On the efficiency of the IMPES method for two phase flow problems in porous media

Carlos Redondo[a,b,1], Gonzalo Rubio[a,b,2], Eusebio Valero[a,b,3]

[a] *ETSIAE (School of Aeronautics and Aerospace Engineering), Universidad Politécnica de Madrid, Plaza de Cardenal Cisneros 3, 28040 Madrid, Spain*

[b] *Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain*

Abstract

In this work, we show a method to efficiently solve the multiphase flow through a porous media in the near wellbore region. The numerical discretisation is based on the IMplicit Pressure Explicit Saturation (IMPES) approach.

While most of the works in the literature rely on Fully Implicit Methods (FIM) to simulate the reservoir, this is not suited for the near wellbore region, where smaller computational and physical times are required, therefore parametric models with history matching are used in this zone. However, parametric models present several uncertainties that can affect the estimation of the pressure drop. An accurate and fast simulation of that region is therefore required.

Here, we propose a new and robust implementation of the IMPES scheme, aiming to reduce the large computational cost of the (implicit) pressure system of equations and increase the robustness and reliability of the scheme. The method takes advantage of the short physical time steps between iterations, observed in the near wellbore region, to produce a new pressure

[1] carlos.redondo@upm.es

[2] g.rubio@upm.es

[3] eusebio.valero@upm.es

field at the cost of a numerical explicit scheme. Two and three dimensional numerical experiments are carried out to show the validity of the proposed discretisation. We show that the cost of the method is reduced in a factor of 7 to 19 when compared to the classical IMPES.

Keywords: IMPES, near wellbore simulation, multiphase flow, porous media

1. Introduction

Simulation of multiphase fluid flow in porous media is currently a topic of interest in many areas such as hydrology and groundwater flow, oil and gas reservoir simulation or waste management. Numerical reservoir simulators are tools widely used by the oil industry and had become the main instrument for evaluating recovery efficiency and economical viability of new oil-drilling. While far-field reservoir areas are well-understood and fairly resolved with current simulation tools, near wellbore regions are normally under-resolved and become a source of inaccuracy in the resolution of the fluid physics in this zone [1, 2]. The under-resolved physics is modelled by including several empirical parameters, such as the *well skin factor* or the *productivity index* [2]. However, the logarithmic pressure drop in the surrounding wellbore region, as well as the complex thermodynamic behaviour of oil often leads to sudden variations in flow properties. These changes modify the production rates in a determinant way and can be hardly predicted by parametric wellbore models. Moreover, new techniques in oil drilling, including crosswise, horizontal wellbores and enhanced oil recovery techniques, increase the uncertainties associated to these models [3] making very difficult to accurately

predict the recovery efficiencies in the mid and long terms. In order to reduce these uncertainties, it is therefore crucial to accurately simulate this region in the most efficient way.

In terms of numerical simulation, the Fully Implicit Method (FIM) [4, 5, 6, 7, 8, 9, 10] and the IMplicit Pressure Explicit Saturation (IMPES) method [11, 12, 13, 14, 15, 16, 17, 18] are the main strategies to solve the system of partial differential equations arising from the discretisation of the mathematical model of the multiphase flow in porous media.

The FIM is regularly used in the simulation of the large far-field reservoir areas. Being unconditionally stable, it allows for selecting large time steps which are able to solve years of operation of complete reservoirs with dozens of producers and injectors in a few hours of computational time [19, 20]. However, when small time steps are required, e.g. to capture fast dynamic behaviour or coupling with other solvers, FIM method is not as computational advantageous. In these scenarios, and despite its limitations, the IMPES method is normally preferred. IMPES is an operator splitting technique [16] based on physics, which solves only the pressure equation implicitly, but updates the saturation explicitly. The pressure equation in this approach is formed by substituting the saturation constraint and Darcy's law into the sum of the mass conservation laws. The computational cost (time and memory) required by the IMPES method is smaller than the FIM at each time step. This advantage is more pronounced for problems with a high number of degrees of freedom and high velocity fluctuations, which induce shorter time steps. The explicit treatment of the saturation equation results in a stability limitation for the time step, especially for highly heterogeneous permeable

media. It should be noticed that this drawback is not so important if there is a physical constraint in the maximum time step, as it happens in the near wellbore region.

However, the IMPES method is not widely used in industrial problems due to the high computational cost of solving the implicit pressure equation, which roughly represents the 90 percent of the solver [14]. Therefore, as already mentioned, modelling that region is the preferred option. To circumvent this problem, Chen, Huan and Li [13] have proposed an improved IMPES method where the implicit pressure equation is only solved after the variation in the saturation exceeds a certain threshold. The main drawback of this method is a lack of accuracy resulting from updating the saturations with incorrect pressure values, which eventually gives rise to nonphysical numerical oscillations of the pressure. This idea has been further extended and improved in the last years [14, 21, 18].

In this work, we propose a novel methodology to improve the performance of the IMPES method by reducing the computational cost of the pressure equation solver. A combination of iterative solver, preconditioner, initial condition and stopping criterion is proposed and analysed in multiple scenarios. Although iterative solvers have been used to solve the IMPES problem before [14], to the authors known no attention has been paid to the effect of the preconditioner, the initial condition or the stopping criterion in the efficiency and accuracy of the solution. From an industrial point of view, the iterative solver strategy is only practical if it is not problem dependent and it does not require parameter tuning. In this work, a non problem dependent iterative

solver strategy is introduced. This strategy drastically reduces the computational cost of the pressure equation solver to the same order of magnitude of the saturation solver, eventually giving rise to a total reduction of the computational time in an order of magnitude while retaining enough accuracy in the solution for most of the applications. The method developed is referred here as iterative IMPES, while the IMPES using direct solver for pressure equation will be denoted as conventional IMPES.

The present paper is organised as follows: First in Section 2 the mathematical model of the physical problem is introduced. Then in Section 3 numerical models are discussed. Special attention is paid to the iterative solver for the pressure equation. In Section 4, five validation test cases are detailed. These simulations are used to analyse the effect of the iterative solver tolerance in both the solution and the efficiency. Lastly in Section 5, overall conclusions for this work are drawn.

2. Mathematical model

The flow of two incompressible immiscible fluids through porous media can be modelled by the saturation equation and the Darcy's law for each phase α. The saturation equation for the phase α is given by:

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \boldsymbol{u}_\alpha = q_\alpha, \qquad \alpha = \text{n,w}, \tag{1}$$

where the subscripts w and n denote the wetting and nonwetting phases. S, $\boldsymbol{u}$ and q are the saturation, velocity and volumetric flux of the phase α and ϕ represents the porosity of the media. The velocity of each phase is given

by the Darcy's law:

$$\boldsymbol{u}_\alpha = -\frac{k_{r\alpha}\mathbf{K}}{\mu_\alpha}\left(\nabla p_\alpha - \rho_\alpha \boldsymbol{g}\right), \qquad \alpha = \text{n,w}, \tag{2}$$

where k_r, μ, p and ρ are the relative permeability, the viscosity, the pressure and the density for the phase α. $\boldsymbol{g}$ is the gravity vector and $\mathbf{K}$ is the absolute permeability tensor. In this work $\mathbf{K}$ is assumed to be diagonal and isotropic, so $\mathbf{K} = k\mathbf{I}$ where $\mathbf{I}$ is the identity matrix.

For two phase immiscible flows, the pressure of the two phases is related through the capillary pressure:

$$p_c(S_w) = p_n - p_w. \tag{3}$$

The saturations of the two phases are constrained by the relation:

$$S_w + S_n = 1. \tag{4}$$

Relative permeabilities, k_r, and capillary pressure, p_c, are modelled using semiempirical functions. For the relative permeability the modified Brooks-Corey relation [22] is used. This model is based on the effective saturation, which is defined as:

$$S_{eff} = \frac{S_w - S_{w,r}}{1 - S_{n,r} - S_{w,r}}, \tag{5}$$

where $S_{w,r}$ and $S_{n,r}$ are the residual or irreducible (or connate for water phase) saturations. The expressions for the relative permeabilities are:

$$\begin{aligned} k_{rn}(S_{eff}) &= k_{rn,max}(1 - S_{eff})^n, \\ k_{rw}(S_{eff}) &= k_{rw,max} S_{eff}^m, \end{aligned} \tag{6}$$

where $k_{rn,max}$ and $k_{rw,max}$ are the relative permeabilities for the *only one flowing phase* limits. Exponents n and m are semi empirical parameters that

depend on the nature of the porous media. Capillary pressure is modelled using a relation that depends on absolute permeability:

$$p_c(S_w) = -\frac{B_c}{\sqrt{k}} \log(S_{eff}), \tag{7}$$

where B_c is a positive parameter related to the porous media and k is the absolute permeability [23].

