Numerical Simulation of Oil-Water Displacements Using a Higher Order Control Volume Formulation in Parallel Computers with Distributed Memory

Rogério Soares da Silva, rogsoares@yahoo.com.br
Darlan Karlo Elisiário de Carvalho, dkarlo@uol.com.br
Paulo Roberto Maciel Lyra, prmlyra@padmec.br
Ramiro Brito Willmersdorf, ramiro@willmersdorf.net
Alessandro Romário Echevarria Antunes, aantunes@yahoo.com.br
Departamento de Engenharia Mecânica, UFPE, Av. Acadêmico Hélio Ramos, s/n, CEP: 50740-530, Recife - PE – Brazil.

Abstract. *In this paper, we present a parallel implementation of a node centered edge-based finite volume formulation (EBFV) to simulate the immiscible and incompressible fluid flow of oil and water in petroleum reservoirs. A modified version of the classical IMplicit Pressure Explicit Saturation (IMPES) method was applied. In the Modified IMPES approach (MIMPES), the pressure equation is solved first and then the velocity field is computed less frequently than, the saturation field. This strategy works well as far as the velocity field varies slowly throughout the simulation implying that the saturation field can be updated several times before the pressure and the velocity fields have to be updated. The elliptic equation is solved implicitly by a variation of Crumpton's two step approach, then velocities are directly computed from the pressure field and the saturation equation is solved explicitly through a higher order MUSCL (Monotone Upwind Scheme for Conservation Laws) method. The computer program was developed using C++ to perform the simulations in serial and in distributed memory parallel computers. Two different alternatives were implemented and compared to perform the iterative solution of the final non-symmetric algebraic system of equations originated from the pressure equation. Several packages are incorporated to the simulator to handle specialized tasks. FMDB (Flexible Mesh Distributed Database) is used to manage the mesh data structure. ParMetis and autopack are also used. The first handles load balancing to minimize communication between partitions, and the latter improves data migration between partitions. The system of equations for the pressure variable is solved through PETSC (Portable, Extensible Toolkit for Scientific Computation) which provides a set of linear solvers and pre-conditioners. This programming model is known as message-passing model and makes use of MPI protocols. Some applications and performance studies are shown to demonstrate the validity of the developed code.*

Keywords*: Finite Volume Method, Edge based, MUSCL, Modified IMPES, Parallel Computers*

1. INTRODUCTION

One of the most popular methodologies used to describe the two phase flow of oil and water in petroleum reservoirs is the IMPES (Implicit Pressure Explicit Saturation) procedure (Ewing, 1983; Carvalho *et al.*, 2009). In this technique, a sequential time stepping procedure is used to split the computation of the pressure and the saturation fields. In the classical IMPES approach, starting, from an initial saturation distribution, the pressure equation is solved implicitly and then, the total velocity is explicitly computed from this pressure field. Following, this velocity field is used as an input for the saturation equation, which is finally solved explicitly. This process is repeated until the end of the simulation. For the incompressible and immiscible two-phase flow of oil and water in rigid porous media, the pressure field is described by an elliptic equation that can have strong discontinuous coefficients (i.e. permeabilities) and, in general, the saturation equation is similar to a convection-diffusion type equation, in which the diffusion coefficients are associated to capillarity effects. The computation of the pressure field, at each time step, involves the solution of a system of equations which is, in general, much more CPU demanding than the explicit computation of the saturation equation. On the other hand, due to explicit solution of the saturation equation, severe time restrictions are imposed on the simulation. For large scale problems, the CPU cost of the classical IMPES procedure can become prohibitive leading researchers to find other ways to make simulation viable. In order to circumvent this problem, we developed a parallel implementation of a modified IMPES method. In the Modified IMPES approach (MIMPES), the pressure equation is solved and the velocity field is updated much less frequently than the saturation field, using the fact that, usually, the total velocity field varies slowly throughout the simulation, implying that the saturation field can be updated several times before we have to update the pressure/velocity fields (Hurtado *et al.*, 2006; Chen *et al.*, 2008).

2. MATHEMATICAL MODEL

In the present section, we briefly describe the governing equations for incompressible and immiscible, two-phase flows of water and oil through rigid porous media. This model is obtained by combining Darcy's Law with the mass conservation equation for each phase. The model adopted here has been successfully used by many authors (Carvalho *et al.*, 2009a), (Chen *et al.*, 2002), (Ewing, 1983) and (Hurtado *et al.*, 2006).

