A CONSERVATIVE PATCHED GRID ALGORITHM FOR TURBULENT FLOW COMPUTATIONS OF 3D COMPLEX AIRCRAFT CONFIGURATIONS

J. Bohbot*, D. Bertin†, and D. Darracq⋆

*European Centre for Research and Advanced Training in Scientific Computation (CERFACS), 42, Avenue Gaspard Coriolis. 31057 Toulouse Cedex 01, France. E-mail: bohbot@cerfacs.fr, darracq@cerfacs.fr, web page: http://www.cerfacs.fr/

†Aerospatiale Matra Airbus, BP M0142/3, 316 route de Bayonne, 31060 Toulouse Cedex 3, France. E-mail: denis.bertin@airbus.aeromatra.com, web page: http://www.aeromatra.com

Key words: compressible flow, complex aircraft configurations, 3D structured grid, conservative patched grid algorithm

Abstract. Body fitted structured grid generation is a crucial step in the computation of flow around complex aircraft configurations. The structured body fitted approach inevitably leads to dense meshes in zones where gradients are expected to be low. The challenge is to develop an efficient way to simplify the grid generation for complex configurations in transonic flow without wasting points. In this paper, a 3D patched grid algorithm is described using Jameson’s central scheme for the inviscid part of the flux and a 2nd order centered discretization for the viscous part of the flux. The patched grid algorithm is associated with the LU-SGS scheme for implicit time integration. The turbulence terms are estimated by the Badwin Lomax algebraic model or by the one equation Spalart Allmaras model. We show the efficiency of the patched grid algorithm with several referenced test cases. Finally, we present computations over the AS28G aircraft (fuselage, wing, pylon, nacelle) using the patched grid algorithm.
1 INTRODUCTION

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. Such a can be used as a tool for an industrial design platform. 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. It makes the grid generation of complex configurations more difficult when clustering grid nodes in regions where the gradients are expected to be high. 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 do not need the same location of grid nodes. The flexibility of this kind of mesh allows mesh refinement and makes it easier to cluster grid points. For transonic flows, the block interface treatment has to be conservative to ensure a correct prediction location of shocks passing through a block interface, and to ensure that no artificial shocks are generated at grid interfaces. The patched grid approach in conservative form is less consuming in memory and CPU time even for 3D complex configurations in comparison with overlapping grid treatment (chimera). For this reason, the patched grid is a good approach for an industrial computation platform. In the first part of this paper, a new conservative patched grid algorithm for the computation of a 3D aircraft configuration is presented with an implicit, centered Navier-Stokes solver. This algorithm is based on the 2nd order central Jameson scheme [7] for the computation of the inviscid part of the fluxes and a centered discretization for the computation of the viscous part of the fluxes. The turbulent models used are the Baldwin-Lomax model and the one-equation model of Spalart-Allmaras. We describe the patched grid algorithm implemented in the Multi-block Navier-Stokes Solver [6] (NSMB). At the patched interface between blocks the inviscid part of the fluxes is fully conservative using calculation of fluxes between a cell adjacent to several other cells. Computations on the NACA0012 and RAE02822 airfoils and the M6 wing in transonic flow show the capability of the algorithm to predict the correct shock location even if the patched interface is partially aligned with the shock. These test cases show that the convergence rate for patched grid is as good as for a grid without patched interface. After this Euler validation of the algorithm, we present the interface treatment of the patched grid for the turbulent Navier-Stokes case. First for the Baldwin-Lomax model, and then subsequently for the one equation model of Spalart-Allmaras. We show in this paper that the main difficulty is the estimation of the gradient through the patched grid interface. In this paper, we present how the patched grid interface affects viscous results with several elementary test cases.
In the second part, we present the industrial platform for grid generation used at AEROSPATIALE MATRA Airbus and CERFACS. We use in this platform, the DAMAS storage device [5] which permits the use of non coincident interfaces between blocks. The patched meshes are generated directly with ICEM HEXA or with a graphical tool called Quickview which allows the visualization of the mesh and node enrichment of blocks or node coarsening of blocks. The surface intersection of the adjacent cells of patched interfaces between blocks are computed with a clipping algorithm developed by Alan Murta [10] from the University of Manchester. This clipping algorithm is directly integrated in the NSMB code. The last part presents a 3D industrial aircraft configuration using patched grid interface. This one is a civil aircraft configuration AS28G with the body, wing, nacelle pylon in a transonic flow. A particularity of the mesh is refinement near the nacelle and the pylon. An implicit Navier-Stokes computation with the Baldwin-Lomax turbulence model is done. The numerical results of the configuration are compared with results without patched interfaces.