For further simplification, we define the mobility of the phase α as:

$$\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}, \qquad \alpha = \text{n,w}. \tag{8}$$

An elliptic equation for the pressure is obtained by combining equations (1), (2) and (4):

$$-\nabla \cdot [\mathbf{K}\lambda_w (\nabla p_w - \rho_w \boldsymbol{g}) + \mathbf{K}\lambda_n (\nabla p_n - \rho_n \boldsymbol{g})] = q, \tag{9}$$

where q is the total source flux $q = q_w + q_n$.

Finally, introducing the capillary pressure equation (3) into (9) the problem can be reduced to a system in two primary variables S_w and p_n:

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot \boldsymbol{u}_w = q_w, \tag{10}$$

$$-\nabla \cdot [\mathbf{K} (\lambda_w + \lambda_n) \nabla p_n] = \nabla \cdot [\mathbf{K} (\lambda_w \rho_w + \lambda_n \rho_n) \boldsymbol{g}] + \nabla \cdot (\mathbf{K}\lambda_w \nabla p_c) - q. \tag{11}$$

The system (10) – (11) is fully coupled through Darcy velocity, $\boldsymbol{u_w}$, that depends on the pressure, and mobilities, λ_α, which are functions of the saturation.

3. Numerical discretisation

In this work, the system of equations in (10) – (11) will be integrated with an implicit explicit (IMEX) method where a splitting approach based

on physics is used. This method, usually known in the literature as IMPES, solves the pressure equation (11) implicitly and updates the saturation equation (10) explicitly. A second order finite difference scheme will be used for the spatial discretisation of (10) – (11) while a Runge-Kutta 2 (RK2) method will be used for the temporal evolution of (10).

The space is discretised by means of a second order finite difference approach on a cartesian staggered grid. Scalar properties such as pressure or saturation are evaluated in cell centres, while vectorial magnitudes such as velocities, pressure gradients or mobilities are evaluated in the midpoints between cell centres. In Figure 1, the complete stencil for the 2d case is shown. It can be seen that x-components of vector magnitudes are evaluated at *x-faces* (white triangles), while y-components are evaluated at *y-faces* (black triangles).

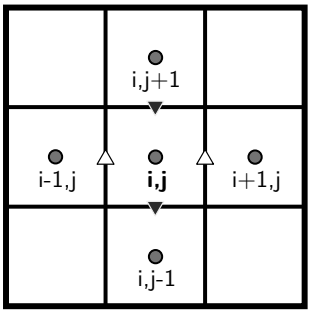

Figure 1: 2d 5-point stencil. White triangles mark x-faces, where x-component of vector magnitudes are evaluated. Black triangles show y-faces where y-component of vector magnitudes are evaluated.

Equations in system (10) – (11) are expressed in divergence form. To numerically solve the system, we define the discretised divergence operator

on a mesh point $\boldsymbol{x}_{i,j} = (x_{i,j}, y_{i,j})$, at a given time t^n as follows:

$$(\nabla \cdot \boldsymbol{F})^n_{i,j} = \frac{F^n_{i+1/2,j} - F^n_{i-1/2,j}}{(x_{i+1,j} - x_{i-1,j})/2} + \frac{F^n_{i,j+1/2} - F^n_{i,j-1/2}}{(y_{i,j+1} - y_{i,j-1})/2}, \tag{12}$$

where F^n are magnitudes evaluated on the faces, for example, $F^n_{i+1/2,j}$ corresponds to the x-component of the vectorial magnitude $\boldsymbol{F}$ evaluated on the x-face between $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{i+1,j}$. In order to apply the discretisation in (12) to the system (10) – (11), the involved magnitudes must be computed at the grid faces. The expressions for the computation on the face between $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{i+1,j}$ are detailed below, where j subscripts are dropped for clarity.

Absolute permeability tensor is assumed to be "uniform-diagonal" (i.e., only elements corresponding to pairwise equal indices are non-zero and constant, $\mathbf{K}_{i,j} = k_{i,j}\mathbf{I}$). The evaluation of this magnitude on a face between two points, also known as hydraulic conductivity, is computed using the harmonic mean:

$$k_{i+1/2} = \frac{2k_i k_{i+1}}{k_i + k_{i+1}}. \tag{13}$$

Pressure gradients are computed in the faces using finite differences:

$$\begin{aligned} \nabla p_n|_{i+1/2} &= \frac{p_n|_{i+1} - p_n|_i}{x_{i+1} - x_i}, \\ \nabla p_c|_{i+1/2} &= \frac{p_c|_{i+1} - p_c|_i}{x_{i+1} - x_i}. \end{aligned} \tag{14}$$

Mobilities, that depend on the advected saturation, are computed using a Total Variation Diminishing (TVD) upwind scheme:

$$\lambda_\alpha|_{i+1/2} = \lambda_\alpha|_i - \varphi\left(\lambda_\alpha|_i - \frac{\lambda_\alpha|_i + \lambda_\alpha|_{i+1}}{2}\right) \quad , \text{if } \boldsymbol{u}_\alpha > 0 \tag{15}$$

$$\lambda_\alpha|_{i+1/2} = \lambda_\alpha|_{i+1} - \varphi\left(\lambda_\alpha|_{i+1} - \frac{\lambda_\alpha|_i + \lambda_\alpha|_{i+1}}{2}\right) \quad , \text{if } \boldsymbol{u}_\alpha < 0 \tag{16}$$

$$\lambda_\alpha|_{i+1/2} = \frac{\lambda_\alpha|_i + \lambda_\alpha|_{i+1}}{2} \quad , \text{if } \boldsymbol{u}_\alpha = 0 \tag{17}$$

where φ is the second order van Leer TVD limiter [24] based on the saturation. Finally, velocities are computed using the variables previously evaluated on the faces:

$$u_w|_{i+1/2} = -\lambda_w|_{i+1/2} \left(\nabla p_n|_{i+1/2} - g_x \rho_w - \nabla p_c|_{i+1/2} \right), \tag{18}$$

$$u_n|_{i+1/2} = -\lambda_n|_{i+1/2} \left(\nabla p_n|_{i+1/2} - g_x \rho_n \right), \tag{19}$$

where g_x is the x-component of the gravity vector $\boldsymbol{g}$.

Introducing discretisation (12)–(17) into (11), pressure equation is transformed into a system of linear equations, with the non–wetting pressure values on each point as unknowns. The system can be expressed in matrix form:

$$A\boldsymbol{p_n} = \boldsymbol{b}, \tag{20}$$

where A is the system matrix, that contains the total transmissibilities between adjacent cells (LHS of equation (11)), $\boldsymbol{p_n}$ is the vector with the values of non–wetting pressure $p_n|_{i,j}$ in all the points of the domain, and $\boldsymbol{b}$ contains the terms related to flux sources, gravity and capillary pressure (RHS of equation (11)).

