INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:855–873 (DOI: 10.1002/fld.378)

Patched grid and adaptive mesh refinement strategies for the calculation of the transport of vortices

A. Benkenida*,†,‡, J. Bohbot§ and J. C. Jouhaud‡

European Center for Research and Advanced Training in Scientific Computation (CERFACS) 42, av Gaspard, Coriolis, 31057 Toulouse Cedex, France

SUMMARY

This paper presents two techniques allowing local grid refinement to calculate the transport of vortices. The first one is the patched grid (PG) method which allows non-coincident interfaces between blocks. Treatment of the non-coincident interfaces is given in detail. The second one is the adaptive mesh refinement (AMR) method which has been developed in order to create embedded sub-grids. The efficiency of these two methods is demonstrated by some validating tests. Then the PG and AMR strategies are applied in the computation of the transport of vortices. We start with a simple vortex flow in a cubic box. Then, the flowfield around a complex aircraft configuration is calculated using the two refinement techniques. Results are compared with a fine, referenced grid calculation. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: wake vortex; structured grid; adaptive mesh refinement; patched grid

1. INTRODUCTION

The rate of decay of airplane wake vortices represents one of the most important parameters in the determination of the separation distance imposed between two successive aircrafts in take-off or landing configuration. Indeed, it is important to ensure there is sufficient distance between aircraft, so to guarantee the following one will be able to traverse the wake safety. Currently, all separation distances are based on aircraft weight and vary from 3 to 6 nautical miles. These rules introduce significant constraints that have already lead to some airports becoming saturated. The evolution of air traffic shows clearly that this situation will be generalized to many other airports in the near future.

* Correspondence to: A. Benkenida, European Center for Research and Advanced Training in Scientific Computation (CERFACS) 42, av. Gaspard Coriolis, 31057 Toulouse Cedex, France.
† E-mail: Adlene.Benkenida@ifp.fr
‡ Senior Researcher.
§ PhD Student.

Contract/grant sponsor: Publishing Arts Research Council; contract/grant number: 98-1846389

Received August 2001
Revised March 2002

Copyright © 2002 John Wiley & Sons, Ltd.

The study of wake vortices allows us to understand their behaviour and consequently explore the means by which to reduce their hazards. This alternative will be beneficial since it will allow us to establish new rules, which are less constraining, for the separation distances between airplanes. By adopting these rules, airport's capacity can be increased.

The analysis of wake vortices by solving the whole flow around the airplane is the most accurate numerical strategy. However, it needs huge grids, especially when we want to follow vortices many wingspans behind the aircraft. In order to reduce the size of the grids (and consequently the cost of computations), some numerical strategies have been developed. In this paper we will present and test two of these strategies: the patched grid (PG) and the adaptive mesh refinement (AMR) techniques. Since the present study is our first work in which we try to couple the physics of wake vortices and meshing strategies, we choose to restrict our attention to inviscid flow. This hypothesis is well justified since we consider only near field wake. Indeed, previous studies [1] have shown that a non-viscous approach gives satisfactory results in the near field wake: the physics of the roll-up phenomenon is dominated by inviscid effects. Turbulent effects start to influence the dynamics of the wake after several wingspans downstream of the plane.

We start this paper by giving some indications about the governing equations and the flow solver. Then we present independently the theory of PG and AMR strategies. In order to validate the two techniques, we perform some numerical tests in the third section. In these tests we show the effect of meshing strategies on different configurations. After that, we apply PG and AMR to the study of wake vortices, which represent the physical topic of this paper. In this section, we use two different configurations. The first one is devoted to the transport of a vortex in a cubic box. This preliminary test is very important, since it allows us to quantify the performance of the PG and AMR strategies (comparison with analytical solution). The second configuration corresponds to a generic aircraft model. We try to follow wake vortices up to 6.5 wingspans downstream of the wing trailing edge. We perform three computations. The first one is based on the use of a fine grid and is considered as a reference. In the second (resp. third) we use the PG (resp. AMR) strategy. A detailed comparison is performed between the three computations. Of course, we end the present study by some concluding remarks, in which we show clearly the great influence of meshing strategies on accurate and efficient wake vortices computations.

2. GOVERNING EQUATIONS AND FLOW SOLVER

2.1. 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 co-ordinates (x, y, z), these equations can be expressed in a conservative form as follows:

$$\frac{\partial W}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = 0 \tag{1}$$

Copyright © 2002 John Wiley & Sons, Ltd.

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

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

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

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

$$p = \rho RT \tag{3}$$

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

2.2. Flow solver

The NSMB (Navier–Stokes multi-block) code is used in this study. This code has been jointly developed by EADS France S.A., CERFACS. EPFL, KTH and Saab [2]. 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 [3], Roe's scheme [4], AUSM+ scheme [5] or HLLE [6]. In order to conserve high gradients in vortices during advection downstream, it is necessary to use a fourth-order central space discretization [7]. Time integration is based on the full matrix implicit method LU-SGS (lower-upper symmetric Gauss–Seidel) [8]. The NSMB has been parallelized using message passing communication MP1.

3. MESHING STRATEGIES

3.1. Patched grid algorithm

The use of domain decomposition techniques has become widespread for complex configurations. The decomposition into multiblock structured meshes facilitates the attainment of the

Copyright © 2002 John Wiley & Sons, Ltd.

desired 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 calculation of gradients. These advantages allow the multi-block structured code to deal with very large industrial configurations. As such, it 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 for 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 PG approach has been studied. With the PG approach, 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. In conservative form, the Navier–Stokes equation can be expressed in generalized co-ordinates (ψ, ξ, ζ) as

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial(\hat{F} - \hat{F}_v)}{\partial \psi} + \frac{\partial(\hat{G} - \hat{G}_v)}{\partial \xi} + \frac{\partial(\hat{H} - \hat{H}_v)}{\partial \zeta} = 0 \tag{4}$$

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

$$\hat{F}_{i+1/2} = (\hat{F}_e - \hat{F}_v)_{i+1/2} \tag{5}$$

One of the most important properties of a PG algorithm for transonic flow is to maintain conservation of the numerical scheme. Benek *et al.* [9] 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 [10] has proposed a conservative PG algorithm for the Euler equations. His method ensures conservation for a PG having a common cell centre 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 [11].

Lerat and Wu [12] have developed a PG algorithm which is conservative and unconditionally stable'for dissipative difference schemes. The block interface treatment does not use flux interpolation in contrast to Raï's method. It consists of computing the numerical flux for each interface divided segment and summing them to get the total numerical flux for each cell face at the PG 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 [12]. We have chosen to extend this method to Jameson's centred scheme and to Ducros's centred scheme for the Euler equations. For the implicit LU-SGS algorithm, fictitious cells at interfaces are filled at each sweep with an area-weighted interpolation of the state vector. These interpolations permit the robustness and the efficiency of the LU-SGS implicit algorithm to be kept.

Copyright © 2002 John Wiley & Sons, Ltd.

