*Proceedings of the 11th Brazilian Congress of Thermal Sciences and Engineering -- ENCIT 2006*
*Braz. Soc. of Mechanical Sciences and Engineering -- ABCM, Curitiba, Brazil,- Dec. 5-8, 2006*

**Paper CIT06-686**

# COMPARATION BETWEEN SOLVERS USED IN ADDITIVE CORRECTION MULTIGRID METHOD WITH UNSTRUCTURED GRIDS

**Susie C. Keller**
susie@sinmec.ufsc.br

**Jonas Cordazzo**
jonas@sinmec.ufsc.br

**Clóvis R. Maliska**
maliska@sinmec.ufsc.br

**Lynceo F. Braghirolli**
lynceo@sinmec.ufsc.br

SINMEC – Laboratory of Numerical Simulation in Fluid Dynamics and Computational Heat Transfer, Dept. of Mechanical Engineering, Federal University of Santa Catarina
Florianopolis, SC, Brazil

***Abstract.*** *This paper shows a study about Additive Correction Multigrid method. The general motivation of multigrid methods is to solve efficiently a large algebraic equations set resulting from the discrete approximation of differential equations using several meshes in order to improve the convergence rate of iterative methods reducing all frequency modes of the error. The Additive Correction Multigrid (ACM) has several advantages over the traditional ones: the discretization is made only on the fine grid, reducing the complexity and the computational cost, as well as eliminating the possibility of inconsistencies among grids; it is a conservative method and it allows the use of an adaptive agglomeration cell scheme. However, the performance of any multigrid method is directly associated with the type of solver used in the several mesh levels. In this paper is studied the use of Gauss-Seidel or ILU to solve the linear system of different levels of ACM. The coarsest level equations, in turn, are solved with a direct-LU solver because the linear system is small becoming possible to obtain the exact solution. Another comparation between the adaptive aglomeration cell scheme and the geometric aglomeration cell scheme is made too. The Element-based Finite Volume Method (EbFVM) is the numerical method chosen to discretize the heat conduction equation in quadrilateral unstructured grids.*

*Keywords. Additive Correction Multigrid (ACM), ILU solver, Adaptive agglomeration scheme, Element- based Finite Volume Method (EbFVM), Unstructured grids.*

## 1. Introduction

In the mathematical and engineering literature several types of multigrid methods have been presented. The philosophy and theory behind such one was presented initially by Brandt (1977). The basic idea of any multigrid strategy is to use a sequence of grids with classical iterative solvers to accelerate the convergence. This is achieved because is possible to smooth the high-frequency modes of the error for each grid. Brandt decomposed the error of the approximate solution to a set of discrete equations in Fourier modes and analyzed how the solvers smooth the various modes. On the finest (original) grid, the high-frequency modes of the error are effectively reduced with classical iterative solvers, but the low-frequency modes are difficult to remove, since they would require coarser grids to attain this goal. In the coarse grids, the low-frequency modes of the error become high-frequency modes being easily removed by the linear solver. Pratical applications and demonstrations of multigrid methods can be found in Briggs (2000) and Trottenberg et al. (2001).

The multigrid methods can be classified in two types: geometric and algebraic ones.The geometric multigrid methods present reasonable performance when the coefficients of the algebraic equations are nearly isotropic and homogeneous, but these methods perform poorly when the coefficients are strongly anisotropic. In the literature we can find several examples of geometric multigrid applications (Brandt, 1977; Vanka, 1986; Volker, 1996, Wesseling and Oosterlee, 2001; Rabi and Lemos, 2003; Mesquita and Lemos, 2005).

The algebraic multigrid methods, on the other hand, appeared to solve the problem related with anisotropic coefficients. Hutchison and Raithby (1986), Elias (1993), Venkatakrishnan and Mavriplis (1996), Francescatto and Dervieux (1998), Stüben (1999, 2001) among others have shown that the choice of block shape can have a drastic effect

on the convergence. Thus, the way as the cells are agglomerated in the fine grid to form the next coarse grid demands special consideration, and has been received lately in this kind of methods.

This article deals with one type of the algebraic multigrid scheme, namely the Additive Correction Multigrid method (Hutchinson and Raithby, 1986), simply called here as ACM. In this method, the coarse grid-equations are generated by the summation of the finer grid equations. Once the solutions on the coarser grid are obtained, they are simply added to the solution on the finer grid. Thus, interpolation or extrapolation operators, commonly used by multigrid classical methods, are not required.

The effects of the use of adaptive agglomeration in problems with anisotropic coefficients and a comparison of the performance of multigrid cycles V, W and F performance applied to unstructured meshes are presented in Keller at al (2004). These authors pointed out the flexibility of the method.

One relevant aspect to be analyzed in this paper is the base solver used to solve the several grid levels. An adequate solver will improve the convergence rate. The most common solvers used in multigrid methods are Gauss-Seidel and Incomplete Lower Upper factorization - ILU. In this work they are tested in order to point out their characteristics and to show the situations where each one is more suitable. For solving the coarsest mesh equations, the LU-direct solver is used because the linear system is small.

As ACM provides the use of adaptive agglomeration scheme besides the geometric agglomeration scheme, one study of these different schemes is also performed in this work. The former scheme consists in adding fine-grid cells with small transport timescales in order to reduce the variation in timescales for propagation of information on a coarse-grid cell, while the last scheme consists in joining the cells in order to form regular blocks.

In all cases analyzed here, the ACM method will be applied to solve the heat conduction problems whose differential equations are discretized by the Element based Finite Volume Method (EbFVM). This numerical method, which is suitable to deal with unstructured grids, will be shortly described here too.

## 2. The additive correction multigrid method

The Additive Correction Multigrid method (ACM) distinguishes from conventional multigrid methods for generating the coarse grid equations without the use of fixed stencils. In ACM, the coarse grid equations are formed by adding the fine-grid equations, resulting in the determination of constant corrections in coarse grid cells, which are summing to the fine grid solution in order to accelerate the convergence.

