ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

SINGULAR MESH GENERATION FROM MULTIPLE OVERSET MESHES: A TOOL FOR INDUSTRIAL APPLICATIONS

Gennaro Abbruzzese[1], Marta Cordero-Gracia[2], Mariola Gómez[3] and Nuno Vinha[4]

Departamento Matemática Aplicada y Estadistica ETSI Aeronauticos,
Universidad Politécnica de Madrid, Spain
[1]gennaro.abbruzzese@upm.es
[2]marta.cordero@upm.es
[3]mariola.gomez@upm.es
[4]n.vinha@upm.es

Keywords: overset, mesh, generation, remeshing, moving, chimera, finite volume.

Abstract. *In this work we present a strategy for single mesh generation from multiple overset grids for an unstructured finite volume vertex-centered Navier-Stokes equations solver. The final aim is to slim down the mesh generation process for complex industrial cases, especially in case of mesh displacement, and boundary layer overlapping where Chimera interpolation schemes are often applied. At the same time the tool developed has to take place in between of the mesh generation and solver, in such a way that it is possible to work on the final mesh with no modification of the grid generator code. The validity of the approach will be verified on the basis of the results obtained by running DLR-TAU solver, directly comparing them with the outcome of Chimera interpolation routines.*

1 INTRODUCTION

Mesh generation can be considered the real bottleneck in the CFD analysis process in industry since meshing time consumption widely exceed the solver running time. Furthermore each solver needs a particular mesh, not only for quality requirement but also for topological requirements. The impossibility of the solvers to process any kind of mesh or damaged mesh can cause a huge time loss from the mesh generation side. In academia the problems coming from a bad mesh are generally overcome by employing refined and up to date routines but in an industrial environment often is not possible to change the tools or put the hands in the source code of the software employed. This is the key to understand the need of a tool which in the CFD investigation process is placed between the mesh generator and the solver.

The intent of this work is to develop a method to build/optimize meshes employed by vertex-centered DLR-TAU solver in such a way that it could be used to process any kind of overset, hybrid, non-matching boundaries, damaged or negative volume grid. The scheme adopted by DLR-TAU comes from the family of the Godunov type methods [1], wide spread for compressible flow prediction, which smooths the fluxes distribution solving the Riemann problem between each cell.

Common practices suggest to use classical geometrical quality criteria to evaluate computational grids. Especially for schemes as vertex-centered finite volume, which manipulate the initial mesh, this evaluation is not sufficient if not misleading. A good looking mesh does not bring always an accurate solution. However the influence of mesh features on the CFD predictions are far from being clear [2] [3] [4] [5] and we lack of actual guidelines to build a high quality mesh.

For viscous flow calculations special feature for the cells are desirable, one being a structured aligned hexahedral mesh in boundary layers. This brings up to several issues and obstacles with regards to automatic of the mesh generation, for example:

- Mesh generation tools often have problems dealing with complex geometries;
- In moving meshes or in the case of relocating components, the mesh must be regenerated;
- The presence of a structured mesh creates a huge amount of cells which are far from boundary layers, for example in the wake and in the free stream regions around aerofoils.

In order to overcome lack of accuracy or resolution in a mesh, generally techniques are employed as adaptation to increase resolution, a-posteriori evaluation of the mesh, accurate schemes on the solver side or interpolation of solution on overlapping grids (Chimera approach). In particular, this last approach is gaining popularity in the industry and interest from the academic community and DLR-TAU has a robust implementation of this scheme. It consists in a set of high quality meshes generated independently and communicating by mean of interpolation procedures in overlapping zones and presents some desirable properties as, on the other hand, some disadvantages [7][8]:

- Provide good quality mesh,
- It is versatile even for highly complex geometries,
- Relatively well tested,
- Allows to skip moving parts re-meshing.