2 GOVERNING EQUATIONS

2.1 Navier-Stokes equations

The governing equations are the unsteady compressible Navier-Stokes equations which describe the conservation of mass, momentum and energy of the flow field. In conservative form, they can be expressed in three-dimensional cartesian coordinates \((x,y,z)\) as:

\[
\frac{\partial U}{\partial t} + \frac{\partial}{\partial x} (F - F_v) + \frac{\partial}{\partial y} (G - G_v) + \frac{\partial}{\partial z} (H - H_v) = 0. \tag{1}
\]

The Euler equations are obtained when the viscous fluxes \(F_v, G_v\) and \(H_v\) are set to zero. The state vector \(U\) and the inviscid fluxes \(F, G\) and \(H\) are given by:

\[
U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ u(\rho E + p) \end{pmatrix}, \quad G = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v w \\ v(\rho E + p) \end{pmatrix}, \quad H = \begin{pmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w^2 + p \\ w(\rho E + p) \end{pmatrix}, \tag{2}
\]

where \(\rho\) is the density, \(u, v\) and \(w\) are the cartesian velocity components, \(p\) is the pressure and \(E\) is the total energy. The viscous fluxes are defined as

\[
F_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u \tau_{xx} + v \tau_{xy} + w \tau_{xz} - q_x \end{pmatrix}, \quad G_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u \tau_{yx} + v \tau_{yy} + w \tau_{yz} - q_y \end{pmatrix},
\]
\[ H_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{zy} \\ \tau_{zz} \\ u \tau_{xx} + v \tau_{zy} + w \tau_{zz} - q_z \end{pmatrix}, \]

with \( \tilde{\tau} \) the shear stress tensor, and \( \tilde{q} \) is Fourier’s thermal conductivity.

### 2.2 Turbulence modeling

Two turbulence models are involved for resolving the boundary layer. The first one is the Baldwin-Lomax algebraic model and the second one is the Spalart-Allmaras one equation model.

- **The Baldwin-Lomax Model**
  The classical Baldwin-Lomax model is an algebraic turbulence model which is computationally low cost consuming and numerically robust. It is based on the empirical mixing length assumption and predicts well equilibrium attached flows. However, its underlying physical assumption limits the complexity of flows which can be adequately predicted. Furthermore, it requires the calculation of certain quantities along wall normals which becomes awkward when unstructured grids or multi-block meshes are used.

- **The Spalart-Allmaras model**
  The Spalart-Allmaras one-equation model is a transport equation model for the eddy viscosity. Although the Spalart-Allmaras model is not designed to reproduce large separation, it showed good results on a large variety of flows [12]. It naturally takes into account history effects and diffusion and is considered to be “local” because it does not need to search for any quantities along wall-normals. This makes it an attractive candidate for complex multi-block computations. Furthermore, it does not require finer grid resolution than the velocity field.

### 3 NUMERICAL METHODS

The NSMB code [6] is used in this study. This code has been developed within an international research project between three research establishments (EPFL, in Lausanne, KTH in Stockholm and CERFACS in Toulouse) and two industrial partners (AEROSPATIALE MATRA Airbus and SAAB). This solver is used in the aircraft design process of AEROSPATIALE MATRA Airbus.

The NSMB code solves the compressible Navier-Stokes equations using a finite volume method with various spatial discretization schemes like Jameson’s central difference scheme, Roe’s scheme, AUSM+ scheme or HLLE. The NSMB code is very efficient thanks
to the implicit LU-SGS (Lower-Upper Symmetric Gauss Seidel) scheme for time integration in scalar and matrix form implemented by Weber [12]. A multigrid algorithm is available to ameliorate the convergence rate. The NSMB code has been parallelized using message passing communication MPI.

In conservative form, the Navier-Stokes equation can be expressed in generalized coordinate as:

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial (\hat{F} - \hat{F}_\nu)}{\partial \xi} + \frac{\partial (\hat{G} - \hat{G}_\nu)}{\partial \eta} + \frac{\partial (\hat{H} - \hat{H}_\nu)}{\partial \zeta} = 0. \quad (3)$$

At the interface between finite volume cells, the numerical fluxes can be expressed as a combination of an inviscid and a viscous part as,

$$\hat{F}_{i+1/2} = \left(\hat{F}_e - \hat{F}_\nu\right)_{i+1/2}. \quad (4)$$

We use the central Jameson scheme for the discretization of the inviscid part of fluxes. This scheme has shown its capability to simulate complex transonic flow and is employed in many industrial code solvers.