To exemplify the ACM, an unstructured mesh is depicted in Fig. (1). Even though the blocks may be formed in any convenient way, in this example the volumes are joined in blocks with five cells on average. We can see that the new coarse-grid blocks are formed by joining fine-grid control volumes in different directions, resulting in an agglomeration with highly unstructured forms.

Initially, we can considerate the system of equations to be solved written in the common form as

$$A_p^i \phi_i = \sum_{nb} A_{nb}^i \phi_{nb} + b_i \,, \tag{1}$$

where $A_p^i$ is the central coefficient of the control volume $i$ considered in the discrete equation for $\phi$, $A_{nb}^i$ are the coefficients connecting the control volume $i$ to the neighbors control volumes, $b_i$ is the source coefficient and $\phi$ is the solution. The coefficients of Eq. (1) can be obtained by applying Finite Volume Method (FVM) or Element based Finite Volume Method (EbFVM) to either structured or unstructured grids (Maliska, 2004).

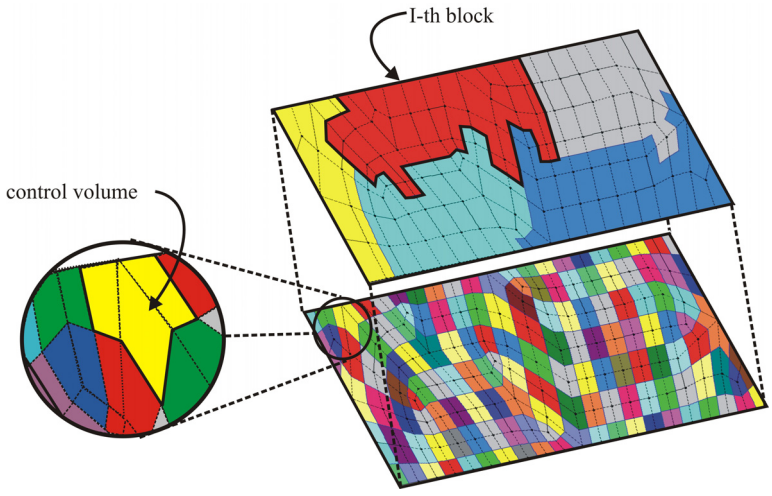



Figure 1. Example of a block agglomeration scheme in unstructured grids.

The core of ACM is the definition of a correction equation that has the role of adding coarse-grid corrections ($\phi^*$) to the best estimate of $\phi$ on the fine grid, as

$$\tilde{\phi}_i = \phi_i + \phi^*_{I,i}\,, \tag{2}$$

where $\tilde{\phi}_i$ is the improved solution in each cell and $\phi^*_{I,i}$ is the correction related to all volumes that lie within the *I* block.

Requiring that the residual be zero for the corrected solution, one obtains the linear system for $\phi^*_{I,i}$ (Hutchinson and Raithby, 1986)

$$A^*_p \phi^*_p = \sum_{nb} A^*_{NB} \phi^*_{NB} + b^*_p\,, \tag{3}$$

where

$$b^*_p = \sum_{i,I} r_i\,, \tag{4}$$

$$A^*_P = \sum_{i,I} A^i_p - \sum A^i_{nb}\,, \tag{5}$$

$$A^*_{NB} = \sum_{i,I} A^i_{nb}\,. \tag{6}$$

In these equations $\sum A^i_{nb}$ denotes the connection among the control volumes inside of one same block and $\sum_{i,I} A^i_{nb}$ represents the connection among volumes of neighbor blocks.

Equation (3) must be solved in order to obtain the correction $\phi^*_{i,I}$. This correction is added to each $\phi$ value of control volumes that lie within the *I*-block, Eq. (2) Thus, the improved estimate $\tilde{\phi}_i$ is obtained, as already stated. Additional details of the ACM method can be found elsewhere (Hutchinson and Raithby, 1986; Maliska, 2004).

As can be seen, using the ACM method, there are no further decisions to be made related to the boundary conditions, transfer of residuals or interpolation of variables.

In the next section the different agglomeration schemes used in this paper are presented.

## 3. Agglomeration scheme

The manner of joining cells in multigrid methods to obtain the coarse grids can affect their efficiency. Moreover, considering the use of unstructured grids, it is expected that this is not a simple task. Following two forms of performing the agglomeration is presented.

### 3.1 Adaptive agglomeration scheme

The nature of the coefficients in the fine mesh equations can strongly affect the convergence behavior of an iterative solver. One can show that there is a natural timescale for a perturbation to travel from a node to the neighbor node (Elias, 1993). This timescale is inversely proportional to the coefficient that connects two nodes. The iterative solver has a time step that approaches a converged solution by propagating disturbances through the solution domain. This time step is defined as the smallest timescale to propagate the information between nodes. The performance of iterative solver degrades when there is a high difference between the solver time step and the timescales to propagate the information between nodes. In order to accelerate the convergence of the discrete linear equations set solution, we can generate a coarse grid equations set by adding cells to reduce the differences between the largest and smallest transport time scales (Elias, 1993). Therefore, an adaptive agglomeration method based on fine mesh coefficients should be used.

The principle of an adaptive agglomeration scheme is to add fine grid cells with small timescale of the transport process (large coefficients), obtaining cells with small variation in the timescale of transport information process.

The adaptive agglomeration scheme begins with a single fine grid cell, and through a set of rules it is determined which of the neighbors should be included in the coarse grid cell. The new members of the coarse cell are examined to determine which of their neighbors should be included, and this is done either until the coarse cell reach the desired size

or there are no more neighbors to be added. Then, another coarser grid can be built agglomerating the new cells in the same way, and so on.

