AGGLOMERATION MULTIGRID SOLUTIONS OF LAMINAR VISCOUS FLOWS

Daniel Strauss – daniels@aer.ita.cta.br
Centro Técnico Aeroespacial, Instituto Tecnológico de Aeronáutica
CTA/ITA/IEAA, 12228-900 - São José dos Campos - SP - Brasil
João Luiz F. Azevedo – azevedo@iae.cta.br
Centro Técnico Aeroespacial, Instituto de Aeronáutica e Espaço
CTA/IAE/ASE-N, 12228-904 - São José dos Campos - SP - Brasil

Abstract. *The paper describes the implementation details and validation results for an agglomeration multigrid procedure developed in the context of hybrid, unstructured grid solutions of aerodynamic flows. The procedure is applied to the solution of 2-D, laminar viscous flows of aerospace interest. The governing equations are discretized with an unstructured grid finite volume method which is capable of handling hybrid unstructured grids. Time march uses an explicit, 5-stage, Runge-Kutta time-stepping scheme. Convergence acceleration to steady state is achieved through the implementation of an agglomeration multigrid procedure which retains all the flexibility previously available in the unstructured grid code. The calculation capability created is applied to 2-D laminar viscous flows of aerospace interest. Studies of the various parameters affecting the multigrid acceleration performance are undertaken with the objective of determining optimal numerical parameter combinations.*

Keywords: *Agglomeration multigrid, Convergence acceleration, Unstructured grids, Finite volume methods.*

1. INTRODUCTION

Many different convergence acceleration methods have been developed and discussed in the literature in the past years. These methods involve various aspects of the numerical solution of partial differential equations and demand implementation efforts which can be very simple or very complex. The implementation of a variable time step option and optimized time stepping schemes yields a solution to the convergence acceleration problem which demands a relatively small effort. On the other hand, time-implicit schemes or multigrid procedures also solve the problem but are much more demanding. A combination of these convergence acceleration methods usually gives the best overall result.

Unstructured meshes present additional difficulties to the implementation of some of the convergence acceleration methods when compared to structured grids. For instance, implicit time-stepping schemes are handicapped by the need of expliciting the connectivity of the mesh, resulting in a difficult implementation and a reduced gain in efficiency. Multigrid procedures are also more difficult in unstructured meshes, mainly due to the burden of obtaining the coarse meshes. However, the gains with a multigrid procedure in unstructured meshes are very significant and, therefore, the additional work is worthwhile.

The main concern in implementing a multigrid procedure in unstructured meshes is the generation of the coarse meshes. There are basically three different ways of treating this problem. In the first approach, the fine meshes are obtained refining the coarse meshes, resulting in nested meshes (Mavriplis, 1988). This approach often results in problems in obtaining an adequate mesh refinement in the regions of interest. Another approach considers coarse meshes that are generated independently from the fine meshes, such that the meshes are unrelated or non-nested (Mavriplis, 1988, 1990, Mavriplis and Jameson, 1990). This approach has the inconvenient of requiring the calculation of the intersections between the coarse and fine mesh volumes, which can be very difficult in the general case. Moreover, the generation of the coarse meshes itself also requires a substantial amount of work. Finally, the third approach is the agglomeration technique (Mavriplis and Venkatakrishnan, 1994, Venkatakrishnan and Mavriplis 1995). In this case, the coarse meshes are generated by agglomerating the neighboring volumes of the fine meshes. Hence, this approach does not have the disadvantages of the others and it was the one chosen by the authors in this work.

This work, then, describes the implementation details and validation results for an agglomeration multigrid procedure. The procedure is developed in the context of a 2-D, laminar Navier-Stokes solver, capable of handling hybrid, unstructured grids. The governing equations are discretized with a centered finite volume method and time march uses an explicit, 5-stage, Runge-Kutta time-stepping scheme. Laminar viscous flow simulations over a flat plate are used to validate the present implementation. Studies of the various parameters affecting the multigrid acceleration performance are undertaken with the objective of determining optimal numerical parameter combinations.

2. MULTIGRID IMPLEMENTATION

2.1 Multigrid Procedure

Multigrid methods have been developed for the solution of general partial differential equations on a discretized domain. Therefore, in this section a general discussion of the implementation of a multigrid procedure will be presented.