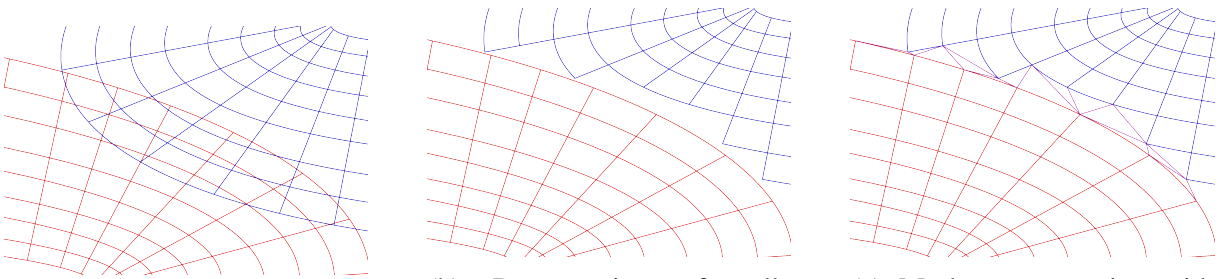
But:

- Higher computational effort,
- Poor solver performance,
- Increased problem setup time,
- Flux non conservative in presence of interpolation and non-matching boundaries.

An attempt to bypass interpolation problems among overset grids and to generate a single dual mesh has been done by Sørensen [9]. In his paper the author builds an algorithm in ten steps to merge two different grids in a single one. The intention of the author is to potentially fully automate the meshing process, to employ the same solver of a single component grid, to avoid interpolation and to have control on the whole domain mesh. Gremmo *et al.*[10] have presented research notes about the implementation of a tool for partial remeshing of background mesh when a solid body and relative mesh are subject to large displacements. In the following paragraph a sequence of actions are shown, to build incrementally a multi-component mesh for a three component *GARTEUR* A310 aerofoil [11][12].

2 THE ALGORITHM

The idea at the basis of the algorithm we are looking for is similar to the one coming from Sørensen [9] with a focus on primary mesh, which makes the implementation simpler and robust. The idea is still to start from two or more grids and supply a single one for the reasons explained above. The outcome of the process has to present the desired features minimising the eventual distortion. The problem is in principle simple to solve, but in practice presents a multitude of issues. First step is, of course identify the zones covered by the different grids. Once identified the new boundaries of each mesh, it is necessary to fill the holes corresponding to the degenerated cells in proximity of the interfaces (fig.1).



(a) Overlapping meshes.

(b) Degeneration of cells around interface.

(c) Mesh reconstruction with constrained triangulation.

Figure 1: Degeneration and reconstruction process in the interface proximity

2.1 Interface definition and cells degeneration

Modifications of the topology of the two meshes will be applied in the overlapping zones. Once identified, these areas must be transformed in single meshes, always taking into account the constraints coming from the non-overlapping meshes. The reason is that it is preferable to maintain the original meshes untouched where possible, since we can assume that these are

generated in order to fulfill some criteria imposed by the user for the particular case. This point raises another problem when the meshes have equal priority. If we try to maintain the topology of the original meshes and at the same time we try to modify the smallest possible number of cells in the mesh we will have to choose only one among the meshes in each zone in the entire overlapping region. The objective is then to split the overlapping area by mean of area interfaces based on mesh priority criteria. The employment of segments in two dimensions or surfaces in the three dimensional case as interface allows us to degenerate the minimum amount of cells in the entire domain and to substitute the cells of each mesh with higher quality cells coming from other mesh. There are many possible formulations for obtaining the interface. A way to define such interface is by mean of a criterion based on the distance from the walls. Simply, the elements lying in the overlapping zones and having points closer to a wall belonging to another mesh will lose priority and replaced by cells from the other mesh or, if needed, by new cells specifically created.

With this approach:

- It allows to deal with the problem of cells with equal priority (and it should be employed for this cases only);
- User defined cells like boundary layer structured mesh for viscous flow simulations will remain untouched as much as possible;
- Assuming that the grids are all built for the same purpose, it is reasonable to think that they will have same y^+ and same growth ratio and, then, that their cells will have similar dimension in proximity of the interface.

