

Studies in multigrid acceleration of some equations of interest in fluid dynamics¹

J P SINGH

Flosolver Unit, CTFD Division, National Aerospace Laboratories,
Bangalore 560 017, India

MS received 28 April 1995; revised 12 September 1995

Abstract. The paper describes the multigrid acceleration technique to compute numerical solutions of three equations of common fluid mechanical interest; Laplace equation, transonic full potential equation and Reynolds averaged Navier–Stokes equations. Starting with the simple and illustrative multigrid studies on the Laplace equation, the paper discusses its application to the cases of full potential equation and the Navier–Stokes equations. The paper also discusses some elements of multigrid strategies like V- and W-cycles, their relative efficiencies, the effect of number of grid levels on the convergence rate and the large CPU time saving obtained from the multigrid acceleration. A few computed cases of transonic flows past airfoils using the full potential equations and the Navier–Stokes equations are presented. A comparison of these results with the experimental data shows good agreement of pressure distribution and skin friction. With the greatly accelerated multigrid convergence, the full potential code typically takes about 10 seconds and the Navier–Stokes code for turbulent flows takes about 5 to 15 min of CPU time on the Convex 3820 computer on a mesh which resolves the flow quantities to good levels of accuracy. This low CPU time demand, made possible due to multigrid acceleration, on one hand, and the robustness and accuracy on the other, offers these codes as designer's tools for evaluating the characteristics of the airfoils.

Keywords. Multigrid; full-potential solution; Navier–Stokes solution; Laplace equation.

¹ Various parts of this paper have been presented at the following conferences; (i) 5th Asian Cong. on Fluid Mech., Taejon, Korea, 1992, (ii) Int. Conf. on Methods of Aerophysical Research, Novosibirsk, 1992, (iii) Fluid Dyn. Symp. in honour of Prof. R Narasimha on his 60th birthday, 1993.

1. Introduction

Starting with the pioneering work of Federenko (1964), the theory and application of multigrid has come of age. The large time saving due to the multigrid acceleration has offered great relief to the workers in many areas of science and technology, especially in the developing countries where multi-megaflop computing machines are not yet common. Even today at many of these institutions, a typical 2-D turbulent Navier–Stokes solution on a grid size of, say, 385×65 , requires 2 to 3 days of turn-around time. The multigrid technique has also provided an opportunity or potential to several codes to be used as designer's tool. The combined effect of the high speed computing machines and the multigrid accelerated codes has led to such realistic design computations that a well converged 3-D Navier–Stokes flow computation past a wing-body-nacelle configuration can be obtained in as little as 4 min of CPU time (Jameson 1993) on some of the high speed computers. Historically, the first publication on multigrid method put to practice was brought out by Brandt (1972) followed by another publication (Brandt 1977) five years later with convincing results which set the pace for rapid progress in multigrid methods and applications (Wesseling 1990). From the same historical angle, it may be mentioned that Hackbusch (1976) (as mentioned in Wesseling 1990) also independently discovered the multigrid method. Today, the multigrid method has crossed its domain of application to elliptic systems for which basically its theory was developed. One can find large scale application of multigrid in solving the time dependent Navier–Stokes equations and several other hyperbolic problems. During the last few years, multigrid has also made inroads in the areas of hydrodynamic stability, integral and integro-differential equations, many-body interactions, semiconductor device simulation, general relativity, theory of material defects and atomic physics, real time optimal control, protein folding, combinatorial optimisation, linear programming etc. (Brandt 1988).

In CFD, one seeks converged solution to viscous flow problems on a grid which is fine enough to resolve all the length scales involved. In most of the cases, the actual number of node points thus obtained is quite large leading to a large number of algebraic equations to be solved resulting from the discretization. Also, most of the relaxation schemes are efficient at smoothing out the high frequency errors as seen on the scale of the grid. The solution on a single fine grid is too expensive for two reasons. (i) due to the large number of algebraic equations resulting from the discretization process, (ii) due to the relaxation schemes taking very long time to smooth out the low frequency errors. Quoting a concept from Brandt (1977), the multigrid method uses a sequence of grids “taking advantage of the relation between different discretizations of the same problem. This method can be viewed in two complementary ways: one is to view the coarse grids as the correction grids, accelerating convergence of a relaxation scheme on the finest grid by efficiently liquidating smooth error components. Another point of view is to regard finer grids as the correction grids by correcting their forcing terms”. The first view of Brandt (1977), which is an early and monumental work on the multigrid methods, refers to the definition of high frequency components with respect to each grid which are smoothed fastest by most relaxation schemes. The second view offers the possibility to “manipulate accurate solutions on coarser grids, with only infrequent ‘visits’ to pieces of finer levels”.

The next section gives a brief description of the basic elements of multigrid technique, followed in the present work, to get accelerated convergence of (a) solution of Laplace

equation on Cartesian grids, (b) solution of transonic potential flow equations, and (c) solution of Navier–Stokes equations for 2-D transonic flow.

2. A brief description of multigrid method

Let $G^h, G^{2h}, \dots, G^{Mh}$ be a hierarchy of grids with h indicating the finest level. Let the equation to be solved be given as

$$L\Phi = f. \tag{1}$$

The discretized form of (1) on grid level h is written as;

$$L^h\Phi^h = f^h. \tag{2}$$

If ϕ is an approximate solution of Φ and ψ is the correction term, then

$$L^h(\phi^h + \psi^h) = f^h, \tag{3}$$

or

$$L^h(\phi^h + \psi^h) - L^h\phi^h = f^h - L^h\phi^h = \mathcal{R}^h. \tag{4}$$

\mathcal{R}^h is known as residual on grid G^h . A certain number of relaxation sweeps are made to obtain ϕ^h . Ideally, when the convergence slows down, the relaxation on the present level is stopped and the solution and the residual is injected down to the next coarser level $2h$ where one can write (4) for this grid as follows;

$$L^{2h}(\mathcal{I}_h^{2h}(\phi^h + \psi^h)) - L^{2h}\mathcal{I}_h^{2h}\phi^h = \tilde{\mathcal{I}}_h^{2h}\mathcal{R}^h, \tag{5}$$

where \mathcal{I}_h^{2h} and $\tilde{\mathcal{I}}_h^{2h}$ are injection operators which need not be the same. Equation (5) is rewritten as;

$$\begin{aligned} L^{2h}\hat{\phi}^{2h} &= \tilde{\mathcal{I}}_h^{2h}\mathcal{R}^h + L^{2h}\mathcal{I}_h^{2h}\phi^h \\ &= \tilde{\mathcal{I}}_h^{2h}(f^h) + (L^{2h}\mathcal{I}_h^{2h}\phi^h - \mathcal{I}_h^{2h}(L^h\phi^h)), \\ &= \tilde{\mathcal{I}}_h^{2h}(f^h) + f^{2h}, \end{aligned} \tag{6}$$

where

$$\hat{\phi}^{2h} = \mathcal{I}_h^{2h}(\phi^h + \psi^h). \tag{7}$$

f^{2h} is the truncation error of the coarse grid relative to the fine grid. It is seen that without the addition of the term f^{2h} in (6), the discretized coarse grid equation has the same form as (2) for fine grid. In fact, when the solution has converged, the coarse grid solution will be same as the injected fine grid solution and f^{2h} will vanish. In that sense, the term f^{2h} , also known as forcing function, is identified as the ‘‘fine to coarse grid defect correction’’. This term can also be seen to represent the interplay between the discretization and the solution process.

This procedure is carried on till the coarsest grid is reached. The corrected coarse grid solution thus obtained needs to be transferred to the fine grid to obtain the updated fine grid solution. This step is written as;

$$\phi_{\text{new}}^h = \phi_{\text{old}}^h + \tilde{\mathcal{I}}_{2h}^h(\phi^{2h} - \mathcal{I}_h^{2h}\phi^h), \tag{8}$$

where $\tilde{\mathcal{I}}$ is a suitable interpolation operator. Note that if the coarse grid solution was simply prolonged to the fine grid without adding correction term as above, it would be devoid of all the high frequency components of the solution and also it would

contain interpolation error. In this sense, the process has taken the best advantage of the high frequency smoothing property of the chosen relaxation scheme on each level and a lot of CPU time is saved by working on coarser grid levels.

3. Application to Laplace equation

The Laplace and Poisson equations (differing only by the absence or the presence of the nonhomogeneous term) are of fundamental importance to fluid dynamics. The Laplace equation occurs as the main governing equation for the incompressible, inviscid and irrotational fluid flow, and the Poisson equation occurs as a component equation in case of incompressible Navier–Stokes flows. Also, the simplicity of this equation makes it a very illustrative platform.

Since the Laplace equation is linear, the operator L in (3) can be linearly distributed over ϕ and ψ and one can proceed to solve directly for correction ψ using what is called the “Correction Scheme”. But in this work, a more general method called Full Approximation Storage (FAS) scheme, as applicable to the nonlinear equations, is used.

Here the interest is to solve the Laplace equation using multigrid acceleration. In this section, the work of Singh (1989) for multigrid strategy will be followed. The Laplace equation is written as:

$$\nabla^2 \phi = 0. \quad (9)$$

The boundary conditions are taken to be as follows; at (i) $y = 0$, $\phi = 0$, (ii) $y = \pi$, $\phi = 0$, (iii) $x = 0$, $\phi = \sin(y)$, and (iv) $x = 1$, $\phi = e \cdot \sin(y)$. The problem is discretized over a domain given by $0 \leq x \leq 1$ and $0 \leq y \leq \pi$ using the Cartesian grid.

The problem being purely elliptic, the equation is discretized by writing the second derivatives using the central difference scheme. The resulting difference equations are solved by successive line relaxation technique. A V-cycle FAS multigrid strategy with independently selected number of sweeps at various grid levels will be used. The results are presented for grid levels up to six.

Figure 1 shows several convergence histories of the solution procedure as a function of number of grid levels used. In each case the solution is sought on the finest grid level of size 65×129 . The figure shows the effect of number of grid levels on the multigrid convergence. The gain in terms of speed of convergence rapidly increases as the number of grid levels is increased from 1 to 4. The gain slows down when 5 grid levels are used. Beyond this, one can see that there is just about no gain in the rate of convergence at all as the number of grid levels is increased from 5 to 6. The results of this study are summarised in table 1 below.