Considering a problem written in the operator form as

$$L\left(u\right)=0 \quad , \tag{1}$$

the discretized problem that has to be solved is

$$L_h\left(u_h\right)= f_h \quad , \tag{2}$$

where the discretization in which a solution is wanted is represented by *h*. With the use of the multigrid procedure, the discretized problem is solved on a coarser mesh *H*. As the *H* mesh has fewer points than the *h* mesh, the solution of the problem in the *H* mesh requires a lower computational cost than in the *h* mesh. Hence, the problem in the *H* mesh can be written as

$$L_H\left(u_H\right)= f_H \quad , \tag{3}$$

with

$$f_H = L_H\left(I_h^H u_h\right) - I_h^H r_h \quad , \tag{4}$$

$$r_h = L_h\left(u_h\right) - f_h \quad , \tag{5}$$

and f_h = 0 in the finest mesh. The I_h^H operator is the restriction operator, which transfers the properties from the fine to the coarse mesh. The solution in the *h* mesh is updated from the solution of Eq. (3) using the prolongation operator I_H^h, which transfers the properties from the coarse to the fine mesh. Hence,

$$u_h^{new} = u_h + I_H^h \left(u_H - I_h^H u_h \right) \; . \tag{6}$$

It should be noted that the multigrid procedure can be applied successively using coarser and coarser meshes. Therefore, Eq. (3) can also be solved using a multigrid procedure, but using a coarser mesh 2*H*.

For correction operators written in delta form, the problem to be solved is

$$L\left(u\right) = N\left(u^{n+1} - u^n\right) + R\left(u^n\right) \; . \tag{7}$$

In the last equation, the *N* operator is associated with the properties update, while the *R* operator is associated with the discretization of the problem equations. In this case, the correction operator can represent a relaxation method or a time marching scheme. Rewriting the operator in Eq. (7) in the discretized form of Eq. (3), the problem that should be solved in the coarse mesh is obtained as

$$L_H\left(u_H\right) = N_H\left(u_H^{n+1} - u_H^n\right) + R_H\left(u_H^n\right) = f_H^n \; . \tag{8}$$

Equations (4) and (5), then, become

$$r_h^n = N_h\left(u_h^{n+1} - u_h^n\right) + R_h\left(u_h^n\right) - f_h^n \; , \tag{9}$$

$$f_H^n = N_H\left(I_h^H u_h^{n+1} - I_h^H u_h^n\right) + R_H\left(I_h^H u_h^n\right) - I_h^H\left[N_h\left(u_h^{n+1} - u_h^n\right) + R_h\left(u_h^n\right) - f_h^n\right] \; . \tag{10}$$

If one defines

$$F_H^n = R_H\left(I_h^H u_h^n\right) - I_h^H\left[R_h\left(u_h^n\right) - f_h^n\right] \; , \tag{11}$$

Equation (10) transforms to

$$f_H^n = F_H^n + N_H\left(I_h^H u_h^{n+1} - I_h^H u_h^n\right) - I_h^H\left[N_h\left(u_h^{n+1} - u_h^n\right)\right] \; . \tag{12}$$

In order to obtain the solution of Eq. (8), it is necessary to calculate all the terms in Eq. (12). However, the two last terms of Eq. (12) are not known because they involve the solution on the fine mesh on the *n* + 1-th time step. Hence, these terms are calculated with a time delay, that is, in the *n*-th time step. Using the definition of the problem, Eq. (8), and this concept of time delay, one can obtain the following results

$$N_h\left(u_h^n - u_h^{n-1}\right) = f_h^{n-1} - R_h\left(u_h^{n-1}\right) \; , \tag{13}$$

$$N_H\left(I_h^H u_h^n - I_h^H u_h^{n-1}\right) = f_H^{n-1} - R_H\left(I_h^H u_h^{n-1}\right) \; . \tag{14}$$

Substituting Eqs. (13) and (14) in Eq. (12)

$$f_H^n = F_H^n + f_H^{n-1} - R_H\left(I_h^H u_h^{n-1}\right) - I_h^H\left[f_h^{n-1} - R_h\left(u_h^{n-1}\right)\right] \; , \tag{15}$$

or, using Eq. (11),

$$f_H^n = F_H^n + f_H^{n-1} - F_H^{n-1} \; . \tag{16}$$

Equation (16) can be used to obtain the value of f_H^{n-1}, resulting

$$f_H^n = F_H^n + F_H^{n-1} + f_H^{n-2} - F_H^{n-2} - F_H^{n-1} \quad . \tag{17}$$

