Proceeding of the XXVII Iberian Latin American Congress on
Computational Methods in Engineering
September 3 to 6, 2006 - Belém, Pará - BRAZIL

A VARIABLE TIMESTEP STRATEGY FOR ACCELERATING THE IMPES SOLUTION ALGORITHM IN RESERVOIR SIMULATION

Fernando Sandro Velasco Hurtado
Clovis Raimundo Maliska
António Fábio Carvalho da Silva
fernando@sinmec.ufsc.br
maliska@sinmec.ufsc.br
afabio@sinmec.ufsc.br

SINMEC Lab, Department of Mechanical Engineering, Federal University of Santa Catarina, 88040-900, Florianopolis, BRAZIL

Abstract. *The IMPES solution algorithm is still popular in reservoir simulation because of their simplicity and low demand of computational resources. In using this algorithm, pressure and saturation equations are partially decoupled in order to solve sequentially each equation for its main variable. However, a severe stability constraint arises from the use of explicit approximations needed for decoupling equations. This is the main disadvantage of the IMPES scheme because extremely small timesteps should be used for overcoming the stability restriction. In order to accelerate the performance of the IMPES algorithm and make it able for solving larger problems, a variable timestep strategy is devised in this work. This strategy consist basically in using larger timesteps for pressure equation than for saturation equation, in order to reduce the number of times that pressure system of equations is solved along a simulation. This is possible because pressure equation is approximated in implicit way, so there is no any stability restriction associated to it. The size of timestep on pressure equation is determined dynamically according to the variation of the total velocity field in a similar way to adaptive timestep schemes used in fully implicit algorithms. By means of a typical problem, we show that the use of this strategy produces large savings in computation time without noticeable increasing of numerical error, compared to conventional IMPES solutions.*

Keywords: *Reservoir Simulation, Solution Algorithms, IMPES Method.*

1. INTRODUCTION

The Implicit Pressure – Explicit Saturation (IMPES) method originated in the earlier years of reservoir simulation. Since then it has become a popular method for solving the system of equations describing multiphase flow in reservoirs. Solving those equations is a difficult task that requires special techniques because their variables are normally coupled in a complicated way.

For the two-phase case, the pressure and the saturation are usually the main variables in the flow equations. The numerical strategy that leads to the IMPES scheme in that case is the use of different time approximations for the pressure and the saturation terms during the discretization of differential equations. While the pressure is approximated according to an implicit scheme, for the saturation is considered an explicit scheme that leads to a partial decoupling of those variables. In this way, the computation becomes remarkably simple because only one linear system for the pressure must be solved per timestep. In contrast, a larger and highly nonlinear system of equations must be solved per timestep if a fully implicit approach were used.

Although the explicit approximation of saturation simplifies considerably the solution of the equations, it is also the origin of one of the major disadvantages of the IMPES algorithm. In order to the IMPES scheme be stable, the so-called CFL condition must be observed and, as a consequence, a severe restriction on the timestep size arises. This is particularly critic in problems with very high fluid velocity and/or necessity of using grids with refined cells. Stable timesteps for simulating those problems can be particularly small and, consequently, the computational time for obtaining a simulation of a whole displacement process can become unreasonably large.

Despite the current tendency of adopting more robust algorithms for reservoir simulation, still a considerable interest exists for using the IMPES scheme in certain applications, and for developing special strategies for improving its performance. Recently, Chen et al. (2004) have published a paper describing some enhancements for the IMPES algorithm. In order to accelerate its performance two main strategies are suggested in that work. Firstly, a timestep control scheme for the saturation equation not based on the CFL condition but on the local maximum variation of saturation. Secondly and more important, it is proposed an approach in which larger timesteps are used for the pressure equation than for the saturation equation. In principle, this is possible because IMPES conditional stability is associated only to the explicit approximation of saturation and not to the approximations made on pressure equation. Larger timesteps for pressure equation mean that pressure linear system must be solved less times during a simulation and this can produce a significant reduction of computation time.

The main objective of this work is to explore the second strategy introduced by Chen et al. (2004) and to extend it further. Although it is possible to use larger timesteps for pressure equation, it is still an open question how large those timesteps can get without damaging the numerical solution. An adaptive control strategy based on the mean variation of total velocity field is proposed in this paper as a way of determining a timestep for pressure equation as large as possible without degrading the quality of the numerical solution. In fully implicit formulations, which do not undergo such severe stability restriction as IMPES does, similar adaptive timestep control strategies are commonly used, although generally based on the maximum variation of saturation and/or pressure. In our approach, timesteps for saturation equation always are determined from the CFL condition in order to prevent oscillations and other abnormal artifacts in the numerical solutions. Some of the results presented by Chen et al. show oscillations because they abandoned the CFL condition as a way of determining stable timesteps for saturation.

The remainder of this paper is outlined as follows. In the next section we briefly describe the numerical model considered for testing the IMPES solution scheme. In the third section the main steps of conventional IMPES algorithm are described and a typical reservoir problem is solved in different grids for testing its performance. Then, in the fourth section we introduce a first approach to improving IMPES algorithm setting large but constant values to the pressure timestep. In the fifth section we introduce the final version of the improved IMPES algorithm, with an adaptive control strategy for calculating pressure timesteps. Finally, in the last section some concluding remarks are drawn.

2. NUMERICAL MODEL

For the sake of simplicity, the improved solution algorithm will be described considering a two-phase incompressible and immiscible flow model and neglecting any influence of capillary pressure. A more general formulation including fluid compressibility, capillary pressure, and gravity is described in Hurtado (2005).