Table 1 shows the level of error obtained using one-level computation and multilevel computations using 2 to 6 grid levels after 100 work units. L_2 (norm) stands for the absolute value of the L_2 -norm of the residual. The drastic gain in the solution convergence is clear. One can see that after 100 work units, the 5-grid level solution has converged to an accuracy of about 14 orders lower than the corresponding convergence on a single level. Of course, it is a different matter that, in practical applications, one never needs to drive convergence to more than, say, 3 to 4 orders of magnitude which is obtained here in far too fewer work units. The basic nature of this observation is true in general for all systems, i.e., the law of diminishing return applies as the number

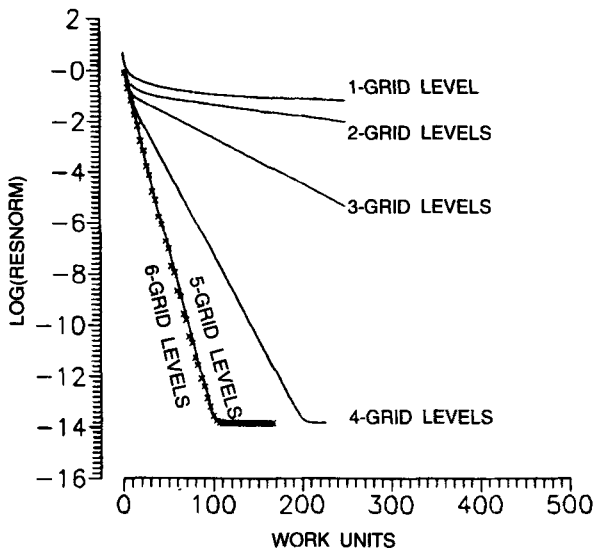


Figure 1. Multigrid convergence and effect of number of grid levels.

of coarser meshes beyond optimum number is increased. Often four to five grid levels are optimum. Of course, as far as the level to which the solution converges after certain number of work units is concerned, it depends on the system being solved (nonlinearity, constraints, presence of discontinuities etc. tend to degrade the multigrid performance). The next two examples of full potential equation and Navier–Stokes solution will elucidate this point.

3.1 *Effect of boundary condition “defect” on multigrid convergence*

In the above, the Laplace equation has been solved with the exact boundary condition. This does not involve any approximation of the boundary values. However, when a problem is solved which involves approximation (and hence error) in the boundary values, that we call “boundary condition defect” here, the performance of the same multigrid accelerated iterative procedure is impaired significantly. To study this, we consider a computationally similar problem (Singh 1992) that involves extrapolation to compute boundary condition and hence, introduces error or “boundary condition defect”. To see this, we present the results of the following demonstration computations, Study-1: The solution of Laplace equation with rectangular boundary and exact boundary conditions for comparison of convergence history, Study-2: the problem in study-1 is recomputed using the difference form of the Neumann boundary condition $\partial\phi/\partial y = \exp(x)\cos(y)$ as the boundary condition on the bottom edge. The truncation

Table 1. Efficiency of multigrid scheme with respect to the number of grid levels used.

	1-level	2-level	3-level	4-level	5-level	6-level
Abs. value of L2 (norm) after 100 work units	8.28	4.7×10^{-2}	2.5×10^{-3}	6.3×10^{-8}	2.8×10^{-14}	2.8×10^{-14}

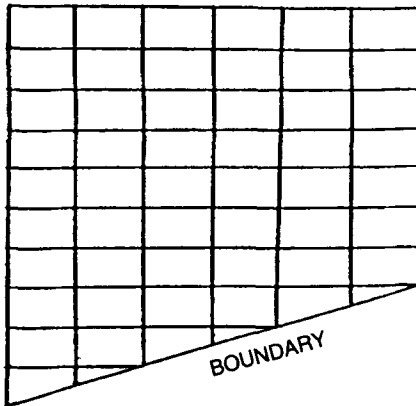


Figure 2. Computational domain for Studies 3 and -4.

error of the difference scheme introduces some error, Study-3: The problem in Study-1 is solved with the following modification; the bottom edge is cut slant as shown in figure 2. The boundary condition implementation at this edge requires values of ϕ at the dummy points for which the exact values are provided. The boundary conditions on the left, right and top edges remain the same, Study-4: all features identical to Study-3 except that the values at the bottom boundary point were obtained using Taylor series expansion as described below. This adds further to the errors.

The value of ϕ at the slant bottom boundary is written as follows by opening a Taylor series around $(i, j - 1)$;

$$(\phi)_{y_b} = (\phi)_{y_{i,j-1}} + (y_b - y_{i,j-1})(\phi)_{yy_{i,j-1}}, \quad (10)$$

where

$$(\phi)_{y_{i,j-1}} = (-3(\phi)_{i,j-1} + 4(\phi)_{i,j} - (\phi)_{i,j+1})/2\Delta y \quad (11)$$

and

$$(\phi)_{yy_{i,j-1}} = ((\phi)_{i,j-1} - 2(\phi)_{i,j} + (\phi)_{i,j+1})/\Delta y^2. \quad (12)$$

The subscript 'b' indicates values on the slant boundary. The values of $(\phi)_{y_b}$ are provided exactly to compute $(\phi)_{i,j-1}$. The higher level of truncation error introduced artificially is obvious. Figure 3 shows the comparison of the convergence histories of the above studies. The study illustrates the level of stagnation residues in various studies. The residue level ($\approx 10^{-7}$ in 32-bit arithmetics, $\approx 10^{-15}$ in 64-bit arithmetics) is the lowest in Study-1 where no "defect" is involved in the boundary condition implementation. Study-2 has relatively higher level of stagnation residue ($\approx 10^{-6}$ in 32-bit arithmetics) where there is a relatively smaller amount of boundary condition "defect". Even though we have considered a slant bottom edge in Study-3, which requires use of the dummy point, the stagnation residue level is the same as that in Study-1 because the value there is provided exactly. Study-4, which has the highest level of "defect", leads to the highest level of stagnation residue ($\approx 10^{-4}$ in 64-bit arithmetics).

The effect of the boundary condition "defect" can also be seen by explicitly adding a perturbation term to the exact boundary condition in Study-1. Let the perturbation term to the boundary condition be chosen as $A \sin(n\pi x)$ where $n = 1, 2, 3, \dots$ with $0 \leq x \leq 1$. The term is explicitly added to the boundary condition on the bottom edge in Study-1. One can infer from figure 4 that as the amplitude A increases, the magnitude of the boundary condition "defect" increases, which leads to convergence process stagnating

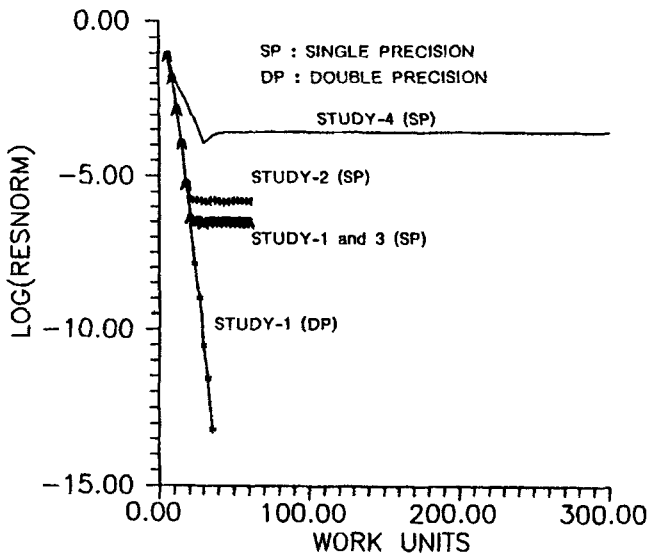


Figure 3. Comparison of convergence rates for studies 1 to 4.

at higher levels. The above study illustrates the sensitivity of the convergence history in the multigrid process as it is affected by the boundary condition "defect".

4. Application to full potential transonic flow computation

Even today, most of the CFD codes that are used in the preliminary aircraft aerodynamic design are based on simpler methods like panel method and full potential

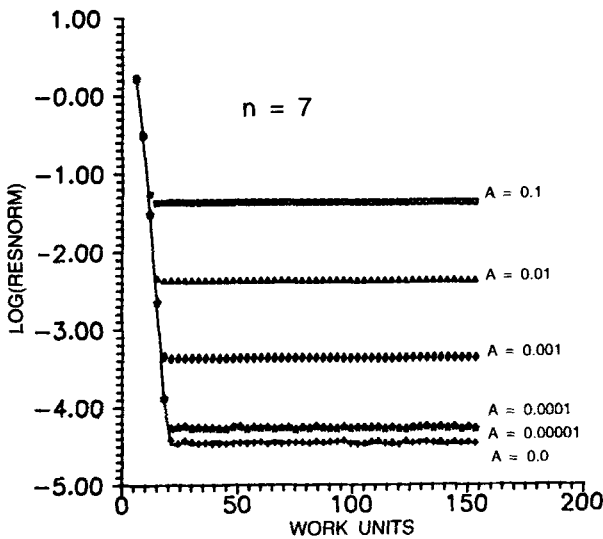


Figure 4. Effect of boundary condition "defect" on multigrid convergence.

equations. Thus, it is useful to have these codes accelerated using techniques like multigrid. In this section, the implementation of multigrid acceleration in a full potential transonic flow solver due to Garabedian & Korn (1971) is described. Historically, the first successful application of the multigrid acceleration to potential function based formulation for the transonic flow was due to South & Brandt (1977). In their work, they presented the accelerated solution of small perturbation equation for non-lifting flow past airfoil. Singh (1989) used the multigrid acceleration to get solutions for flow past airfoils using the code of Garabedian & Korn (1971) modified and adapted by him at NAL. An acceleration of about 5 to 7 times (in terms of work units) was obtained using the multigrid strategy described. The basic method of Garabedian & Korn (1971) is described below.

