CMNE/CILAMCE 2007
Porto, 13-15 Junho, 2007
© APMTAC, Portugal 2007

ADDITIVE CORRECTION MULTIGRID APPLIED TO PETROLEUM RESERVOIR SIMULATION

Jonas Cordazzo*, Leonardo Karpinski, Susie Keller, António F. C. Silva, Clovis R. Maliska

SINMEC – Laboratory of Numerical Simulation in Fluid
Dynamics and Computational Heat Transfer
Department of Mechanical Engineering
Federal University of Santa Catarina - UFSC
Zip Code: 88040-900 Florianopolis, SC, Brazil
email: jonas@sinmec.ufsc.br web: http://www.sinmec.ufsc.br

Key words: multigrid, reservoir simulation, unstructured grids, finite volume methods.

Summary. *In recent years, multigrid methods have received considerable attention due to its capability to reduce drastically the computation time in the solution of linear systems. In this article the Additive Correction Multigrid method is analyzed and applied to petroleum reservoir simulation problems, employing structured and unstructured grids. The results of test-problems shown in this paper are strongly encouraging and demonstrate that the multigrid methods can be used with great advantage in reservoir simulation too, similarly to other areas of engineering where this kind of method has been routinely used.*

1. INTRODUCTION

It is very well known that around 70% of the computational effort spent to obtain a numerical solution is consumed in the solution of the linear systems. Some solvers are robust and efficient only for the solution of linear systems with small number of unknowns, but the effort can exponentially increase with the grid size. The multigrid solvers, on the other hand, theoretically present linear effort with the grid size, and have a great potential to be explored.

The efficiency of a multigrid method can be understood through the spectral analysis of solution errors. These errors can be represented superposing waves of different frequencies ([1][2] and others). One can show that only the errors with wavelength of same order of the grid size are properly reduced by iterative solvers. The other errors, with lower frequencies relatively to this grid, are hardly reduced causing a poor rate of convergence, requiring a prohibitive amount of computational work to significantly advance to the exact solution.

This observation leads to the need of different grids to efficiently reduce the entire error frequency spectrum. Utilizing successively coarser grids, the errors that have wavelength with the size of the grid are eliminated very fast in each one the grids. This procedure accelerates the solution because errors of all frequencies are efficiently eliminated.

1.1. A brief review of literature

The multigrid methods started to be investigated in 1960´s with the Fedorenko [3] and Bakhvalov [4] researches. However, the good characteristics of these methods were pointed out only with Brandt [5] in 1970´s and 1980´s with the geometric multigrid. In this type of multigrid, the coarsening process is strictly done based on geometrical aspects of the grid, and there is the need of discretization on coarses grids too.

The difficulties related to the discretization on all grids motivated the development of algebraic multigrid methods [6][7], which do not demand any information about the grid and the discretization scheme. In these methods it suffices to have the coefficients, the independent vector and the initial estimative of variables. The fundamental idea is that this method may be a black box solver that is not linked to the numerical procedures.

The algebraic multigrid based on the additive correction was developed based on the Settari and Aziz´s works [8] , and it differs from the previous ones because physics is taken into account in its conception, due to the enforcement of the conservation of properties within each coarse grid cell [9][10]. The coefficients of coarse grid cells are obtained by summing up the fine grid equations, following a procedure that does not demand the need of fixed stencils. This is an important characteristic that enables this method to be applied to more complex situations, as when unstructured grids are employed.

The robustness of the additive correction multigrid, referred herein as ACM, is pointed out in the work of Raw [11], where different Navier-Stokes problems are solved without modification of solver parameters. Elias [9] showed that even for cases with anisotropic coefficients, the ACM can work very well, since the agglomeration scheme takes into account the different time scales. Elias [12] showed also that the iterative methods suffer convergence problems when the discretized equations have anisotropic coefficients. He proposed to form

the coarse grid aiming to reduce the different information time scales caused by the anisotropy.

In petroleum reservoir simulation, the multigrid methods were applied with important results [13]-[16], even though the most of commercial simulators do not have the option of this kind of solver yet. The lack of flexibility of classical multigrid methods, mainly the geometric ones, has caused difficulties for application in real cases.

1.2. Scope of this work

As already seen in the previous section, the ACM has a great potential that motivates its investigation and application in petroleum reservoir simulation. As we will see, this method can be employed successfully in solving the linear systems provided by the classical Finite Volume Method (FVM) [17][18], which employs structured grids (also called finite difference methods in petroleum literature), and the Element-based Finite Element Method (EbFVM), which uses unstructured grids in a conservative framework [18].

Initially, the multigrid method will be described, pointing out the ACM concepts. The coarse grid equations are derived, the cycles are defined and some details of the computational implementation are discussed. Finally, some examples of two-phase (water and oil) flow and two-dimensional domain are shown in order to demonstrate the potentialities of the method outlined here, comparing different multigrid cycles and iterative solvers.

2. MECHANISMS INVOLVED IN A MULTIGRID METHOD

There are three main mechanisms in a multigrid method:

1. Transfer of information from the coarse to the fine grid;
2. Coarse grid solution;
3. Transfer of information from the fine to the coarse grid.

Even though, at a first glance, it seems to be more intuitive to have these mechanisms in the reverse order, that is, from 3 to 1, it is more didactic, for the ACM method, to describe them in this order.

2.1. Transfer of information from the coarse to the fine grid

This procedure is often called “prolongation” and consists of updating the estimative of fine grid variable using a correction value aiming to eliminate the low frequency errors. In ACM, it is proposed that the coarse grid provides a vector with corrections to be added in the estimative of fine grid. The increment will be of the type

$$\Phi_P^* = \Phi_P + \delta_I \tag{1}$$

where Φ_P^* is the improved solution of variable Φ calculated for node P of fine grid, and will be used as estimative in the next iteration, Φ_P is the last estimative of this variable in

the fine grid, that is, the most recent one, and δ_I is the correction coming from the coarse grid. In this equation, the subscript P represents each one of control volumes composing the cell I. Figure 1 shows an example of the coarsening process, where we can see the fine and the coarse grids, as well as the cell and control volumes definition.

