ICAS-92-4.3R

MELINA - A MULTI-BLOCK, MULTI-GRID 3D EULER CODE WITH SUB BLOCK TECHNIQUE FOR LOCAL MESH REFINEMENT

S. Rill and K.Becker
Deutsche Airbus GmbH
Bremen, Germany

Abstract

The Euler code MELINA is based on an explicit 5-stage Runge-Kutta time stepping scheme with blended 4^{th} and 2^{nd} order artificial dissipation proposed by Jameson. This scheme is among the most efficient algorithms for the solution of the unsteady Euler equations. MELINA already has a considerable impact on the aerodynamic design of complex 3D flow problems at Deutsche Airbus (DA) such as laminar glove design, integration of propulsion systems like propfan or turbofan, or simulation of flap track fairings on the wing.

The objective of this investigation is to improve the speed of convergence of the code as well as the spatial accuracy of the results in order to allow for an even more routinely application of 3D flow simulation in the aerodynamic design process at DA.

The speed of convergence of MELINA can be drastically improved by using a multi-grid procedure to expell the disturbances more rapidly through the outer boundary. In order to reduce CPU time even further it is desirable to cluster grid points only in regions of large gradients such as at the wing leading and trailing edge. To achieve this goal, we overlay sub grid blocks in the regions of interest. This leads to the desirable property that the code still vectorizes well and that the refined local sub block can readily be integrated in the multi-grid procedure.

1 Introduction

Due to considerable advances in Computational Fluid Dynamics (CFD) in the last decade it has become possible to simulate flows around 3D complex configurations like complete transport aircraft with the Euler equations on a routinely basis. Even the first pilot applications of Navier-Stokes simulations around such complicated geometries are emerging today.

The purpose of this paper is to describe the newest developments that have been introduced to the proven tool package INGRID-MELINA-PISA, [1] for flow simulation around 3D complex configurations at Deutsche Airbus (DA).

The Multi-grid, multi-block Euler integration algorithm MELINA is continuously being upgraded and adapted for the applications which are of interest to the transport aircraft designer at DA. It may be classified as a 'Jameson-type' explicit, cell vertex scheme with central differences, artificial dissipation and Runge-Kutta time-stepping.

With the Interactive, algebraic grid generator INGRID, [2] several tasks can be tackled. INGRID is tailored to the generation of body-fitted meshes around various transport aircraft configurations. It serves as a geometry definition and manipulating system to create the configuration to be evaluated from given input data. In a second step the surface is covered with a surface grid with full user control of the node distribution. Then a global multi-block mesh can be generated for a wing/body configuration. To add further components, a suitable block frame is cut out of the prior blocks and filled with the surface of those components and the respective 3D mesh. Thirdly, additional smaller components like flap track fairings are put into place. One or more of the prior blocks are split at a suitable mesh plane and free space is generated by pushing the split plane aside using spline techniques. The space is then filled with new mesh blocks which are called implanted blocks.

The Practical interactive solution analysis system PISA is the tool package that has been developed for post processing .

In order to enhance the accuray of the inviscid computation with MELINA and INGRID we simulate the local decambering effect of the boundary layer along the wing span such that experimental and theoretical load distributions match. In each span section of the mesh, the wing airfoil is decambered and rotated in order to let the global angle of attack of the Euler calculation fit to the projected flight or wind tunnel angle and to adapt the local lift and momentum coefficients to the predicted or measured local coefficients.

An impression of the status quo concerning the complexity that can be treated and accuracy that can be obtained with MELINA and INGRID is presented in Figs. 1 and 2. Fig. 1 shows the global configuration of a wing/-body/pylon/engine/flap track fairing combination with the pressure distribution projected onto it, whereas in Fig. 2 a more quantitative comparison between the experimental and theoretical pressure distribution is plotted.

Major problems that still persist for routinely complex 3D flow simulations are, instead of the dramatic improvements in the field of CFD and computer technology, the problems of computational efficiency and spatial accuracy in regions of strong gradients like at the leading edge. Computational efficiency has become of special importance at DA since MELINA is heavily beeing used by the aerodynamic design department in the process of wing design and for the optimization of engine-airframe integration.

For complex configurations it is extremly difficult, if not impossible, to resolve all flow features with the same accuracy on a single mesh. In the multi-block approach, implemented in MELINA and INGRID from the beginning, the flow field is partitioned into distinct zones, each discretized as a single grid block. Different types of grid topologies as well as enrichment and coarsening can be used in each block to improve mesh efficiency.

Among the developments in CFD in the last years, the multi-grid technique and the method of mesh refinement can be identified as major contributions for convergence acceleration and improved spatial resolution, respectively. Impressive improvements in convergence rates have been obtained by several authors using multi-grid technique in combination with similar multi-block schemes like MELINA, [3],[4] and [5].

Contributions to the topic of mesh refinement can be found in [6] and [7] where good experiences with local mesh refinement have been reported for 2D and 3D test cases including computations on an isolated Onera M6 wing.

The objective of the new release of MELINA and INGRID was to combine the benefits of both approaches. This can be achieved by introducing locally refined sub blocks and several global coarse block levels and connect them via a multi-grid technique. The advantage of this approach is to exploit the higher spatial resolution of the locally refined sub blocks in critical regions without spending too much grid points in smooth regions and without paying the price of reduced convergence speed (the convergence rate using multi-grid technique is independend of the spatial resolution).

2 Governing Equations

Inviscid flow can be modelled by the continuum mechanic's conservation equations for mass, momentum and energy. In 3D these are five scalar integral equations which are augmented by the thermic and caloric gas equations for ideal gas as material laws to form a closed system.

