A98-31516

ICAS-98-2,7,3

A Multigrid Algorithm for Inviscid Flow Computations on Unstructured Grids

T. Berglind* and L. Tysell**
The Aeronautical Research Institute of Sweden
Bromma, SWEDEN

Abstract

This work demonstrates the efficiency of multigrid convergence acceleration technique for an overset grid method. The flow solver scheme is based on an explicit, vertex-based, finite volume scheme with central difference type of spatial discretization. In contrast to multigrid for structured grids, the grid quality inevitably degrades with the level of the coarse grids. The importance of robust coarse grid operators for multigrid algorithms on unstructured grids is discussed. A new coarse grid operator is proposed. The capabilities of the multigrid scheme are assessed by solving the flow equations over configurations of varying geometrical complexity. The convergence by multigrid alone is accelerated by an order of magnitude in both 2D and 3D.

1 Introduction

Since explicit schemes only make use of local information, a large number of iterations is required to transmit information across the computational domain. This becomes extensively CPU-demanding for large problems. The multigrid method is a powerful method of accelerating the convergence to steady-state of a numerical scheme. The usefulness of multigrid for hyperbolic equations was first demonstrated by Ni[1] and later by Jameson[2]. Multigrid methods offer an alternative to implicit methods in order to efficiently solve large problems, requiring low additional memory overhead.

The multigrid methods for unstructured grids imply additional challenges. Coarser grids cannot be generated in a simple way from a fine grid as is the case for structured grids. In the last decade two main approaches to multigrid for unstructured grids have been explored, overset meshes and control volume agglomeration.

The *method of overset grids* implies that coarse grids are generated completely independent of the fine grid and may not even contain any common points. This is possible because the underlying theory of multigrid methods does not assume any relation between the various grids, only that information can be transferred back and forth between the grid levels. This technique was first demonstrated by Löhner and Morgan[3] and later by Mavriplis[4]. The approach provides great flexibility in determining the appearance of the various grid levels.

The main disadvantage of the method of overset grids is that the coarse grid generation generally is non-automatic. Techniques have been proposed to circumvent this disadvantage. This usually involves the removal of selected fine grid vertices and the triangulation of the remaining grid points[5] [6]. Olliver-Gooch[7] has presented an interesting approach for which the coarse triangulation is derived incrementally from the fine triangulation by removing vertices, guaranteeing a valid triangulation at every step in the procedure. The possibility to extend this technique to three dimensions is however unclear.

The *method of control volume agglomeration*, introduced by Lallemand et al [8], means that control volumes are fused together to larger control volumes. For a vertex-based scheme the control volumes are taken as the cells defined by the dual mesh formed by drawing the triangle median segments. The idea of the agglomeration method is to agglomerate neighboring fine grid control volumes, creating a smaller set of control volumes.

In this work we have adopted the method of overset grids to implement in the FFA flow solver *FLUID* for unstructured grids[9] [10]. This flow solver is applicable to both 2D and 3D flow computations. The numerical method is described in section 3.

2 Grid Generation

2.1 Grid generation

All grids are generated by TRITET[11], an in-house code for generation of unstructured 2D and 3D grids. The code is based on the advancing front technique. The

---

* Senior Scientist ** Research Scientist

21st ICAS Congress
13-18 September 1998
Melbourne, Australia

The code is based on the advancing front technique. The ADT data tree structures[12] is implemented in order to reduce the computational time. The CPU time follows the asymptotic Nlg(N) behavior of the ADT algorithm.

Input to the grid generation are a set of surface patches and the background nodes. The background nodes are automatically connected to a background grid by a Delaunay algorithm.

Adaptive grids are generated by use of a gradient sensor. Several key-variables can be monitored simultaneously. The sensor permits directionally stretched grids. As key-variables all the normalized primitive variables are used. Thus, it should be possible to detect main features in the flow. The adaptation can start from very coarse initial grids.

2.2 Coarse grid construction

The generation of very coarse unstructured grids is difficult, especially in three dimensions. The preservation of short characteristic edges in the geometry may result in some cells of low quality, since the specified cell sizes may be much larger than the length of these edges. The coarse grids are generated by the same background grid as the fine grid, but the specified cell sizes (transformation matrix) are multiplied by a scalar factor.

In order to achieve the best multigrid performance, the coarsening ratio between subsequent grid levels should be 1:4 in 2D and 1:8 in 3D. It is usually possible to generate four grid levels, even for complex configurations. In cases we have been able to generate a fifth grid level, it has not improved the multigrid efficiency.

3 Numerical Scheme

3.1 The Euler flow model

The Euler equations which govern inviscid compressible 3D flow written in the integral form for a control volume Ω with the boundary $\delta\Omega$ read:

$$\frac{\partial}{\partial t}\int_{\Omega} \vec{q}\, dV + \oint_{\partial\Omega} \bar{F}(\vec{q}) \cdot d\vec{S} = 0 . \qquad (1)$$

Here $\vec{q}$ represents the conserved quantities: density, the components of momentum, and total energy. $\bar{F}(\vec{q})$ represents the inviscid flux density tensor, V the volume and $\vec{S}$ the surface vector.