Initially, we assume that the phase velocities obey the Darcy's law, which, ignoring gravitational effects can be written for phase *i*, as

$$\vec{v}_i = -\underset{\sim}{\lambda}_i \nabla P_i \tag{1}$$

where $\underset{\sim}{\lambda}_i$ is the phase mobility tensor. Henceforth, we will assume incompressible medium and fluids. We will also ignore the capillary pressure and assume that $P = P_w = P_o$, where (w) and (o) stand, respectively, for the wetting (water) and the non-wetting (oil) phases. Additionally, conservation of mass for each phase *i* can be written as

$$-\nabla \cdot \left(\rho_i \vec{v}_i\right) + q_i = \frac{\partial \left(\phi \rho_i S_i\right)}{\partial t} \tag{2}$$

In Equation (2), ϕ is the porosity, i.e. fraction of the rock which can be occupied by fluids, q_i denotes sources or sinks, ρ_i is the phase density and S_i is the saturation of phase *i*, which represents the percentage of the available pore space occupied by this phase. Due to this last definition, we can write

$$S_o + S_w = 1 \tag{3}$$

Combining Eq. (2) to Eq. (3) and after some algebraic manipulation we obtain the following pressure equation

$$\nabla \cdot \left(\underset{\sim}{\lambda} \nabla P\right) = -Q \text{ or } \nabla \cdot \vec{v} = Q \tag{4}$$

where, $\underset{\sim}{\lambda} = \underset{\sim}{\lambda}_o + \underset{\sim}{\lambda}_w$ is the total fluid mobility tensor, $\vec{v} = \vec{v}_o + \vec{v}_w = -\underset{\sim}{\lambda} \nabla P$ is the total velocity field which represents the sum of phase velocities and $Q = Q_w + Q_o$, with $Q_i = \left(q_i / \rho_i\right)$, is the total injection or production specific rate. By introducing the fractional flow function $f_i = \lambda_i / \left(\lambda_o + \lambda_w\right)$, we can also derive a hyperbolic equation for the water saturation, which can be written as

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot \vec{F}_w(S_w) = Q_w \tag{5}$$

The term $\vec{F}_w = f_w \vec{v}$ is the flux function which is strongly dependent on the water phase saturation. As it can be seen, the pressure and saturation fields are connected through the total velocity $\vec{v}$.

3. NUMERICAL FORMULATION

In order to discretize the pressure and the saturation equations, i.e., Eq. (4) and (5), respectively, we have adopted a vertex centered, median dual finite volume (FV) method, in which the coefficients necessary to our calculation are associated to the edges and to the vertex of the computational mesh (Lou *et al.*, 1995; Carvalho *et al.*, 2009b). These edge and node coefficients are pre-computed in a pre-processing stage from the more traditional element data structure which is commonly used in the finite element method.

The median dual control volumes adopted here are built connecting centroids of elements to the middle point of the edges that surround a specific mesh node, even though alternative control volumes could be used (e.g. centroid dual). In edge-based vertex centered schemes, fluxes are usually integrated on the dual mesh through one or more loops over the edges, and the computational cost is, essentially, proportional to the number of edges of the mesh. To properly handle porous media (i.e. material) discontinuities, we perform the integration over the whole domain in a sub-domain by sub-domain approach, where a sub-domain is defined by a group of elements that share the same physical properties such as permeability and porosity.

In this work, we have chosen to use the cell distributed methodology due to the easiness of associating rock properties to sub-domains which naturally fit to reservoir bed boundaries. For a detailed description of all steps to obtain the discretized pressure, velocity and saturation equations see (Carvalho *et al.*, 2009a).

3.1. Implicit pressure equation

In Equation (6), we present the discretized form of the pressure equation for 2-D and 3-D problems.

$$\sum_{R=1}^{Ndom}\left[\sum_{L=1}^{NN(\Omega_R)}\left(-\underset{\sim}{\lambda}_{IJ_L}^{\Omega_R}\left(\frac{\left(\nabla\hat{p}_I^{\Omega_R}+\nabla\hat{p}_{J_L}^{\Omega_R}\right)}{2}-\left(\frac{\left(\nabla\hat{p}_I^{\Omega_R}+\nabla\hat{p}_{J_L}^{\Omega_R}\right)}{2}\cdot\vec{L}_{IJ_L}\right)\vec{L}_{IJ_L}+\frac{\left(\hat{p}_{J_L}-\hat{p}_I\right)}{\left|\Delta_{IJ_L}\right|}\vec{L}_{IJ_L}\right)\right)\cdot\vec{C}_{IJ_L}^{\Omega_R}\right]+$$
$$\sum_{R=1}^{Ndom}\left[\sum_{L=1}^{NN(\Gamma_R)}\left(\underset{\sim}{K}^{\Omega_R}\hat{p}_{IJ_L}^{\Omega_R}\right)\cdot\vec{D}_{IJ_L}^{\Omega_R}\right]=\sum_{R=1}^{Ndom}Q_I^{\Omega_R}V_I^{\Omega_R} \tag{6}$$