In this equation, the terms F_H^{n-1} cancel each other, so there is no contribution of the n – 1-th instant of time. After successive application of Eq. (17), the only contribution left will be that of the initial instant of time, which can be neglected. Therefore, the final problem that has to be solved is

$$N_H\left(u_H^{n+1} - u_H^n\right) + R_H\left(u_H^n\right) = F_H^n \quad , \tag{18}$$

with

$$F_H^n = R_H\left(I_h^H u_h^n\right) - I_h^H\left[R_h\left(u_h^n\right) - f_h^n\right] . \tag{19}$$

2.2 Agglomeration Technique

The previous section discussed the multigrid implementation in a general way. There was no attempt to treat the practical issues associated with such implementation. In this section, the actual implementation of the agglomeration procedure for generating the coarse meshes will be addressed.

The coarse meshes for the multigrid procedure are generated by agglomerating or grouping fine mesh volumes to form one coarse mesh volume. A “seed” volume is chosen in the fine mesh and, then, all the volumes that have a node or an edge in common with this “seed” volume are grouped and they form the coarse mesh volume. Another “seed” volume is selected and the agglomeration procedure continues grouping all the fine mesh volumes. It should be noted that during the agglomeration procedure only the volumes that have not been already agglomerated may be grouped to form a coarse mesh volume. This is a necessary condition in order to guarantee that there is no volume overlapping in the coarse mesh.

Better coarse mesh quality can be obtained if the selection of the “seed” volumes is not random. Therefore, a queue containing all the fine mesh volumes is generated prior to the agglomeration procedure. In this work, the queue is formed such that the first volumes are the volumes next to a boundary and, after them come the interior volumes. This approach is very simple, easy to implement and adds a very low additional computational cost. Although it does not necessarily provide the best agglomeration of the interior volumes, it results in good quality coarse mesh volumes close to the boundaries.

As the spatial discretization scheme used in this work is linear, a simplification can be made in the coarse meshes. This simplification consists in eliminating the nodes that belong to only two volumes. The justification for this procedure comes from the fact that the flux passing between the two volumes is the same whether the boundary separating the two volumes is discretized by one or many edges, provided that the discretization scheme is linear. Therefore, a significant amount of storage space can be saved by doing this mesh simplification. Figure 1 presents an example of such node elimination, where the darker lines represent the original boundary separating two coarse volumes and the dashed line represents the boundary edge after the node elimination.

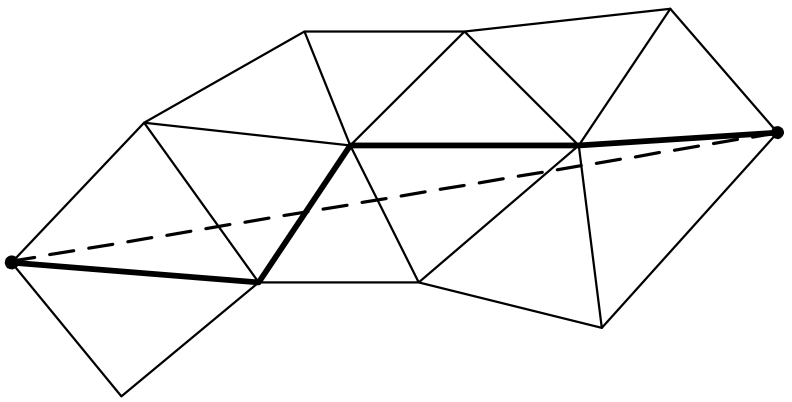


Figure 1: Mesh simplification by the node elimination procedure.

Although the mesh simplification previously described can reduce the total storage space required by the code, it brings a complication related to the connectivity of the nodes in the mesh. As some nodes in the mesh are not used, it is necessary to ensure that the remaining nodes are properly counterclockwise oriented in order to have the normal vectors of each edge correctly pointing outwards. This is accomplished using the node orientation in the fine mesh volumes to orient the nodes in the coarse mesh volumes.

The agglomeration procedure can be summarized, then, in three steps. The first step consists in defining the queue of volumes of the fine mesh. In the second step, the fine mesh volumes are agglomerated to form the coarse mesh volumes, following the queue generated in the first step. During this step, the mesh simplification described is adopted, and only the nodes that belong to three or more coarse mesh volumes are stored. The third and final step is the verification of the node orientation in each volume of the coarse mesh, and the correction of the orientation where it is needed.

The actual implementation of this agglomeration procedure was designed to require the minimum amount of storage possible. Therefore, the only extra information that has to be stored in each mesh, besides the usual information associated with the solution procedure, is the number of the coarse mesh volume that contains each of the fine mesh volumes.