3.2 Spatial discretization

A vertex based scheme is chosen since it requires substantially less memory overhead than a cell centered scheme. The control volume for a vertex consists of the outer faces of all contiguous tetrahedra. This implies that the control volumes will be overlapping. The scheme is however conservative in the q quantities since, for each contribution to the summation of convective fluxes for a vertex, a term with the same magnitude but with opposite sign will appear in a similar summation for a neighbor vertex. The integral formulation of the governing Equations (1) is applied to each control volume

$$\frac{\partial}{\partial t}(V_i \vec{q}_i) + \sum_{j=1}^{n_i} \bar{F}_j \cdot \Delta\vec{S}_j = 0 , \qquad (2)$$

where i is the node at the center of the control volume, n_i is the number of faces of the control volume, and ΔS_j is the surface vector for the face. The flow variables inside the flux summation are the arithmetic means over variables at the face vertices.

This discretization corresponds to a finite-volume formulation using the trapezoidal rule of integration.

3.3 Dissipation

A proper form of numerical dissipation for a central scheme on structured grids consists of a blend of second and fourth differences according to Jameson et al. [13]. A similar construction can be made for unstructured grid schemes. In this case the dissipative operator consists of a blend of a harmonic and a biharmonic operator, see Jameson and Mavriplis[14],

$$D(q_i) = \sum_{k=1}^{m_i} (\varepsilon_{ik}^{(2)}(q_k - q_i) + \varepsilon_{ik}^{(4)}(\nabla^2 q_i - \nabla^2 q_k))\frac{(\lambda_k + \lambda_i)}{2} \qquad (3)$$

∇^2 represents the undivided Laplacian operator on a unstructured grid. $\varepsilon_{ik}^{(2)}$ is a pressure switch and $\varepsilon_{ik}^{(4)}$ is a coefficient for the biharmonic operator, λ is the spectral radius and m_i is the number of edges contiguous to the node i. In contrast to structured grid schemes, the harmonic and biharmonic operators are isotropic.

We define,

$$\nu_i = \kappa^{(2)} \frac{\left| \sum_{k=1}^{m_i} (p_k - p_i) \right|}{\sum_{k=1}^{m_i} (|p_k| + |p_i|)} \qquad (4)$$

where $\kappa^{(2)}$ is an empirically determined coefficient. The pressure switch $\varepsilon_{ik}^{(2)}$ is taken as the maximum value of ν_i at the two end vertices of the edge,

$$\varepsilon_{ik}^{(2)} = max(\nu_i, \nu_k) . \qquad (5)$$

In the vicinity of a shock wave the biharmonic operator tend to produce overshoots. It is therefore turned off

by

$$\varepsilon_{ik}^{(4)} = max(0, \kappa^{(4)} - \varepsilon_{ik}^{(2)}) \quad (6)$$

where $\kappa^{(4)}$ is a constant coefficient.

Since, in Equation (3), both $\varepsilon_{ik}^{(2)}$ and $\varepsilon_{ik}^{(4)}$ is symmetric with respect to i and k, also the dissipation operator is conservative with respect to the conservative quantities q.

3.4 Boundary conditions

An impermeable wall boundary condition is applied at solid walls and at the symmetry plane. The contributions to the boundary integral in Equation (1) are zero except the pressure terms in the momentum equations. This boundary condition is only set through the fluxes, the so called *weak formulation*. This means that the velocities at the boundary are in general not tangential to the boundary for a not yet converged solution.

For the alternative, the *strong formulation*, the normal velocities are explicitly set to 0. It is however quite complicated to implement the strong formulation for this type of discretization since we know the surface normals not at the vertices but at the mid points of the surface elements. This implies that for each update of the boundary conditions, a system of equations has be solved. This becomes quite expensive in 3D.

At inflow-outflow boundaries, characteristic boundary conditions are applied, see Berglind[9]. At a subsonic outflow boundary it is possible to specify either the ingoing characteristic, the pressure or the normal velocity.

4 Multigrid Algorithm

4.1- The FAS-scheme

The FAS-scheme is a generalization of the multigrid correction scheme for linear operators to non-linear operators[18]. The coarse grid equation can be written as

$$L_H \bar{u}_H = -I_h^H r_h - (-L_H \bar{I}_h^H \bar{u}_h) \quad (7)$$

where L is the nonlinear operator, u is the solution and r is the residual. The subscript h refers to the fine mesh and the subscript H refers to the coarse grid. I_h^H is the restriction operator for the residual and $\bar{I}_h^H$ the restriction operator for the solution.

The right hand side of this expression is called the forcing function. Once the coarse grid equations have been advanced in time, the fine grid variables are updated as,

$$\bar{u}_h^{new} = \bar{u}_h^{old} + I_H^h(\bar{u}_H^{new} - I_h^H \bar{u}_h^{old}) = \bar{u}_h^{old} + v_H \quad (8)$$

, where v_H is the coarse grid correction.

If the transferred residual in Equation (7) is 0 then the coarse grid solution will equal the transferred solution. According to Equation (8) this implies that no corrections of the fine grid solution will be performed. Consequently at steady state, the solution calculated by the multigrid FAS-scheme will have the same accuracy as the solution of the fine grid problem.

4.2 Intergrid transfer operators

The flow variables are interpolated using bilinear interpolation, i.e. the flow variables in the coarse grid vertex C1 are interpolated from the fine grid triangle F1-F2-F3 enclosing the vertex, Figure 1. The corrections of the prolongation are interpolated in the same way.

It is desirable that the transfer of residuals is conservative, i.e. the sum of residuals of the fine grid should equal the sum of the transferred residuals to the coarse grid. According to Mavriplis[4] this can be achieved by, for each fine grid residual at a vertex FI in Figure 1, distribute the residual value to the three vertices of the enclosing coarse grid triangle C1-C2-C3. By using the same weights as given by the bilinear interpolation, the restriction operator becomes the transpose of the prolongation operator. Therefore, the same interpolation tables are used for both the restriction and the prolongation operator. Each coarse grid vertex FI receives residual contributions from all fine grid points which lie in a coarse grid triangle contiguous to P.