The terms of equations N_{dom}, NN, Ω_R, $\nabla\hat{p}_I$, $\vec{L}_{IJ_L}$, $\vec{C}_{IJ_L}$, Q_I, V_I refers to, respectively, the number of domains, number of neighbors nodes connected to node I, the domain, the approximated pressure gradient, the *IJ* edge vector, the orthogonal vector to control surface, the source/sink term and the control volume. After the pressure field is obtained, the mid-edge velocity field is computed by Eq. (7), as

$$\vec{v}_{IJ_L}^{\Omega_R}=-\underset{\sim}{\lambda}_{IJ_L}^{\Omega_R}\left(\frac{\left(\nabla P_I^{\Omega_R}+\nabla P_{J_L}^{\Omega_R}\right)}{2}-\left(\frac{\left(\nabla P_I^{\Omega_R}+\nabla P_{J_L}^{\Omega_R}\right)}{2}\cdot\vec{L}_{IJ_L}\right)\vec{L}_{IJ_L}+\frac{\left(P_{J_L}-P_I\right)}{\left|\Delta_{IJ_L}\right|}\vec{L}_{IJ_L}\right) \tag{7}$$

In this paper, we have dealt with isotropic and anisotropic porous media assuming that $\underset{\sim}{\lambda}_{IJ_L}^{\Omega_R}=\underset{\sim}{K}_{IJ_L}^{\Omega_R}\lambda_{IJ_L}$, where $\underset{\sim}{K}_{IJ_L}^{\Omega_R}$ can be a full tensor satisfying the ellipticity condition $K_{xx}K_{yy}\geq K_{xy}^2$ where K_{xx}, K_{yy} and K_{xy} are the entries of the permeability tensor $\underset{\sim}{K}$. The edge values of the scalar mobility terms are approximated using a mid-point rule in order to formally guarantee second order accuracy, i.e. $\lambda_{IJ_L}=\left(\lambda_I+\lambda_{J_L}\right)/2$ and viscosity is constant under the assumption of incompressible and isothermal flow. Therefore, we can redefine Eq. (6) using the flux function approximation as

$$\sum_{R=1}^{Ndom}\left(\sum_{L_{I(\Omega_R)}}\vec{F}_{IJ_L}^{\Omega_R}\cdot\vec{C}_{IJ_L}^{\Omega_R}+\sum_{L_{I(\Gamma_R)}}\vec{F}_{IJ_L}^{\Gamma}\cdot\vec{D}_{IJ_L}^{\Omega_R}\right)=\sum_{R=1}^{Ndom}Q_I^{\Omega_R}V_I^{\Omega_R} \tag{8}$$

where $\vec{F}_{IJ_L}^{\Omega_R}$ stands for the flux through control surfaces.

3.2. Explicit saturation equation

Usually, in petroleum reservoir simulators, the discretization of the advective term that characterizes the hyperbolic saturation equation is performed by the classical first order upwind (FOU) method, which is capable of completely eliminating spurious oscillations at the cost of introducing a large amount of artificial diffusion (Ewing, 1983). On the other hand, pure second order schemes produce physically unrealistic results, with overshoots and/or undershoot in the vicinity of sudden changes in the saturation field (i.e. shocks). By integrating Eq. (5) and applying the divergence theorem we can write

$$\int_\Omega\phi\frac{\partial S_w}{\partial t}\partial\Omega+\int_\Gamma\vec{F}_w\left(S_w\right)\cdot\vec{n}\,\partial\Gamma=\int_\Omega Q_w\partial\Omega \tag{9}$$

For which the edge-based finite volume discretized form is given by Eq. (10), as

$$\hat{S}_w^{n+1}=\hat{S}_w^n-\sum_{r=1}^{Ndom}\frac{\Delta t}{\phi^{\Omega_r}V^{\Omega_r}}\left(\sum_{L=1}^{NN^{(\Omega_r)}}\vec{F}_{IJ_L(w)}^{\Omega_r}\cdot\vec{C}_{IJ_L}^{\Omega_r}+\sum_{L=1}^{NN^{(\Gamma_r)}}\vec{F}_{IJ_L(w)}^{\Gamma_r}\cdot\vec{D}_{IJ_L}^{\Omega_r}+Q_I^{\Omega_r}V_I^{\Omega_r}\right)^n \tag{10}$$

The source term, is non zero only at production wells. For a particular mesh node *I*, the second term in the left hand side of Eq. (9) is approximated as

$$\int_{\Gamma_I}\vec{F}_w\left(S_w\right)\cdot\vec{n}\,\partial\Gamma_I\cong\sum_{L_I}\vec{F}_{IJ_L(w)}\cdot\vec{C}_{IJ_L}=\sum_{L_I}\frac{1}{2}\left[\left(\vec{F}_I^-\left(S_{I(w)}^-\right)+\vec{F}_{J_L}^+\left(S_{J_L(w)}^+\right)\right)\cdot\vec{C}_{IJ_L}-\alpha_{IJ_L}\left(S_{J_L(w)}^--S_{I(w)}^+\right)\right] \tag{11}$$

where $\alpha_{IJ_L} = \left|\vec{v}_{IJ_L}\right|\left|\Delta f_{IJ_L(w)}/\Delta S_{IJ_L(w)}\right|$, with $\Delta f_{IJ_L(w)}/\Delta S_{IJ_L(w)} = \left(f_{J_L(w)} - f_{I(w)}\right)/\left(S_{J_L(w)} - S_{I(w)}\right)$ and the superscripts (-) and (+) are used to indicate that fluxes are computed using the following linear extrapolated saturation values.