2.3 Restriction and Prolongation Operators

In the description of the multigrid procedure, the restriction and prolongation operators were introduced. Their actual mathematical definition was not presented in the last sections, but it will be discussed in this section. The restriction operator for the conserved properties, as well as for the residuals, will be presented, together with two different prolongation operators for the conserved properties.

The restriction operator transfers a variable from a fine mesh to a coarse mesh. The operator used in the present work for the conserved properties restriction is the volume weighted average. Therefore, the restricted conserved properties of a coarse mesh volume are equal to the sum of the conserved properties of all the fine mesh cells that form this coarse mesh volume, weighted by their volumes. On the other hand, the restriction of the residuals is accomplished by simple addition of the fine mesh residuals. Hence, the residual of a coarse mesh volume is equal to the sum of the residuals of all the fine mesh volumes that are contained by this coarse mesh volume. The restriction operator for the residuals is different from the restriction operator for the conserved properties because the residuals can be interpreted as line integrals in finite volume schemes. Consequently, as the residuals of the fine mesh are summed, the interior edges contribution will cancel each other, leaving only the contribution of the edges that form the coarse mesh volume.

The prolongation operator, in opposition to the restriction operator, transfers a variable from a coarse mesh to a fine mesh. As discussed in the Multigrid Procedure section, only the conserved properties corrections have to be prolonged. Hence, only one prolongation operator has to be defined. In this work two such different operators were used. The first operator uses direct injection of the coarse mesh values into the fine mesh. With this operator, the correction of a fine mesh volume is equal to the correction of the coarse mesh volume that contains this volume. Although this operator is very simple and easy to implement, it is not able to transfer exactly even a linear distribution. The second prolongation operator uses an averaging process to obtain the corrections in the fine grids. The averaging consists in, for each edge of the fine mesh, arithmetically averaging the corrections of the coarse mesh volumes corresponding to the two volumes that contain the edge. For each volume, then, these averaged corrections are summed and the result is divided by the number of edges of the volume. This operator is also very easy to implement, and it has the advantage of being able to transfer a linear distribution with less error than the first prolongation operator. As it will be shown latter in the work, the second option results in better convergence rates for viscous flows.

3. THEORETICAL FORMULATION

The agglomeration multigrid procedure previously described is applied to the solution of the laminar Navier-Stokes equations, which can be written in integral form as

$$\frac{\partial}{\partial t}\int_V Q\,dV + \int_S \left(E\,dy - F\,dx\right) = 0 \quad , \tag{20}$$

where

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{Bmatrix} , \quad E = \begin{Bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ (e+p)u - \tau_{xx}u - \tau_{xy}v + q_x \end{Bmatrix} , \quad F = \begin{Bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 - \tau_{yy} \\ (e+p)v - \tau_{xy}u - \tau_{yy}v + q_y \end{Bmatrix} . \tag{21}$$

Using a cell centered based finite volume scheme, the discrete conserved variables vector is defined as an average over the cell of the continuous properties. Hence, for the i-th volume the discrete property vector is

$$Q_i = \frac{1}{V_i}\int_{V_i} Q\,dV \quad . \tag{22}$$

The definition of the discrete vector Q_i can be used to rewrite Eq. (20), resulting

$$\frac{\partial}{\partial t}\left(V_i Q_i\right) + \int_{S_i}\left(E\,dy - F\,dx\right) = 0 \quad . \tag{23}$$

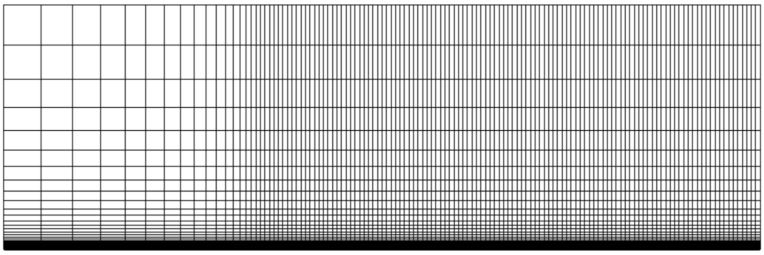
The surface integral in Eq. (23) is discretized using a centered scheme, such that

$$\int_{S_i}\left(E\,dy - F\,dx\right) = \sum_{k=1}^{n}\left[E\left(Q_{ik}\right)\Delta y_{ik} - F\left(Q_{ik}\right)\Delta x_{ik}\right] \quad . \tag{24}$$