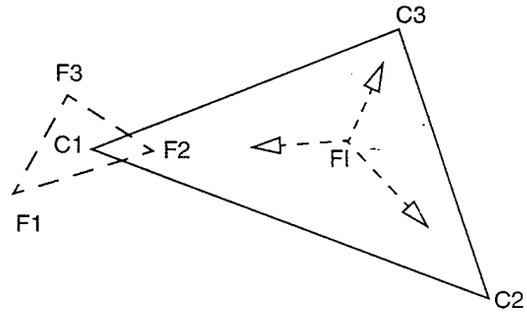

Fig. 1: Illustration of conservative residual restriction.

In order to locate a particular cell which encloses a particular node, one must search over the grid cells. A search over the entire grid requires $O(N^2)$ operations, where N is the number of vertices of the grid. For large grids it will require an extensive amount of CPU-time. For this reason the ADT algorithm[12], developed for our grid generator, is applied to find the cell that encloses a particular vertex.

On strictly concave boundaries, the boundary vertices will be exterior or coincide with vertices in the interpolation grid. The vertex P in Figure 2 will not lie inside any of the searched grid cells. Therefore the distances to all boundary grid cells, at the same boundary in the interpolating grid are computed. The boundary grid cell closest to the vertex P is used in the interpolation.

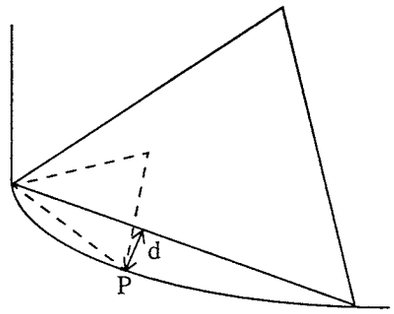

Fig. 2: Computation of the distance to a boundary grid cell.

The interpolation tables are computed prior to the flow computations. With the described method, the total time required for the computation of the interpolation coefficients constitutes a small fraction of the CPU-time required for the flow computations.

4.3 Multigrid time stepping.

The time stepping scheme plays an important role in the multigrid algorithm in damping high frequency error modes. A five-stage hybrid time stepping scheme, is used to drive the multigrid algorithm. The time stepping scheme is given by,

$$w^{(j)} = w^{(0)} - \alpha_j \frac{\Delta t}{\Omega}(C(w^{(j-1)}) - D_{j-1}) \qquad (9)$$

, where j = 1, 2,..., 5. C is the convective operator and D is the dissipative operator updated according to,

$$\begin{aligned} D_0 &= D_1 = D(w^{(0)}) \\ D_2 &= D_3 = \beta D(w^{(2)}) + (1-\beta)D_0 \\ D_4 &= \gamma D(w^{(4)}) + (1-\gamma)D_2 \end{aligned} \qquad (10)$$

The dissipative operator D(w) is evaluated only at the first, third and fifth stages of the scheme and is employed to construct the D_i operator, which is a linear combination of present and previous evaluations of D(w). The values of the coefficients are chosen according to Jameson and Mavriplis[15], $\beta = 0.56$, $\gamma = 0.44$ and α_1=1/4, α_2=1/6, α_3=3/8, α_4=1/2 and α_5=1/4.

The forcing function in Equation (7) can be written as,

$$P = R' - R(w') \qquad (11)$$

, where R' is the transferred residual and w' is the transferred solution. On the coarse grids the time stepping proceeds as follows

$$w^{(j)} = w^{(0)} - \alpha_j \Delta t(R(w^{(j-1)}) + P) \quad . \qquad (12)$$

In the first stage the calculated residuals on the coarse grid are cancelled by the second term in the forcing function P leaving only the R' term.

4.4 Coarse grid dissipation

For overset and agglomeration multigrid techniques it is inevitable with a degradation of the grid quality for coarser grid levels. In order to achieve a good multigrid efficiency, stable coarse grid operators are required. A way to ensure a more stable coarse grid operator is, according to Venkatakrishnan and Mavriplis[23], to formulate a first order Roe's upwind solver on the coarser grid levels. However, it was found that the Roe's upwind solver is considerably more expensive than the central difference scheme.

An alternative is to formulate a first order central difference scheme as a coarse grid operator. If the biharmonic operator in (3) is omitted and the pressure switch in front of the harmonic operator is replaced with a constant $\varepsilon^{(0)}$

$$D(q_i) = \sum_{k=1}^{m_i} \varepsilon^{(0)}(q_k - q_i)\frac{(\lambda_k + \lambda_i)}{2} \qquad (13)$$

, a new scheme that is formally first order accurate is obtained. The constant $\varepsilon^{(0)}$ is for the computations presented in this work set to 0.02.

This coarse grid operator in Equation (13) is found to significantly improve the robustness of the multigrid scheme. In most cases it allowed an increase of the CFL-value enabling speedups in CPU-time up to 50%.

In some cases we have also been using a lower CFL-value on the coarser grid levels.

4.5 Smoothing of corrections

The robustness of an overall multigrid scheme can, according to Radespiel and Swanson [19], be improved by smoothing the coarse grid corrections before they are passed on to the finest grid. The smoothing reduces high-frequency oscillations introduced by the linear interpolation of the coarse grid corrections.