$$S^{+}_{I(w)} = S_{I(w)} + \frac{\psi^{*}_{I}}{2}\left(\nabla S_{I(w)} \cdot \overrightarrow{IJ_L}\right) \text{ and } S^{-}_{J_L(w)} = S_{J_L(w)} + \frac{\psi^{*}_{J_L}}{2}\left(\nabla S_{J_L(w)} \cdot \overrightarrow{IJ_L}\right) \tag{12}$$

where $\overrightarrow{IJ_L}$ is the length vector in the edge direction (i.e. $\vec{x}_{J_L} - \vec{x}_I$), and $\psi^{*}_{I} = \psi_I \psi_{IJ_L}$ is a slope limiter which must smoothly varies from one (second order scheme) to zero (first order scheme) in the vicinity of saturation shocks. ψ_I is responsible for switching the scheme from second order to first order whenever necessary and ψ_{IJ_L} is responsible for the edge interpolative boundedness, i.e., it guarantees that the extrapolated values of the saturation throughout the edge remain between $S_{I(w)}$ and $S_{J_L(w)}$. Whenever using elements with high aspect ratios, which are common in mesh adaptive processes, other alternatives, such as the gradient extrapolation approach or the artificial dissipation scheme have, respectively, produced erroneous solutions with noticeable over and undershoots or overly diffusive solutions (Carvalho *et al.*, 2009a).

4. MODIFIED IMPES APPROACH

As previously mentioned, the IMPES method is a segregated type method in which the flow equations are manipulated in order to produce an elliptic pressure equation, solved implicitly and a hyperbolic type saturation equation, which is then solved explicitly. In classical IMPES method the pressure and the saturation fields are updated assuming a common time step. On the other hand, the Modified Implicit Pressure Explict Saturation approach (MIMPES) consists in assigning larger time steps $\left(\Delta t_p\right)$ to the implicit pressure equation than those used to solve the saturation equation $\left(\Delta t_{CFL}\right)$ which is constrained by CFL condition. The latter equationis solved repeatedly until the saturation filed reaches the same time of the pressure field, i.e. the summation of all Δt_{CFL} must be equal to Δt_p. Then, a new Δt_p is calculated and a new summation of Δt_{CFL} is performed. The major advantage of this strategy is that the implicit pressure equation, which represents more than 90% of all calculations, is solved several times less than the explicit saturation equation. The MIMPES algorithm which we are using in the present paper, is an edge-based implementation of the original element-based algorithm proposed by Hurtado *et al.* (2006) with a slightly modification. The time step control strategy is based on the velocity field variation. An algorithm that implements this idea has the following steps:

Set velocity variation tolerance (DVTOL)
while simulation not finished
  Set time-step counter $\Delta t_{sum} = 0$ *;Set MIMPES:= true Set New Implicit Time-step; NITS=true,*
  Solve matrix system (pressure field) Eq. (8)
  while MIMPES is true
    Compute velocity field, Eq. (7),
    Compute CFL restricted time-step Δt_{CFL} *and velocity norm,* $\left|\Delta \vec{v}_T\right|^n$
    Compute:
      $\Delta t_p^{n=0} = \Delta t_{CFL}$
      $\Delta t_{sum} = \Delta t_{sum} + \Delta t_{CFL}$;
      if NITS is true
        $\Delta t_p^{n+1} = \dfrac{DVTOL}{\left|\Delta \vec{v}_T\right|^n} \Delta t_p^n$;
        NITS=false
      end
    if $\Delta t_{sum} \geq \Delta t_p^{n+1}$
      MIMPES = false
    end
    Compute saturation field, Eq. (10)
  end
end

After calculating the velocity field, a L_2 norm, $\left|\Delta\vec{v}_T\right|^n$, of all mid-edge velocities is computed and the new time-step $\left(\Delta t_P^{n+1}\right)$ for the pressure equation is computed by

$$\Delta t_P^{n+1} = \frac{DVTOL}{\left|\Delta\vec{v}_T\right|^n}\Delta t_P^n \tag{13}$$

where *DVTOL* represents an empiric number that turns the simulation more conservative (closer to classical IMPES and consequently slower) or more dynamic (faster but less accurate). Previous tests with different DVTOL values (Carvalho *et al.*, 2009a) showed that 0.05 can give good results, i.e., faster than classical IMPES and with acceptable accuracy.

At the very first simulation time step, Δt_P^n assumes the value defined by the CFL restriction of the explicit formulation adopted to solve the saturation equation, i.e., $\Delta t_P^{n=0} = \Delta t_{CFL}$.

