

Coupled Patched-Grid / AMR Meshing Technics for Transonic Aircraft Design.

Julien Bohbot*, Jean Christophe Jouhaud†, Denis Darracq‡

CERFACS - CFD Team, Toulouse (France)

40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2002

Session :Applied Aerodynamics

Abstract

Grid generation and grid refinement are crucial problems in the computation of flow around complex aircraft configurations. A coupling between two meshing technics have been implemented in a structured solver to make easier the grid generation and to reduce the number of nodes. The first technic is the patched grid (PG) possibility which allows no-coincident interfaces between block. PG is used conjointly with an Adaptive Mesh Refinement (AMR) technic to refine local regions. The efficiency of this kind of association in order to refine local region of interest are demonstrated with several referenced test cases. Finally, computations over the AS28G aircraft (fuselage, wing, pylon, nacelle) using the PG/AMR will be presented.

Introduction

During the last decade, meshing facilities and meshing refinement possibilities become the determining factor in the choice of a code solver. Unstructured solver is commonly used for their advantages in mesh generation and mesh refinement. However, the complexity to implement numerical algorithm and to avoid numerical dissipation, due to gradient calculation, limits their numerical possibility. In an other hand, structured solver give accurate results and are easier to implement a new numerical technic. In vectorized computer, structured solver is less CPU time consuming than unstructured solver.

Structured grids require common coincident interfaces between block which imposes constraint on the

grid generation and consequently mesh refinement (boundary layer refinement, geometric shape refinement, strong gradient...) propagate to the farfield boundaries. As a consequence, the grid generation of complexe configurations is more difficult when grid nodes clustering is needed in region where local gradients are assumed to be high. With structured grid, performing local mesh refinement is difficult and generally costly in nodes number.

To avoid all these disadvantages of the structured solver, the patched grid approach have been studied conjointly with Adaptive Mesh Refinement (AMR) technics. With the patched grid, blocks must have common interfaces but do not need the same location of grid nodes. The flexibility of this kind of mesh allows mesh refinement and makes easier to cluster grid points. AMR technics allow local mesh refinement with embedded subblocks. This technic have been introduced by Berger and Collela¹³ for two dimensionnal unsteady supersonic flow. Jouhaud¹⁵ have extended AMR technics for three dimensionnal steady transonic flow with using multigrid coupling strategies.

The conjointly use of these two meshing technics give new prospect to the structured solver in order to be an efficient industrial tool.

*PhD student, CERFACS, 42 av. Coriolis F-31057 Toulouse Cedex France, email: bohbot@cerfacs.fr, phone: +33 5 61 19 30 62, fax: +33 5 61 19 30 00

†Senior Researcher, CERFACS, 42 av. Coriolis F-31057

‡Senior Researcher, AIAA member, CERFACS, 42 av. Coriolis F-31057

In a first part, PG and AMR technics are briefly introduced. In a second part, validation test cases are performed to show the numerical efficiency of the meshing technics. In the last part, an industrial test case of a large civil aircraft configuration is computed using both PG and AMR technics.

1 Governing equations and flow solver

1.1 Euler equations

The governing equations are the unsteady Euler equations which describe the conservation of mass, momentum and energy of an inviscid flow field. Using cartesian coordinates (x, y, z) , these equations can be expressed in a conservative form as follows :

$$\frac{\partial W}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = 0. \quad (1)$$

The state vector W and the inviscid fluxes f , g and h are given by :

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho E + p) \end{pmatrix}, \quad (2)$$

$$g = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(\rho E + p) \end{pmatrix}, \quad h = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(\rho E + p) \end{pmatrix}, \quad (3)$$

where ρ is the density, u , v and w are the cartesian components of velocity, p is the pressure and E is the total energy. As the system (1) contains six unknowns (ρ , u , v , w , E , p) for only five equations, it is necessary to add another equation (a state equation). For a caloric perfect gas, this equation is given by :

$$p = \rho R T \quad (4)$$

where R is the gas constant equal to 287 (J/kgK) for air. The temperature T is then defined as a function of the conservative variables.