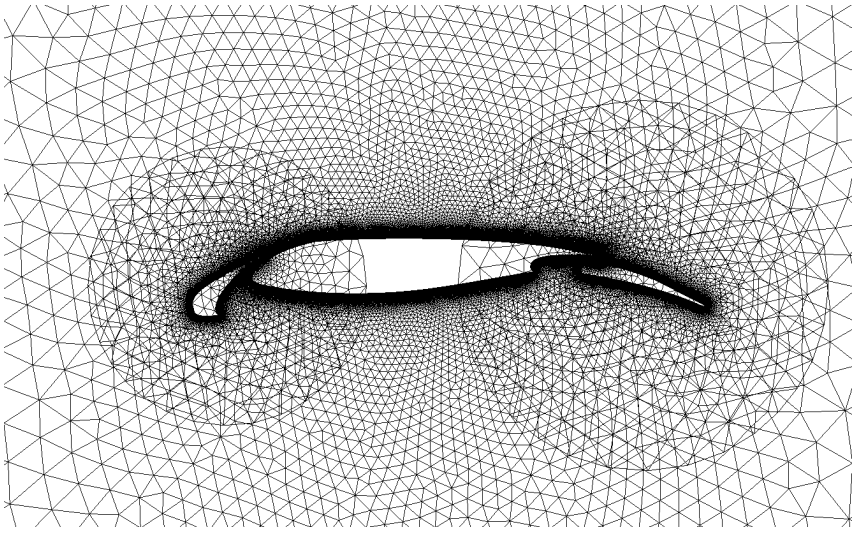


Figure 2: Overlapping meshes.

The outcome of such degeneration is shown in figure 3. It is possible to see that the slat lies in the boundary layer of the wing. This is a typical case where unstructured grid generation software find difficult to handle the complexity of the geometry. In the picture 4 the median line between the two walls is shown which is implicitly defined by the criterion used. All the cells crossing the median line are deleted. By the way the cells to be eliminated have been marked with a simple distance check and with the AABB tree routine in *CGAL* [13] to speed up the calculation.

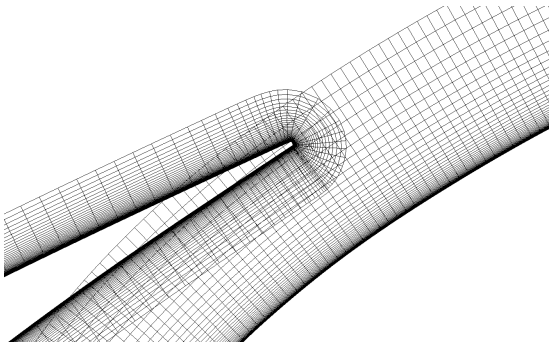


Figure 3: Overlapping boundary layers of wing and slat.

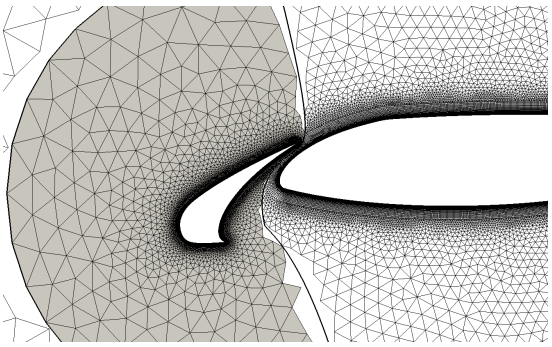


Figure 4: Degeneration of cells external to area defined by distance from teh walls.

2.2 Single mesh reconstruction

The simplest and most efficient way to reconstruct the mesh in arbitrary shaped hull (an example in fig.5) is the triangulation. Of course we can expect that some bad quality elements will be created in this way, but also it will be possible to apply smoothing tools once the single mesh will be created. For this reason we resorted to the *CGAL* [13] library. The use of this tool will guarantee a high quality grid as output and a high level of automation of this step. After the degeneration the two meshes are not overlapping anymore and they are separated by an empty hull. In order to refill this hull *CGAL* constrained Delaunay 2D triangulation routine has been employed. This triangulation still creates a convex polygon as a simple Delaunay triangulation but forces the creation of the edges on some constraints. In this case the borders of the void hull are imposed as constraints in order to obtain a triangulation which matches original meshes elements. The polygon cells outside the void hull borders are ignored and the remaining are added to the full mesh to patch the two degenerated meshes. The full single mesh is in this way generated.