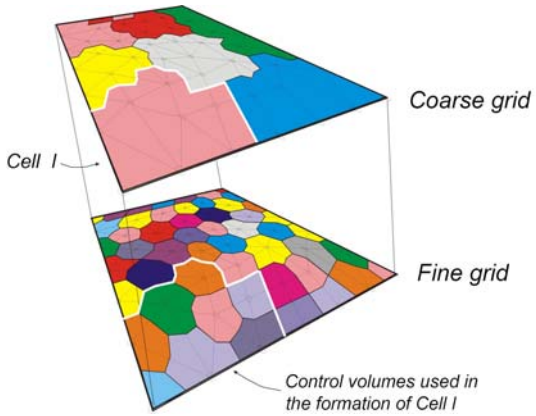


Figure 1. Coarsening process

Note that, even though the discretized equations are obtained using a method based on elements – EbFVM, these equations in the fine grid are the result of the enforcement of conservation principles over control volumes. This is the essence of every finite volume method [17][18]. In ACM it is desired to extend these principles to the coarse cells also, what justifies the agglomeration of control volumes in Figure 1, instead of elements (triangles), what seems to be easier to do. The control volumes are formed in EbFVM joining lines from the triangle barycenter to the median edges. More details about this method and the control volumes construction are out of the scope of this work and can be obtained elsewhere [18][19].

2.2. Coarse grid solution

In order to obtain consistent correction values (δ_I), it is reasonable establish equations in the coarse grids strongly linked with the physics. This is the reason why in most multigrid methods the governing equations are discretized in the coarse grids too. However, the algorithm associated to this practice is complex, producing inconsistencies with the grids, especially with unstructured ones. To avoid these difficulties, it is employed a different procedure in ACM, outlined as follows.

The variable δ_I is defined such that the residual of each control volume equation in the fine grid becomes zero. This is, of couse, the goal of any multigrid method, as well as any solver. A condition that can be explored is the fact that when the solution of linear system is exact, the residual of any equation is zero, as well as the piecewise sum of these residuals. In other words, the summing of residuals of any control volumes in the domain must be zero

$$\sum_P \mathsf{R}_P^* = 0 \tag{2}$$

where R_P^* is the most recent residual equation of control volume P and, due to the discretization using FVM or EbFVM, will have the form

$$\mathsf{R}_P^* = b_P - A_P \Phi_P^* - \sum_{nb \text{ of } P} A_{nb} \Phi_{nb}^* \tag{3}$$

where nb represents the P neighbor nodes, A is the coefficient matrix and b is the independent vector.

Substituting the variable Φ_P^* of Eq. (1) in (3), we have

$$\mathsf{R}_P^* = b_P - A_{P,P}\left(\Phi_P + \delta_I\right) - \sum_{nb \text{ of } P} A_{P,nb}\left(\Phi_{nb} + \delta_{nb}\right) \tag{4}$$

However, recognizing the expression of for R_P in the previous equation, given by

$$b_P - A_{P,P}\Phi_P - \sum_{nb \text{ of } P} A_{P,nb}\Phi_{nb} = \mathsf{R}_P \tag{5}$$

Eq. (4) can be written as

$$\mathsf{R}_P^* = \mathsf{R}_P - A_{P,P}\delta_I - \sum_{nb \text{ of } P} A_{P,nb}\delta_{nb} \tag{6}$$

Given some agglomeration scheme to form the cell I, and summing their residuals according to Eq. (2), one obtains

$$0 = \sum_{P \in I}\left\{\mathsf{R}_P - A_{P,P}\delta_I - \sum_{nb \text{ of } P} A_{P,nb}\delta_{nb}\right\} \tag{7}$$

This equation can be written in the usual form by

$$A_{I,I}\delta_I + \sum_{NB \text{ of } I} A_{I,NB}\delta_{NB} = b_I \tag{8}$$

where NB indicates the neighbor cells of I in the coarse grid, and

$$A_{I,I} = \sum_{P \in I} A_{P,P} + \sum_{P \in I}\left\{\sum_{(nb \text{ of } P) \in I} A_{P,nb}\right\} \tag{9}$$

$$A_{I,NB} = \sum_{P \in I} \left\{ \sum_{(nb \text{ of } P) \notin I} A_{P,nb} \right\} \tag{10}$$

$$b_I = \sum_{P \in I} \{R_P\} \tag{11}$$

The solution of the linear system of Eq. (8) will provide the corrections to be summed to the fine grid solution, and that will satisfy the condition establish in Eq. (2). This solution can be obtained using any iterative solver, as the Gauss-Seidel or even a direct solver. The coefficient determination represented by Eqs. (9) to (11) requires a very simple procedure, even for unstructured grids, since it consists only on summing coefficients of the fine grid.

This procedure of obtaining the coarse grid equations has an important physical support. Elias [20] demonstrated that it is equivalent to enforce the flux conservation at the interfaces of coarse grids. This is highly desirable because the finite volume methods are based on conservation principles. Therefore, all equations must be written using the conservative approach. However, it is important to mention also that the procedure just described is not equivalent to discretize the coarse grid equations, what we wanted to avoid.

2.3. Transfer of information from the fine to the coarse grid

This process is usually called restriction or interpolation. However, differently of most classical methods, these names do not apply to the multigrid in consideration. The reason is that in the ACM there is no transfer of primary variables information from the fine to the coarse grid. Instead, as mentioned in the previous section, the fine grid only provides the coefficients and the equation residuals to be summed in each cell.

3. MULTIGRID CYCLES

Often, the use of only two grids – one fine and one coarse – is not enough to speed up the convergence, since errors of all frequencies are not properly eliminated with only two grids. So, it is common to employ several coarse grids so that the several frequency modes can be eliminated. The order in which each grid is used in the solution procedure is defined as a multigrid "cycle", which can be flexible or fixed.