The 2-D full potential flow equation in Cartesian coordinates is;

$$(c^2 - u^2)\phi_{xx} - 2uv\phi_{xy} + (c^2 - v^2)\phi_{yy} = 0, \quad (13)$$

where ϕ is the velocity potential, u, v the velocity components in x and y directions, and c the local speed of sound given by the Bernoulli's law

$$(u^2 + v^2)/2 + c^2/(\gamma - 1) = (1/2) + M^2/(\gamma - 1). \quad (14)$$

Using Sell's conformal mapping, the entire infinite flow field is mapped on to the inner circle of unit radius such that the perimeter of the circle corresponds to the airfoil contour and the centre corresponds to infinity. In this plane, using (r, θ) coordinates, the full potential equation, (13), is written as;

$$\begin{aligned} & (c^2 - r^2 f^{-2} \phi_\theta^2)\phi_{\theta\theta} - 2r^4 f^{-2} \phi_\theta \phi_r \phi_{\theta r} + r^2(c^2 - r^4 f^{-2} \phi_r^2)\phi_{rr} \\ & + r(c^2 - r^2 f^{-2} \phi_\theta^2 - 2r^4 f^{-2} \phi_r^2)\phi_r + f^{-3}(r^2 \phi_\theta^2 + r^4 \phi_r^2) \\ & (f_\theta \phi_\theta + r^2 f_r \phi_r) = 0, \end{aligned} \quad (15)$$

where $f = r^2 |F'(re^{i\theta})|$ and F is the mapping function defined as $x + iy = F(re^{i\theta})$. The boundary conditions for the problems are (a) the flow tangency condition on the surface, i.e.,

$$\hat{c}\phi/\hat{c}n = 0, \quad (16)$$

and (b) outer boundary condition written as;

$$\Phi = (\Gamma/2\pi)\tan^{-1}((1 - M_\infty^2)^{1/2} \tan(\theta + \alpha)), \quad (17)$$

where Φ is the full potential, α is the angle of attack, θ is the coordinate in the (r, θ) plane and M_∞ is the free stream Mach number. Equation (15) is discretized using the mixed finite difference scheme due to Murman & Cole (1971).

4.1 Multigrid acceleration of full potential equation

The nonlinearity of the full potential transonic flow equation with shock sitting on the airfoil surface poses special problems in the multigrid process by making the solution very sensitive. It turns out that the high frequency errors introduced during the coarse grid correction interpolation, especially in the vicinity of the shock, keep offsetting the smoothness obtained on the finer levels. This impairs the rate of convergence.

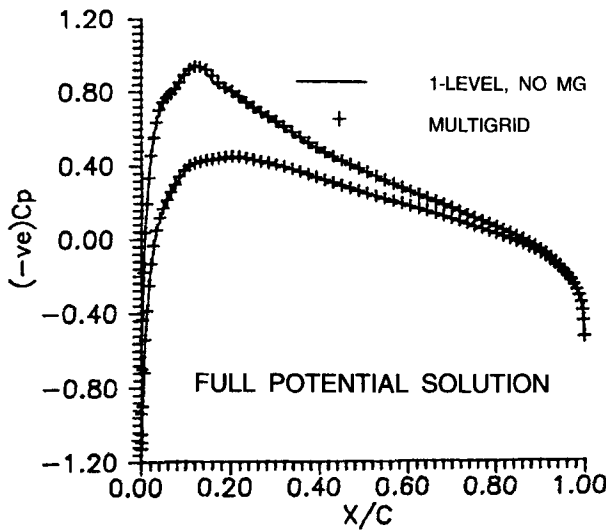


Figure 5. Comparison of pressure distribution with and without multigrid, NACA 0012, $M = 0.7$, $\alpha = 1^\circ$.

The multigrid strategy selected here also is the FAS scheme starting at the finest of the 3-level grid hierarchy with V-cycle. The finest grid is 160×32 . The next two coarser grids are obtained by omitting the alternate points in both r and θ directions. The coarse grid correction is interpolated using simple algebraic averaging of the values at the two coarse grid nodes. The relaxation sweep strategy adopted for the results presented here consists of a constant number of sweeps at all levels in both restriction as well as in the prolongation phases (though a choice exists to select any number of sweeps at any level). The boundary conditions are updated at all the grid levels. The computations are terminated when the L_2 -norm of the residuals becomes less than 10^{-4} .

Using this simple strategy, two typical results are presented for flow computations made for (i) NACA 0012 airfoil at $M_\infty = 0.7$ and $\alpha = 1^\circ$ and (ii) Garabedian-Korn

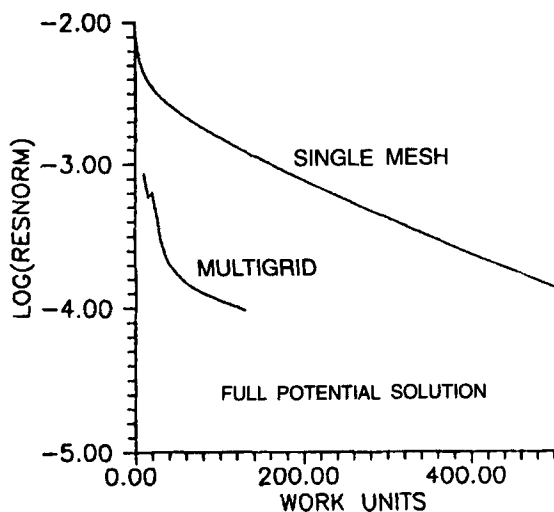


Figure 6. Comparison of convergence with and without multigrid, NACA 0012, $M = 0.7$, $\alpha = 1^\circ$.

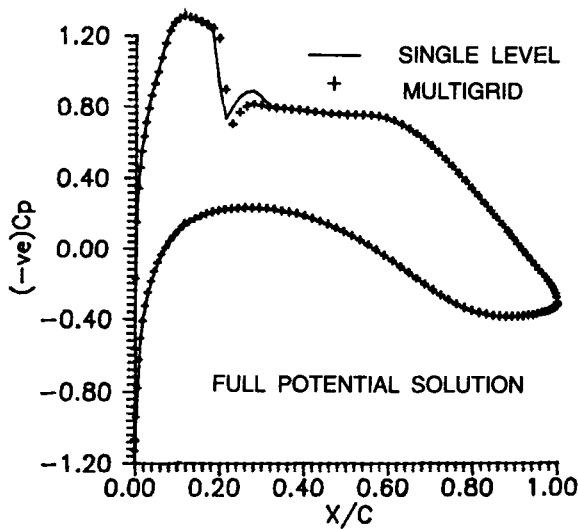


Figure 7. Comparison of pressure distribution with and without multigrid, G-K airfoil, $M = 0.702$, $\alpha = 1.1^\circ$.

airfoil, $M_\infty = 0.702$ and $\alpha = 1.1$. Figure 5 shows the pressure distribution for case (i) which is supercritical though without a shock. The agreement between the single grid computation and multigrid solution is just about total. Figure 6 shows the convergence history for this case. The single grid computation took 562 work units to converge to the specified tolerance level while the multigrid solution took only 125 work units leading to an acceleration of about 4.5 times in terms of work units. Figure 7 shows the comparison of pressure distribution for case (ii). This is a supercritical case with shock.

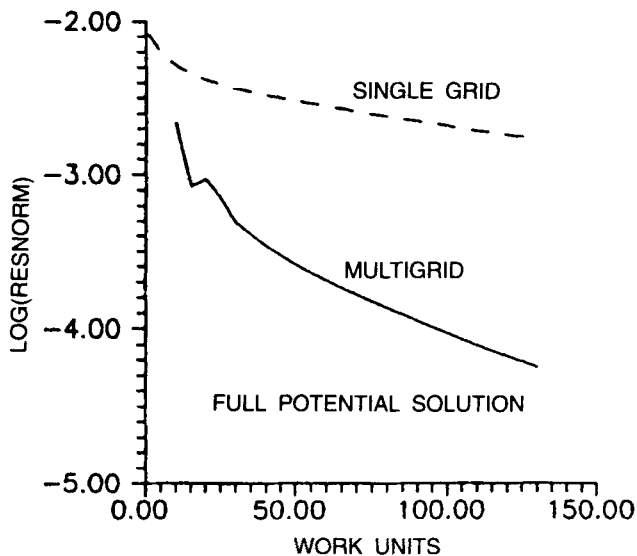


Figure 8. Comparison of convergence with and without multigrid, G-K airfoil, $M = 0.702$, $\alpha = 1.1^\circ$.

Like the earlier case, the agreement between the single grid computation and multigrid solution is good except for a small difference in the vicinity of the shock. The reason for this is explained earlier which is associated with the interpolation of coarse grid correction. Figure 8 shows the convergence history for this case. The single grid solution took 696 work units to converge to the specified tolerance level, whereas the multigrid solution took only 100 work units giving nearly 7 times faster convergence speed. The gain in convergence speed for more difficult cases like the ones with strong shock and high angle of attack is higher. This is because the instability of shock which greatly impairs the single grid convergence is effectively handled by the smoothing capability of the basic scheme at multiple levels with the added advantage of the coarse grid correction of the multigrid process. (The simple grid sequencing technique also derives the advantage of multiple levels though without the advantage of any interaction between the solution process and the discretization seen in the form of coarse grid correction.) In fact, this speed-up can be further improved by adopting strategies like implementing convergence rate based switching criterion from fine to crude grid in the injection phase, working on additional coarser levels, using W-cycles and full multigrid, trying out a better smoothing of corrections in the prolongation phase etc. Next, in the following section, implementation of multigrid acceleration to obtain solution of a more complex system, i.e., compressible Reynolds averaged Navier–Stokes equations, is described.

5. Multigrid acceleration of Navier–Stokes solver

The full Navier–Stokes equations are believed to provide complete physics of the various fluid flow phenomena. Thanks to the developments in the area of efficient algorithms and acceleration techniques, particularly the multigrid technique, on one hand, and the faster computing machines, on the other, that it has become possible to compute Navier–Stokes flows around practical configurations in reasonable turn-around time. Still when it comes to simulating turbulence, the situation is not very encouraging at least as far as the computing resources are concerned. In spite of these difficulties, the solutions of Navier–Stokes equations are of great interest where one solves what is called Reynolds averaged Navier–Stokes equations which involve introduction of an empirical model to simulate turbulence. The details on these aspects are now available in the text books and only a brief description is presented here for ready reference. Some important aspects of discretization of these equations, the solution process and the implementation of multigrid will be described in the next sections. This will be followed by discussions of a few cases of multigrid solution of the transonic flow computations. The performance of the multigrid convergence of the present Navier–Stokes code, based on the described strategy, has been tested over a range of Mach numbers between 0.1 to about 4.0 (Singh *et al* 1993).

Most of the progress in the multigrid solution of the Navier–Stokes equations has taken place during the last ten years or so. Several algorithms and methods have been devised to handle specific flow situations. A good number of these Navier–Stokes solvers use the finite volume discretization and Runge–Kutta time stepping scheme with addition of an explicit artificial viscosity to stabilise the solution. The solver in the present work is based on this method. (for details see Jameson *et al* (1981), Jameson (1985), Martinelli (1987), Kloppmann *et al* (1993), Singh *et al* (1993) and additional references therein).

