Seventh International Conference on
Computational Fluid Dynamics (ICCFD7),
Big Island, Hawaii, July 9-13, 2012

ICCFD7-1603

Development of an agglomeration multigrid technique in the hybrid solver elsA-H

C. Marmignon [*], B. Cantaloube[*], M.C. Le Pape[*],
M. de la Llave Plata[*], V. Couaillier[*], and M. Gazaix[*]
Corresponding author: claude.marmignon@onera.fr

[*] ONERA - The French Aerospace Lab, FR-92322 Châtillon, France.

Abstract: This paper presents an agglomeration multigrid procedure developed in the context of hybrid grids. A key issue for using multigrid acceleration techniques on unstructured meshes is the generation of the coarse grid levels. Here, the grid coarsening technique relies on the agglomeration of cells based on their distribution on an octree data structure. The used hybrid solver is called *elsA-H* and its aim is to leverage the full potential of structured solvers and unstructured mesh generation by enabling any type of grid to be used within the same simulation process. The implementation uses a cell-centered, face-based data structure. The aim of this work is therefore to accelerate *elsA-H* solver by implementating the multigrid concept.

Keywords: Agglomeration multigrid, convergence acceleration, finite volume method, unstructured grids.

1 Introduction

Over the past few years, ONERA has been working on the extension of the multiblock structured solver *elsA* [1] to hybrid grid configurations [2, 3, 4]. The new hybrid solver is called *elsA-H*, in which a new unstructured solver has been built by taking advantage of the modular approach provided by the object-oriented framework. The aim of *elsA-H* is therefore to leverage the full potential of structured solvers and unstructured mesh generation by enabling any type of grid to be used within the same simulation process. The main challenge lies in the numerical treatment of the hybrid-grid interfaces where two different blocks meet. This paper reports recent progress on the development of a multigrid (MG) algorithm for unstructured grid simulations using *elsA-H*. This constitutes the first step towards the development of a more general hybrid MG technique, within the framework of the *elsA-H* project.

Multigrid is considered as one of the most effective acceleration techniques [5, 6, 7]. MG methods are based on the discretization of the equations onto a number of grids at different levels of resolution. The targeted solution to the equations is assumed to be well represented on the finest grid, whereas the rest of the grids provide a coarser approximation of the same solution, and are used to accelerate the overall convergence of the simulation process.

In the context of unstructured methods, a major difficulty of the MG technique lies in the generation of the coarse levels. There are different ways of handling this problem [8, 9, 10]. The strategy adopted here is based on unstructured volume-agglomeration to generate the coarse levels from the fine grid [11]. Grid cells are fused together to form a smaller set of larger polyhedral control volumes. The main difficulty of this technique is the selection of the cells to be agglomerated so that the new cells have acceptable aspect ratios. This constraint is easier to respect for a MG method implemented in the context of node-centered schemes. In this study, the methodology used in the generation of the coarse grids is that developed by Mahmutyazicioglu [12]. The method relies on the agglomeration of cells based on their distribution on an octree data structure. This is a hierarchical data structure which is based on the successive subdivision of the space into eight equal-size octants.

The developed multigrid algorithm is similar to the one used in *elsA*'s structured method [1], in which the grid transfer operators have been adapted to the unstructured framework.

This work describes the implementation and first validation results for this agglomeration multigrid procedure. The governing equations are discretized using a centered Jameson-Smith-Turkel (JST) scheme [13, 14] or the Roe's TVD upwind scheme with MUSCL reconstruction [15]. Time discretization uses a multi-stage Runge-Kutta scheme or an implicit backward Euler scheme. The implicit scheme implemented is the LU-SSOR (Lower-Upper Symmetric Successive Over Relaxation) technique. Bidimensional flows using quadrilateral meshes are considered in order to compare structured and unstructured MG algorithms. The unstructured MG algorithm and the agglomeration technique are tested on the Euler flows around the bidimensional NACA 0012 airfoil and the Onera M6 wing.

In order to be consistent with *elsA-H*'s face-based format, the connectivity information that defines the different grids is stored into the CGNS input file following this same data structure. The generation of the coarse grids is therefore considered as a pre-processing step in the simulation process.

2 Hybrid Flow Solver

The adaptation of elsA to deal with hybrid grids is based on an Object-Oriented framework in which new abstract (generic) parent classes have been introduced, from which purely structured and unstructured objects and methods inherit. A large number of Fortran routines have been reused (with minor or no changes at all) in the development of new unstructured algorithms. As an example, regarding turbulence modelling most of the structured classes can be used, as the numerical scheme is formally identical to the structured method. Dedicated unstructured classes have been introduced for the computation of the gradient terms and the application of the boundary conditions.

This modular approach allows us to develop a new unstructured solver within elsA, which means that there is no need for external code coupling.

Following elsA's own numerical choices, elsA-H is based on a multi-block cell-centered finite-volume approach. elsA-H's unstructured solver relies on a face-based strategy. Therefore, a pre-processing stage is necessary in order to transform the element-based connectivity information provided by the grid generator into a face-based format.

elsA-H supports multi-element grids. So far, it is able to work on hexahedra, tetrahedra, pyramids and prisms, which is sufficient for the definition of multi-block hybrid grids with conforming interfaces.

The exchange of data between blocks is done through one single layer of ghost cells, which requires adapted numerical schemes at the block boundaries.

The main input data format supported by elsA-H is the CFD General Notation System (CGNS) [16], which allows an easy representation of structured, unstructured and hybrid grids in one single file.

The discretization of the convective terms within elsA can be done by using either the centered Jameson scheme with pressure-velocity sensor, or Roe's scheme with MUSCL interpolation.

3 Multigrid Technique

3.1 Multigrid procedure

The multigrid method using a sequence of fine to coarse grids, denoted h-multigrid, has been extensively used for many years in block-structured finite volume codes in the CFD community. It is considered to be one of the best convergence acceleration techniques.