In order to avoid "*pressure time-steps*" that are too large, which could affect the accuracy, or time-steps that are too small, extremely diminishing the efficiency of the procedure, the following control procedure (Hurtado *et al.*, 2006) was adopted.

$$0.75 \le \frac{\Delta t_P^{n+1}}{\Delta t_P^n} \le 1.25 \tag{14}$$

Therefore, if Δt_P^{n+1} calculated by Eq.(13) is out of the bounds determined by Eq.(14), the new implicit time-step is set: $\Delta t_P^{n+1} = 0.75\Delta t_P^n$ or $\Delta t_P^{n+1} = 1.25\Delta t_P^n$. In order to use this methodology in a "safer way", it is strongly recommended to check its effect on the accuracy of the numerical solutions in order to obtain the best accuracy/CPU time ratio.

Although, the elliptic pressure equation is not explicitly dependent on the time variable (t), differently from the saturation equation, it must be emphasized that Δt_P^n represents how long the pressure field will be held constant while the saturation field is calculated for successive time steps.

5. PARALLEL IMPLEMENTATION ISSUES

In this work, all calculations are performed using a parallel reservoir simulator written in C++, in which we have incorporated some open source packages, such as the FMDB, used for mesh management, the ParMetis, used for mesh partitioning and the PETSc used as a library of linear solvers and pre-conditioners.

5.1 Partition and load balance

Parallel simulations with unstructured meshes need a special attention to distribute nodes and elements among processes. The first step towards parallel efficiency is to guarantee that each partition receives the same number of nodes to avoid load unbalance which represents a case where a process works much more than others. The second step is related to the interface among processes for which the number of nodes with remote copies must be a minimum, in order to reduce the overhead caused by parallel communications.

5.2. Data structure for distributed meshes

The use of unstructured meshes by parallel simulators demands a "mesh manager" capable, for instance, to perform data migration among processors to satisfy the load balance. The open source library FMDB (Flexible Distributed Mesh Database) (Seol, 2005) is a parallel mesh manager written in C++ which uses the libraries ParMetis and Autopack for mesh partitioning and efficient parallel message passing, respectively.

5.3. Parallel iterative solvers and pre-conditioners

In the present work, the diffusion term given by Eq. (6) can be written in the following matrix form:

$$\left(EF\hat{p} + G\hat{p}\right) = A\hat{p} = b \tag{15}$$

where $\hat{p}$ is vector of the approximated pressure field, [G] is the matrix that comprises the contribution component of the flux parallel to the mesh edges and the product [E][F] contains the contributions of the fluxes orthogonal to the mesh edges. Based on the fact that solvers like GMRES or Conjugated Gradient performs their calculations using only matrix-vector products, we have applied a matrix-free scheme to overcome the high computational cost of performing

the addition operation between two matrices $(EF+G)\hat{p}$ which have different sparsity profiles. Throughout the simulation the matrices [E], [F] and [G] are assembled but the final matrix A is not. Then, to solve the system $A\hat{p}=b$, the product $A\hat{p}$, which can be easily implemented into the code using PETSc library, is reached by three steps using only two vectors:

$$\begin{aligned} z &= F\hat{p} \\ y &= G\hat{p} \\ y &= y + Ez \end{aligned} \tag{16}$$

As the final matrix is non-symmetric, the GMRES (Generalized Minimal RESidum) solver algorithm and the ASM (Additive Schwarz Method) pre-conditioner were used. The relative convergence tolerance and the number of iterations at which GMRES restarts adopted were 1e-6 and 30, respectively.

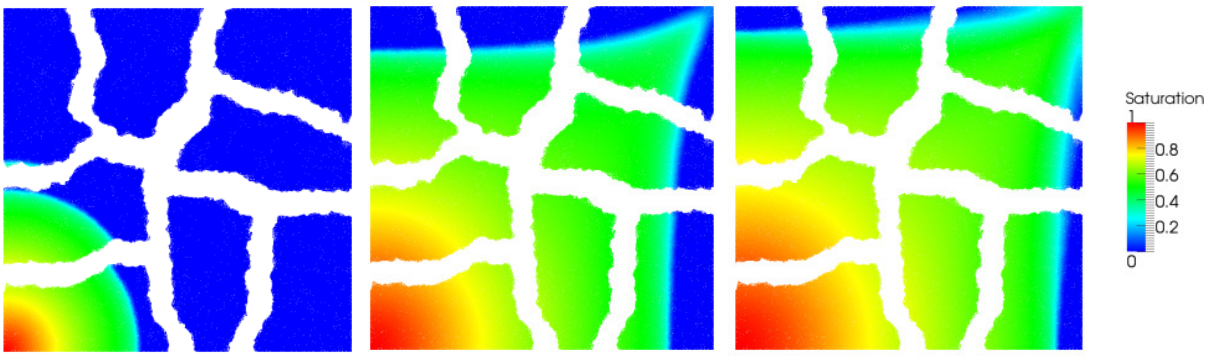
6. NUMERICAL RESULTS

The problem analyzed here, which was adapted from Durlofsky (1993), consists in a classical ¼ of five spot problem. The porous media is assumed to be homogeneous and isotropic with $K = \underset{\sim}{I}$ throughout the whole domain. We also assume that porosity is homogeneous even though its actual value is not relevant because we are only using it to define the non-dimensional time or PVI (Pore Volumes Injected) which is given by Eq. (17), as

$$PVI = \int \frac{Qdt}{V_p} \tag{17}$$

where V_P and Q are the total porous volume and the total flow rate, respectively.