### 3.1 Jameson’s central scheme

For inviscid flows the basic central derivative scheme does not contain any dissipative term and is intrinsically unstable. In order to prevent oscillations near shock waves or stagnation points, artificial dissipation terms are added to the governing discrete equations. The introduction of appropriate dissipation in the vicinity of shock waves allows an entropy condition to be satisfied.

For a steady state problem, we use the central scheme in conjunction with the artificial dissipation model of Jameson & al. [7]. The artificial dissipation model of Jameson is a combination of second and fourth order differences. The second-difference term allows shock capturing without oscillations, while the linear fourth-difference term provides the important background dissipation.

The interface flux function is given by (omitting indices for the other two directions):

$$\hat{F}_{i+1/2} = \hat{F} \left(\frac{U_{i+1} + U_i}{2}, s_{i+1/2}\right) - d_{i+1/2}, \quad (5)$$

with $\vec{s}$ the surface vector and the dissipation flux term

$$d_{i+1/2} = \epsilon_{i+1/2}^{(2)} (U_{i+1} - U_i) - \epsilon_{i+1/2}^{(4)} (U_{i+2} - 3U_{i+1} + 3U_i - U_{i-1}). \quad (6)$$

The coefficients $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are used to locally adapt the dissipative flux and are directionally scaled by the factor $r_{i+1/2}$.
\[
\epsilon^{(2)}_{i+1/2} = k^{(2)} r_{i+1/2} \max(\psi_{i+1}, \psi_i), \\
\epsilon^{(4)}_{i+1/2} = \max (0,0, k^{(4)} r_{i+1/2} - \epsilon^{(2)}_{i+1/2}).
\]

The scaling factor \(r_{i+1/2}\) is calculated as the average of the spectral radii at the cell face

\[
r_{i+1/2} = \frac{1}{2} \left( \lambda(A)_{i+1/2}^I + \lambda(A)_{i+1}^I \right),
\]

where \(\lambda(A)_{i}^I\) is the spectral radius of the Jacobian matrix \(A = \partial(F \cdot s)/\partial U\) evaluated at the cell center using the mean surface vector \(1/2 (s_{i+1/2} - s_{i-1/2})\) in the \(I\)-direction. The pressure sensor \(\psi\) controls the second-order dissipation near shock waves. It is constructed using the absolute value of the normalized second-order differences of the pressure:

\[
\psi_i = \left| \frac{p_{i+1} - 2p_i + p_{i-1}}{p_{i+1} + 2p_i + p_{i-1}} \right|.
\]

Typical values for the constant \(k^{(2)}\) are between 0.5 and 1.0, the values for \(k^{(4)}\) are typically between 0.01 and 0.03. Furthermore, when performing Navier-Stokes simulations, the artificial dissipation is usually damped in the boundary layer by multiplying the wall normal artificial dissipation fluxes with a function of the local Mach number which tends to zero at the wall and to one at the edge of the boundary layer [8].

### 3.2 Numerical viscous fluxes

The viscous terms are discretized using a second-order central discretization using the finite volume method. The viscous numerical flux is calculated similar to its inviscid counterpart as

\[
(F_v)_{i+1/2} = (F_v)_{i+1/2} \cdot s_{i+1/2},
\]

where the viscous flux tensor is given by \(F_v = (f_v, g_v, h_v)\). The velocity and temperature gradients in the viscous fluxes are calculated at the cell surfaces. These gradients are defined by the average over a shifted control volume \(P\) straddling a cell surface.

### 4 PATCHED GRID ALGORITHM

In this section, we now describe the patched grid algorithm which is implemented in NSMB. One of the most important properties of a patched grid algorithm for transonic flow is to maintain conservation of the numerical scheme. Benek, Steger and Dougherty [2] 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. Raï [11] 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 Berger [3].

Lerat and Wu [9] 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 Raï’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 Wu [9]. We have chosen to extend this method to Jameson’s centered scheme for the Euler equations. The same method is used for the numerical viscous fluxes and for the Spalart-Allmaras eddy viscosity fluxes. 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.

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 in [1].

