

**An Upwind Multigrid Method for Solving Viscous
Flows on Unstructured Triangular Meshes**

by

Daryl Lawrence Bonhaus

B.S. June 1990, University of Cincinnati

A Thesis submitted to

The Faculty of

The School of Engineering and Applied Science

of The George Washington University in partial satisfaction

of the requirements for the degree of Master of Science

August 13, 1993

ABSTRACT

A multigrid algorithm is combined with an upwind scheme for solving the two-dimensional Reynolds-averaged Navier-Stokes equations on triangular meshes resulting in an efficient, accurate code for solving complex flows around multiple bodies. The relaxation scheme uses a backward-Euler time difference and relaxes the resulting linear system using a red-black procedure. Roe's flux-splitting scheme is used to discretize convective and pressure terms, while a central difference is used for the diffusive terms. The multigrid scheme is demonstrated for several flows around single and multielement airfoils, including inviscid, laminar and turbulent flows. The results show an appreciable speedup of the scheme for inviscid and laminar flows, and dramatic increases in efficiency for turbulent cases, especially those on increasingly refined grids.

ACKNOWLEDGMENTS

The author gratefully acknowledges the technical assistance of Dr. W. Kyle Anderson and the helpful technical discussions with Dr. Dimitri J. Mavriplis, N. Duane Melson and Gary P. Warren. Special thanks also to Jerry C. South, Jr. and Dr. James L. Thomas for technical reviews of the manuscript.

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGMENTS	ii
LIST OF FIGURES	v
LIST OF SYMBOLS	viii
Chapter	
I. INTRODUCTION	1
II. MULTIGRID	4
Linear Systems	4
Nonlinear Systems	6
Multigrid Cycles	7
Intergrid Transfers	8
III. THE MODEL PROBLEM — LAPLACE’S EQUATION	13
Spatial Discretization	13
Iteration Scheme	14
Results	16
IV. VISCOUS FLOW SOLVER	19
V. RESULTS	20
Euler Solution	20
Laminar Navier-Stokes Solution	21
Turbulent Navier-Stokes Solutions	22

VI. CONCLUSIONS	25
APPENDIX A: CODING OF MULTIGRID CYCLE USING RECURSION	26
APPENDIX B: DESCRIPTION OF ORIGINAL VISCOUS FLOW SOLVER	28
Governing Equations	28
Time Integration	29
Spatial Formulation	31
REFERENCES	32
FIGURES	34

LIST OF FIGURES

Figure 1 Schematic of two-level multigrid cycle.	34
Figure 2 Schematic of V cycle for four grid levels.	34
Figure 3 Schematic of W cycle for four grid levels.	34
Figure 4 Example of a fine-grid node P that will not contribute information to the coarse grid if linear interpolation is used.	35
Figure 5 Discretization of a curved boundary surface for both a fine and a coarse grid.	35
Figure 6 Effective interpolation near viscous surfaces. The diagonal edges cutting across the quadrilateral cells are omitted for visual clarity.	35
Figure 7 Effective interpolation near viscous surfaces for structured grids.	35
Figure 8 Median-dual control volume for node 0.	36
Figure 9 Contribution of an individual edge to the median-dual control volume for node 0.	36
Figure 10 Sample unstructured grid for a square.	37
Figure 11 Effect of relaxation scheme on multigrid performance. All runs are made using a 4-level V cycle with a direct solution on the coarsest mesh.	37
Figure 12 Effect of coarse grid convergence level on convergence of the multigrid cycle. Although not shown, convergence for $MITER = 10$ are virtually identical to those for $MITER \rightarrow \infty$	38
Figure 13 Effect of multigrid cycle on convergence. Both cases use 4 grid levels with 1 damped-Jacobi relaxation sweep on the coarsest grid.	38