1.2 Flow solver

The NSMB (Navier-Stokes Multi-Block) code⁶ is used in this study. This code is jointly developed by three research establishments (EPFL in Lausanne, KTH in Stockholm and CERFACS in Toulouse) within an European research project supported by two industrial partners (Aérospatiale MATRA Airbus and SAAB). The NSMB code solves the compressible Navier-Stokes equations using a finite volume method with various spatial discretization schemes like Jameson's central difference scheme⁷, Roe's scheme¹⁶ or AUSM+scheme¹⁷. Time integration is based on the full matrix implicit method LU-SGS (Lower-Upper Symmetric Gauss-Seidel). NSMB has been parallelized using message passing communication MPI.

2 Patched grid algorithm

The use of domain decomposition techniques becomes widespread for complex configurations. This decomposition in multiblock structured meshes facilitates the distribution of the mesh points and reduces the memory required for the numerical solver. It also allows an efficient use of parallel computers. The structured numerical solver is less CPU time consuming thanks to the vectorization of the algorithm. Furthermore, structured meshes simplify the gradient calculation. These advantages allow the multi-block structured code to deal with very large industrial configurations.

However the structured grid requires common interfaces between blocks which imposes constraints on the grid generation. Consequently mesh refinement in regions where gradients are strong propagate to the farfield boundaries. To avoid this disadvantage of the structured grids, the patched grid approach has been studied. With a patched grid, blocks must have common interfaces but don't need the same location of grid nodes as figure 1 pictured it. The flexibility of this kind of mesh allows mesh refinement.

One of the most important properties of a patched grid algorithm is to maintain conservation of the numerical scheme. Benek, Steger and Dougherty² have illustrated the loss of accuracy when non-conservative interfaces are used. Their transonic bi-dimensional flow calculations of an airfoil with a smaller embedded grid around a flap, show a very distorted computed shock when the shock passes through the grid interface. This phenomenon is well

known, since the numerical computation of a discontinuous solution requires a numerical scheme in conservation form. Rai¹¹ has proposed a conservative patched grid algorithm for the Euler equations. His method ensures conservation for a patched grid having a common cell center line at the interface. The governing equations are integrated in each block in conjunction with a zonal boundary scheme which allows proper information transfer across grid interfaces. This method could be viewed as a particular case of the flux interpolation method of Berger³. Lerat and Wu⁹ have developed a patched grid algorithm which is conservative and unconditionally stable for dissipative difference schemes. The block interface treatment doesn't use flux interpolation as Rai's method. It consists in computing the numerical flux for each interface divided segment and summing them to get the total numerical flux for each cell face at the patched grid interfaces. Furthermore, this method is linearly equivalent to an area-weighted interpolation of the state vector.

The method which is described in this paper uses the splitting and dividing method of the numerical fluxes as described by Lerat and Wu⁹. The implementation is done in a way to guarantee conservation for all the parts of the numerical flux. For the implicit LU-SGS algorithm, fictive cells at interfaces are filled at each sweep with an area-weighted interpolation of the state vector. These interpolations permit to keep the robustness and the efficiency of the LU-SGS implicit algorithm.

2.1 Numerical method

Let us discuss in detail an interface condition for a two dimensional patched grid illustrated in figure 1. Block 1 and block 2 have a common boundary line of nodes. The indices (i, j) refer to the cell-center locations of block 1 and the indices (l, m) refer to the cell-center locations of block 2. We assume that the interface is located at index $i = 1/2$ for block 1 and at index $l = 1/2$ for block 2. To simplify the understanding of the algorithm detailed here, we assume that there is no overlap or gap between cell at the block interface. We also suppose that surface vectors of the cell's face located at the block interfaces are collinear. For the general patched grid case, we redefine cell volume and surface vector as been clearly explained in¹.