The time integration of the system of ordinary differential equations is carried out using a multi-stage Runge-Kutta scheme (the Backward Euler scheme corresponds to a one-step scheme) or an implicit backward Euler scheme . In general, classical iterative approaches are well adapted for rapidly damping high frequency error components on a given grid. The remaining errors, associated with the smoother low frequency error components, are responsible for the slow convergence. These low frequency error components on the fine grid appear as higher frequencies on the coarser grid. Thus, to ensure fast convergence of the solution to steady state, a multigrid acceleration technique is used. This technique uses a sequence of successively coarser grids to efficiently damp the perturbations. Let us denote the grid level by a subscript, a sequence of grids h_1,...,

h_m, ..., h_M are then defined, where h_1 denotes the finest grid and h_M represents the coarsest grid. The multigrid strategy chosen for the present work is the Full Approximation Storage (FAS) scheme in conjunction with the Runge-Kutta time stepping proposed by Jameson. This strategy is used to improve the convergence rate of a multi-block solver for the solution of the Euler and the Reynolds-Averaged Navier-Stokes equations. Jameson's FAS algorithm for a simple V-Cycle can then be summarised as follows :

- Compute the residual R_{h_1} and start with a q-stage Runge-Kutta time stepping to update the solution on the finest level.

The following steps are repeated right up to the coarsest level $m = 2, ..., M$ which corresponds to the Restriction step :

- Recompute the residual $R_{h_{m-1}}(u_{h_{m-1}})$ on the previous level and calculate the modified residual to be transferred from grid level h_{m-1} to the level h_m:

$$R^{(*)}_{h_{m-1}} = R_{h_{m-1}}(u_{h_{m-1}}) + P_{h_{m-1}},$$

where $P_{h_{m-1}}$ is the added forcing function defined below with P_{h_1} on the finest level.

- Transfer the solution and residual vectors from the previous grid h_{m-1} to the next coarser grid h_m using respectively the fine to coarse transfer operators $T^{h_m}_{h_{m-1}}$ and $\hat{T}^{h_m}_{h_{m-1}}$:

$$\begin{aligned} \overline{u}_{h_m} &= T^{h_m}_{h_{m-1}} \; u_{h_{m-1}} \\ \overline{R}_{h_m} &= \hat{T}^{h_m}_{h_{m-1}} \; R^{(*)}_{h_{m-1}} \end{aligned}$$

- Compute the forcing function for the residuals on the grid level h_m which is the difference between aggregated residuals transferred from grid h_{m-1} and the residuals recalculated on h_m :

$$P_{h_m} = \overline{R}_{h_m} - R_{h_m}(\overline{u}_{h_m}),$$

- where $R_{h_m}(\overline{u}_{h_m})$ is the residual vector computed on the grid level h_m using the transferred solution vector $\overline{u}_{h_m}$ from the previous grid level h_{m-1}.

- Start Runge-Kutta time stepping on the coarse level h_m using the following reformulated version, to take into account the forcing function as well as to include subiterations if necessary, coupled with an implicit smoothing technique (IRS or LU) :

$$\left\{ \begin{array}{l} u^{(0)}_{h_m} = \overline{u}_{h_m} \; ou \; u^{(q)}_{h_m} \\ \\ \Delta\tilde{u}^{(1)}_{h_m} = \alpha_1 \dfrac{\Delta t_{h_m}}{\Omega_{h_m}} \left[R_{h_m}(u^{(0)}_{h_m}) \;\; + P_{h_m} \right] \\ \Theta_{h_m} \; \Delta u^{(1)}_{h_m} = \Delta\tilde{u}^{(1)}_{h_m} \\ u^{(1)}_{h_m} = u^{(0)}_{h_m} + \Delta u^{(1)}_{h_m} \\ \vdots \\ \Delta\tilde{u}^{(q)}_{h_m} = \alpha_q \dfrac{\Delta t_{h_m}}{\Omega_{h_m}} \left[R_{h_m}(u^{(q-1)}_{h_m}) + P_{h_m} \right] \\ \Theta_{h_m} \; \Delta u^{(q)}_{h_m} = \Delta\tilde{u}^{(q)}_{h_m} \\ u^{(q)}_{h_m} = u^{(0)}_{h_m} + \Delta u^{(q)}_{h_m} \end{array} \right.$$

Note that upon convergence, when the residual on the finest level goes to zero, the term $R_{h_m}(u^{(0)}_{h_m}) \; + P_{h_m}$ in the above equation which can be rewritten as $R_{h_m}(u^{(0)}_{h_m}) \; + [\overline{R}_{h_m} - R_{h_m}(\overline{u}_{h_m})]$ goes as well to zero. Thus, no correction is computed on the coarser levels and driven back to the finest level.

- Updated solution on coarse grid h_m :

$$u_{h_m} = u_{h_m}^{(q)} \quad .$$

The accumulated corrections from each coarser grids are then successively passedback to finer levels by interpolation $(m = M, ..., 2)$. This represents the prolongation step :

- Transfer the correction from the grid level h_m to the next finer one h_{m-1}

$$u_{h_{m-1}}^{(+)} = u_{h_{m-1}} + I_{h_m}^{h_{m-1}} \; (u_{h_m}^{(+)} - \overline{u}_{h_m}) \quad avec \quad u_{h_M}^{(+)} \equiv u_{h_M} \quad ,$$

  where $I_{h_m}^{h_{m-1}}$ is the coarse to fine grid prolongation or interpolation operator from grid h_m to the next finer one h_{m-1}.

3.2 Transfer operators

The implemented strategies on hybrid grids include V and W cycles strategies. A sequence of coarse grids is extracted from the initial given fine grid. Further, the boundary conditions on the coarse grids are treated in the same way as in the fine grid.

Special attention is given to the intergrid transfer operators in the cell-centered formulations in which the variables are located at cell centers.Thus, the transferred variable locations change from one grid to another, which is not the case in a cell-vertex or node centered formulation.

The restriction operator exchanges information from a grid level to the next coarser mesh. The prolongation operator interpolates from a grid level to the next finer one. For the restriction operators, the standard approach is used. The restriction operator used for the conservative variables is the volume weighted average and the restriction of the residuals is accomplished by simple addition of the fine mesh residuals. The transfer of flow variables conserves mass, momentum and energy by the rule:

$$\overline{u}_{h_m} = \frac{\sum \Omega_{h_{m-1}} u_{h_{m-1}}}{\sum \Omega_{h_{m-1}}}$$

and the residual transferred to grid h_m is the sum of the residuals computed on the cells of the fine grid :

$$\overline{R}_{h_m} = \sum R_{h_{m-1}}^{(*)}$$

where the summations range over the cells on the fine grid composing each cell on the coarser grid.