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

\[
\int \hat{F}^{[1]}(\xi = 1/2, \eta)d\eta = \int \hat{F}^{[2]}(\xi = 1/2, \eta)d\eta. \tag{11}
\]

The equality 11 can be written in discretized form like:

\[
\hat{F}^{[1]}_{1/2,j} = \left( \alpha_1 \hat{F}^{[2]}_{1/2,m-1} + \alpha_2 \hat{F}^{[2]}_{1/2,m} \right), \tag{12}
\]

with \(\alpha_j\) 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 $\alpha_1$ 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 $A_{(1/2,j)}$.

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

To enforce eq. 12, we split the numerical fluxes in an inviscid part and a viscous part. With this splitting of the numerical fluxes, we have two equalities like eq. 12 to check. In a first part, we will discuss the inviscid conservative patched grid treatment. In a second part, the viscous fluxes calculation is described with the turbulent modeling.

### 4.1 Inviscid flux treatment

Let us detail Jameson’s numerical flux at the patched grid interface illustrated in figure 1. Using 5, the incoming numerical fluxes at the interface of block 2 is given by:

$$\begin{align*}
\hat{F}^{[2]}_{1/2,m+1} &= \hat{F}\left(\frac{U^{[2]}_{1,1,m+1} + U^{int}_{1,1,m+1} + s_{1/2,m+1}}{2}, s_{1/2,m+1}\right) - d^{[2]}_{1/2,m+1}, \\
\hat{F}^{[2]}_{1/2,m} &= \hat{F}\left(\frac{U^{[2]}_{1,1,m} + U^{int}_{1,1,m}}{2}, s_{1/2,m}\right) - d^{[2]}_{1/2,m}.
\end{align*}$$ (13)

The numerical flux of block 1 at the patched grid interface can be expressed like:

$$\begin{align*}
\hat{F}^{[1]}_{1/2,j} &= \hat{F}\left(\frac{U^{[1]}_{1,j} + U^{int}_{1,j}}{2}, s_{1/2,j}\right) - d^{[1]}_{1/2,j},
\end{align*}$$ (14)

with $U_{int}$ the state vector at the cell face. To ensure conservation of the scheme, we have to satisfy eq. 12 between the numerical fluxes at the cell faces of the non-coincident interface. It means to consider independently Jameson’s dissipative term and the central derivative term.
If we assume that surface vector of the cell faces \((1/2, j)\) of block 1 and the cell faces 
\((1/2, m)\) and \((1/2, m + 1)\) are collinear, we take the value of the state vector at the cell face to ensure the first equality of equations 15 as:

\[
U_{m}^{\text{int}} = U_{m+1}^{\text{int}} = U_{1,j}^{[1]}.
\]  
(16)

For Jameson’s dissipative term, we split the flux of the cell face \((1/2, j)\) in two elementary fluxes for the cell faces \((1/2, m)\) and \((1/2, m + 1)\) like:

\[
\begin{align*}
\hat{F} \left( \frac{U_{[1]}^{[1]} + U_{[1]}^{\text{int}} + U_{[2]}^{[1]} + U_{[2]}^{\text{int}}}{2}, s_{\frac{1}{2}, j} \right) &= \alpha_1 \hat{F} \left( \frac{U_{[1]}^{[1]} + U_{[1]}^{\text{int}} + U_{[2]}^{[1]} + U_{[2]}^{\text{int}}}{2}, s_{\frac{1}{2}, m+1} \right) + \alpha_2 \hat{F} \left( \frac{U_{[1]}^{[1]} + U_{[1]}^{\text{int}} + U_{[2]}^{[1]} + U_{[2]}^{\text{int}}}{2}, s_{\frac{1}{2}, m} \right), \\
d_{[2]}^{[1]} &= \alpha_1 d_{[2]}^{[2]} + \alpha_2 d_{[2]}^{[2]} + m + 1.
\end{align*}
\]  
(15)

If we assume that surface vector of the cell faces \((1/2, j)\) of block 1 and the cell faces 
\((1/2, m)\) and \((1/2, m + 1)\) are collinear, we take the value of the state vector at the cell face to ensure the first equality of equations 15 as:

\[
U_{m}^{\text{int}} = U_{m+1}^{\text{int}} = U_{1,j}^{[1]}.
\]  
(16)