5.1 The Navier–Stokes equations

The time-dependent Navier–Stokes equations for two-dimensional flow are written as (Singh *et al* 1993a);

$$\frac{\hat{c}}{\hat{c}t} \iiint \mathbf{W} d(\text{vol}) + \iint \mathbf{H} \cdot \hat{\mathbf{n}} dS = 0, \quad (18)$$

where vector \mathbf{W} is the cell averaged mass, momentum and total energy and is written as;

$$\mathbf{W} = (\rho, \rho u, \rho w, \rho E)^T, \quad (19)$$

and $\hat{\mathbf{n}}$ is the unit vector normal to the cell boundaries. u and w are the Cartesian velocities. The energy equation is written as;

$$E = e + (u^2 + w^2)/2, \quad (20)$$

where e is the mass averaged internal energy given as;

$$e = p/(\rho(\gamma - 1)), \quad (21)$$

and p is the pressure. The flux tensor \mathbf{H} is given as;

$$\mathbf{H} = \begin{bmatrix} \rho u & \rho w \\ \rho u^2 - \sigma_x & \rho w u - \tau_{zx} \\ \rho u w - \tau_{xz} & \rho w^2 - \sigma_z \\ (\rho E - \sigma_x)u - \tau_{xz}w + q_x & (\rho E - \sigma_z)w - \tau_{zx}u + q_z \end{bmatrix}, \quad (22)$$

with

$$\sigma_x = -p + \frac{2}{3}\mu(2u_x - w_z)$$

$$\sigma_z = -p + \frac{2}{3}\mu(2w_z - u_x)$$

$$\tau_{xz} = \tau_{zx} = \mu(u_z + w_x)$$

q_x and q_z are the components of the heat flux vector.

$$q_x = -kT_x, \quad q_z = -kT_z.$$

μ is the molecular viscosity and is computed using the Sutherland law. k is the thermal conductivity. For turbulent flows, the Baldwin–Lomax (1978) model for eddy viscosity has been used.

5.2 Spatial discretization

The time dependent Navier–Stokes equations (18) are discretized based on the method of lines in which the spatial derivatives are discretized using the cell centred finite volume scheme. Applying these conservation equations to one of these cells, we can write

$$\frac{d\mathbf{W}}{dt}_{i,k} + (\mathbf{Q}_c + \mathbf{Q}_v - \mathbf{D})_{i,k} = 0. \quad (23)$$

In the above, \mathbf{Q}_c and \mathbf{Q}_v are the convective and viscous flux vectors. This discretization amounts to central differencing and requires explicit addition of artificial viscosity to provide (a) a high order background dissipation to avoid the odd-even decoupling and

(b) a second component active in the vicinity of the shock to prevent shock-induced oscillations. This term is indicated by \mathbf{D} in (23) and is described in the sub-section 5.3.

5.3 Artificial dissipation

In the present work, the Jameson–Schmidt–Turkel (Jameson *et al* 1981) scheme for the numerical dissipation has been used which contains a blend of second and fourth differences of the flow variables. The second difference term damps oscillations in the vicinity of the shock sensed by a pressure based sensor. The fourth difference term helps to avoid the odd-even decoupling and provides the background dissipation in the smooth regions of the flow. This scheme is described below. The Jameson–Schmidt–Turkel numerical dissipation scheme (Jameson *et al* 1981; Martinelli 1987) is given as;

$$\mathbf{D}_{i,k} = d_{i+1/2,k} - d_{i-1/2,k} + d_{i,k+1/2} - d_{i,k-1/2}, \quad (24)$$

where

$$d_{i+1/2,k} = \lambda_{i+1/2,k} [\varepsilon_{i+1/2,k}^{(2)} \delta_x d_{i,k} - \varepsilon_{i+1/2,k}^{(4)} \delta_x^3 d_{i-1,k}]. \quad (25)$$

δ_x is the forward difference operator. The coefficients $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ are written as;

$$\varepsilon_{i+1/2,k}^{(2)} = k^{(2)} \max(v_{i+1}, v_i), \quad (26)$$

$$\varepsilon_{i+1/2,k}^{(4)} = \max(0, k^{(4)} \varepsilon_{i+1/2,k}^{(2)}). \quad (27)$$

The pressure sensor v is given as;

$$v = |p_{i+1,k} - 2p_{i,k} + p_{i-1,k}| / (|p_{i+1,k}| + |2p_{i,k}| + |p_{i-1,k}|). \quad (28)$$

The Navier–Stokes grids are highly stretched. The grid spacing near the wall in the direction normal to it is typically of the order of $10^{-5}c$ (c being the airfoil chord) to resolve the boundary layer leading to streamwise aspect ratio of the cells very much larger than unity in the vicinity of the airfoil. This leads to a large value for artificial dissipation leading, in turn, to a large drag. In the far field, the situation is reversed as a result of which the cell aspect ratio defined as before is less than and often much less than unity. This disparity leads to very different high frequency modes in the two directions, and cause difficulty for both the convergence and accuracy of the steady-state flow computation as well as adversely affecting the multigrid procedure. To surmount this difficulty, the variable scaling factor (Martinelli 1987; Swanson & Turkel 1987) Λ has been used instead of λ in (25). For this, λ is written as

$$\lambda_{i+1/2,k} = 0.5 [(\Lambda_x)_{i,k} + (\Lambda_x)_{i+1,k}], \quad (29)$$

where

$$\Lambda_x = \phi_x \lambda_x, \quad \phi_x = 1 + (\lambda_z/\lambda_x)^\sigma,$$

λ_x and λ_z are the scaled spectral radii of the Jacobian matrices of the convective fluxes, σ is a constant. With this, it has been possible to handle highly stretched grids with significantly improved damping properties.

5.4 Runge–Kutta time stepping

The semi-discrete equation (23) may be written as;

$$(d\mathbf{W}_{i,k}/dt) + \mathcal{R}(\mathbf{W}_{i,k}) = 0. \quad (30)$$

\mathcal{R} is the discretized flux term and is known as residual. A hybrid 5-stage Runge–Kutta scheme has been used to advance the solution of (30) in time. Following Jameson (1985), this is written as

$$\begin{aligned}\mathbf{W}^{(0)} &= \mathbf{W}^{(n)}, \\ \mathbf{W}^{(m)} &= \mathbf{W}^{(0)} - \alpha_m \Delta t [Q_c(\mathbf{W}^{(m-1)}) + Q_c(\mathbf{W}^{(0)}) - D^{(m-1)}], \\ \mathbf{W}^{(n+1)} &= \mathbf{W}^{(5)},\end{aligned}\quad (31)$$

m is the Runge–Kutta stage number (1 to 5 for 5-stage), n the time index, $\mathbf{W}^{(0)}$ the starting solution or the solution at previous time step. The convective fluxes are computed at every stage. For time economy, the viscous fluxes, considered in full, are computed only at the first stage and the numerical fluxes are computed at alternate stages employing a combination of present and previous values as follows

$$\begin{aligned}D^{(0)} &= D^{(1)} = D(\mathbf{W}^{(0)}), \\ D^{(2)} &= D^{(3)} = \beta D(\mathbf{W}^{(2)}) + (1 - \beta)D^{(0)}, \\ D^{(4)} &= \gamma D(\mathbf{W}^{(4)}) + (1 - \gamma)D^{(2)}.\end{aligned}\quad (32)$$

The weighting factors used above are $\beta = 0.56$ and $\gamma = 0.44$. The coefficients α_m are taken as $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, $\alpha_3 = 3/8$, $\alpha_4 = 1/2$, $\alpha_5 = 1$. It has been shown by Martinelli (1987) that this scheme has a large parabolic stability limit. Since only the steady state solution is of interest here, a local time stepping scheme is used to accelerate the solution process wherein the time steps for each cell are taken to be the maximum permitted by the local CFL criterion.

5.5 Implicit residual smoothing

A second technique that is used to accelerate the solution procedure is that of implicit residual smoothing which enlarges the stability range of the basic time stepping scheme enabling larger time steps to be taken. Also, the implicit residual smoothing is known to significantly enhance the smoothing characteristics of the time marching scheme for use with multigrids (Wigton & Swanson 1990). In two dimensions, the residual smoothing operation is defined as follows;

$$(1 - \varepsilon_x \delta_{xx})(1 - \varepsilon_x \delta_{zz})\mathcal{R}^* = \mathcal{R} \quad (33)$$

where \mathcal{R}^* is the smoothed residual. The smoothing parameters ε 's are different for each cell because of different aspect ratios. The coefficients ε developed by Martinelli (1987) are quite effective for highly stretched meshes and have been used in the present scheme. These locally varying coefficients are given as;

$$\varepsilon_x = \max \left\{ 0, \frac{1}{4} \left[\left(\frac{\text{CFL}}{\text{CFL}_0} \frac{\lambda_x}{\lambda_x + \lambda_z} \phi_x \right)^2 - 1 \right] \right\}. \quad (34)$$

CFL_0 is the natural Courant–Friedrich–Lewis (CFL) number of the basic scheme. The expressions for ε_z can be written in a similar way. While this variable coefficient implicit procedure requires additional CPU time to compute ε 's for each cell, this is more than compensated for by the fact that it allows CFL numbers two to three times larger than the original scheme. The highest CFL number tried in the present computations is about 7.5.

5.6 Boundary conditions

The airfoil surface is treated as an adiabatic wall on which no-slip boundary condition is applied and zero normal pressure gradient is assumed. The formulation being cell centred, the variables are not specified on the surface. This requires considering a dummy cell adjacent to the boundary. To apply the wall boundary condition, simple extrapolation of pressure to the dummy cell is done and the density is found by symmetry. The velocity components are set antisymmetric.

The farfield boundary condition is based on the use of Riemann invariants for one-dimensional flow normal to the boundary. The boundary conditions for normal velocity component and speed of sound are constructed by adding and subtracting the invariants (for details see Kroll & Jain 1987). For subsonic free streams, the tangential velocity components and entropy are extrapolated from the interior at an outflow boundary and are equated to freestream values at the inflow boundary. With this, the flow variables in farfield are computed.