Saturation equation (10) is updated in time with a RK2 scheme. The discretised algorithm, after introducing (18) in (10), is listed below:

Data: Fluid and porous media properties.

initialization of S_w^0, $\boldsymbol{u}_w^0$;

for $n = 0$ **to** *Number of iterations* **do**

Solve $A^n \boldsymbol{p}_w^n = \boldsymbol{b}^n$ and compute $\boldsymbol{u}_w^n$ with (18)

Compute Δt with (23)

$$S_{w;i,j}^{n+1/2} = S_{w;,i,j}^n + \frac{\Delta t}{2\phi}\left(q_{w;i,j}^n - \left(\frac{u_{w;i+1/2,j}^n - u_{w;i-1/2,j}^n}{(x_{i+1,j} - x_{i-1,j})/2} + \frac{u_{w;i,j+1/2}^n - u_{w;i,j-1/2}^n}{(y_{i,j+1} - y_{i,j-1})/2}\right)\right)$$

Solve $A^{n+1/2} \boldsymbol{p}_w^{n+1/2} = \boldsymbol{b}^{n+1/2}$ and compute $\boldsymbol{u}_w^{n+1/2}$ with (18)

$$S_{w;i,j}^{n+1} = S_{w;,i,j}^n + \frac{\Delta t}{\phi}\left(q_{w;i,j}^n - \left(\frac{u_{w;i+1/2,j}^{n+1/2} - u_{w;i-1/2,j}^{n+1/2}}{(x_{i+1,j} - x_{i-1,j})/2} + \frac{u_{w;i,j+1/2}^{n+1/2} - u_{w;i,j-1/2}^{n+1/2}}{(y_{i,j+1} - y_{i,j-1})/2}\right)\right)$$

$t^{n+1} = t^n + \Delta t$

end

Algorithm 1: IMPES algorithm

As can be seen, pressure equation (11) has to be solved twice each time step. The system of linear equations resulting from the spatial discretisation is solved using an iterative preconditioned Generalised Minimal RESidual (GMRES) solver [25]. This strategy reduces drastically the computational time required for the implicit part of the IMPES algorithm compared to direct solvers. Iterative pressure solver is discussed in detail in Section 3.1.

Time step is computed using the stability criteria from [26, 27] and imposing a maximum variation in saturation ΔS_{max} per time step. The stability criteria for a two dimensional problem reads:

$$\Delta t_{\mathrm{CFL}} = \mathrm{CFL} \cdot \frac{\phi \Delta x \Delta y}{2\Psi p_c' \sum T + 4 f_w' \left(|q_x| + |q_y|\right)}, \tag{21}$$

where $\Psi = \lambda_w \lambda_n / (\lambda_n + \lambda_w)$, $p_c' = \mathrm{d}p_c / \mathrm{d}S_w$, $\sum T$ represents the total transmissibilities defined as: $\sum T = 2k\Delta y/\Delta x + 2k\Delta x/\Delta y$, $f_w' = \mathrm{d}f_w/\mathrm{d}S_w$ is the derivative of the fractional flow $f_w = \lambda_w/(\lambda_w + \lambda_n)$ and $q_x = \Delta y(\boldsymbol{u}_{x,w} + \boldsymbol{u}_{x,n})$,

$q_y = \Delta x(\boldsymbol{u}_{y,w} + \boldsymbol{u}_{y,n})$.

The solution is stable and free of oscillations for CFL < 1 [27]. The additional time step restriction is defined as:

$$\Delta t_{\Delta S_{max}} = \frac{\Delta S_{max}\phi}{q_w - \nabla \cdot \boldsymbol{u}_w}. \tag{22}$$

Iteration time step is the minimum value among all the points in the domain for both criteria:

$$\Delta t = \min\left(\Delta t_{\mathrm{CFL}}, \Delta t_{\Delta S_{max}}\right). \tag{23}$$

A value for the maximum variation of saturation per iteration of $\Delta S_{max} = 0.1$ is imposed to avoid instabilities in the firsts iterations. After that, the CFL condition is more restrictive and it is enough to guarantee stability.

It should be mentioned that Algorithm 1 does not include any artificial limit for the saturation, even for test cases with saturation injection of $S_w = 1$ or $S_n = 1$ and residual saturations $S_{r,w} = S_{r,n} = 0$. These artificial limitations have been avoided as they usually result in errors in phase conservation [10]. The discretisation used in this work conserves the volume of each phase, which is crucial on long runs in reservoir simulation.

Finally, as far as the boundary conditions are concerned, no flow condition is imposed on all boundary faces. Additionally, flux or pressure conditions are introduced in well form using Peaceman model [2].

3.1. Iterative pressure solver

As it was stated in the previous section, the IMPES strategy for solving system (10) – (11) requires two resolutions of a sparse linear system of equations (20), associated to the discretisation of the pressure, per iteration.

The resolution of this system is the most computational expensive part of Algorithm 1, so an efficient method to perform this operation is essential. This system can be solved using direct methods, which are robust and compute the solution of the system up to machine precision. However, they are expensive from the point of view of computational time and present high memory requirements. Iterative solvers, which approximate the solution of the system starting from an initial guess in a iterative way, are the alternative to direct ones. The iterative nature of these methods make them capable of approximating the solution up to a prescribed tolerance. Iterative methods are much less robust than direct ones and their efficiency relies on the initial guess and the preconditioning that, in general, are problem dependent. For a through review of iterative methods see [28].

The matrix of the pressure equation (20) is numerically non–symmetric and typically ill conditioned, for these reasons, a preconditioned GMRES method is used in this work for solving the pressure equation.

The iterative solver strategy is only practical if is not problem dependent and requires no parameters tuning. In this section, the initial condition, the preconditioner and the stopping criterion for the GMRES are detailed.

3.1.1. Initial guess

The explicit time step restriction imposed by the stability criteria is exploited to improve the performance of the GMRES method. In particular a small time step means that the pressure variation between consecutive time steps is also small, therefore a high-quality initial guess is available beforehand. To further exploit this fact, the initial guess for the iterative solver is

computed by linear extrapolation of the two previous pressure solutions.

3.1.2. Stopping criterion

The tolerance in pressure equation (20) can be used as stopping criterion but its relationship with the error in saturation distribution is problem dependent. Instead, the error in total velocity per time step is used as problem independent control parameter,

$$\mathrm{DIV}_{\mathrm{tol}} = \max\left[\nabla \cdot \left(\boldsymbol{u_w}|_{i,j} + \boldsymbol{u_n}|_{i,j}\right) + q_w|_{i,j} + q_n|_{i,j}\right] \Delta t. \tag{24}$$

This tolerance can be computed with negligible cost every time step but it is costly to compute at every GMRES iteration. To circumvent this shortcoming, the GMRES residual (norm of the last vector in the GMRES Arnoldi iteration) is used. This parameter is problem dependent but shows the same convergence behaviour as the error in total velocity as shown in Figure 2. This tolerance, GMRES_{tol}, is set in the time step $k+1$ from their value in the previous time step, k, using the expression:

$$\mathrm{GMRES}_{tol}^{k+1} = \mathrm{GMRES}_{tol}^{k} \frac{\tau}{\mathrm{DIV}_{\mathrm{tol}}^{k}}, \tag{25}$$

where τ is the desired tolerance in divergence of total velocity.

As can be observed in Figure 2, the factor between the GMRES residual and the error in total velocity divergence may also vary in the course of a simulation however the proposed stopping criterion is able to track the desired tolerance accurately.

Equation (24) is exactly zero if a direct solver is used for the pressure equation. On the contrary, if it is not exactly zero, an error of order $\mathrm{DIV}_{\mathrm{tol}}$ acts as a source term in Equation (10). Prescribing a tolerance of the same

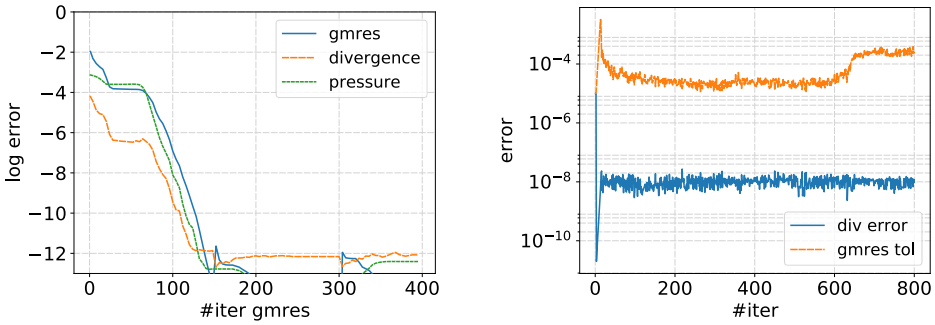

Figure 2: Left: example of evolution of GMRES residual (gmres), error in divergence of total velocity (divergence) and error in pressure equation (pressure) with the number of GMRES iterations. Right: example of evolution of GMRES tolerance and error in total velocity divergence using (25) for a target tolerance of 10^{-8}.

order of magnitude as the discretisation error, lead to solutions indistinguishable from the ones obtained with direct methods, but with a reduced cost.

