A MsCV framework using a non-orthodox MPFA-D for the simulation of two-phase flows on truly unstructured grids.

[1]**Artur Castiel Reis de Souza, Eng.**

arturcastiel@gmail.com

[2]**Lorena Monteiro Cavalcanti Barbosa, PhD**

lorenamcavalcanti@gmail.com

[3]**Darlan Karlo Elisiário de Carvalho, PhD**

dkarlo@uol.com.br

[4]**Paulo Roberto Maciel Lyra, PhD**

prm@padmec.org

[1]Departamento de Engenharia Civil, UFPE, Av. da Arquitetura, S/N, CEP 50740-550 Recife/PE - Brazil. [2,3,4]Departamento de Engenharia Mecnica, UFPE, Av. da Arquitetura, S/N, CEP 50740-550 Recife/PE - Brazil

Abstract. *Modern geocellular models may contain up to hundreds million cells, while petroleum reservoir models handle at most a fraction of this quantity turning the direct numerical simulation of multiphase flow in heterogeneous and anisotropic media infeasible. To overcome this problems Multiscale Finite Volume Methods (MsFVM) uses operators to project the fine-scale system of equations onto a coarse-scale space originated from a lower-resolution grid. The resulting coarse system is solved and by using the multiscale operator projected back onto the higher-resolution grid. Nonetheless, the MsFVM operators fail to capture high-resolution geological properties on unstructured grids as they often rely on TPFA, which is only consistent for k-orthogonal grids. Furthermore, MsFVM possess no framework capable of generating the geometric entities needed to the simulation on unstructured coarse-scale meshes. The Multiscale Restricted Smoothed Basis (MsRSB) method creates this framework and expands the multiscale approach to unstructured coarse grids. However, it fails to produce consistent solutions on fine-scale unstructured grids and for arbitrary permeability tensors as it uses TPFA method. In this article, we couple a Multi-Point Flux Approximation with a Diamond stencil to the MsRSB to extend its use to general unstructured grids. We call our framework the "Multiscale Control Volume" (MsCV) method.*

Keywords: *Multiscale Reservoir Simulation, MsRSB, MsCV, MPFA-D, Unstructured Meshes*

1 INTRODUCTION

The advances in geostatistical modeling and characterization allows information from different scales to be integrated in order to generate geocellular models whose resolution typically range from 10^8 to 10^9 blocks, meanwhile the standard models of flow simulation in porous media can handle 10^6 to 10^7 blocks. In this way, multiple direct simulations on these high-resolution grids become infeasible. To overcome this limitation, scale-transferring methods have been devised. In essence, they allow high-resolution geostatistical data to be integrated onto the flow simulation grid. Among them, two branches of schemes stand out: the upscaling and the multiscale methods. The first generally employs a sort of homogenization (Farmer, 2002), even when there is no formal separation between the scales. In these schemes a solution is found at the coarse-scale space leading to fast and robust results, which often are unable to preserve details of the physical properties causing deterioration of the accuracy of the studied physical phenomena representation. On the other hand, the multiscale schemes develop a set of operators, which are capable of projecting the fine-scale discrete system onto the coarse-scale space, this low-resolution system is then solved and the multiscale operator projects the solution back to the high-resolution grid (Hou and Wu,1997; Jenny, Lee and Tchelepi, 2006; Zhou and Tchelepi,2008). This preserves the natural coupling between scales avoiding inconsistencies and loss of fine-scale information inherent of most upscaling methods. The main difference between the Multiescala methods concerns the strategy used to impose mass conservation (Kippe, Aarnes and Lie, 2008). While methods in the Multiscale Mixed Finite Element Method (MSMFEM) family(Hou and Wu,1997; Kippe, Aarnes and Lie, 2008) impose mathematically the conservation of flow by creating multiscale operators that by definition produce conservative velocity fields, schemes in the Multiscale Finite Volume (MsFV) family use an additional auxiliary dual mesh to reconstruct a conservative flux field. Although the use of another mesh increases the amount of data stored, it eliminates considerably the number of degrees of freedom by removing the need to calculate an operator that integrates information simultaneously of pressure and velocity fields. Nevertheless, most of the MsFVM suffer from a severe deficiency, as they are unable to work on unstructured grids. This is due to three different factors; the use of Two-Point Flux Approximation (TPFA), which is only consistent for k-orthogonal grids, the difficulties presented in generating geometrical entities, such as auxiliary grids and coarse centers, and calculating multiscale operators accordingly. In this context, a great effort has been made in the development of schemes capable of enabling the simulation on unstructured grids on both scales. Krogstad et al. (2009) developed a variant of MsMFEM capable of working with coarse meshes whose volumes are nearly degenerated and unstructured. Moyner and Lie (2015) presented the Multi-scale Restriction-Smoothed Basis (MsRSB) in which the multiscale prolongation operator is calculated interactively smoothing the indicator function of each coarse volume restricted to corresponding support region enabling simulation using coarse-scale unstructured meshes. Parramore et al. (2016) developed the Neumann-D, a new set of boundary conditions to calculate the prolongation operator and coupled with a Multi-Point Flux Approximation (MPFA) method. In this article, we modify the MsRSB replacing the TPFA by a non-orthodox MPFA method with a diamond stencil (MPFA-D) to create a framework that allows the simulation on fine and coarse scales

unstructured grids. Additionally, we have developed a geometrical coarsening algorithm that creates lower-resolution grids in which the coarse volumes are capable of adapting themselves according to the underlying geological features of the media. We call our new approach, that incorporates the MPFA-D and the slightly modified MsRSB formulations, the "Multiscale Control Volume" (MsCV) method.

2 Mathematical Modeling

In the following section, we present a brief introduction on the laws and assumptions used to derive the equations that governs oil and water flow inside the reservoir rock. First, we assume a water-oil isothermal incompressible flow described by Darcy's law throughout an also incompressible fully saturated media. Gravitation fields and capillarity effects are neglected. We use a formulation in which the physical phenomena is segregated into two PDEs that govern respectively pressure distribution and phase transportation. The first being expressed by:

$$\vec{\nabla} \cdot \vec{v} = Q \tag{1}$$

where total fluid injection/production rate Q is denoted by $Q = Q_w + Q_o$, which is the sum of the sink/source term of each phase and $\vec{v}$ is the total fluid velocity $\vec{v} = \vec{v}_w + \vec{v}_o$ in which $\vec{v}_i$ is the Darcy's phase velocity defined as:

$$\vec{v}_i = -\lambda_i \underline{\underline{K}} \vec{\nabla} p \quad \text{where i = w,o (water/oil)} \tag{2}$$

and where p stands for the pressure, $\underline{\underline{K}}$ is the absolute rock permeability tensor and λ_i the phase mobility. The latter expressed by:

$$\phi \frac{\partial S_w}{\partial t} = -\vec{\nabla} \cdot \vec{F}(S_w) + Q_w \tag{3}$$

where ϕ is the rock porosity, S_w the saturation of the water phase and $\vec{F}(S_w)$ is defined as:

$$\vec{F}(S_w) = f_w(S_w)\vec{v} \text{ with } f_w = \lambda_w/(\lambda_w + \lambda_o) \tag{4}$$

The assumption the media is fully saturated gives the closure equation that states:

$$S_i + S_w = 1 \tag{5}$$

2.1 Initial and Boundary Conditions

For the PDEs described by Eq. (1) and (3) represent a well-posed problem, initial and boundary conditions must be provided, which a are often a combination of the following conditions:

$$\begin{aligned}
&p(\vec{x}, t) = g_D \quad \text{on} \quad \partial\Omega_D \times [0, t] \\
&\vec{v} \cdot \vec{n} = g_N \quad \text{on} \quad \partial\Omega_N \times [0, t] \\
&S_w(\vec{x}, t) = \bar{S}_w \quad \text{on} \quad \partial\Omega_W \times [0, t] \\
&S_w(\vec{x}, 0) = \bar{S}_w^o \quad \text{on} \quad \Omega
\end{aligned} \tag{6}$$

where Ω is the physical domain, $\partial\Omega_D$ and $\partial\Omega_N$ represent respectively Dirichlet and Neumann boundaries, Ω_W injection and production wells, g_d prescribed pressure, g_n prescribed flux, $\bar{S}_w$ is the prescribed saturation in a injection well and $\bar{S}_w^o$ is the initial saturation distribution.

3 Numerical Formulation

In this article, we use a variant of the classic IMplicit Pressure Explicit Saturation (IMPES) which decouples Eq. (1) and (3) to compute sequentially the pressure field implicitly to construct the velocity field and then calculating the saturation field. This process starts from an initial saturation field and it is repeated at each time step until the final time of the simulation is met. In the variant used in this paper, the pressure solver and velocity field steps are calculated using the MsCV method for the pressure equation. It is worth noting that although there are methods in the MsFV family to solve transport equation (Zhou,2010), this work focus on multiscale schemes for pressure equation only. In the following subsections, we present the strategies used to discretize the pressure and saturation equation along with Multiscale Finite Volume technique employed on this work.