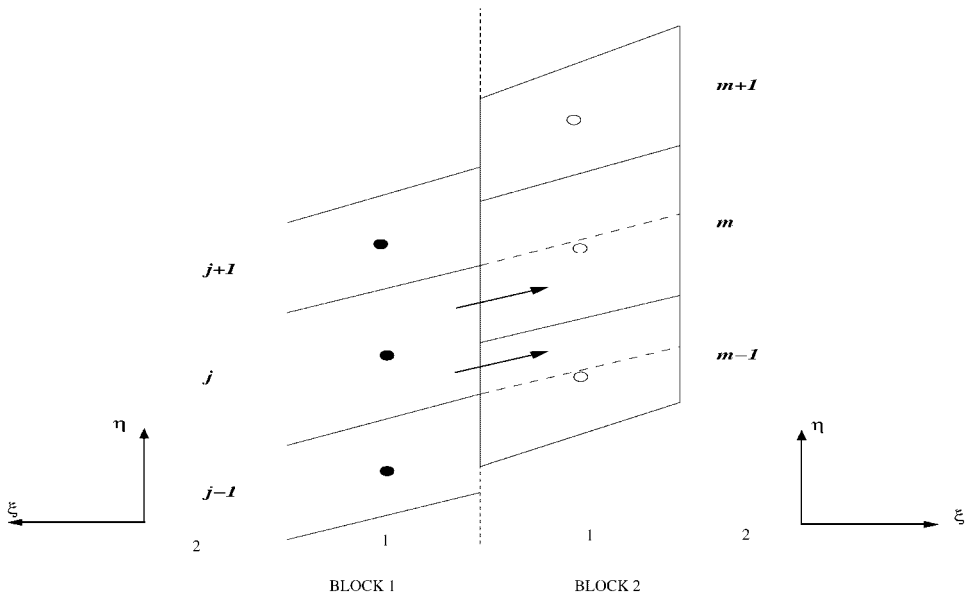

Figure 1. Patched grid.

Let us discuss in detail an interface condition for a two-dimensional PG illustrated in Figure 1. Blocks 1 and 2 have a common boundary line of nodes. The indices (i,j) refer to the cell-centre locations of block 1, and the indices (l,m) refer to the cell-centre locations of block 2. We assume that the interface is located at index $i=\frac{1}{2}$ for block 1, and at index $l=\frac{1}{2}$ for block 2. To simplify the understanding of the algorithm detailed here, we assume that there is no overlap or gap between cells 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 PG case, we redefine cell volume and surface vector as has been clearly explained in Reference [13].

Defining the spatial numerical flux $\hat{F}$ in the (ξ,η) generalized co-ordinates, the global conservation can be maintained by enforcing spatial flux conservation along the PG interfaces as

$$\int \hat{F}^{[1]}(\xi=1/2,\eta)\,\mathrm{d}\eta = \int \hat{F}^{[2]}(\xi=1/2,\eta)\,\mathrm{d}\eta \tag{6}$$

The equality (6) can be written in discretized form like

$$\hat{F}^{[1]}_{1/2,j} = (\alpha_1 \hat{F}^{[2]}_{1/2,m-1} + \alpha_2 \hat{F}^{[2]}_{1/2,m}) \tag{7}$$

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

Copyright © 2002 John Wiley & Sons, Ltd.

surfaces $A_{(1/2,j)}$ and $A_{(1/2,m-1)}$ normalized with the area of $A_{(1/2,j)}$.

$$\alpha_1 = \frac{\int A_{(1/2,j)} \bigcap A_{(1/2,m-1)} \, \mathrm{d}S}{\int A_{(1/2,j)} \, \mathrm{d}S}$$

Let us discuss the inviscid treatment for the second-order scheme of Jameson and the fourth-order scheme of Ducros.

3.1.1. Inviscid flux treatment. In this section, the treatment of the block interfaces without coincident points is described for the Jameson second-order centred scheme and the Ducros fourth-order scheme. For these two schemes, the artificial dissipation fluxes are treated similarly. Just the derivative terms need a particular treatment for the two different schemes. Then we just describe the treatment of the diffusive term in the Jameson's central scheme part.

Second-order Jameson's central scheme: Let us detail Jameson's numerical flux at the PG interface illustrated in Figure 1. The incoming numerical fluxes at the interface of block 2 is given by

$$\begin{aligned} \hat{F}^{[2]}_{1/2,m-1} &= \hat{F}\left(\frac{U^{[2]}_{1,m-1} + U^{\mathrm{int}}_{m-1}}{2}, s_{1/2,m-1}\right) - d^{[2]}_{1/2,m-1} \\ \hat{F}^{[2]}_{1/2,m} &= \hat{F}\left(\frac{U^{[2]}_{1,m} + U^{\mathrm{int}}_{m}}{2}, s_{1/2,m}\right) - d^{[2]}_{1/2,m} \end{aligned} \tag{8}$$

The numerical flux of block 1 at the PG interface can be expressed like

$$\hat{F}^{[1]}_{1/2,j} = \hat{F}\left(\frac{U^{[1]}_{1,j} + U^{\mathrm{int}}_{j}}{2}, s_{1/2,j}\right) - d^{[1]}_{1/2,j} \tag{9}$$

with U^{int} the state vector at the cell face. To ensure the conservation of the scheme, we have to satisfy Equation (7) 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

$$\begin{aligned} \hat{F}\left(\frac{U^{[1]}_{1,j} + U^{\mathrm{int}}_{j}}{2}, s_{1/2,j}\right) &= \alpha_1 \hat{F}\left(\frac{U^{[2]}_{1,m-1} + U^{\mathrm{int}}_{m-1}}{2}, s_{1/2,m-1}\right) \\ &\quad + \alpha_2 \hat{F}\left(\frac{U^{[2]}_{1,m} + U^{\mathrm{int}}_{m}}{2}, s_{1/2,m}\right) \\ d^{[1]}_{1/2,j} &= \alpha_1 d^{[2]}_{1/2,m} + \alpha_2 d^{[2]}_{1/2,m-1} \end{aligned} \tag{10}$$

If we assume that the surface vectors at the cell face $(\frac{1}{2}, j)$ of block 1 and at cell faces $(\frac{1}{2}, m)$ and $(\frac{1}{2}, m-1)$ are collinear, we can take the value of the state vector at the cell face to ensure

Copyright © 2002 John Wiley & Sons, Ltd.

the first equality of Equations (10) as

$$U_m^{\text{int}} = U_{m-1}^{\text{int}} = U_{1,j}^{[1]} \tag{11}$$

For Jameson's dissipative term, the flux at the cell face $(\frac{1}{2}, j)$ can be split into two elementary fluxes for cell faces $(\frac{1}{2}, m)$ and $(\frac{1}{2}, m-1)$ as

$$\begin{aligned} d_{1/2,m-1}^{[2]} &= \varepsilon_{1/2,m-1}^{(2)}(U_{1,m-1} - U_{1,j}) \\ &\quad - \varepsilon_{1/2,m-1}^{(4)}(U_{2,m-1} - 3U_{1,m-1} + 3U_{1,j} - U_{2,j}) \\ d_{1/2,m}^{[2]} &= \varepsilon_{1/2,m}^{(2)}(U_{1,m} - U_{1,j}) \\ &\quad - \varepsilon_{1/2,m}^{(4)}(U_{2,m} - 3U_{1,m} + 3U_{1,j} - U_{2,j}) \end{aligned} \tag{12}$$

with the scaling factor in $(\frac{1}{2}, m)$ and $(\frac{1}{2}, m-1)$ evaluated at the cell centre using the mean surface vector $\frac{1}{2}(\vec{s}_{1/2,m} - \vec{s}_{3/2,j})$ and $\frac{1}{2}(\vec{s}_{1/2,m-1} - \vec{s}_{3/2,j})$, respectively. The pressure sensor is evaluated at the cell face using the normalized second-order difference of the pressure (for example at cell face $(\frac{1}{2}, m)$ such as

$$\psi_{1,m} = \left| \frac{p_{2,m} - 2p_{1,m} + p_{1,j}}{p_{2,m} + 2p_{1,m} + p_{1,j}} \right| \tag{13}$$

$$\psi_{0,m} = \left| \frac{p_{1,m} - 2p_{1,j} + p_{2,j}}{p_{1,m} + 2p_{1,j} + p_{2,j}} \right| \tag{14}$$