For Jameson’s dissipative term, we split the flux of the cell face \((1/2, j)\) in two elementary fluxes for the cell faces \((1/2, m)\) and \((1/2, m + 1)\) like:

\[
\begin{align*}
d_{\frac{2}{2}, m+1}^{[2]} &= \epsilon_{\frac{2}{2}, m+1}^{[2]} (U_{1,m+1} - U_{1,j}) - \epsilon_{\frac{2}{2}, m+1}^{[4]} (U_{2,m+1} - 3U_{1,m+1} + 3U_{1,j} - U_{2,j}), \\
d_{\frac{2}{2}, m}^{[2]} &= \epsilon_{\frac{2}{2}, m}^{[2]} (U_{1,m} - U_{1,j}) - \epsilon_{\frac{2}{2}, m}^{[4]} (U_{2,m} - 3U_{1,m} + 3U_{1,j} - U_{2,j}),
\end{align*}
\]  
(17)

with the scaling factor in \(\frac{1}{2}, m, \frac{1}{2}, m + 1\) evaluated at the cell center using the mean surface vector \(\frac{1}{2}(s_{\frac{1}{2}, m} - s_{\frac{1}{2}, j})\) and \(\frac{1}{2}(s_{\frac{1}{2}, m+1} - s_{\frac{1}{2}, j})\). The pressure sensor is evaluated at the cell face using the normalized second order difference of the pressure (for example at cell face \((1/2,m)\) such as:

\[
\psi_{1,m} = \left| \frac{p_{2,m} - 2p_{1,m} + p_{1,j}}{p_{2,m} + 2p_{1,m} + p_{1,j}} \right|,  \\
\psi_{0,m} = \left| \frac{p_{1,m} - 2p_{1,j} + p_{2,j}}{p_{1,m} + 2p_{1,j} + p_{2,j}} \right|.
\]  
(18)

The sensor is then taken in \((1/2,m)\) as:

\[
\nu_{1/2,m} = \max(\mu_{0,m}, \mu_{1,m}).
\]  
(19)

Furthermore, when performing Navier-Stokes simulations, the artificial viscosity is usually damped in the boundary layer. We have to treat the damping function at patched grid interfaces to maintain the conservativity. The damping function must have the same value at each side of the interfaces.

With this conservative treatment of the inviscid part of the numerical fluxes, we can compute transonic flow. However, it is well known that refinement induces oscillations. Jameson’s dissipative term of the central scheme can reduce these oscillations and stabilize the numerical scheme even with a high volume ratio between adjacent cells. This means that, we have obtained a robust and efficient algorithm to compute transonic flow with no-coincident grids.
4.2 Viscous flux treatment

The treatment of the numerical viscous fluxes through a non-coincident interface is less developed in literature than the inviscid numerical fluxes. Biedron and Thomas [4] have developed a generalized non-conservative algorithm for 3D applications using a MUSCL interpolation method for the conservative variables in the ghost cell. They have shown neglective effect non-conservative algorithm in a viscous layer. However, their algorithm is limited to flows without shock waves.

We have chosen to treat the numerical viscous fluxes in a pseudo-conservative way. We split the numerical viscous fluxes like the numerical Euler fluxes in elementary fluxes to enforce conservation. However, shear stress tensors are evaluated with an interpolation of the state vector in the ghost cell. This approximation decreases the CPU time consumption of the non coincident interface.

We consider the numerical treatment of the interface illustrated in figure 1.

Like the inviscid fluxes, we match the flux of block 1 with two elementary fluxes of block 2:

\[
\mathbf{\hat{F}}_{v,[1]}^{[1]} = \left( \alpha_1 \mathbf{\hat{F}}_{v,[1]}^{[2]} + \alpha_2 \mathbf{\hat{F}}_{v,[2]}^{[2]} \right),
\]

(21)

We consider in first the momentum equation and then the treatment of the energy equation.

- Momentum equations

The numerical viscous flux can be expressed at the cell face \((1, j)\) of block 1 as:

\[
\mathbf{\hat{F}}_{v,[1]}^{[1]} = \mathbf{\tau}_{1//2,j}, \mathbf{s}_{1//2,j},
\]

(22)

and at the cell face \((1, m)\) and \((1, m + 1)\) like:

\[
\mathbf{\hat{F}}_{v,[1]}^{[2]} = \mathbf{\tau}_{1/2,m}, \mathbf{s}_{1/2,m},
\]

\[
\mathbf{\hat{F}}_{v,[2]}^{[2]} = \mathbf{\tau}_{1/2,m+1}, \mathbf{s}_{1/2,m+1}.
\]

(23)

If we assume that the surface vectors are collinear, equation 21 can be expressed like:

\[
\mathbf{\tau}_{1//2,j} = \alpha_1 \mathbf{\tau}_{1/2,m} + \alpha_2 \mathbf{\tau}_{1/2,m+1}
\]

(24)

with \(\mathbf{\tau}\) the shear stress tensor.