The mathematical model for that kind of flow can be reduced two a pair of differential equations (Peaceman, 1977; Aziz & Settari, 1979), the so-called Buckley-Leverett form of the saturation equation,

$$\phi\, \partial_t s + \vec{\nabla} \cdot (F\, \vec{\mathbf{v}}_T) = 0 \tag{1}$$

and the pressure differential equation,

$$\vec{\nabla} \cdot (\lambda_T \vec{\vec{\mathbf{K}}} \cdot \vec{\nabla} P) = 0 \tag{2}$$

Here, ϕ and $\vec{\vec{\mathbf{K}}}$ are the porosity and the tensor of absolute permeability of the medium, respectively. The main variables on each equation are the saturation of water s and the pressure P. Furthermore, $\lambda_T = \lambda_w + \lambda_o$ is the total mobility (w stands for water phase and o stands for oil phase), and $F = \lambda_w / \lambda_T$ is known as fractional flux function, all of them being functions of the saturation. Finally, $\vec{\mathbf{v}}_T = \vec{\mathbf{v}}_w + \vec{\mathbf{v}}_o$ is the total velocity, a variable that couples the pressure and saturation equations, since it can be expressed as

$$\vec{\mathbf{v}}_T = -\lambda_T \vec{\vec{\mathbf{K}}} \cdot \vec{\nabla} P \tag{3}$$

In order to discretize in space the system of differential equations, the Element-based Finite Volume Method (EbFVM) will be considered. This approach follows the basic guidelines of the conventional finite volume method, namely the integration of differential equations over control volumes in a way that conservation is automatically enforced. However, a significant improvement in flexibility is introduced through the concept of element as the basic geometrical entity for discretization of the solution domain, since in this way the use of unstructured grids is readily affordable. Here we only give a very brief introduction to the application of EbFVM, more details can be found elsewhere (Maliska, 2004; Hurtado, 2005; Hurtado et al., 2005).

The main geometric entities considered in EbFVM are shown in Fig. 1. The grid is formed by *elements*, which are quadrilateral in this work. These entities are used for defining the discretized geometry of the solution domain as well as for defining the spatial variation of physical properties of the medium. The unknowns of the problem are calculated at the *nodes*, located at every element corner. Around every node is built a *control volume*, formed by portions of the elements sharing a common node. Every control volume is delimited by a certain number of *faces*, obtained joining the center of every neighboring element with the midpoint of its two sides sharing the node around which the control volume is built. Normally, a discretized equation for a control volume represents some kind of balance, thus, it becomes necessary to compute fluxes across the faces. As surface integrals defining fluxes are usually approximated by the midpoint rule, the face center points are also important entities in EbFVM and they are commonly known as *integration points*.

After integration of differential equations (1) and (2) over a generic control volume, they can be approximated, respectively, as

$$\frac{s_p^{n+1} - s_p^n}{\Delta t^n}\, \phi_p \Delta V_p + \sum_e \left\{ \sum_i F_i^\theta (\vec{\mathbf{v}}_T)_i^\theta \cdot \Delta\vec{\mathbf{S}}_i \right\}_e = 0 \tag{4}$$

$$\sum_e \left\{ \sum_i (\lambda_T)_i^\theta \left[\vec{\vec{\mathbf{K}}}_e \cdot (\vec{\nabla} P)_i^\theta \right] \cdot \Delta\vec{\mathbf{S}}_i \right\}_e = 0 \tag{5}$$

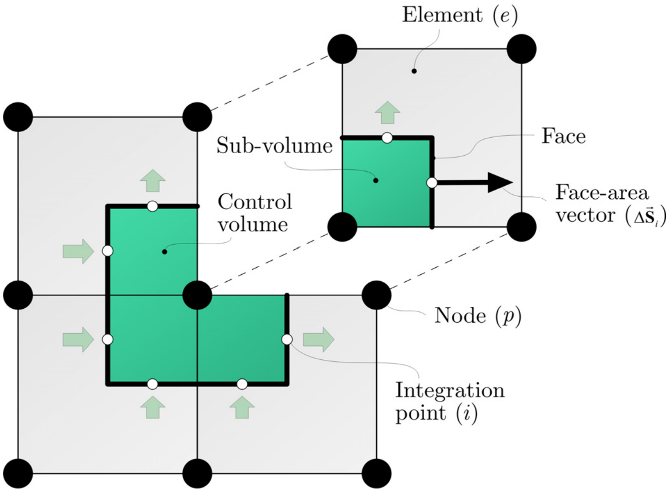

Figure 1 - Main geometrical entities on the Element-based Finite Volume Method.

Here, subscripts specify the points where the variables are evaluated and superscripts specify the time level. Moreover, ΔV_p is the volume of the control volume, $\Delta\vec{\mathbf{S}}_i$ is the area vector of a face limiting the control volume, and Δt^n is the timestep. The outer summations in Eqs. (4) and (5) involve all elements contributing to the control volume. On the other hand, the inner summations involve the two integration points that lies over the control volume faces inside a contributing element. After the assembling of the equations corresponding to all control volumes in the solution domain, a system of algebraic equations is obtained. The only unknowns in those equations must be the nodal values of saturation and pressure. For more details concerning to the discretization process, see Hurtado et al. (2005).

In Eqs. (4) and (5), θ is the time level at which the volumetric fluxes are calculated. Considering $\theta = n+1$ leads to the fully implicit approach. Since in that case the saturation and pressure equations are fully coupled and they are highly nonlinear, usually Newton-like methods are the only alternative to solve them (Mattax & Dalton, 1990). A much simpler approach, the conventional IMPES scheme, is obtained when $\theta = n$. With this approximation it is possible to decouple the pressure equation from the saturation equation and solve each one independently for its own variable. This scheme will be analyzed in more detail in the following section.

3. THE CONVENTIONAL IMPES ALGORITHM

The main steps of the IMPES algorithm are summarized in Fig. 2. It is assumed that at time level n a discrete approximation of the saturation field is available. If that time level is the first one for the simulation, that saturation field should come from the initial condition of the problem. As saturation is known, total mobility can be evaluated everywhere and consequently the only unknown variable in Eq. (5) is the pressure. Thus, as first step on the algorithm, a linear system based in Eq. (5) with $\theta = n$ should be assembled and solved for the pressure.

The pressure field obtained in the previous step, which corresponds also to the time level n, permits to compute the total velocity at integration points. This operation, which should be performed using a discretized form of Eq. (3), constitutes the second step of the algorithm.