Water and oil viscosities are, respectively, $\mu_w = 1.0$ and $\mu_o = 4.0$, therefore, the viscosity ratio (essentially the mobility ratio) is $M = (\mu_o/\mu_w) = 4.0$. Boundary conditions are: no-flux at all external boundaries, $S_i = 1.0$ in the injection well, and $P_{ul} = P_{br} = 0.0$ at the upper left and bottom right corners. A computational mesh with 45.459 nodes and 232.084 tetrahedral elements has been used to simulate the two phase oil-water flow for 1.0 PVI. Durlofsky (1993) solved this problem using a combined, mixed finite element (used to solve the pressure/velocity problem) and finite volume (used to solve the saturation problem) approach. Figures (1a) to (1c) present the saturation field for PVI=0.1, 0.45 and 0.6, respectively, using the described methodologies. The simulation was performed in parallel using 8 processors and the MIMPES approach with DVTOL = 0.05. All figures deliberately show the 8 mesh partitions.



(a) (b) (c)

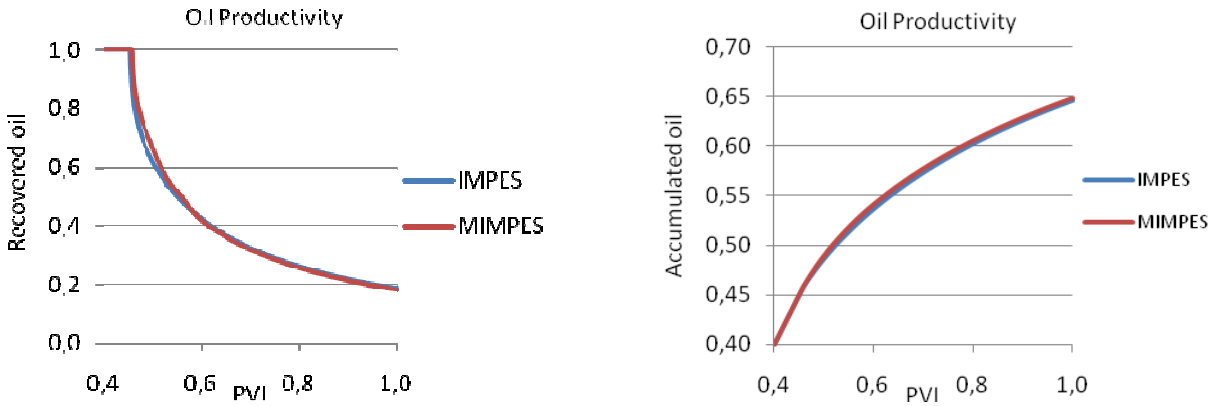
Figure 1. Saturation fields at different simulation times: 0.1 PVI (a), 0.45 PVI (b) e 0.6PVI (c). These results were obtained using the MIMPES approach for 8 processors.

Figures 2a and 2b show the recovered oil and accumulated oil, respectively for the IMPES and MIPES approaches. The x-axis for both figures varies from 0.4 to 1.0 PVI to highlight the accuracy of MIMPES compared to the IMPES approach. The recovered oil plot shows the relative oil flow through production well. In the beginning, the fluid flow production is 100% oil until the breakthrough (the instant in which the water phase reaches the production well). The second plot, Fig. 2b, shows the relative oil production related to the original oil in place, i.e, the quantity of oil exploited from the reservoir.

In Figure 3a, we present the IMPES and MIMPES approaches using one processor (i.e., one core) to show how fast the latter is compared to the former. Then, the MIMPES simulation is repeated using different numbers of cores (1, 2, 4, 8, 16 and 32) to obtain the speed-up curve showed in Fig. 3b.