5.7 Multigrid strategy

The basic features of the multigrid scheme used here are along the lines developed by Jameson (1985). The specific features of the present method can be seen in Singh *et al* (1993) and Kloppmann *et al* (1993). The Runge–Kutta time marching procedure quickly smoothes out the high frequency errors (short waves) on the fine meshes where further smoothing operations would become progressively slower. Note here that in the context of multigrid, the schemes like Gauss–Seidel, predictor–corrector, Runge–Kutta and so on are seen as smoothers rather than solvers. This is because one intends to get a good coarse grid approximation from the fine grid solution by having the fine grid starting approximation smoothed, i.e., high frequency errors diminished. The solution with the remaining error components are passed on to the next coarser mesh until that coarsest mesh is reached where the low frequency errors (long waves) of the finer meshes are seen as high frequency errors. The basic Runge–Kutta smoother is again used to remove these long wave (with respect to the finest mesh level on which the solution is sought) errors.

The scheme may use any number of levels of mesh refinement which is read as input in the code. In all the results presented here, the Full Approximation Storage scheme is used in which one starts with an initial solution on the finest mesh. In the present computations, three or four mesh levels have been used with V or W-cycles. The scheme for the residual transfer and correction interpolation are important steps for a good multigrid performance. The scheme should be such that the coarse grid correction improves the finer grid solution in smooth regions as well as correcting the discontinuity location. Towards this requirement, the flow variables are transferred from fine to coarse grid as follows (in the following, the subscript will indicate the grid level identifier).

$$\mathbf{W}_{2h}^{(0)} = \sum V_h \mathbf{W}_h / V_{2h}, \quad (35)$$

where V is the cell volume and h and $2h$ refer to the finer and coarser meshes levels, respectively. The summation is performed over the four fine cells constituting one coarse cell. This volume weighted averaging preserves mass, momentum and energy

conservation. The residuals are transferred as simple sums of the residuals in the four constituent fine cells. The eddy viscosity is computed only on the finest mesh and is carried on to the coarser mesh by simple averaging. To update the solution on the coarse mesh, the residual is modified by adding the forcing function defined as;

$$f_{2h} = - \sum R_h(\mathbf{W}_h) + R_{2h}(\mathbf{W}_{2h}^{(0)}) \tag{36}$$

The summation is performed as before. With this, the multistage scheme on the coarser mesh is written as;

$$\begin{aligned} \mathbf{W}_{2h}^{(0)} &= \mathbf{W}_{2h}^{(n)} \\ \mathbf{W}_{2h}^{(1)} &= \mathbf{W}_{2h}^{(0)} - \alpha_1 \Delta t (R_{2h}^{(0)} - f_{2h}), \\ &\dots\dots\dots \\ \mathbf{W}_{2h}^{(m)} &= \mathbf{W}_{2h}^{(0)} - \alpha_m \Delta t (R_{2h}^{(m-1)} - f_{2h}), \\ &\dots\dots\dots \\ \mathbf{W}_{2h}^{(n+1)} &= \mathbf{W}_{2h}^{(m)} \end{aligned} \tag{37}$$

where n is time level and m is the number of stages. Note that in the first stage above, the entire corrected residual reduces to $\sum R_h(\mathbf{W})_h$ implying that the evolution on the coarse mesh is influenced by the residuals on the finer mesh. This procedure is followed until the coarsest mesh is reached. Only a single fine grid boundary condition update offers computation time economy without sacrificing the accuracy of the solution. The corrections computed on the coarse mesh are prolonged to the finer mesh by simple linear interpolation. This step introduces high frequency errors causing degradation of the convergence which is more pronounced if the grids are highly stretched. To improve the situation, the corrections are smoothed before update. An effective way to take into account the effect of high grid stretching is to use the implicit residual smoothing. The coarse grid correction is transferred to the finer mesh by

$$\mathbf{W}_h = \mathbf{W}_{h(\text{old})} + \tilde{\mathcal{I}}_{2h}^h (\mathbf{W}_{2h} - \mathcal{I}_h^{2h} \mathbf{W}_{h(\text{old})}), \tag{38}$$

where \mathcal{I} and $\tilde{\mathcal{I}}$ are suitable interpolation operators.

5.8 Results from Navier–Stokes computations

The multigrid accelerated Navier–Stokes code developed and based on the above strategy has been extensively used to compute several flow conditions on a variety of airfoils. Table 2 lists the four typical cases of airfoils and the corresponding flow conditions for which the computed results will be presented.

For the cases presented, a 257×41 size C-grid with a total of 64 points in the wake or 321×65 size C-grid with a total of 128 points in the wake has been used. The farthest grid points are placed at about 12 to 18 chords away from the airfoil and the grid line just above the airfoil is about 10^{-5} chords away to enable a good resolution of the boundary layer. The computations have been performed on the Convex 3820 computer at the CSIR Centre for Mathematical Modelling and Computer Simulation, Bangalore. Depending on the mesh size and the flow conditions, the solutions, converged to the level of acceptable tolerance, take just about 5 to 15 min of CPU time. The corresponding single grid solution takes about 2–3 hours of CPU time for similar

Table 2. Cases selected for Navier–Stokes computations.

Case	Airfoil	Grid	M_∞	α (deg)	Re(10^6)	Xtr
1.	CAST-7	257 × 41	0.70	2.00	4.0	7%
			0.70	2.64	6.0	8%
			0.70	3.60	6.0	8%
2.	RAE 2822	321 × 65	0.73	2.79	6.5	3%
3.	NACA 66, 2-215	257 × 41	0.601	2.0	1.5	64% up 73% low
4.	NACA 0012	257 × 41	0.6	8.437	9.0	5%
			0.6	9.0	9.0	5%

xtr – transition location as % of chord

and, in most cases, higher levels of residuals. This feature of the code, along with its robustness and accurate prediction capabilities, offers itself as a designer's tool for evaluating the characteristics of the airfoils.

5.8a CAST-7 airfoil: This is 11.8% thick supercritical airfoil with moderate rear loading, designed by the Dornier Company. This airfoil has been extensively studied by this author during the early phase of the work with this code without multigrid (Singh 1990). Table 3 shows the details of the studies made on this airfoil and presented in this report.

The single level computation and 3-level grid sequencing computations at serial numbers 1 and 2 are included from the point of view of illustrating the effect of factors other than multigrid as will be explained. The single grid computation (no. 1) refers to

Table 3. Comparison of performance of Navier–Stokes code with and without multigrid (Cast-7 airfoil).

Computational strategy	c_l	c_d	Work units	Convergence level	CFL	Remarks
1. Single grid	0.8197	0.01905	6500	$\approx 1.97 \times 10^{-1}$	0.6	4-stg RK, No IRS std JST
2. 3-Level grid sequencing	0.8175	0.01899	3437	—	0.6	4-stg RK, No IRS std JST
3. Single grid	0.8117	0.01898	3000	$\approx 10^{-5}$	1.6	5-stg hybrid RK, VC-JST, VC-IRS
4. 3-Level MG V-cycle	0.8174	0.01926	147	2.06×10^{-4} for stabilised Cl	1.6	5-stg hybrid RK, VC-JST, VC-IRS
5. 3-Level MG V-cycle	0.8172	0.01921	496	9.51×10^{-8} for convergence tolerance of 10^{-7}	1.6	5-stg hybrid RK, VC-JST, VC-IRS

4- and 5-stg RK – 4- and 5-stage standard Runge–Kutta scheme

IRS – implicit residual smoothing

VC – variable coefficient (eigenvalue weighted)

JST – Jameson–Schmidt–Turkel artificial dissipation

CFL – normalised CFL number defined as CFL/CFL_0 (see (34))

the original status of the code when the work on the development of the accelerated Navier–Stokes code was started under the NAL–DLR (Göttingen) collaboration. Computation no. 2 refers to the first stage of the acceleration work using grid sequencing (Singh 1990). Both these computations use a standard 4-stage Runge–Kutta scheme with the Jameson–Schmidt–Turkel dissipation (Jameson *et al* 1981). Also, to stabilise the solution properly, a normalised CFL number of 0.6 has been used which gives slow convergence rates in these two cases. The single grid version at serial number 3 and the multigrid version have additional features like eigenvalue scaled dissipation coefficients, hybrid 5-stage Runge–Kutta time integration and variable coefficient implicit residual smoothing whose effects are described in §§ 5.3 to 5.5. These elements offer the multigrid version of the code not just the weighted dissipation but higher CFL too. The normalised CFL used in the presented multigrid example is 1.6. These points should be borne in mind while making comparisons based on the data in table 3.

Figure 9 shows the comparison of the pressure distributions obtained using single grid (dashed line), 3-level grid sequencing results (shown by *) and 3-level multigrid V-cycle results (full line). This study presents the validation test to check the faithful reproduction of earlier verified and well-studied single mesh and 3-level grid sequencing computations (Singh 1990) by the present multigrid computations. Under these flow conditions, the airfoil develops a shock of moderate strength on the upper surface at around 35–37% of chord. The agreement of C_p -distributions obtained from the three computations is almost total. Figure 10 shows a similar comparison of the skin friction (normalised with respect to free stream quantities) obtained from the single grid version (dashed line), the grid sequencing version (no. 2 with *) and the multigrid version (no. 4, full line). As in the case of C_p , the agreement of skin friction obtained from these versions is also nearly total. The skin friction is a more sensitive quantity

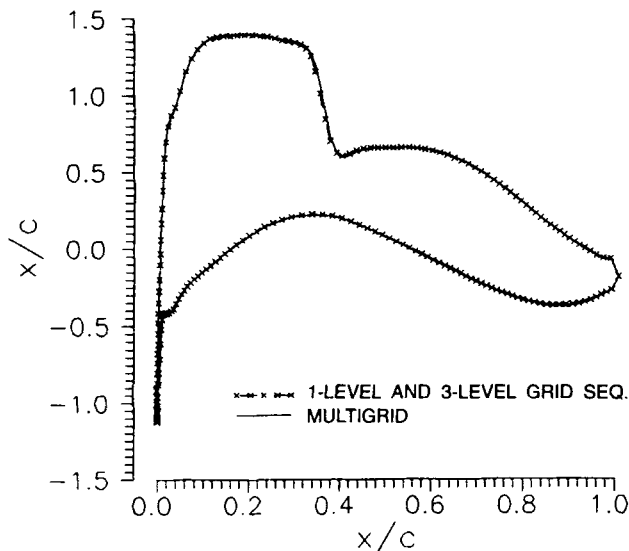


Figure 9. Comparison of pressure distribution with and without multigrid, N–S solution, Cast-7, $M = 0.7$, $\alpha = 2^\circ$.

