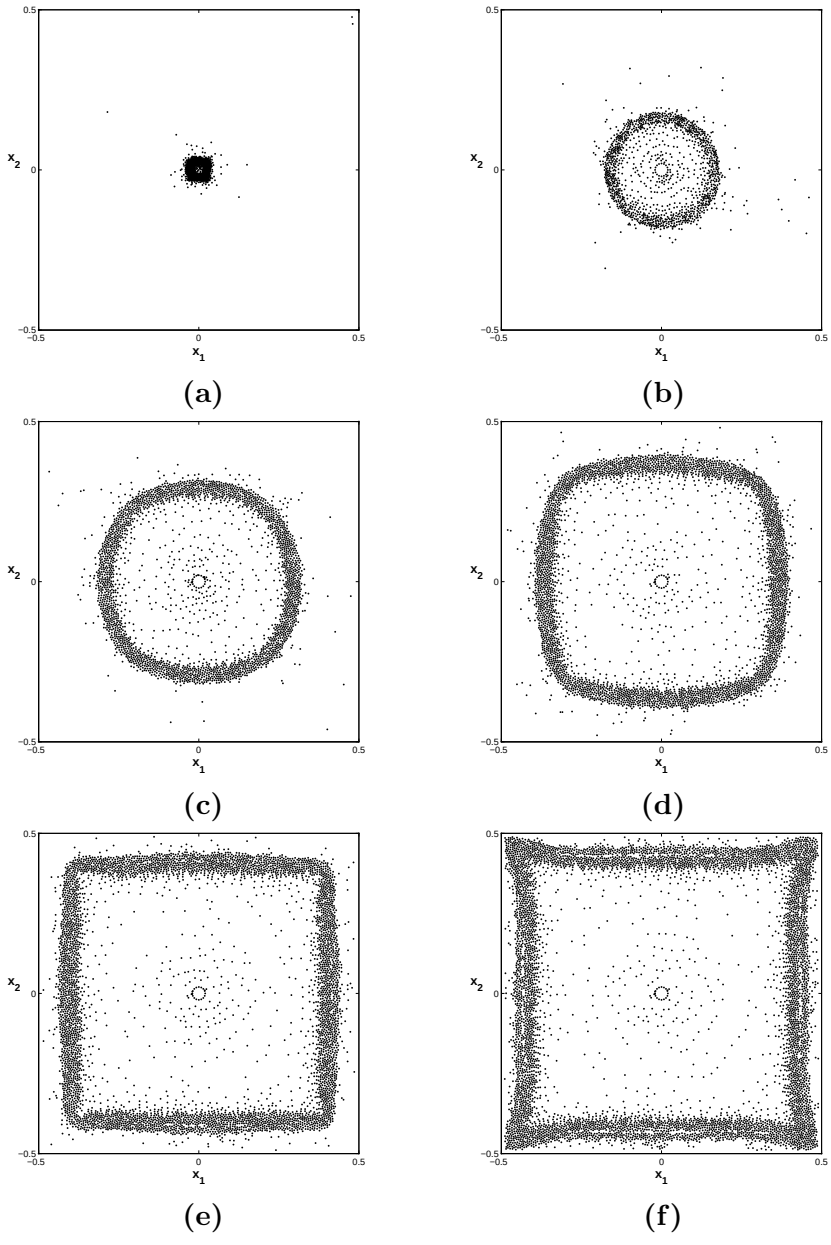

Figure 5: Five-spot problem. Adaptive node distribution during the simulation by **AMMoC** at six different times, **(a)** $t = t_0$, **(b)** $t = t_{420}$, **(c)** $t = t_{840}$, **(d)** $t = t_{1260}$, **(e)** $t = t_{1680}$, and **(f)** $t = t_{2100}$.