The sensor is then taken in $(\frac{1}{2}, m)$ as

$$\nu_{1/2,m} = \max(\psi_{0,m}, \psi_{1,m}) \tag{15}$$

Furthermore, when performing Navier–Stokes simulations, the artificial viscosity is usually damped in the boundary layer. We have to treat the damping function at PG 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 non-coincident grids.

Fourth-order Ducros's scheme: Improvements due to this fourth-order scheme have been confirmed for the estimation of wake vortices [7]. This scheme is widely used to compute wake vortices and is used in this paper for the vortex studies. The PG algorithm extends to this high-order central scheme without any difficulties.

Copyright © 2002 John Wiley & Sons, Ltd.

The skew symmetric form of the Ducros's scheme can be expressed at the cell's interfaces $(\frac{1}{2}, j)$, as

$$\hat{F}^{[1]}_{1/2,j} = \hat{F}(U_j^{\text{int}'}, U_j^{\text{int}}, U_{1,j}, U_{2,j}) \tag{16}$$

To ensure conservation of the numerical fluxes, we split the fluxes in two elementary fluxes associated with the adjacent cells

$$\hat{F}^{[1]}_{1/2,j} = (\alpha_1 \hat{F}^{[2]}_{1/2,m-1} + \alpha_2 \hat{F}^{[2]}_{1/2,m}) \tag{17}$$

with α_i the area-weight coefficient. These elementary fluxes are then expressed in this form

$$\begin{aligned} \hat{F}^{[2]}_{1/2,m} &= \hat{F}(U_{-1,j}, U_{0,j}, U_{1,m}, U_{2,m}) \\ \hat{F}^{[2]}_{1/2,m-1} &= \hat{F}(U_{-1,j}, U_{0,j}, U_{1,m-1}, U_{2,m-1}) \end{aligned} \tag{18}$$

This treatment keeps the conservation of the scheme even at non-coincident interfaces.

3.2. *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 an 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 Berger [14], Colella [15] and Quirk [16], an AMR method is presented, associated to the NSMB solver. In a previous paper [17], 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 vorticies with a high degree of precision.

3.2.1. Hierarchical grid structure. The AMR algorithm consists of a sequence of integration on different grid levels ($0 \leqslant l \leqslant l\max$). A grid G_l is required to be a union of sub-blocks in which the same discretization 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) $\forall l\, 1 \leqslant l \leqslant l\max$, $G_l \subset G_{l-1}$: 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 1-1, except for external or wall boundaries. In the case of the NSMB solver, for programming facilities, we do not authorize overlapping at different grid levels; (4) a fine elementary mesh must be contained in only one coarser elementary mesh. Finally, the grid hierarchy is generated by a factor 2 sub-division of selected cells: a coarse cell becomes eight fine cells.

3.2.2. 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 is aimed at achieving good parallel performance. The second stage is essential for making a coupling between grid levels: on each level; a composite residual is constructed. By this way, the coarse grids are corrected.

Copyright © 2002 John Wiley & Sons, Ltd.

3.2.3. 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 strategies [8]. 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 is 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 $l\max$, equations are discretized with an implicit phase which can be cast into the following compact formulation:

$$A_{l\max}\Delta U_{l\max}^{n+1} = -\frac{\Delta t}{|\Omega|} R(U_{l\max}^{n}) \tag{19}$$

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

$$\forall l\ 1 \leqslant l \leqslant l\max - 1 \tag{20}$$

$$A_l\,\Delta U_l^{n+1} = -\frac{\Delta t}{|\Omega|} R_l^{\text{comp}} \tag{21}$$

$$\Delta U_l^{n+1} = U_l^{n+1} - U_l^{\text{comp}} \tag{22}$$

where R_l^{comp} and U_l^{comp} are defined as follows:

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

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

The main advantage of this coupling lies in its conservation preserving property and its independance in terms of the solver, which is a great advantage in a big CFD code such as NSMB.

3.2.4. Automatic grid adaption/GAME process. The keystone of AMR is provided by its efficiency to generate automatically refined zones over regions of interest (discontinuities, boundary layers, large truncation errors, ..., etc). In the NSMB context, each level of refinement is generated by GAME. GAME is a 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 algorithm [16]. In the present method, a vortex-sensor is used to detect wake vorticies. 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), on the most coarse grid, a primary level of refinement is built with the help of GAME. Then, a second computation is realized with the first level and so forth, the hierarchical grid structure is progressively generated. The user decides the number of refinement levels.

Copyright © 2002 John Wiley & Sons, Ltd.

4. NUMERICAL VALIDATION

4.1. Patched grid technique validation

To assess the accuracy of solutions calculated when using the PG algorithm presented in this paper, we have evaluated a transonic test case. Comparisons between the solution obtained with a non-coincident grid and with a similar coincident grid were made.

NACA0012—Inviscid transonic flow: This test case involves inviscid transonic flow past an NACA0012 airfoil, with free-stream. Mach number $M_\infty = 0.85$ and an angle of attack $\alpha = 1°$. This test case has been chosen for its sensitivity to the accuracy of the numerical treatment. In each case, we use 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 PG 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 PG interface is aligned with the strong shock on the upper side of the airfoil. This PG contains 10 500 nodes. In Figure 3, the pressure coefficient distributions on the airfoil show that the PG 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 PG algorithm for inviscid flow. Figure 2 shows the Mach isolines on the PG and Figure 3 the convergence histories for the coincident grid and the PG. The convergence histories are quite similar for the two calculations and demonstrates the stability characteristics of the PG algorithm.

4.2. Adaptive mesh refinement technique validation

The AS28G wing configuration generated at EADS Airbus SA represents a 3D test-case. This inviscid flow corresponds to cruise conditions with a free stream Mach number $M_\infty = 0.8$, an

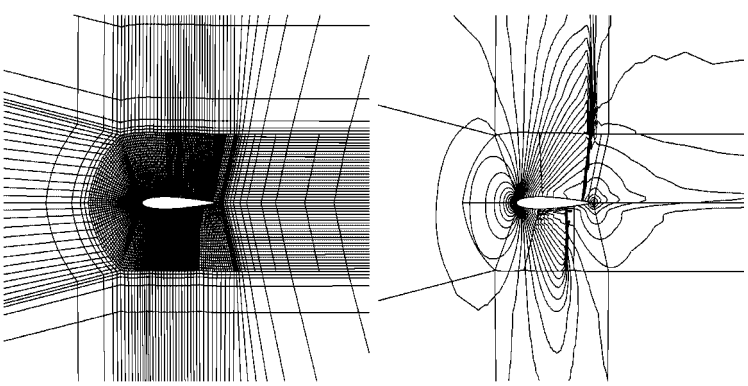

Figure 2. NACA0012 airfoil patched grid, Mach isolines $M_\infty = 0.85$, $\alpha = 1°$.