Figure 14	Effect of number of grid levels on convergence of the multigrid scheme. All cases use a V cycle with 10 damped-Jacobi relaxation sweeps on the coarsest grid.	39
Figure 15	Portion of grid around a NACA 0012 airfoil.	39
Figure 16	Convergence history for nonlifting potential flow over a NACA 0012 airfoil at zero incidence.	40
Figure 17	Surface pressure coefficient distribution on a NACA 0012 airfoil at zero incidence.	40
Figure 18	Convergence histories for several V cycles versus both cycle number and computer time for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).	41
Figure 19	Convergence histories for several V cycles versus both cycle number and computer time for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).	41
Figure 20	Comparison of performance of V and W cycles versus both cycle number and computer time for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).	42
Figure 21	Effect of number of subiterations on performance of a 4-level W cycle for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).	42
Figure 22	Comparison of performance of V and W cycles for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).	43
Figure 23	Comparison of performance of V and W cycles for laminar flow over a NACA 0012 airfoil ($M_\infty = 0.5$, $\alpha = 3^\circ$, $Re = 5000$).	43
Figure 24	Grids around NACA 0012 airfoil used for laminar-flow case.	45

Figure 25 Comparison of performance of V and W cycles for turbulent flow over an RAE 2822 airfoil ($M_\infty = 0.75$, $\alpha = 2.81^\circ$, $Re = 6.2 \times 10^6$).	45
Figure 26 Comparison of surface pressure coefficient distribution with experiment on RAE 2822 airfoil ($M_\infty = 0.75$, $\alpha = 2.81^\circ$, $Re = 6.2 \times 10^6$).	46
Figure 27 Comparison of skin friction coefficient distribution with experiment on RAE 2822 airfoil ($M_\infty = 0.75$, $\alpha = 2.81^\circ$, $Re = 6.2 \times 10^6$).	46
Figure 28 Geometry of 3–element airfoil.	47
Figure 29 Comparison of performance of V and W cycles for turbulent flow over a 3–element airfoil ($M_\infty = 0.2$, $\alpha = 16.21^\circ$, $Re = 9 \times 10^6$).	47
Figure 30 Comparison of performance of V and W cycles for turbulent flow over a 3–element airfoil ($M_\infty = 0.2$, $\alpha = 21.34^\circ$, $Re = 9 \times 10^6$).	47
Figure 31 Comparison of performance of V and W cycles for turbulent flow over a 3–element airfoil ($M_\infty = 0.2$, $\alpha = 16.21^\circ$, $Re = 9 \times 10^6$) on a grid of 309,000 nodes.	48
Figure 32 Comparison of distributions of surface pressure coefficients for V cycle with experiment for turbulent flow over a 3–element airfoil ($M_\infty = 0.2$, $\alpha = 16.21^\circ$, $Re = 9 \times 10^6$) on a grid of 309,000 nodes.	48

LIST OF SYMBOLS

Δ	indicates a change in the associated quantity
α	angle of attack, relaxation parameter
γ	ratio of specific heats
η	cell aspect ratio ($\Delta x / \Delta y$)
μ	viscosity
ρ	density
$\tau_{xx}, \tau_{xy}, \tau_{yy}$	shear stress
ϕ	potential function
A	area, constant in equation of a plane
$[\mathbf{A}]$	Jacobian matrix for nonlinear system
B, C	constants in equation of a plane
C^*	constant in Sutherland's law
C_l	section lift coefficient
C_p	pressure coefficient
$[\mathbf{D}]$	diagonal matrix
E	total energy
$\vec{\mathbf{F}}$	flux vector
I	grid transfer operator
\mathbf{I}	identity matrix
L	left hand side operator
M	Mach number, edge weights
$[\mathbf{O}]$	matrix of off-diagonal terms
Pr	Prandtl number
\mathbf{Q}	vector of conserved variables
R, \mathbf{R}	residual
Re	Reynolds number
T	temperature
W_1, W_2, W_3	interpolation coefficients
a	speed of sound
f	forcing function

\mathbf{f}, \mathbf{g}	flux vector components
$h, 2h, 4h, \dots$	representative grid spacing
i	subiteration number
\hat{i}, \hat{j}	unit vectors in Cartesian coordinates
n	iteration/cycle number
$\hat{\mathbf{n}}$	unit normal vector
$\vec{\mathbf{n}}$	directed area
p	pressure
q	general field quantity
q_x, q_y	heat transfer terms
t	time
u	Cartesian velocity component, general solution vector
\tilde{u}	exact solution vector
v	Cartesian velocity component, error vector for linear multigrid
	scheme
x, y	Cartesian coordinates

Subscripts:

∞	free stream quantity
\mathbf{i}	inviscid
t	turbulence quantity
\mathbf{v}	viscous

I. INTRODUCTION

The recent increase in international competition in the commercial aircraft industry has resulted in a renewed interest in high-lift aerodynamics. High-lift configurations for commercial transports are characterized by multiple bodies generally in close proximity of each other, resulting in complex, highly viscous flows involving merging turbulent shear layers, laminar separation bubbles, and extensive regions of separated flow. The combination of landing and takeoff flight conditions and aircraft size result in a high Reynolds number flow at a relatively low free-stream Mach number. Circulation around these configurations can be so great that supersonic flow can exist even at low free-stream Mach numbers.

Given the above characteristics, it is clear that a computational method used to simulate such flows must itself have certain features. First, it must be able to deal with multiple bodies. This dictates that either unstructured grids or block-structured grids be used. Second, it must resolve the flow features, namely the merging shear layers, among others. This would require either extreme global grid refinement or an adaptive-grid capability. Third, it must take into account compressibility effects. Finally, to capture viscous effects, turbulence and transition modelling are required. While either grid methodology is capable of performing the above modelings, the focus of this study is on an unstructured-grid method.

Since it is relatively easy to implement an adaptive strategy with unstructured grids, these methods have been very popular for solving high-lift flows. Their main drawbacks, however, are the memory overhead associated with storing grid connectivity information and the computer time associated with indirect addressing. In addition, due to the lack of grid structure, it is difficult to implement simple implicit schemes such as approximate

factorization, while explicit schemes suffer from slow convergence. Present solvers are either implicit schemes utilizing iterative matrix solvers[1][2] or explicit schemes using acceleration techniques such as implicit residual smoothing and multigrid[3][4].

The multigrid method is particularly appealing since, in theory, the number of iterations required to reach a given level of convergence can be independent of the grid size. In other words, the work required to achieve a given level of convergence depends linearly on the number of grid points. Past efforts with structured-grid solvers have shown that remarkable gains in efficiency can be achieved through the use of a multigrid algorithm[5][6]. However, the implementation of a multigrid algorithm in an unstructured-grid environment is much more difficult. The lack of directionality and structure in the grid makes grid coarsening somewhat ambiguous, and intergrid transfers are not straightforward.

The recent effort of Mavriplis[3] to use multigrid with an unstructured-grid solver has been very successful. In his method, the solver and grid generator are closely coupled. Structured grids are generated around each solid body in the flow field, and the resulting points are overlaid and triangulated. For viscous meshes, the structured grid is maintained near the surface so that interpolation coefficients can be calculated. The scheme is a multistage Runge-Kutta scheme with residual smoothing and the various levels of grid refinement used in the multigrid algorithm are generated independently.

The work presented here is an implementation of multigrid acceleration for an existing implicit upwind solver using several of Mavriplis' techniques. Modifications have been made to eliminate the need for grid structure near the surface and hence uncouple the solver from the grid generation process. The details of this work are presented in the following chapters.

First is a more detailed description of multigrid methods as well as details of the

implementation used in this work. Next, a model problem is described that was used to test the multigrid method. Following this is a description of the original implicit upwind flow solver, followed by results showing the marked improvement in efficiency obtained by using multigrid.

II. MULTIGRID

Many relaxation schemes damp high-frequency error components relatively quickly, but are generally slower to damp low-frequency components. By interpolating the solution to a coarser grid, these low frequency errors appear as higher frequencies that can be damped well by the relaxation scheme. The coarse grid can be used to compute a correction to the fine-grid solution to eliminate its low-frequency errors. By using successively coarser grids recursively, lower and lower frequency components of the fine-grid solution error can be eliminated, and by performing a direct solution on the coarsest grid, the convergence rate of the multigrid cycle can be the same as that of the relaxation scheme for only the high frequencies.

In the following sections are descriptions of the basic multigrid methods for both linear and nonlinear equations, details of common multigrid cycles and their implementation using recursion, and specific details on the implementation of the intergrid transfer operators used in the current work.