3.1.3. Preconditioner

The preconditioner is a key aspect when dealing with iterative solvers. On the one hand, advanced preconditioners such as ILUT or more sophisticated such as algebraic multigrid show good performance reducing the number of GMRES iterations but at high computational cost. On the other hand, other preconditioners such as Jacobi or ILU(0) are less efficient accelerating GMRES convergence but their computation and application is cheaper. As a result, each problem should be analysed in detail to get a proper balance between iteration cost and number of iterations. It should be noticed that a good initial guess plus a relaxed prescribed tolerance reduce the benefits of an expensive preconditioner, as the solution might be converged with a small

number of iterations, even with a cheap one.

In Figure 3 the convergence of the GMRES is analysed. Three preconditioners are compared: ILU(0), ILUT(4) and ILUT(8). ILU(0) uses an incomplete LU factorization with the same sparsity pattern as the matrix A. ILUT(p) uses an incomplete LU factorization retaining only the p values with higher magnitude on each row in L and U. As expected, more advanced preconditioners achieve better convergence rates in term of number of iterations but for moderate tolerances (10^{-7} in Figure 3), a cheap preconditioner such as ILU0 performs better from the point of view of simulation time. This behaviour is similar in all the analysed cases. The importance of this result will be emphasised in the following section, where it will be shown that moderate tolerances ($\sim 10^{-6}$) are adequate to obtain solutions accurate enough in most of the scenarios.

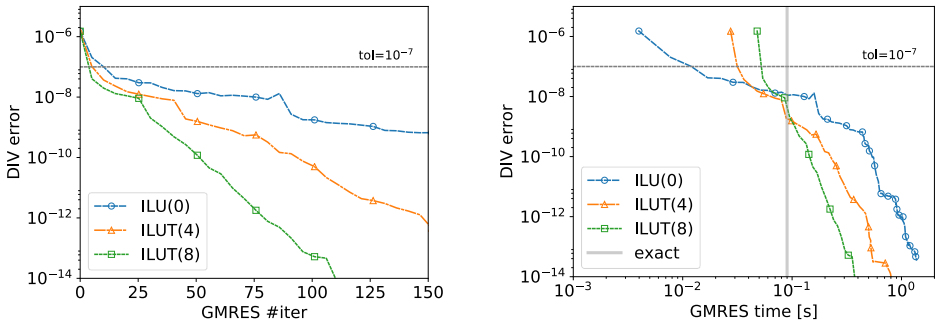

Figure 3: Convergence of error in divergence of total velocity function of the number of GMRES iterations (left) and computational time (right). The impact of the preconditioner is shown for ILU(0), ILUT(4) and ILUT(8). The tolerance of the previous time step is set up to 10^{-7}. Computational time needed by the direct solver(exact) is shown on the right plot. Markers are plotted every 25 iterations.

4. Numerical experiments

In this section numerical experiments to validate and test the performance of the developed discretisation technique for multiphase flow in porous media are carried out. The impact of the tolerance, defined in Equation (24), in the solution accuracy is analysed in a set of problems of interest, next the impact of the tolerance in the computational time is presented. For this purpose, we define the efficiency of the iterative solver strategy as:

$$\text{efficiency} = \frac{\text{Simulation time using direct solver}}{\text{Simulation time using GMRES}} \tag{26}$$

The solver is coded in Fortran and compiled with Intel Fortran Compiler 2016. The code is parallelised with OpenMP. Intel-MKL Pardiso is used for the direct solver and an implementation based on Intel-MKL RCI FGMRES interface is used for the iterative linear solver. Experiments have been carried out on an workstation with a 4-core Intel(R) i7-4790 processor.

4.1. Test cases

In this section, four test cases are considered. These test cases have been selected because they represent a wide spectrum of physical problems of interest. In particular, special attention is paid to gravity, capillary pressure and localised high flow velocity effects (both geometrically and permeability induced).

Case 1 is extracted from the SPE10 dataset 1, where the effect of the gravity can be measured [29]. Case 2 is a banded domain from [23] with capillary pressure. Case 3 is a five-spot simulation on a filtered random

domain. Case 4 is a 3d injection problem in a realistic permeability field. Details of the cases are shown in Table 1.

Parameters	Case 1	Case 2	Case 3	Case 4
Domain (m)	762×15.24	500×270	100×100	$100 \times 100 \times 23$
Resolution (points)	100×20	200×100	100×100	$75 \times 75 \times 30$
ϕ	0.2	0.2	0.2	0.0159 – 0.1872
k (md)	10^{-3} – 10^{3}	1 – 100	10^{-2} – 10^{3}	0.12 – 3731.82
μ_w (Pa·s)	10^{-3}	$4.5 \cdot 10^{-4}$	10^{-3}	10^{-4}
μ_n (Pa·s)	10^{-5}	10^{-3}	10^{-2}	10^{-5}
ρ_w (kg·m^{-3})	700	1000	1000	1000
ρ_n (kg·m^{-3})	1	750	750	100
n	5	2	2	2
m	5	2	2	2
B_c (bar md$^{1/2}$)	0	50	0	0
gravity (m·s^{-2})	9.81	0	0	9.81

Table 1: Parameters for validation cases

4.1.1. Case 1

This case is part of the SPE10 Comparative Solution Project. The dataset 1 consists of a 2d two phase (oil-gas) model with a simple 2d vertical section geometry [29, 30]. Gas is injected from an injector located on the left of the model and dead oil is produced from a well on the right of the model. Relative permeability parameters and fluid properties are taken from [29] and listed in Table 1. Residual saturation for gas phase is $S_{n,r} = 0.1$ and for oil $S_{w,r} = 0.25$. Initially, the media is fully saturated with oil. The permeability

distribution for this case, shown in Figure 4, presents horizontal bands with high permeability. These zones are associated with high flow velocities that will impose small time steps to maintain stability. In the following section the effect of these heterogeneities on the performance of the method is analysed. The solution show a good agreement with the results shown in [29].

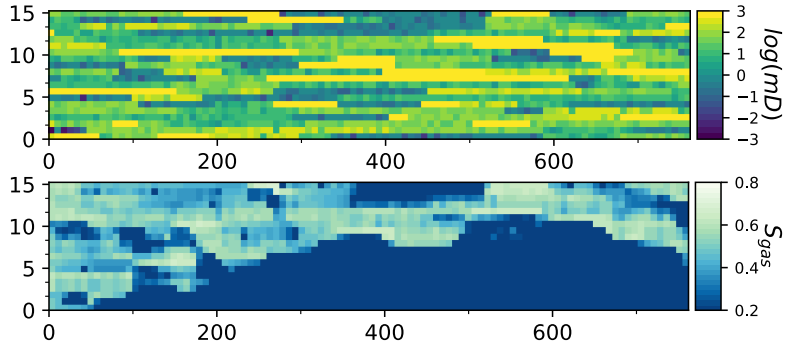

Figure 4: Case 1. Top: Absolute permeability field. Bottom: gas saturation distribution with gravity effects. Solution after 3 years with uniform gas injection rate at $6.6 \cdot 10^{-7}$ m/s on the left boundary.

In Figure 5 the impact of the solver tolerance in the saturation distribution is shown. It can be observed that a tolerance of 10^{-6} is enough to obtain a valid approximation for most of the applications. Solver tolerance has also an impact in the production history but as can be drawn form Figure 6 a tolerance of 10^{-6} is enough to obtain a production curve similar to the one obtained using the direct solver for the pressure equation.