Following the nomenclature of a typical family tree as proposed by Elias et al (1997), we call the fine grid cell, whose neighbors are being examined, as the father, and its neighbors included in the same coarse cell as its children. The current father's father is known as the grandfather. This nomenclature is very convenient because the growth of a coarse cell occurs in a tree-like fashion.

There are two main rules used to decide which fine grid cells are added to the new coarse grid cells considered (Elias, 1993). The first agglomeration rule states that the coefficient that connects the father with the possible son must be of the same order or larger than half of the coefficient that connects the father with the grandfather. Representing the father by the index *f*, the possible son by *s* and the grandfather by *g*, we have that *s* is a possible son of *f* if:

$$\max\left(a_{fs}, a_{sf}\right) \geq \max\left(a_{gf}, a_{fg}\right)/2 \,. \tag{7}$$

The second rule states that a cell is excluded if the interface timescale is very large. The concept of what would be "large" is defined relative to all other timescales which affect the cell in question. A possible son can be agglomerated with the father if the coefficient that connects the father with the son will be of the same order or larger than half of the coefficient that connects the possible son with its neighbors. Thus, taking the coefficients again and representing the possible son's neighbors by the index *n*, *s* is agglomerated with *f* if:

$$\max\left(a_{fs}, a_{sf}\right) \geq \max\left(a_{sn}, a_{ns}\right)/2 \,. \tag{8}$$

In order to improve the efficiency of this agglomeration scheme some details must be also considered. For instance, to simplify the choice of the first fathers for coarse grid cells and ensure that all cells are agglomerated in an only sweep of the fine grid, the first fathers are selected in the same order of sweep of the iterative solver. Moreover, as the initial father does not have any grandfather and any preferential timescale, the rule one can be ignored.

It is possible to find in the end of this process remaining cells in the fine grid without any neighbor that can be used as a son. To eliminate wasted computational effort with those single cells, any father cells which do not have children are forced to join the coarse grid cell of their most strongly connected neighbor.

### 3.2 Geometric agglomeration scheme

The geometric agglomeration scheme consists in adding fine grid cells in order to obtain, as much as possible, regular coarse blocks with good aspect ratio. Unlike the previous scheme, the cells coefficients are not important in this task. A cell is added in a block if it is a neighbor of other cell or cells that lie in this block.

## 4. The linear solvers used in ACM

The kind of solver used in each grid level is other important decision. A good relaxation scheme is one that is well able to reduce error components in the large coefficient direction of a wavelength extending as far as possible. Three different solvers will be considered here: Lower Upper (LU) factorization to the coarsest grid, and Gauss-Seidel and Incomplete Lower Upper factorization – ILU to the other grids.

### 4.1 Lower Upper Factorization (LU)

The LU factorization is a direct method, where the *U* matrix is obtained with the same algorithm used in Gauss Elimination and the *L* matrix is constructing with the multiplicative factors used in the building of the *U* matrix.

Considering the linear system

$$[A][\phi] = [L][U][\phi] = [B] \tag{9}$$

where *A* is the coefficient matrix, *B* is the independent terms vector, the $\phi$ is unknowns vector, *L* is the lower triangular matrix and *U* is the upper triangular matrix.

The system of Eq. (9) can be solved finding *D* in the following linear system:

$$[L][D] = [B] \tag{10}$$

and, after, determining the unknown $\phi$ with the Eq. (11)

$$[U][\phi] = [D]. \tag{11}$$

As the LU factorization deals with the full matrix, it is useful only to solve small linear systems. For this reason this solver is used only in the coarsest grid of ACM method.

The iterative linear solvers used in other grid levels of ACM are presented in the next section.

### 4.2 Gauss-Seidel

This method belongs to the point-to-point iterative methods because it solves the linear system in serial manner, iteratively, using the values of variables at the previous iterative level and the values already calculated. The point-to-point iterative methods have convergence extremely slow when a large equation system must to be solved. However, they have been applied in multigrid methods due to the fact that there is no need of obtaining the complete convergence in each grid level (only a correction factor is calculated).

The iterative process consists in the calculation of $\phi_i$, Eq. (1) using an initial estimative. After, the convergence must be checked and this procedure must be repeated case the convergence has not been reached. Other possibility is the procedure of Gauss-Seidel to be repeated a certain number of times, This number is defined by the user and it occurs when Gauss-Seidel is used as the solver base of the ACM method.

### 4.3 Incomplete Lower Upper Factorization (ILU)

This method is usually used as preconditioner for non-stationary methods like GMRES, conjugated-gradient and others. However, due to its strong implicit characteristics, the Incomplete Lower Upper Factorization – ILU can be used also as a base solver in multigrid methods. This method deals with a storage system where only no-null values are saved, and consequently, only these values can be modified by LU factorization, justifying the name: "Incomplete". Some details about how it works are given in sequence.

Considering the linear system:

$$[A][\phi] = [B], \tag{12}$$

where $A$ is the coefficient matrix, $B$ is the independent terms vector and $\phi$ is unknowns vector.

The philosophy of ILU methods is given by

$$[L][U] = [A] + [A']. \tag{13}$$

where $L$ is the lower triangular matrix, $U$ is the upper triangular matrix and $A'$ is a matrix created to accelerate the convergence and that do not influence in the final solution.

When the LU factorization originates the $A$ matrix, the method is a direct method, but how is created the matrix $A'$, the method is iterative and has the following form

$$[A + A'][\phi]^{k+1} = [A + A'][\phi]^{k} - \left\{[A][\phi]^{k} - [B]\right\} \tag{14}$$

where $k$ is the iterative level. When the solution converged is obtained, the second term of the right hand of Eq. (14) becomes zero, resulting in $\phi^{k+1} = \phi^{k}$.