With total velocity and saturation totally defined at time level n, the only remaining unknown in Eq. (4) is the nodal saturation value at the next time level. Therefore, Eq. (4) can be rearranged and written as

$$s_p^{n+1} = s_p^n - \frac{\Delta t^n}{\phi_p \Delta V_p} \sum_e \left\{ \sum_i F_i^n (\vec{\mathbf{v}}_T)_i^n \cdot \Delta\vec{\mathbf{S}}_i \right\}_e \tag{6}$$

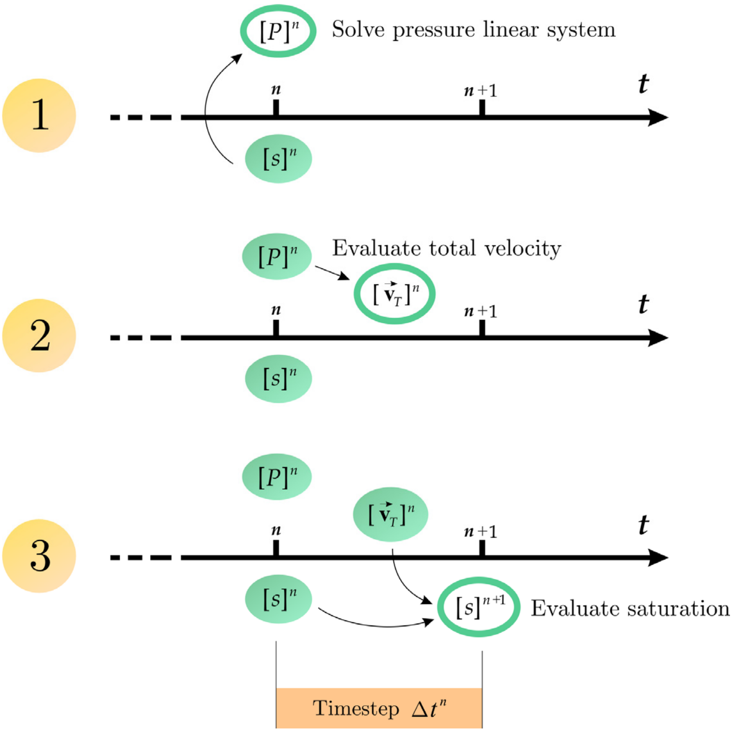

Figure 2 – Main steps in conventional IMPES solution algorithm.

Consequently, the last step in an IMPES solution block is the calculation of the saturation field at time level $n+1$, evaluating Eq. (6) node by node. The explicit approximation of the flux terms in Eqs. (4) and (5) has resulted not only in the decoupling of pressure and saturation, but also in a set of decoupled equations for the nodal values at the new time level. For that reason, there is no actual need of solving a linear system of equations for updating saturation when using the IMPES algorithm.

The major advantage of the IMPES algorithm over the fully implicit scheme thus becomes evident. Instead of solving a highly nonlinear system of $2N$ equations per time level, where N is the number of grid nodes, it is only needed to solve a linear system of N equations when using the IMPES scheme. The little requirement of computational resources and the simplicity of implementation have turned the IMPES into a very popular solution scheme since the earlier years of reservoir simulation.

There is however a several drawback to using IMPES algorithm for solving certain problems. The explicit approximation of flux terms in Eq. (4) is only conditionally stable. Therefore, in order to IMPES algorithm be stable, the so-called CFL condition should satisfied. In general, this condition can be stated as (Coats, 2001)

$$\max_p \left(\frac{\Theta_p \Delta t^n}{\phi_p \Delta V_p} \right) < 1 \tag{7}$$

where Θ_p is some function of flow-rates and reservoir and fluid properties, associated to control volume p. This condition can be written in the equivalent form

$$\Delta t^n < \min_p \left(\frac{\phi_p \Delta V_p}{\Theta_p} \right) \tag{8}$$

Hence, the CFL condition sets a constraint on the timestep size for performing a simulation. In situations when grids with very small control volumes have to be used or very high fluid velocities arise during the simulation, the timestep limitation can make the IMPES scheme totally impractical (Mattax & Dalton, 1990).

In order to evaluate the performance of the conventional IMPES algorithm, initially we solve the displacement problem depicted in Fig. 3. The domain corresponds to one quarter of a five-spot configuration, with a heterogeneous but isotropic distribution of absolute permeability. Water is injected into the reservoir at a constant volumetric rate and displaces the resident oil to the production well. This displacement process was simulated using different two-dimensional Cartesian grids and the conventional IMPES algorithm was employed in all cases. The objective of this test was to check the increase of computation time as grids are refined, which is a customary test for evaluating the performance of numerical schemes.

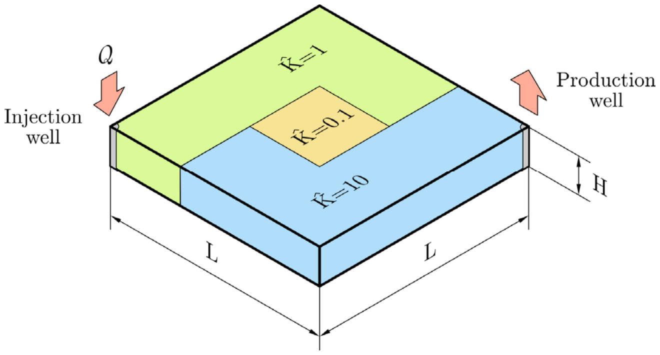

Figure 3 – Sketch of the test problem.

The following saturation functions were considered for all tests

$$\hat{\lambda}_T(\hat{s}) = 2\hat{s}^3 + (1-\hat{s})^5 \tag{9}$$

$$F(\hat{s}) = \frac{2\hat{s}^3}{2\hat{s}^3 + (1-\hat{s})^5} \tag{10}$$

where $\hat{\lambda}_T$ is the total mobility normalized by $\lambda_o^{\max}$, the maximum value of oil mobility. Moreover, $\hat{s}$ is the normalized saturation, defined as

$$\hat{s} \doteq \frac{s - s_{wi}}{1 - s_{or} - s_{wi}} \tag{11}$$

Here, s_{wi} is the irreducible water saturation and s_{or} is the residual oil saturation. In fact, in our implementation all variables were treated as non-dimensional variables. For instance, time, velocity, and pressure were non-dimensionalized, respectively, according to the expressions

$$\hat{t} \doteq \frac{Q\,t}{\phi L^2 H} \tag{12}$$

$$\hat{\mathbf{v}}_T \doteq \frac{LH}{Q}\mathbf{v}_T \tag{13}$$

$$\hat{P} \doteq \frac{H\lambda_o^{\max}}{Q}P \tag{14}$$

where Q is the injection rate and L and H are geometric dimensions defined in Fig. 3. All results presented afterward correspond to non-dimensional variables defined by preceding equations.