In the flexible cycles, the algorithm chooses, as function of the convergence behavior, whether the solver has to iterate more in a grid or not. The number of sweeps of the solver is variable in each grid. The main drawback of this cycle is the difficult of establishing suitable and general criteria able to guide the decision of moving from one grid to another.

The fixed cycles, on the other hand, are built defining previously the sequence of different grids and the number of sweeps in each one. The fixed cycles most used are the V, W and F, and are shown in Figure 2 for different levels (quantity of grids). Note that these three cycles are identical for the cases where there are only two levels. With three levels the W and F-cycles are still identical, while for other cases these three cycles are

completely different. The configuration of each cycle for more quantities of levels depicted in Figure 2 can be obtained following the same formation rule.

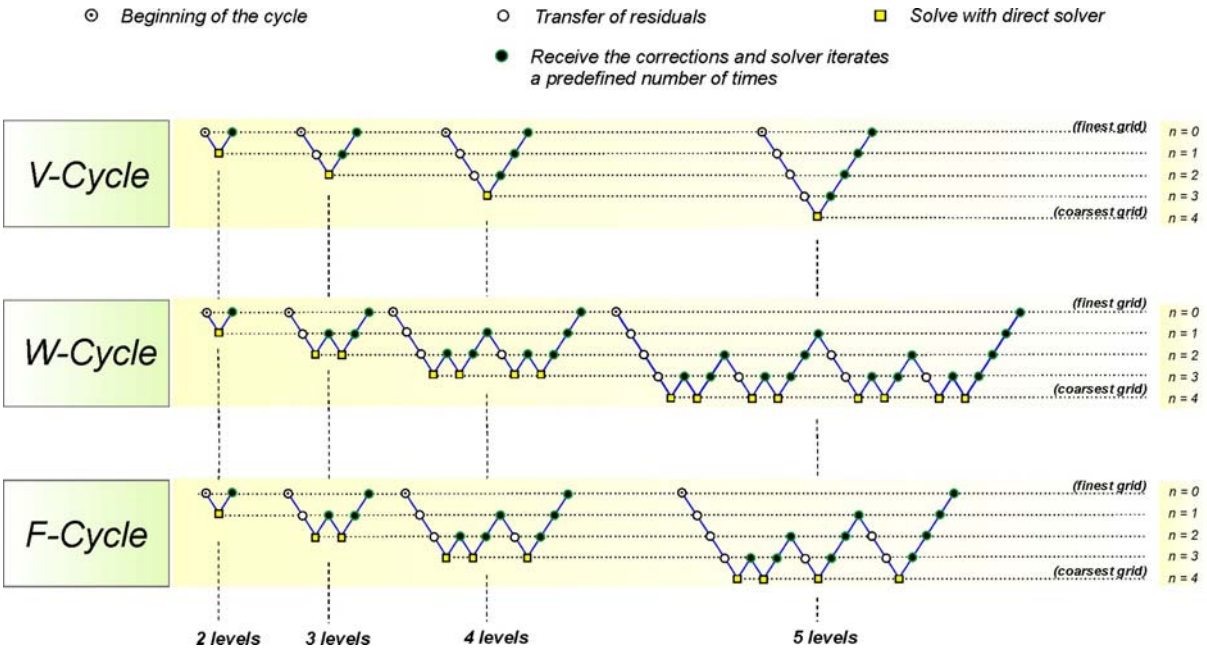


Figure 2. Multigrid cycles with different levels (quantity of grids), where *n* indicates the level number

The computational effort for solving the linear system is proportional in each grid with V-cycle, while using the W and F-cycles more effort is spent in the coarse grids. As the coarse grid solutions are quickly obtained, it is expected that the multigrid methods that employ W and F-cycles be faster than the multigrid methods that use V-cycle. However, independently of cycle type, often is utilized a direct solver to obtain the solution in the coarsest grid (identified with a square in Figure 2), because the linear system of this grid is relatively small.

Therefore, we can summarize Figure 2 saying that when the level number (n) increases, the information from the fine to the coarse grid occurs through the transfer of the the residuals (section 2.3). When n decreases, the solution of coarse grid is carried out (section 2.2), followed by the prolongation procedure, which is the transfer of the variable correction from the coarse to the fine grid (section 2.1).

4. AGGLOMERATION PROCESS

The efficiency of multigrid method depends decisively on how the control volumes are agglomerated in the coarse grid formation. As the same correction is applied to all variable inside an agglomerated control volume, a good practice would be to join control volumes demanding similar correction values. This indicates, for instance, that the inclusion, in a cell, of control volumes whose variables have values significantly different should be avoided. This

strategy does not apply if a block o control volumes contains a geological faults, in which pressure fields may not be connected.

A criterion based on the analysis of coefficients can be a good measure to lead the agglomeration process. If the coefficient connecting two control volumes *i* and *j*, for instance, is smaller than the coefficient connecting control volumes *j* and *k*, probably the expected values for a variable calculated in *j* and *k* are more similar than the values of *i* and *j*. Therefore, the control volumes *j* and *k* would be more adequate to be in the same cell than the control volumes *i* and *j*.

Moreover, an agglomeration scheme based on coefficients can contribute for the reduction of discrepancies between the time step implied by an iterative solver, and the time scale required for the transport of the physical information between nodes in the domain [20]. The difference among coefficients appears due to the geometrical and physical anisotropy, and frequently cause the solver stalling. This motivates the use of this strategy in this work.