To ensure conservation of the scheme, we have:

\[
\mathbf{\tau}_{1//2,j} = \alpha_1 \mathbf{\tau}_{1//2,j} + \alpha_2 \mathbf{\tau}_{1//2,j+1}
\]

(25)
with
\[
\begin{align*}
\tau_{1,j}^m &= \tau_{1,j}^m, \\
\tau_{1,j}^{m+1} &= \tau_{1,j}^{m+1}.
\end{align*}
\] (26)

In a general way, shear stress tensor entries can be expressed like,
\[
\tau_{1,j} = \frac{1}{2}(\mu_{1,j}) \left( \bar{a} \cdot (\nabla u)_{1/2,j} + \bar{b} \cdot (\nabla v)_{1/2,j} + \bar{c} \cdot (\nabla w)_{1/2,j} \right).
\] (27)

with \( \mu \) the dynamic viscosity and \( \bar{a}, \bar{b}, \bar{c} \) constant vectors. At the patched grid interface, for shear stress tensor \( \tau_{1,j}^m \) evaluated at the cell face \((1/2, j)\), the second equation 26 is met if:
\[
\begin{align*}
\frac{\mu_{1,j}}{\mu_{1,m}} &= \frac{\mu_{1,0}^m}{\mu_{1,m}^m}, \\
\nabla u_{1/2,j} &= \nabla u_{1/2,m}, \quad \nabla v_{1/2,j} = \nabla v_{1/2,m}, \quad \nabla w_{1/2,j} = \nabla w_{1/2,m}.
\end{align*}
\] (28)

To ensure the first equation of system 28, we take at the cell face \((1/2,j)\):
\[
\frac{\mu_{1,j}}{\mu_{1,m}} = \frac{\mu_{1,0}^m}{\mu_{1,m}^m} = \frac{1}{2} (\mu_{1,j} + \mu_{0,m}).
\] (29)

For the other equations of system 28, we have to evaluated the gradient at the cell face of the patched grid interface. Enforcing this relation is expensive in CPU time. The gradient at the interface is computed using a weighted average of the state vector in the ghost cell:
\[
U_{0,j} = \alpha_1 U_{1,m} + \alpha_2 U_{1,m+1}.
\] (30)

The geometry used for the ghost cell \((0,j)\) is the geometry of the interior cell \((1,j)\). This gradient evaluation is accurate and reduces the conservation error of the numerical scheme.

- **Energy equation**

The numerical flux can be expressed in a general form at the cell face \((1/2,j)\) as,
\[
\tilde{F}^{[1]}_{1/2,j} = (\tilde{\tau}_{1/2,j} \tilde{u}_{1/2,j}) + k.\nabla T_{1/2,j}.
\] (31)

Like the momentum equation, we write the numerical flux at the patched interface illustrated in figure 1. We obtain the system:
\[
\begin{align*}
\tau_{1,0}^m \tilde{v}_{1/2,j} + k.\nabla T_{1/2,j} &= \alpha_1 (\tau_{1,0}^m \tilde{v}_{1/2,j} + k.\nabla T_{1/2,j}) + \alpha_2 (\tau_{1,0}^{m+1} \tilde{v}_{1/2,j}^{m+1} + k.\nabla T_{1/2,j}^{m+1}) \\
\tau_{1,0}^m \tilde{v}_{1/2,j}^{m+1} &= \tau_{1,0}^{m+1} \tilde{v}_{1/2,j}^{m+1} \\
k.\nabla T_{1/2,j} &= k.\nabla T_{1/2,j} \\
k.\nabla T_{1/2,j}^{m+1} &= k.\nabla T_{1/2,j}^{m+1}
\end{align*}
\] (32)
We compute the velocity and the temperature gradient with a weighted average at the cell face. We assume that the conductivity coefficient $k$ is constant.

### 4.3 Turbulent term treatment

In this section, we present the patched grid interface treatment for the turbulent term. As the viscous term, lack of conservation is less crucial than for the inviscid part of the numerical flux. Furthermore, Spalart-Allmaras uses a non conservative first order upwind discretization of the convective parts. However, the Spalart-Allmaras turbulence model contains a transport equation for the eddy viscosity. This transport equation has to be treated correctly to allow a correct calculation of the eddy viscosity numerical fluxes at a non-coincident interface.