Defining a correction to $\phi$ given by

$$[\delta]^{k+1} = [\phi]^{k+1} - [\phi]^{k} \tag{15}$$

and the solution residue as

$$[R]^{k} = [A][\phi]^{k} - [B] \tag{16}$$

we can substitute Eq. (15) and Eq. (16) in Eq. (14) resulting in

$$[A + A'][\delta]^{k+1} = -[R]^{k}, \tag{17}$$

and substituting the Eq. (13) in Eq. (17), we obtain

$$[L][U][\delta]^{k+1} = -[R]^k . \tag{18}$$

which is a equation to determine the correction. The solution is obtained through two process of successive replacements, since the matrices $L$ and $U$ are lower triangular and upper triangular matrices, respectively. Thus, defining a new vector $V$, it can be determined by

$$[L][V] = -[R]^k \tag{19}$$

and knowing $V$, we can obtain $\delta$ by

$$[U][\delta]^{k+1} = -[V]^k . \tag{20}$$

Thus, $\phi$ is updated in the following form

$$[\phi]^{k+1} = [\phi]^k + [\delta]^{k+1} . \tag{21}$$

The residue is calculated again by Eq. (16) and the iterative process follows until the residue will be less than tolerance specified, or for a definite number of times.

In the next section, it is shown the method used to discretize the differential equation in all problems analyzed in this paper.

## 5. Element-based Finite Volume Method (EbFVM)

In a finite volume methodology the computational domain is covered by non-overlapping control-volumes where the balances are made. In the EbFVM, the element concept and the geometric representation inherited from the finite element method are utilized, and its definition precedes the creation of the control volume. Figure (2) shows an example of an EbFVM grid, emphasizing the differences between grid elements (entity defined by the nodes) and control-volumes (built around the nodes). Besides, it is shown the integration points where the conservation equations are integrated.

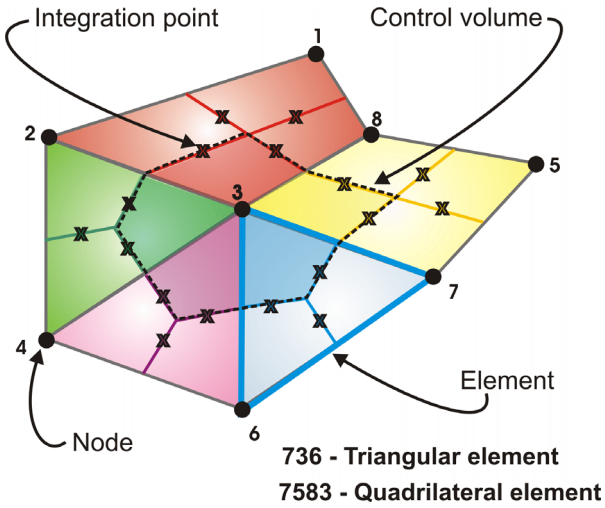



Figure 2. Control volume in a unstructured grid of triangular and quadrilateral elements.

The control volumes are created by the method of the medians, which consists of joining the center of the elements to their medians. The resulting control volume is formed by portions (sub-control volumes) of neighboring elements.

More details involving the deduction of EbFVM discretized equations can be found elsewhere (Raw, 1985; Maliska, 2004).

## 6. Results

Two example problems will be presented to demonstrate the efficiency of the ACM method using different base solvers and agglomeration schemes. Considering grids with 1024, 4096, 16384 and 65536 nodes for these two problems, each block is agglomerated with four cells on average in each level. Depending on the number of cells that the original grid has, the number of levels is defined in order to obtain always the coarsest grid with around 64 cells, which is solved directly with LU method. The maximum allowed RMS residue to the solution in the original grid was $10^{-5}$. The results are obtained with the use of the V, W and F cycles (Trottenberg et al., 2001) in all cases analyzed here.

### 6.1 Comparison between the results from ACM with different base solvers

**Test-Case 1.** The first problem considered is a diffusion problem with *x*-boundaries of the domain insulated and the *y*-boundaries partly insulated, as represented in Fig. (3). The EbFVM was used for discretizing this problem because it treats the grids in an unstructured fashion, even though it is used here grids geometrically structured with nodes or control-volumes equally in both *x* and *y* directions. Because of the problem geometry, the coefficients related to nodes in the interior of domain are greater in the *y* direction than in *x* direction.

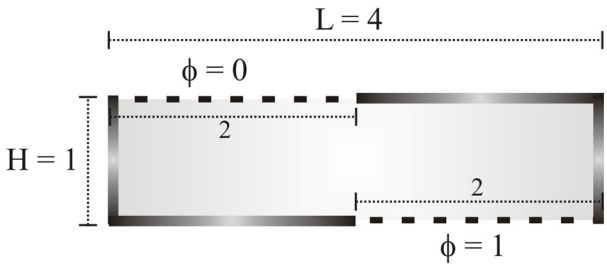
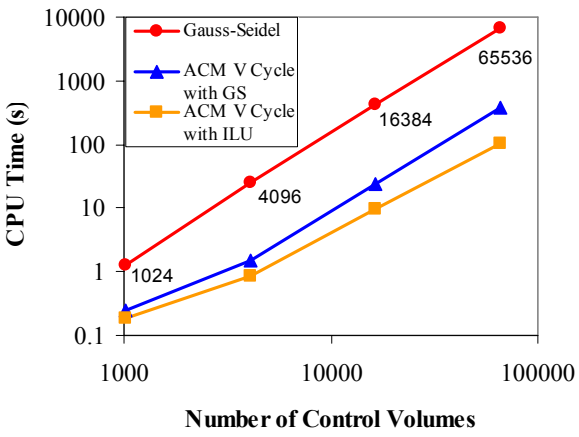
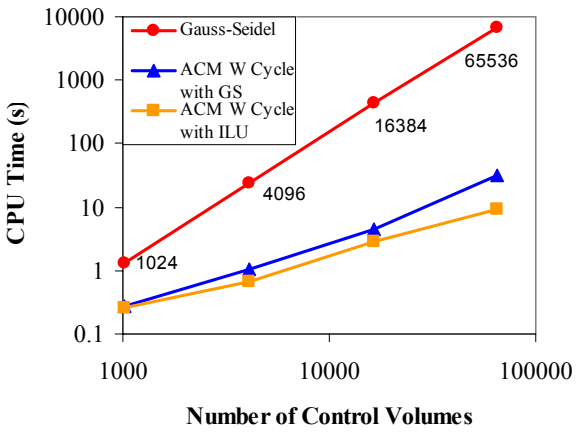



Figure 3. Geometry of the first example problem.