For any control volume V the conservation of mass, momentum and energy result in the system of Euler equations

mass (continuity):

$$\frac{\partial}{\partial t}\int_V \rho\, dV = -\oint_{\partial V} \rho\,(\vec{v}\cdot\vec{n})\, dO \qquad (1)$$

momentum:

$$\frac{\partial}{\partial t}\int_V \rho\vec{v}\, dV = -\oint_{\partial V} \rho\vec{v}\,(\vec{v}\cdot\vec{n}) + p\vec{n}\, dO \qquad (2)$$

energy:

$$\frac{\partial}{\partial t}\int_V \rho E\, dV = -\oint_{\partial V} \rho H\,(\vec{v}\cdot\vec{n})\, dO \qquad (3)$$

∂V is the surface of the control volume V and $\vec{n}$ is the outward unit normal vector on ∂V.

The pressure p can be calculated from

$$p = (\kappa-1)\rho\left[E - \frac{u^2+v^2+w^2}{2}\right].$$

wheras $\rho, \vec{v}, E$ and κ are the density, velocity vector, total energy per unit volume and ratio of specific heats of the fluid, respectively.

Eqs. (2)-(4) can be abbreviated by the vector valued equation

$$\frac{\partial}{\partial t}\int_V \vec{W}\, dV = -\oint_{\partial V} \bar{\bar{F}}(\vec{W})\cdot\vec{n}\, dO \qquad (4)$$

where $\bar{\bar{F}}$ is the flux tensor and $\vec{W} = (\rho, \rho u, \rho v, \rho w, \rho E)$ is the vector of conservative variables.

3 Spatial Discretization

3.1 Spatial Cell Vertex Discretization

In order to discretize Eq. (4), a boundary-fitted finite volume mesh, generated by INGRID, is used. In the cell vertex discretization the vertices of the finite volumes are taken as locations for the unknown variables $\vec{W}$. Following Jameson et.al. [8] and others, the discretization by the method of lines decouples time and space directions.

$$\frac{d}{dt}\vec{W}_{i,j,k} = \frac{1}{V_{i,j,k}}(\vec{Q}_{i,j,k}) \qquad (5)$$

where

$$\vec{Q}_{i,j,k} = \oint_{\partial V_{i,j,k}} \bar{\bar{F}}(\vec{W})\cdot\vec{n}\, dO$$

and

$$\frac{\partial}{\partial t}\int_{V_{i,j,k}} \vec{W}\, dV \approx V_{i,j,k}\frac{\partial\vec{W}}{\partial t}|_{i,j,k}$$

This discretization leads to equations attached to finite volumes rather than grid points. Therefore a summation of the equations is applied which takes into account every eight cells connected to one common grid point.

$$\frac{d}{dt}\vec{W}_S = \frac{1}{V_S}\vec{Q}_S = \frac{1}{V_S}\sum_{l=1}^{8}\vec{Q}_l$$

$V_S = \sum_{l=1}^{8} V_l$ is the volume of the supercell constructed from the participating grid cells. If we now denote a supercell and its center (cell vertex) by the index i,j,k and add an artificial filter term to prevent odd-even decoupling or spurious high frequency oscillations we obtain

$$\frac{d}{dt}\vec{W}_{i,j,k} = \frac{1}{V_{i,j,k}}(\vec{Q}_{i,j,k} + \vec{D}_{i,j,k}) \qquad (6)$$

The artificial filter $\vec{D}_{i,j,k}$ consists of two parts: The so-called background filter is necessary to ensure convergence for the whole range of flow speeds to be treated. In Jameson's approach it consists of 4th differences of the solution vector multiplied by an approximation of the largest eigenvalue of the Jacobian matrix $\frac{\partial\bar{\bar{F}}}{\partial\vec{W}}$ and a user-defined constant $\varepsilon^{(4)}$. The second filter term is a second difference. It is designed to avoid oscillations of the solution near shocks and is switched on only in those regions by the use of a special pressure based sensor function.

3.2 Boundary Conditions

Various boundary conditions are implemented in MELINA. We can distinguish between physically motivated boundaries and algorithmic boundaries of the computational domain.

The computational domain may be bounded by solid walls, the farfield, fan face and exhaust planes of the engine and the symmetry plane if one exists. On a solid body the velocity component normal to the wall is zero. Hence, the flux integral parts along the faces of the control volume that are aligned with the body reduce to the pressure integrals of the momentum equations.

The definition of the farfield boundaries is based on characteristic theory, which, if applied to the locally linearized hyperbolic system of Euler equations, leads to a set of conditions for the characteristic variables. A transformation of these variables to the conservative variables $\vec{W}$ used in the rest of the flow field results in proper equations for the values of $\vec{W}$ at the outer boundary.

The boundary condition of symmetry is enforced with the use of guard cells which are mirror images of the cells abutting the symmetry plane.

The boundary condition at the fan face satisfies the massflow specified by the engine data at the flow condition to be evaluated. The boundary condition at the exit of the engine is formulated such that the jumps in stagnation pressure and stagnation temperature given in the engine data are realized.

The category of algorithmic boundary conditions consists of inner cut conditions that manage the exchange of grid coordinates and solution at the nodes from one block to the other depending on the type of blocks that have a common block segment. Practically this is done by introducing guard cells for each block face which leads to an overlapping of blocks that are devided by an inner cut and updating these cells from a inter-block boundary exchange buffer. In the simple case with a one to one correspondance of points on both abutting block faces, the grid coordinates and data at the nodes are simply copied from the corresponding cell of the neighbouring block to the buffer and then to the guard cell.

In order to optimize the distribution of nodes, the options of enrichment and coarsening from one block to an other have been implemented. Enrichment means the splitting of each finite volume into two in each desired cooordinate direction and linear interpolation of the coarse node data, whereas in the case of coarsening every other grid node is dropped in the specified direction.