The prolongation operator has to be defined only for conservative variables. In this work, a canonical prolongation operator is used. A direct injection of the coarse mesh values into the fine mesh is applied. This operator is very simple and easy to implement. Afterwards, this operator can be improved if defects of convergence are observed [6].

3.3 Strategy for turbulent flows

In the case of the RANS equations, the approach adopted is to compute the viscous terms on the coarser grids too. Thus, their influences are also taken into account in the forcing functions on the coarser grids. Different turbulence models are available in the solver ranging from algebraic models to two-equation models. These models are used to compute the turbulent quantities only on the finest grid level. On the coarser grids, they are obtained by interpolating the values from the finest level. This leads to a very direct approach with algebraic models, while with one or two equation models, the corresponding turbulence model equations are solved separately decoupled from the flow equations. In the solver, one Runge-Kutta iteration is carried out to update the turbulent quantitites on the fine grid. Thus, different new turbulence models can easily be included in the present environment.

In order to ensure robustness in V cycles without multigrid on turbulent quantities, sub-iterations are performed on the corresponding equations (let say 2 subiterations on $k - \omega$ system for a V cycle with 2 or 3 grids).

3.4 Agglomeration approach

The coarse mesh levels used in the multigrid algorithm are generated by an agglomeration technique. Grid cells are fused together to form a smaller set of larger polyhedral (polygonal in 2D) control volumes. One of the main difficulties of this technique is the selection of the cells to be agglomerated so that the newly generated cells have acceptable aspect ratios. The grid coarsening method is based on localization of the grid cells on an octree data structure.Three dimensional unstructured/structured grid cells are agglomerated via an octree data structure to obtain the coarse grid levels. The grid coarsening algorithm refers to the Emel Mahmutyazicio thesis [12]. It can be divided into two steps. Octrees are spatial data structures that successively partition a region of space into 8 equally-sized octants. The first step is to form the octree data structure hierarchy of the input grid file. A parallelepiped (rectangle in 2D) is defined to contain the original mesh. Then, thanks to the octree algorithm (quadtree in 2D) the domain is decomposed into octants (quadrants in 2D), containing a specified number of cells of the previous finer mesh. Figures 1 and 2 show the resulting octree space partitioning for a 2D Naca0012 airfoil and a 3D M6 Onera wing. Then the cells contained in each octant are agglomerated and form a coarse cell. For the sake of numerical scheme accuracy, connexity of coarse cells is one of the constraints of the agglomeration algorithm. Most often, the agglomerated cells share a common face. The cells, in each octant, which are not connex, are agglomerated to the cells of an adjacent octant. Similar situations can be encountered when the size of an octant is greater than the local dimension of a body. It occurs frequently in the vicinity of the trailing edges where an octant could encompass cells on the upper side and on the lower side. In that case, the non connex resulting coarse cell must be divided into two single connex cells. If the two resulting cells are not large enough (insufficient number of fine elements), they are agglomerated to an adjacent cell. The figures 3 and 4 show the resulting agglomeration respectively for the whole domain and in the vicinity of the trailing edge of a Naca0012 airfoil. We notice that the original geometry of the airfoil is not affected by the coarsening. The agglomeration of the adjacent cells to the M6 wing is shown in figure 5. In 3D cases, the boundaries of coarse cells being also that on the finest cells, it is less easy than in 2D to distinguish the coarse cells. The process is repeated in the same way for the higher coarsening levels. For the second level, the reference grid is then the first level coarse cells. Figures 6 and 7 show two levels of coarsening around the Naca012 airfoil. In figure 7, we can see again that the airfoil geometry is not affected by the coarsening.

This algorithm is well adapted for the agglomeration of isotropic meshes, such as those composed of tetrahedral elements. For anisotropic meshes, e.g. prismatic elements in the boundary layer, it fails. The development of a node-based agglomeration scheme [17] is in progress and will be coupled to the octree algoritm to deal with a wide variety of meshes.

4 Time integration

4.1 Implicit scheme

Efficiency in the numerical solutions of non-linear systems requires implicit schemes. This section describes the time implicit discretization schemes implemented in the elsA-H hybrid software. An implicit LU-SSOR (Lower-Upper Symmetric Successive Over Relaxation) approximate factorization is implemented for 3D hybrid unstructured grids. A grid reordering algorithm is proposed for efficient implementation of the LU-SSOR method.

The implicit method consists in two steps: a step for the construction of the operator and another technique for its resolution. Here, the inviscid part of the implicit operator results from the linearization of the van Leer numerical flux and only concerns the first-order scheme. If the resolution of a block-matrix requires a relatively high unit cost, it allows us to obtain an important intrinsic efficiency. Nevertheless, in order to reduce the computational cost, a diagonalization of all or parts of the blocks of the implicit matrix is also considered.

For structured grids, the LU-SSOR sweeps are usually performed by using hyperplanes $i + j + k = C^{tt}$. The aim is to order the matrix in lower and upper triangular matrices. A forward sweep updates point (i, j, k) using already updated values at $(i-1, j, k)$, $(i, j-1, k)$,$(i, j, k-1)$ while a backward sweep uses $(i-1, j, k)$, $(i, j-1, k)$ and $(i, j, k-1)$. However, this procedure is particularly well adapted to structured grids and does not extend easily to unstructured grids. In order to get a similar LU-SSOR algorithm for unstructured grids, a special grid re-ordering procedure is required.

This re-ordering procedure was proposed by Soetrisno et al. [18]. The cells are tidied up by planes. The member i of the plane p is the cell n and is given by the new cell numbering system (i, p).

The resolution of the implicit operator is based on an approximation of the exact matrix $(L + D + U)$ by $(L + D)D^{-1}(U + D)$. The system is approximated by a method of relaxation with forward and backward sweeps across the domain. Each relaxation cycle can be written in terms of two stages:

$$(L + D)\Delta W^{(p+1/2)} = -Rhs - U\Delta W^{(p)}, \tag{1}$$

$$(U + D)\Delta W^{(p+1)} = -Rhs - L\Delta W^{(p+1/2)}, \tag{2}$$

The previous reordering of cells provides a clear definition of the lower and upper matrices L and U and the proposed algorithm can be applied.

For approximate implicit methods, the consistency of having the same numerical scheme for both implicit and explicit operators is not required, since the spatial accuracy of converged steady-state solutions depends only on the explicit operator. In order to guarantee the diagonal dominance, an implicit operator based on a first-order accurate upwind scheme is implemented. Here, a linearization of van Leer's FVS scheme is considered.