Copyright © 2002 John Wiley & Sons, Ltd.

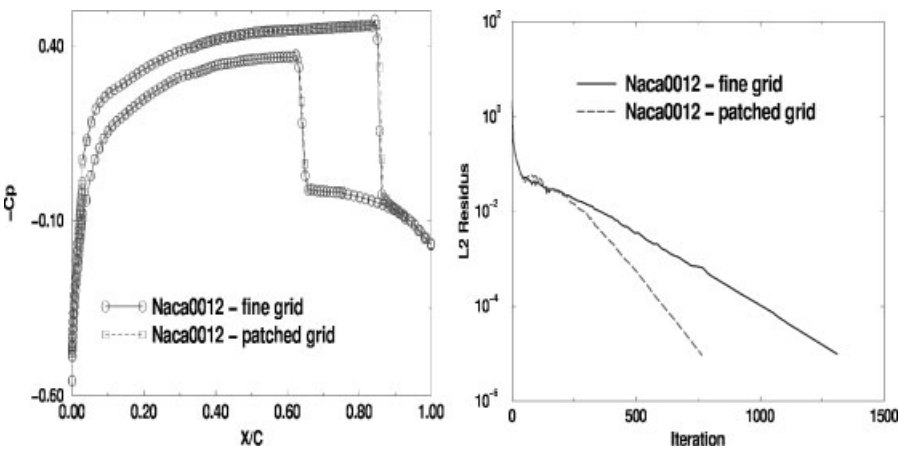

Figure 3. Cp on NACA0012 $M_\infty = 0.73$, $\alpha = 1°$—Convergence histories.

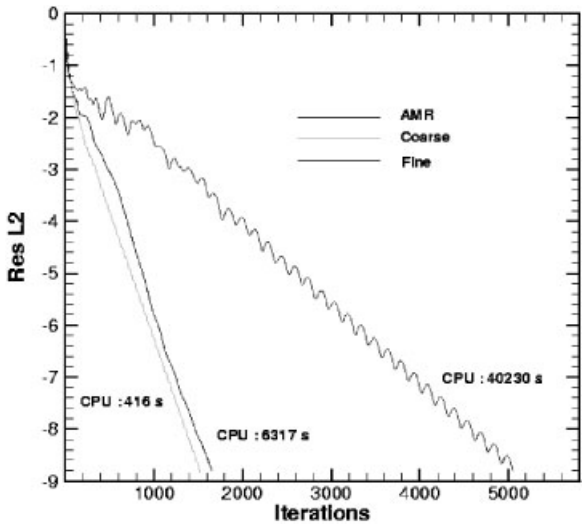

Figure 4. Convergence histories.

angle of attack $\alpha = 2.2°$. Two different meshes are considered for this computation. The first one is a globally refined mesh who contains 1 626 690 nodes. The second one is a composite mesh, containing 502 084 nodes, with two levels of twice refined grids. The most fine grid (level 2) has the same density as the globally refined mesh. In fact, the initial coarse grid is obtained by removing four nodes in all the directions from the first mesh.

The LU-SGS scalar method is applied to this computation. Jameson's dissipation coefficients are set at $k^{(2)} = 0.5$ and $k^{(4)} = 0.02$. From the convergence histories (Figure 4), we can see that AMR leads to significant improvements both in terms of number of time steps to reach

Copyright © 2002 John Wiley & Sons, Ltd.

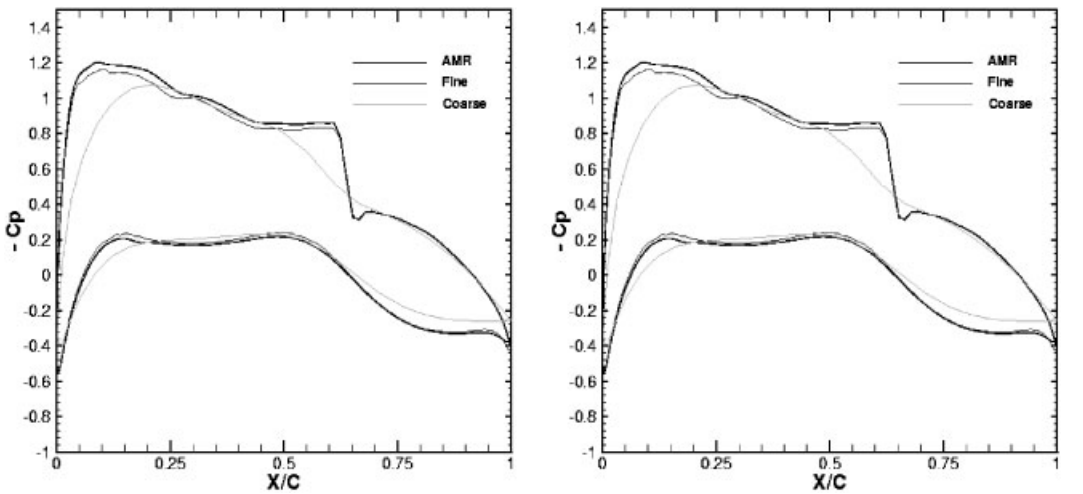

Figure 5. C_p stations ($y = 5$ and 13)—$M_\infty = 0.8$, $\alpha = 2.2.^\circ$

a convergence level (3 times less), and in terms of CPU time to carry out the computation on a VPP700 (6 times less). Moreover, the number of nodes is divided by a factor 3.

In Plates 1–3, the Mach number contours of the composite solution are plotted on the wing's skin. The shock continues to be captured better as the grid is refined.

The pressure coefficient distributions plotted in Figure 5 present another comparison between AMR and globally refined results. It shows some small discrepancies due to the very strong influence of the fine-coarse boundary conditions. Nevertheless, the shock locations are the same for the two cases. In conclusion, we have nearly the same precision.

5. APPLICATION TO VORTEX CALCULATION

5.1. Vortex transport

In order to evaluate the performance of the two meshing strategies (PG and AMR) in the computation of wake vortices, we start with a simple vortex flow. We impose a particular boundary condition at the inlet of a cubic box. This condition is characterized by a Lamb–Oseen vortex, defined by the following equation:

$$V(r,t) = 1.40\, V_{\mathrm{m}}(r/\mathrm{rc})^{-1}[1 - \mathrm{e}^{-1.25(r/\mathrm{rc})^2}] \tag{23}$$

where V_m is the maximum value of the tangential velocity and rc is the core radius, r indicates the radial position from the centre of the vortex. Using the NSMB code, we try to transport the vortex across to the outlet boundary by imposing a constant inlet axial velocity.

Three computations are performed. In the first one, we use a 'fine' grid. This computation is considered as the reference of comparison. The second (resp. the third) simulation is based on the use of PG (resp. AMR) strategy.

Figure 6 shows the inlet boundary plane of the different grids. We observe that all the grids have the same refinement level in the central zone, which is the region where the vortex is

Copyright © 2002 John Wiley & Sons, Ltd.