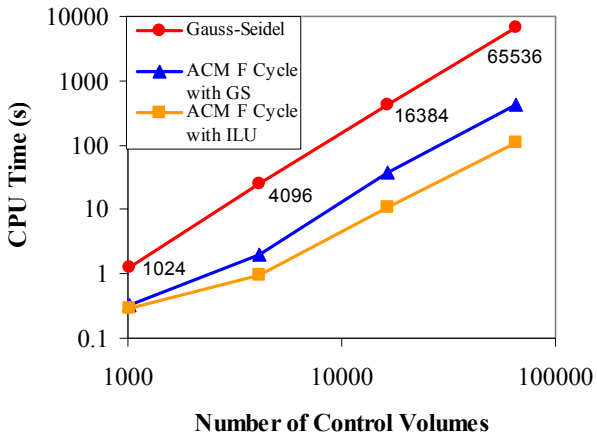
In the results presented in Fig. (4) is used the adaptive agglomeration scheme. Thus, the agglomeration is always made in the direction of the greater coefficients. We can see for the cycles V, W and F, that the ACM method with ILU solver presented the smallest computational cost.




(a)




(b)




(c)

Figure 4. Comparison between iterative solvers used in ACM method to the first conduction problem in (a) V cycle, (b) W cycle and (c) F cycle, in relation to the traditional Gauss-Seidel solver.

**Test-Case 2.** The second example is similar to the first one, except for the boundary conditions on $x$ and $y$ coordinates that are interchanged, as shown in Fig. (5). In this example, the information about the boundary conditions is transmitted mainly in the direction of smaller coefficients, so we expect a larger computational effort required for Gauss-Seidel to reach the convergence.

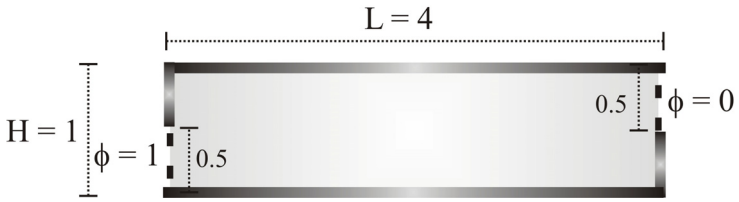
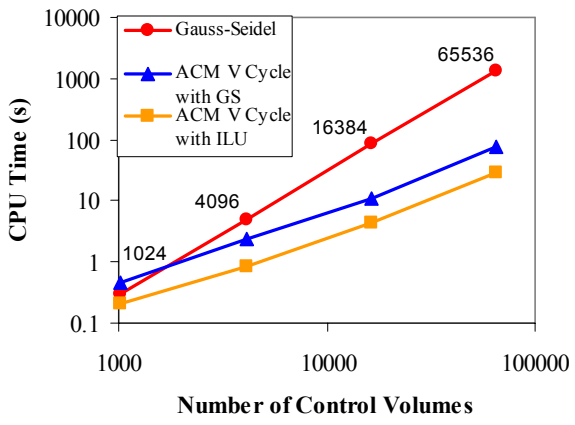
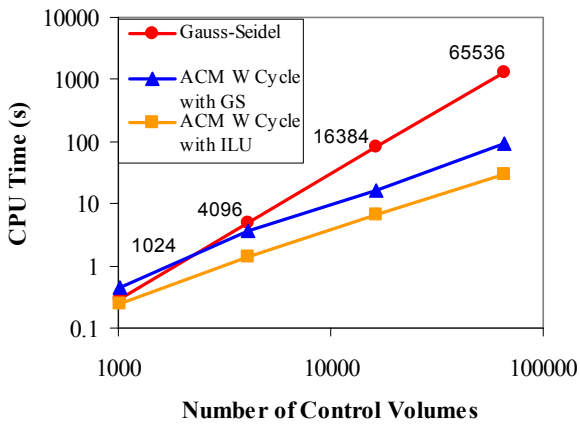



Figure 5. Geometry of the second example of diffusion problem.

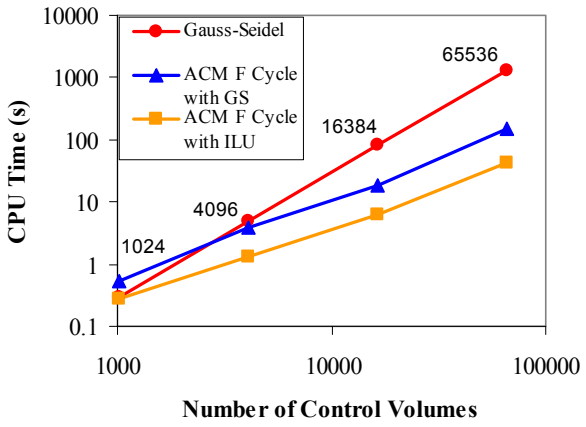
The results presented in Fig. (6) show again that the ILU solver used in ACM method has the smallest computational cost and thus it is better than ACM with Gauss-Seidel and the Gauss-Seidel alone, despite of the convergence behavior to be different of the first example because the boundary conditions are interchanged. Here, is used too the adaptive agglomeration scheme.