Linear Systems

A system of linear equations can be written as

$$L(u) = f \tag{1}$$

where L is a linear operator, u is the solution vector, and f is a forcing function. The discrete approximation of the system on a grid characterized by spacing h is written

$$L^h(\tilde{u}^h) = f^h \tag{2}$$

where \tilde{u}^h is the exact solution to the discrete system. Let u^h be the current approximation to the exact solution \tilde{u}^h and now define the error v^h as

$$v^h = \tilde{u}^h - u^h \tag{3}$$

Now equation 2 can be written as

$$L^h(u^h + v^h) = f^h \quad (4)$$

which, since L is a linear operator, can be rewritten as

$$L^h(u^h) + L^h(v^h) = f^h \quad (5)$$

The error v^h can be represented on a coarser grid characterized by spacing $2h$ provided that it is sufficiently smooth to prevent aliasing of high-frequency components on the coarse grid.

An approximation to v^h can be calculated on the coarse grid by writing equation 5 for the coarse grid

$$L^{2h}v^{2h} = I_h^{2h}(f^h - L^h u^h) \quad (6)$$

where I_h^{2h} is referred to as the restriction operator, which transfers quantities from the fine grid h to the coarse grid $2h$. The implementation of this operator is described in a following section. Equation 6 can be simplified by defining $f^{2h} = I_h^{2h}(f^h - L^h u^h)$ to obtain

$$L^{2h}v^{2h} = f^{2h} \quad (7)$$

Once v^{2h} is obtained, the fine grid solution can be corrected using

$$(u^h)_{\text{new}} = (u^h)_{\text{old}} + I_{2h}^h v^{2h} \quad (8)$$

where I_{2h}^h transfers quantities from the coarse grid to the fine grid and is called the prolongation operator. Details of this operator are presented in a later section.

Low-frequency error components can be efficiently eliminated on coarse grids at a fraction of the cost of a fine grid calculation. Eliminating these error components on the

fine grid is very costly, as many more relaxation cycles are required than would be on the coarse grid. In addition, this process can be performed recursively on successively coarser grids (i.e. with spacings $4h$, $8h$, etc.) with each coarse grid used to compute a correction to the next higher grid level. Details on this recursive process are presented in the section titled “Multigrid Cycles.”

Nonlinear Systems

For systems of nonlinear equations, the step taken between equations 4 and 5 in the previous section cannot be performed, so a different formulation must be used. Following is a description of the Full Approximation Storage (FAS) scheme [7].

Starting with equation 2, subtract $L^h u^h$ from both sides to obtain

$$L^h(u^h + v^h) - L^h u^h = f^h - L^h u^h = R^h \quad (9)$$

Written for the coarse grid, this equation becomes

$$L^{2h}(I_h^{2h} u^h + v^{2h}) - L^{2h}(I_h^{2h} u^h) = I_h^{2h}(f^h - L^h u^h) \quad (10)$$

By rearranging terms and defining the coarse grid forcing function as

$$f^{2h} = I_h^{2h}(f^h - L^h u^h) + L^{2h}(I_h^{2h} u^h) \quad (11)$$

equation 10 can be written as

$$L^{2h} u^{2h} = f^{2h} \quad (12)$$

Once u^{2h} is calculated, the fine-grid solution is updated using

$$(u^h)_{\text{new}} = (u^h)_{\text{old}} + I_h^h [u^{2h} - I_h^{2h}(u^h)_{\text{old}}] \quad (13)$$

Note that the difference in formulations does not preclude the use of the recursive processes referred to in the preceding section and described in the next.

Multigrid Cycles

The recursive formulations referred to in the preceding sections are described below and closely follow those of Briggs [8]. A particular implementation of a recursive coarse-grid correction scheme is referred to as a multigrid cycle. Two specific multigrid cycles used in the present work are described along with a generalization of those cycles into a single procedure.

The simplest multigrid cycle is one involving only two levels. A schematic diagram of such a cycle is illustrated in figure 1. Each cycle begins with one or more relaxation sweeps on the fine grid. Next, the restriction operator transfers the residual and solution vectors to the coarse grid. One or more relaxation sweeps are then performed on the coarse grid. Finally, a correction is prolonged from the coarse grid to the fine grid and applied to the fine grid solution. This cycle works most efficiently when the coarse grid is solved directly, but for most cases, this is still impractical.