A scalar version of the LU-SSOR method is also implemented. The matrix D is then defined with diagonal blocks and depends on the spectral radius of the jacobian matrices of convective flux. The off-diagonal terms are calculated by using the homogeneity properties of an upwind explicit flux [19].

5 Numerical results

5.1 Two-dimensional structured grids

The unstructured MG algorithm has been tested on several 2D quadrilateral meshes, so that the unstructured solution can be compared to that of the structured MG solver. It allows us to evaluate the effect of the various simplications made in the transfer operators and to know if these ones are enough to insure good convergence acceleration properties.

The first application concerns the inviscid flow around a 2D NACA 0012 airfoil. The original structured grid is composed of 8192 cells (256×32). Figure 9 compares the residuals obtained from the inviscid simulation of a 2D NACA 0012 airfoil using the structured (dashed lines) and unstructured (solid lines) solvers respectively. The flow conditions correspond to transonic inviscid flow at Mach number 0.85 and angle of attack of 1^o. The simulation was run using Roe's scheme and Runge-Kutta time-stepping. The calculations have been performed using V and W cycles. Note that when a single coarse level is used the behaviour of both solvers is essentially the same. The differences between both solvers then increase when two levels are considered, and in particular when the w-cycle algorithm is used. Despite the simplifications adopted in the unstructured method, these preliminary results demonstrate the overall good performance of the newly implemented MG unstructured solver.

The second application is the turbulent flow over a 2D bump. The studies in wind tunnel relative to this case were made in the transonic channel S8A at Onera [20]. The experimental channel is 120 mm wide and 100 mm heigh. A profile representing symmetric bumps is arranged on the lower and upper walls of the channel to accelerate the flow up to slightly supersonic speeds. A second sonic throat of adjustable section is placed at the channel exit to create shock wave by effect of blocking, and isolates the flow in the channel of possible perturbations which could come from the downstream. In the calculations we shall not represent this second throat, and the effect of blocking will be obtained from the level of prescribed outlet pressure which we shall impose. The computational and boundary conditions are the following : $Re_{i0} = 2.078\ 10^6$, $T_{i0} = 300\ K$, $P_{i0} = 96.\ 10^3\ Pa$.

The index "$i0$" denotes the inlet stagnation state. At the inlet boundary, total pressure P_{i0} , total temperature T_{i0} and the velocity direction aligned with the channel are prescribed. At the outlet the static pressure P_s is prescribed with different values in order to fit experimental and computational shock location : $0.630 \leq \frac{P_s}{P_{i0}} \leq 0.650$.

The computational domain represents the half of the channel with a symmetry condition on the upper boundary. An adiabatic no-slip condition is applied on the lower wall. The basic mesh is an hexahedral mesh composed of 180x64 cells. This calculation has been performed using a k-ω model. The interest of this test case is the consideration of dissipative effects. One more, the convergence rate is satisfactory (figure 10).

5.2 Unstructured two-dimensional grid

The inviscid flow around the bidimensional NACA 0012 airfoil is now simulated with an unstructured grid. The mesh consists of 68620 triangles. The two grid levels are obtained by application of the octree based grid coarsening method to a triangular mesh generated using Delaunay triangulation (figure 7). These coarse grids contain 26094 and 9194 cells from the second to the third grid levels respectively. The coarser grid level is shown in black color. The simulation was run using Jameson's scheme. The calculations have been performed with V and W cycles. Figure 11 illustrates convergence histories performed with the implicit LUSSOR method respectively without multigrid (1 grid), and with multigrid with two coarse grid levels. For this test case, we have a ratio gain of almost 8 for a 2 coarse grid computation with a W cycle.

5.3 Unstructured grid over ONERA M6 wing

The Onera M6 wing is a basic 3D test case widely presented in the literature in order to validate numerical methods. The flow field is computed here by the solution of the Euler equations at a free Mach number of 0.836, an angle of attack of 6.06^o. Spatial discretization is achieved using the Jameson centered scheme, and time integration is carried out using the implicit LUSSOR technique. The unstructured grid is shown in figure 12 contains 176750 cells. The two levels of coarse grids are generated with the octree based grid coarsening technique. The coarse grids contain 18461 and 1969 cells. Figure 13 shows a very good convergence acceleration for the multigrid computations. For this test case, we have a ratio gain of 10 for a 2 coarse grid computation with a W cycle.

6 Conclusion

An agglomeration multigrid procedure has been developed in the context of hybrid grids in the *elsA-H* solver and tested on bidimensional and tridimensional configurations. At first, inviscid and turbulent flow calculations on several 2D quadrilateral meshes have been performed in order to evaluate the tranfer operators. Satisfactory results have been obtained. Then, the agglomeration technique based on the localization of unstructured grid cells on an octree data structure has been tested for 2D and 3D inviscid flow simulations. It provides large convergence acceleration. The method has now to be extended to 3D turbulent flow simulations. IEven though this algorithm is well adapted to the agglomeration of isotropic meshes, such as those composed of tetrahedral elements, it usually fails for anisotropic meshes with prismatic elements in the boundary layer. The development of a node-based agglomeration scheme is in progress and will be coupled to the octree algorithm to deal with a wide variety of meshes.

References

[1] L. Cambier and M. Gazaix. elsa : An efficient object-oriented solution to cfd complexity. AIAA Paper 2002–0108, 2002.

[2] G. Puigt, M. Gazaix, M. Montagnac, M.-C. Le Pape, M. de la Llave Plata, C. Marmignon, J.F. Boussuge and V. Couaillier. Development of a new hybrid compressible solver inside the CFD elsA sofware. AIAA Paper 2011–3379, 2011.

[3] M. de la Llave Plata, V. Couaillier, M.-C. Le Pape, C. Marmignon and M. Gazaix. An all-in-one structured/unstructured solver for the simulation of internal and external flows. Application to turbomachinery. Proceedings of the 4th European Conference for Aerospace Sciences, 2011.

[4] M. de la Llave Plata, V. Couaillier, M.-C. Le Pape, C. Marmignon, M. Gazaix and B. Cantaloube. Further developments in the multiblock hybrid CFD solver elsa-H. AIAA Paper 2012–1112, 2012.

[5] D.J. Mavriplis. Multigrid techniques for unstructured Meshes. ICASE Report No 95-27, 1995.