(a)




(b)




(c)

Figure 6. Comparison between iterative solvers used in ACM method to the second conduction problem in (a) V cycle, (b) W cycle and (c) F cycle, in relation to the traditional Gauss-Seidel solver.

**Discussion.** We can see too that the ACM W cycle showed the better performance in the problem 2 (Fig. (6.b)), while in problem 1 the three cycles presented similar performance (Fig. (4)).

In problem 1, Fig. (4), the traditional Gauss-Seidel solver had a good performance only in the mesh with 1024 cells. It occurs because the coefficients are large in the *y* direction and the effect of the Dirichlet boundary conditions is rapidly transmitted to interior of domain. Thus, the residue to be eliminated in the *x* direction is small and Gauss-Seidel is sufficient to solve this diffusion problem in relatively coarse meshes. However, with the increasing of the grid size, it is necessary the use of ACM method to reduce the computational cost.

In Fig. (6), we can see the high computational time gain of ACM over Gauss-Seidel solver in all grids considered. It occurs because the Dirichlet boundary conditions are prescribed in the weak coefficient direction (see Fig. (5)). Thus, Gauss-Seidel solver propagates the boundary condition information in this direction one node per iteration, while ACM propagates the information over the whole domain in only one iteration.

### 6.2 Comparison between the results from ACM with different agglomeration schemes

Now, we will compare two forms of performing the agglomeration: adaptive and geometric, both explained in section 3. From now on, the ACM method with ILU solver is the default, since this showed to be the best configuration to the ACM method in the cases analyzed. The two cases considered earlier and described in Figs. (3) and (5) will be considered again. As the only difference among them is the boundary conditions, their coarse grids, being calculated either by the adaptive or geometrically, must be the same and are presented in the sequence.

At first, in Fig (7) is shown the original grid with 1024 cells, where the control volumes are represented by the blue lines, the nodes by the circles and the elements by the hatched lines.

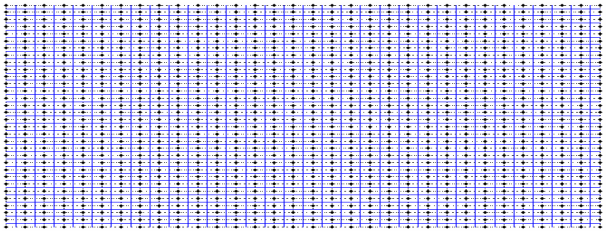
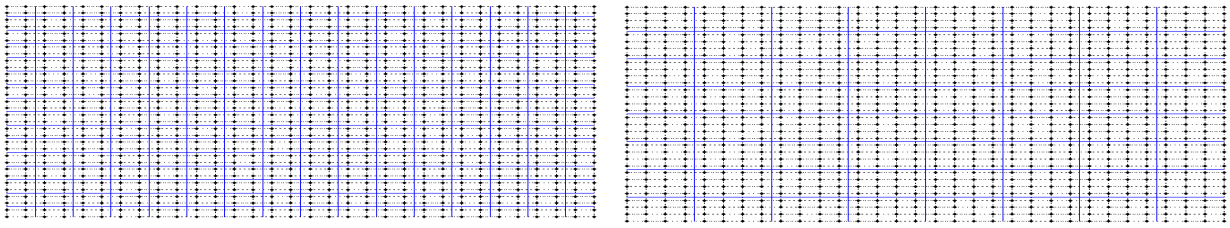


Figure 7. Original grid of two examples with 1024 cells.

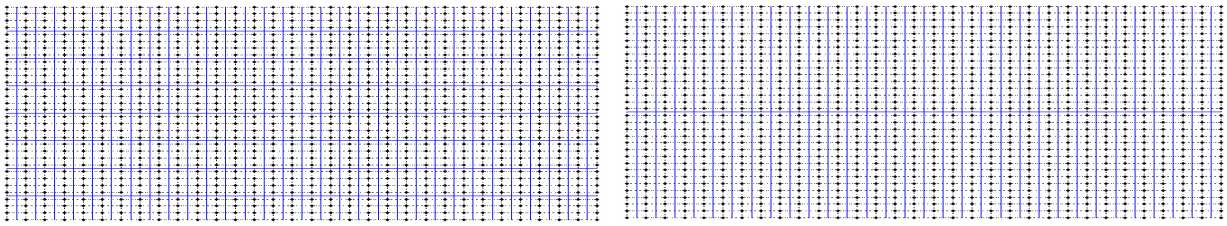
Figure (8) shows two levels of geometric agglomeration with four cells, resulting in 256 new coarse cells, Fig. (8.a) and 64 new coarse cells, Fig. (8.b).

The two coarse grids, with 256 and 64 cells, determined following the rules of the adaptive agglomeration are shown in Fig. (9). Since this problem presents anisotropic coefficients due to the geometry with the greater coefficients in y direction, the adaptive agglomeration will occur in that direction, as shown in Figs. (9.a) and (9.b).



(a) (b)

Figure 8. Geometric agglomeration with (a) 256 cells and (b) 64 cells.



(a) (b)

Figure 9. Adaptive agglomeration with (a) 256 cells and (b) 64 cells.

In the graphics presented in Figs. (10) and (11) are shown the total CPU time needed to reduce the RMS residual to a pre-determined value (in this case $10^{-5}$) in different size grids obtained using the ACM with adaptive agglomeration, ACM with geometric agglomeration and an iterative solver, in this case Gauss-Seidel. Taking the example 1 for each cycle we obtain the results showed in Fig. (10).