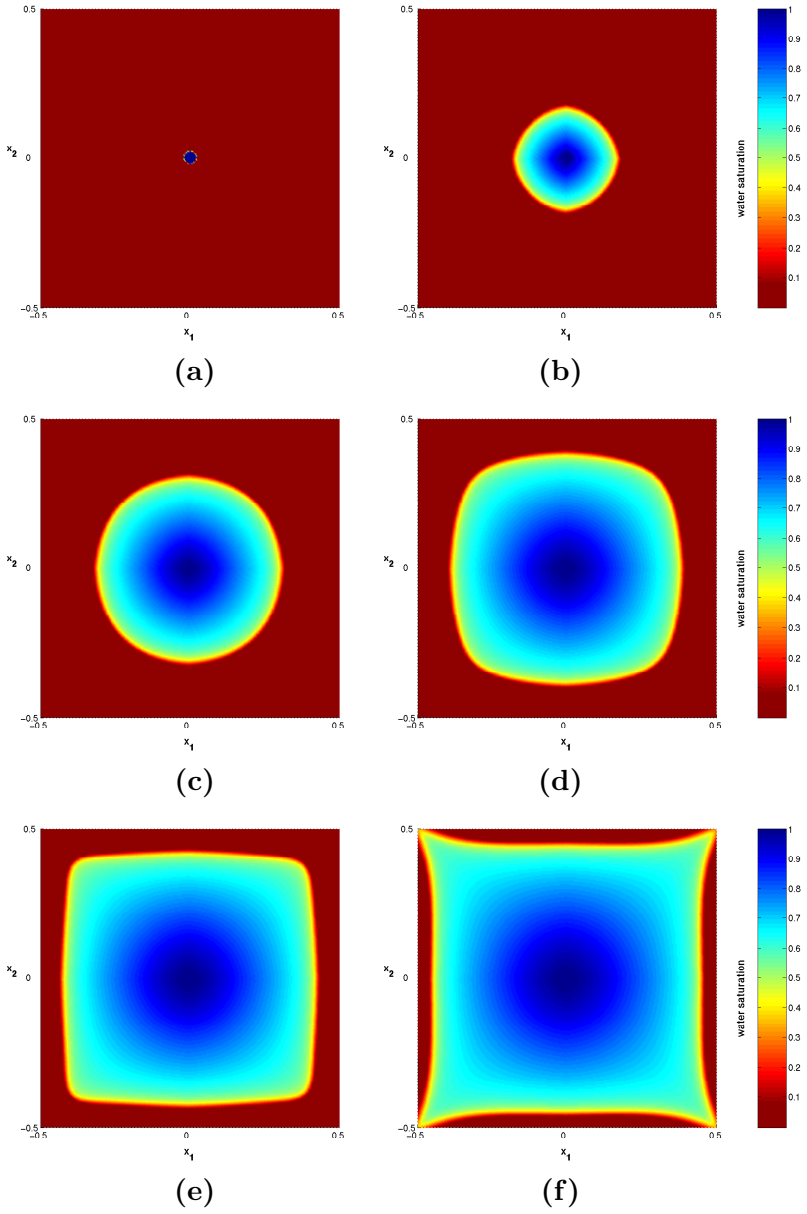

Figure 6: Five-spot problem. Solution obtained by ECLIPSE. The color plots indicate the water saturation u during the simulation at six different times, **(a)** $t = t_0$, **(b)** $t = t_{420}$, **(c)** $t = t_{840}$, **(d)** $t = t_{1260}$, **(e)** $t = t_{1680}$, and **(f)** $t = t_{2100}$.

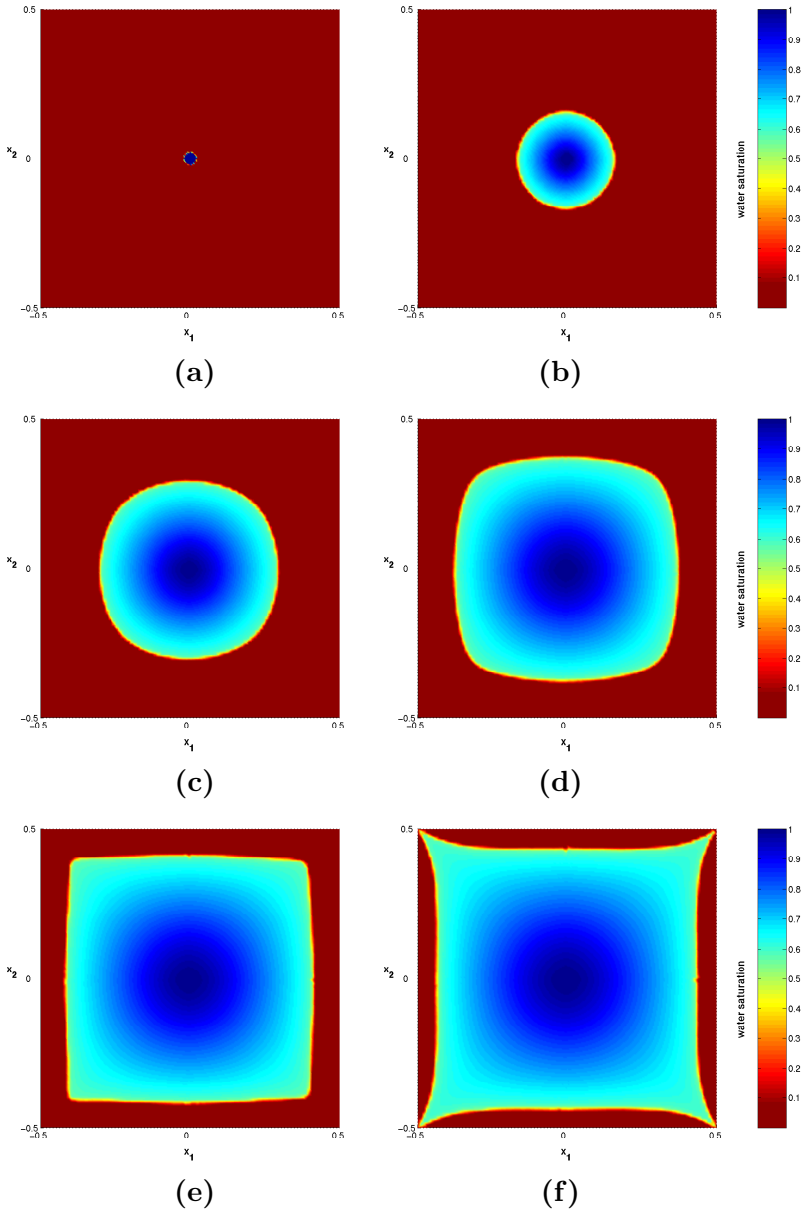

Figure 7: Five-spot problem. Solution obtained by FrontSim. The color plots indicate the water saturation u during the simulation at six different times, **(a)** $t = t_0$, **(b)** $t = t_{420}$, **(c)** $t = t_{840}$, **(d)** $t = t_{1260}$, **(e)** $t = t_{1680}$, and **(f)** $t = t_{2100}$.