[6] E.D.V. Bigarella, J.L.F. Azevedo and L.C. Scalabrin. Centered and upwind multigrid turbulent flow simulations of launch vehicle configurations. *Journal of Spacecraft and Rockets*, 44(1), 2007.

[7] E. Mahmutyazicioglu, I.H. Tuncer and H. Aksel. Octree based unstructured grid coarsening method for 3D multigrid applications. ECCOMAS CFD, 2010.

[8] D.J. Mavriplis. Multigrid solutions of the two-dimensional Euler equations on unstructured triangular meshes. *AIAA Journal*, 26(7):824–831, 1988.

[9] D.J. Mavriplis, A. Jameson. Multigrid solutions of the Navier-Stokes equations on triangular meshes. *AIAA Journal*, 28(8):1415–1425, 1990.

[10] D.J. Mavriplis, V. Venkatakrishnan. Agglomeration multigrid for viscous turbulent flows. Proceedings of the 25th AIAA Fluid Dynamics Conference, 1994.

[11] M. Lallemand, H. Stave, A. Dervieux. Unstructured multigriding by volume agglomeration: current status. *Journal of Spacecraft and Rockets*, 21(3):397–433, 1992.

[12] Emel Mahmutyazicioglu. *Development of an octree based grid coarsening and multigrid flow solution.* PhD thesis, Middle East Technical University, Turkey, 2010.

[13] A. Jameson and T. J. Baker. Improvements to the aircraft Euler method. Aiaa paper 1987–0452, 1987.

[14] A. Jameson, W. Schmidt and E. Turkel. Approximate Riemann solvers, parameter vectors and different schemes. AIAA Paper 1981–1259, 1981.

[15] P.L. Roe. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *Journal of Computational Physics*, 43(2):357–372, 1982.

[16] steering sub-committee of the AIAA CFD committee on standards. The CFD General Notation System. Standard Interface Data Structures. AIAA r-101a-2005, 2007.

[17] P. Sumant, A. Cangellaris, N. Aluru. A node-based agglomeration AMG solver for linear elasticity in thin bodies. *Numerical Methods in Engineering*, 25(3):219–236, 2009.

[18] M. Soetrisno, S. T. Imlay and D. W. Roberts. A zonal implicit procedure for hybrid structured-unstructured grid. AIAA Paper 1994-0645, Reno, Nevada, 1994.

[19] G. Alléon, S. Champagneux, G. Chevalier, L. Giraud and G. Sylvand. Parallel distributed numerical simulations in aeronautic applications. *Applied Mathematical Modelling*, 30:714–730, 2006.

[20] J. Délery. Experimental investigation of turbulent properties in transonic shock/boundary layer interaction. *AIAA Journal*, 21(3), 1983.