3.1 Implicit formulation for the pressure equation using the MPFA-D

This subsection is focused on briefly deriving and explaining the Multi Point Flux Approximation with a Diamond stencil (MPFA-D) proposed by Gao and Wu (2010) and originally adapted for two phase fluid flow in petroleum reservoirs by Contreras et al. (2016). This method is a non-orthodox member of the Finite Volume family which can be seen as a generalization of the classical TPFA. In this scheme, the flux on each edge of the mesh is expressed as a combination of the unknown of the control volumes (CVs) to the left and to the right sides of the edge along with the two nodes which comprise the referring edge. In order to obtain a truly cell-centered method, these nodes are then written as a linear combination of the unknowns of the surrounding CVs using a mass conservative interpolation. We present a straightforward lemma (Gao and Wu, 2010) used for deriving the MPFA-D flux expression.

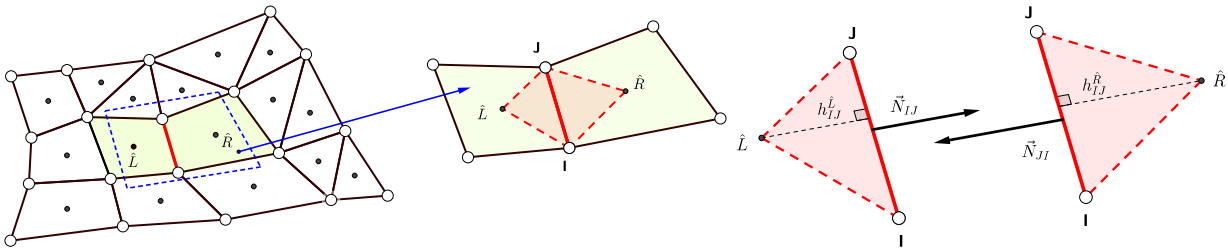
Lemma 1. *Let $\triangle OPQ$ be a triangle with vertices O, P, Q ordered counterclockwise. For a general pressure p defined on $\triangle OPQ$, we have:*

$$\vec{\nabla} p \simeq \frac{p_q - p_p}{|PQ|^2}\overrightarrow{PQ} + \frac{\mathcal{R}\overrightarrow{PQ}}{|PQ|^2}[(p_p - p_o)\cot\angle PQO + (p_q - p_o)\cot\angle OPQ] \tag{7}$$

where $\mathcal{R}$ is the algebraic operator which rotates a vector $90°$ clockwise.

Let IJ be a general edge inside the discretization of a physical domain Ω as shown in Fig. 1a. Regardless of the shape of any adjacent volumes inside this domain, two auxiliary triangular CVs can be formed by connecting the nodes I and J edge to the centroids of the volumes on the left $\hat{L}$ and on the right $\hat{R}$ of IJ. We assume the physical domain Ω and its boundary $\partial\Omega$ are partitioned by a finite number of CVs Ω_k. By integrating Eq. (1) over a $\Omega_{\hat{k}}$ and applying the Gauss divergence theorem, we obtain:

$$\int_{\partial\Omega_k} \vec{v}\cdot\vec{n}dA = \int_{\Omega_k} Q dV \tag{8}$$



(a) General edge of two adjacent volumes inside the discrete domain.

(b) Diamond stencil is generated by connecting center of left and right CV to IJ.

Figure 1: Illustration of the diamond stencil on a general grid.

where $\vec{n}$ is the unitary normal vector to the surface $\partial\Omega_k$ of the control volume Ω_k. Henceforth, we define $|\vec{N}_{IJ}| = |IJ|$.

The left and right hand sides of Eq. 8 can be written as:

$$\int_{\partial\Omega_k} \vec{v}\cdot\vec{n}dA = \sum_{IJ\in\partial\Omega_k} \vec{v}_{IJ}\cdot\vec{N}_{IJ} \quad \text{and} \quad \int_{\Omega_k} QdV = \bar{Q}_k V_k \tag{9}$$

In which V_k is the volume (or area in 2d) of Ω_k and $\vec{v}_{IJ}$ and $\bar{Q}_k$ are respectively the average Darcy's velocity across the edge IJ and the average source/sink term obtained using the mean value theorem. Finally, we rewrite Eq. (1) in its discrete form:

$$\sum_{IJ\in\partial\Omega_k} \vec{v}_{IJ}\cdot\vec{N}_{IJ} = \bar{Q}_k V_k \tag{10}$$

To obtain a fully discrete system of equations, we need consistent approximation for Darcy's average velocity on the edge IJ. In order to do so, we use the Lemma 1 to obtain a discrete form of the gradient pressure in Eq. (2) and derive a flux approximation based both triangles of Fig. 1b, ie:

$$\frac{h_{IJ}^{\hat{L}}}{\lambda_{IJ}K_{IJ(\hat{L})}^{n}}\vec{v}_{IJ}^{\hat{L}}\cdot\vec{N}_{IJ} \simeq -\frac{1}{|IJ|}\left((p_I-p_{\hat{L}})\frac{\overrightarrow{J\hat{L}}\cdot\overrightarrow{JI}}{|IJ|}+(p_J-p_{\hat{L}})\frac{\overrightarrow{I\hat{L}}\cdot\overrightarrow{IJ}}{|IJ|}\right)-(p_I-p_J)h_{IJ}^{\hat{L}}\frac{K_{IJ(L)}^{t}}{K_{IJ(L)}^{n}} \tag{11}$$

$$\frac{h_{IJ}^{\hat{R}}}{\lambda_{IJ}K_{IJ(\hat{R})}^{n}}\vec{v}_{IJ}^{\hat{R}}\cdot\vec{N}_{IJ} \simeq -\frac{1}{|IJ|}\left((p_J-p_{\hat{R}})\frac{\overrightarrow{J\hat{R}}\cdot\overrightarrow{JI}}{|IJ|}+(p_I-p_{\hat{R}})\frac{\overrightarrow{I\hat{R}}\cdot\overrightarrow{IJ}}{|IJ|}\right)-(p_I-p_J)h_{IJ}^{\hat{R}}\frac{K_{IJ(R)}^{t}}{K_{IJ(R)}^{n}} \tag{12}$$

For a consistent finite volume approximation, the flux must be continuous and unique across IJ, this means:

$$\vec{v}_{IJ}^{\hat{L}}\cdot\vec{N}_{IJ}+\vec{v}_{IJ}^{\hat{R}}\cdot\vec{N}_{JI}=0 \tag{13}$$

Finally, we can use (13) in (12), take the average of the resulting expression and equation (11) to derive an expression for a unique and continuous flux on IJ:

$$\vec{v}_{IJ}\cdot\vec{N}_{IJ} \simeq \tau_{IJ}[p_{\hat{R}}-p_{\hat{L}}-\upsilon_{IJ}(p_J-p_I)] \tag{14}$$

where the scalar transmissibility τ_{IJ}, and the non-dimensional tangential parameter v_{IJ} are defined as:

$$\tau_{IJ} = -\lambda_{IJ}|IJ|\frac{K^n_{IJ(\hat{L})}K^n_{IJ(\hat{R})}}{K^n_{IJ(\hat{L})}h^{\hat{R}}_{IJ} + K^n_{IJ(\hat{R})}h^{\hat{L}}_{IJ}} \tag{15}$$

$$v_{IJ} = \frac{\overrightarrow{IJ}\cdot\overrightarrow{\hat{L}\hat{R}}}{|IJ|^2} - \frac{1}{|IJ|}\left(\frac{K^t_{IJ(\hat{L})}}{K^n_{IJ(\hat{L})}}h^{\hat{L}}_{IJ} + \frac{K^t_{IJ(\hat{R})}}{K^n_{IJ(\hat{R})}}h^{\hat{R}}_{IJ}\right) \tag{16}$$

This ensures that the pressure gradient is piecewise constant and the pressure field is piecewise linear over the two triangles that form the diamond stencil. Moreover, we have obtained a consistent FV flux approximation, which is a generalization of the classic TPFA. Note that the tangential term v_{IJ} disappears if $\underset{\sim}{K}$ is isotropic or the mesh is k-orthogonal.

3.2 Boundary Condition Treatment

A flux expression for edges subjected to Dirichlet Boundary Conditions can be derived by using $g_D(I)$ and $g_D(J)$, prescribed pressures, in Eq. (14). It follows:

$$\vec{v}_{IJ}\cdot\vec{N}_{IJ} \simeq -\frac{\lambda_{IJ}K^n_{IJ}}{h^{\hat{L}}_{IJ}|IJ|}\left[(\overrightarrow{J\hat{L}}\cdot\overrightarrow{JI})g_D(I) + \overrightarrow{I\hat{L}}\cdot\overrightarrow{IJ})g_D(J) - p_{\hat{L}}|IL|^2\right] - K^t_{IJ}\left(g_D(J) - g_D(I)\right) \tag{17}$$

Similarly, for edges subjected to Neumann boundary conditions is described as:

$$\vec{v}_{IJ}\cdot\vec{N}_{IJ} = g_D|IJ| \tag{18}$$

where g_D stands for a prescribed velocity normal to edge IJ.