4.1.2. Case 2

In this case [23] [10] we consider a 2d heterogeneous domain composed of horizontal layers with alternate absolute permeability oriented along the main

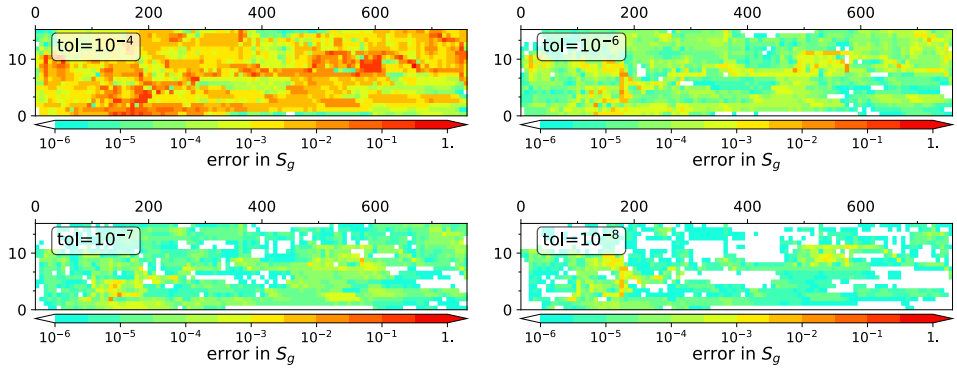

Figure 5: Error in gas saturation for case 1 with gravity effects for different tolerance values.

flow direction. Wetting phase is injected uniformly across the left boundary at a rate of 0.11 PVI/year. The domain is initially saturated with the non wetting phase. Values for residual saturation are $S_{r,w} = S_{r,n} = 0$. Saturation profile for the wetting phase, at a PVI=0.5, is shown in Figure 7. Important flow features, such as wetting phase accumulation in permeability jump interfaces when permeability dependent capillary pressure is used, are accurately captured.

In Figure 8 the impact of the tolerance in the saturation distribution is shown. At high tolerances, $\sim 10^{-4}$, the error in saturation is of the same order of saturation itself but the solution is stable and important features as front positions are accurately solved. A tolerance of $\sim 10^{-6}$ is enough to obtain maximal error values of order 10^{-1}.

4.1.3. Case 3

This case is a five spot simulation on a 2d squared domain. Permeability distribution has been computed with an in-house random porous media gen-

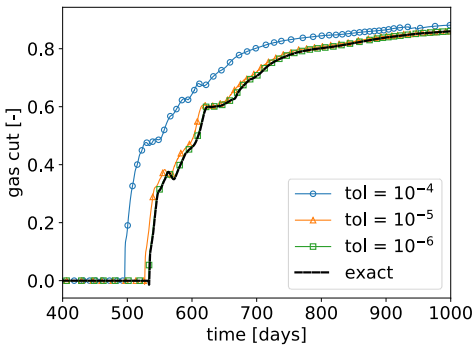

Figure 6: Case 1. Impact of iterative solver tolerance on the gas-cut history in the producer located on the right boundary of the domain

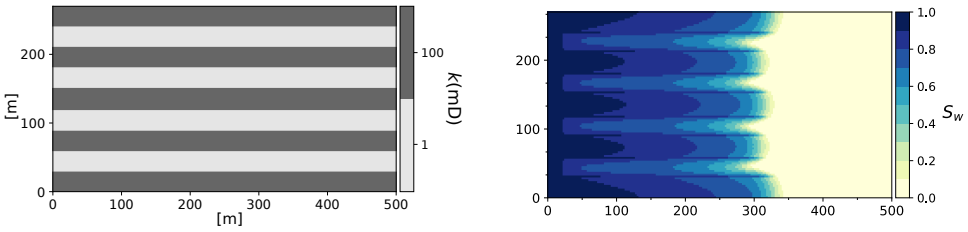

Figure 7: Case 2. Left: Absolute permeability distribution. Right: Wetting phase saturation distribution after 0.5 PVI. Domain size of 500×270 m×m on a 200×100 mesh.

erator. Wells are surrounded by permeable zones to avoid high unphysical pressures. Water is injected from the four corners of the domain, which is initially saturated with oil. Values for residual saturation are $S_{r,w} = S_{r,n} = 0.1$. Flow is produced from a well in the centre of the domain. Permeability distribution is shown in Figure 9. Injection rate is fixed for each injector to 10^{-5} m^2/s.

This test case has been included because the radial flow configuration is of high interest for oil industry and it is usually a challenge for explicit

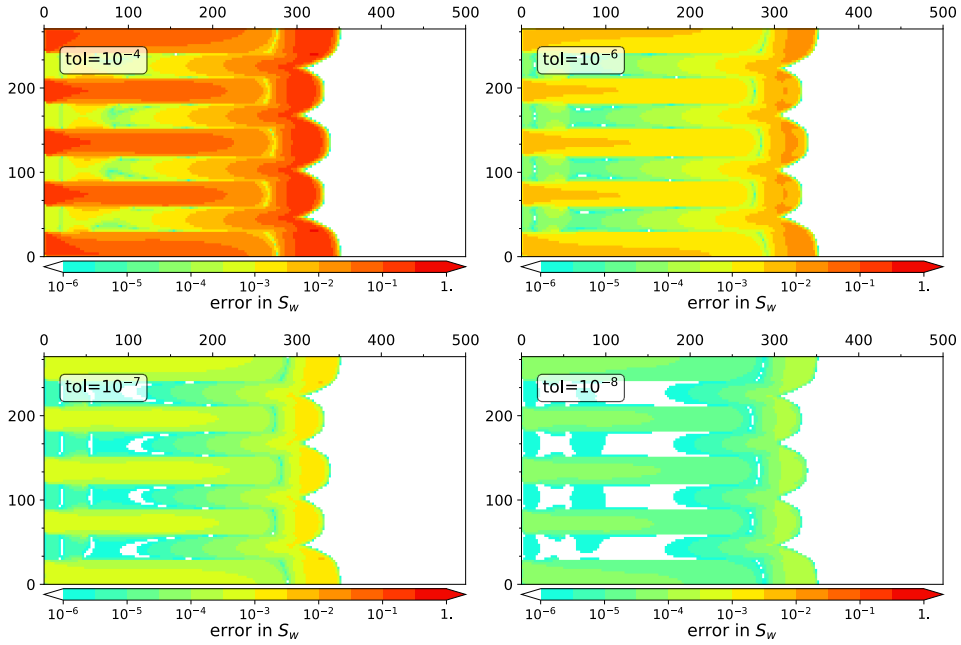

Figure 8: Error in saturation for case 2 for different tolerance values.

methods, as small time steps are required to maintain stability due to the high velocity regions.

In this case, due to the viscosity ratio of the fluids involved, fingering patterns are formed as can be observed in Figure 9. These structures, associated to the unstable interface between the fluids, are very sensitive to high tolerances as sown in Figure 10. For tolerances above 10^{-5} fingering patterns are not correctly captured and the flow configuration is wrongly predicted. This effect can be also observed in the production curve in Figure 11 where the production curves for tolerances 10^{-4} and 10^{-5} differ notably from the exact solver curve. Again, as observed in the previous cases, a tolerance value of 10^{-6} is enough to produce a solution qualitatively similar to the solution

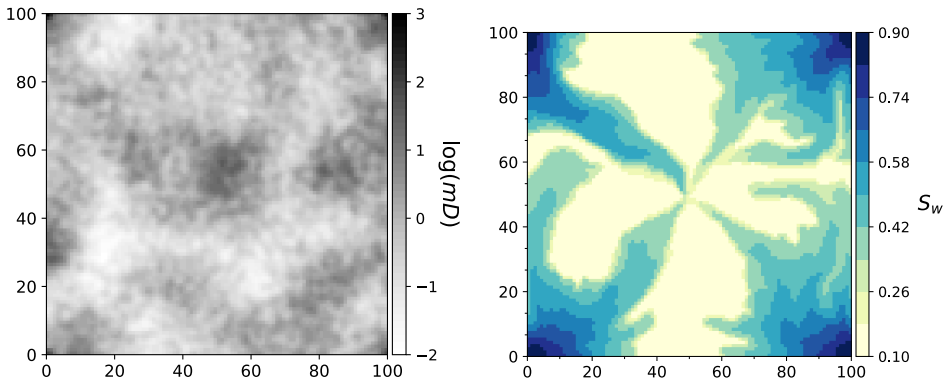

Figure 9: Case 3. Left: Absolute permeability distribution. Right: Wetting phase distribution at water breakthrough for the four injector wells. Domain size of 100×100 m×m with 100×100 mesh.

computed with the exact pressure solver. It should be mentioned that for the entire range of tolerances analysed, no oscillations are observed in flux production when CFL < 1 is used. The appearance of oscillations is reported in [13] when ΔS_{max} criterion is used for computing the time step.