Defining the spatial numerical flux \hat{F} in the (ξ, η) generalized coordinates, the global conservation can be maintained by enforcing spatial flux conservation along the patched grid interface as ,

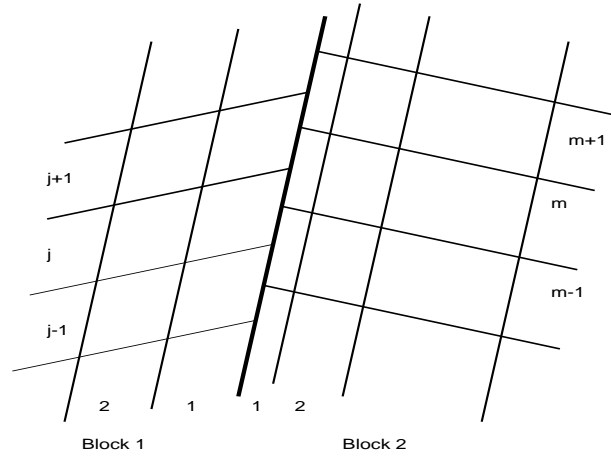


Fig. 1: Patched grid

$$\int \hat{F}^{[1]}(\xi = 1/2, \eta) d\eta = \int \hat{F}^{[2]}(\xi = 1/2, \eta) d\eta. \quad (5)$$

The equality 5 can be written in discretized form like :

$$\hat{F}_{\frac{1}{2},j}^{[1]} = \left(\alpha_1 \hat{F}_{\frac{1}{2},m-1}^{[2]} + \alpha_2 \hat{F}_{\frac{1}{2},m}^{[2]} \right), \quad (6)$$

with α_j the area weighted coefficient. We note by $A_{(1/2,j)}$ the surface defining the cell face $(1/2, j)$, and $A_{(1/2,m)}$, $A_{(1/2,m-1)}$ the surfaces defining the cell faces $(1/2, m)$ and $(1/2, m-1)$. The factor α_1 is the coefficient related to the intersection area between geometric surfaces $A_{(1/2,j)}$ and $A_{(1/2,m-1)}$ normalized with the area of $A_{(1/2,j)}$.

$$\alpha_1 = \frac{\int A_{(1/2,j)} \cap A_{(1/2,m-1)} dS}{\int A_{(1/2,j)} dS}.$$

Full details of the method and validation test cases can be found on reference⁵.

2.2 No coincident mesh generation

The main advantage of the patched grid possibility is that we can use the original block splitting for the refinement process. Unlike the AMR technique, the patched grid refinement do not need a first calculation. The mesh refinement is made during the mesh generation. For the aircraft configuration, we have chosen to refine the block of the wing and of the wake. With keeping the refinement on the wing, we describe with accuracy the vortex creation. The body and outflow domain are not refined. With this kind of refinement, the grid contains 55% fewer nodes than the fine grid.

3 Adaptive Mesh Refinement

Despite constant advances realized by both numerical schemes and computers, improving the accuracy and decreasing the computational costs are still two major objectives of the CFD. The use of global mesh refinement for structured grids requires unnecessary large number of nodes, which results in an increase of computational work and storage. Local grid refinement methods allow computational CPU savings for transonic steady flows using local multigrid type strategies. Following the pioneering work of Berger¹², Colella¹³ and Quirk¹⁴, an adaptive mesh refinement method (AMR) is presented, associated to NSMB solver. In a previous paper¹⁵, 2D steady flows have already been considered for the Euler equations. Here, the purpose is to apply the same method for a 3D complex computation. We hope to capture wake vortices with a high degree of precision.

3.1 Hierarchical Grid Structure

The AMR algorithm consists of a sequence of integration on different grid levels ($0 \leq l \leq lmax$). A grid is G_l required to be union of sub-blocks in which the same discretisation procedure is applied : $G_l = \cup_k G_{l,k}$ where $G_{l,k}$ are elementary sub-blocks. The hierarchical grid structure respects the ‘‘properly nested property’’ which is based on the three following rules :

- (1) $\forall l \leq l \leq lmax, G_l \subset G_{l-1}$, : inclusion of underlying grids ;
- (2) $G_{l,k} \cap G_{l,h} = \emptyset$ if $k \neq h$: no overlapping ;
- (3) adjacent cells of G_l must only belong to the level $l-1$, except for external or wall boundaries.

In the case of NSMB solver, for programming facilities, we don’t authorize overlapping at different grid levels :

- (4) a fine elementary mesh must be contained in only one coarser elementary mesh.

At last, the grid hierarchy is generated by a factor 2 sub-division of selected cells : a coarse cell becomes eight fine cells.