The coefficient analysis employed here is based on two rules. The first one determines if a control volume *P* neighbor of a cell block may be agglomerated to it. If yes, the second rule determines if there is another cell block, neighbor of this same control volume, which has better conditions of agglomerating it. These rules can be written in a pseudo code form as

$$\textit{Rule 1:}\ \max\left(A_{i,P}, A_{P,i}\right) \geq \alpha \max\left(A_{i,h}, A_{h,i}\right) \tag{12}$$

$$\textit{Rule 2:}\ \max\left(A_{i,P}, A_{P,i}\right) \geq \alpha \max\left[\max\left(A_{P,nb}, A_{nb,P}\right)\right]_{(nb \text{ of } P)\notin I} \tag{13}$$

where α is a factor. The value $\alpha = 1/2$ worked very well in the cases analyzed. The subscripts *P*, *i*, *h* and *nb* represent control volumes in a situation shown in Figure 3, so that *P* is the index of the control volume that can be joined to a cell I , *h* means a reference control volume within the cell, *i* is the neighbor control volume of *P* located inside the cell block *I*, and *nb* represents all neighbor control volumes of *P* located outside of this cell block.

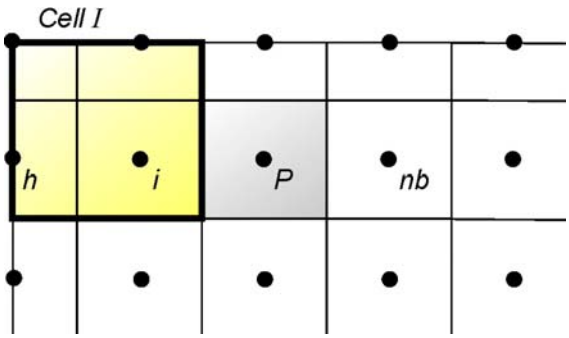


Figure 3. Control volumes involved in the formation of a cell *I*

It is important to note that the terms related to advective flux, often calculated using upwind (UDS) schemes, originate asymmetric coefficients, i.e., $A_{i,P}$ may be significantly

different from $A_{P,i}$. Even though this situation creates a strong connection in one direction and a weak in the other, it does not cause difficulties in the use of rules 1 and 2, because they are based on the largest coefficient value. Other agglomeration scheme guidelines can be found in [19][20], although there is no definite rules for agglomeration.

The agglomeration algorithm outlined here is recursively used to form all grid levels. The last level will be the one that has few cells, that is, a small number defined by the user. In this work this number was 50.

5. COMPUTATIONAL IMPLEMENTATION

The ACM solver was implemented in C++ using the object-oriented programming, polymorphisms, templates etc. [21][22]. Although, this kind of programming model does not seem as efficient as the classical structural programming, the support of this choice comes from the fact that the algorithm becomes easily readable, of simple maintenance and reusability.

The main idea is to make each grid level be a computational object composed by matrices, vectors and scalars. This solver was implemented considering block matrices, composed by smaller matrices side-by-side. Implicit methods usually employed in reservoir simulation produce this kind of matrix. In this case, each linear system of coarse grids is also composed by block matrices, and its agglomeration process may follow the same guidelines applied to the matrices without blocks (previous section). One has, however, to define which parameter has to be used in Eqs. (12) and (13), to decide to which cell of a coarse grid a control volume belongs. An illustration of different coefficients within the blocks can be seen in Eq. (14).

$$\begin{pmatrix} \begin{bmatrix} \frac{\partial R_{T_1}}{\partial p_1} & \frac{\partial R_{T_1}}{\partial s_{w1}} \\ \frac{\partial R_{w_1}}{\partial p_1} & \frac{\partial R_{w_1}}{\partial s_{w1}} \end{bmatrix} & \cdots & \begin{bmatrix} \frac{\partial R_{T_1}}{\partial p_n} & \frac{\partial R_{T_1}}{\partial s_{wn}} \\ \frac{\partial R_{w_1}}{\partial p_n} & \frac{\partial R_{w_1}}{\partial s_{wn}} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} \frac{\partial R_{T_n}}{\partial p_1} & \frac{\partial R_{T_n}}{\partial s_{w1}} \\ \frac{\partial R_{w_n}}{\partial p_1} & \frac{\partial R_{w_n}}{\partial s_{w1}} \end{bmatrix} & \cdots & \begin{bmatrix} \frac{\partial R_{T_n}}{\partial p_n} & \frac{\partial R_{T_n}}{\partial s_{wn}} \\ \frac{\partial R_{w_n}}{\partial p_n} & \frac{\partial R_{w_n}}{\partial s_{wn}} \end{bmatrix} \end{pmatrix} \begin{Bmatrix} \begin{Bmatrix} \Delta p_1 \\ \Delta s_{w1} \end{Bmatrix} \\ \vdots \\ \begin{Bmatrix} \Delta p_n \\ \Delta s_{wn} \end{Bmatrix} \end{Bmatrix} = -\begin{Bmatrix} \begin{Bmatrix} R_{T_1} \\ R_{w_1} \end{Bmatrix} \\ \vdots \\ \begin{Bmatrix} R_{T_n} \\ R_{w_n} \end{Bmatrix} \end{Bmatrix} \tag{14}$$

This linear system is the result of the application of Newton´s Method to a two-phase flow Darcy's equations, where the first line of each block contains the derivatives of total conservation equations residuals, while the second line carries the derivatives of water conservation equations residuals [23]. These derivatives are with respect to pressure and water saturation (p and s_w). The key question is to define which coefficient has to be used in the rules 1 and 2.

In this work, the agglomeration process employs the magnitude of the coefficient

multiplying Δp, i.e., the value $\partial R_T / \partial p$ in each block. The reason is that the pressure equation expresses the total conservation, and therefore is more representative than the water conservation equation. Besides, the coefficient multiplying Δp is present in flux term, instead of the coefficient of Δs_w directly involved in the transient term, which affects only the diagonal coefficients. The choice of pressure coefficients for the agglomeration algorithm is also the procedure when dealing with the Navier-Stokes equation, since pressure acts over the flow, and in some sense carries the influence of all variables.

6. RESULTS

This section is divided into two subsections, according to the numerical formulation employed to solve the non-linear system [23]:

(1) Test-cases using the IMPES (*IMplicit Pressure, Explicit Saturations*) method, which result in a linear system for pressure;
(2) Test-cases using fully implicit method (Newton´s Method), which produce a linear system with a block matrix containing two equations and variables for each node. These variables are pressure and water saturation variations (cf. Eq. (14)).