3.3 Mobility Approximation

In this work we use the approximation of Friis and Evje (2012) that proposes the total mobility projected on an edge IJ can be obtained by the average of the approximation of the mobility on each node of this edges (J and I).

$$\lambda_{IJ} = \frac{\lambda_I + \lambda_J}{2} \tag{19}$$

where the nodal mobilities λ_I and λ_J are approximated by the arithmetic mean of the mobilities surrounding a given a node:

$$\lambda_i = \frac{\sum_{\hat{k}=1}^{n_i}\lambda_{\hat{k}}V_{\hat{k}}}{\sum_{\hat{k}=1}^{n_i}V_{\hat{k}}} \tag{20}$$

where $i = I, J$ and n_i is number of control volumes around node i.

3.4 Linear Preserving Weight Interpolation

As we previously mentioned, in order to derive a fully discrete cell-centered system based on the pressure equation discretization, we must find an linear-preserving interpolation to write the pressures on nodes I and J as a linear combination of the pressure on the surrounding CVs, ie:

$$p_I = \sum_{k=1}^{n(I)} w_k p_k \tag{21}$$

where $n(I)$ is the number of volumes around I and w_k is the weights attributed to pressure p_k. Gao and Wu (2010) proposed an interpolation that is robust for anisotropic and heterogeneous media and meets the linear-preserving criteria. This interpolation is derived by imposing flux continuity on all $I\bar{k}$ edges of the auxiliary control volume generated by connecting the middle edges around a node I (See Fig. 2a). The weights are then defined as :

$$w_k = \frac{\psi_k}{\sum_{k=1}^{n(I)} \psi_k} \tag{22}$$

where

$$\psi = \overline{K}^n_{\hat{k},1} \eta^1_k \xi_k + \overline{K}^n_{\hat{k},2} \eta^2_k \xi_{k+1} \tag{23}$$

$$\eta^1_k = \frac{|I\bar{k}|}{h^k_k} \tag{24}$$

$$\eta^2_k = \frac{|I\overline{k+1}|}{h^k_{k+1}} \tag{25}$$

in which:

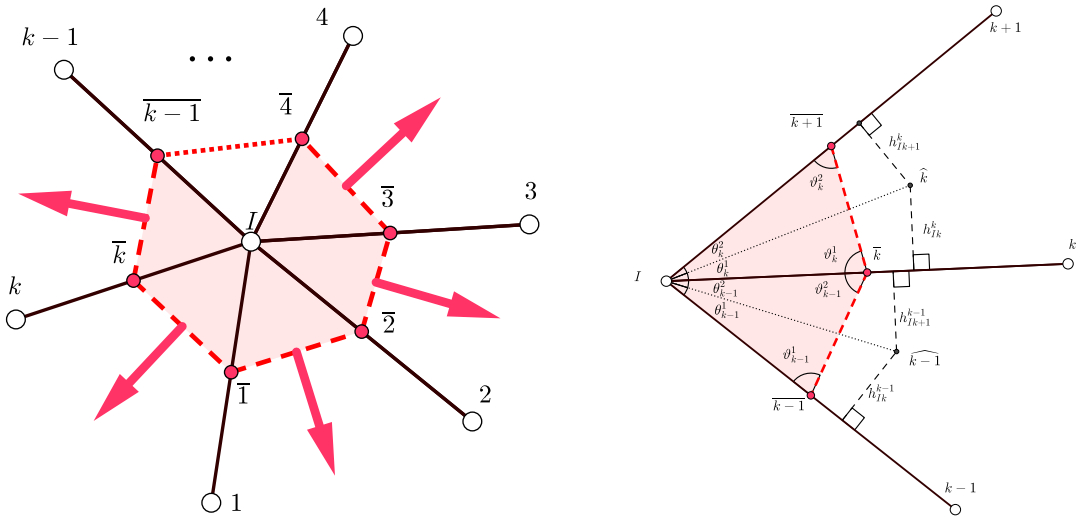
$$\xi_k = \frac{\overline{K}^t_{\hat{k-1}} - \overline{K}^t_{\hat{k}} + \overline{K}^n_{\hat{k-1}} \cot \vartheta^1_{\hat{k-1}} + \overline{K}^n_{\hat{k}} \cot \vartheta^2_{\hat{k}}}{K^{t,1}_k - K^{t,2}_{k-1} + K^{n,2}_{k-1} \cot \theta^2_{k-1} K^{n,1}_k \cot \theta^1_k} \tag{26}$$

Nodes laying on the boundary of the computational domain must be treated accordingly. For nodes subjected to Dirichlet BC one can simply set the values of node I and J to the value of the prescribed pressure. In cases of nodes lying in a Neumamm BC, weights must be calculated taking into account incoming and outcoming fluxes giving the following results:

$$\xi_k = \begin{cases} \frac{\bar{K}^n_1 \cot \vartheta^2_1 - \bar{K}^t_1}{K^n_1 \cot \vartheta^1_1 - K^t_1}, & k = 1; \\ \text{Use equation (26)}, & 2 \leqslant k \leqslant n(I); \\ \frac{\bar{K}^n_{n(I)} \cot \vartheta^2_{n(I)} + \bar{K}^t_{n(I)}}{K^n_{n(I)} \cot \vartheta^1_{n(I)} - K^t_{n(I)}}, & k = n(I)+1; \end{cases} \tag{27}$$

where k is counted counterclockwise from the edge 1 until the last edge around this node $n(I)+1$.

Check (Gao and Wu, 2010; Contreras et al., 2016) for a detailed derivation of the LPEW weights.



(a) Auxiliary Control Volume around node I created by connecting middle edges.

(b) Sketch of the geometrical entities used in the LPEW-2 deriving process.

Figure 2: Interaction region for the LPEW adapted from Gao and Wu (2010).

3.5 Explicit formulation of the Saturation Equation

To derive a discrete form of the saturation equation, we integrate Eq. (3) in the time interval t_o and t and in a CV Ω_k obtaining:

$$\int_{t_o}^{t}\int_{\Omega_k}\frac{\partial S_w}{\partial t}dVdt = -\int_{t_o}^{t}\int_{\Omega_k}\frac{1}{\phi}\vec{\nabla}\cdot\vec{F}(S_w)dVdt + \int_{t_o}^{t}\int_{\Omega_k}\frac{1}{\phi}Q_w dVdt \tag{28}$$

After applying Gauss Theorem in the first integral on the RHS equation (28) is rewritten as:

$$\int_{t_o}^{t}\int_{\Omega_{\hat{k}}}\frac{\partial S_w}{\partial t}dVdt = -\int_{t_o}^{t}\frac{1}{\phi_{\hat{k}}}\int_{\partial\Omega_{\hat{k}}}\vec{F}(S_w)\vec{n}dsdt + \int_{t_o}^{t}\int_{\Omega_{\hat{k}}}\frac{1}{\phi_{\hat{k}}}Q_w dVdt \tag{29}$$

By using the mean value theorem, we approximate the terms:

$$\frac{1}{\phi_{\hat{k}}}\int_{\partial\Omega_{\hat{k}}}\vec{F}(S_w)\vec{n}dsdt \simeq \frac{1}{\phi_{\hat{k}}}\sum_{IJ\in\Omega_{\hat{k}}}\vec{F}(S_w)_{IJ}\cdot\vec{N}_{IJ} \tag{30}$$

$$\frac{1}{\phi_{\hat{k}}}\int_{\Omega_{\hat{k}}}Q_w dV \simeq \frac{1}{\phi_{\hat{k}}}\overline{Q}_w \tag{31}$$

$$\int_{\Omega_{\hat{k}}}\frac{\partial S_w}{\partial t}dV \simeq \frac{\partial\overline{S_w}}{\partial t}V_{\hat{k}} \tag{32}$$

And using a Forward Euler approximation to integrate the LHS of Eq. (28), we have:

$$\int_{t_o}^{t}\frac{\partial S_w}{\partial t}dt = S_{w,\hat{k}}^{n+1} - S_{w,\hat{k}}^{n} \tag{33}$$

where $S_{w,\hat{k}}^{n+1}$ and $S_{w,\hat{k}}^{n}$ are, respectively, the water saturation on $\hat{k}$ at the time level n and $n+1$. By using Eq. (30) to (33) in (28) and manipulating we obtain the fully discrete saturation equation approximation:

$$S_{w,\hat{k}}^{n+1} = S_{w,\hat{k}}^{n} - \frac{\Delta t}{\phi_{\hat{k}} V_{\hat{k}}} \sum_{IJ \in \Omega_{\hat{k}}} \vec{F}(S_w)_{IJ} \cdot \vec{N}_{IJ} + \frac{\Delta t \overline{Q}_w}{\phi_{\hat{k}} V_{\hat{k}}} \tag{34}$$

where the hyperbolic term is $\vec{F}(S_w)_{IJ} = f_w\left(S_{w,\hat{k}}^{n}\right)\vec{v}_{IJ}$ and the time step is $\Delta t = t^{n+1} - t^{n}$ and in which $f_w(S_w) = \lambda_w/\lambda$. In addition, the FOU approximation states that flux across a surface IJ is approximated according to:

$$f_w(\vec{v}_{IJ} \cdot \vec{N}_{IJ}) = \begin{cases} (f_w)^{\text{left}} \, (\vec{v}_{IJ} \cdot \vec{N}_{IJ}) & \text{if} \quad \vec{v}_{IJ} \cdot \vec{N}_{IJ} < 0 \\ (f_w)^{\text{right}} \, (\vec{v}_{IJ} \cdot \vec{N}_{IJ}) & \text{if} \quad \vec{v}_{IJ} \cdot \vec{N}_{IJ} > 0 \end{cases} \tag{35}$$

where $(f_w)^{\text{left}}$ and $(f_w)^{\text{right}}$ are respectively the water fractional flux of the element to the left and to the right of the surface IJ.