3.2 Grid Cycling

For steady flows, various options of cycles are possible as in multigrid strategies. For the present case, we apply two sequences : The first stage consists in integrating all the sub-blocks at the same time. This strategy of integration aims a good parallelized programming. The second stage is essential by making a coupling between grid levels : on each level ; a composite residual is constructed. By this way, the coarse grids are corrected.

3.3 Coarse grid correction - Composite residual

To ensure transfers of information from fine grids to coarse grids, we employ locally a forcing function, as in multigrid strategies⁷. This type of communication is essential for two reasons. The first one is to obtain a more accurate coarse grid solution for the interpolation of dummy cells at fine-coarse boundaries. The second reason must be connected with the adaption process. In fact, while fine grids capture fine structures, this information is required on the coarse grids in order to keep, and eventually to increase, the refinement there. On the refined level $lmax$, equations are discretized with an implicit phase which can be cast into the following compact formulation :

$$A_{lmax} \Delta U_{lmax}^{n+1} = -\frac{\Delta t}{|\Omega|} R(U_{lmax}^n) \quad (7)$$

where $\Delta U_{lmax}^{n+1} = U_{lmax}^{n+1} - U_{lmax}^n$ is the time solution increment and $R(U_{lmax}^n)$ is the residual. After the first stage, local flow solution U_{lmax} and residual $R(U_{lmax})$ are collected and recursively transferred down to coarser grid using a conservation preserving operator T :

$$\forall l \ 1 \leq l \leq lmax - 1, \quad (8)$$

$$A_l \Delta U_l^{n+1} = -\frac{\Delta t}{|\Omega|} R_l^{Comp}, \quad (9)$$

$$\Delta U_l^{n+1} = U_l^{n+1} - U_l^{Comp}, \quad (10)$$

where R_l^{Comp} and U_l^{Comp} are defined as follow :

$$R_l^{Comp} = \begin{cases} R_l(U_l^n) & \text{for non-refined cells} \\ T R_{l+1}^{comp} & \text{for refined cells} \end{cases}$$

$$U_l^{Comp} = \begin{cases} U_l^n & \text{for non-refined cells} \\ T U_{l+1}^{comp} & \text{for refined cells} \end{cases}$$

The main advantage of this coupling lies in its conservation preserving property and its independance in terms of the solver. In fact, the residual composite method is multi-solvers, what is a great advantage in a big CFD code such NSMB.

3.4 Automatic Grid Adaption/GAME process

The keystone of AMR is provides by its efficiency to automatically generate refined zones over regions of interest (discontinuities, boundary layers, large truncation errors, ...). In the NSMB context, each level of refinement is generated by GAME. GAME is

subdivision process based on several monitor functions called sensors. These sensors lead to thresholds which have to be tuned. Afterwards, the flagged cells are grouped into a patch of sub-grids using a grouping/clustering algorithm¹⁴. In the present method, a vortex- sensor is used to detect wake vortices. In the case of NSMB, the grid adaption process (GAME) is not directly coupled with the integration process.(NSMB code). After a first stage of integration (converged computation), one the most coarse grid, a primary level of refinement is built with the help of GAME. Then, a second computation is realised with the first level and so forth, the hierarchical grid structure is progressively generated. The user decides the number of refinement levels.

4 Test Cases

To assess the accuracy of solutions obtained using the PG/AMR algorithm presented in this paper, we have evaluated several test case.

4.1 NACA0012 - Inviscid transonic flow

The first test case involves the inviscid transonic flow past a NACA0012 airfoil, with free-stream Mach number $M_\infty = 0.84$ and an angle of attack $\alpha = 1^\circ$. This test case has been chosen for its sensitivity to the accuracy of the numerical treatment. In each case, we have used the LU-SGS implicit matrix method. We use the Jameson central scheme with the dissipation coefficients $k^{(2)} = 0.5$ and $k^{(4)} = 0.04$. For this test case, a rather strong shock occurs on the upper side and a weaker shock on the lower side of the airfoil. Two computations were made on two different meshes. The first one is a np-coincident grid which has 12 blocks and xxx nodes. The topology contains C blocks around the airfoil and H blocks in the other regions. The PG is obtained with un-enrichment of the H block and with some refinements in several blocks belonging to the C topology (cf. figure 2). The PG/AMR grid is obtained with the refinement of shock in the lower and upper side of the airfoil (cf. figure 3) In figure 4, the pressure coefficient distributions on the airfoil show that the PG/AMR treatment predicts a correct shock location.