The two-level cycle can approach its maximum performance without a direct solution on the coarse grid by using what is known as a V cycle. The relaxation step on the coarse grid is now replaced by another two-level cycle. This repeats recursively until the coarsest grid available is reached. An example of the resulting cycle for four grid levels is illustrated in figure 2 . During the first half of the cycle at each intermediate grid level, residual and solution vectors are received from the finer grid, the solution is relaxed a given number of times, and residual and solution vectors are then passed to the coarser grid below. When the coarsest level is reached, a correction is passed successively upward until finally reaching the finest grid.

A further improvement to the two-level cycle can be made by replacing the coarse-grid relaxation with a pair of two-level cycles. This is again done successively at the

coarser levels. The resulting cycle is called a W cycle and an example having four grid levels is illustrated in figure 3.

The two cycles above can be combined into a single recursive procedure by specifying a cycle index, μ . The resulting generalized cycle is referred to as a μ cycle. Since the ultimate result of a multigrid cycle is a correction to the fine grid, the cycle can be expressed as a function whose parameters are the current residual and solution vectors and whose result is the new solution vector. Stated mathematically,

$$u^h \leftarrow \mu^h \left(R^h, u^h \right) \quad (14)$$

The μ cycle can now be described by the following recursive procedure:

1. Relax n times on grid h .
2. If grid h is not the coarsest level,
 - a. Restrict residual and solution vectors to grid $2h$.
 - b. Evaluate $u^{2h} \leftarrow \mu^{2h} \left(R^{2h}, u^{2h} \right)$ μ times.
 - c. Prolong correction from grid $2h$ to grid h .

Appendix A gives details of the coding of this procedure.

Intergrid Transfers

The restriction of the solution from a fine grid to a coarser grid and the prolongation of the correction from a coarse grid to a finer grid both utilize bilinear interpolation. The procedure is described here for the restriction operator, but the process is identical for prolongation.

To get information to the coarse grid, a bilinear interpolation is performed using the data at the three vertices of the fine grid cell that encloses each coarse grid node. The coordinates of the vertices along with the quantity being interpolated form a plane in

a three-dimensional space. Finding the value of the transferred quantity at the coarse grid node amounts to solving the equation of the plane at that node. This plane can be expressed mathematically as:

$$Ax + By + C = q \quad (15)$$

where A, B, C are constant coefficients and q is the quantity being transferred. The coefficients are determined by assembling a system of equations using the data at the cell vertices as follows:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (16)$$

where the subscript denotes a particular vertex of the fine grid cell. Solving this system yields the following expressions for the coefficients:

$$A = \frac{q_1(y_2 - y_3) + q_2(y_3 - y_1) + q_3(y_1 - y_2)}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)} \quad (17)$$

$$B = \frac{q_1(x_3 - x_2) + q_2(x_1 - x_3) + q_3(x_2 - x_1)}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)} \quad (18)$$

$$C = \frac{q_1(x_2y_3 - x_3y_2) + q_2(x_3y_1 - x_1y_3) + q_3(x_1y_2 - x_2y_1)}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)} \quad (19)$$

When equation 15 is evaluated using the coordinates of the coarse grid nodes, it is apparent that the value of the quantity at the coarse grid node may be written as the sum of geometric weights multiplied by the values of the quantity at the vertices of the enclosing fine grid cell, i.e.

$$q_n = W_1q_1 + W_2q_2 + W_3q_3 \quad (20)$$

Given the coordinates (x_n, y_n) of the coarse grid node and the coefficients A, B, C substituted into equation 15, the quantities W_1, W_2, W_3 are found by inspection to be

$$W_1 = \frac{x_n(y_2 - y_3) + y_n(x_3 - x_2) + (x_2y_3 - x_3y_2)}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)} \quad (21)$$

$$W_2 = \frac{x_n(y_3 - y_1) + y_n(x_1 - x_3) + (x_3y_1 - x_1y_3)}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)} \quad (22)$$

$$W_3 = \frac{x_n(y_1 - y_2) + y_n(x_2 - x_1) + (x_1y_2 - x_2y_1)}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)} \quad (23)$$

It is easily verified that the sum of these three weights is unity.