4.1.4. Case 4

Here the developed technique for the efficient resolution of the IMPES method in a problem of industrial scale. The results are obtained with a three dimensional version of the scheme introduced in section 3. The objective is twofold: first to show the validity of the scheme to solve a problem of industrial scale and second to show the efficiency of the scheme for a *real size* problem.

In particular, we simulate the near wellbore region of an injection well situated at the center of a domain of 100m×100m. As far as the depth is concerned, the injection zone is located at $4 - 14$m while the total depth of

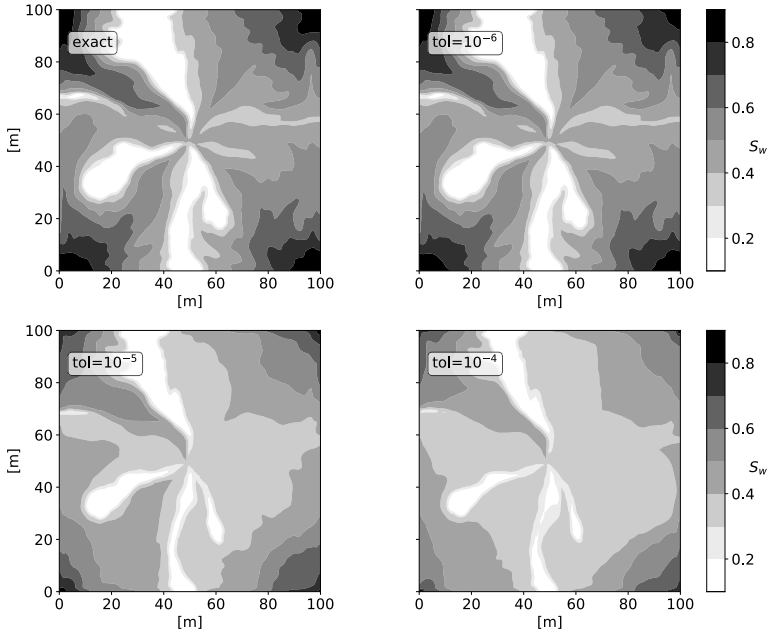

Figure 10: Case 3. Impact of solver tolerance in the saturation distribution after a year of injection.

the domain is 23 m. The porosity and permeability maps can be seen in Figures 12-13. It should be noticed that the high permeability zone situated at the bottom of the injection zone is behind the injection well. This fact will have an important effect on the results.

Initially, the domain is fully saturated with the non wetting, lighter phase while the wetting, heavier phase is injected through the well . The properties of the domain and the fluid properties are detailed in Table 1. The residual saturation of both phases is set to zero ($S_{rw} = S_{rn} = 0$).

As far as the boundary conditions are concerned, the pressure at the top of the reservoir and at the top of the well are set to 320 bar and 320.1

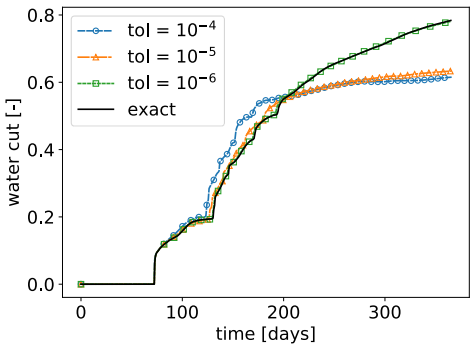

Figure 11: Case 3. Impact of the solver tolerance on the production history.

bar respectively. The hydrostatic pressure is taken into account both at the reservoir and at the well.

The saturation contour of the wetting phase after 10 days of simulation can be seen in Figure 14. There is a strong effect of the gravity on the distribution of saturation, as expected. It should be noticed that, although no wetting phase is injected in the high permeability zone at the bottom of the domain, it arrives there due to the gravity effects. The wetting phase accumulates at this zone as it can not cross a non permeable wall located at the bottom of the domain. Results shown in Figure 14 have been obtained with the iterative IMPES method with a tolerance of 10^{-6}.

In Figure 15 an error plot is shown. The error is calculated in total flow of wetting phase injected at day 10 and it is defined as the difference between the one obtained with the classical IMPES and the iterative IMPES divided by the classical IMPES.

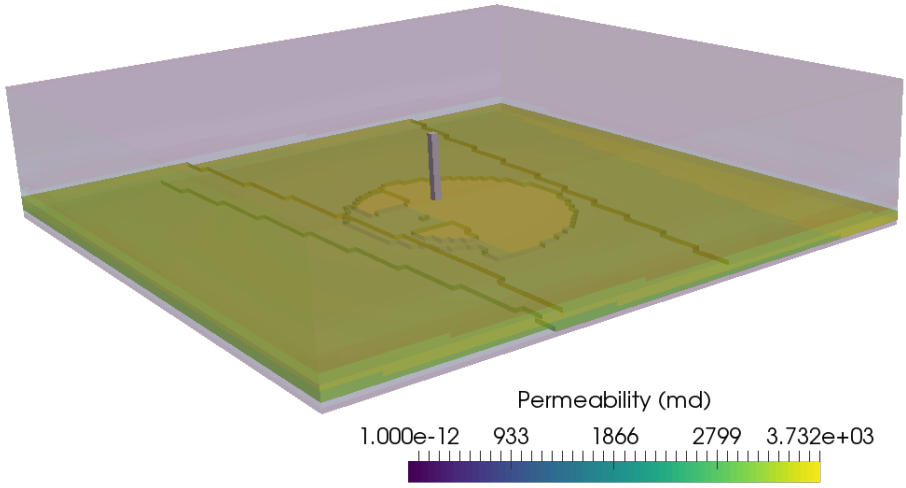

Figure 12: Permeability field in the near wellbore. The injection well is represented at the center of the domain.

4.2. Iterative-IMPES efficiency

From the results in the test cases a tolerance in divergence of 10^{-4} is enough to guarantee stability while a tolerance of 10^{-6} leads to solutions indistinguishable from the ones obtained by using a direct solver.

The impact of the solver tolerance and the preconditioner for test cases is shown in Figure 16. For fixed values of the solver tolerance higher than 10^{-6}, it can be seen that GMRES–ILUT(8) does not offer an overall computational time reduction due to its high computational cost. GMRES–ILUT(4) is robust and outperforms conventional IMPES by a factor up to 3 in all the analysed scenarios. Both GMRES–ILUT(8) and GMRES–ILUT(4) present an approximately constant relative efficiency with respect to the solver tolerance, as most of the computational cost is due to the computation of the preconditioner, i.e., the effect of the number of iterations of the GMRES is

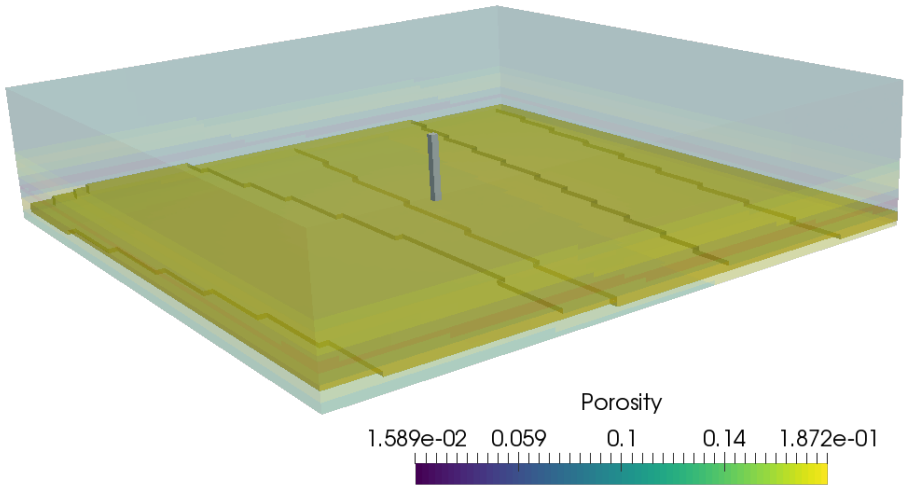

Figure 13: Porosity field in the near wellbore. The injection well is represented at the center of the domain.

small. On the contrary, GMRES–ILU(0) presents a highly variable relative efficiency with respect to the solver tolerance. The computational cost of constructing the preconditioner is small in this case and, as a result, the relative effect of the number of iterations of the GMRES (directly linked to the solver tolerance) is bigger. It should be noticed that GMRES–ILU(0), in spite of showing variable performance, far outperforms the other tested methods for solver tolerances higher than 10^{-7}. Using the ILU0 preconditioner, efficiencies in the range of 3–12 for low tolerances (10^{-7}) and 10–25 for moderate tolerances (10^{-5}) have been measured.