- Baldwin-Lomax model
  The Baldwin-Lomax model uses two different formulations to evaluate the eddy viscosity in an inner region and the outer region. For the inner region the Prandtl mixing length theory is used. This theory needs the calculation of the magnitude of vorticity in the ghost cell. This magnitude of vorticity is calculated at a patched grid interface using a weighted average of velocity.

- Spalart-Allmaras model
  The Spalart-Allmaras model is a one equation transport of the eddy viscosity. The numerical turbulent fluxes can be split in a convective part and a diffusive part. If we consider the patched interface illustrated in figure 1, we split the turbulent numerical flux at the cell face $(1/2, j)$ of block 1 in two elementary fluxes calculated at cell face $(1/2, m)$ and $(1/2, m + 1)$ of block 2. The eddy viscosity gradients are computed using a weighted average in the ghost cell. This treatment improves the convergence rate.

### 5 MESH GENERATION

The first set of tools used to create/check/modify a structured multi-block is ICEM DDN, COMAK, MULCAD, PADDAM and HEXA at AEROSPATIALE MATRA Airbus and at CERFACS. ICEM MULCAD mesh generator allows to create patched grid interfaces with node locations which can be different on both sides of a block interface. Moreover, ICEM HEXA just allows the refinement of blocks in each topological direction. The patched grid algorithm can be used as a partial connectivity block interface treatment. We can also create patched grid interfaces with graphical tools developed at AEROSPATIALE MATRA Airbus named QUICKVIEW / LE TO.
We store the topological and geometrical data in a DAMAS database [5]. The DAMAS storage device is commonly used to store grid data and data relative to CFD results. It includes the non-coincident interface description necessary for the NSMB calculation.
To avoid serious effects of the lack of conservation for Navier-Stokes calculations, we have to generate patched grid interfaces without big gaps or overlap. With ICEM HEXA generate patched interface which have no overlap even near wall boundary. For this reason, ICEM HEXA is a good way to generate patched grid interfaces for transonic flow simulations. This tools allows the refinement on the wall boundary without loss of geometric accuracy.

If the interfaces have different grid points distributions on either side of the interface, then gaps or overlap may occur between the cells and the requirements for conservative patched grid algorithm is to compute new cell volumes and the new cell surface vectors which are necessary to our algorithm (cf. [1]). To compute weighted area intersection between cells, we use a clipping algorithm developed by A.Murta [10] from computers graphics.

6 TEST CASES

To assess the accuracy of solutions obtained using the patched grid algorithm presented in this paper, we have evaluated several test cases which contain the main kind of patched grid configurations. For all these test cases, we have made comparisons between the solution obtained with a non-coincident grid and with a similar coincident grid. These test cases show the conservative quality of the algorithm implement in NSMB.

- NACA0012 - Inviscid transonic flow
  The first test case involves 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. We have made two computations on two different meshes. The first one is a coincident grid which has 12 blocks and 14300 nodes. The topology contains C blocks around the airfoil and H blocks in the other regions. The patched grid is obtained with un-enrichment of the H block and with some refinements in several blocks belonging to the C topology. A block is partially refined in the supersonic region, and is refined in two directions in the shock region (cf. figure 2). A patched grid interface is aligned with the strong shock on the upper side of the airfoil. This patched grid contains 10500 nodes. In figure 3, the pressure coefficient distributions on the airfoil show that the patched grid interface treatment predicts a correct shock location even with a patched interface aligned with the strong shock. This calculation assesses the conservative implementation of the patched grid algorithm for inviscid flow. Figure 2 shows the Mach isolines on the patched grid and figure 3 the convergence histories for the coincident grid and the
patched grid. The convergence histories are quite similar for the two calculations and assesses the stability characteristic of the patched grid algorithm.

![Figure 2: NACA0012 airfoil patched grid, Mach isolines $M_\infty = 0.85, \alpha = 1^\circ$](image)

![Figure 3: -$C_p$ on NACA0012 $M_\infty = 0.73, \alpha = 1^\circ$, Convergence histories](image)

- 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 Reynolds number $Re = 6.5 \times 10^6$. 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 coincident and contains 90774 nodes. The patched one contains 77500 nodes and is obtained with un-enrichment of the block outside the C mesh and in the wake block (cf.
figure 4). For this flow configuration, three Gauss-Seidel sweeps are performed per iteration.

We obtain similar results with the two different meshes. The convergence histories are similar for the two different calculations which gives a clear indication of the good stability of the patched grid algorithm. Figure 4 shows the Mach isolines of the patched grid calculation. Pressure distributions and skin friction coefficients are compared between the two calculations in figure 5.