In the first test were used four regular Cartesian grids, with 21×21, 30×30, 42×42, and 60×60 elements, respectively. For solving the pressure linear system was used the GMRES method with a SSOR preconditioning matrix (Saad, 2003), stipulating a residual tolerance of 10^{-8}. Simulations were performed on each grid for the time interval $\hat{t} \in [0,1]$.

Figure 4 shows a comparison of accumulated CPU times for each one of the basic operations performed in conventional IMPES algorithm, as well as the overall CPU time for each one of the simulations run. The first interesting characteristic in Fig.4 is that the total CPU time raised approximately two orders of magnitude while the number of grid nodes barely increased one order of magnitude. Although the actual values of CPU time are strongly influenced by both hardware and code implementation characteristics, the results show that a simulation with IMPES algorithm even in a barely refined grid can demand a prohibitive time of computation. This is caused mainly by the stability restriction, Eq. (8), which establishes a reduction of the stable timestep as control volume sizes decrease when the grid is refined. For instance, in the 60×60-element grid the stable timestep was as small as $\Delta\hat{t} \approx 1.7 \times 10^{-5}$, which means that approximately 59000 basic IMPES operations were performed during a simulation for the interval $\hat{t} \in [0,1]$ in that grid.

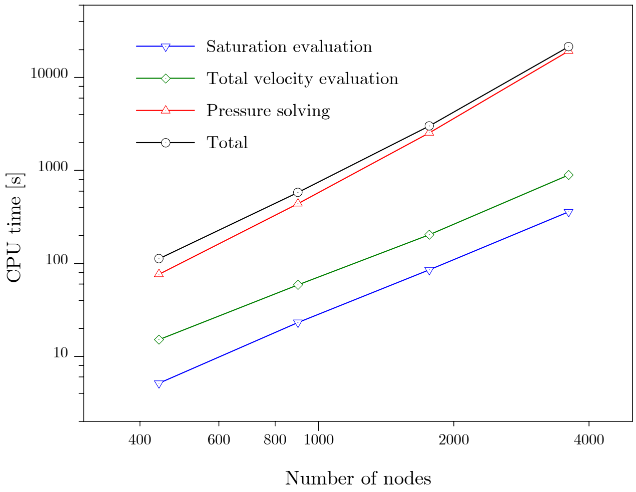

Figure 4 – CPU time for different operations performed in conventional IMPES algorithm.

Figure 4 shows also another remarkable feature. That is, the major percentage of the overall CPU time per simulation corresponds to the operation of solving the pressure linear system. This can be observed more clearly in Fig. 5, which shows the time percentages for each operation. In the 60×60-element grid, solving linear systems for the pressure throughout the simulation

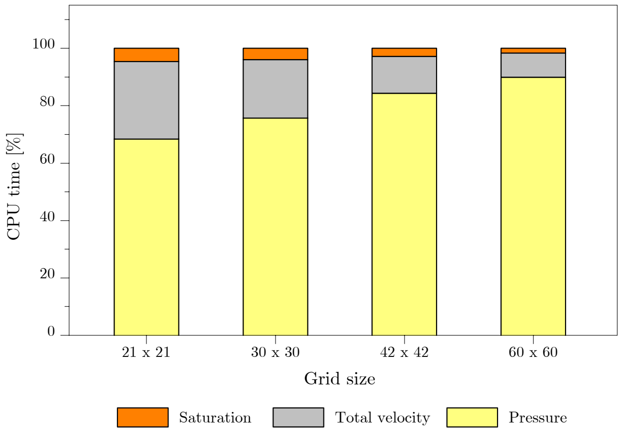

Figure 5 – Comparison of CPU time percentages for each operation in conventional IMPES algorithm.

demanded almost 90% of the total time. This behavior is rather logic because solving a linear system is always a more demanding task than simply evaluating equations, as is made with saturation and total velocity. If would be possible to reduce the number of times that pressure linear system ought to be solved during a simulation, certainly the overall CPU time per simulation would be reduced significantly. This is one of the main ideas behind the improved forms of the IMPES algorithm that will be described in the two following sections.

4. A FIRST APPROACH FOR ACCELERATING IMPES ALGORITHM

The severe stability constraint that suffers IMPES algorithm is caused by the explicit approximation considered for saturation (Peaceman, 1977; Mattax & Dalton, 1990). As pressure is approximated according to an implicit scheme, in principle there not should be any stability restriction associated to this variable. Therefore, we could devise a solution algorithm for two-phase flow in which saturation would be updated using small timesteps, hence obeying the CFL condition, but maintaining pressure and total velocity fields frozen a certain number of times. Only after that, pressure linear system would be solved again and afterward total velocity would be updated. The alternative third step for this modified IMPES algorithm is depicted in Fig. 6, where for each pressure and total velocity updating, saturation is updated m times, where $m > 1$ is an integer number.

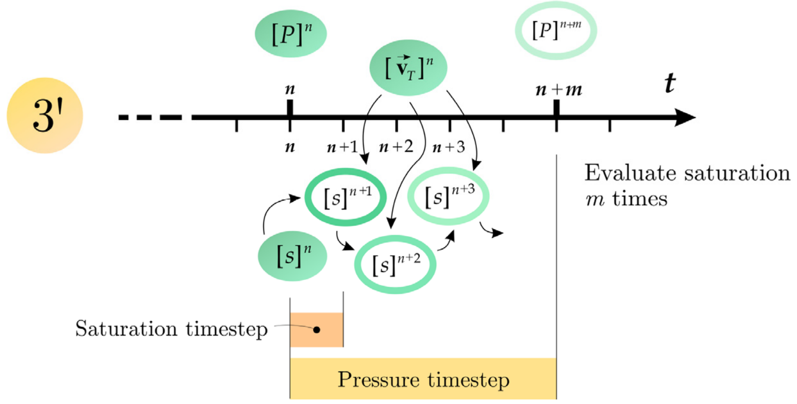

Figure 6 – Alternative third step for improved IMPES solution algorithm.