4.2 RAE28022 - Turbulent transonic flow using the Spalart-Allmaras turbulence model

The objective of the NSMB flow solver is to be used as a design tool for aircraft in a transonic regime. The flight conditions are free-stream Mach number $M_\infty = 0.73$, an angle of attack $\alpha = 2.79^\circ$, and a

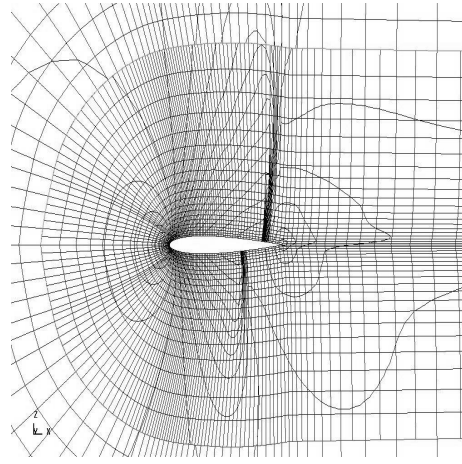


Fig. 2: Mach isolines (PG) on NACA0012 $M_\infty = 0.84, \alpha = 1^\circ$

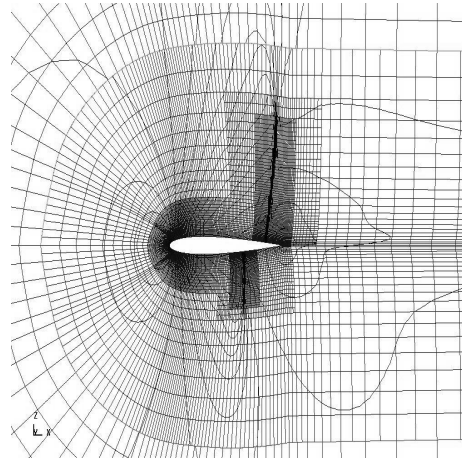


Fig. 3: Mach isolines (PG/AMR) on NACA0012 $M_\infty = 0.84, \alpha = 1^\circ$

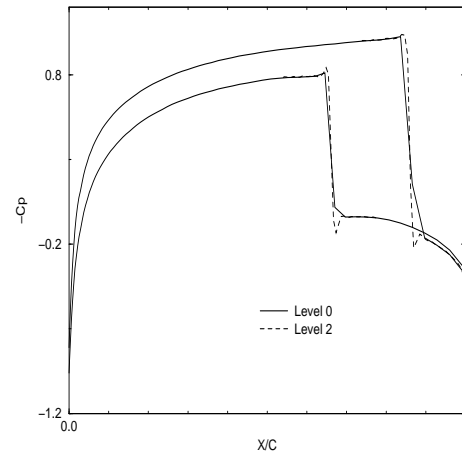


Fig. 4: $-C_p$ on NACA0012 $M_\infty = 0.84, \alpha = 1^\circ$

Reynolds number $Re = 6.510^6$. Navier-Stokes calculations are performed with the Spalart-Allmaras turbulence model and the LU-SGS implicit matrix version. Jameson's dissipation coefficients are fixed to $k^{(2)} = 0.5$ and $k^{(4)} = 0.015$. We use two different meshes. The first one is a no-coincident mesh and is obtained with un-enrichment of the block outside the C mesh (cf. figure 5). It contains xxx nodes. The second one is refined in region near the shock and contains xxx nodes. (cf. figure 6). For this flow configuration, three Gauss-Seidel sweeps are performed per iteration. Similar results are obtained with the two different meshes.

The numerical results of the pressure distribution are very close to the experimental data except at the foot of the shock (cf. figure 7). We can now expect to improve the accuracy of the calculation with some mesh refinement at the foot of the shocks with the PG/AMR functionalities.