As a centered scheme is being used to discretize the equations, artificial dissipation terms must be added in order to control nonlinear instabilities (Jameson and Mavriplis, 1986). The artificial dissipation operator chosen in the present work is formed by the sum of an undivided Laplacian and a biharmonic operator (Mavriplis 1988, 1990). The expressions for this operator can be found, for instance, in Azevedo and Korzenowski (1998). The time integration of Eq. (23) is accomplished by a fully explicit, 2nd-order accurate, 5-stage Runge-Kutta time-stepping scheme (Mavriplis, 1990)

4. RESULTS AND DISCUSSION

The multigrid procedure described in the previous sections is applied in the solution of laminar viscous flow over a flat plate configuration at zero angle of attack. This quite simple problem was chosen in order to provide a model problem which could be used to assess the convergence acceleration performance of the multigrid procedure. Different parameters of the multigrid procedure were tested in an attempt to determine the settings that result in the best convergence acceleration.

(a)  9417 nodes, 9216 volumes

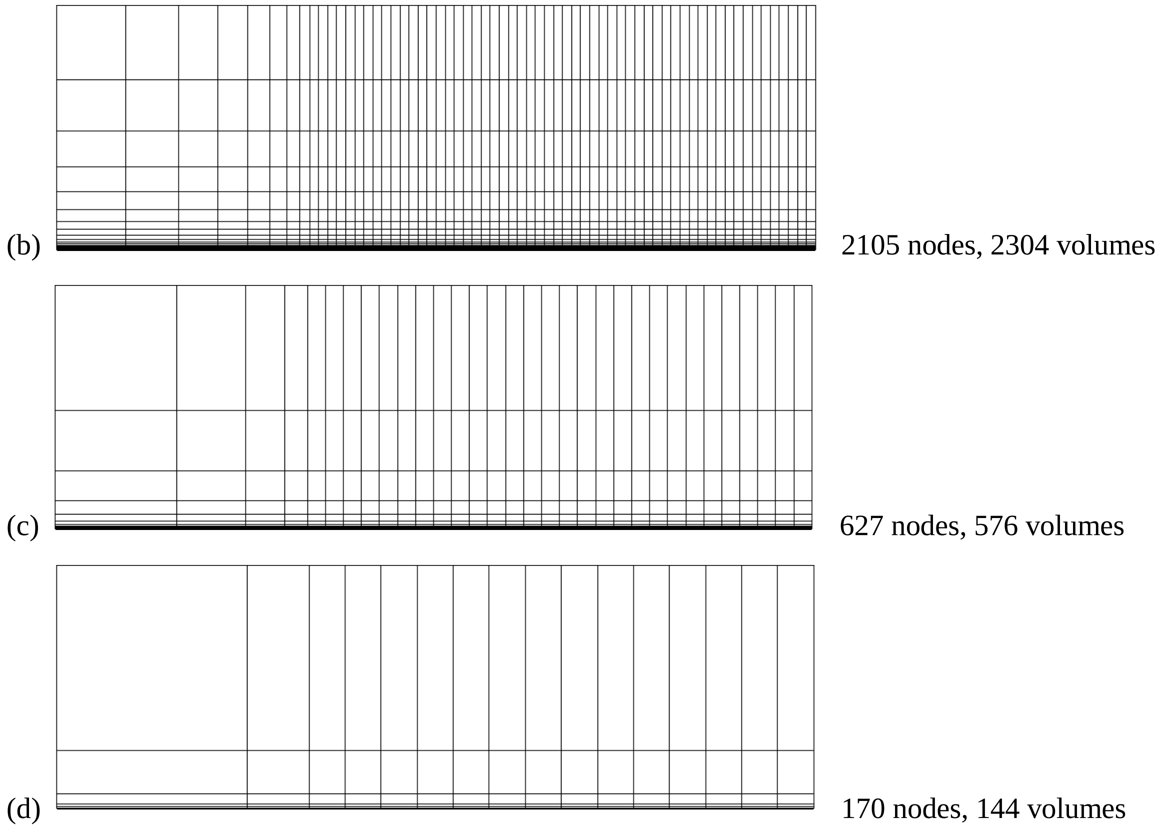


Figure 2: Fine (a) and coarse agglomerated meshes (b, c, d) used for the flat plate calculation.