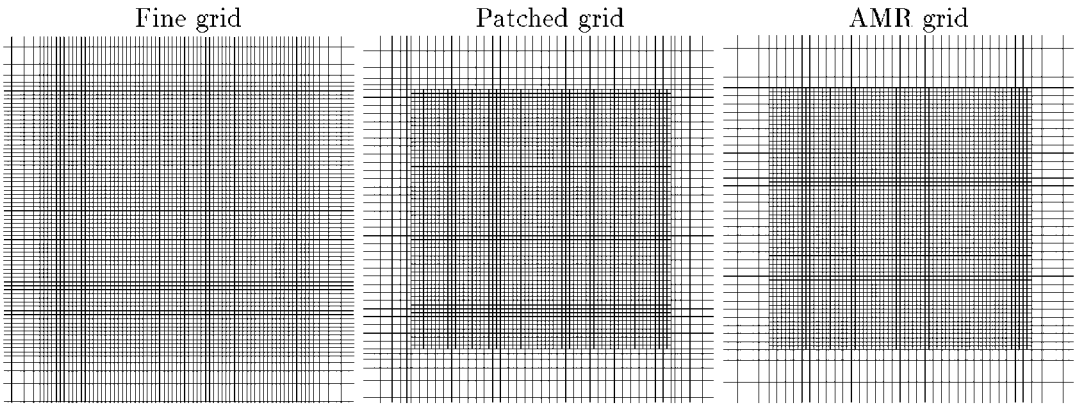

Figure 6. Comparison of inlet planes.

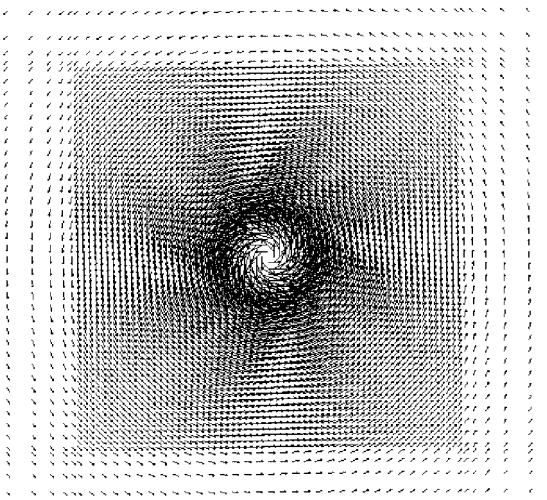

Figure 7. Velocity field in the inlet plane.

Table 1. Mesh size for each computation.

	Mesh size (points)
Fine	2.5×10^6
PG	1.5×10^6
AMR	1.4×10^6

localized (see Figure 7). Outside this region both the patched and AMR grids are coarser than the reference grid. The level of refinement shown in Figure 6 is maintained up to the outlet plane, in order to capture the vortex well along the entire length of the box. Table I shows the size of grids.

Copyright © 2002 John Wiley & Sons, Ltd.

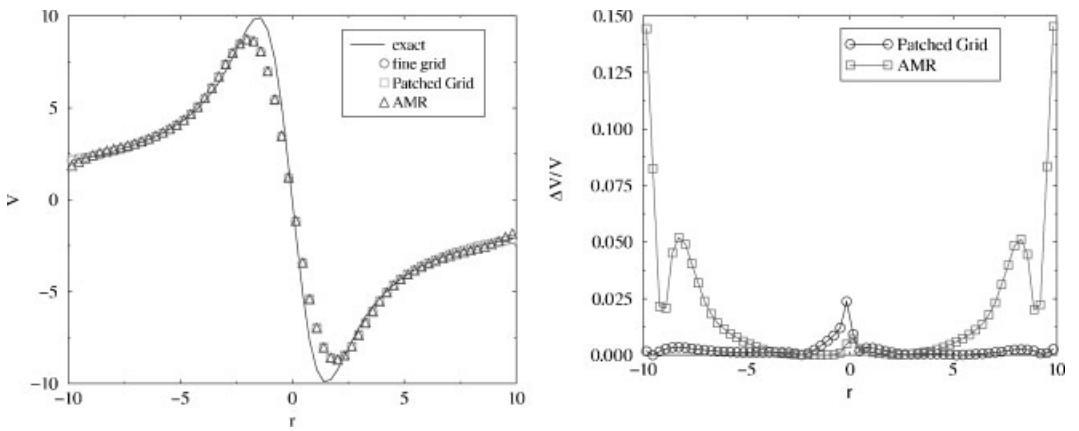

Figure 8. Comparison of velocity profiles—relative error of the velocity in comparison with uniform fine grid computation.

Figure 8 allows us to compare velocity profiles plotted at the outlet plane with the theoretical velocity profile (inlet flow) represented by a solid line. In this figure, it is clear that the numerical dissipation related to the used space discretization scheme, leads to a decrease of the maximum value of the tangential velocity. However, the conservative property of NSMB code causes the increase of the core radius, so we have the same total circulation in any plane which is perpendicular to the vortex axis. This problem is well known in the computation of compressible flows. In the present study, our goal is not to propose a solution for this question, but to test meshing strategies in the computation of vortex flows. Figure 8 shows that both PG and AMR lead to a satisfactory result: the velocity profile is well transported, if we compare it with that obtained using the fine grid. This example shows clearly that it is not necessary to have a fine grid everywhere if we want to study a vortex flow. It is sufficient to refine only the zone traversed by the core of the vortex in order to compute the flow well.

In order to have a more accurate idea about the accuracy of the results, relative error curves are plotted. Figure 8 is obtained by taking the fine grid result as reference to compute relative errors. As we can see, the difference between PG and fine grid does not go above 2.5%. The situation is slightly different for the AMR. In the core region, AMR leads to a very good results (the relative error is about 1.25% at maximum). However, out of the core region the difference is significant.

5.2. *Aircraft model*

As was specified in the introduction, we will try to use the techniques, which described in Section 3, in the computation of an aircraft wake vortices. The airplane model used in this study is presented in Figure 9. This model is essentially characterized by wings (NACA-4412) at zero angle of incidence and $\frac{2}{3}$ span flaps (NACA-0012, flap setting 20°). Note that the fuselage is extended until the end of the computational domain to avoid any generation of secondary vortices. The length of the domain downstream of the trailing edge, *Lw*, is 6.5 wingspans. It is important to point out that this airplane is a generic aircraft model, which was used in the European project C-Wake.

Copyright © 2002 John Wiley & Sons, Ltd.

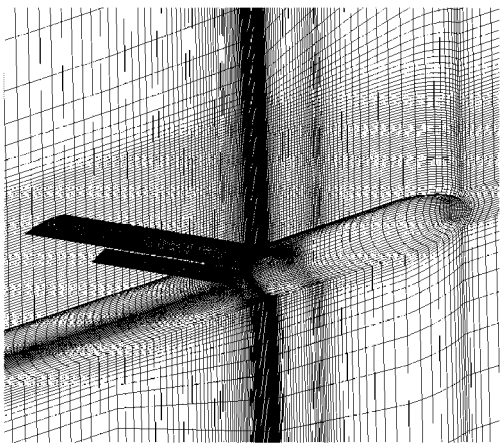

Figure 9. A general view of the grid near the surface of the airplane model.