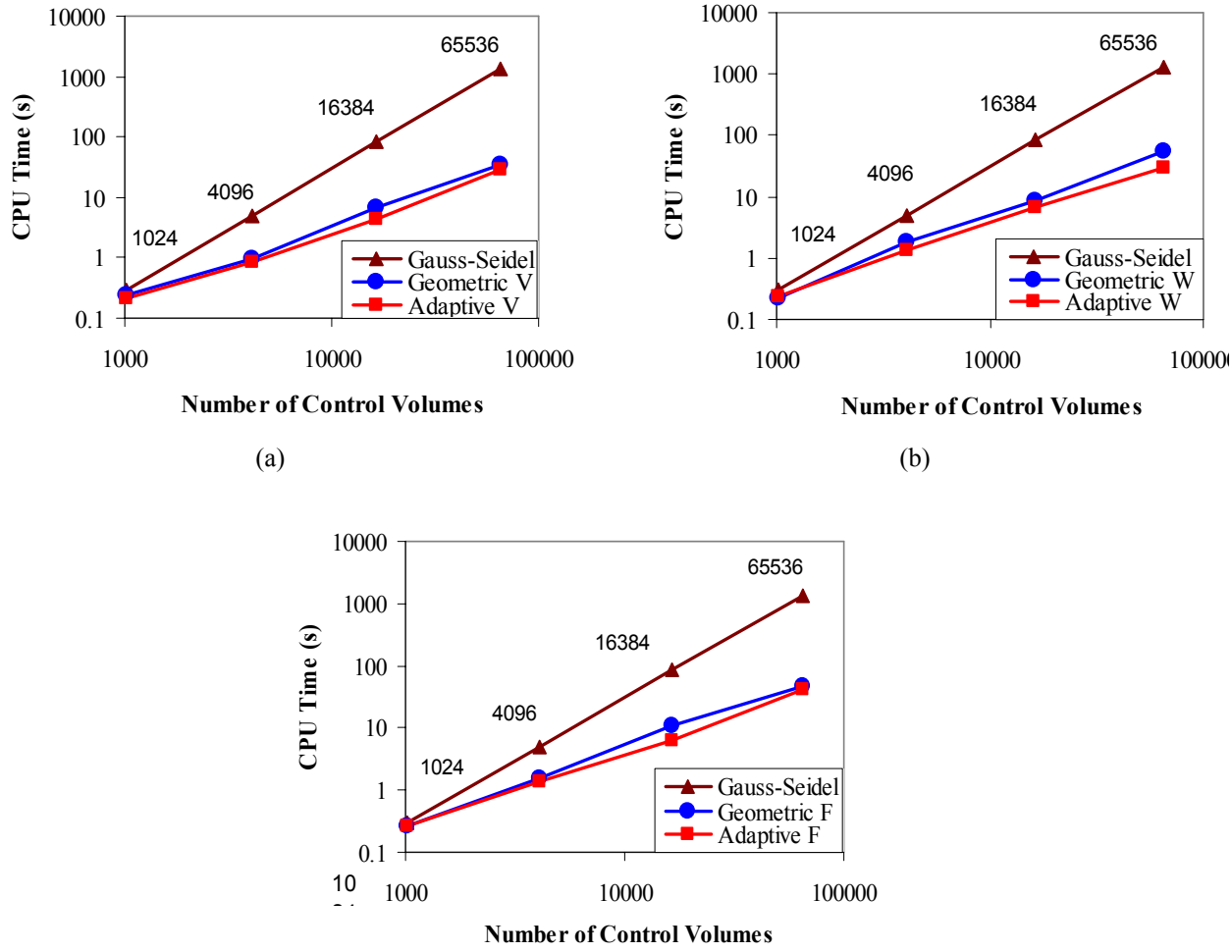
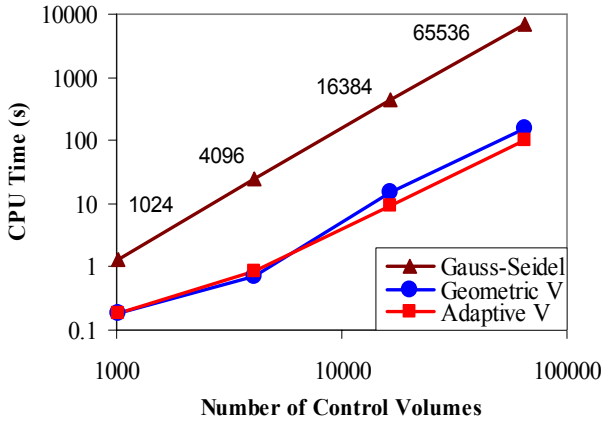


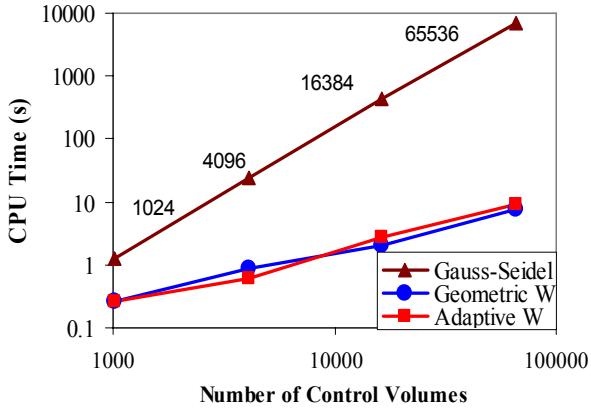

Figure 10. Comparison between agglomeration schemes used in ACM method to the first conduction problem in (a) V cycle, (b) W cycle and (c) F, in relation to the traditional Gauss-Seidel solver.

We can see that ACM with adaptive agglomeration scheme reach the converged solution more quickly than the ACM with geometric scheme and for this it is considered better. However the two agglomeration schemes had gotten good results (considering the CPU time), especially in comparison with Gauss-Seidel solver that it is a slow solver.