In each case, the performance of different solvers is examined comparing the CPU effort for solving the first linear system of the transient simulation. It seems reasonable to apply the conclusions to the other time levels of the simulation.

The test-problem to be analyzed is the very well known ¼ of five-spot [23]. This geometry was discretized using the classical diagonal grid, that is, Cartesian formed by equally spaced quadrilaterals. The discrete solution was obtained integrating the conservation equations using two methods: the Finite Volume Method (FVM) and the Element-based Finite Volume Method (EbFVM) [18], in order to compare the multigrid's efficiency for each method. The former uses only structured grids, producing matrices with well defined band structure.. The EbFVM is a method that combines the flexibility of using unstructured grids with the concept of assembling by elements, largely used in Finite Element Methods, enforcing local and global conservation.

In each discretized problem, a reference solution is produced by solving the linear system with a very severe convergence criteria. The solution of each solver was considered converged when the solution satisfied some specified criteria when compared withy the reference solution. The default parameters utilized in the multigrid method in most of the cases in this paper are shown in Table 1.

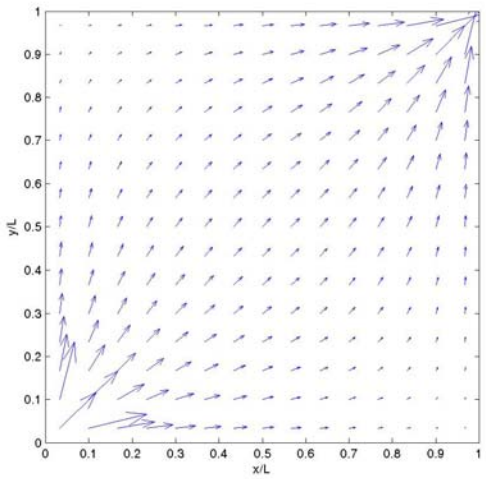


Figure 4. Total velocity field obtained using a Cartesian grid with 15x15 quadrilateral elements

Cycle type	*F*
Iterative solver	Gauss-Seidel (GS)
Number of solver iterations in each grid	2
Number of control volumes in each cell	≈ 7
Maximum number of cells to solve directly	50

Table 1. Default parameters of the solver

6.1. Test-cases using IMPES algorithm

Figure 5 compares the CPU effort of different solvers to obtain the converged solution for the test-case just described. These solvers are Gauss-Seidel, GMRES (preconditioned with SSOR - Symmetric Successive Over Relaxation) and ACM multigrid (within a Gauss-Seidel solver, as already mentioned in Table 1). CPU efforts for solving the linear systems obtained using FVM and EbFVM are also presented in Figure 5. The CPU was a Pentium 4, 1.7 GHz, 1 GB RAM.

It can be seen in Figure 5 that the ACM multigrid shows a linear increasing in the CPU effort as a function of the number of the unknowns. In theory, this behavior is what is sought for any solver. As expected, the Gauss-Seidel solver presented a quadratic increase of the CPU effort with the number of variables. GMRES behavior lies among the two just mentioned.

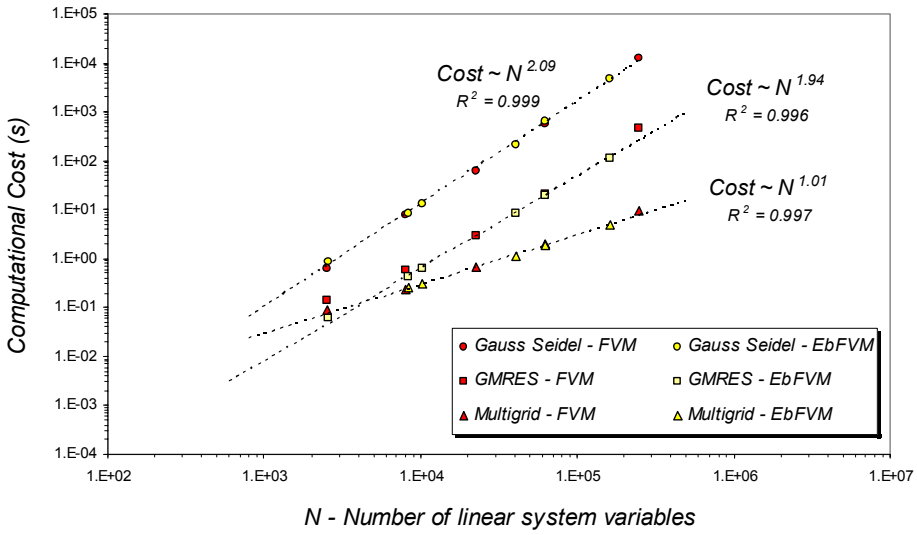
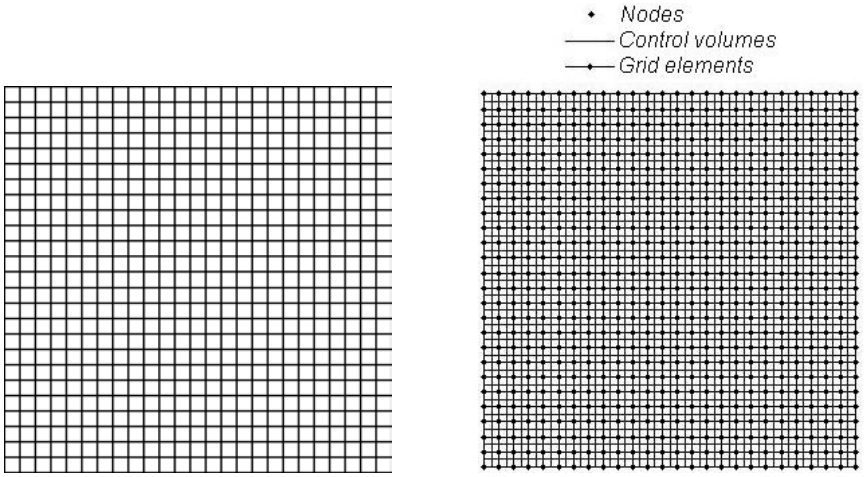