3.3 Sub Block Refinement

Computing a 3D flowfield around complex configurations it is difficult to resolve all features of the solution to the same accuracy with a uniform mesh. This problem is already somewhat relaxed by using multi-block meshes with the option of enrichment and coarsening of the mesh from one block to the other. The drawback of this feature is the fact that the desirable surfaces of block splitting and the requirements for mesh refinement usually do not coincide. Another pitfall is, that for a posteriori changes of the spatial resolution a complete change of the block topology is required.

This leads to the development of sub block refinement. The idea of sub block refinement is to simply "patch" locally refined mesh blocks onto the existing mesh and connect the additional fine sub blocks with the mesh via a multi-grid technique. At the time beeing, sub blocks are defined a priori or a posteriori to the solution of the flow problem in the grid generator INGRID. Their orientation or extension in the computational domain can be judged and changed interactively with mouse and menu technique. The only limitation to the topology is that a sub block has to lie completely in a grid block of the existing mesh. But a grid block may have various sub blocks and a sub block may have several sub blocks itself.

In the actual version of INGRID, the mesh points of the locally refined sub blocks are constructed with a simple trilinear interpolation of the coarse cells with a special interpolation of the points on the surface using Coons' patches. Thus yielding a doubling of the grid density in all three coordinate directions. Although the results with MELINA on meshes obtained with this strategy are very encouraging, as will be shown later, it is a matter of future investigations to refine the method of sub block construction in such a way that smoother variations in grid stretching are obtained. This is of special importance with respect to the application of the sub block technique to Navier-Stokes simulations.

The sub block approach can be viewed as a compromize between structured an unstructered meshes, combining the benefit of high computational efficiency on structured meshes and of clustering grid points in a "quasi unstructured" way by scattering sub blocks and even further refined blocks in regions of strong gradients. It is envisaged but not yet realized to use this method for solution adaptive mesh refinement if the regions of sub block refinement are determined automatically during the iteration by suitable sensor functions.

The interior grid points of the sub blocks are interacting with the coarser block via a multigrid technique that will be described in the next section. For the boundary points of the sub blocks we use the same boundary condition routines as for the coarser blocks. For the cut conditions along the edges of the fine sub block we again introduced guard cells that are updated by an additional inter-block boundary exchange buffer. This buffer field had to be added, because with a sub block the coarse block requires not only data exchange with other blocks along its "external" block faces but also with one or more sub blocks along "internal" block faces.

4 Solution Procedure

4.1 Time-stepping Scheme

For the time-stepping scheme a hybrid, explicit 5-stage Runge-Kutta time stepping scheme is chosen. If we define the residual in equation (6) as

$$\vec{R}^k = Q(\vec{U}^k) + \sum_{m=0}^{k} \gamma_{k,m} D(\vec{U}^m)$$

we obtain

$$\begin{aligned} \vec{U}^0 &= \vec{W}^n \\ \vec{U}^k &= \left(\vec{U}^0 - \frac{\alpha_k \Delta t}{V} \vec{R}^{k-1}\right), k = 1,2,3,4,5 \\ \vec{W}^{n+1} &= \vec{U}^5 \end{aligned} \qquad (7)$$

For convergence acceleration additional techniques like local time stepping, implicit residual averaging with variable coefficients and enthalpy damping (if applicable) are used. The coefficients $\gamma_{k,m}$ are chosen such that we obtain a hybrid (5,2) Runge-Kutta scheme that is to evaluate the filter terms only in the first two of the five Runge-Kutta steps. The parameters α_k are

$$\alpha_1 = \frac{1}{4}, \alpha_2 = \frac{1}{6}, \alpha_3 = \frac{3}{8}, \alpha_4 = \frac{1}{2}, \alpha_5 = 1.$$

4.2 Multi-grid Acceleration

A general outline of the multi-grid procedure for the special case of a 3-level W-cycle with 1-stage local sub block refinement is presented in Fig. 3. There are various possibilities for the implementation of a multi-grid algorithm in the context of a multi-block environment which can lead to extrem discrepancies in the convergence behaviour or even to divergence if not properly done, Rossow [5]. The reasons for these convergence problems are the time lags that may occur between the grid blocks during a multi-grid cycle.

The multi-grid implementation that has been optimized with respect to robustness and minimization of IO-operations will be presented here. For the organization of the operations that have to be performed on the grid blocks during the multi-grid procedure we define the following nomenclature:

We distinguish between three different types of grid blocks.

- The first class is categorized as global fine blocks. All blocks of this category together form the global fine mesh around the configuration.
- The second category consists of the sub blocks which are local refinements of some of the global fine blocks. These sub blocks must not but may be connected to each other. One may think of them as beeing scattered over the computational domain in an unstructured manner in regions of strong gradients in the solution or geometry. Sub blocks may have locally refined sub blocks themselves.
- The third category consists of blocks that are defined as global coarse blocks. These blocks are obtained from global fine blocks by coarsening the latter m times depending on the number of multi-grid levels that shall be used in the time-stepping scheme.

From these block categories we form block groups, symbolized by the squares in Fig. 3. The block groups are the building blocks of the multi-grid cycles (V-cycles, W-cycles etc.) in the sense that the same operations are perfomed on each of the blocks of the group at any stage of the multi-grid cycle.

In order to precondition the starting solution of the multigrid cycle we apply the full multi-grid strategy. That is in our case to use three iteration levels coarse-medium-fine. The coarse iteration level is obtained from the fine level by coarsening each of the blocks, including the global coarse blocks, twice and reducing the depth of the multi-grid cycle by 2. (See Fig. 3). The medium iteration level is generated in the same way, except that the blocks are coarsend only once and the depth of the multi-grid cycle is reduced by one level. Usually we perfom about 30 multi-grid cycles on the coarse and medium iteration levels until we start the final fine multi-grid cycles.