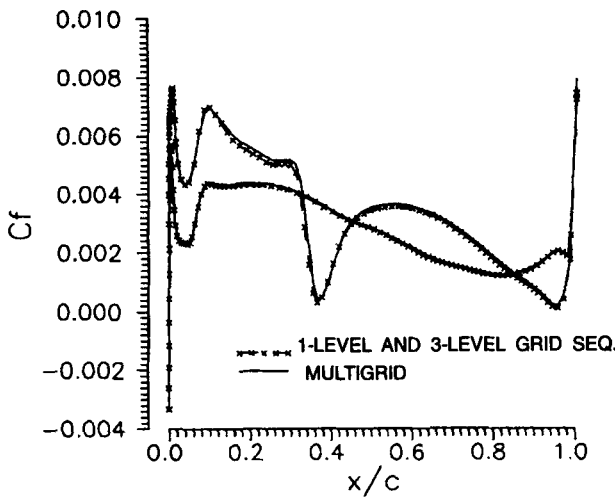


Figure 10. Comparison of skin-friction, N-S MG solution, Cast-7 airfoil, $M = 0.7$, $\alpha = 2^\circ$.

than C_p and these agreements confirm the general reliability of the codes as well as that of the multigrid implementation. Figure 11 shows the comparison of convergence rates of the single grid computation (case 3) and multigrid computations. This figure shows that to achieve a drop of five orders in the normalised maximum residual, the multigrid requires approximately 280 work units, whereas the single grid computation requires approximately 2800 work units. Figure 12 shows the comparison of evolution of the lift. It is seen that while the single grid computation takes about 1200–1300 work units to reach stabilised lift level (but with higher residue level), the multigrid computation shows a rapid and smooth evolution stabilising in about 200 work units with one order smaller residue level. This rapid stabilisation of lift coefficient, which is sustained without any oscillation, indicates a reliable, robust and stable computation.

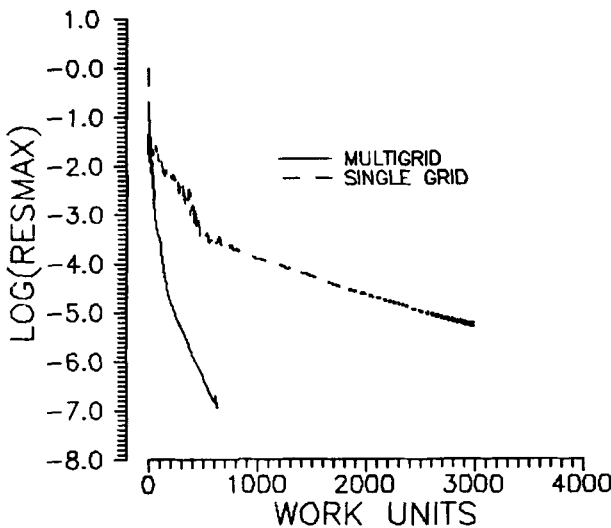


Figure 11. Comparison of convergence history, N-S solution, Cast-7, $M = 0.7$, $\alpha = 2^\circ$.

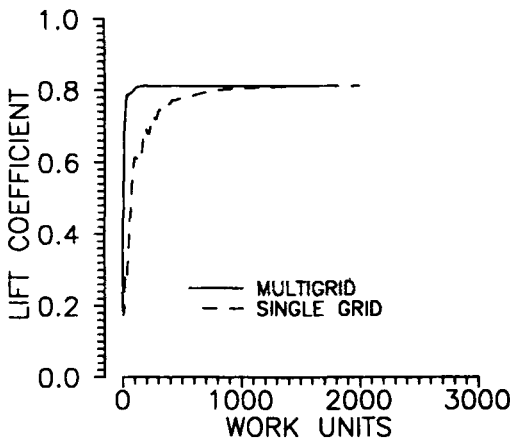


Figure 12. Comparison of lift evolution, N-S MG solution, Cast-7, $M = 0.7$, $\alpha = 2^\circ$.

Before proceeding further to discuss a few computations and their comparisons with experimental data, it would be of interest to discuss here the effects of V and W multigrid cycles and the number of smoothings at each level on the performance of multigrid technique. Figure 13 shows the effect of V and W-cycles on the convergence history of flow past RAE 2822 airfoil. The two computations have all other inputs identical. A marginal gain in convergence rate is seen with W-cycle (Kloppmann *et al* 1993). Figure 14 shows the effect of number of smoothings performed at each grid level and also the effect of V and W-multigrid cycle strategies for flow past NACA 0012 airfoil. The numbers in the legend indicate the strategy used. For example, "case 1: 3-lvl V-cycle 8/4/2 & 0/0/0" indicates that the computation in case 1 uses 3 level V-cycle with 8 smoothings at the coarsest level, 4 at the medium level and 2 at the finest level in the injection phase and no smoothing during the prolongation phase. This experiment showed that most computations, especially the ones using W-cycle, failed to converge if no smoothing was performed during the prolongation phase. The figure also indicates faster drop of residual with larger number of smoothings performed on the coarser

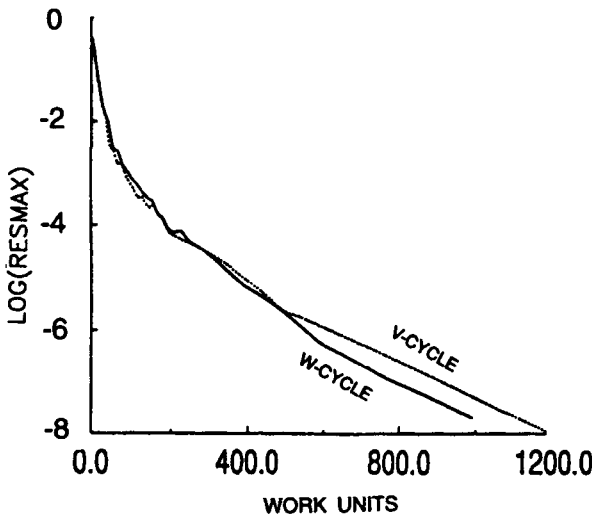


Figure 13. Effect of V- and W-cycles on the multigrid convergence.

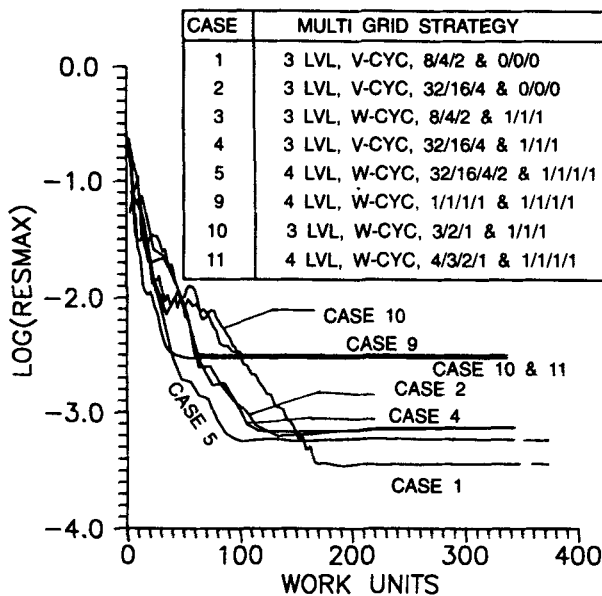


Figure 14. Effect of number of smoothings on the multigrid convergence.

levels. The fastest convergence is obtained with the combination used in case 5 whereas the residual is driven to the lowest level in case 1.

Next, the results of some computations for transonic flow past airfoils will be presented and compared with the experimental data to demonstrate the robustness and accuracy of the code and the efficiency obtained due to multigrid technique.

Figure 15 and 16 show the comparisons of the predicted pressure distributions with experimental data (Stanewski *et al* 1979) for flow past Cast-7 airfoil at $M = 0.7$ at angles of attack of (i) 2.64° and (ii) 3.6° . The Reynolds number for both cases is about 6×10^6 with transition forced at 8% chord. The agreement for the case (i) is good everywhere including the shock location and its intensity. The agreement for the case (ii) is also good but the computed shock intensity is somewhat stronger. The grid for this case is a 257×41 C-grid with 64 points in the wake and 193 points on the airfoil. The lateral separation of the grid is such that there are nearly 25 grid lines inside the boundary layer and the grid lines in the outer flow are sparsely located. This indicates that if the flow inside the boundary layer is well resolved, the outer mesh density plays a secondary role. This fact is well borne out not just by the good pressure prediction but also by the well predicted drag. Table 4 summarises the quantitative comparison for these two cases.

5.8b RAE 2822 airfoil: For this study, the standard transonic test case number 9 from the AGARD set of data (Stanewski *et al* 1979) is selected. This computation has been carried out on a fairly fine mesh of size 321×65 . The experimental incidence for this case is 3.19° . The present computations have been made at a corrected angle of attack of 2.79° as suggested by the experimenters (Stanewski *et al* 1979) to accommodate the effect of wind tunnel wall interference. Figure 17 shows the comparison of computed pressure distribution with the AGARD data (Stanewski *et al* 1979). The agreement is seen to be good almost everywhere including at the peak plateau and

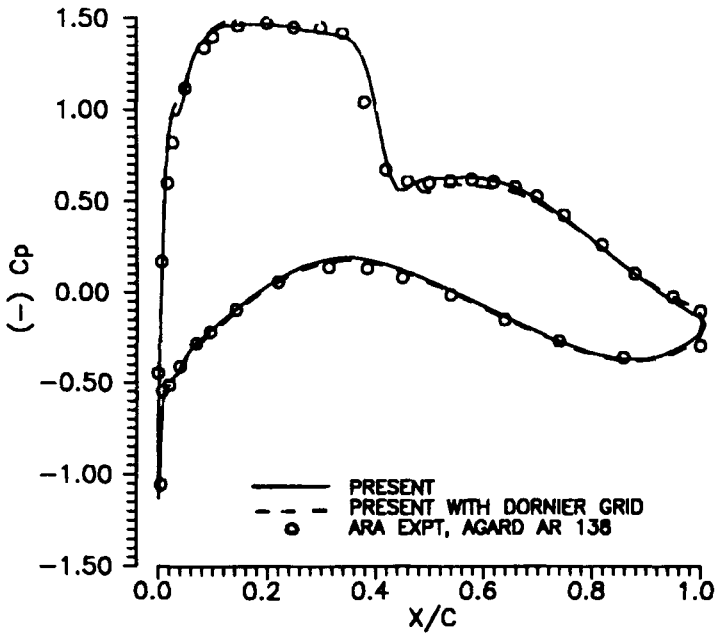


Figure 15. Comparison of C_p -distribution, Cast-7, $M = 0.7$, $\alpha = 2.64^\circ$, $Re = 6 \times 10^6$.