Notice that we have derived an explicit approximation which only relies on current known information to calculate the saturation at next time step. Therefore, to ensure stability, Δt must meet Courant-Friedrichs-Lewy condition (CFL):

$$max_{IJ \in \Omega_{\hat{k}}} \left[\left(\frac{\Delta f_w(S_w)}{\Delta S_w} \right)_{IJ} \vec{v}_{IJ} \cdot \vec{N}_{IJ} \right] \frac{\Delta t}{V_{\hat{k}}} \leqslant \sigma \tag{36}$$

where the CFL condition is σ, and $\left(\frac{\Delta f_w(S_w)}{\Delta S_w}\right)_{IJ}$ is a discrete approximation of $(\frac{\partial f_w(S_w)}{\partial S_w})_{IJ}$ (Souza,2015).

4 Multiscale Formulation

The very definition of Multiscale Methods arises from the need to exchange information in different grids. As a direct simulation on a higher-resolution grid (fine-scale mesh, Ω_f or Ω) is not feasible, we project the high-resolution system of equation onto lower-resolution space (primal coarse-scale mesh, Ω_c) using the multiscale operators. This low-resolution coarse system is solved and the solution is projected back onto the higher-resolution grid using the prolongation operator. Thus, the lower resolution system works as an auxiliary basis with a considerable fewer degrees of freedom created to enable the simulation, meanwhile the multiscale operators capture the coupling between these scales. Zhou and Tchelepi (2008) developed an algebraic notation based on the work of Jenny, Lee and Tchelepi (2006) that describes these scale-transferring operators in a matrix form as a Restriction and Prolongation operator respectively defined as:

$$P_c = \underline{R}_{op} P_f \tag{37}$$

$$P_f = \underline{P}_{op} P_c \tag{38}$$

where P_f is a $n_f \times 1$ fine-scale pressure solution, P_c is a $n_c \times 1$ coarse-scale pressure solution, $\underline{P}_{op}$ $n_f \times n_c$ matrix and $\underline{R}_{op}$ must be a $n_c \times n_f$ matrix. Notice that, as the main idea of multiscale methods is to exchange computational cost by time in order to approximate a solution on the fine-scale space, Eq. (38) is an approximate result.

Coarse System

Let Eq. (39) be a fine-scale discrete system assembled using a FV flux approximation:

$$\underset{\sim}{T}_f P_f = Q_f \tag{39}$$

To derive a coarse scale system, we insert the definition of the Prolongation Operator presented on Eq. (38) in (39) and we apply it on both sides the Restriction Operator which results in the following equation:

$$\underset{\sim}{R}_{op} \underset{\sim}{T}_f \underset{\sim}{P}_{op} P_c = \underset{\sim}{R}_{op} Q_f \tag{40}$$

which is a $n_c \times n_c$ coarse-scale system of equations:

$$\underset{\sim}{T}_c P_c = Q_c \tag{41}$$

where

$$\underset{\sim}{T}_c = \underset{\sim}{R}_{op} \underset{\sim}{T}_f \underset{\sim}{P}_{op} \quad \text{and} \quad Q_c = \underset{\sim}{R}_{op} Q_f \tag{42}$$

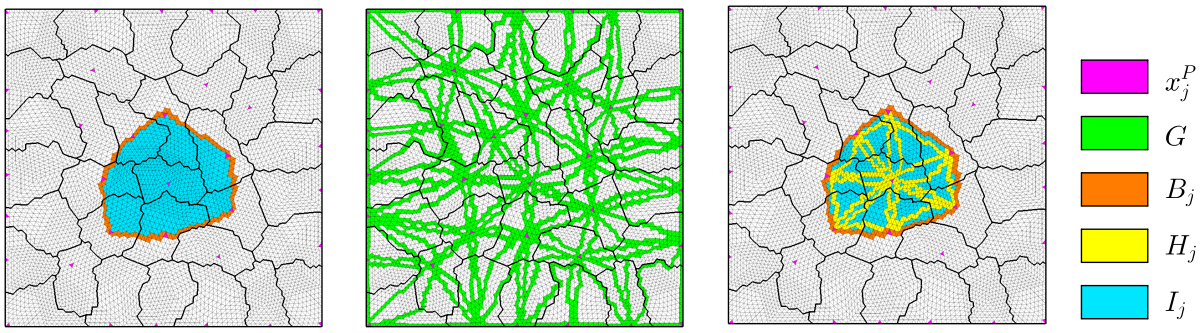
4.1 Multiscale Restriction Smoothed-Basis Method - MsRSB

The major advantage of the MsRSB over other multiscale methods in the MsFV family is the possibility to work with coarse-scale unstructured meshes. This flexibility allows constructing primal coarse-scale grids that adapt to media with faults, channels and pinched cells. Moyner and Lie (2015) has developed a framework that inherits and extends basic concepts of the MsFV method such as fine scale mesh, primal coarse mesh, primal coarse center and the support of a coarse center remain identical, however the preprocessing algorithms used to obtain them are different. Another advantage the MsRSB possess which is especially appropriate to IMPES like simulations is that prolongation operator is iteratively computed using a modified version of the Weighted Jacobi method. This means that no more than a few iterations are needed to recalculate the prolongation operator found on the previous time step decreasing significantly the coast associated with updating the basis functions. Nevertheless, the MsRSB, as most methods in the MsFV family, employs TPFA which is only consistent on k-orthogonal fine-scale grids. This severely limits its application on complex geometries restricting its use to mostly structured fine-scale grids. This article deals with this limitation by replacing the TPFA by MPFA-D method. This gives birth to a framework that works for unstructured grids on both scales.

MsRSB geometric entities

Hereby, we present the basic geometric concepts used by the MsRSB method.

Fine Scale Mesh (Ω^f or Ω): This is the higher-resolution grid derived from the discretization of the physical domain. Traditional Multiscale Methods generally employ some kind of k-orthogonal grid discretization for this mesh. These are the light gray volumes in Fig. 3.



(a) Support region (I_j) and (B_j) support boundary. **(b) Global Boundary, $G = B_1 \cup B_2 \cup B_3 \cup ... \cup B_k$.** **(c) Global boundary in Ω_j^P $H_j = I_j \cap G$.**

Figure 3: Basic concepts of a general coarse volume Ω_j^P MsRSB

Primal Coarse Mesh (Ω_c^p or Ω^p): This is a lower-resolution grid generated by clustering fine-scale volumes. In most traditional multiscale methods, it inherits the k-orthogonality of the underlying discretization giving birth to simple rectangular shapes. They are represented by the solid dark lines in 3.

Primal Coarse Center (x^P): This the fine-scale volume which will represent the center of a coarse volume. The classical approach is to use the fine volume closest to the coarse cell centroid, however more recent work (Barbosa,2017) proposes shifting the primal coarse center of coarse volumes laying on the boundary of the domain. In the next section we describe a different algorithm (Moyner and Lie,2015) to be used for general unstructured coarse meshes.

Support Region of a Primal Coarse Volume j (I_j): The definition of the support region of a primal coarse volume serves the same purpose of to dual coarse volumes of the MsFVM which is to calculate the prolongation operator and to reimpose mass conservation on the boundaries of the coarse volumes. It consists in all fine-scale volumes where:

$$(\underline{\underline{P}}_{op})_{i,j} > 0 \;\; \forall i \in I_j, \text{ otherwise } (\underline{\underline{P}}_{op})_{i,j} = 0 \tag{43}$$

I_j defines a influence zone of the coarse cell center x_j^P. The MsRSB uses the primal coarse cells and the coarse center to create the support region of a primal coarse volume in order to define all other geometric entities. Figure 3a illustrates the support region of the center coarse cell as the blue volumes around it. Note that the pink coarse, center of the middle coarse volume, is not included on the support region.

Support Boundary Region, (B_j): The support of a boundary region consists in all volumes that share at least one edge with I_j but are not contained in it. Figure 3a and 3c depict these as orange volumes. It is worth noting that no primal coarse center x_j^P are included in this group.

Global Support Boundary, (G): The Global Support Boundary Region is the union of all support boundaries associated with all coarse volumes of the primal coarse mesh. Figure 3b illustrates as the green fine-scale volumes.

Global Support Boundary in a Support Region, H_j: The region consists in the intersection of a general I_j with G. They are represented as the yellow volumes on Fig. 3c.