Although for incompressible flow the pressure equation does not have any term involving time, in the conventional IMPES scheme a same time scale is assumed for solving both pressure and saturation equations. In the modified version of the algorithm outlined above, a larger timestep results for pressure equation than for saturation equation. Since in general pressure and total velocity fields evolve more slowly than saturation, it is expected that this modification does not introduce noticeable additional errors into the numerical solution. At least if some care is taken in using that accelerated form of IMPES algorithm.

The strategy of using larger timesteps for pressure equation than for saturation as a way of accelerating the performance of IMPES algorithm was suggested by Chen et al. (2004), but also by Hurtado et al. (2004) for simulating two-phase displacements in core samples. However, none of that works suggest a practical way for determining how large the pressure timestep could be set for a specific problem. It is not showed clearly either how much CPU time savings can be obtained using this strategy or how can it modify the numerical solution. In order to obtain some insight about those questions, results of some tests will be presented in the following section. But before that we will describe an improved version of the algorithm introducing an adaptive timestep control for pressure equation.

5. THE IMPES SCHEME WITH ADAPTIVE PRESSURE TIMESTEP CONTROL

In order to systematize further the application of the modified version of the IMPES algorithm, we will introduce now an adaptive control strategy for the pressure timestep. As mentioned in section 2, total velocity is the linking variable between pressure equation and saturation equation. Eventually, any variation of the pressure field will be perceived by the saturation through variations of the total velocity field. Therefore, a technique for determining pressure timesteps aiming a constant variation of the total velocity field throughout the whole simulation is in fact a good option. In order to achieve this we will consider an adaptive control technique similar to the used with fully implicit formulations (Aziz & Settari, 1979). Hence, the following relationship will be used for determining a new timestep for pressure

$$\Delta \hat{t}_P^{n+1} = \frac{DVTOL}{\left\langle \Delta \hat{\mathbf{v}}_T \right\rangle^n} \Delta \hat{t}_P^n \tag{15}$$

where $DVTOL$ is a prescribed value for limiting the variation of the total velocity field and $\left\langle \Delta \hat{\mathbf{v}}_T \right\rangle^n$ is a numerical value that characterizes that variation during the time interval $\Delta \hat{t}_P^n$. Actually, many options do exist to characterize the variation of a discrete approximation of a field. Performing several numerical experiments we found that the following mean value produces in general smooth variations of the pressure timestep along a simulation,

$$\left\langle \Delta \hat{\mathbf{v}}_T \right\rangle^n = \frac{1}{4N_e} \sum_e \left\{ \sum_i \frac{\left| \left[(\hat{\mathbf{v}}_T)_i^n - (\hat{\mathbf{v}}_T)_i^{n-1} \right] \cdot \Delta \vec{\mathbf{S}}_i \right|}{\left| \Delta \vec{\mathbf{S}}_i \right|} \right\}_e \tag{16}$$

Here, N_e is the total number of elements on the grid and $\left| \Delta \vec{\mathbf{S}}_i \right|$ is the modulus of the face area vector. The outer summation includes all elements in the grid and the inner summation includes all integration points inside an element. In fact, Eq. (16) represents the mean value of the variations of the normal component of total velocity at the faces of all control volumes in the grid. We chose to calculate the velocity variations based on the normal component at control volume faces because this is the component used in Eq. (6) for update saturation. In order to prevent too fast increase or too fast decrease of the pressure timestep, limiting values were introduced for the ratio $\Delta \hat{t}_P^{n+1} / \Delta \hat{t}_P^n$. Typically, the constraints $\Delta \hat{t}_P^{n+1} / \Delta \hat{t}_P^n > 0.75$ and $\Delta \hat{t}_P^{n+1} / \Delta \hat{t}_P^n < 1.25$ were considered.

The accelerated version of IMPES algorithm including the adaptive timestep control strategy was tested with the same problem solved in section 3. Initially we considered the 60×60-element grid and simulated the displacement process for the time interval $\hat{t} \in [0,1]$. Values for $DVTOL$ between 10^{-4} and 10^{-2} were considered. Those values correspond to variations on total velocity fields non-dimensionalized according to Eq. (13). In all cases an initial pressure timestep $\Delta \hat{t}_P^0 = 10^{-3}$was prescribed. Figure 9 shows the accumulated CPU times for each operation in the algorithm, besides the total CPU time for the whole simulations. As can be seen, impressive reductions in the total CPU time are obtained using the accelerated algorithm. From a value of $\sim 2 \cdot 10^4 [s]$ with conventional IMPES (see Fig. 4), the total CPU time dropped to $\sim 3 \cdot 10^2 [s]$ with $DVTOL = 10^{-2}$, that is, there was a reduction of two orders of magnitude in the time of computation. For the latter values of $DVTOL$, the accumulated CPU time corresponding to the pressure solving dropped below the accumulated CPU time corresponding to the updating of saturation. Consequently, this latter time began to dominate the total CPU time.

Figure 10 shows the ratio between the pressure timestep and the saturation timestep throughout the simulations with the different values of $DVTOL$. This figure evidences the way in which the adaptive timestep control works. When the total velocity field begins to experiment significant variations, the pressure timesteps are reduced significantly. This occurred, for instance, near the time $\hat{t} \approx 0.4$, when the water front arrived to the production well. After that event, the total velocity field began to vary more slowly and consequently the timesteps were increased gradually.