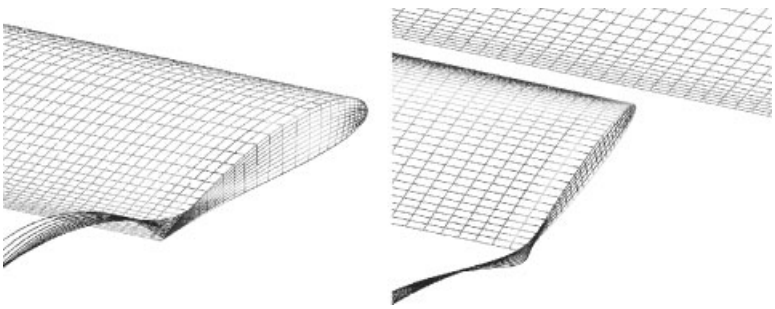

Figure 10. A perspective view of the grid at wing and flap tips.

The symmetry of the model, the inviscid property of the flow to be computed and the assumption that vortices are interacting symmetrically, allows us to use grids that correspond only to one half of the model (see Figure 9 which represents the fine grid). Figure 10 shows the grid at wing and flap tips. It also allows to see the starting of streamlines which represent the wake vorticies.

5.2.1. Results and discussion. We performed three computations on the VPP700 of météo France. The first one corresponds to the fine grid. It is considered as the reference case. In the second (resp. third) computation we use the PG (resp. AMR) technique. All these computations were performed for the value of the angle of attack: $A \circ A = 4^\circ$ and the free-stream Mach number: $M = 0.176$. Note that we use the free-stream velocity U_0 and the wingspan b as velocity and length scales, respectively.

The strategy used in both the PG and AMR computations consists of using an initial coarse grid obtained from the fine one by dividing the number of points by two in each spatial direction. In the PG computation we refine the blocks of the coarse grid which are crossed by wake vortices. However, in the AMR calculation we perform a more local refinement by

Copyright © 2002 John Wiley & Sons, Ltd.

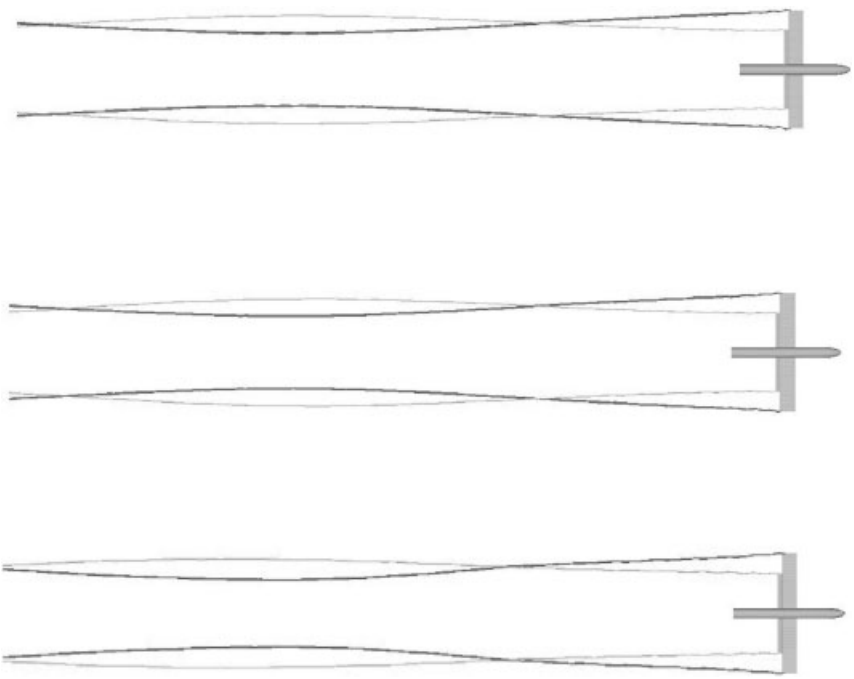

Figure 11. The top view of streamlines in wake vortices. Result obtained using the fine grid, PG and AMR strategies (from top to bottom).

selecting only the parts of these blocks traversed by the vortices (Game process). For this, we use a sensord based on the evaluation of the total pressure gradient. The comparison of Plates 4–6 allow us to have a general idea about the different grids. As we can see on the visualized planes, the refinement of the AMR grid is the most local. The difference between the fine grid and the Patched one can be seen on the fuselage.

Streamline views: We can carry out a qualitative comparison between the different computations by visualizing streamlines in the wake vortices. As we can see in Figure 11 showing the top view of the wake, the evolution of vortices resulting from the use of the PG technique seems to be similar to the reference case. However the AMR result is slightly different. Indeed, the interaction between the two vortices starts later in the AMR computation.

5.2.2. Quantitative analysis of results. Now we focus our attention on the analysis of a plane situated in the wake zone where we have observed a difference between results. Let us fix the position of this plane at 5.2 wingspans downstream of the wing trailing edge.

Velocity profiles: In this section we compare velocity profiles through wing and flap vortices. This alternative allows us to compare the core radius (rc) and the maximum velocity (V_m) for each vortex. As we can see in Figures 12, we plot the vertical component of velocity w versus the spanwise position y. In these figures y_{ci}, where $i = f$ for the flap and $i = w$ for the wing, indicates the spanwise position of the vortex centre.

Copyright © 2002 John Wiley & Sons, Ltd.

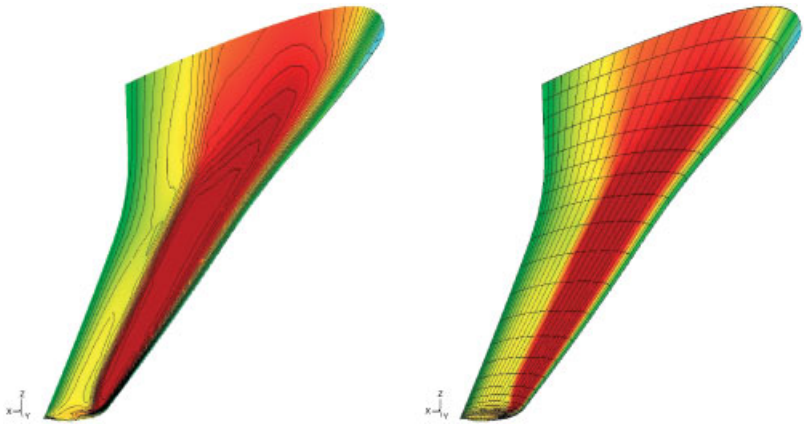

Plate 1. Level 0—mesh and Mach number contournf—$M_\infty = 0.8,\ \alpha = 2.2°$.