In an explicit scheme, solution updates are directly proportional to the computed residuals. It therefore

seems reasonable to use the procedure for residual smoothing to smooth the corrections.

Implicit residual smoothing equation for unstructured grids implies that we solve the smoothing equation,

$$(1+n_k)\overline{R}_i - \varepsilon \sum_{k=1}^{n_k} \overline{R}_k = R_i, \qquad (14)$$

where R is the initial residual and $\overline{R}$ is the smoothed residual. The system of equations (14) is strongly diagonally dominant and can therefore be solved by Jacobi iterations. For residual smoothing, two Jacobi iterations is found to give sufficient accuracy.

The influence of number of iterations for smoothing of the corrections on the convergence rate is depicted in Figure 3. The AGARD01 test case for a NACA0012 profile is computed with a 4 level V-cycle for different number of smoothing iterations. In Figure 3 it can be seen that one smoothing iteration is sufficient to smooth the high-frequency oscillations of the corrections. Without smoothing of the corrections this case did not converge unless the CFL number is lowered.

The value of the coefficient ε is in all computations presented in this paper set to 0.2.

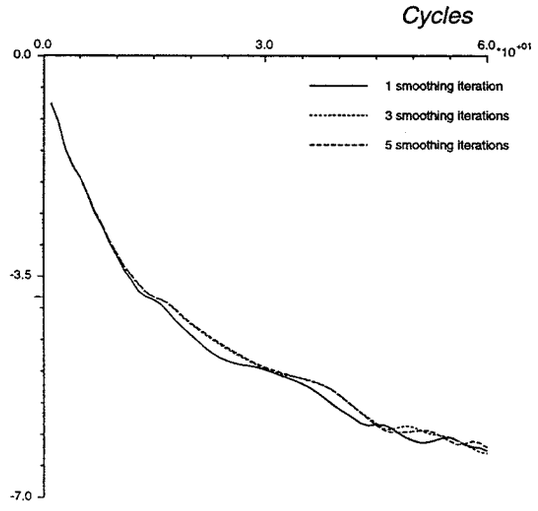

Fig. 3: The influence of the number of smoothing iterations for the corrections on the convergence rate. The computations are performed for a 4-level V-cycle for the AGARD01 case for a NACA0012 airfoil.

5 Test Cases

In the test examples presented here, a multigrid V-cycle is applied. The number of pre- and post-smoothings used on the various levels is shown in Table 1 .

The computational cost for the V-cycle is 2.20 work units in 2D an 1.54 work units in 3D, where one work unit is the computational work required for a single grid computation on the finest grid level.

Level	Pre-smooth.	Post-smooth.
2	2	1
3	3	1
4	4	

Table 1: Pre- and post-smoothings for the 4-level V-cycle.

In the convergence diagrams the L_2 norm of the density residual is plotted as a function of both the number of multigrid cycles or work units.

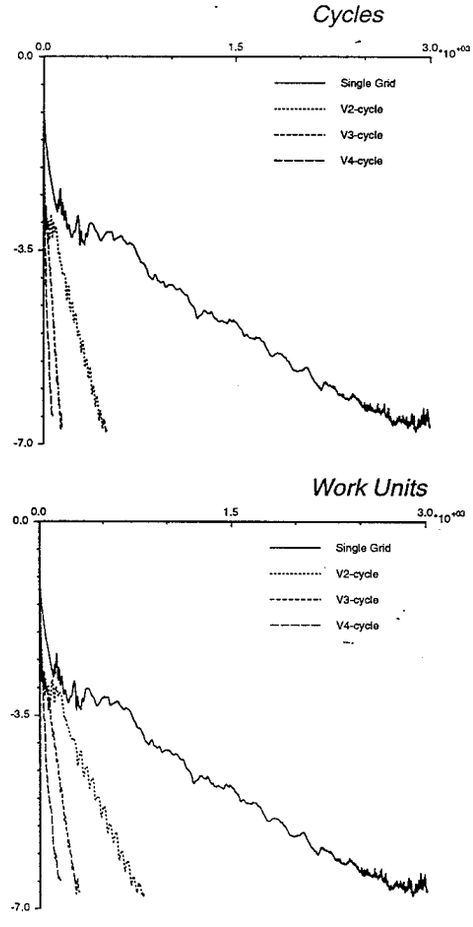

Fig. 4: Convergence rates for different number of grid levels of the multigrid V-cycle for the NACA0012 airfoil. The convergence rates are plotted versus a. the number of cycles and b. as work units.

5.1 Single element airfoil

Flow past a NACA0012 airfoil at a free stream Mach

number of 0.8 and an angle of attack of 1.25°, which is known as the AGARD01 case, is calculated. A sequence of four grid levels around the NACA0012-profile is applied, see Figure 7. The finest grid contains 21224 vertices.

The Cp-distribution in Figure 8 is compared with a solution on a structured grid obtained by the AGARD working group 07 [20]. The structured grid contained 23040 vertices, i.e. approximately the same number as the unstructured grid. The agreement is good everywhere except in the vicinity of the shocks where the resolution is somewhat better for the unstructured grid solution. The absence of oscillations in the vicinity of the shock and the sharpness of the shock demonstrates the efficiency of the artificial dissipation operator in providing smooth solutions.