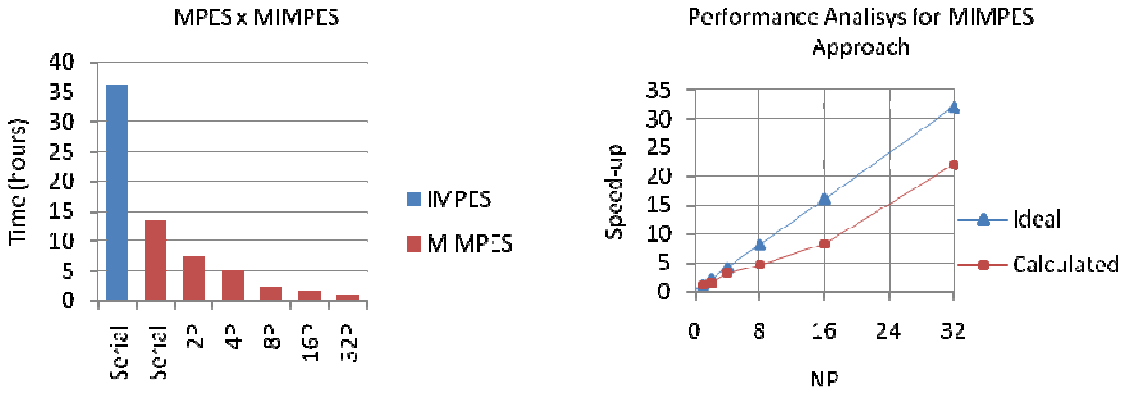
In Figure 3a, the total simulation time (in hours) of both approaches are compared. In the blue column, the classical IMPES, and, in the red columns, the MIMPES. With 32 cores, the MIMPES was 44 times faster than the IMPES with one core which took more than one day to complete the simulation while the former finished the same simulation in less than an hour.

The speed-up curve in Fig. 3b tells us how fast a parallel simulation is, when compared to the sequential running. In general, parallel communication overhead among processes increases as more cores are used, what leads the curve to depart from the ideal one.



(a) (b)

Figure 2. Oil productivity: recovered (a) and accumulated (b) oil analysis highlighting the water cutting moment.



(a) (b)

Figure 3. Comparison between simulations approaches (a): IMPES and MIMPES. The former is performed only sequentially and the later with 1, 2, 4, 8, 16 and 32 processors. Speed-up curve (b) showing performance of the parallel simulation with the MIMPES approach with 1, 2, 4, 8, 16 and 32 processors.

7. NUMERICAL ISSUES

The strategy of using a matrix-free scheme to avoid the extremely high computational cost of adding two matrices with different sparsity profiles can be too expensive for large problems due the excessive number of floating point operations of the scheme described in Eq. (16). Therefore, for large scale problems, a different approach has been adopted to achieve better CPU performance when solving the system of equations. Based on the good properties of matrix [G], which is symmetric, positive-defined and diagonal dominant, the solution is obtained by a deffect-correction, approach, as

$$[G]\hat{p}^{k+1} = q - [E][F]\hat{p}^{k} \tag{18}$$

The matrix of the system of equation is [G] and it can be solved by the Conjugated Gradient (CG) algorithm. Matrices [E] and [F] are part of the right hand side. For meshes with small aspect ratio and isotropic or mildly

anisotropic problems the system of equations is solved few times (2 to 7) to achieve convergence. But in general, for highly anisotropic porous media, or for meshes with high aspect ratio, about 20 iterations can be necessary. Another interesting feature of this iterative procedure is related to the fact that the product [E][F], in the right hand side, is performed just once until convergence is reached which reduces the number of floating point operations significantly.

Two examples simulating an elliptic homogeneous isotropic and an orthotropic 3-D ¼ of five-spot were chosen to evaluate the new approach and then compare the results with the matrix-free approach originally adopted. A 3-D mesh of tetrahedral with 259.662 degrees of freedom were used to discretize a domain with dimensions 1 x 1 x 0.05, with mobility ratio oil/water 4.0 and relative tolerance 1e-5 for CG solver using the Jacobi method as preconditioner. Permeability tensors K_1 and K_2 were applied for test 1 and 2, respectively.

$$K_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad K_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

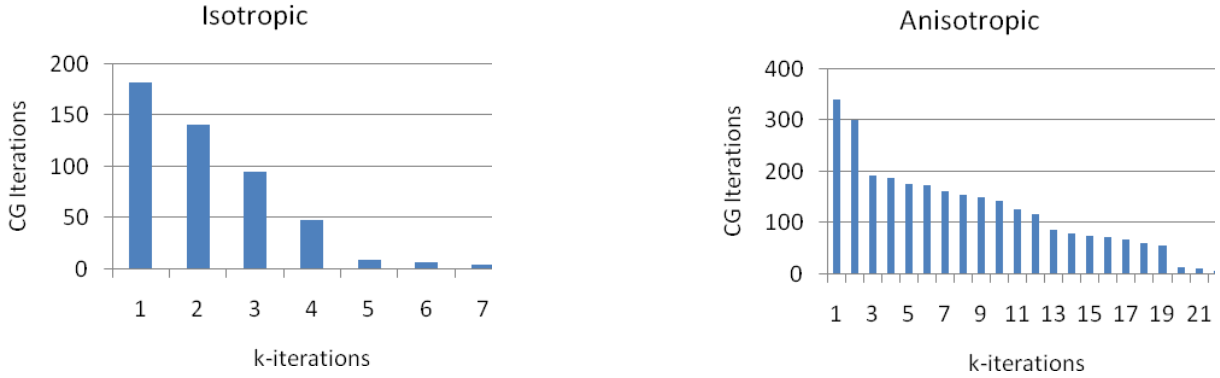
Results for the two solver schemes are showed in Tab. 1. The defect-correction scheme produced very acceptable results with a considerable reduction in CPU time for both tests with isotropic and anisotropic tensors when compared to the matrix-free scheme. However, the reduction decreases as the anisotropic effect increases as expected.