shock location. It may be pointed out that very often in the literature, a significant difference is seen in the peak plateau region. The experimental values of lift and drag coefficients ($c_l = 0.803$ and $c_d = 0.0168$ respectively) agree well with the predicted values ($c_l = 0.799$ and $c_d = 0.019$ respectively). As seen in figure 18, the comparison of skin

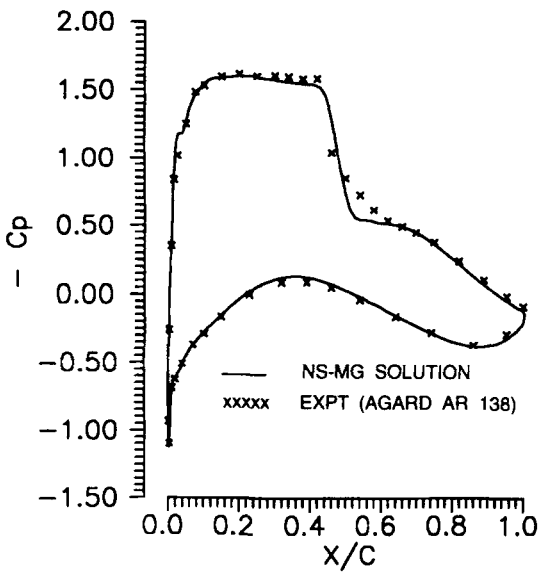


Figure 16. Comparison of C_p -distribution, Cast-7, $M = 0.7$, $\alpha = 3.6^\circ$, $Re = 6 \times 10^6$.

Table 4. Quantitative comparison with experimental data Cast-7 Airfoil.

Case	M_∞	α (deg)	Re 10^6	Xtr	c_l (present)	c_l^*	c_d (present)	c_d^*
(i)	0.70	2.64	6.0	8%	0.8809	0.867	0.02548	0.01879
(ii)	0.70	3.60	6.0	8%	1.0289	1.027	0.0393	0.03286

*Reference – Stanewski (1993b)

friction (normalised with respect to the quantities at the boundary layer edge to be in conformity with the experimental data) is also in good agreement with the experimental data.

5.8c *NACA66,2-215 low drag airfoil*: This is one of the low drag airfoils (what is now known as NLF airfoils) designed at the erstwhile NACA during the 40's. Its characteristics are studied and reported (Graham *et al* 1945). The airfoil was designed to perform at high subsonic Mach numbers. The transition points on the upper and lower surfaces, as given in table 2, were fixed at locations approximately predicted using the Granville criterion as the related information was not available. Figure 19 shows the comparison of the computed pressure distribution with experimental data (Graham *et al* 1945). The agreement is seen to be good. Figure 20 shows the corresponding convergence history. The residual drops to about 3×10^{-5} in about 400 work units.

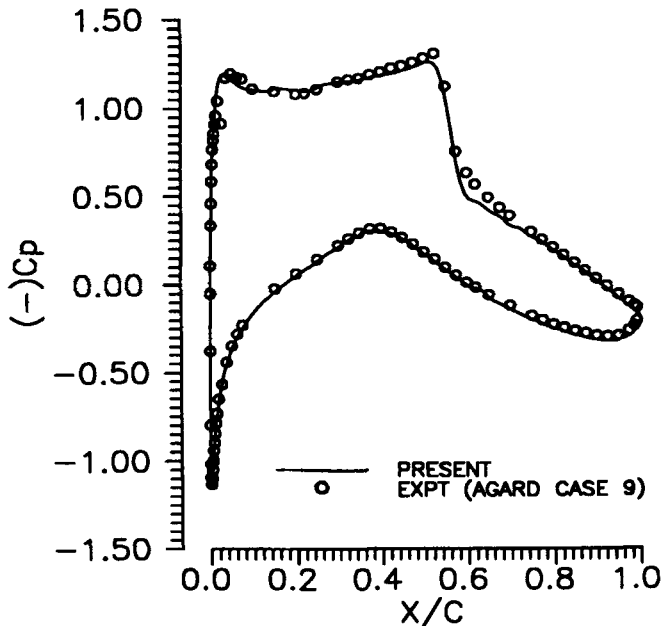


Figure 17. Comparison of C_p -distribution. N-S MG solution, RAE 2822, $M = 0.73$, $\alpha = 2.79^\circ$, $Re = 6.5 \times 10^6$.

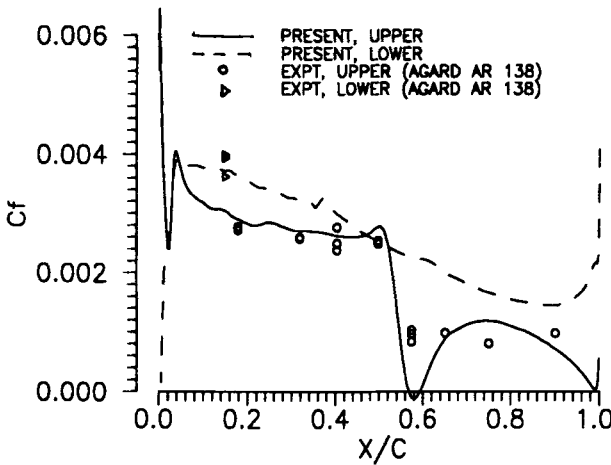


Figure 18. Comparison of skin-friction. N-S MG solution, RAE 2822 airfoil, $M = 0.73$, $\alpha = 2.79^\circ$, $Re = 6.5 \times 10^6$.

5.8d *NACA 0012 airfoil*: This airfoil was chosen to demonstrate the application of the multigrid Navier–Stokes code to compute flows at high angles of attack with large trailing edge separation. The Baldwin–Lomax model for turbulence is not adequate when the flow is largely separated. With the present experience, the Granville correction to this model appears to give results in much better agreement under separated flow conditions (Singh *et al* 1993). However, under the nonequilibrium conditions of shock induced separation, this model does not seem to be as effective as under the conditions of the trailing edge flow separation at low speeds. Figure 21 shows the comparison of C_p distribution at $M_\infty = 0.6$, $\alpha(\text{corrected}) = 8.437^\circ$, $Re = 9 \times 10^6$ and transition fixed at 5% chord. This case corresponds to the stall condition with

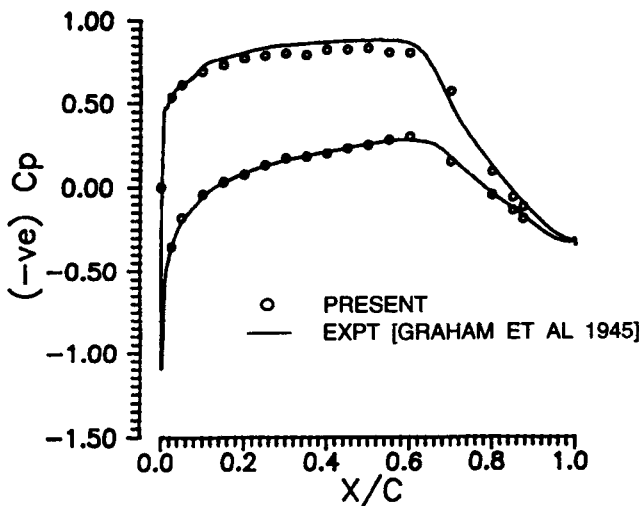


Figure 19. Comparison of pressure distribution, NACA 66, 2-215 airfoil, $M = 0.601$, $\alpha = 2^\circ$, $Re = 1.5 \times 10^6$.

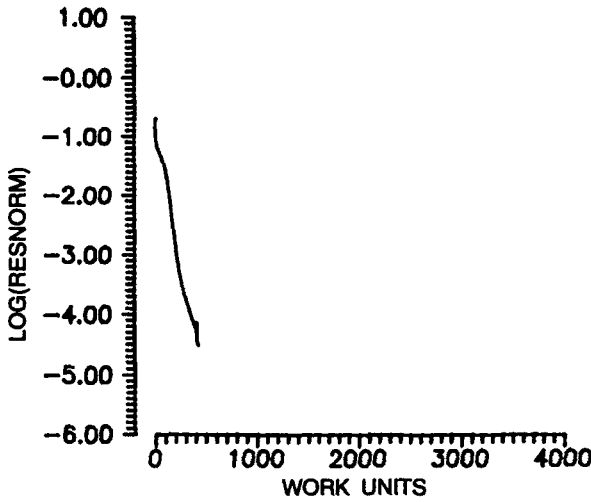


Figure 20. Convergence history, NACA 66, 2-215 airfoil, $M = 0.6$, $\alpha = 2^\circ$, $Re = 1.5 \times 10^6$.

$\alpha_{\text{expt}} = 9.86^\circ$. The results are compared with the experimental data of Harris (1981). The predicted pressure distribution with the Granville correction is seen to be in better agreement with the experimental data than that predicted using the original Baldwin-Lomax model. Figure 22 shows the computed skin friction (quantities normalised with respect to free stream values). A separation bubble between about 16% to 23% of chord is seen with a brief reattachment followed by separation on the entire upper surface. Figure 23 shows streamline pattern around this airfoil under the same conditions but at slightly higher (corrected) angle of attack of 9° . The separation bubble, which was very small for the case with $\alpha = 8.437^\circ$ to offer good graphic representation is slightly enlarged in the present case and can be seen clearly in figure 24. The flow reattaches briefly after the bubble and separates again over the

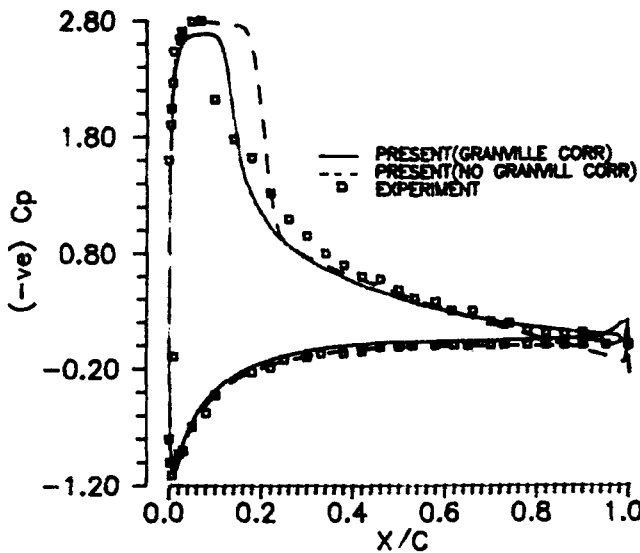


Figure 21. Comparison of pressure distribution, NACA 0012 airfoil, $M = 0.6$, $\alpha(\text{corr}) = 8.437^\circ$, $Re = 19 \times 10^6$, $x_{tr} = 5\%$, stall condition.