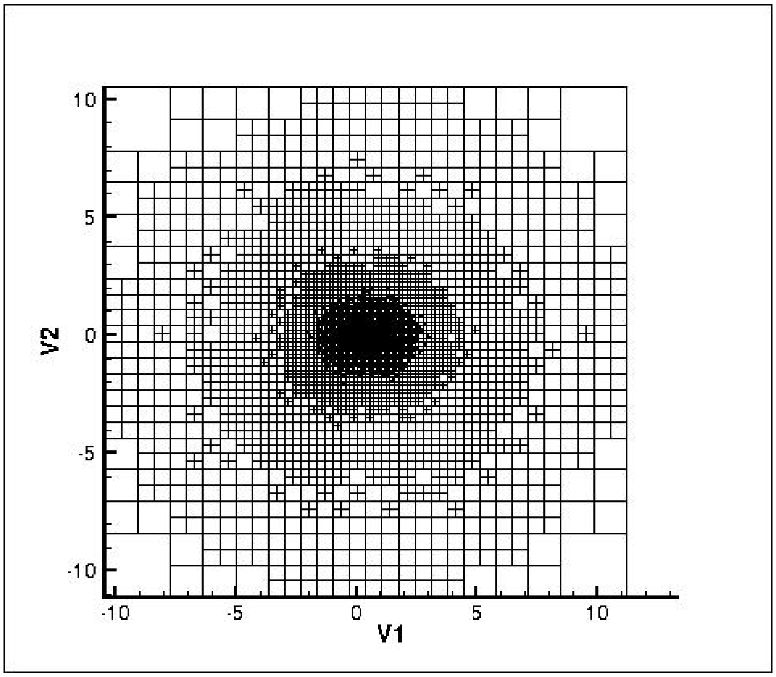

Figure 1: Naca0012 quadtree discretization

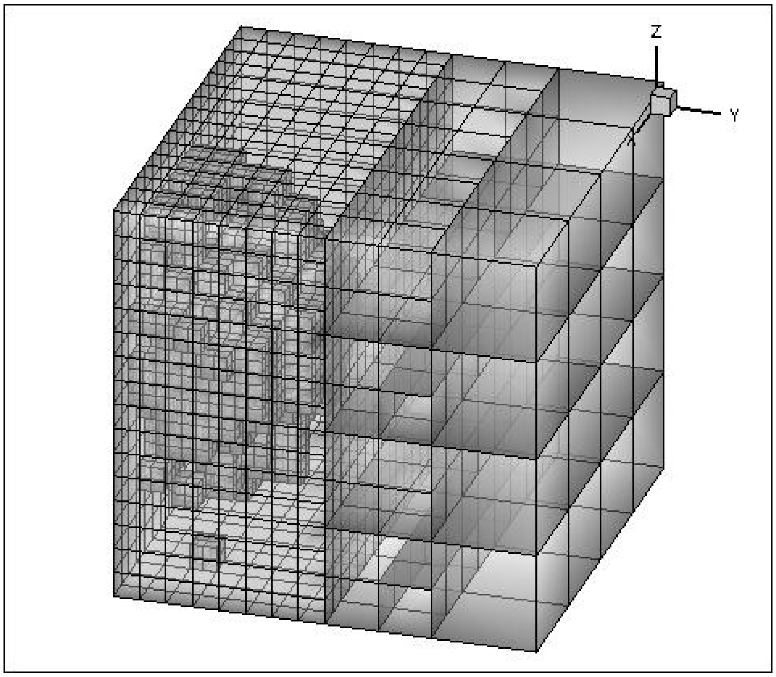

Figure 2: M6 wing octree discretization

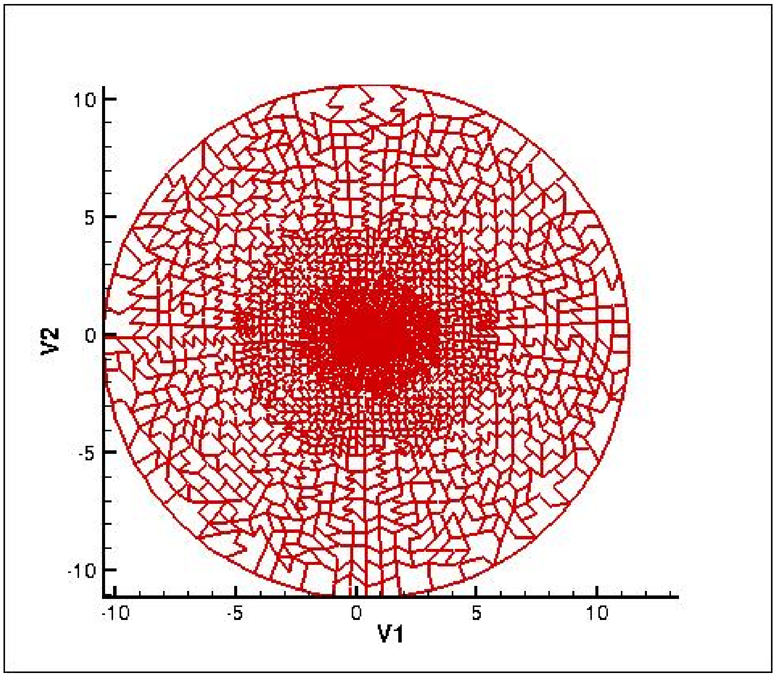

Figure 3: First level agglomerated cells

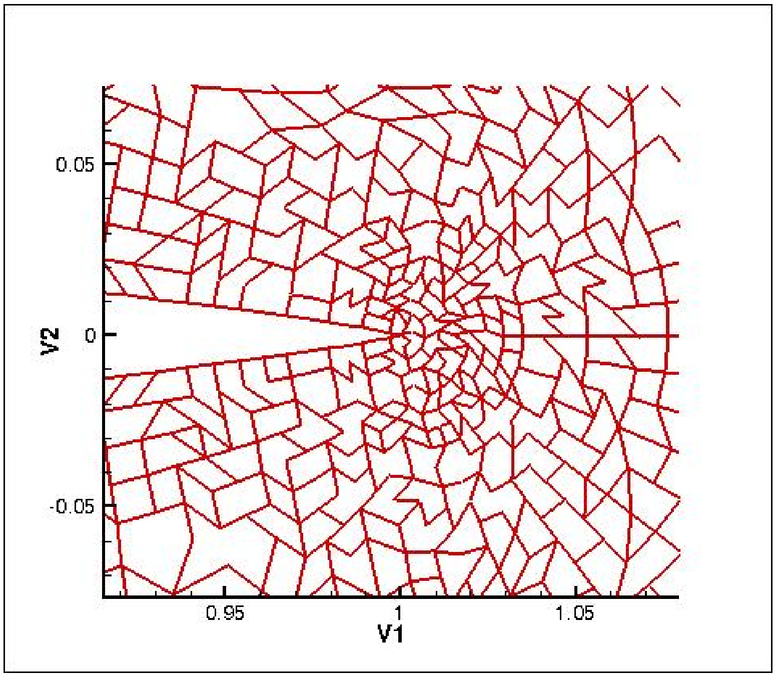

Figure 4: First level agglomerated cells, trailing edge close view

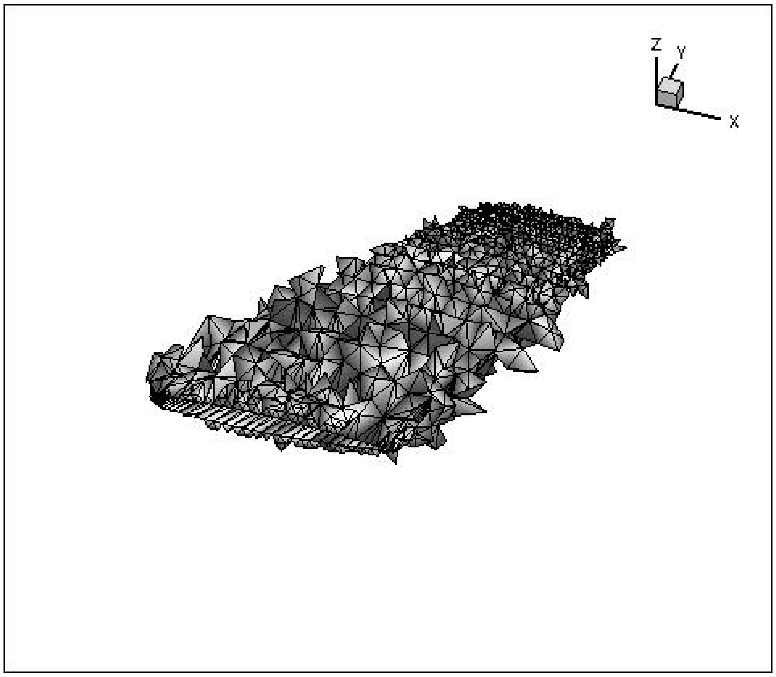