The numerical results of the pressure distribution are very close to the experimental data except at the foot of the shock. We can now expect to improve the accuracy of the calculation with some mesh refinement at the foot of the shocks with the patched grid capability. This refinement would allow the NSMB code to predict the lambda shock which intersect the boundary layer, and consequently the good pressure distribution.

Figure 4: RAE28022 airfoil patched grid - Mach isolines $M_\infty = 0.73, \alpha = 2.79^\circ$

Figure 5: $-C_p$ pressure coefficient distributions and skin friction coefficients for the RAE28022
M6 - Turbulent transonic flow using Spalart-Allmaras turbulence modeling

This test case involves a 3-dimensional transonic flow over a ONERA-M6 wing. The flight conditions are free-stream Mach number $M_\infty = 0.8395$, an angle of attack $\alpha = 3.06^\circ$, and a Reynolds number $Re = 11.72 \times 10^6$. Navier-Stokes calculations are performed with the Spalart-Allmaras turbulence model and the LU-SGS implicit matrix scheme. We perform 3 sweeps for the implicit LU-SGS algorithm. Jameson’s dissipation coefficients are fixed to $k^{(2)} = 0.5$ and $k^{(4)} = 0.015$. The artificial dissipation is damped near the wall. We use two different meshes. The first one is coincident and contains $6.10^5$ nodes. The patched one contains $3.10^5$ nodes and is obtained with unenrichment of the block outside the C mesh and in the wake block (cf. figure 6). We remove nodes in the span direction and in the chord direction. The pressure distributions are compared with wind tunnel results in figures 7 and 8. Figure 8 shows the convergence histories. The patched grid has a similar convergence as for the coincident grid calculation. The shock location is well predicted for all these computations as figures 7, 8 confirm. Theses figures show that the strong shock on the upper side has more dissipation on the patched grid calculation than the coincident grid calculation. This phenomenon is produced by the block coarsening in the supersonic region outside the C blocks which generates more artificial dissipation. This artificial dissipation propagates in the C block and spreads the strong shock at the upper side. We can diminish this artificial dissipation for the patched grid calculation, by decreasing the Jameson’s dissipation coefficients without loss of convergence rate.

Figure 6: M6 wing with patched grid. On the surface, pressure isolines are shown
7 AS28G CONFIGURATIONS

The AS28G aircraft configurations generated at AEROSPATIALE MATRA Airbus represents a realistic test for industrial calculation. 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 $Re = 11.16 \times 10^6$. Two different meshes are considered. The first one is coincident and contains 550,000 nodes. The second one is a patched grid with 395,000 nodes. The patched grid is obtained with removing 1 on 2 nodes in all the directions in the block which are not interesting for our calculation. We have kept all the nodes in the wing pylon and nacelle blocks. The
nodes near the fuselage and at the far field are removed. We have kept nodes in the
wake of the wing, nacelle and pylon. The LU-SGS scalar method is applied to this
computation. Jameson’s dissipation coefficients value used are $k^{(2)} = 0.5$ and $k^{(4)} = 0.02$.
The wall normal artificial dissipation is damped in the boundary layer. We use for this
computation the Baldwin-Lomax turbulence model. The convergence histories can be
seen in figure 12. The convergence evolutions are similar for the two different meshes and
corroborate the stability property of the patched grid algorithm even for 3-dimensional
calculations. Figure 9 show the isolines of pressure and the mesh on the patched grid.
The pressure distributions at four locations in the span-wise direction on the wing are
displayed in figures 10, 11 and compared with wind tunnel results. The skin friction lines
on the interior side of the pylon are displayed in figure 12.

Figure 9: AS28G patched grid and pressure isolines

Figure 10: $-C_p$ stations 1 (17% of half-span), $-C_p$ stations 2 (29% of half-span)- $M_\infty = 0.8, \alpha = 2.2^\circ$
8 CONCLUDING REMARKS

A conservative patched grid algorithm has been developed for a multi-block structured solver based on Jameson's central scheme. 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. Patched interfaces can intersect viscous layers and limited test cases have indicated no problem. However, it is clear that non-coincident interfaces induce numerical perturbations which may create numerical dissipation. The patching algorithm has been applied to the AS28G complex aircraft configurations. The turbulent flow solutions show qualitative agreement with experiment data and with the numerical solution on a similar coincident grid. The next goal is to improve the non-coincident treatment in turbulent boundary layers.
References