If the linear interpolation outlined above is used for the transfer of residuals from a fine grid to a coarser grid, a situation may arise as shown in figure 4, where a nonzero residual at fine grid node P is not utilized on the coarse grid, since none of the fine-grid cells having node P as a vertex enclose any coarse-grid nodes; hence, much of the benefit of multigrid is lost. In addition, the residual is actually the surface integral of the fluxes around the boundary of the control volume and is therefore related to the time rate of change of the conserved variables. In order for this rate of change to be the same for all grids, it is necessary that the residual transfer be conservative, that is, that the sums of the residuals on the fine and coarse grids be equal. For these reasons, the restriction process for residuals is handled in the following manner.

For a given fine grid node, the coarse grid cell that surrounds the node is determined. The residual for the fine grid node is then distributed to the vertices of the surrounding coarse grid cell. The weights used for the distribution are the same weights used in the linear interpolation from the coarse grid to the fine grid. This process ensures that all fine-grid residuals contribute to the coarse grid, and that the total residual is conserved, since the weights multiplying the residual at any given fine-grid node sum to unity.

To transfer information from one grid to another using the above interpolation operators requires knowledge of which cell of one grid encloses each node of the other grid. To determine this information, a tree search similar to that used by Mavriplis[9] is used. In this procedure, the nodes of the first grid are first ordered into a list such that

a given node has an immediate neighbor appearing earlier in the list. The search then proceeds as described in the following paragraph.

For the first node, an arbitrary cell of the second grid is chosen to start the search. If the cell does not enclose the node, the immediate neighbors of the cell are added to a list of cells to check (provided the neighboring cells have not already been checked). Next, the neighbors of the neighboring cells are checked, and so on until the enclosing cell is found. For the remaining nodes, the cell enclosing a neighboring node appearing earlier in the list of nodes (i.e. one whose enclosing cell is known) is used as a starting point for the search.

The search algorithm can encounter problems near boundaries, where the grid is actually a planar discretization of a curved surface, as illustrated in figure 5. Alternating nodes on the fine-grid boundary are displaced away from the coarse-grid boundary. The problem is aggravated by concave surfaces and highly stretched viscous meshes, where several interior nodes may lie outside the coarse-grid domain, and fine-grid interior nodes very close to the surface may receive interpolated information from coarse-grid cells farther away from the surface. This latter case is illustrated in figure 6.

Simple structured-grid algorithms perform intergrid transfers in computational space, where grid lines match nicely and operators are straightforward. The equivalent situation in physical space is illustrated in figure 7. The following procedure is a way of approximating this behavior on unstructured grids by preserving the distance to the boundary for each node in a prescribed region. The procedure is described for a fine- to coarse-grid transfer. The reverse operation is similar.

First, the list of boundary faces for each grid is sorted such that adjacent faces are in order and the face with the greatest x -coordinate is first in the list. Then, starting at the first boundary face of each grid, the boundaries are matched by determining which coarse-

grid face is closest to each fine-grid node. Each fine-grid boundary node is assigned interpolation coefficients by projecting the node onto the coarse-grid face and computing a linear interpolation along the face. The physical displacement required to move the fine-grid node to the coarse-grid face is also stored for later use.

Next, a region near the fine-grid surface is defined in which nodes will be shifted to maintain their position relative to the boundary. This is done by first tagging the nodes of the grid lying on viscous boundaries. The edges of the grid are then cycled through a prescribed number of times. Each cycle through the edges, nodes neighboring tagged nodes are themselves tagged. The result for the particular case of a triangulated structured grid is that a certain number of layers of grid points have been tagged. In general, nodes in a region surrounding the viscous boundary nodes will be tagged.

Each tagged node is then associated with the boundary face it is nearest. The node is projected onto the boundary face, and the previously computed boundary-node displacements are then used to determine the displacement to be applied to the interior node via linear interpolation. Note that these shifted node coordinates are used only in the above procedure for calculating the interpolation coefficients and not in the rest of the flow calculation.