After outlining the multi-grid acceleration, we would like to discuss in more detail the individual operations that have to be perfomed on the block groups in each step of the multigrid cycle, symbolized by the filled squares in Fig. 3 . Let us consider the interaction between two grid levels, fine and coarse with spatial step sizes h and 2h, respectively. On the fine level we perfom the Runge-Kutta time steps as before, (Equ. (7)). With the updated solution $\vec{W}_h^{n+1}$ and the final residual $\vec{R}_h^5$ we determine the forcing function on the coarser mesh $\vec{f}_{2h}^0$ as

$$\begin{aligned} \vec{U}_{2h}^0 &= INJ_{h->2h}(\vec{W}_h^{n+1}) \\ \vec{f}_{2h}^0 &= RES_{h->2h}(\vec{R}_h^5) - \vec{R}_{2h}^0 \end{aligned} \qquad (8)$$

and carry out five Runge-Kutta steps on the coarse mesh

$$\begin{aligned} \vec{U}_{2h}^k &= \left(\vec{U}_{2h}^0 - \frac{\alpha_k \Delta t}{V}(\vec{R}_{2h}^{k-1} + \vec{f}_{2h}^0)\right), k = 1,2,3,4,5 \\ \vec{W}_{2h}^{n+1} &= \vec{U}_{2h}^5 \end{aligned} \qquad (9)$$

The correction of the solution on the coarser mesh is computed from

$$\vec{C}_{2h} = \vec{W}_{2h}^{n+1} - \vec{W}_{2h}^n$$

and interpolated to update the solution on the finer mesh

$$\vec{W}_h^{n+1} = \vec{W}_h^{n+1} + INT_{2h->h}(\vec{C}_{2h}) \qquad (10)$$

RES,INJ and INT in Eqs. (8) and (10) are the operators for the restriction of the residual (full weighting), for the injection of the fine solution to the corresponding points of the coarse grid and for the trilinear interpolation of the corrections from the coarse grid onto the grid points of the fine mesh, respectively.

The corresponding block diagram is depicted in Fig. 4. It is easy to identify the elements of the conventional scheme with no multi-grid acceleration, like updating the boundary conditions and Runge-Kutta time stepping, in the main path of the algorithm. The additional operations on each block group for the multi-grid technique depend, generally speaking on "from where are we coming" and "where are we going" in the cycle. For the robustness of the scheme it has been found as beeing essential that the inter-block boundary exchange buffer is updated only after all blocks of one block group had been updated by the Runge-Kutta steps. This insures that there are no time lags between the various grid blocks of one block group.

5 Results

Among the applications of the CFD software package INGRID-MELINA-PISA are quite complex 3D geometries like wing/body/pylon/engine with flap track fairings mounted on the wing. The simulations around such configurations are performed on a routinely basis at DA and are part of the integrated aerodynamic design process to supplement and/or replace wind tunnel experiments for the validation of distinct leaps in the iterative aerodynamic design cycle, [9].

The target of this investigation is to demonstrate the capability of the new multi-grid approach and the technique of local sub block refinement to accelerate the convergence of the time iteration and to improve the spatial resolution of the scheme, respectively. The new version of MELINA was applied to two configurations: a generic quasi 2D NACA0012-wing and a wing/body combination of a modern transport aircraft.

The mesh for the NACA0012-wing test case was generated with a 2D grid generator using the technique of konformal mapping. The 2D O-mesh around the NACA0012 airfoil is copied and shifted in the spanwise direction to form a 3D mono-block mesh around an unswept NACA0012-wing. In spanwise direction the wing is bounded by solid wall conditions, to simulate a flow that is comparable to an experiment in a 2D wind tunnel test section with adaptive upper and lower walls. In order to study the influence of mesh block splitting, the 3D mesh can be split at every desired position. It is also possible to add locally refined sub blocks to test the influence of sub block refinement.

Various block topologies without sub block refinement have been tested and compared with respect to convergence rate and robustness. With the specific implementation of the multi-block, multi-grid procedure depicted in Figs. 3 and 4 we observed only very limited differences between multi-block and mono-block meshes concerning these criteria such that we concentrate on the special topology of one mesh block on the upper wing side and one block below the wing which is the basic block topology for transport aircraft configurations at DA.

An impression of the global configuration of the NACA0012-wing can be drawn from Fig. 5 that presents a plot of the pressure distribution on the surface of the wing and in a plane normal to the wing surface for the standard NACA0012 test case conditions of $Ma = 0.8$ and 1.25^0 angle of attack. The mesh is split in an upper and lower mesh block along the symmetry plane of the wing. The orientation of two locally refined sub blocks in the region of the leading edge of the wing is also indicated by their boundary surfaces.

A more quantitative evaluation of this test case is illustrated in Figs. 6 to 10. Let us first focus on the problem of spatial accuracy. Fig. 6 shows a plot of lines of constant total pressure loss for the coarse mesh with 80 cells around the crossection and 16 cells normal to the wing. The total pressure gains and losses at the leading edge are no physical phenomena and result exclusively from spatial discretization errors. In the case of the coarse mesh, the total presure losses, that add up to a peak value of 7.3 percent, are convected downstream.

If we compare the distribution of the total pressure losses of the corresponding test case with local sub block refinement at the leading edge in Fig. 7, we can observe a reduction of the peak level by a factor of approximately 4. This is the improvement that one would expect for a second order accurate scheme (doubling of the grid density leads to a reduction of discretization errors by a factor of 4). A comparison with a global fine calculation with 160*32*8 finite volumes confirms that it is possible to obtain locally the same accuracy with sub block refinement as with a global fine mesh. In the plot of lines of constant Mach number in Fig. 8 for the test case with sub block refinement, the isomach lines, which are a quite sensitive indicator for spatial disretization errors, leave the region of local refinement almost with no distortion. This behaviour has been observed for all cases computed so far, indicating that the boundary condition for the cut interface from locally refined sub blocks to the coarser blocks are treated correctly by MELINA.