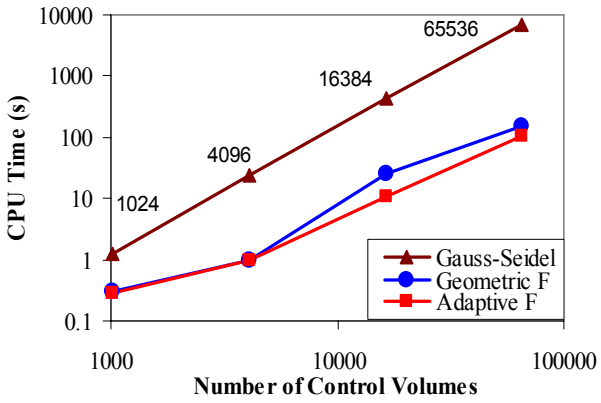
In Fig. (11) will be shown the results obtained to example 2, Fig. (5), comparing again ACM with adaptive agglomeration, ACM with geometric agglomeration and Gauss-Seidel solver.




(a)




(b)




(c)

Figure 11. Comparison between agglomeration schemes used in ACM method to the second conduction problem in (a) V cycle, (b) W cycle and (c) F cycle, in relation to the traditional Gauss-Seidel solver.

In Fig. (11.b), where is shown the performance of the two agglomeration schemes in W cycle, it is not defined which scheme is the best. In Fig. (11.a) and (11.c) the adaptive agglomeration scheme had better performance that the geometric scheme in the more large grids.

## 7. Conclusions

This paper has outlined some of the ACM main features. It was shown that the choice of which iterative method will be used in the grid levels influences the performance of the ACM method. It was seen that the solution obtained with the multigrid is accelerated even more when the ILU factorization is used as a base solver.

Two agglomeration types were studied: adaptive and geometric. The adaptive scheme considers the nature of the coefficients, joining cells with small timescale of the transport process (large coefficients). The geometric scheme joins cells trying to obtain only a regular coarse blocks with good aspect ratio.

In the cases analyzed here, the two agglomeration schemes had similar performance. However the schemes must be exhaustingly tested in other kinds of anisotropic problems to obtain a more conclusive idea regarding their behavior.

Finally, we can say that this paper studied some important features of ACM method that must to be chosen when one decides to implement and use this kind of method. However more analyses in unstructured meshes must be done, mainly foreseeing its application to a coupled linear equation system.

## 8. References


Brandt, A., 1977, "Multi-Level Adaptive Solutions to Boundary-Value Problems", Mathematics of Computation, vol 31, pp. 333-390.

Briggs, W. L., Henson, V. E., McCormick, S. F., 2000, "A Multigrid Tutorial", Society for Industrial and Applied Mathematics", EUA, 193p.

Elias, S. R., 1993, "Enhancements to Additive Correction Multigrid", Master thesis, Waterloo, Ontario, Canada.

Elias, S.R., Stubley, G.D., Raithby, G.D., 1997, "An Adaptative Agglomeration Scheme for Additive Correction Multigrid", International Journal of Numerical Methods in Engineering, Vol. 4 (5), pp. 887-903.

Francescatto, J., Dervieux, A., A Semi-Coarsening Strategy for Unstructured Multigrid based on Agglomeration, International Journal for Numerical Methods in Fluids, Vol. 26, pp. 927-957, 1998.

Hutchinson, B. R., Raithby, G. D., 1986, "A Multigrid Method Based on the Additive Correction Strategy", Numerical Heat Transfer, vol. 9, pp. 511-537.

Keller, S. C., Cordazzo, J., Hinckel, P. H., Maliska, C. R., 2004, "The Additive Correction Multigrid Method for Unstructured Grids", CILAMCE – XXV Iberian Latin-American Congress on Computational Methods in Engineering, 10-12 de Novembro, Recife, Pernambuco, Brazil.

Keller, S. C., Cordazzo, J., Hinckel, P. H., Maliska, C. R., 2004, "Additive Correction Multigrid Method Applied to Diffusion Problems with Unstructured Grids", ENCIT – 10th Brazilian Congress of Thermal Engineering and Sciences, 29 de Novembro - 3 de Dezembro, Rio de Janeiro, Brazil.

Maliska, C. R., 2004, "Transferência de Calor e Mecânica dos Fluidos Computacional", Ed. LTC, Rio de Janeiro, Brazil, 453 p.

Mesquita M. S., Lemos, M. J. S., 2005, "Effect of Medium Properties on Convergence Rates of Multigrid Solutions of Laminar Flows in Permeable Structures", International Journal of Dynamics of Fluids Vol. 1, No.1, pp. 69-86.

Raw, M. J., 1985, "A New Control-Volume-Based Finite Element Procedure for the Numerical Solution of the Fluid flow and Scalar Transport Equations", Doctor thesis, Waterloo, Ontario, Canada.

Rabi, J. A., Lemos, M. J. S., 2003, "Multigrid correction-storage formulation applied to the numerical solution of incompressible laminar recirculating flows". Applied Mathematical Modelling 27, pp. 717-732.

Stüben, A., 1999, "Algebraic Multigrid (AMG)", An Introduction with Applications, GMD-Report 70.

Stüben, A., 2001, "A Review of Algebraic Multigrid", Journal of Computational and Applied Mathematics, Vol. 128, pp. 281 –309.

Trottenberg, U., Oosterlee, C., Schüller, A., 2001. "Multigrid", Academic Press, 631p.

Vanka, S. P., 1986, "Block-Implicit Multigrid Calculation of Two-Dimensional Recirculating Flows", Computer Methods in Applied Mechanics and Engineering, 59, pp. 29- 48.

Venkatakrishnan, V., Mavriplis, D. J., 1996 "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids", Journal of Computational Physics, Vol. 127, pp. 380-397.

Volker, S., Burton, T., Vanke, S. P., 1996, "Finite Volume Multigrid Calculation of Natural-Convection Flows on Unstructured Grids", Numerical Heat Transfer, Part B, 30:1-22.

Wesseling, P, Oosterlee, C. W., 2001, "Geometric multigrid with applications to computational fluid dynamics", Journal of Computational and Applied Mathematics, 128, pp. 311-334.