III. THE MODEL PROBLEM — LAPLACE’S EQUATION

For the purposes of developing and debugging code for the multigrid cycle and its associated intergrid transfer mechanisms, it is beneficial to decouple the difficulties associated with the numerics of a problem involving a system of nonlinear equations from those associated strictly with information transfer between grids. To this end, the two-dimensional Laplace equation ($\nabla^2\phi = 0$) is coded and used as a test vehicle. The boundary conditions are chosen such that the problem represents two-dimensional nonlifting potential flow. Namely, $\frac{\partial\phi}{\partial n} = 0$ on the airfoil surface(s) and $\phi = x$ at the outer boundary (i.e. free stream conditions).

Spatial Discretization

The equation is solved in integral form, i.e.

$$\int_S \nabla^2\phi dA = 0 \quad (24)$$

At each node, the equation is integrated over a control volume consisting of the cells surrounding the node. This control volume surrounding a given node 0 is shown as the dark outline in figure 8 and is known as the median-dual control volume. The median dual mesh around a given node is formed by joining the centroids of the cells surrounding the node and the midpoints of the edges joining to the node. Using Green’s theorem, the integral equation becomes

$$\oint_C \nabla\phi \cdot \hat{\mathbf{n}} dl = 0 \quad (25)$$

and the integral over the control volume can be evaluated by integrating around its boundary.

In reference 1, it is shown that integrating over the control volume described above is equivalent to a Galerkin finite-element approximation. Now contributions from each edge

that connects to a given node can be considered individually given the nodes that form the edge and the centroids of the cells on either side of the edge. Each edge contributes two segments to the boundary of the median-dual control volume, as shown in figure 9.

The Laplacian operator can now be discretized as the sum of weights associated with edges of the mesh multiplied by the difference in the solution along the edge, i.e.

$$(\nabla^2 \phi)_0 = \sum_{i \in \chi_0} M_i (\phi_i - \phi_0) \quad (26)$$

where χ_0 is the set of edges connecting to node 0 and M_i is the edge weight. Reference 1 presents a detailed derivation of these edge weights for the Laplacian operator, and a brief summary of the results is presented here. Using the notation of figure 9, the edge weights can be expressed as:

$$M_i = -\frac{1}{4} \left[\frac{\hat{n}_{i+1/2} \cdot \hat{n}_{i+1}}{A_{i+1/2}} - \frac{\hat{n}_{i-1/2} \cdot \hat{n}_{i-1}}{A_{i-1/2}} \right] \quad (27)$$

where $A_{i-1/2}$ and $A_{i+1/2}$ are the areas of the cells to the right and left of the edge connecting node 0 and node i , respectively. Identities are then used to express this in terms of the single edge. The resulting expression is:

$$M_i = -\frac{1}{4} \left[\frac{\frac{1}{4} \hat{n}_i \cdot \hat{n}_i - 9 \hat{n}_{R_i} \cdot \hat{n}_{R_i}}{A_{i+1/2}} + \frac{\frac{1}{4} \hat{n}_i \cdot \hat{n}_i - 9 \hat{n}_{L_i} \cdot \hat{n}_{L_i}}{A_{i-1/2}} \right] \quad (28)$$

Note that these weights depend only on the geometry and not on the solution, so they can be precomputed and stored for the duration of the calculation.

Iteration Scheme

Jacobi iteration with a relaxation parameter is used to advance the solution. Starting with equation 26, the residual at iteration n is defined as

$$R_0^n = \sum_{i \in \chi_0} M_i (\phi_i^n - \phi_0^n) \quad (29)$$

Laplace's equation is now discretized using ϕ_0 at iteration $n + 1$ and ϕ_i at iteration n as follows:

$$\sum_{i \in \chi_0} M_i (\phi_i^n - \phi_0^{n+1}) = 0 \quad (30)$$

This equation is then solved for ϕ_0^{n+1} to yield

$$\phi_0^{n+1} = \frac{\sum_{i \in \chi_0} M_i \phi_i^n}{\sum_{i \in \chi_0} M_i} \quad (31)$$

Now subtract ϕ_0^n from both sides to yield:

$$\phi_0^{n+1} - \phi_0^n = \frac{\sum_{i \in \chi_0} M_i (\phi_i^n - \phi_0^n)}{\sum_{i \in \chi_0} M_i} \quad (32)$$

The increment to the solution $\Delta\phi_0^n$ is then calculated by:

$$\Delta\phi_0^n = \phi_0^{n+1} - \phi_0^n = \frac{R_0^n}{\sum_{i \in \chi_0} M_i} \quad (33)$$

Over- or underrelaxation is accomplished simply by adding a relaxation parameter α as follows:

$$\Delta\phi_0^n = \alpha \frac{R_0^n}{\sum_{i \in \chi_0} M_i} \quad (34)$$

If α is greater than unity, the solution is overrelaxed, and if it is less than unity (but greater than zero), the solution is underrelaxed. Of course, Jacobi iteration is recovered when α is equal to unity.