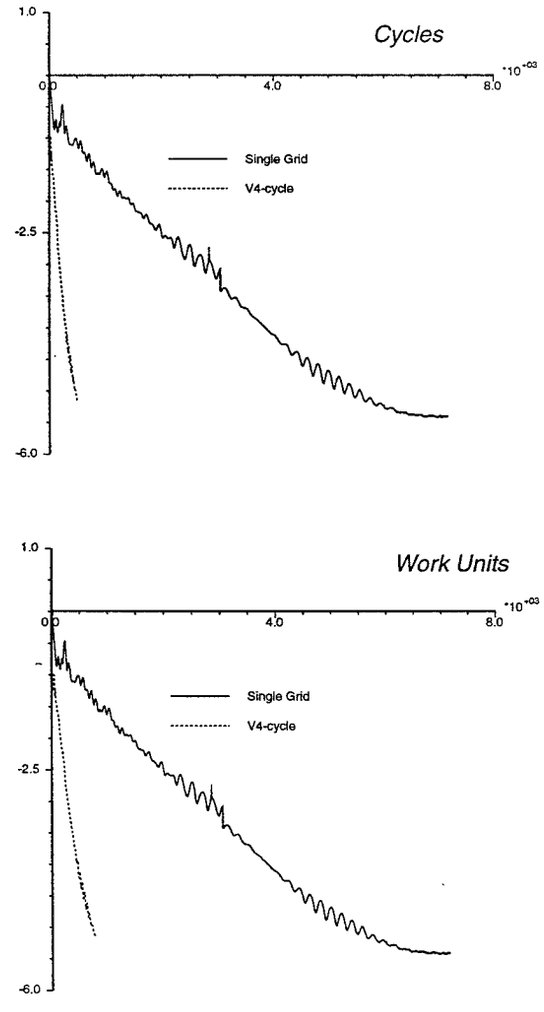

Fig. 5: Convergence rates with and without multigrid for computation of transonic flow over a transport wing. The convergence rates are plotted versus a. the number of cycles and b. the number of work units. M=0.5, α=2.2°.

Multigrid convergence rates for the V-cycle with different number of grid levels are depicted in Figure 4. Already two levels of multigrid speeds up the convergence a factor 8 in iterations and a factor 3.7 in work units. The speedup for four multigrid levels is 37 in terms of iterations and 17 in terms of work units. In this case convergence to machine accuracy is reached within 75 multigrid cycles.

5.2 A Transport Wing

The multigrid algorithm is also tested for flow around a transport wing. A transonic case with the free stream Mach number 0.8 and the angle of attack 2.2° is computed. At these conditions the flow is transonic and a λ-shaped shock is formed over the wing. The grid consisted of 483979 vertices. A sequence of four grid levels is generated, Figure 9. The grid has been adapted to gradients in the flow in three steps. The Mach-contours are depicted in Figure 10. The multigrid convergence rate for this case is depicted in Figure 5. The multigrid speeds up the computation a factor 11.6 in terms of iterations and a factor 7.5 in terms of work units.

5.3 Fighter configuration

A transonic flow case past the Swedish fighter airplane JAS 39 Gripen configuration is computed. The grid consisted of 477 525 grid points and a sequence of four grid levels is generated, Figure 11. The grid has been adapted in three steps to gradients in the flow solution.

The free stream Mach number is 0.95 and the angle of attack is 6.4°. The specified inlet mass flux ratio at the air intake C_A is 0.7. The flow at the air intake is subsonic, which means that one characteristic variable propagates into the computational domain, i. e. one boundary condition should be applied. In this case it is natural to specify the normal velocity since the average normal velocity is known, see [22]. We assume that the velocity profile is uniform.

Mach contours on the configuration surface are depicted in Figure 12. The solution contains crisp shocks on the canard, the delta wing and behind the canopy. This demonstrates that the grid resolves relevant flow features of the solution.

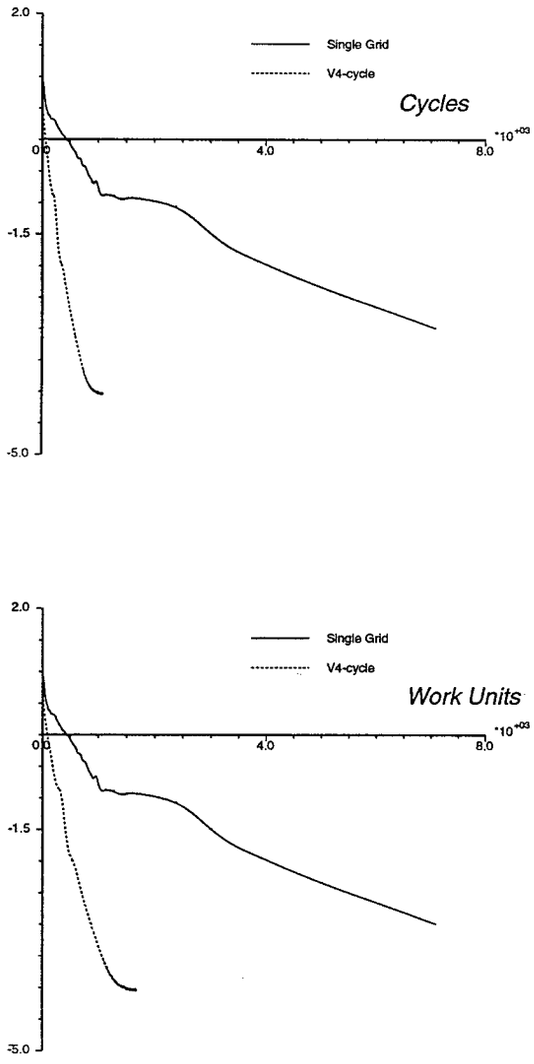

Fig. 6: Convergence rates with and without multigrid for computation of transonic flow over the JAS 39 Gripen configuration at M=0.95 and α = 6.4°. The convergence plotted versus a. the number of cycles and b. the number of work units.