Figure 5: First level agglomerated cells adjacent to the M6 wing

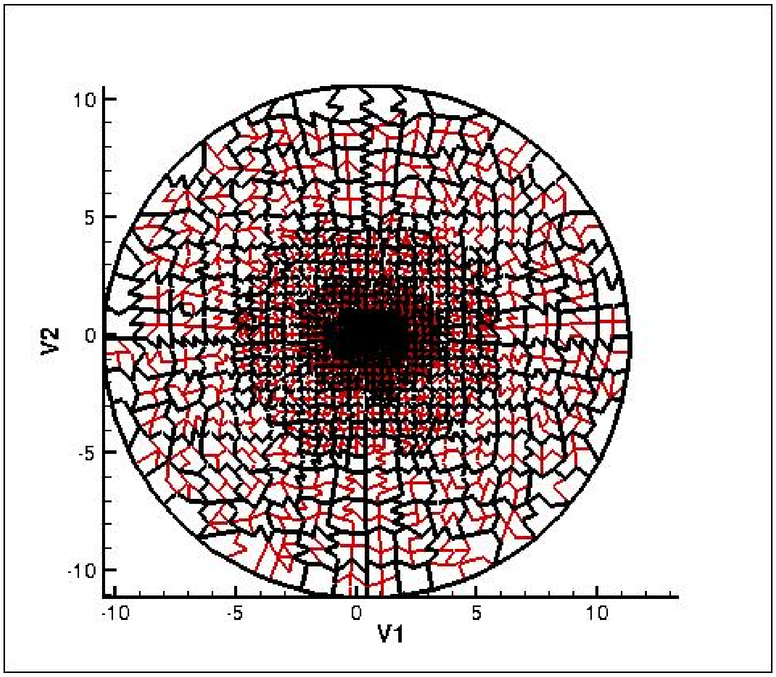

Figure 6: Coarse grid levels

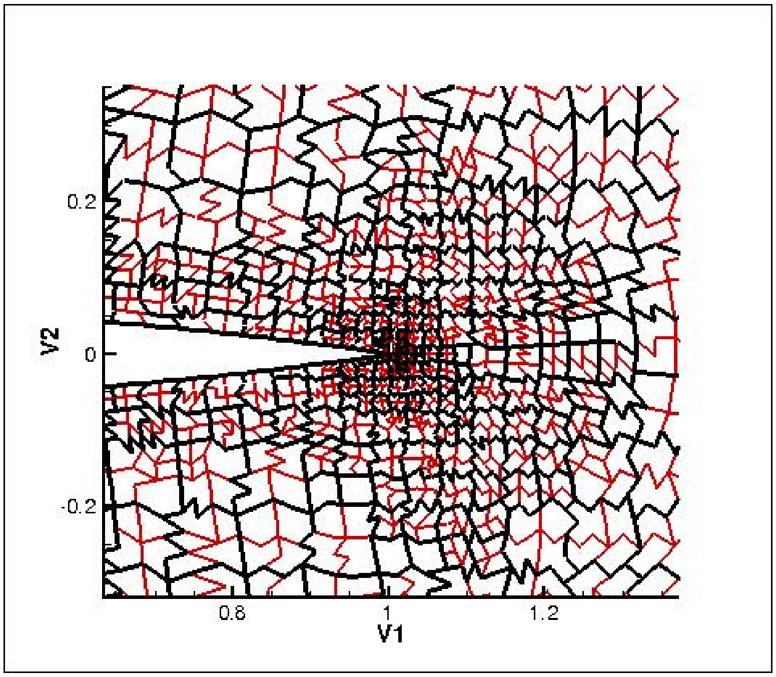

Figure 7: Coarse grid levels, trailing edge close view

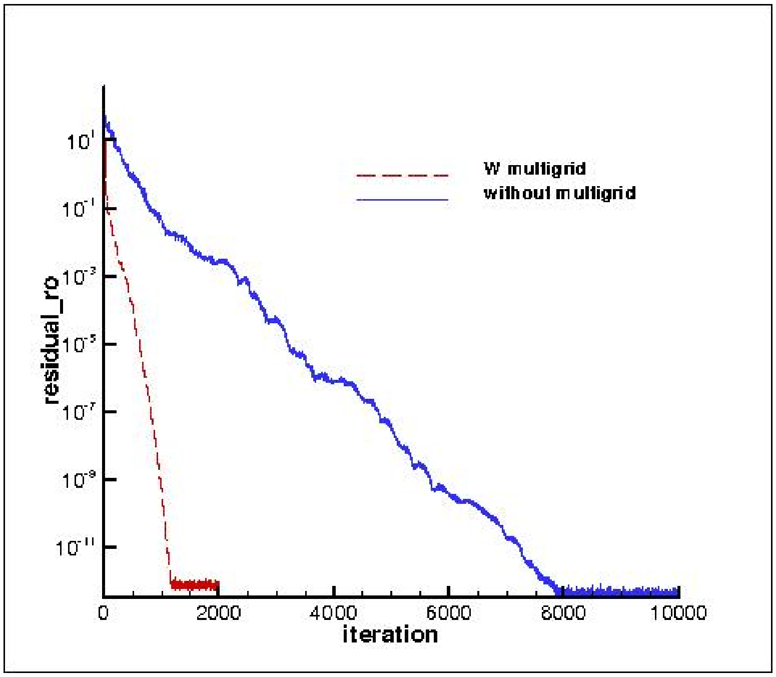

Figure 8: Two coarse levels

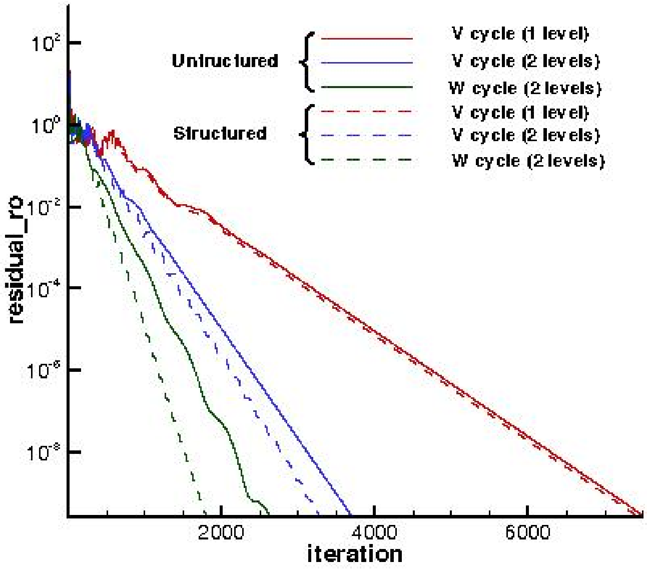

Figure 9: Evolution of residuals - NACA 0012 airfoil.