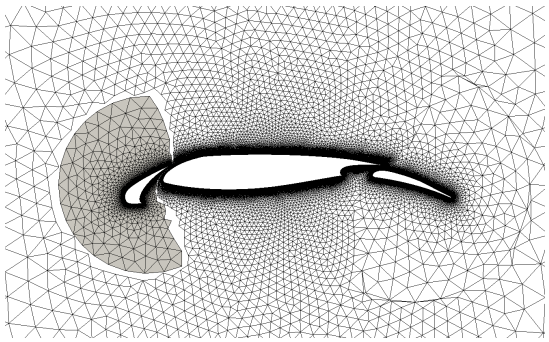


Figure 5: Slat mesh added to the wing background mesh right before joining.

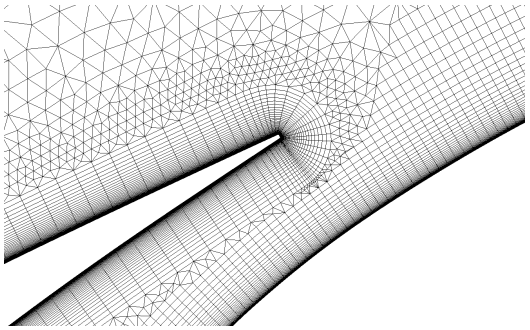


Figure 6: Boundary layers of wing and slat.

In the picture 5 it is possible to see the mesh where the flap mesh has been already degenerated and mended and the same process has to be applied to the slat. In the figure 6, the mesh between the slat and the wing has been magnified. As predicted the triangulation in this zone has a favorable quality since it has been preformed in a space defined by elements of comparable dimension. In the figure 7 we can see the full single mesh after reconstruction and observe that the same said for near wall zones is not valid walking away from the walls. Furthermore, the cells have been intentionally left untreated to check the influence of such small amount of

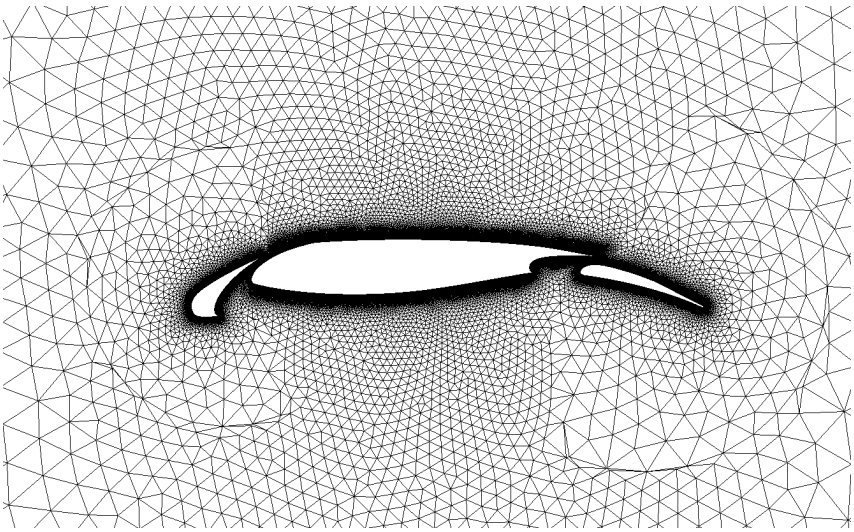


Figure 7: Final incrementally generated mesh.

bad cells. In this case for example, the cells created in the process, by the addition of the slat, are 292 out of a total number of 191025 cells in the final full mesh.