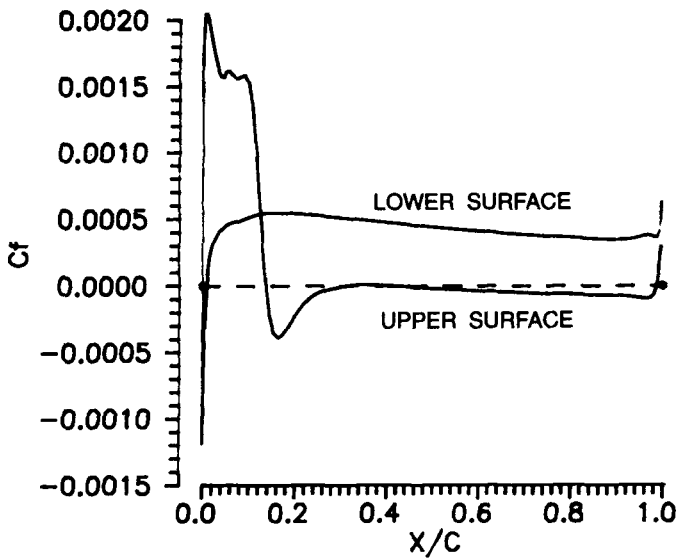


Figure 22. Computed skin friction NACA 0012 airfoil, $M = 0.6$, $\alpha(\text{corr}) = 8.437^\circ$, $Re = 9 \times 10^6$.

entire upper surface. The skin friction inset in figure 24 clearly shows the location of the bubble and the separated flow on the upper surface. The flow evolution for this computation is wavy with lift coefficient oscillating over a couple of hundreds of work units with the amplitude of oscillation of lift continuously reducing and finally settling down. This indicates that the flow has still not become unsteady and the results are fairly acceptable for design estimates as indicated by the stabilised lift. The normalised residual could not be driven below about 10^{-2} .

6. Conclusions

The multigrid technique is, perhaps, the best known acceleration technique for iterative solution procedures. It offers the possibility of a realistic turn-around time for many computations which otherwise take hours and days on most of the commonly used

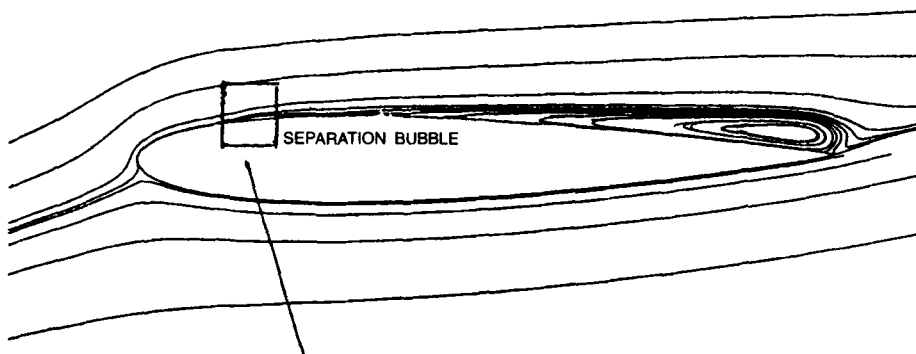


Figure 23. Streamline pattern, N-S MG solution, NACA 0012 airfoil, $M = 0.6$, $\alpha(\text{corr}) = 9^\circ$ (Granville correction to B-L model, stall condition).

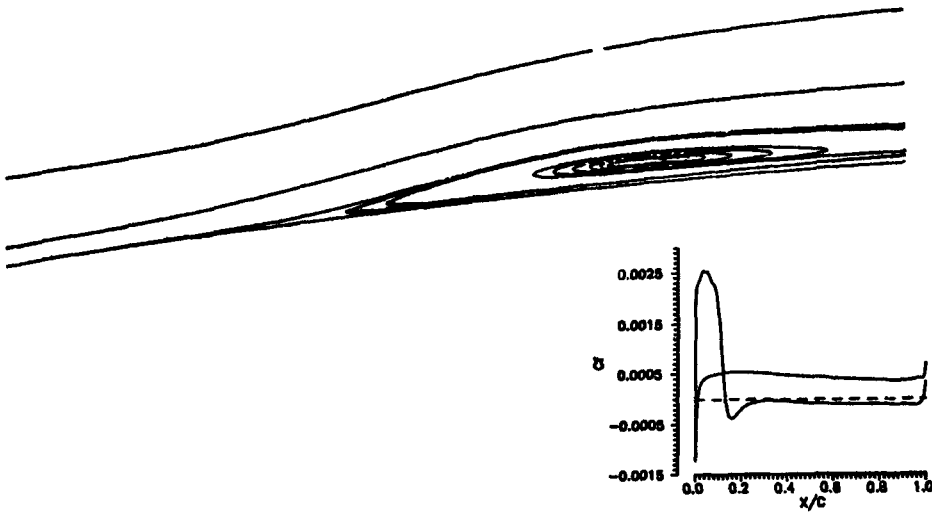


Figure 24. Magnified view of the separation bubble of figure 23. The inset shows the skin friction. The reversed flow inside the bubble and the fully separated flow downstream is clear from the negative skin friction.

computing machines. Its application to three common fluid dynamical problems is demonstrated and the quantitative gains are seen to be in the range of 3 to 10 times in terms of work units. The accelerated codes, which have been demonstrated to be fairly robust and accurate, offer themselves as designer's tools for evaluating the characteristics of the airfoils, especially when used on high speed computing machines.

References

- Baldwin B S, Lomax H 1978 Thin layer approximation and algebraic model for separated turbulent flows. AIAA Paper 78-257
- Brandt A 1972 Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. *Proc. of Third ICNMF D. Lecture Notes in Phys. 18* (Berlin: Springer Verlag) vol. 1, pp 82-89
- Brandt A 1977 Multi-level adaptive solutions to boundary-value problems. *Math. Comput.* 31: 333-390
- Brandt A 1989 The Weizmann Institute Research in multilevel computation: 1988 Report. *Proc. of 4th Copper Mountain Conf. on Multigrid Methods*
- Federenko R P 1964 On the speed of convergence of an iterative process. *USSR Comput. Math. Math. Phys.* 4: 227-235
- Garabedian P, Korn D 1971 Analysis of transonic airfoils. *Commun. Pure Appl. Math.* 24: 841-851
- Graham D J, Nitzberg G E, Olson R N 1945 A systematic investigation of pressure distribution at high speeds over five representative NACA low drag and conventional airfoil sections. NACA TR-832
- Hackbusch W 1976 Ein iteratives Verfahren zur schnellen Aufloesung elliptischer Randwertprobleme. Report 76-12, Universitaet Koeln
- Harris C D 1981 Two-dimensional aerodynamic characteristics of the NACA 0012 airfoil in the Langley 8-foot transonic pressure tunnel. NASA TM 81927

- Jameson A, Schmidt W, Turkel E 1981 Numerical simulation of the Euler equations by finite volume methods using Runge–Kutta time stepping-schemes. *AIAA Paper* 81–1259
- Jameson A 1985 Multigrid algorithm for compressible calculations. MAE Report 1743, Princeton University
- Jameson A 1993 Numerical wind tunnel – vision or reality. *AIAA 24th Fluid Dynamics Conference*, Orlando, FL, *AIAA Paper* 93–3021
- Kloppmann Ch, Schwamborn D, Singh J P 1993 Multigrid solutions of the 2-D Navier–Stokes equations for transonic internal and external flows. *4th European Conference on Multigrid*, Amsterdam
- Kroll N, Jain R K 1987 Solution of two-dimensional Euler equations – Experience with a finite volume code. DFVLR-FB 87-41, DLR, Braunschweig
- Martinelli L 1987 *Calculation of viscous flows with a multigrid method*. PhD dissertation, Mechanical & Aerospace Engineering Department, Princeton University, Princeton, NJ
- Murman E M, Cole J D 1971 Calculation of plane steady transonic flow. *AIAA J.* 9: 114–121
- Singh J P 1989 Solution of Laplace equation using full approximation storage multigrid technique. NAL PD CF 8927, NAL, Bangalore
- Singh J P 1992 Multigrid accelerated relaxation solution of transonic full potential flow equation. NAL PD CF 8942. Also *Proc. Int. Conf. on Methods of Aerophysical Research*, Novosibirsk
- Singh J P 1990 Accelerated Navier–Stokes solution. Part I: Grid sequencing. *IB* 221–90 A 20, DLR, Göttingen
- Singh J P 1992 An illustrative study of the impact of boundary condition defect on driving the multigrid procedure. *5th Asian Congress on Fluid Mechanics* (Taejon, Korea: Department of Aerospace Engineering, KAIST)
- Singh J P, Schwamborn D, Kloppmann Ch 1993a Development of a general purpose multigrid accelerated Navier–Stokes solver. NAL SP-9315, NAL, Bangalore
- Singh J P, Schwamborn D, Kloppmann Ch 1993b Multigrid Navier–Stokes solution of flow past airfoil around stall. *5th Int. Symp. on CFD* (Sendai, Japan: Jpn. Soc. Comput. Fluid Dynamics)
- South J C, Brandt A 1977 Application of a multi-level grid method to transonic flow computations. In *Transonic flow problems in turbomachinery* (eds) T C Adamson, M F Platzer (Washington: Hemisphere)
- Stanewski E, Puffert W, Mueller R, Bateman T E B 1979 Supercritical airfoil Cast-7 – surface pressure, wake and boundary layer measurements. In *Experimental data base for computer program assessment*, AGARD AR 138
- Swanson R C, Turkel E 1987 Artificial dissipation and central difference schemes for the Euler and Navier–Stokes equations. *AIAA Paper* 87–1107
- Wesseling P 1990 Multigrid methods in computational fluid dynamics. *Z. Angew. Math. Mech.* 70: T337–T347
- Wigton L B, Swanson R C 1990 Variable coefficient implicit residual smoothing. *Proceedings of the 12 ICNMF. Lecture Notes in Phys.* 371 (Berlin: Springer-Verlag)