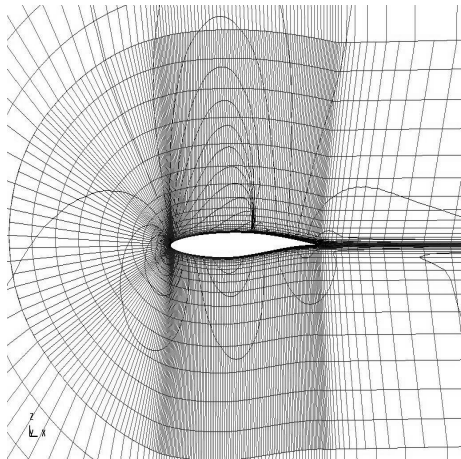


Fig. 5: Mach Isoline on RAE (PG) $M_\infty = 0.73, \alpha = 2.79^\circ$

4.3 AS28G CONFIGURATIONS

The AS28G aircraft configuration generated at EADS Airbus SA represents a realistic test for industrial calculations. This aircraft complete configuration (fuselage, wing, nacelle, pylon) is used to study the engine/airframe integration where viscous effect are important (cf. ⁸). Under certain flight conditions, flow separation can occur in this region which might lead to buffeting phenomena. To predict with accuracy this phenomenon, refinement in the pylon region is required. The AS28G computation presented here corresponds to cruise conditions with a free stream Mach number $M_\infty = 0.8$, an angle of attack $\alpha = 2.2^\circ$, and a Reynolds number

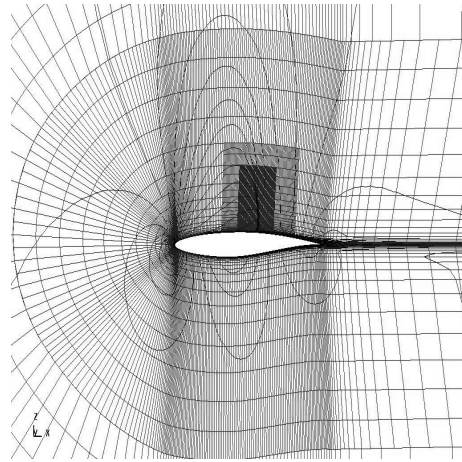


Fig. 6: Mach isoline on RAE (PG/AMR) $M_\infty = 0.73, \alpha = 2.79^\circ$

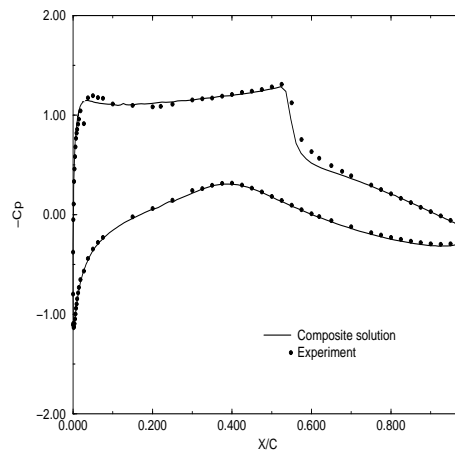


Fig. 7: $-C_p$ distribution on RAE $M_\infty = 0.73, \alpha = 2.79^\circ$

$Re = 11.1610^6$.

Results will be in the final paper.

5 Conclusion

A PG/AMR algorithm has been developed for a multi-block structured solver. We have shown the capabilities of this algorithm. It can be used to remove nodes in regions of the flow field where high resolution is not required or to refine zones where gradients are expected to be high. Furthermore, the conservative property of the non-coincident treatment allows accurate transonic flow simulations. PG/AMR block can intersect viscous layers and limited test cases have indicated no problem.