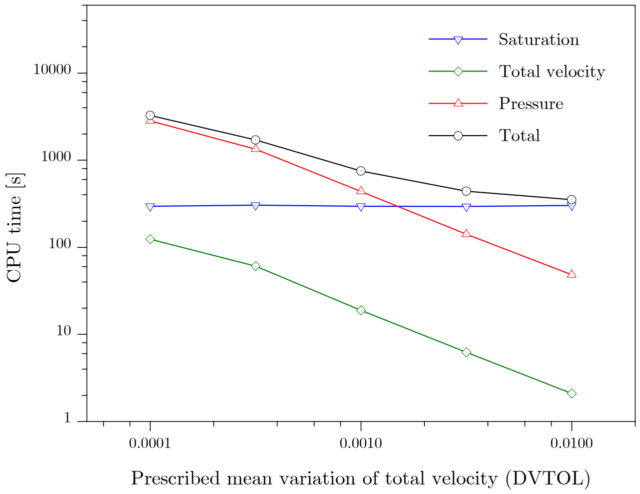

Figure 9 – CPU times as function of the prescribed mean variation of total velocity, using the improved version of IMPES algorithm with adaptive pressure timestep.

As Fig. 10 shows, depending on the prescribed value of $DVTOL$, the pressure timestep can attain a magnitude 1000 times greater than saturation timestep. Figure 11 complements Fig. 10, because it shows the actual mean variation of total velocity along the simulations. As can be seen the actual variation follows adequately the prescribed value in practically all cases.

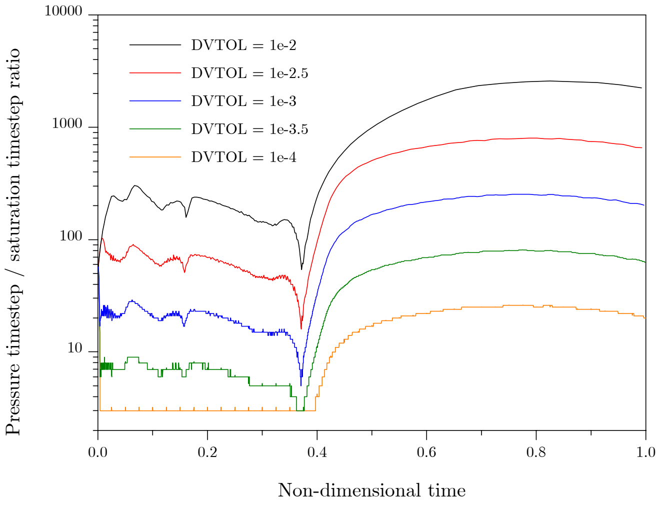

Figure 10 – Variation of the ratio of pressure timestep to saturation timestep throughout the simulation in the 60×60-element grid, for different values of prescribed mean variation of total velocity.

Although it has been showed that the modified IMPES algorithm can notably reduce the computation time for a simulation, still nothing has been commented about the quality of the numerical solutions. In order to evaluate this, the solution obtained with the conventional IMPES algorithm will be taken as reference solution. Figure 12 shows four snapshots of the water saturation field obtained with conventional IMPES. Every discrete saturation field obtained with the modified IMPES algorithm has been subtracted from the correspondent saturation field in the reference solution, and the resulting fields are shown in Fig. 13 for some of the values of $DVTOL$ tested. It is evident that the discrepancies are minor and are confined to the water front neighborhood. Only in the latter time of the solution with $DVTOL = 10^{-2}$ the deviation reach significant values. Maybe even more illustrative is Fig. 14 that depicts the 2-norm of the deviation

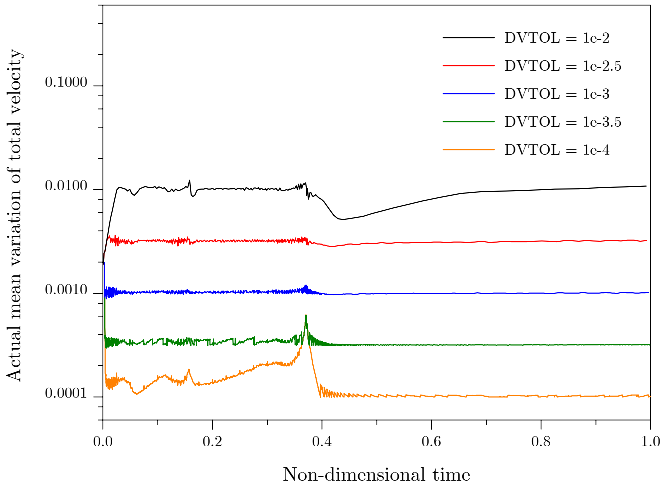

Figure 11 – Actual mean variation of total velocity throughout a simulation in the 60×60 -element grid, for different values of prescribed mean variation of total velocity.

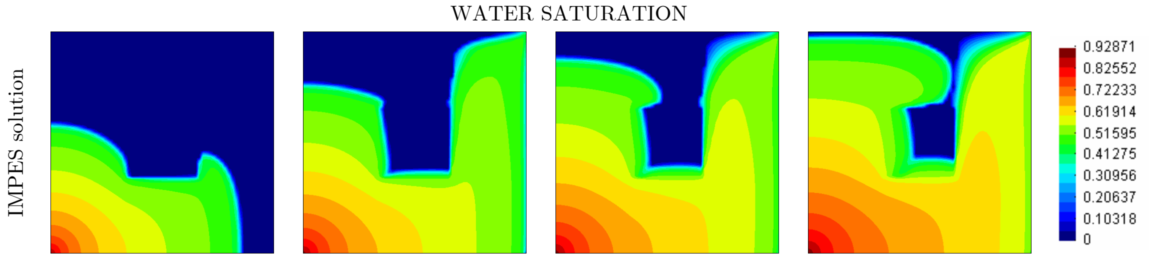

Figure 12 – Evolution of the water saturation field, obtained with the conventional IMPES algorithm in the 60×60 -element grid.