3 A BRIEF TEST

3.1 Cases setup

In this paragraph we show a small test with *DLR-TAU* finite volume Navier-Stokes equations solver. The simulations have been run in order to get an idea of the possible loss of accuracy of the prediction in a manipulated mesh comparing it with a mesh fully generated simultaneously for the three components by the automated mesh generator available in *CENTAUR*[14]. The mesh is shown in figure 8. We can notice that, although it has a number of points comparable to the one of the incrementally generated mesh (tab.1), it has a different distribution of points. From the zoom of the boundary layers between slat and wing in fig.9 we can observe that the complexity of the geometry is an obstacle for the mesh generator since the growth of the boundary layers is limited by the space between slat and wing.

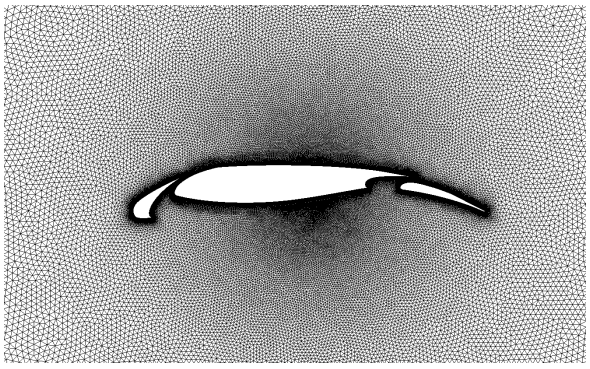


Figure 8: *CENTAUR* mesh simultaneously generated for the three components.

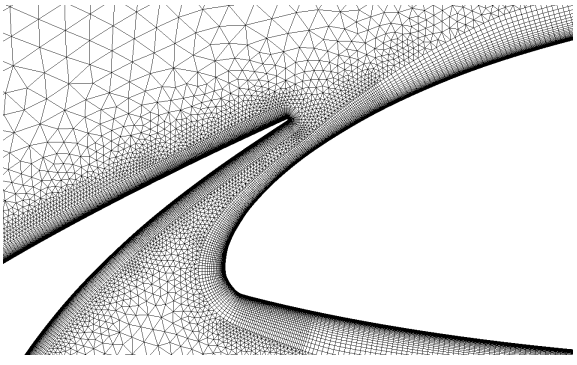


Figure 9: Zoom of the boundary layers between slat and wing in *CENTAUR* mesh simultaneously generated for the three components.

Furthermore, in order to evaluate the advantages coming from the presented method, the same meshes of the components individually generated (still with *CENTAUR*) have been used to run a Chimera simulation without applying any topology modification (fig. 2). The table 1 shows the number of points of the meshes.

Mesh	Number of points
Incremental	***345740***
3-Components	***330784***
Chimera (slat)	*29892*
Chimera (wing)	*165212*
Chimera (flap)	*156722*
Chimera (total)	***351826***

Table 1: Meshes dimension.

The cases are a quasi 2D simulations with surface mesh with $y = 0$ extruded to $y = 1$. Boundary conditions imposed in TAU are *farfield* boundary conditions for farfield faces, *viscous wall* for the solid bodies faces and *symmetry plane* for the plane in constant $y = 0$ and $y = 1$ [15]. All the meshes have been run with Spalart-Allmaras one-equation turbulence model under the conditions summarized in table.2.

operating conditions	
Re	0.605×10^7
Mach	0.224
AoA	12.19°
wing configuration	*take-off*

Table 2: Operating conditions for the three cases.

3.2 Results

The aim of this test is to reveal any possible malfunctioning of the mesh incrementally generated with the method explained above. It is certainly not detailed enough to fully validate the newly assembled mesh but as we will see, it becomes useful to confirm some valuable features of the algorithm. All the simulations have been run until convergence of the lift and drag coefficients. It is important to point out that the Chimera scheme could not reach a calculation speed over 35% of the speed of the scheme employed in the other two simulations. All of them have shown a rapid convergence of these parameters while the residuals of the density seem to have encountered a resistance around 10E-5.