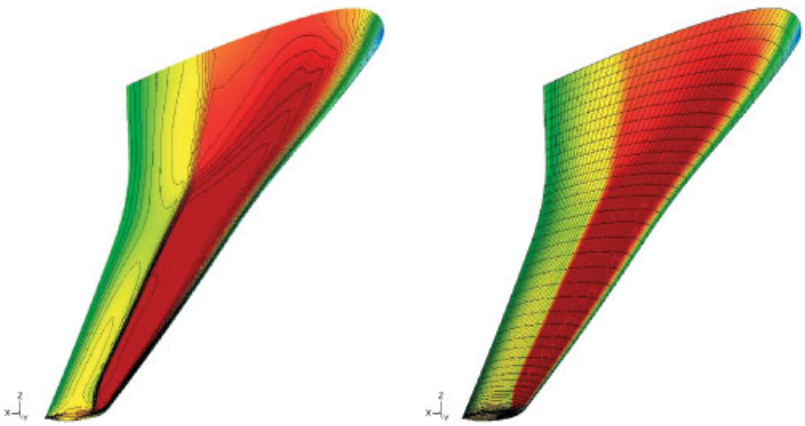

Plate 2. Level 1—mesh and Mach number contours—$M_\infty = 0.8,\ \alpha = 2.2°$.

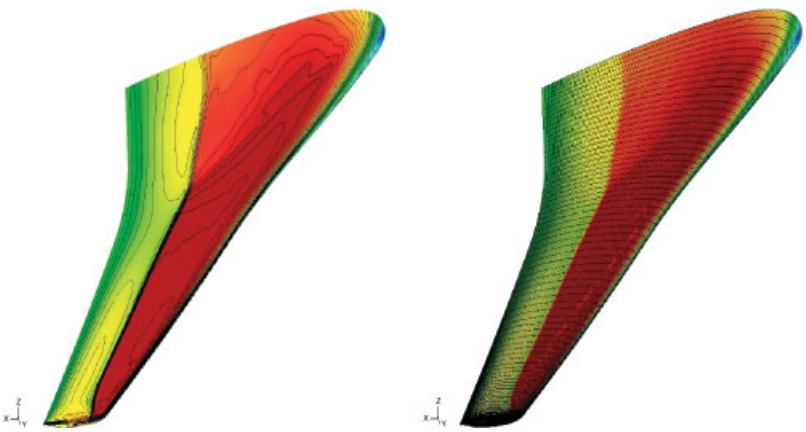

Plate 3. Level 2—mesh and Mach number contours—$M_\infty = 0.8,\ \alpha = 2.2°$.

Copyright © 2002 John Wiley & Sons, Ltd.

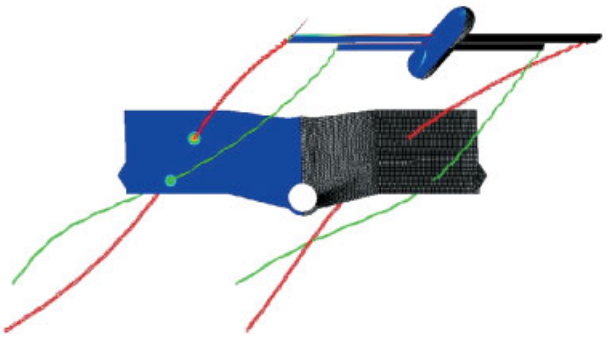

Plate 4. Grid and total pressure in a plane situated at $x/b = 2.9$ downstream of the wing trailing edge. Result obtained using the fine grid.

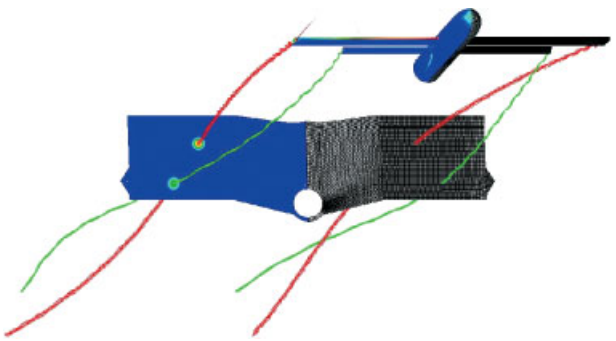

Plate 5. Grid and total pressure in a plane situated at $x/b = 2.9$ downstream of the wing trailing edge. Result obtained using the PG strategy.

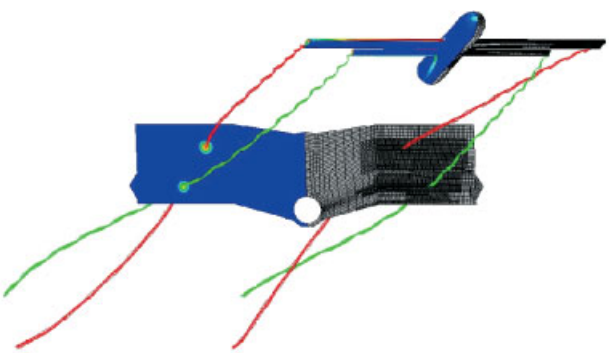

Plate 6. Grid and total pressure in a plane situated at $x/b = 2.9$ downstream of the wing trailing edge. Result obtained using the AMR strategy.

Copyright © 2002 John Wiley & Sons, Ltd.

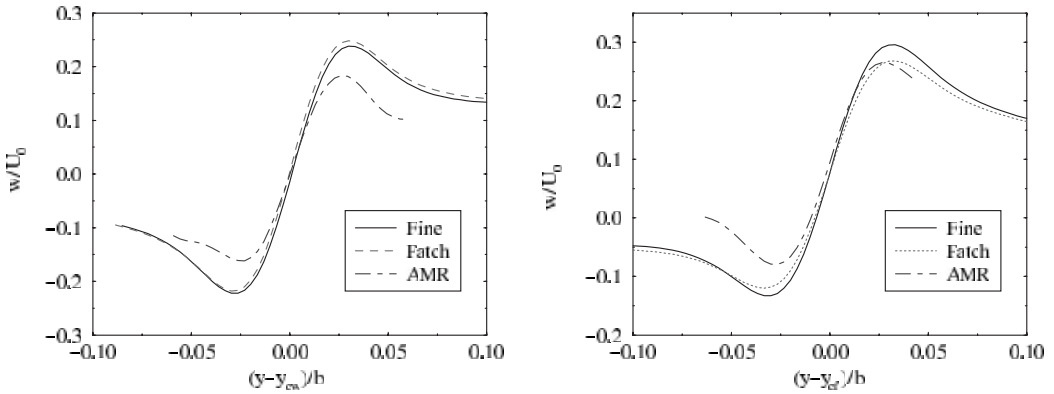

Figure 12. Vertical velocity profile through the wing vortex in a plane situated at 5.2 wingspans downstream of the wing trailing edge—vertical velocity profile through the flap vortex in a plane situated at 5.2 wingspans downstream of the wing trailing edge.

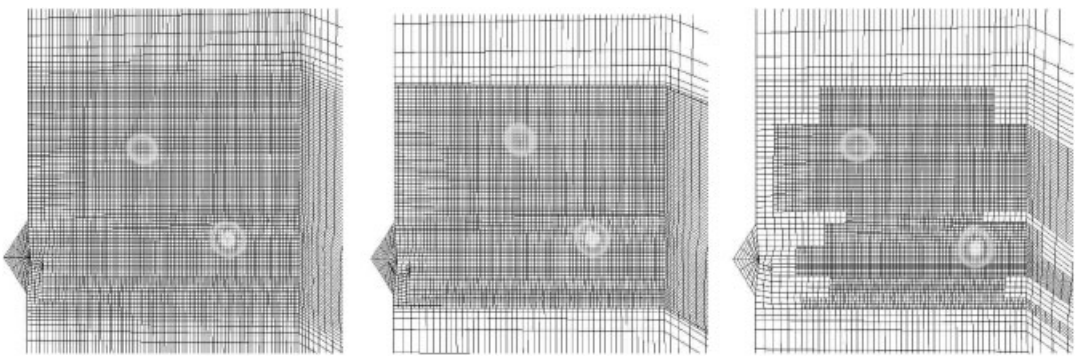

Figure 13. Grid for the reference computation—fine grid—patched grid—AMR grid.