MsRSB Multiscale Operators

The MsRSB inherits the Restriction operator from the traditional MsFV which proposes using the Volume Summation (Chen and Hou, 2002; Jenny, Lee and Tchelepi, 2006; Zhou, 2010; Moyner and Lie, 2015).This operator literately works as a switch identifying which fine cells Ω_i^f influence the coarse volume Ω_j^c. As a consequence, j^{th} row of the operator is a $1 \times n_f$ boolean vector with 1 for all fine cells that belong to Ω_j^c and 0 otherwise. It can be defined as:

$$(\underset{\sim}{R}_{op})_{ij} = \begin{cases} 1 & \Omega_i^f \in \Omega_j^c; \\ 0, & \text{otherwise} \end{cases} \quad \text{where } 1 \leqslant j \leqslant n_c \text{ and } 1 \leqslant i \leqslant n_f \tag{44}$$

As we have previously mentioned, the MsRSB prolongation operator is solved iteratively inside each support region assuming 1 on x^P and outside it. As iterative methods require an initial solution, we choose as an initial guess the indicator function of each corresponding coarse volume, i.e.:

$$(\underset{\sim}{P}_{op}^0)_{ij} = \underset{\sim}{R}_{op}^T \tag{45}$$

On the other hand, the prolongation operator computed by solving the homogeneous elliptic part of Eq. (1) using a set of boundary conditions devised to decouple the domain generating multiple smaller problems. As a result, each column of this operator stores a normalized pressure distribution that varies from 1 on the coarse cell center to 0 outside the support of the corresponding cell. Herein, we define the Prolongation Operator using the following procedure:

1. First we calculate the classical Weighted Jacobi increment.

$$\widehat{d_j} = -\omega D^{-1} \underset{\sim}{T}_f^{pre} (\underset{\sim}{P}_{op})_j^{\eta} \tag{46}$$

where ω is the damping parameter of the Weighted Jacobi method, which is set to 2/3 in this paper, D is the diagonal matrix containing the diagonal of the preconditioned transmissibility matrix $\underset{\sim}{T}_f^{pre}$ which is defined by:

$$(\underset{\sim}{T}_f^{pre})_{ij} = \begin{cases} (\underset{\sim}{T}_f)_{ij} & i \neq j; \\ (\underset{\sim}{T}_f)_{ij} - \sum_{k=1}^{n} (\underset{\sim}{T}_f)_{ik} & i = j; \end{cases} \tag{47}$$

2. The following piecewise function modifies the increment ensuring that partition of unity is not lost and restricting growth to the inner side of the support region.

$$d_{ij} = \begin{cases} \frac{\widehat{d_{ij}} - (\underset{\sim}{P}_{op})_{ij}^{\eta} \sum_{k \in H_k} \widehat{d_{ik}}}{1 + \sum_{k \in H_k} \widehat{d_{ik}}} & \Omega_j^f \in H_j; \\ \widehat{d_{ij}} & \Omega_j^f \in I_j \text{ and } \Omega_j^f \notin H_j; \\ 0 & \Omega_j^f \notin I_j; \end{cases} \tag{48}$$

The last sub-function in Eq. (48) zeroes increments of fine-scale volumes as they get to the boundaries of the support region. To avoid losing partition of unity this first sub-function redistributes this increment among all other support regions that share these fine-scale volumes. In essence, while the second sub-function spreads the influence inside each Prolongation Operator column $(\underset{\sim}{P}_{op})_j^{\eta}$, the last sub-function restricts this influence to fine-scale volumes contained in each support region and while the first spreads this lost influence by normalizing the Prolongation Operator rows $(\underset{\sim}{P}_{op})_i^{\eta}$ of fine-scale volumes that belong to the Global Support Boundary. Thus, sub-functions 1 and 3 are a sort of Reduced Boundary conditions. It worth remembering that the primal coarse cell centers (x^P) by definition do not belong to any support region nor any boundary region, therefore the values attributed by Eq. (45) remain unchanged. In other words, the centers take 1 for values on the Prolongation Operator Column $(\underset{\sim}{P}_{op})_{x^P j}^{\eta}$ associated with the primal coarse volume which they are centers and 0 otherwise.

3. Finally, the Prolongation Operator is updated using the modified increment.

$$(\underset{\sim}{P}_{op})_j^{\eta+1} = (\underset{\sim}{P}_{op})_j^{\eta} + d_j \tag{49}$$

4. Local error is defined for volumes not in the Global Support of Boundary Region.

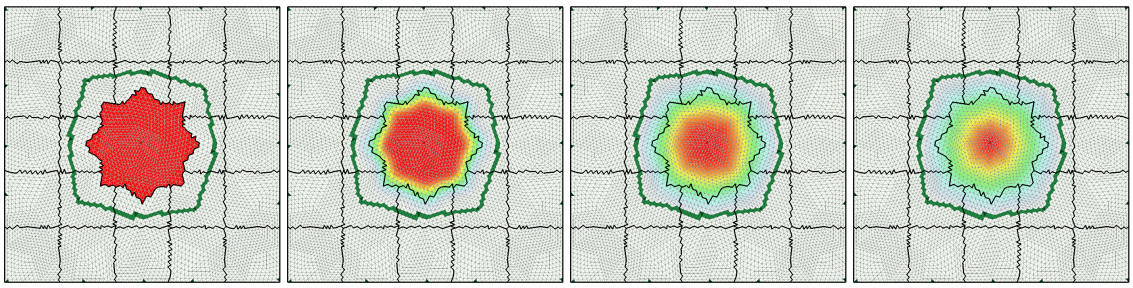
$$e_j = \max_i(\|\widehat{d_{ij}}\|), \qquad i \notin G \tag{50}$$

5. Check for convergence. If $\|e\|_\infty > tol$ the current $(\underset{\sim}{P}_{op})_j^{\eta}$ is fed to Step 1, otherwise set $(\underset{\sim}{P}_{op}) = (\underset{\sim}{P}_{op})_j^{\eta}$.

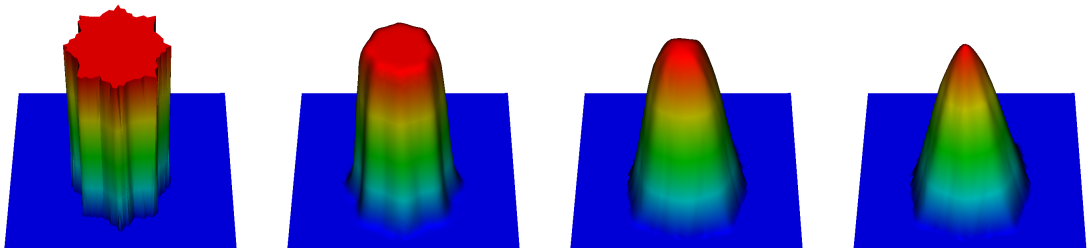
The original Multiscale Restricted Smoothed Basis is named after the process reported above. Figure (4) illustrates how the initial indicator function is smoothed restricted to the corresponding support region.

Flux Reconstruction

The difficulty in finding a consistent approximation for the transport equations lays on the fact it is mandatory to obtain a conservative velocity field in all grids. FV approximations are by definition conservative, however, the multiscale pressure solution derived from prolongating the solution of the coarse-scale system gives birth to a velocity field that is only locally conservative, which means they are discontinuous on the surface of each support region but continuous inside it. This is due to the use of boundary conditions in Eq. (48) that neglected normal flux out of the support region in order to uncouple the domain. Therefore, to obtain the MsCV approximation we need to reconstruct a fully conservative fine-scale velocity field. This is done in two straightforwards steps (Jenny, Lee and Tchelepi,2006). As a consequence of fluxes being discontinuous on surfaces of the support region, the interface coarse volumes remains conservative. The first step is to retrieve these fluxes, in order to use them as Neumann Boundary Conditions to solve Eq. (1) restricted to each primal coarse volume. The second step is to use the resulting pressure fields to calculate a new velocity field. The reconstructed flux is a composition of the velocities found in these two steps. For the boundaries of each coarse volume, we use the velocity calculated on the first step, inside, we use the second velocity.



(a) Initial guess. **(b) 31 iterations.** **(c) 171 iterations.** **(d) Converged.**



(e) Initial guess. **(f) 31 iterations.** **(g) 171 iterations.** **(h) Converged.**

Figure 4: Illustration of the MsRSB iteration process.