The advantage of using the proposed iterative solver strategy also grows with the problem size as can be observed in Figure 17. This property increases the suitability of the strategy for industrial size problems.

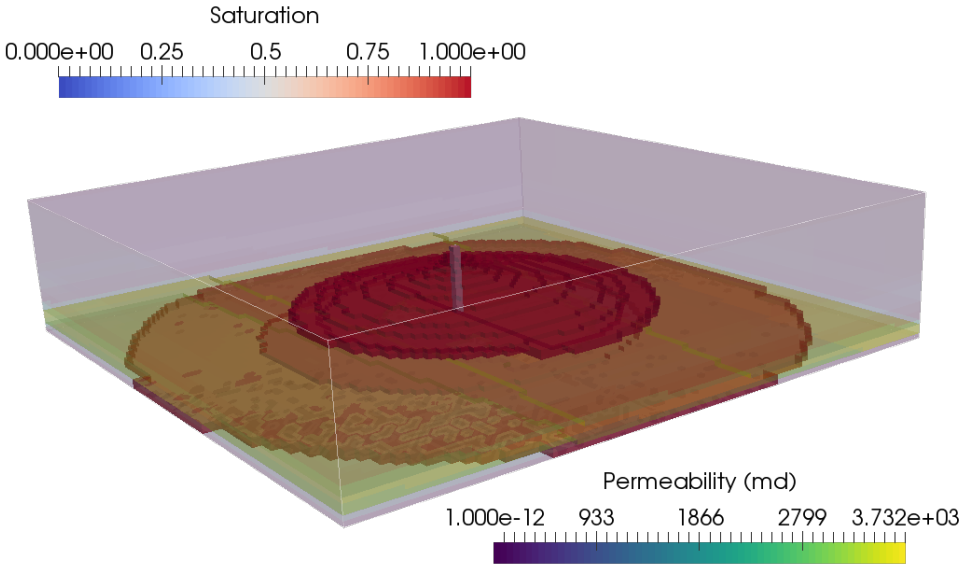

Figure 14: Saturation distribution after 10 days of simulation.

4.2.1. Comparisons with alternative approaches

It has been shown that a solver tolerance value of $\mathrm{DIV}_{\mathrm{tol}} = 10^{-6}$ coupled with a ILU0 preconditioner results in a proper balance between accuracy and performance for all the test cases. In Table 2 detailed timing data is listed. It can be observed that for a tolerance of 10^{-6}, that yields accurate results in all tested scenarios, the iterative IMPES strategy produces speed ups ranging from 7 to 19 due to a drastic reduction in the pressure solver cost.

These results should be put into perspective by comparing them with other techniques found in the literature to reduce the computational cost for solving the system (10) – (11) with respect to the conventional IMPES method.

The improved IMPES technique [13], solves the pressure equation every

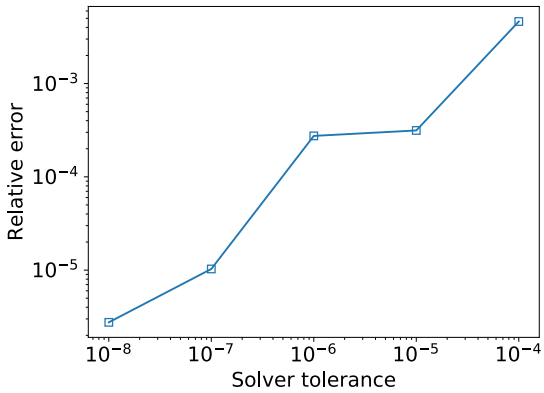

Figure 15: Impact of tolerance in the relative error of injected flow rate for case 4.

n time steps letting the error in total velocity divergence grow up to a tolerance, then the pressure field is updated. This strategy can obtain great computational time reductions compared to conventional IMPES when high tolerance values $\mathrm{DIV}_{\mathrm{tol}} \approx 10^{-2} \sim 10^{-3}$ are used. With pressure and saturation solver times for conventional IMPES shown in Table 2, a maximal speed up of 40 can be obtained with this method. However, as detailed in Section 4.2 , a value for total velocity tolerance of $\sim 10^{-6}$ is required to avoid unphysical saturation distributions. With these tolerances, improved IMPES is no longer efficient [14], while iterative IMPES still outperforms conventional IMPES by a factor of up to 20 in the analysed cases.

5. Conclusions

A method to solve the multiphase flow through a porous media has been introduced. The method is very efficient to solve problems where temporal accuracy is required. The developed numerical scheme takes advantage of

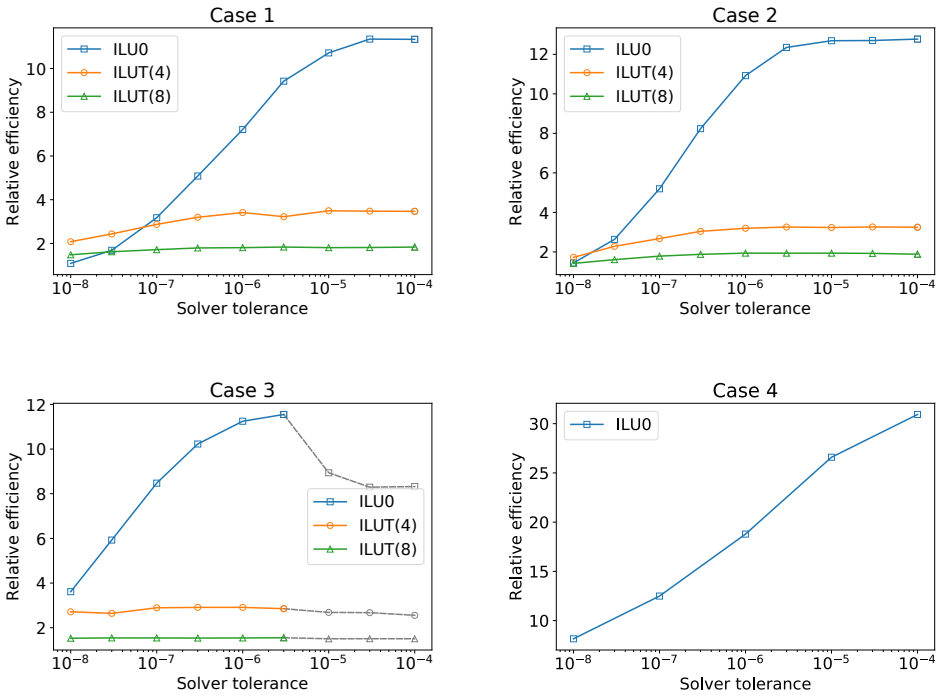

Figure 16: Relative efficiency of the iterative IMPES solver for the test cases with three different preconditioners, ILU0, ILUT(4) and ILUT(8). Dashed lines show simulations with wrong saturation distributions.

the time step restriction of the IMPES method to reduce the computational cost of the implicit pressure solver by using an iterative linear solver. Furthermore, the relation between the iterative linear solver tolerance and the accuracy of the solution has been studied. The validity of the developed strategy to improve the performance of the method without compromising the accuracy of the solution has been shown for a wide range of complex problems with wells, gravity and capillary effects.

An improvement in computational time of an order of magnitude with

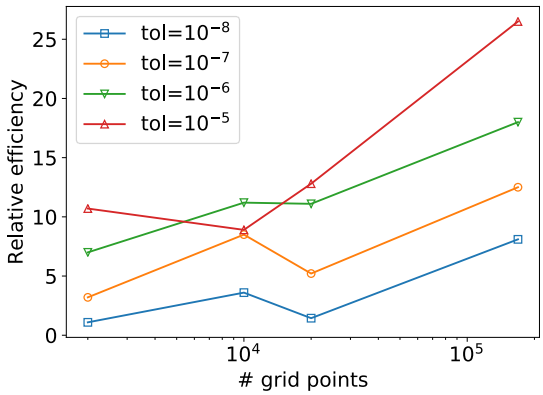

Figure 17: Scalability of the efficiency of the iterative solver strategy with the problem size. Data extracted from the test cases using ILU0 preconditioner.

respect to the conventional IMPES has been measured in a wide range of simulations. This efficiency improvement is achieved while retaining high temporal accuracy. This feature can be relevant in some applications of high interest such as detailed near wellbore simulations or transient processes.