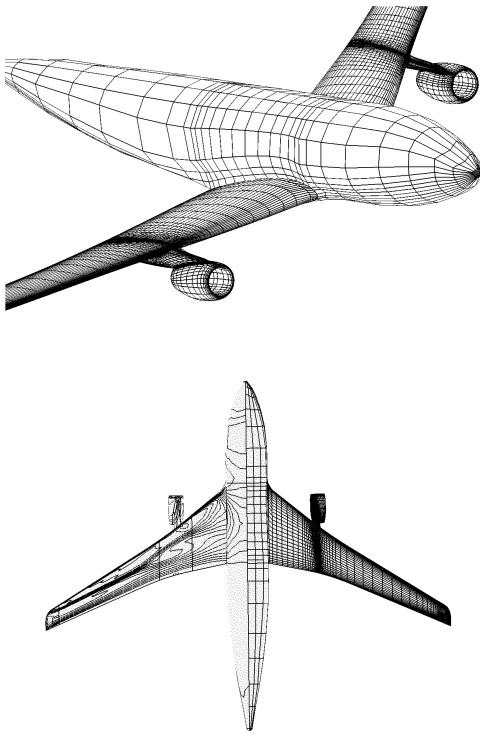


Fig. 8: AS28G patched grid and pressure isolines

References

- ¹K.S. Abdol-Hamid, J.R. Carlson and S.P. Pao, "Calculation of turbulent flows using mesh sequencing and conservative patch algorithm". *AIAA Paper*, **95-2336**, presented at the 31st Joint Propulsion Conference and Exhibit, San Diego, July (1995)
- ²J.A. Benek, J.L. Steger and F.C. Dougherty, "A flexible embedding technique with application to the Euler equations". *AIAA Paper*, **83-1944**, presented at the Computational Fluid Dynamic Conference, Denvers, MA, July, 1983.
- ³J.M. Berger, "On conservation at grid interfaces". *SIAM J. Numer. Anal.*, **24**, 967-984 (1987).
- ⁴R.T. Biedron and J.L. Thomas, "A generalized patched grid algorithm with application to the F-18 forebody with actuated control strake". *Computing System in Engineering*, **1**, 563-576 (1990).
- ⁵J. Bohbot, G. Grondin, D. Darracq and A. Corjon, "A Parallel Multigrid Conservative Patched/Sliding Mesh Algorithm for Turbulent Flow Computation of 3D Complex Aircraft Configurations". *AIAA Paper* **2001-1006**, (Reno 2001)
- ⁶J.B. Vos, A.W. Rizzi, A. Corjon, E. Chaput and E. Soinne, "Recent Advances in Aerodynamics inside the NSMB (Navier-Stokes Multiblock) Consortium". *AIAA Paper* **98-0225**, (1998).
- ⁷A. Jameson, "Analysis and design of numerical schemes for gas dynamics, 1 : artificial diffusion, upwind biased, limiters and their effect on accuracy and multigrid convergence", *Computational Fluids Dynamics*, **4**, 171-218, (1995)
- ⁸C. Gacherieu, "Etude d'un modèle de turbulence algébrique 3D: Application au calcul Navier-Stokes de l'écoulement autour d'une installation motrice d'avion de transport", PH.D. Thesis, CERFACS/IMFT, Toulouse FRANCE, (1996).
- ⁹A. Lerat and Z.N. Wu, "Stable conservative multi-domain treatments for implicit Euler solvers". *Journal of Computational Physics*, **123**, 45-64 (1996).
- ¹⁰A. Murta, "Exact Clipping of General 2D Polygons". *Submitted to the ACM Annual Symposium on Computational Geometry*, Hong Kong, June 2000.
- ¹¹M.M. Rai, "Stable conservative multi-domain treatments for implicit Euler solvers". *Journal of Computational Physics*, **62**, 472-503 (1986).
- ¹²M. J. Berger, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," Ph.D. thesis, Computer Science Dept., Stanford University. 1982.
- ¹³M. J. Berger and Colella, "Local Adaptive Mesh Refinement for Shock Hydrodynamics,". *J. Comput. Phys.* **82**, **67-84**, 1989.
- ¹⁴J. J. Quirk, "An Adaptive Grid Algorithm for Computational Shock Hydrodynamics,". *J. Comput. Phys.* **82**, **67-84**, 1989.
- ¹⁵J. Jouhaud, M. Borrel, "A Hierarchical Adaptive Mesh Refinement Method : Application to 2D Flows,". *Third ECCOMAS Computational Fluid Dynamics Conference*, **268-274**, 1996.
- ¹⁶P.L. Roe, "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes" *J. Comp. Phy.*, Vol. **43**, 357-372, (1981).
- ¹⁷M.-S. Liou and C.J. Stephen, "A New Flux Splitting Scheme". *J. Comp. Phy.*, Vol. **107**, 23-29, (1993).