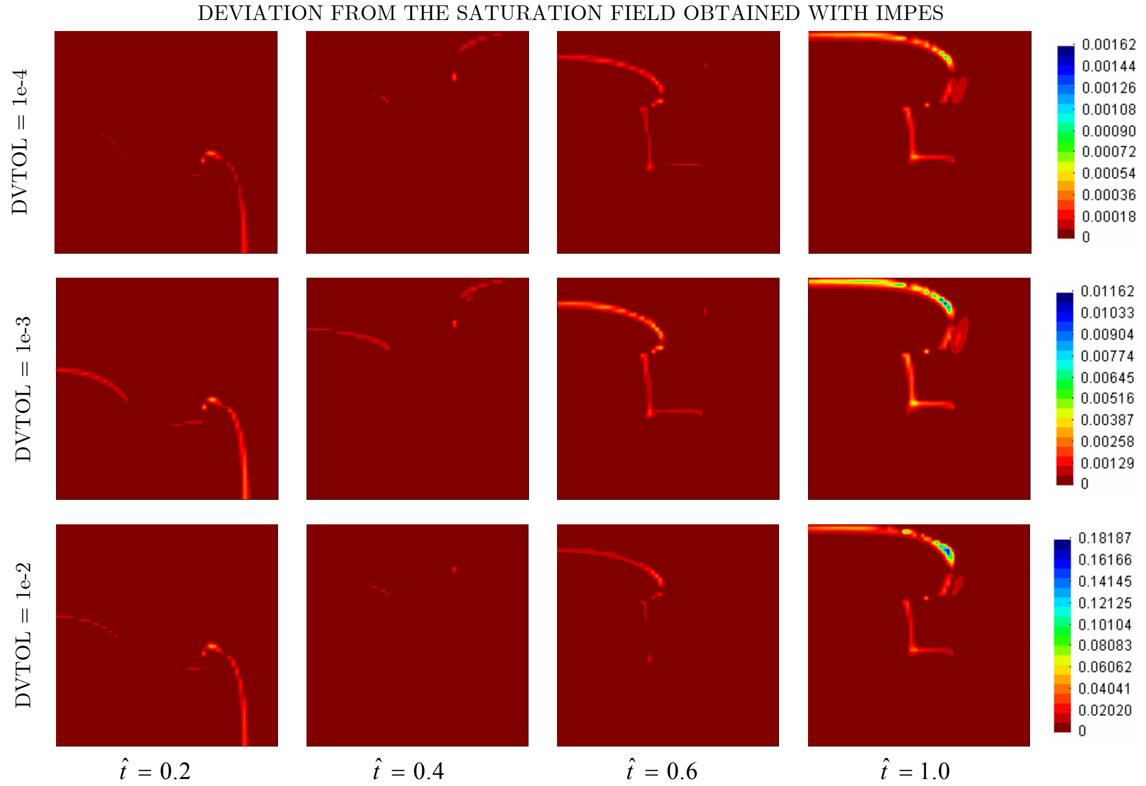

Figure 13 – Deviation of the water saturation field from the correspondent with conventional IMPES algorithm, for three values of prescribed mean variation of total velocity.

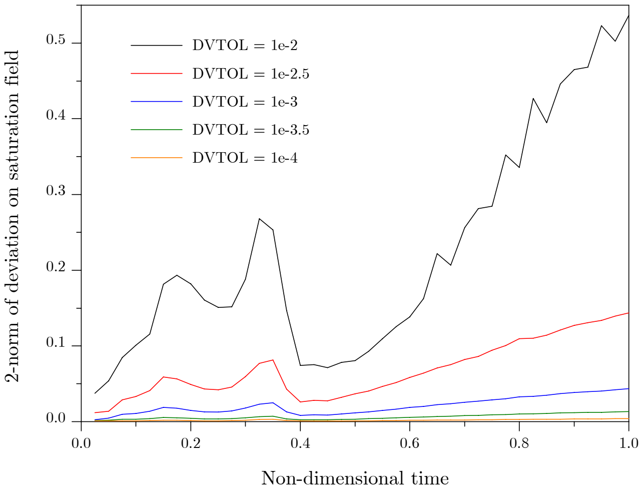

Figure 14 – 2-norm of the deviation on saturation field for different values of prescribed mean variation of total velocity.

on saturation field throughout the simulations. As supposed before, only the simulation with $DVTOL = 10^{-2}$ shows a rather uncontrolled growing of the error after the breakthrough. The remainder simulations show smaller deviations.

It is important also to compare results concerning global parameters such the accumulated oil production and the water-cut. In Figs. 15 and 16 the curves obtained with conventional IMPES algorithm are compared with the curves obtained with different values of $DVTOL$ in the accelerated version of IMPES. Figure 15 shows water-cut curves, while Fig. 16 shows accumulated oil production curves. Since at first sight no significant discrepancy appears to be among all curves, a close-up of a portion of them is showed in detail. Whereas in the water-cut case a periodic discrepancy results evident for the curve obtained with $DVTOL = 10^{-2}$, in the oil production case not any difference is noticeable. All those results show that the cost-benefit relationship of the modified IMPES algorithm is very advantageous because great computational time savings are obtained without introducing significant additional error into the numerical solutions.

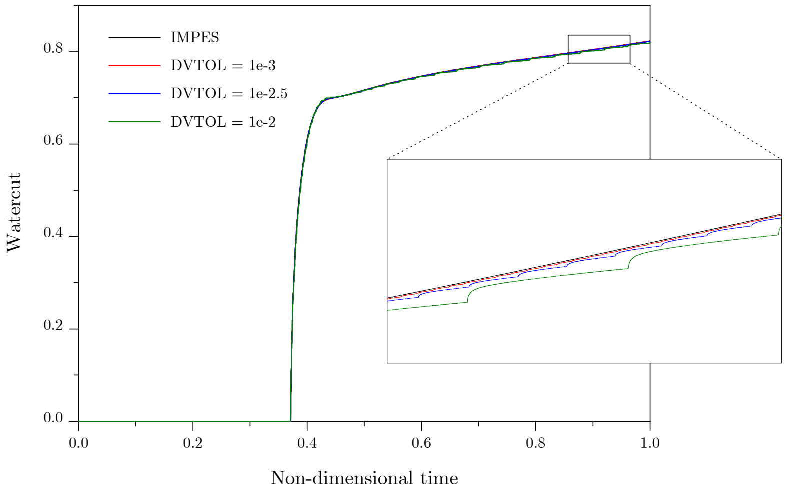

Figure 15 – Comparison of water-cut curves.