Acknowledgments

This work has been partially supported by REPSOL under the research grant P130120150 monitored by Dr. Angel Rivero. This work has been partially supported by Ministerio de Economía y Competitividad (Spain) under the research grant TRA2015-67679-C2-2-R. The authors acknowledge the computer resources and technical assistance provided by the Centro de Supercomputación y Visualización de Madrid (CeSViMa).

	Conventional IMPES			Iterative IMPES				
	time (s)	p (s)	S (s)	time (s)	p (s)	S (s)	#Iter	speedup
Case 1	138	133	3.2	19	14	3.2	8426	7.2
Case 2	1074	1037	21	98	68	21	6337	11.0
Case 3	4424	4281	91	393	250	91	57663	11.2
Case 4	13800	-	-	736	-	-	-	18.8

Table 2: Performance comparison between conventional and iterative IMPES ($\mathrm{DIV}_{\mathrm{tol}} = 10^{-6}$). Time corresponds to the total simulation time, p corresponds to the time required by the pressure equation solver and S by the saturation equation solver and mobilities computation.

References

[1] D. Ding, Coupled simulation of near-wellbore and reservoir models, Journal of Petroleum Science and Engineering 76 (1-2) (2011) 21–36. doi:10.1016/j.petrol.2010.12.004.

[2] F. A. Dumkwu, A. W. Islam, E. S. Carlson, Review of well models and assessment of their impacts on numerical reservoir simulation performance, Journal of Petroleum Science and Engineering 82-83 (2012) 174–186. doi:10.1016/j.petrol.2011.12.005.

[3] T. Leeftink, C. Latooij, W. Rossen, Injectivity errors in simulation of foam EOR, Journal of Petroleum Science and Engineering 126 (2015) 26–34. doi:10.1016/j.petrol.2014.11.026.

[4] T. B. Tan, N. Kalogerakis, A Fully Implicit, Three-Dimensional, Three-Phase Simulator with Automatic History-Matching Capability ({SPE}-

21205), Proc. of the 11TH SPE Reservoir Simulation Symp. (Anaheim, California) (1991) 35–46doi:10.2118/21205-MS.

[5] D. Collins, L. Nghiem, An efficient approach to adaptive-implicit compositional simulation with an equation of state, SPE Reservoir Engineering 7 (May) (1992) 259–264.

[6] C. N. Dawson, H. Klie, M. F. Wheeler, C. S. Woodward, A parallel , implicit , cell-centered method for two-phase flow with a preconditioned Newton–Krylov solver, Computational Geosciences 1 (Sep) (1997) 215–249. doi:10.1023/A:1011521413158.

[7] Y.-s. Wu, G. Qin, A Generalized Numerical Approach for Modeling Multiphase Flow and Transport in Fractured Porous Media, Communications in Computational Physics 6 (1) (2009) 85–108. doi:10.4208/cicp.2009.v6.p85.

[8] P. Sun, G. Xue, C. Wang, J. Xu, Fast numerical simulation of two-phase transport model in the cathode of a polymer electrolyte fuel cell, Commun. Comput. Phys. 6 (1) (2009) 49–71.

[9] T. Lee, M. Leok, N. H. McClamroch, Geometric numerical integration for complex dynamics of tethered spacecraft, Proceedings of the 2011 American Control Conference (March) (2011) 1885–1891. arXiv:1010.1724, doi:10.1002/nme.

[10] H. Yang, C. Yang, S. Sun, Active-set reduced-space methods with nonlinear elimination for two-phase flow problems in porous media, SIAM Journal on Scientific Computing 38 (4)

(2016) 593–618. arXiv:http://dx.doi.org/10.1137/15M1041882, doi:10.1137/15M1041882.

[11] J. W. Sheldon, B. Zondeck, W. J. Cardwell, One dimensional, incompressible, noncapillary two-phase fluid flow in porous media, Trans, AIME 216 (1959) 290–296.

[12] H. L. Stone, A. O. Garder Jr., Analysis of Gas-Cap or Dissolved-Gas Drive Reservoirs, Society of Petroleum Engineers Journal 1 (2). doi:10.2118/1518-G.

[13] Z. Chen, G. Huan, B. Li, An improved impes method for two-phase flow in porous media, Transport in Porous Media 54 (3) (2004) 361–376. doi:10.1023/B:TIPM.0000003667.86625.15.

[14] F. Hurtado, C. Maliska, A. Silva, A Variable Timestep Strategy for Accelerating the IMPES Solution Algorithm in Reservoir Simulation, Proceedings of the XXVII Iberian Latin American Congress on Computational Methods in Engineering. Belém, Brasil: UFPA (September) (2006) 14.

[15] J. Kou, S. Sun, A new treatment of capillarity to improve the stability of impes two-phase flow formulation, Computers & Fluids 39 (10) (2010) 1923 – 1931. doi:http://dx.doi.org/10.1016/j.compfluid.2010.06.022.

[16] J. Kou, S. Sun, On iterative IMPES formulation for two phase flow with capillarity in heterogeneous porous media, International Journal of Numerical Analysis and Modeling 1 (1) (2010) 20–40.

[17] P. Horgue, C. Soulaine, J. Franc, R. Guibert, G. Debenest, An open-source toolbox for multiphase flow in porous media, Computer Physics Communications 187 (2015) 217 – 226. doi:http://dx.doi.org/10.1016/j.cpc.2014.10.005.

[18] R. S. da Silva, P. R. M. Lyra, R. B. Willmersdorf, D. K. E. de Carvalho, A higher resolution edge-based finite volume method for the simulation of the oil-water displacement in heterogeneous and anisotropic porous media using a modified IMPES method, International Journal for Numerical Methods in Fluids 82 (12) (2016) 953–978. doi:10.1002/fld.4254.

[19] R. Künze, I. Lunati, S. H. Lee, A multilevel multiscale finite-volume method, Journal of Computational Physics 255 (2013) 502–520. doi:10.1016/j.jcp.2013.08.042.

[20] H. Liu, K. Wang, Z. Chen, J. Luo, H. Deng, Y. He, Numerical simulation of multi-phase flow in porous media on parallel computers (2016). arXiv:1606.00556.

[21] D. Karlo, E. D. Carvalho, R. Soares, P. Roberto, M. Lyra, 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, Proceedings of COBEM 2009.

[22] R. Brooks, A. Corey, Hydraulic Properties of Porous Media, Colorado State University Hydrology Papers, Colorado State University, 1964.

[23] H. Hoteit, A. Firoozabadi, Numerical modeling of two-phase flow in heterogeneous permeable media with different capillar-

ity pressures, Advances in Water Resources 31 (1) (2008) 56–73. doi:10.1016/j.advwatres.2007.06.006.

[24] D. Kuzmin, On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection, Journal of Computational Physics 219 (2) (2006) 513–531.

[25] Y. Saad, M. H. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869. doi:10.1137/0907058.

[26] M. Todd, P. O'Dell, G. Hirasaki, Methods for increased accuracy in numerical reservoir simulators, Society of Petroleum Engineers Journal 12 (6) (1972) 515–530.

[27] K. Coats, IMPES Stability: The CFL Limit, SPE Journal 8 (03) (2003) 291–297.

[28] Y. Saad, Iterative Methods for Sparse Linear Systems: Second Edition, Society for Industrial and Applied Mathematics, 2003.

[29] J. Franc, P. Horgue, R. Guibert, G. Debenest, Benchmark of different CFL conditions for IMPES, Comptes Rendus Mécanique 344 (10) (2016) 715 – 724. doi:http://dx.doi.org/10.1016/j.crme.2016.08.003.

[30] M. Christie, M. Blunt, Tenth SPE comparative solution project: A comparison of upscaling techniques, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2001. doi:10.2118/66599-MS.