The PG computation gives satisfactory results. For the wing vortex, rc and V_m are very close to the reference results. The situation is slightly different for the flap vortex. Although the vortex has the same size (i.e. the same core radius), the maximum velocity is smaller if we compare it with the reference case.

The result obtained using the AMR technique is relatively different if we compare it to the other computations. For the two vortices, both rc and V_m are smaller. This result explains in some manner the difference which we observe when we compared streamline views. But, what is the origin of this non-satisfactory result?

It is certainly due to the local characteristic of AMR technique. In other words, the refined part used in AMR computation is not sufficient to capture all the dynamics of vortices. This argument is well justified by Figure 13 which shows the position of vortices on the different grids in the selected plane. If the vortex is well surrounded by a refined zone it's dynamics are well captured. This is the case for wing vortex in the PG computation. The comparison of the grids in Figure 13 allows us to explain not only the AMR results but the PG results

Copyright © 2002 John Wiley & Sons, Ltd.

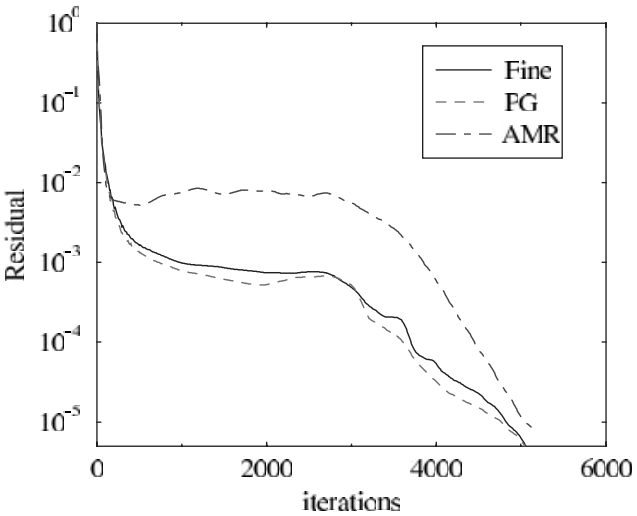

Figure 14. Convergence plots for residual (L_2 norm).

Table 2. CPU time and mesh size for each computation.

	CPU time(h)	Mesh size (points)
Fine	53.75	4.3×10^6
PG	23.33	2×10^6
AMR	13.94	1.4×10^6

too. Indeed, on this last grid (PG) the flap vortex is situated not very far from the extremity of the refined zone. This explains the differences which we observe in Figure 12.

Comparison of computations characteristics: In order to estimate well the benefit realized using the PG and AMR strategies, it is necessary to compare CPU times and mesh sizes. As we can see in Figure 14, the adopted strategy consists of doing the necessary number of iterations until arriving at a pre-fixed value of the residual (5×10^{-6}). The AMR strategy seems to be the most advantageous in terms of CPU time, since it converges quicker than the PG technique. In Table II, we can notice the remarkable difference between the calculations. Using mesh refinement strategies allows us to reduce mesh size and consequently the CPU time (between the reference case and AMR computation (resp. PG computation), there is a ratio of 3.85 (resp. 2.3)).

6. CONCLUSION

PG and AMR strategies have been presented for 3D steady inviscid flows. The two techniques have shown their ability to increase locally the number of nodes. They are well suited to wake vortices computations. However, for these kind of flows, it is not sufficient to refine locally the characteristic structures. Wake vortices calculations need a large encapsulation to avoid the merging of the vortices.

Copyright © 2002 John Wiley & Sons, Ltd.

The AMR strategy is encouraging thanks to the CPU performance but the refinement is too local. We expect to define a more sophisticated sensor in order to refine more widely the vortices. A good solution could be to work with sensors based on an estimation of second derivatives.

For these kind of flows, the PG strategy is well adapted: we keep the same precision by diminishing the number of nodes.

In the future, we will combine these two strategies to use the advantages of both methods.

REFERENCES

1. Laporte F, Corjon A. Simulations Numériques 3D Stationnaires et Instationnaires de sillages d' Avions Gros Porteurs. *Asso. Aéro. et Astro. de France*, 36e Colloque d' Aérodynamique Appliquée; France, March 2000.
2. Vos J, Chaput E, Arlinger B, Rizzi A, Corjon A. Recent advances in aerodynamics inside the NSMB (Navier–Stokes multi-block) consortium. *AIAA Paper*, January 1998; 98-0802.
3. Jameson A, Schmidt W, Turkel E. Numerical solutions of the euler equations by finite volume methods using Runge–Kutta time stepping. *AIAA Paper*, 1981; 81-1259.
4. Roe PL. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 1981; **43**:357–372.
5. Liou M-S, Stephen CJ. A new flux splitting scheme. *Journal of Computational Physics* 1993; **107**:23–29.
6. Weber C. Dévelopement de méthodes implicites pour les équations de Navier–Stokes moyennées et la simulation des grandes échelles: Application á l'aérodynamique externe. *Ph.D. Thesis*, CERFACS/IMFT. Toulouse FRANCE, 1999.
7. Ducros F, Laporte F, Soulères T, Guinot V, Moinat P, Caruelle B. High-order fluxes for conservatives skew-symmetric-like schemes in structured meshes: application to compressible flows. *Journal of Computational Physics* 2000; **161**:114–139.
8. Jameson A, Yoon S. LU implicit schemes with multiple grids for the Euler equations. *Proceedings of 24th Aerospace Science Meeting*, *AIAA Paper*, Reno, 1986; 86-0105.
9. Benek JA, Steger JL, Dougherty FC. 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.
10. Raï MM. Stable conservative multi-domain treatments for implicit Euler solvers. *Journal of Computational Physics* 1986; **62**:472–503.
11. Berger JM. On conservation at grid interfaces. *SIAM Journal of Numerical Analysis* 1987; **24**:967–984.
12. Lerat A, Wu ZN. Stable conservative multi-domain treatments for implicit Euler solvers. *Journal of Computational Physics* 1996; **123**:45–64.
13. Abdol-Hamid KS, Carlson JR, Pao SP. 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 Degio, July 1995.
14. Berger MJ. Adaptive mesh refinement for hyperbolic partial differential equations. *Ph.D. Thesis*, Computer Science Dept., Stanford University, 1982.
15. Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* 1989; **82**:64–84.
16. Quirk JJ. An adaptive grid algorithm for computational shock hydrodynamics. *Ph.D. Thesis*, College of Aeronautics, Cranfield Institute of Technology, 1991.
17. Jouhaud J, Borrel M. A hierarchical adaptive mesh refinement method: application to 2D flows. *Third ECCOMAS Computational Fluid Dynamics Conference*, 1996; 268–274.

Copyright © 2002 John Wiley & Sons, Ltd.