Error Evaluation

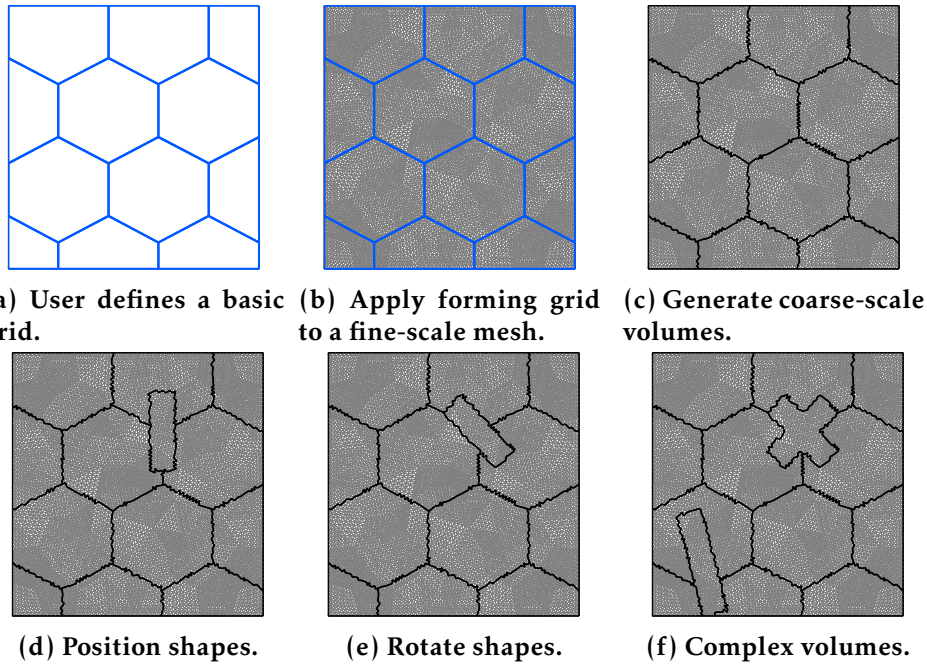
By construction, multiscale methods rely on the numerical scheme used to discretize the flux on the underlying high-resolution mesh. Therefore, the quality of the multiscale solution is always bounded by the direct simulation on the fine grid or reference solution, which becomes a theoretical upper limit. For a proper comparison of the two-phase flow, the saturation field needs to be evaluated after the same amount of simulation time. Thus, the relative error and the average error are defined as:

$$e_{rel} = \frac{||\vec{S}^{ref} - \vec{S}^{ms}||}{||\vec{S}^{ref}||} \qquad \overline{e} = \frac{1}{N}\sum_{j=1}^{N}(e_{rel})_i \tag{51}$$

where $\vec{S}^{ms}$ is a $\vec{S}^{ref}$ are the multiscale and the reference solution for the saturation field. The mean error is defined as the arithmetic mean of the error calculated for each CV in a grid with N control volumes.

4.2 MsCV Preprocessing Algorithms

In order to make a MsCV compatible with general grids we must understand the peculiarities these grids posses to propose algorithms to generalize the ideas behind the MsFVM. The first main difference is the MsCV allows coarse volumes to assume general non-regular shapes. This makes the coarsening algorithm not as trivial as agglomerating fine-scale cells in rectangular volumes. The second main difference is that the fine-scale volumes may also be fully unstructured. As a consequence, we must define coarsening algorithms, calculate the primal cell centers and define support regions. The preprocessing stage consists in all these three steps. For the fine



(a) User defines a basic grid. **(b) Apply forming grid to a fine-scale mesh.** **(c) Generate coarse-scale volumes.**

(d) Position shapes. **(e) Rotate shapes.** **(f) Complex volumes.**

Figure 5: Geometric partitioner to generate coarse.

mesh discretization step, we use Gmsh (Geuzaine and Remacle,2009), a finite-element mesh generator released under General Public License (GPL) and broadly used in the numerical simulation community.

Coarsening Algorithms

We have developed a geometric partitioner for generating primal coarse-scale grids. This partitioner is comprised of a set of routines that allows us to literally draw the contours of the coarse volumes. This approach showed better results in most cases in comparison to coarse grids generated using Metis (Karypis and Kumar,2009). These routines use a basis algorithm that checks whether the centroid of a fine-scale volume lays inside of a general polygon. This algorithm takes as input a generally defined polygon, all fine-scale volume are checked and those with center laying inside or on the boundaries of the polygon are defined as belonging to the resulting coarse volume. The first routine we have developed creates a base coarse mesh using simple geometric shapes and checks which fine cell belongs to each coarse volume. Figure 5 illustrates this process. First, the user inputs which basic shape will be used along with the number of elements in the X and Y axes, this creates a basic forming grid (Fig. 5a) that is applied upon a fine-scale mesh (Fig. 5a). Each volume in this basic forming coarse grid uses the basis algorithm to check the fine-scale volumes that belong to them. As a consequence, the resulting coarse-scale volumes have contours that resemble but are not necessarily equal to those in the original forming grid (Fig. 5c). Additionally, we have also developed routines that let the user to draw different shapes on the coarse-scale grid, which in turn uses inpolygon to create additional coarse volumes. This allows the user to generate complex, and non-conforming coarse-scale meshes taking in account the physical and geometrical properties of the media. Figure 5 illustrates some of its features. It is possible to input the size and position where a basic shape is applied

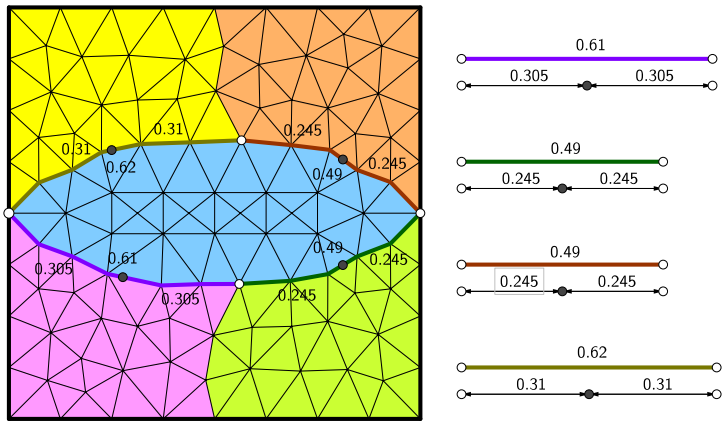


Figure 6: Finding the center of the interfaces.

(Fig. 5d). User can rotate and manipulate (Fig. 5e) as well as instance as many shapes as needed to compose a complex cell (Fig. 5f). After these steps, we apply an algorithm that checks the coarse-scale mesh integrity by ensuring that all fine-scale volumes in a coarse-volume share at least one edge with each other and relocate volumes which do not meet this criteria. The ideas behind these functions is that in the future after in-depth studies we can create a fully automatic coarsening routine that reads the geometry of the problem and choose the number and the shapes of the coarse volumes accordingly.

Cell Center Calculation for Coarse Mesh

The classical MsFV approach that uses the fine-scale volume closest to the centroid of the primal coarse volume is not suitable for general unstructured grids as convex-shaped volumes may have centroids that lay too close to the boundaries of the polygon or even outside it. Moyner and Lie (2015) proposes a three step algorithm to find x^P. Figure 6 illustrates an unstructured coarse mesh with an unitary domain. The first step is to find the interfaces between the blue coarse volume and its neighbors (colorful thick lines around the center coarse volume). Second, we calculate the middle of these interface by finding the point that splits in two the length of each interface. Finally, we use these points to calculate the Geometric Median. By definition, this median finds a point which minimizes the sum of the distances of the sampled points. In other words, the center point of a coarse volume is the point which has the lowest sum of distances to the middle edges interfaces. The center of a primal coarse volume (x^P) is the fine-scale cell whose coordiantes of the center lays closest to its geometric median.

Support Region Generation

We employ the algorithm proposed by Moyner and Lie (2015) with slightly modifications. Figure 7 illustrates the process that generates the support region and the support boundary associated with the central primal coarse volume whose center is green. Notice that all other required geometric entities, such as the Global Support Boundary and Global Support Boundary in a Support Region, are calculated by using their very definition.

(a) The first step is to define which primal coarse volumes are neighbors of the center coarse volume. We define neighbors volumes that share at least one node.

(b) We find the middle of the interface of the central volume with each of its neighbors (blue nodes). We also find the middle of the interface among all neighbor volumes (red nodes).

(c) We use these points plus the center points of the neighbor coarse cell center (black nodes in pink volumes) to perform a Delaunay triangulation.

(d) For a consistent support region, no neighbor coarse cell center may lay inside of this region. Then, we perform a search among the triangles generated by the Delaunay triangulation to find those that do not meet this condition. In other words we find those triangles that prevent the coarse cell center nodes (black nodes) to be on the boundary of the resulting shape.

(e) We remove the triangles found on the previous step from the Convex Hull of the points found at the (b) step.

(f) We use the basis algorithm using the resulting polygon to find which fine-scale volumes lay inside it (blue volumes).

(g) We search for fine-scale volumes that share at least two edges with this region.

(h) We add these volumes to the blue region to formally define the support region of the middle coarse volume.

(i) Finally, we define as the support boundary all volumes that share at least one node with the volumes inside the support region.

5 Results

For the two examples presented in this section we consider a unitary domain $\Omega = [0\ 1]\times[0\ 1]$ in a 1/4 of five spot configuration, in which, flux is prescribed ($Q_{inj} = 1$) at the injector well and pressure is prescribed ($p_{prod} = 0$) at the producer well. Additionally, water is being injected at the injection well ($S_w = 1$) in a field initially saturated by oil only ($S_o = 0$). Moreover, the permeability tensors used in these examples are defined as: $K_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $K_2 = \begin{bmatrix} \frac{1}{1000} & 0 \\ 0 & \frac{1}{1000} \end{bmatrix}$ and $K_3 = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}$.