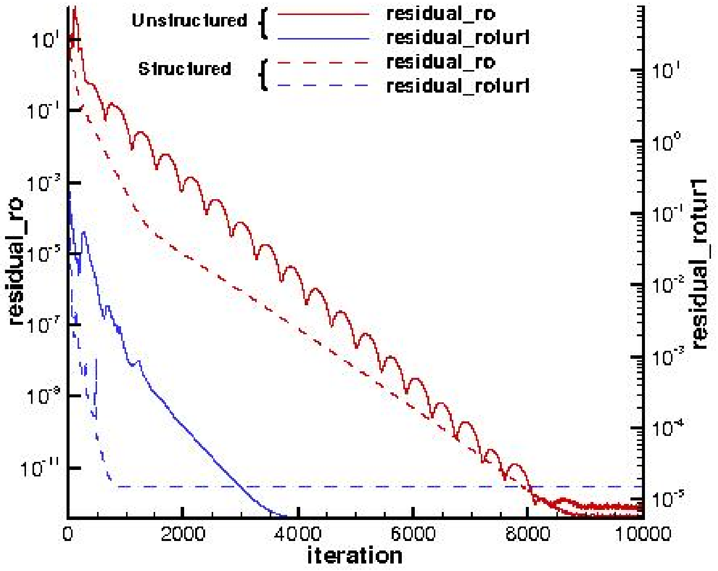

Figure 10: Evolution of residuals - 2D bump.

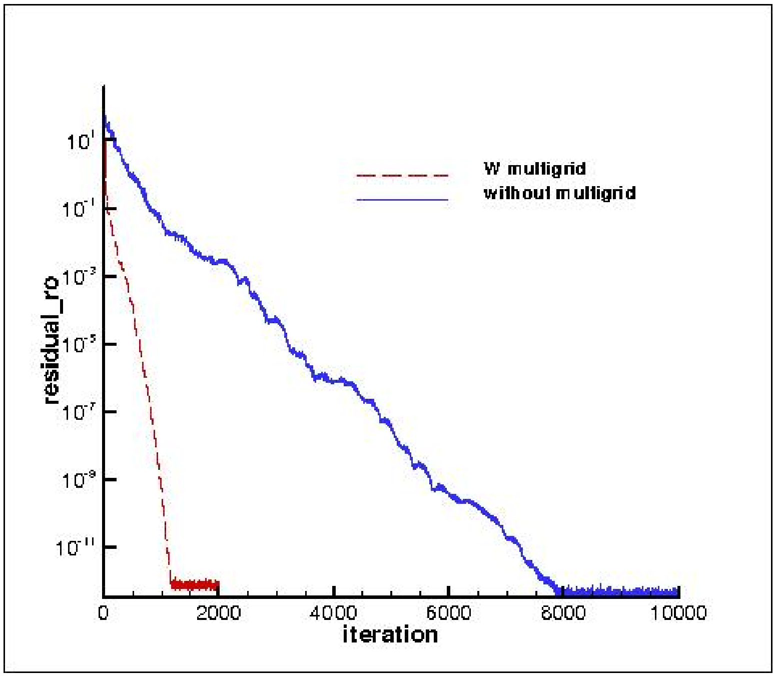

Figure 11: Evolution of the residuals - unstructured Naca0012 grid

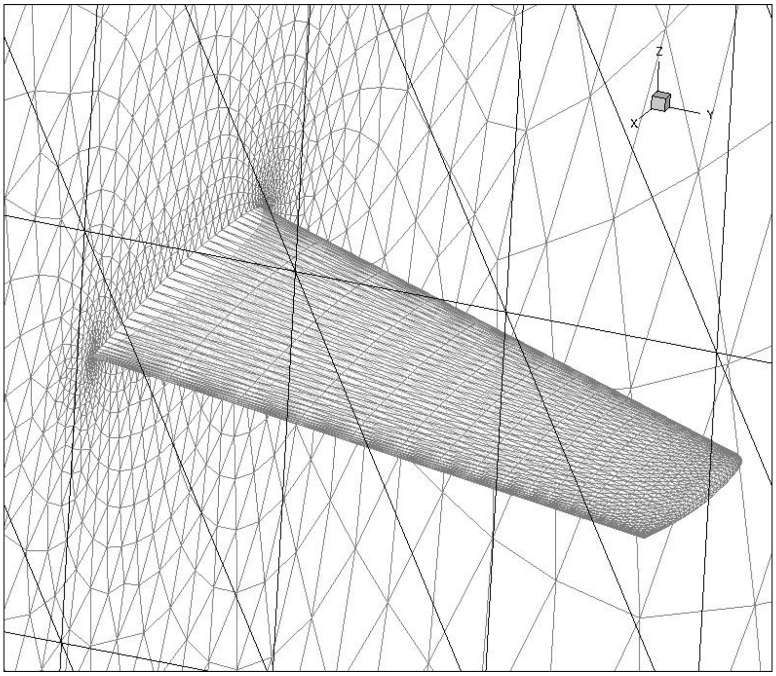

Figure 12: Mesh of Onera M6 wing.

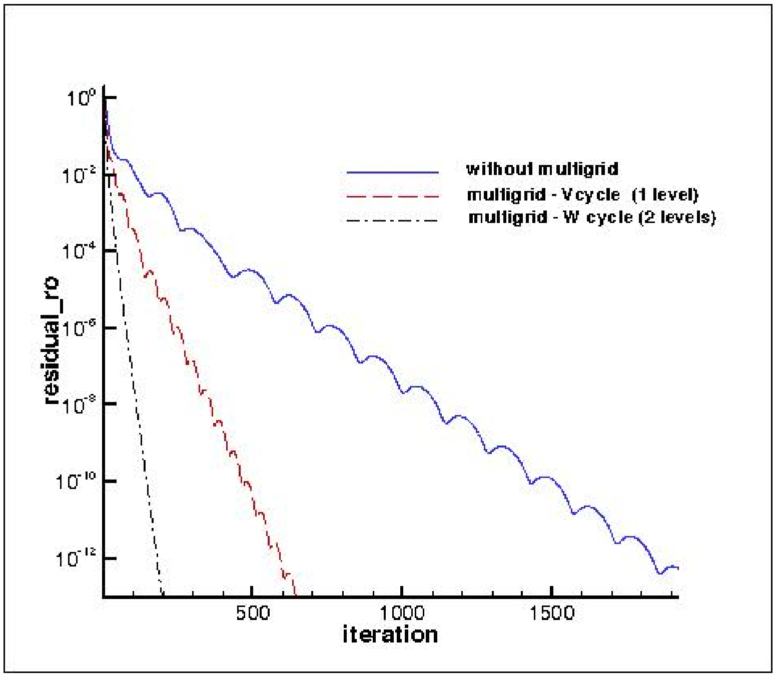

Figure 13: Evolution of the residuals - M6 wing.