Figure 2 presents the meshes used for the flat plate calculation. Although these meshes look like structured meshes, the authors emphasize that they were treated in a fully unstructured way. Moreover, only the mesh (a) in Figure 2 was provided as entrance data. The coarse meshes (b), (c) and (d) were generated by the code using the agglomeration procedure previously described. The decision of using an unstructured mesh generated from a structured mesh was made because a structured mesh is very easy to generate in this case, and the use of quadrilateral cells is more adequate to the treatment of viscous flows adjacent to solid walls. On the other hand, since the complete procedure treats all meshes as fully unstructured, the capability could be directly applied to any other configuration.

The flow characteristics used in the simulation were $M_\infty = 0.3$ and $Re = 1.0 \times 10^5$. Many different numerical parameters were tested in order to assess the optimal settings that result in the best convergence ratios. A summary of the cases tested is presented in Table 1, and the convergence histories corresponding to these cases are presented in Figure 3. In this figure, the convergence history for the case without the use of the multigrid procedure is also shown, and it is labeled as single grid case.

Table 1: Description of the cases tested.

Case	Description
1	3 coarse mesh levels, 1 pre and 1 pos-sweep iterations, W cycle, 20 iterations on the coarsest grid level, averaging prolongation operator
2	Same as case1 but with 4 coarse mesh levels
3	Same as case 2 but with zero pre-sweep iterations
4	Same as case 2 but with direct injection prolongation operator
5	Same as case 2 but with 1 pre and 1 pos-sweep iterations on each coarse grid level
6	Same as case 5 but with 2 iterations on the coarsest grid level
7	Same as case 5 but with 40 iterations on the coarsest grid level

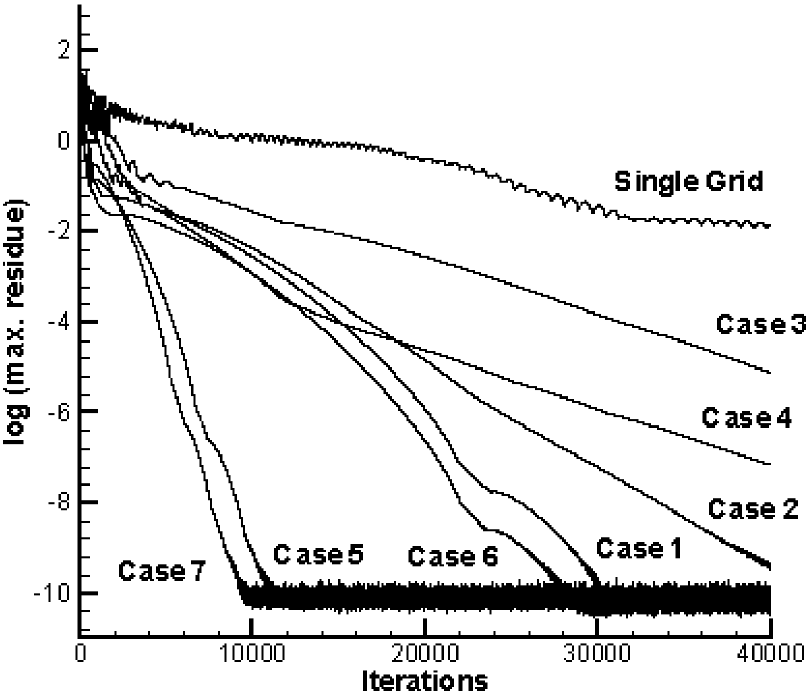


Figure 3: Convergence histories for the cases described in Table 1.

The convergence histories presented in Figure 3 cannot be used to compare directly the performance of the multigrid procedure, as the computational cost per iteration is different in each case tested. Therefore, a comparison in terms of computational time is presented in Table 2. From this table it can be seen that it is worthwhile to expend more in the solution of the coarse meshes in order to obtain the best overall computational time reduction. Although there was no attempt to converge the case where the multigrid procedure was not used to machine zero, a comparison between this case and the best case with multigrid can be made in terms of computational cost to reduce the logarithm of the residual of a few orders of magnitude. Hence, the cost to converge the solution without multigrid by four orders of magnitude is 27000 *s*, while the cost with multigrid is 11300 *s*, which means a reduction to 42% of the original computational time.

Table 2: Computational time required to reduce the residual to machine zero. The computational time for the cases marked with * correspond to 40000 iterations.