5.1 Two-Phase Flow in 1/4 of Five Spot with a Central Low Permeability Barrier

In this context, our first example (See Fig. 8*a*) consists in a low permeability zone shaped as a square with size (L=0.5) and concentric with the unitary domain. In this example, the simulation was performed on a fine-scale grid containing 2432 volumes on coarse-scale 7x7 coarse grid. Under this setting, we were capable of obtaining results that qualitatively stayed very close to the reference solution (See Fig. 9). Notice that the multiscale solution was capable of retaining most of the features important for the flow. These results are supported by the curves on Fig. 11a that show almost

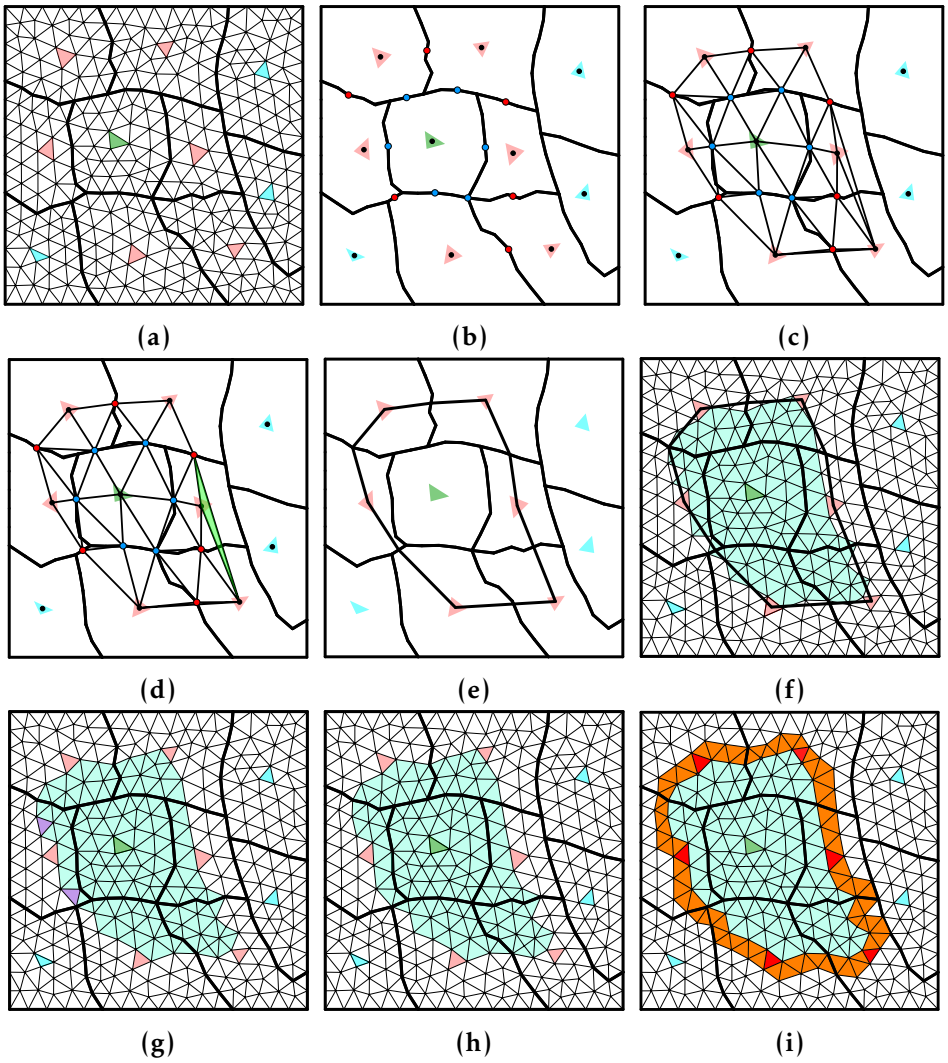
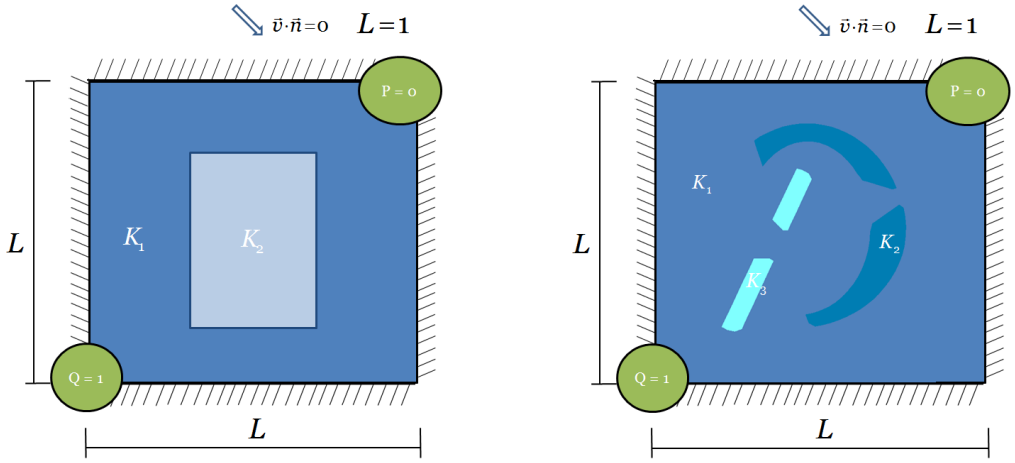


Figure 7: Support region and boundary support region generation for general unstructured grids.



(a) Example 5.1 **(b) Example 5.2**

Figure 8: Permeability fields of the examples.

identical cumulative oil plots. Nonetheless, the multiscale solution has anticipated the breakthrough. Moreover, Table 1 shows the average error on the saturation field is as low as $||s||_2 = 1.89\%$.

5.2 Two-phase flow in a 1/4 of five- spot with two unconnected channels and two curved barriers

On the second example, we present a geological formation for which classical multiscale methods would have trouble to represent as they would require high-resolution fine-grids to obtain a discrete system capable of preserving the complex features of this geological formation (See Fig. 8*b*). By using this framework, we were capable of running a very accurate simulation with only 3060 fine-scale volumes and 16 coarse volumes. Figure 10 shows that qualitatively the solutions are very close showing that this scheme was capable of preserving high-resolution features of the flow. The curves presented on Fig. 11 support this affirmation by showing very close Cumulative Oil curves. Moreover, in this example the breakthrough in the multiscale and reference simulation take place at the same time. Table 1 also shows a very low average error $||s||_2 = 5.28\%$.

Table 1: Average error comparison

PVI	Example 5.1 $\|\|s\|\|_2$	Example 5.2 $\|\|s\|\|_2$
0.1	9.10%	14.06%
0.4	6.91%	10.86%
0.5	5.24%	8.88%
1	1.89%	5.28%

6 Conclusions

In this article, we have presented a framework that extends the original MsRSB formulation by using a non-orthodox MPFA-D, thus enabling two-phase flow simulation of oil and water in heterogeneous and anisotropic petroleum reservoirs using truly unstructured grids. We also present the algorithms used on the pre-processing stage, essential to generate the geometric entities required to run the multiscale simulation on unstructured grids. As the examples have shown, the saturation solutions provided by this framework are capable of preserving high-resolution features of the flow. Additionally, unstructured fine-scale grids can improve the discretization of complex geological formations enabling simulation on fine meshes with a significantly lower number of control volume in comparison to the structured grid counterparts. In conclusion, the framework we have presented can produce accurate solutions at reasonably computational costs on the simulation of two-phase flow in heterogeneous and anisotropic porous media using general unstructured grids.

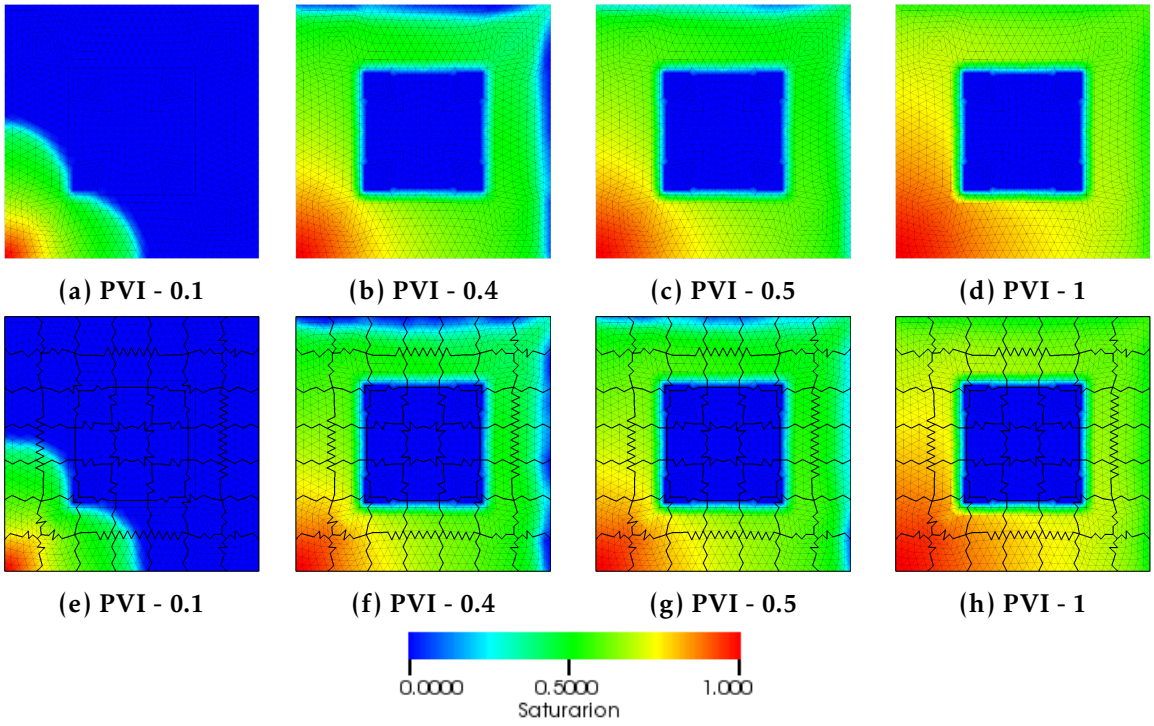


Figure 9: Reference and multiscale solution for the saturation field of example 1.