Figure 5. Performance of different solvers in the solution of linear system provided by Finite Volume Method (FVM) and the Element-based Finite Volume Method (EbFVM)

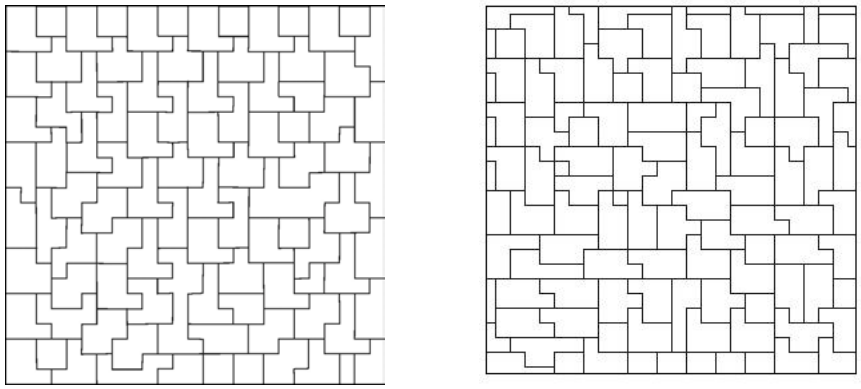
The results using FVM and EbFVM (the mains difference is the type of the grid) are almost identical, demonstrating that the use of unstructured grids does not cause any difficulty in the convergence rate.

Considerations on the agglomeration. In the ACM multigrid, the grids are simply abstractions, since they are only used to graphically characterize the connectivity of the coarse grid equation. However, in order to have an idea of the shape of cells in coarse grids, Figure 6 presents two levels generated when a Cartesian grid of 25x25 control volumes using the FVM and a grid of 625 elements using the EbFVM are employed. In this figure, the coarse grids are formed joining 7 control volumes in each cell. In the EbFVM, the control volumes are built around nodes, while in the FVM where the control volumes and grid elements are the same geometrical entities. As already stated, the ACM multigrid creates coarse grids always joining control volumes, no matter the numerical method employed. It is important to note that, even though the EbFVM grid seems structured in this figure, the node and element numeration is arbitrary, what define this grid as unstructured, since the resulting matrix obeys the nodes ordination.



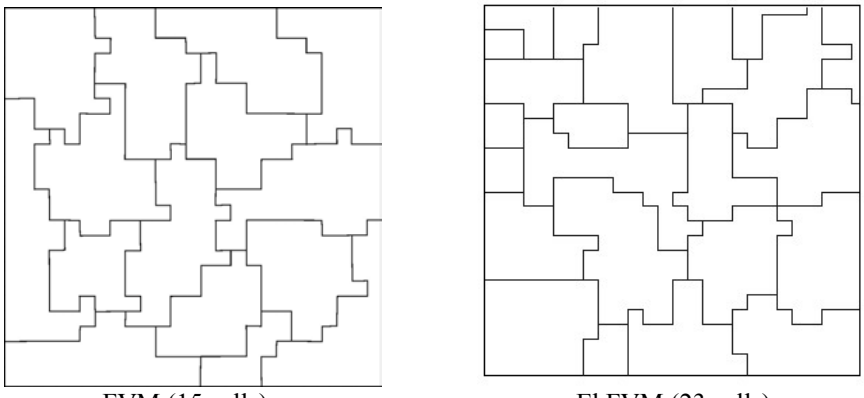
FVM (25x25 control volumes) EbFVM (625 elements and 676 nodes)

(a) Level 0



FVM (96 cells) EbFVM (111 cells)

(b) Level 1



FVM (15 cells) EbFVM (23 cells)

(c) Level 2

Figure 6. Fine grid *(a)* and coarse grids generated by ACM multigrid (*b* and *c*) utilized in the solution of the problem described in Figure 4 employing different methods (FVM and EbFVM)

Cycles and iterative solvers. Now, the performance of ACM multigrid for reservoir simulation problems will be analyzed employing different cycles and iterative solvers. For the comparisons, a grid with 62500 elements (structured with 250x250) and 63001 nodes will be used in the EbFVM method, with the convergence criterion used in the last section. Figure 7 presents the behavior of Euclidian norm residual during the solution of linear system, using different iterative solvers in the multigrid: Gauss-Seidel (GS) and ILU(0). Usually the ILU (Incomplete LU factorization) is used as a preconditioner, but in this work it works as an iterative solver [18][19], contributing to improve the multigrid results, as will be seen. To simplify the comparisons, the residual was normalized by the residual calculated using the initial estimative of variables.

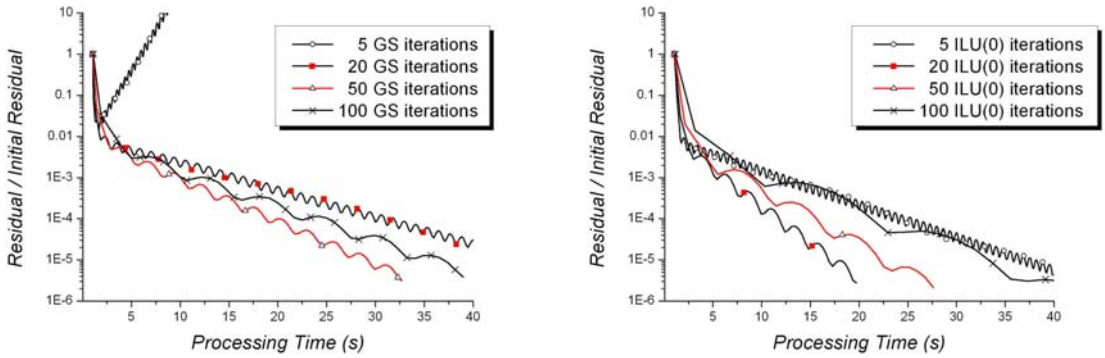
As can be seen in Figure 7, the multigrid performance practically does not changed using W and F-cycles, even with different iterative solvers and different fixed iterations. Employing the V-cycle, however, the ACM multigrid requires more effort to reach the same results. In addition, convergence problems were verified with this cycle, mainly when few GS iterations were performed in each level. On the other hand, using the iterative solver ILU(0), the multigrid residual with V-cycle reduced faster than using GS. This behavior did not match with the results obtained using W and F-cycles, in which the multigrid performance increased slightly utilizing GS. The monotonic residual reduction is an outstanding feature of the ACM using W and F-cycles, as shown in Figure 7.