The multigrid speeds up the computation a factor 12.5 in terms of iterations and a factor 8 in terms of work units, Figure 6.

6 Summary and Conclusions

This work demonstrates the efficiency of multigrid convergence acceleration technique for an overset grid method. The speedup in CPU-time due to multigrid alone is more than an order of magnitude in 2D and somewhat less than an order of magnitude in 3D. A new alternative for a coarse grid operator, for which the dissipation is modified such that a first order scheme is applied on the coarser grid levels, is proposed. In most cases it enables the use of higher CFL-numbers.

7 Acknowledgments

The authors are grateful to our colleague Petter Eneroth for generating the grids and making the pictures of the JAS computation. We are also grateful to NFFP, the National Aeronautical Research Program of Sweden, for sponsoring this project.

8 References

[1]Ni, R. H., "A multiple Grid Scheme for Solving the Euler Equations", AIAA Journal, Vol 20,1982, pp. 1565-1571.

[2]Jameson, A., "Solution of the Euler Equations by a Multigrid method", Applied Mathematics and Computation, Vol 13, pp. 327-356.

[3]Löhner, R. and Morgan, K., "Unstructured Multigrid Methods", 2nd European Conf. on Multigrid Methods, Cologne, FRG, Oct. 1985.

[4]Mavriplis D. J.,"Multigrid Solution of the 2-D Euler Equations on Unstructured Triangular Meshes", AIAA Journal, Vol. 26 NO. 7 April 1988, pp 824-831.

[5]Guillard, H., "Node nested multigrid with Delauney Coarsening. INRIA Report No. 1898, 1993.

[6]Chan, T., F. and Smith, B., "Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes", UCLA CAM Report 93-42, Dept of Mathematics, University of California, Los Angeles, December 1993.

[7]Olliver-Gooch, C., F. "Multigrid Acceleration of an Upwind Euler Solver on Unstructured Grids", AIAA Journal Vol. 33, No. 10, October 1995.

[8]Lallemand M., Steve, H. and Dervieux, A., "Unstructured Multigridding by Volume Agglomeration: Current status", Computers and Fluids, 21(3):397-433, 1992.

[9]Berglind T., "Finite Volume Solution of 2D Euler Equations on Unstructured Triangular Grids, FFA TN 1994-17, FFA, 1996.

[10]Berglind T., "Multigrid Solution of the Euler Equations on Triangular and Tetrahedral Grids", FFA TN 1997-27, FFA, 1997.

[11]Tysell L. G.," An advancing Front Grid Generation System for 3D Unstructured Grids", Proceedings of the 19th ICAS Congress, 1994, ICAS-94-2.5.1, Anaheim, CA.

[12]Bonet J. and Periare J., "An Alternating Digital Tree(ADT) Algorithm for 3D Geometric Searching and Intersection Problems", International Journal for Numerical Engineering, Vol. 31, 1991.

[13]Jameson, A.,Schmidt, W. and Turkel, E. "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes", AIAA Paper 81-1259, AIAA 14th Fluid Dynamics and Plasma Dynamics Conference, Palo Alto, 1981.

[14]Jameson, A.,Mavriplis, D., "Two Dimensional Euler Equations on a Regular Triangular Mesh",AIAA Paper 83-0435, AIAA 23th Aerospace Sciences Meeting, Reno, NV, 1985.

[15]Mavriplis D. J. and Jameson A., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes", *AIAA Journal*, Vol. 28 NO. 8 August 1990, pp 1415-1425.

[16]Barth, T. J. and Jespersen, D., "The Design and Application of Upwind Schemes on Unstructured Meshes", AIAA Paper 89-0366, Jan. 1989.

[17]Venkatakrishnan V. and Mavriplis D. J., "Agglomeration Multigrid for the Three-dimensional Euler Equations", AIAA Journal, Vol. 33 NO. 4, April 1995, pp 633-640.

[18]Mavriplis D. J, "Multigrid Techniques for Unstructured Meshes", in von Karman Institute Lecture Series Computational Fluid Dynamics, 1994.

[19]Radespiel R., Swanson R. C., "Progress with multigrid schemes for hypersonic flow problems", NASA Contractor Report 189579, ICASE Report No.91-89.

[20]AGARD Advisory Report No. 211, "Test Cases for Inviscid Flow Field Methods", AGARD-AR-211.

[21]Thibert J.J., "The Garteur High Lift Research Program", ONERA, 92322 Chatillon , France.

[22]Berglind T., "Flow Simulation around a Realistic Fighter Airplane Configuration", Proceedings of the 17th Congress of ICAS 1990, ICAS-90-6.4.1, Stochholm, Sweden, 1990.