Figure 10 shows the plot of lift coefficient history for the three simulations. We can observe that all of them are converging rapidly with a small percentage of error to the final value. The *CENTAUR* mesh has at the beginning high fluctuations of the value while the other two have a smoother behavior. With this data is still not possible to determine if the smooth curves are similar because of the similarities in mesh topology or to a higher amount of cells in the boundary layers. The converged values of the lift coefficient are summarized in the table 3 with the error relative to the experimental results [11].

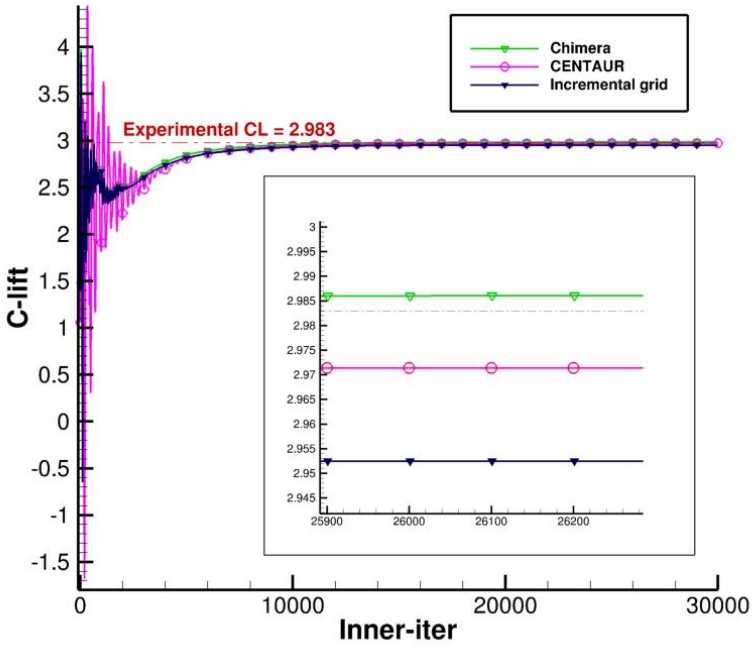
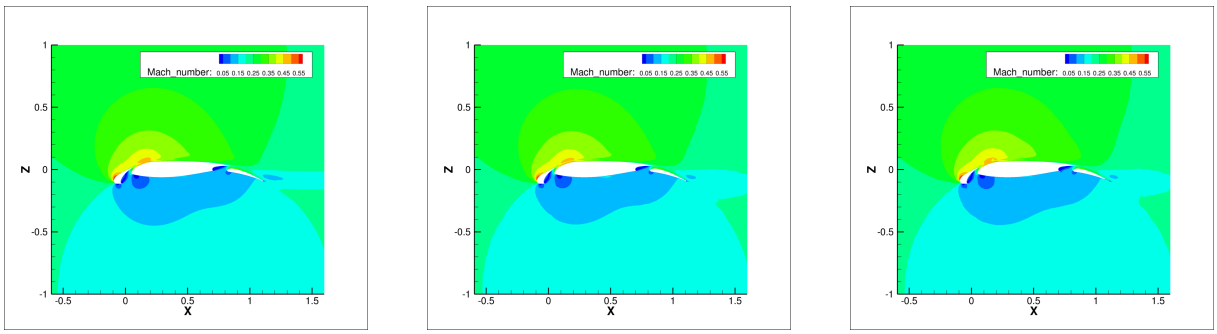


Figure 10: Lift coefficient history for the three simulations.

lift coefficients (exp = 2.983)		
	CL	*err(%)*
Incremental	2.953	1.006
CENTAUR	2.971	0.4022
CHIMERA	2.987	0.4

Table 3: Lift coefficient result and error relative to experimetal results [11].