Finally, Fig. 17 compares the performance of conventional and accelerated version of IMPES algorithm for different grids. The CPU times for conventional IMPES correspond to total times in Fig. 4 whereas the CPU times for accelerated IMPES were obtained with $DVTOL = 10^{-2.5}$. According the previous tests this value of *DVTOL* presents an adequate compromise between reduction of computational time and increase of the error associated to the accelerated scheme. That value worked well also with other type of reservoir problems tested.

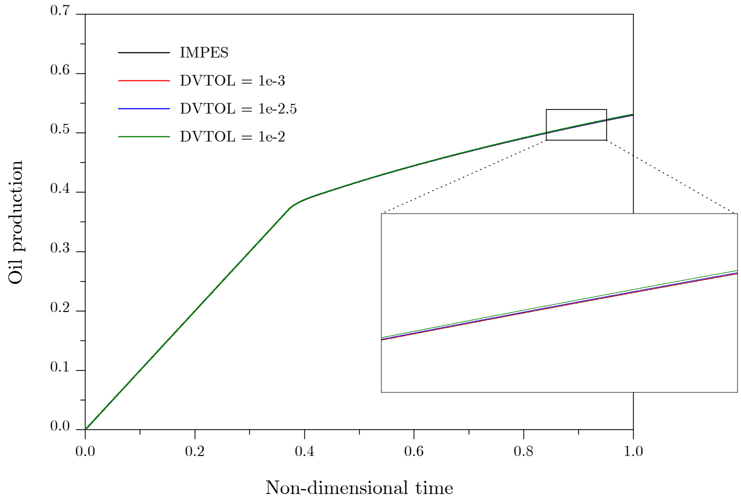

Figure 16 – Comparison of accumulated oil production curves.

6. CONCLUDING REMARKS

A modified version of the IMPES algorithm aiming an improvement in its performance has been presented in this work. The basic strategy consists in advancing pressure equation with larger timesteps than saturation equation in order to reduce the amount of computation associated with the solution of linear system of equations for the pressure. The novel feature of the algorithm proposed herein is an adaptive control strategy in order to systematize the determination of adequate timesteps for pressure. Moreover, the pressure timestep variation is associated to the mean variation of total velocity field, attempting to maintain it constant along a simulation. The results presented show that remarkable reduction in the computation time can be obtained using the proposed modified IMPES algorithm, without introducing significant additional error into the numerical solutions of displacement processes.

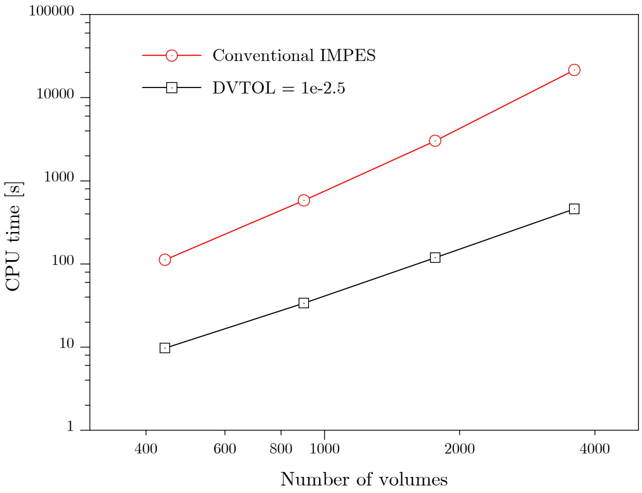

Figure 17 – Comparison of performances of the conventional IMPES and the accelerated IMPES with DVTOL = 1e-2.5, for different grid sizes.

7. ACKNOWLEDGEMENTS

This work was supported by Agência Nacional do Petróleo (ANP), through PRH - Programa de Formação de Recursos Humanos em Petróleo e Gás (PRH 09, Universidade Federal de Santa Catarina).

8. REFERENCES

Aziz, K., Settari, A., 1979. *Petroleum reservoir simulation*. Applied Science Publishers Ltd., London, England.

Chen, Z., Huan, G., Li, B., 2004. *An improved IMPES method for two-phase flow in porous media.* Transport in Porous Media, Vol. 54, pp. 361-376.

Coats, K. H., 2001. *IMPES stability: the CFL limit*. SPE Paper 66345, presented at the SPE Reservoir Simulation Symposium, Houston/TX, USA, February 11-14.

Hurtado, F. S. V., 2005. *An element-based finite volume formulation for the simulation of two-phase immiscible flow in porous media*. M.Sc. Dissertation (in Portuguese), Mechanical Engineering Department, Federal University of Santa Catarina, Brazil.

Hurtado, F. S. V., Maliska, C. R., Silva, A. F. C., Cordazzo, J., Ambrus, J. & Contessi, B. A., 2004. *An element-based finite volume formulation for simulating two-phase immiscible displacements in core samples*. Proceedings of the 10th Brazilian Congress of Thermal Sciences and Engineering, November 30-December 3, Rio de Janeiro, Brazil.

Hurtado, F. S. V., Maliska, C. R., Silva, A. F. C., Cordazzo, J, 2005. *An element-based finite volume formulation for reservoir simulation*. Proceedings of the XXVI Iberian Latin-American Congress on Computational Methods in Engineering – CILAMCE 2005, Gurarpari/ES, Brazil, October 19-21.

Maliska, C. R., 2004. *Transferência de calor e mecânica dos fluidos computacional*, 2ª edição revista e ampliada. Livros Técnicos e Científicos Editora S. A., Rio de Janeiro, Brasil.

Mattax, C. C. and Dalton, R. L., 1990. *Reservoir Simulation*. SPE Monograph Series, Volume 13, Society of Petroleum Engineers.

Peaceman, D. W., 1977. *Fundamentals of numerical reservoir simulation*, In Developments in Petroleum Science, Vol. 6, Elsevier Scientific Publishing Company.

Saad, Y., 2003. *Iterative methods for sparse linear systems*. Second edition, Society for Industrial and Applied Mathematics, Philadelphia, USA.