Figs. 9 and 10 present a comparison between the convergence behaviour of the new code running in the 4-level multi-grid mode without sub blocks and in the 3-level multi-grid mode with sub block refinement as well as the convergence history of the scheme on a single mesh. The results correspond to the test cases discussed already in Figs. 6 to 8. The convergence behaviour is plotted as the reduction of the residual $\vec{R}_h^5$ on the finest mesh (defined in Section 4.1) and the development of the lift coefficient of the configuration as functions of the time step or the CPU time, in Figs. 9 and 10 respectively. In all three cases, the mesh density of the finest grid-level corresponds to the density of a 160*32*8 mesh. The full multi-grid strategy (See Fig. 3) was applied by starting the iteration on the medium iteration level.

If we look at the development of the lift in Fig. 9 as a function of timesteps we observe the dramatic improvement of the lift convergence behaviour that is typically achieved with multi-grid acceleration. The same applies to the reduction of the residual. With multi-grid acceleration we obtain reduction rates of the residual in the order of .88 to .89 in contrast to a rate of .98 to .99 in single mesh applications. In this graph we count one timestep on the single mesh and one multi-grid W-cycle equally as one timestep, which is obviously not "fair" since a W-cycle is more expensive than a timestep on the fine mesh alone. The curve is also somewhat misleading since the 30 iterations on the medium iteration level (full multi-grid strategy) are counted like timesteps on the fine iteration level.

A more realistic comparison is shown in Fig. 10 where residual and lift convergence are plotted as functions of CPU time. Although the gain in lift convergence with the multi-grid approach is diminished, we still come up to a reduction of 70 percent of the CPU time with the multi-grid method relative to the single mesh. Concerning lift convergence a noticable improvement of the 3-level multi-grid mode with sub block refinement relative to the 4-level multi-grid mode has not been observed. Also it seems to be somewhat fortitious that the lift converged to exactly the same values on the global fine meshes (4-level multi-grid and single mesh) compared to the coarse mesh with local sub block refinement, since instead of the good resolution at the leading edge, with local refinement there are still some discrepancies in the solutions especially in the shock region.

The mesh for the more complex case of a wing/body configuration of a modern transport aircraft configuration was generated with INGRID and has an H-type structure in streamwise direction and is of O-type normal to that direction. The mesh consists of approximately 700000 cells in 4 blocks (See Fig. 11). The two interior blocks 1 and 2, that cover the upper and lower surface, are separated by the wing plane and the two exterior blocks extend to the farfield. In order to optimize the distribution of finite volumes, the option of coarsening and enrichment from one mesh block to the other is used extensively. Because wing/body computations are usually performed as baseline computations for engine installation prediction at DA, a high mesh density is required in block 1 from which the engine block may be cut out and reconfigurated. The external block 3 is coarsened relative to block 1 in all three coordinate directions. The upper side blocks 2 and 4 are treated likewise except that both are coarsend in j-direction relative to the lower side blocks 1 and 3. The strategy of coarsening and enrichment should not be mixed up with the technique of local sub block refinement, but means that the individual blocks of the global fine mesh level have different grid densities in the 3 coordinate directions.

Because of the coarsening it was only possible to use 3 multi-grid levels. Again we applied the full multi-grid approach starting with 40 W-cycles on the medium iteration level and 100 W-cycles on the fine iteration level. The lift converged after 70 cycles on the fine iteration level and the residual convergence rate was determined to be 0.89, (See Fig. 12). With respect to CPU time we achieved a reduction from 120 min on a VP200 vector computer on the single mesh for this configuration to 43 min, which corresponds to a reduction of 65 percent.

6 Conclusions

It has been demonstrated that the convergence behaviour of the multi-block Euler code MELINA is dramatically improved by the implementation of a multi-grid acceleration technique. The CPU time has been reduced by up to 70 percent.

The technique of local sub block refinement also fits nicely in the multi-grid algorithm and offers the possibilty to improve the spatial resolution of the scheme. The method of sub block refinement can be viewed as a compromise between structured and unstructured discretization, combining the advantages of both approaches. The basic scheme without sub block refinement is of second order accuracy in space. Investigations of the total pressure loss in regions of local refinement have shown that the technique of local sub block refinement conserves this second order accuracy.

The developments implemented in the grid generation system INGRID and the Euler code MELINA discussed in this paper will enable us to apply the 3D Euler simulation to more flow problems than in the past and to perform more iteration steps in the aerodynamic design of complex 3D geometries.

Future developments will treat the generalization of the method for the simulation of viscous flows. This can be done by introducing viscous sub blocks that are placed on the surface of the configuration and connecting these to the global mesh again via the multi-grid approach.

References

[1] Rill, S., Becker, K., "Simulation of Transonic Inviscid Flow over a Twin Jet Transport Aircraft", AIAA Paper 91-0025, 1991.

[2] Becker. K., "Interactive Algebraic Mesh Generation for Twin Jet Transport Aircraft", it Proc. 3rd Int. Conf. Num. Grid Generation, Barcelona, June 1991.

[3] Atkins, H.L., "A Multi-Block Multigrid Method for the Solution of the Euler and Navier-Stokes Equations for Three-Dimensional Flows", AIAA Paper 91-0101, 1991.

[4] Radespiel, R.,Rossow, C., Swanson, R.C., "Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations", AIAA Journal, Vol. 28, NO. 8, pp. 1464-1472, 1990.

[5] Rossow, C. C., " Computation of Inviscid Flow Fields Around Complex Configurations Using a Multiblock Multigrid Method", *5th Copper Mountain Conference on Multigrid Methods*, March 31 - April 5, Colorado, USA, 1991.

[6] Berger, M.J., "Automatic Adaptive Grid Refinement for the Euler Equations", AIAA Journal, Vol. 23, NO. 4, pp. 561-568, 1985.

[7] Eriksson, L.-E., "Numerical Solution of the Euler Equations using Local Grid Refinement", FFA Report 143, Flygtekniska Försöksanstalten, The Aeronautical Research Institute of Sweden.