Case	Computational Time per Iteration (*s*)	Computational Time (*s*)
Single Grid	0.486	19440 *
1	2.08	62400
2	1.66	66400 *
3	1.20	48000 *
4	1.67	66800 *
5	2.77	33240
6	2.27	63560
7	3.14	31400

The numerical solution obtained is compared with the exact Blasius solution for the incompressible flat plate in Figure 4. In this figure, the two numerical solutions correspond to the case where the multigrid procedure was used and to the case where it was not. The velocity profiles are plotted as a function of the nondimensional coordinate $\eta = \frac{y}{x}\sqrt{Re_x}$, and correspond to a station in the middle of the plate. The small difference between the two curves is due to the worst convergence of the case where the multigrid procedure was not used. In the case where the multigrid procedure was used, the solution presented was converged to machine zero. One can see from the figure that the numerical solution is very close to the exact Blasius solution.

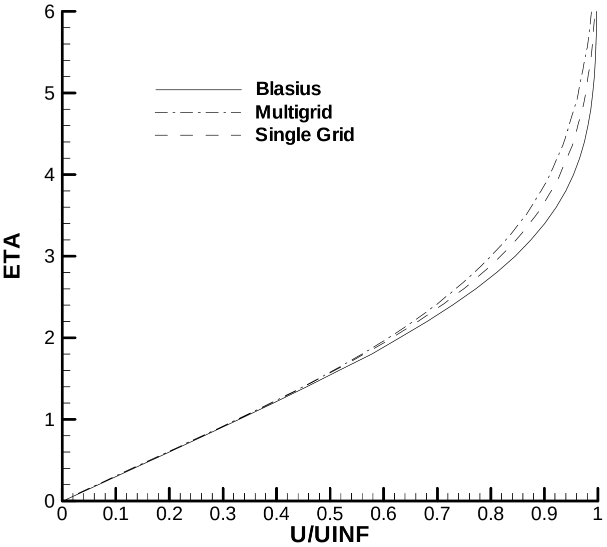
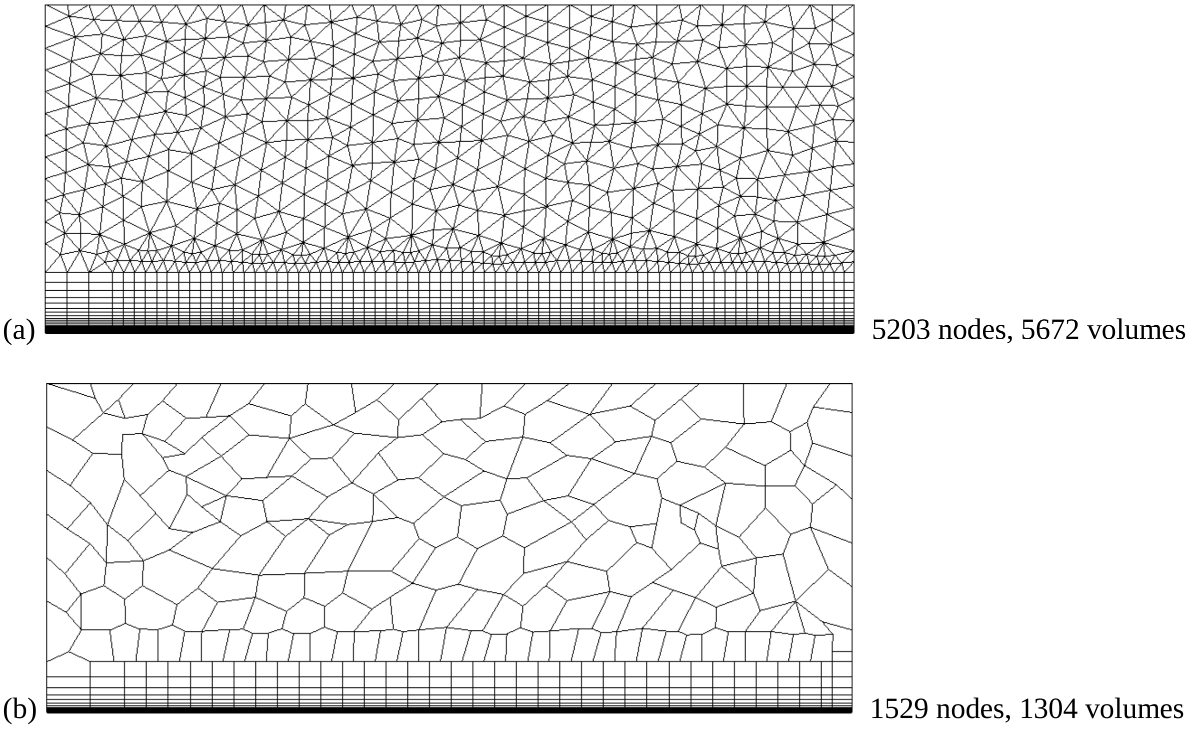


Figure 4: Comparison of the numerical results obtained with the Blasius solution for the flat plate ($M_\infty = 0.3$ and $Re = 1.0 \times 10^5$).