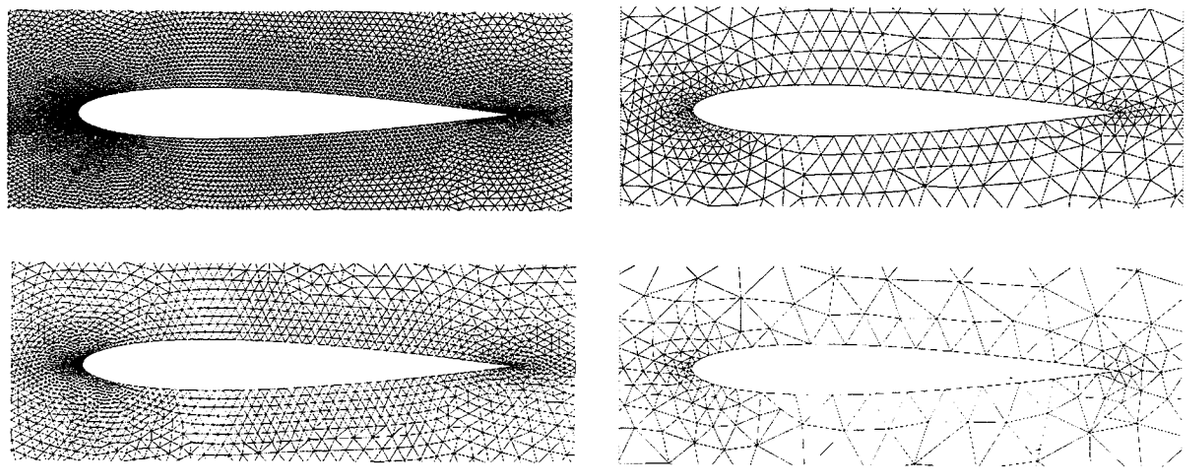

Fig. 7: A sequence of grids employed to compute transonic flow over a NACA0012 profile. The finest grid contains 21224 grid points and the coarsening ratio for the subsequent grids is approximately 1:4.

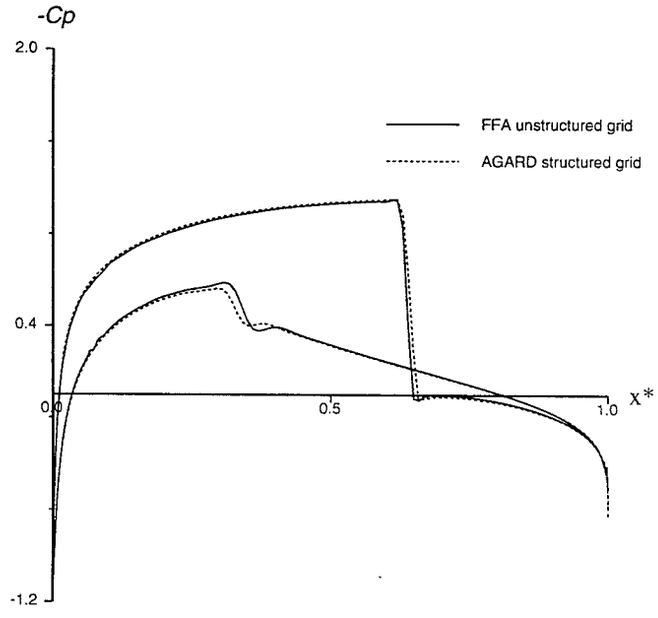

Fig. 8: Cp-distribution on the NACA0012 profile for the AGARD01 test case (M=0.80, α=1.25°).