[8] Jameson, A., Schmidt, W., Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes", AIAA Paper 81-1259, 1981.

[9] Greff, E., Becker, K., Karwin, M., Rill, S., "Integration of High By-pass Ratio Engines on Modern Transonic Wings for Regional Aircraft", DGLR Paper 91-160, *DGLR-Jahrestagung*, Berlin, 1991.

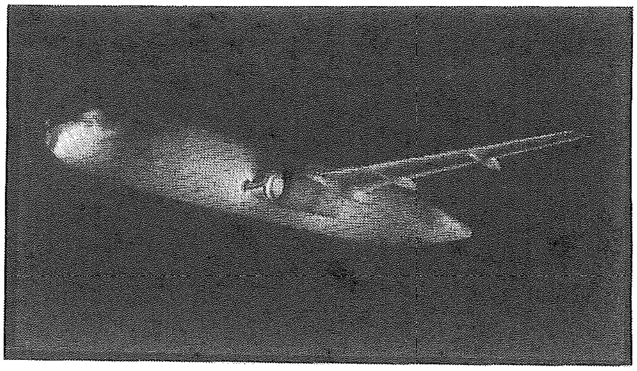

Fig. 1 Application of MELINA to transport aircraft configuration

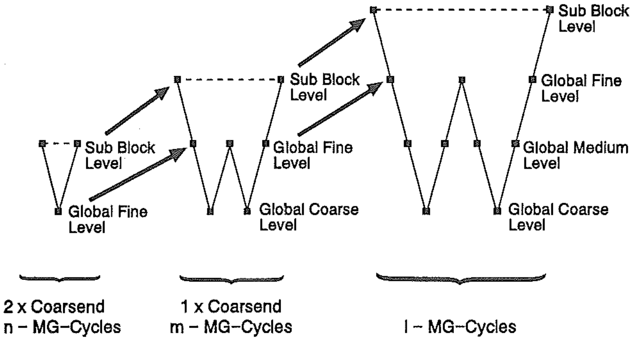

Fig. 3 Full multi-grid procedure for 3-level W-cycle with 1-stage sub block refinement