6.2. Test-cases using Fully Implicit algorithm

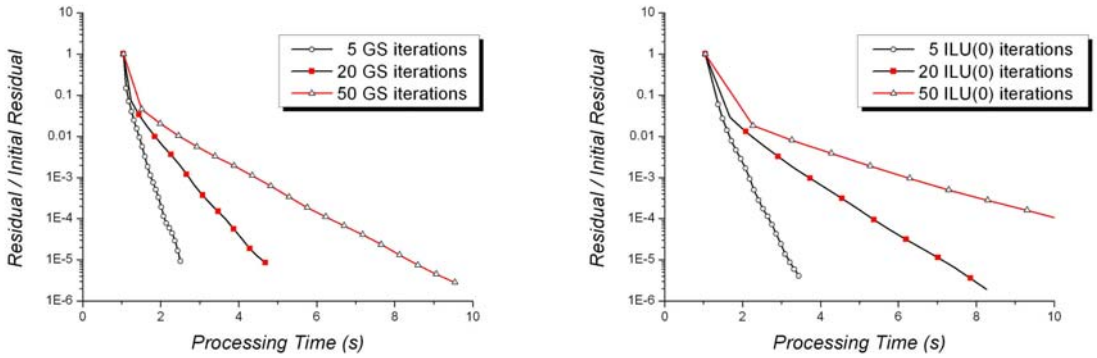
The multigrid performance in the solution of linear systems composed by block matrices is the topic of this section. As already stated, the Fully Implicit (FI) formulation produces block matrices of (2x2) for two-phase flow in porous media. The test-problem analyzed is the same of previous section (¼ of five-spot), as well are the multigrid parameters (Table 1).

Although not demonstrated in this paper, the agglomeration algorithm for block matrices produced coarse grids very similar to the ones produced in the IMPES case. This shows that the agglomeration scheme employed is adequate, because, in fact, the coarse grid geometry should not depend on how the non-linear discretized equations are solved.

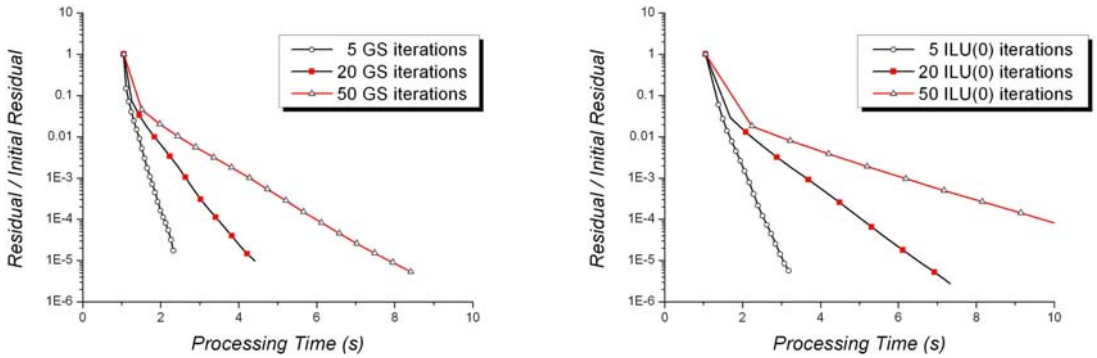
Figure 8 compares the CPU effort of the ACM multigrid for the IMPES and the FI methods. For this comparison, a given computational cost is obtained using different fine grids in each method, because the FI produces matrices with double size when compared to the ones when the IMPES is used. Overall, the computational effort trend is identical for these methods, showing a slightly larger increase for the FI method. The overhead costs, such as determining block matrices in each coarse grids, summing coefficients in each block, and others demands to deal with block matrices, justify the FI's additional work.



(a) V-Cycle



(b) W-Cycle



(c) F-Cycle

Figure 7. Convergence of ACM multigrid employing several cycles, iterative solvers (GS - Gauss-Seidel and ILU(0)) and fixed number of iterations in each level

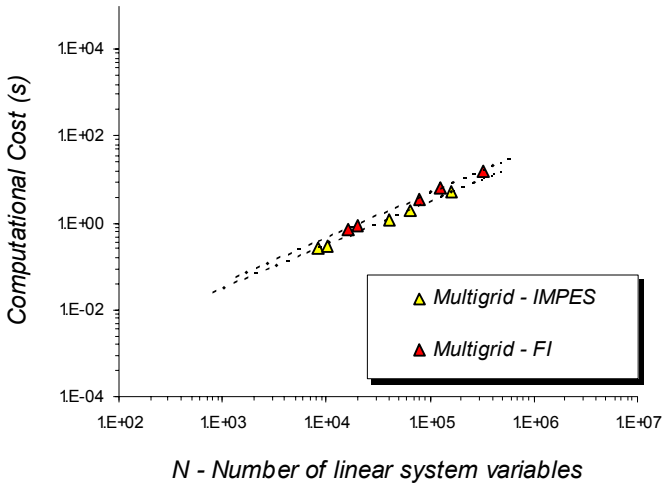


Figure 8. Multigrid computational effort in solving linear systems provided by IMPES and FI (Fully Implicit) formulations

7. CONCLUSIONS

The ACM multigrid implemented and tested in this work has shown to be appropriated to solve the linear systems generated by the most used formulations (IMPES and FI) in petroleum reservoir simulation. The outstanding feature of this solver, comparing with standard iterative solvers, is the linear behavior of the computational effort with the grid size.