Although all the results presented were obtained using the meshes in Figure 2, the agglomeration procedure developed is able to handle more difficult mesh topologies. For instance, Figure 5 presents an example of a hybrid mesh and two agglomerated meshes that were obtained from this mesh. This topology represents a typical case for the treatment of viscous boundary layers in the proximity of solid walls. It has the advantage of being capable of adequately capture the viscous effects close to the body while reducing the number of points in the regions far from the body, where they are not needed.



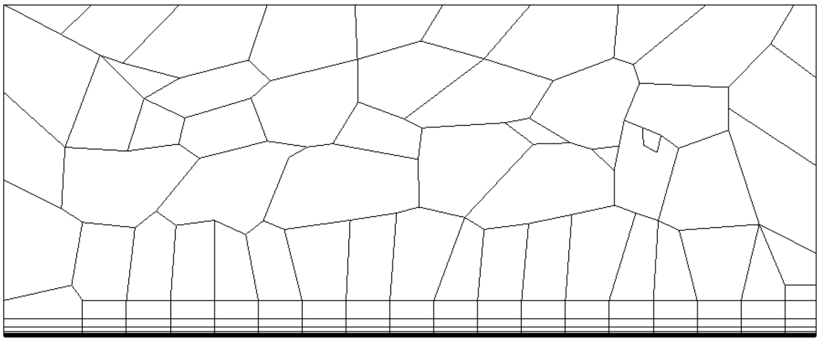
(c)  393 nodes, 324 volumes

Figure 5: Hybrid meshes for a flat plate. Mesh (a) is the original mesh, while meshes (b) and (c) are coarse agglomerated meshes.

5. CONCLUSIONS

An agglomeration multigrid capability was developed in the context of a 2-D, laminar viscous solver which was capable of handling hybrid unstructured meshes. The multigrid procedure as well as the agglomeration technique were discussed in detail. Moreover, the restriction and prolongation operators used were presented, together with some considerations about simplifications that can be made to reduce the computational cost of the multigrid procedure.

Laminar viscous flow over a 2-D flat plate was used to validate the implemented capability. Studies of the various parameters affecting the multigrid acceleration performance were undertaken with the objective of determining optimal numerical parameter combinations. The results obtained show a significant improvement in convergence in comparison with the case without the use of the multigrid procedure. An example of the agglomerated meshes of a typical hybrid grid is presented, and this is an indication of the generality of the agglomeration procedure implemented.

Acknowledgments

The authors gratefully acknowledge the partial support of Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under the Integrated Project Research Grant No. 522413/96-0.

REFERENCES

Azevedo, J.L.F. and Korzenowski, H., 1998, Comparison of unstructured grid finite volume methods for cold gas hypersonic flow simulations, AIAA Paper No. 98-2629, Proceedings of the 16th AIAA Applied Aerodynamics Conference, June, Albuquerque, New Mexico, pp. 447-463.

Jameson, A. and Mavriplis, D.J., 1986, Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh, AIAA Journal, vol. 24, n. 4, pp. 611-618.

Mavriplis, D.J., 1988, Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes, AIAA Journal, vol. 26, n. 7, pp. 824-831.

Mavriplis, D.J., 1990, Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes, AIAA Journal, vol. 28, n. 2, pp. 213-221.

Mavriplis, D.J. and Jameson, A., 1990, Multigrid solution of the Navier-Stokes equations on triangular meshes, AIAA Journal, vol. 28, n. 8, pp. 1415-1425.

Mavriplis, D.J. and Venkatakrishnan, V., 1994, Agglomeration multigrid for viscous turbulent flows, AIAA Paper No. 94-2332, Proceedings of the 25th AIAA Fluid Dynamics Conference, June, Colorado Springs.

Venkatakrishnan, V. and Mavriplis, D.J. 1995, Agglomeration multigrid for the three-dimensional Euler equations, AIAA Journal, vol. 33, n.4, pp.633-640.