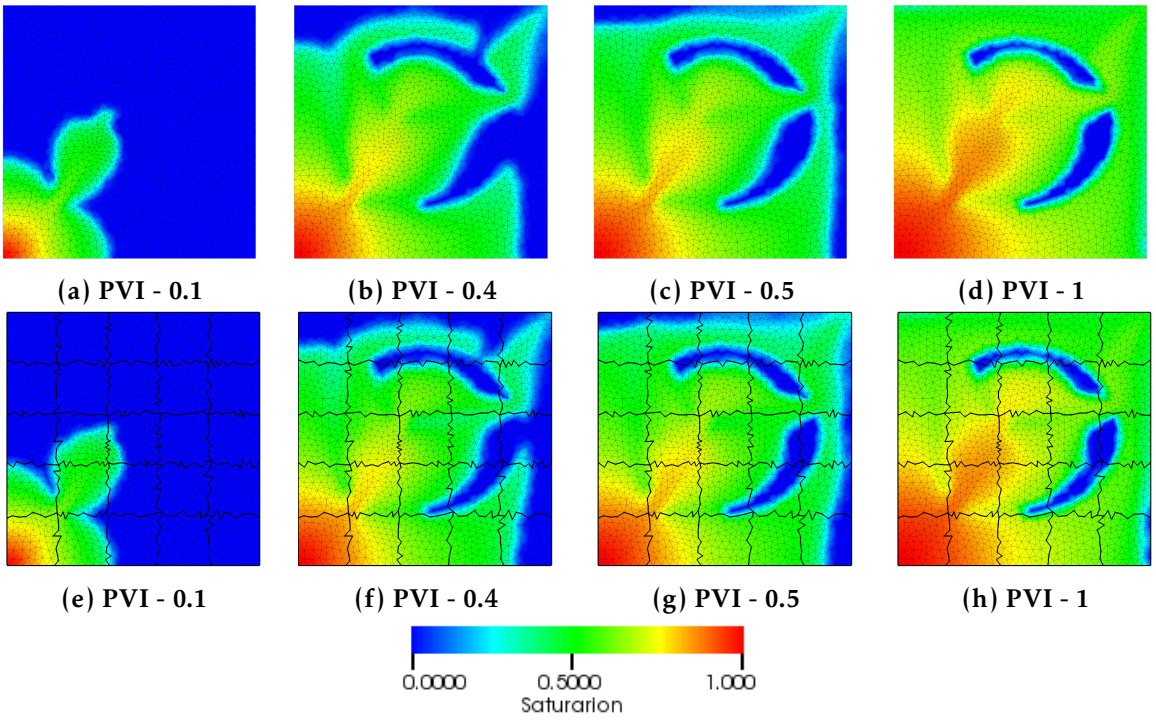
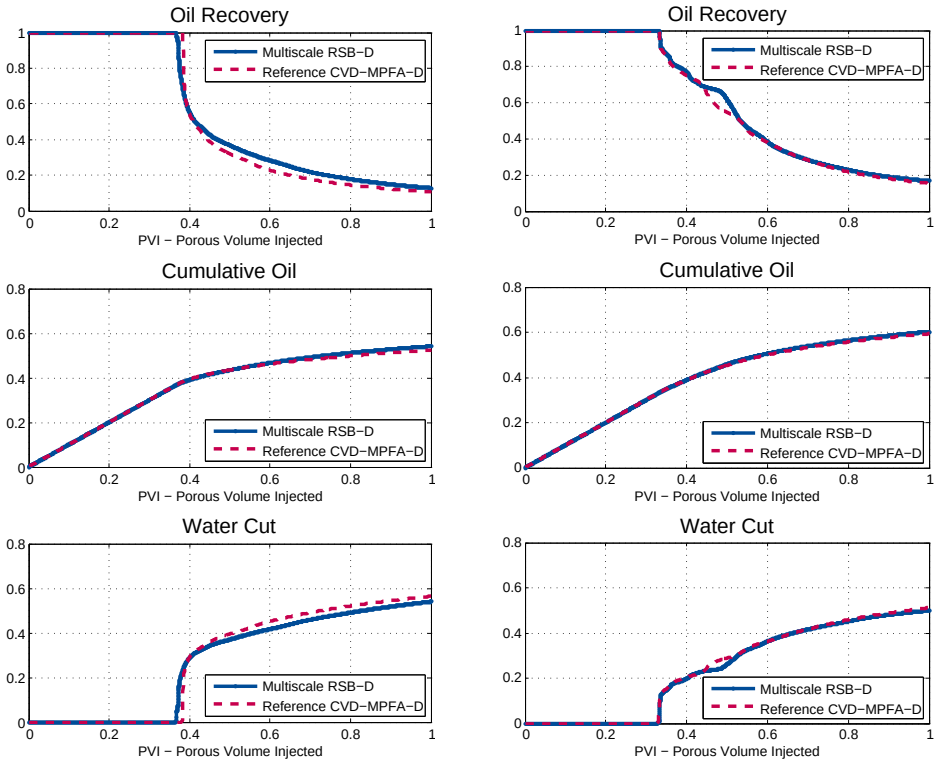


Figure 10: Reference and multiscale solution for the saturation field for example 2.



(a) Example 5.1. **(b) Example 5.2.**

Figure 11: Production curves.

ACKNOWLEDGEMENTS

The author of this article would like to thank the Science Support Foundation of the State of Pernambuco (FACEPE) and the National Council for Scientific and Technological Development (CNPq) for funding this research.

REFERENCES

BARBOSA, L. M. C. *Investigação de Formulações Multi-Escala Localmente Conservativas para a Simulação de Reservatórios de Petróleo Muito Heterogêneos.* Thesis (PhD) — Federal University Of Pernambuco, Recife, Brazil, 2017.

CHEN, Z.; HOU, T. Y. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Mathematics of Computation*, v. 72, n. 242, p. 541–577, 2002.

CONTRERAS, F. et al. A cell-centered multipoint flux approximation method with a diamond stencil coupled with a higher order finite volume method for the simulation of oil-water displacements in heterogeneous and anisotropic petroleum reservoirs. *Computers & Fluids*, v. 127, p. 1–16, 2016.

FARMER, C. L. Upscaling: a review. *International Journal for Numerical Methods in Fluids*, v. 40, n. 1-2, p. 63–78, 2002.

FRIIS, H. A.; EVJE, S. Numerical treatment of two-phase flow in capillary heterogeneous porous media by finite-volume approximations. *Int. J. Numer. Anal. Mod*, v. 9, p. 505–208, 2012.

GAO, Z.; WU, J. A linearity-preserving cell-centered scheme for the heterogeneous and anisotropic diffusion equations on general meshes. *International Journal for Numerical Methods in Fluids*, v. 67, n. 12, p. 2157–2183, 2010.

GEUZAINE, C.; REMACLE, J.-F. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, v. 79, n. 11, p. 1309–1331, 2009.

HOU, T. Y.; WU, X.-H. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, v. 134, n. 1, p. 169–189, 1997.

JENNY, P.; LEE, S.; TCHELEPI, H. Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media. *Journal of Computational Physics*, v. 217, n. 2, p. 627–641, 2006.

KARYPIS, G.; KUMAR, V. *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*. 2009.

KIPPE, V.; AARNES, J. E.; LIE, K.-A. A comparison of multiscale methods for elliptic problems in porous media flow. *Computational Geosciences*, v. 12, n. 3, p. 377–398, 2008.

KROGSTAD, S. et al. A multiscale mixed finite-element solver for three-phase black-oil flow. *Society of Petroleum Engineers*, 2009.

MOYNER, O.; LIE, K.-A. A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids. *Journal of Computational Physics*, v. 304, p. 46–71, 2015.

PARRAMORE, E. et al. Multiscale finite-volume cvd-mpfa formulations on structured and unstructured grids. *Multiscale Modeling & Simulation*, v. 14, n. 2, p. 559–594, 2016.

SOUZA, M. R. de A. *Simulação Numérica de Escoamento Bifásico em Reservatórios de Petróleo heterogêneos e Anisotrópicos Utilizando um Método de Volumes Finitos Verdadeiramente Multidiensional com Aproximaação de Alta Ordem.* Thesis (PhD) — Federal University of Pernambuco, 2015.

ZHOU, H. *Algebraic Multiscale Finite-Volume Methods for Reservoir Simulation*. Thesis (PhD) — Stanford University, 2010.

ZHOU, H.; TCHELEPI, H. A. Operator-based multiscale method for compressible flow. *SPE Journal*, v. 13, n. 02, p. 267–273, 2008.