The mesh run with Chimera scheme seems to have the best behavior among the three. Still the data is not sufficent for a verdict. With only this test is not possible to determine if the error is sistematic. What is possible to see, nevertheless, is that the assembled mesh has an error on the prediction which does not overcome the 1%. Furthermore, although no mesh smoothing has been applied, the convergence is not affected by the bad cells generated with constrained triangulation, as it could be expected. We continue the observation of the results with the plot of the contours of the Mach number (fig.11). This time the Chimera seems to have an inferior performance among the three, showing a ”pleating” of the contours in proximity of the wing downwind to the slat, whereas the assembled mesh seems to be the one which better captures the gradient between the flows coming from the upper part of the slat and the channel composed by slat and wing.



(a) *CENTAUR* mesh. (b) Incremental mesh. (c) Chimera mesh.

Figure 11: Contours of the Mach number for the three simulations

4 CONCLUSIONS

A method for mesh assebling by incremental components addition has been implemented and tested. With the small amount of data available is not possible to have a verdict on the performances once the mesh is used for aerofoil flow simulation. Nevertheless, it has been shown that the method has some valuable features. The algorithm is a simple way to give flexibility to the meshing problems when neither the solver nor the mesh generator codes are available. It is easy to automate and presents some of the advantages of the Chimera approach but with no need to introduce interpolation with a remarkable time saving and with a better accuracy for complex cases. Hence, it would be interesting to develop the algorithm and further studies can be carried out with a more comprehensive set of test-cases but also introducing a smoothing of the newly created cells and, of course, expanding it to real non symmetrical 3D cases.

REFERENCES

[1] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics : a practical introduction*. Springer, Berlin, New York, 1997.

[2] F. J. Dannenhoffer, *Correlation of grid quality metrics and solution accuracy for supersonic flows*. AIAA-2012-0610, 2012.

[3] B. Diskin and J. L. Thomas, *Accuracy of Gradient Reconstruction on Grids with High Aspect Ratio*. NIA Report No. 2008-12, 2008.

[4] B. Diskin and J. L. Thomas, *Mesh effects on accuracy of finite-volume discretization schemes*. AIAA-2012-0609, 2012.

[5] B. Diskin and J. L. Thomas, *Notes on accuracy of finite-volume discretization schemes on irregular grids*. Applied Numerical Mathematics, 2009.

[6] S. Crippa, *Application of novel hybrid mesh generation methodologies for improved unstructured cfd simulations*. 28th AIAA Applied Aerodynamics Conference - CFD Drag Prediction Workshop Results, 28 June - 1 July 2010, 2010.

[7] S. Crippa, *Application of chimera with hexahedral blocks in solar meshes*. Third Symposium Simulation of Wing and Nacelle Stall, 21st - 22nd June 2012, 2012.

[8] S. Crippa, *Application of Chimera with Hexahedral Blocks in Solar Grid*, PerPre-Deliverable 09.2011, $C^2A^2S^2E$, Institute of Aerodynamics and Flow Technology, DLR.

[9] K. A. Sørensen, *Multi-component modeling with the dual mesh connect approach*. Computational Flight Testing: Results of the Closing Symposium of the German Research Initiative ComFliTe, Braunschweig, Germany, June 11th-12th, 2012.

[10] S.Gremmo, I.Antonik, R.Nigro, C.Hirsch, and G.Coussement, *Partial remeshing strategy for cfd simulations when large displacements are taken into account*. 23rd International Meshing Roundtable (IMR23), Research Notes, 2014.

[11] F. Manie, O. Piccin, and J.P. Ray, *Test Report of the 2D model M1 in the Onera F1 Wind Tunnel*, GARTEUR High Lift Action Group AD (AG-08), TP041, October 1989.

[12] M.C.Iorio, L. M. Gonzalez and A. Martinez Cava, *Global Stability Analysis of a Compressible Turbulent Flow around a High-Lift Configuration*. 45th AIAA Fluid Dynamics Conference, 22-26 June 2015, Dallas, Texas, USA. pp. 1-25.

[13] *CGAL*, Computational Geometry Algorithms Library.

[14] *CENTAUR*, www.centaursoft.com.

[15] TAU-Code User Guide Release 2015.1.0