Different multigrid cycles were examined, and the F-cycle multigrid presented the smallest CPU time with the desired monotonic residual reduction for the cases analyzed. In spite of the limited number of test-problems considered here, the robustness of ACM multigrid could be noticed. Its performance practically did not changed, solving scalar or block matrices. The agglomeration process created adequate coarse grids, independently whether the fine grid is structured or not. This avoids the need of constructing coarse cells using fixed stencils, a common practice that limit the applicability of classical multigrid methods.

ACKNOWLEDGMENTS

The authors are grateful to Agência Nacional do Petróleo (ANP), through its PRH-Program, through a scholarship for the first author.

REFERENCES

[1] W.L. Briggs, V.E. Henson and S. Mccormick, *A multigrid tutorial*, 2nd edition, SIAM Books, Philadelphia ,(2000).

[2] U. Trottenberg, C. Oosterlee, A. Schüller, A. Brandt., P. Oswald and K. Stüben, *Multigrid*, Academic Press, Inc., Orlando, (2001).

[3] R.P. Fedorenko, "The speed of convergence of one iteration process", *USSR Comput. Math. Math. Phys.*, Vol. **4**, pp. 227-235, (1964).
[4] N.S. Bakhvalov, "On the convergence of a relaxation method under natural constraints on an elliptic operator", *USSR Comput. Math. Math. Phys*., Vol. **6**, pp.101-135, (1966).
[5] A. Brandt, Multi–level adaptive technique (MLAT) for fast numerical solution to boundary value problems, in *Proceedings* of the Third International Conference on Numerical Methods in Fluid Mechanics, H. Cabannes and R. Teman, eds., Vol. **18** of Lecture Notes in Physics, Springer–Verlag, pp. 82–89, Berlin, (1973), *apud* [2].
[6] A. Brandt, S.F. Mccormick, and J.W. Ruge, Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations, Technical Report, Institute for Computational Studies, Colorado State University, (1982), *apud* [7].
[7] A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, T.A. Manteuffel, S.F. Mccormick, G.N. Miranda, and J.W. Ruge,. "Robustness and scalability of algebraic multigrid", *SIAM J. Sci. Comput*. Vol. **21**, Dec, pp.1886-1908, (1999).
[8] A. Settari and K. Aziz, "A generalization of additive correction methods for the iterative solution of matrix equations", *SIAM Journal of Numerical Analysis*, Vol. **10**, pp. 506 – 521, (1973).
[9] B.R. Hutchinson and G.D. Raithby,. "A multigrid method based on the additive correction strategy", *Numerical Heat Transfer*, Vol. **9**, pp. 511-537, (1986).
[10] J.P. Van Doormaal, A. Turan and G.D. Raithby, Evaluation of new techniques for the calculation of internal recirculating flows, in *Proceedings* of the AIAA 25th Aerospace Sciences Meeting, January 12-15, Reno, (1987).
[11] M. Raw, Robustness of coupled algebraic multigrid for the navier-stokes equations, in *Proceedings* of the AIAA 96-0297, 34th Aerospace and Sciences Meeting & Exhibit, January 15-18, Reno, (1996).
[12] S.R. Elias, G.D. Stubley, and G.D. Raithby, "An adaptive agglomeration method for additive correction multigrid", International Journal for Numerical Methods in Engineering, Vol. **40**, pp. 887-903, (1997).
[13] P. Bastian and R. Helmig, "Efficient full-coupled solution techniques for two-phase flow in porous media – parallel multigrid solution and large scale computations", *Advances in Water Resources*, Vol. **23**, pp. 199-216, (1999).
[14] A. Silva, A.L.G.A. Coutinho and I. Parsons, "Métodos multigrid para a simulação por elementos finitos de escoamentos imiscíveis em meios porosos", *Métodos Numéricos para Cálculo y Diseño en Ingeniería*, Vol. **16**, pp. 311-326, (2000).
[15] H. Klie, M.F. Wheeler, T. Clees and K. Stüben, Deflation AMG solvers for highly ill-conditioned reservoir simulation problems, *SPE paper* 105820 presented at the Reservoir Simulation Symposium, 26-28 February, Houston, Texas, (2007).
[16] K. Stüben, T. Clees, H. Klie, B. Lu, and M.F. Wheeler, Algebraic multigrid methods (AMG) for the efficient solution of fully implicit formulations in reservoir simulation, *SPE paper* 105832 presented at the Reservoir Simulation Symposium,

26-28 February, Houston, Texas, (2007).
[17] S.V. Patankar, *Numerical heat transfer and fluid flow*, Hemisphere Publishing Corporation, Washington, (1980).
[18] C.R. Maliska, *Heat transfer and computational fluid flow*, 2nd. edition in Portuguese, Livros Técnicos e Científicos Editora, Rio de Janeiro, (2004).
[19] J. Cordazzo, Petroleum *reservoir simulation using the element-based finite volume method and algebraic multigrid*, Ph.D. Thesis in Portuguese, Federal University of Santa Catarina, Florianopolis, Brazil, (2006).
[20] S.R. Elias, *Enhancements to additive correction multigrid*, Master thesis, University of Waterloo, Canada, (1993).
[21] N.M. Josuttis, *The C++ standard library - a tutorial and reference*, Addison-Wesley, USA, (1999).
[22] B. Stroustrup, *The C++ programming language*, Special Edition. Addison Wesley. USA ,(2000).
[23] K. Aziz and A. Settari, *Petroleum reservoir simulation*, Applied Science Publishers, London, (1979).
[24] E.F. D'Azevedo, P.A. Forsyth and Wei-Pai Tang, Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems, *SIAM J. Matrix Anal. Appl.*, Vol. **13**, pp. 944-961, (1992), *apud* [20].