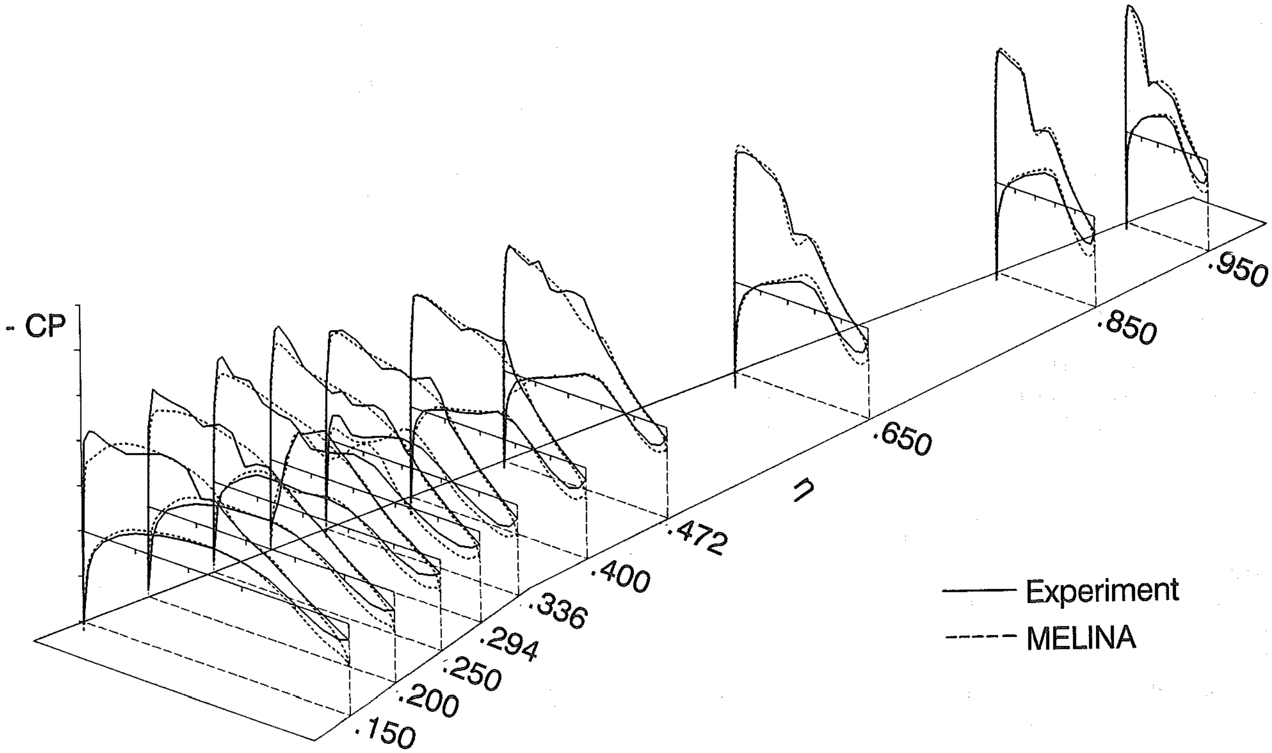

Fig. 2 Comparison of pressure distribution on transport aircraft

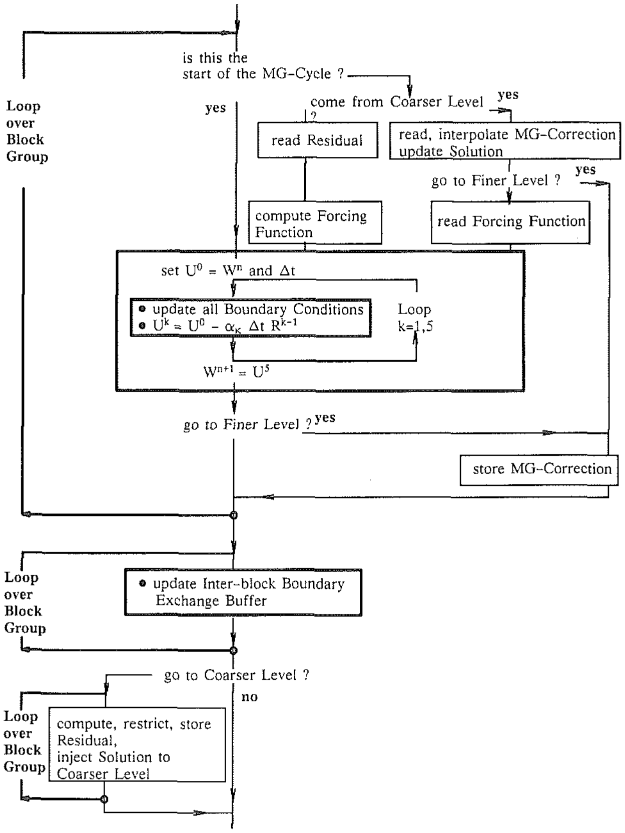
is this the start of the MG-Cycle ?
Loop over Block Group
yes
come from Coarser Level ?
yes
read Residual
read, interpolate MG-Correction update Solution
go to Finer Level ?
yes
compute Forcing Function
read Forcing Function
set $U^0 = W^n$ and Δt
• update all Boundary Conditions
• $U^k = U^0 - \alpha_K \Delta t R^{k-1}$
Loop k=1,5
$W^{n+1} = U^5$
go to Finer Level ?
yes
store MG-Correction
Loop over Block Group
• update Inter-block Boundary Exchange Buffer
go to Coarser Level ?
no
Loop over Block Group
compute, restrict, store Residual, inject Solution to Coarser Level

Fig. 4 Block diagram for a step in the multi-block, multi-grid cycle

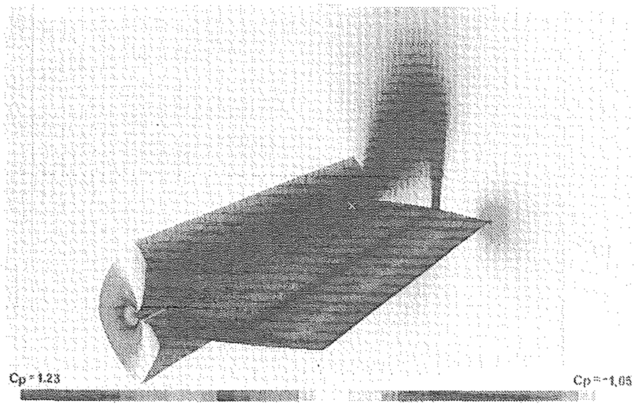

Fig. 5 NACA012-wing test case - pressure coefficient 80*16*4 mesh with sub blocks at leading edge $Ma = 0.8$ and $\alpha = 1.25^o$

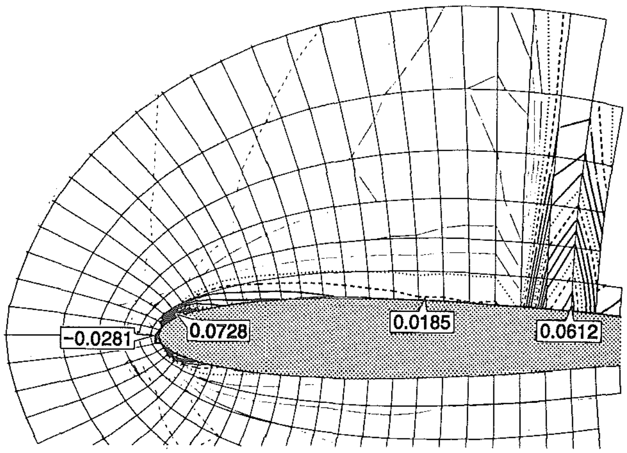

Fig. 6 Total pressure loss on 80*16*4 mesh NACA0012-wing, $Ma = 0.8$ and $\alpha = 1.25^o$

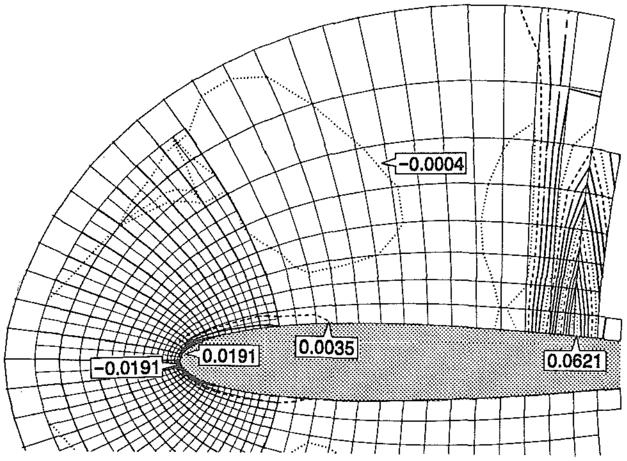

Fig. 7 Total pressure loss on 80*16*4 mesh with sub block NACA0012-wing, $Ma = 0.8$ and $\alpha = 1.25^o$