Table 1. Performance comparison between the matrix-free and the iterative defect-correction schemes. Both were tested with an isotropic (1) and an orthotropic (2) permeability tensors.

Test	Time (s)		Iterations	
	Matrix-free	Defect-correction	Matrix-free[(1)]	Defect-correction[(2)]
1	145	61	399	7
2	488	352	1372	22

[(1)]: GMRES iterations
[(2)]: *k*-iterations

For the isotropic tensor, Fig. 4a, solver convergence required only seven k iterations which means that the Conjugated Gradient solver has been performed 7 times. For the orthotropic tensor, Fig 4b, twenty two iterations were necessary to reach convergence. The convergence criteria adopted here relies on the residuum L_2 norm over solutions at iterations k-1 and k. The iterative procedure stops when the norm becomes less than a specified tolerance, here 1E-6.



(a) (b)

Figure 4. Conjugate Gradient behavior for each k-iteration of the defect-correction scheme for (a) test 1 with an isotropic tensor and (b) test 2 with an orthotropic tensor.

8. CONCLUSIONS

In the present paper, we have briefly presented a node-centered, edge-based, higher order finite volume method with a "Modified Implicit Pressure, Explicit Saturation" (MIMPES) approach using parallel computers with distributed memory capable to model the 3-D incompressible and immiscible two-phase flow of water and oil porous media. This approach produced very acceptable results with a considerable reduction in CPU time. The use of parallel computers turned this approach even more attractive. Two different approaches were used to solve the implicit pressure equation. Two tests were performed with isotropic and orthotropic tensors and, for these examples, the results showed that the

defect-correction approach was faster than the matrix-free approach which was initially adopted for the analyzed cases. In the near future, alternatively, we plan to try another implementation of the previous algorithms, exploiting the advantages of performing mat-vec products edge-by-edge.

9. ACKNOWLEDGEMENTS

The authors would like to thank the following agencies: "Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)", "Agência Nacional de Petróleo (ANP PRH-26)", the "Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), to "Núcleo de Atendimento em Computação de Alto Desempenho" (NACAD) COPPE/RJ, and to "Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE)", for their financial and infrastructure support during the development of this work.

10. REFERENCES

Carvalho, D. K. E., Silva, R. S., Lyra, P. R. M. and Willmersdorf, R. B., 2009a, "A 3-D edge-based higher order finite volume procedure for the simulation of two phase flow of oil and water in porous media using a modified IMPES approach.", 20th International Congress of Mechanical Engineering, Gramado-RS, Brazil.

Carvalho, D. K. E., Willmersdorf, R. B. and Lyra, P. R. M., 2009b, "Some results on the accuracy of an edge-based finite volume formulation for the solution of elliptic problems in non-homogeneous and non-isotropic media." Inter. J. for Num. Methods in Fluids DOI: 10.1002/fld.1948.

Chen, Z., Huan, G., Li, B., Li, W. and Espin, D., 2002, Comparison of Practical Approaches to Reservoir Simulation, 2nd Meeting on Reservoir Simulation, Buenos Aires, Argentina.

Crumpton, P. I., Moinier, P., Giles, M. B. T. J., 1997, An Unstructured Algorithm for High Reynolds Number Flows on Highly Stretched Grids, Numerical Methods in Laminar and Turbulent Flow, Taylor C and Cross J T Pineridge Press Swansea 561–572.

Durlofsky, L. J. A., 1993, Triangle Based Mixed Finite Element-Finite Volume Technique for Modeling Two Phase Flow Through Porous Media, J. of Comp. Phys., vol105, p252-266.

Ewing, R. E., 1983, The Mathematics of Reservoir Simulation, Siam, Philadelphia.

Helmig, R. and Huber, R., 1998, Comparison of Galerkin-Type Discretization Techniques for Two-Phase Flow Problems in Heterogeneous Porous Media, Advances in Water Resources, vol21, p697-711.

Hurtado, F. S. V., Cordazzo, J., Maliska, C. R. and Silva, A. F. C., 2006, Advanced Numerical Techniques for Improving Reservoir Simulation, Rio Oil and Gas.

Seol, E. S., 2005, FMDB: Flexible Distributed Mesh Database For Parallel Automated Adaptive Analysis, Rensselaer Polytechnic Institute, Troy, New York.

Luo, H., Baum, J. D. and Löhner, R., 1995, An Improved Finite Volume Scheme for Compressible Flows on Unstructured Grids, Technical Report, AIAA Paper, p95-0348.

11. RESPONSIBILITY NOTICE

The following text, properly adapted to the number of authors, must be included in the last section of the paper:
The author(s) is (are) the only responsible for the printed material included in this paper.