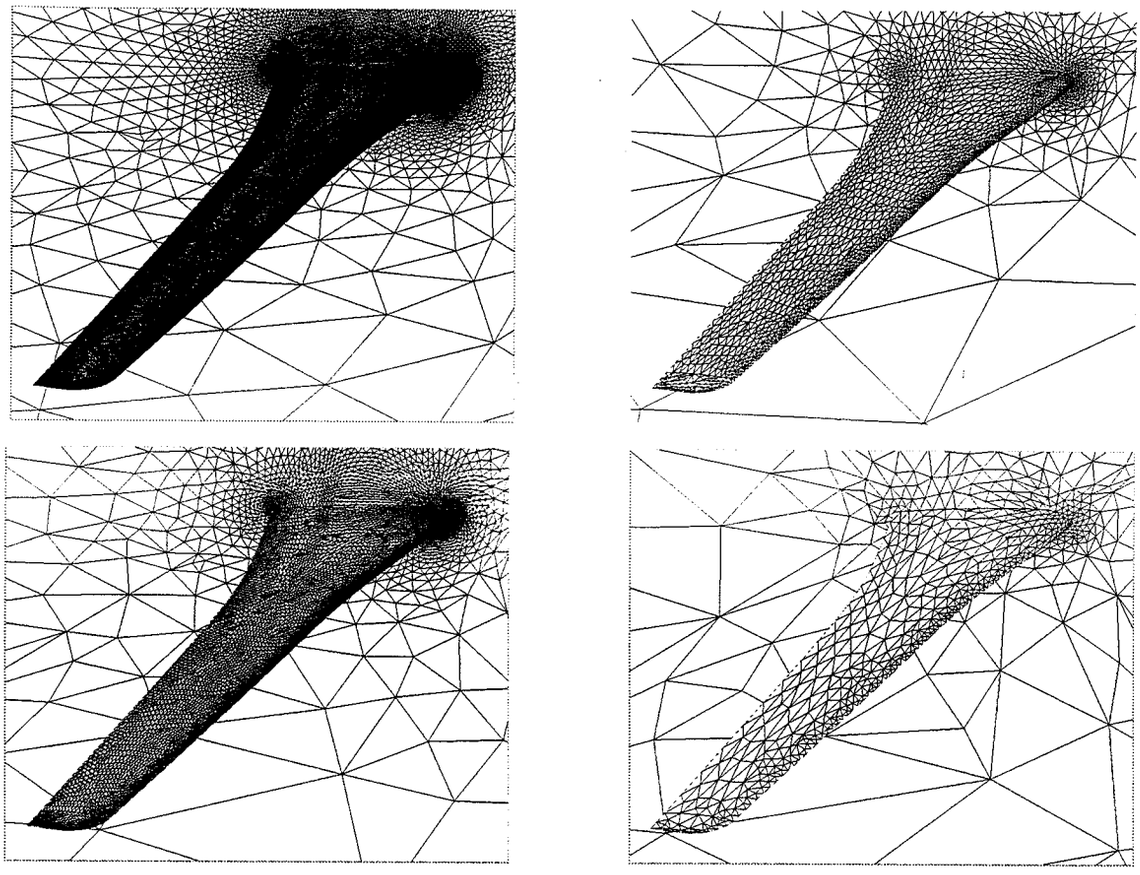

Fig. 9: A sequence of grids employed to compute flow over a transport wing. The finest grid contains 483979 vertices and the coarsening ratio between the subsequent grid levels is approximately 1:8.

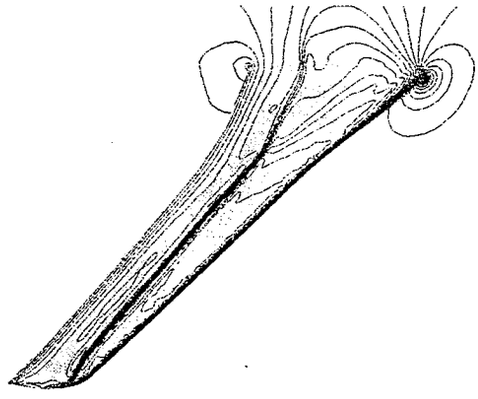

Fig. 10: Mach-contours for computed flow over a transport wing configuration at M=0.8 and $\alpha = 2.2^{o}$. $\Delta M = 0.05$.

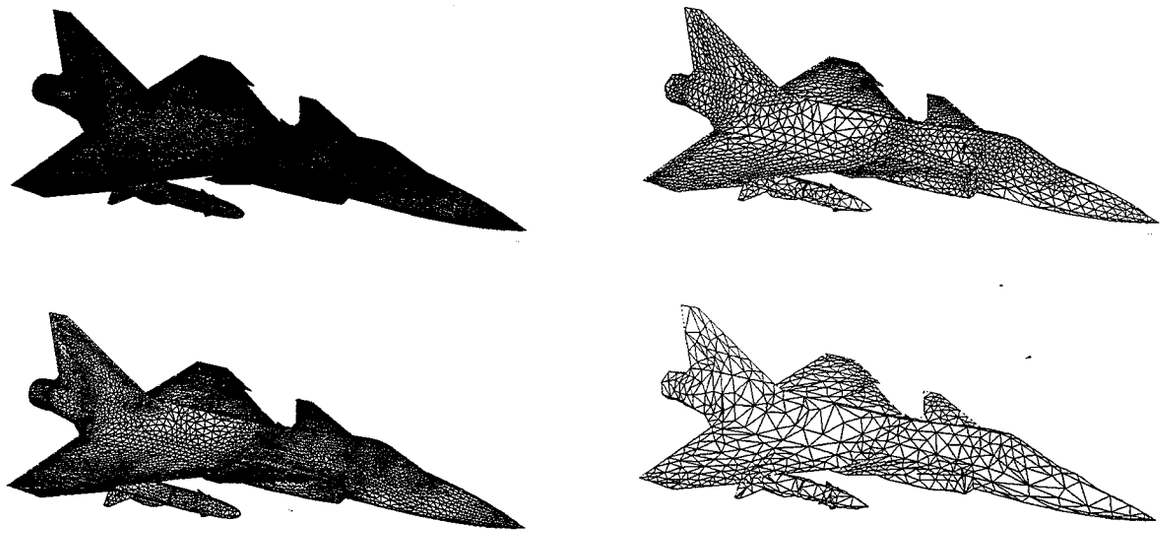

Fig. 11: A sequence of grids employed to compute transonic flow over the JAS 39 Gripen configuration. The finest grid contains 477 525 vertices and coarsening factor between the subsequent grid the levels is approximately 1:8.

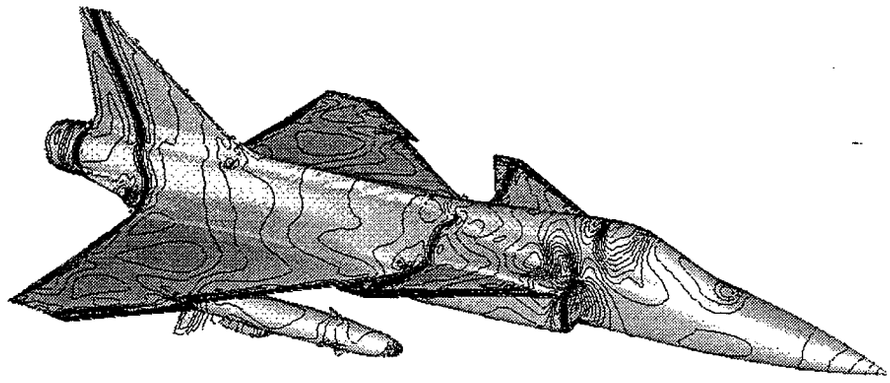

Fig. 12: Cp-distribution for computed flow about the JAS 39 Gripen configuration at M=0.95 and $\alpha = 6.4^{o}$. $\Delta Cp = 0.05$.