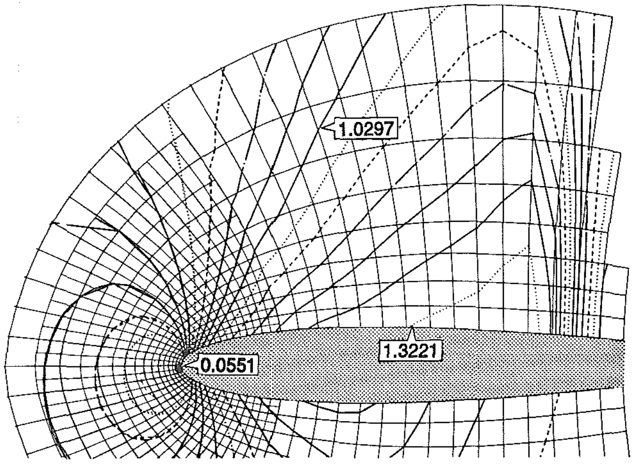

Fig. 8 Isomach lines on 80*16*4 mesh with sub block NACA0012-wing, $Ma = 0.8$ and $\alpha = 1.25^o$

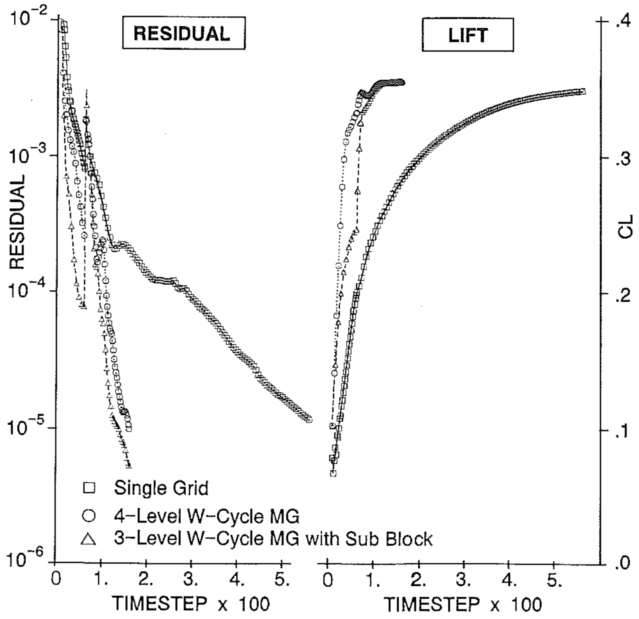

Fig. 9 Convergence history for NACA0012 test cases as function of timesteps

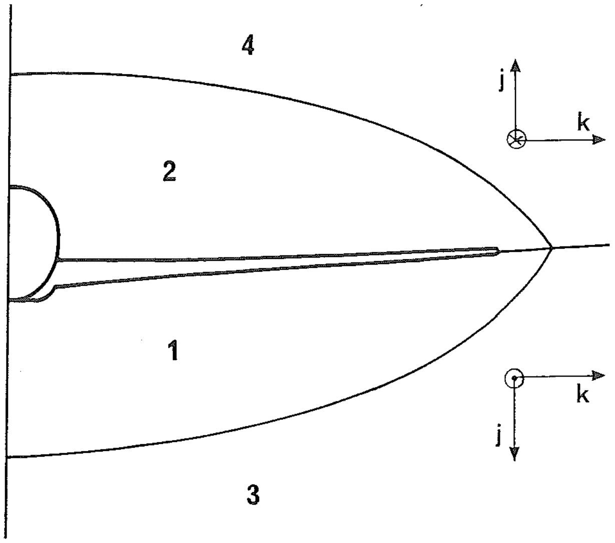

Fig. 11 Block topology for wing/body configuration

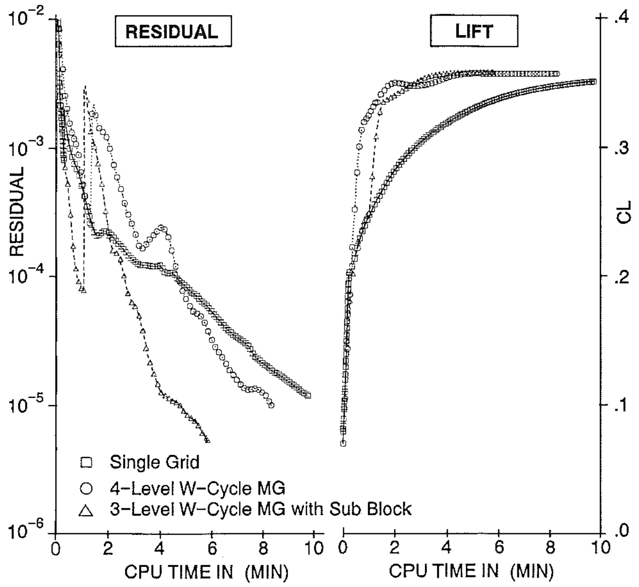

Fig. 10 Convergence history for NACA0012 test cases as function of CPU time

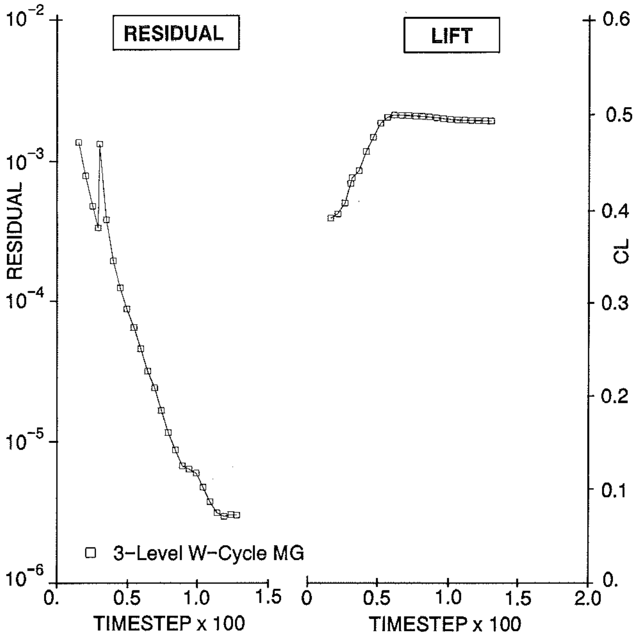

Fig. 12 Convergence history for wing/body configuration