Alternately, a red-black scheme may be used. In this scheme, the grid is divided into two "colors" — red and black — depending on whether the node number is even or odd, respectively. The increment to the solution is calculated as follows:

1. Calculate the residual for all nodes.
2. Update nodes colored red.
3. Recalculate the residual for all nodes.
4. Update nodes colored black.

The scheme requires an additional residual calculation at each iteration, but exhibits better smoothing properties from the perspective of multigrid methods.[8]

Results

The Laplace solver was run in different modes for two cases. The first case is the simple case of a square with homogeneous Dirichlet boundary conditions on all its edges. The grids consisted of Cartesian meshes with each cell cut across one of its diagonals to form triangles. An example of one of these meshes is shown in figure 10. An initial condition of unit potential was used on the interior grid. Finally, the solver was used to simulate nonlifting potential flow over a NACA 0012 airfoil at zero angle of attack.

Square Domain

The first study, shown in figure 11, is a demonstration of the effect of the relaxation scheme on the performance of the multigrid scheme. All three cases were run with a direct solution on the coarsest grid and used a 4-level V cycle. Clearly, the Jacobi scheme is inferior, as it poorly damps the high frequency components of the solution error. The damped Jacobi scheme ($\alpha=0.5$) exhibits better performance since underrelaxation greatly improves the damping of the high frequencies[8]. The red-black scheme shows excellent performance. The damped Jacobi scheme is used for the remaining studies on the square domain.

In practice, a direct solution on the coarsest grid is not used, but by performing more relaxations on the coarse grid, improving the cycle, and using additional grid levels, the performance of the scheme can approach that which is achieved by a direct solution on the coarse grid. The following studies illustrate this point.

The next study shows the effect of the level of convergence of the coarsest grid on the convergence rate of the multigrid scheme. Figure 12 shows convergence histories for

a 4-level V cycle. The parameter $MITER$ is the number of damped Jacobi relaxation sweeps performed on the coarsest mesh during each multigrid cycle. The curve marked $MITER \rightarrow \infty$ corresponds to a direct solution on the coarsest grid. It is obvious that a relatively meager increase in the number of relaxation sweeps on the coarsest grid has a profound impact on the convergence of the multigrid scheme. Although not shown, the convergence history for $MITER = 10$ is virtually identical to that of a direct solution on the coarsest mesh. These extra sweeps on the coarsest mesh are cheap in terms of computational work in comparison with the finest mesh. It will be shown, however, that the same increase in performance can be achieved by changing the multigrid cycle.

Figure 13 shows that using the W cycle instead of the V cycle results in the same improved convergence of the multigrid cycle. Both cases use 4 grid levels with 1 damped-Jacobi relaxation sweep on the coarsest mesh. The residual is now plotted against computer time to reflect the actual computational work done, since a W cycle requires more computational work per multigrid cycle than does a V cycle.

In figure 14, the benefit of using additional grid levels in the multigrid cycle is apparent. All cases use a V cycle with 10 damped-Jacobi relaxation sweeps on the coarsest grid. Each level of coarsening allows the relaxation scheme to damp lower and lower frequency components of the solution error more effectively. This figure along with figure 12 shows that the scheme can approach its best performance without a direct solution on the coarsest grid.

NACA 0012 Airfoil

To demonstrate the ability of the Laplace solver to calculate nonlifting incompressible flows, results for a NACA 0012 airfoil at zero incidence are presented. The red-black scheme has been used for this case with one relaxation sweep at each grid level. At the

airfoil surface, the derivative of the potential normal to the surface vanishes, and free stream potential is enforced at the far field boundary. Four levels of grid refinement were generated, the finest consisting of 14,269 nodes. A portion of this grid is shown in figure 15. Table 1 summarizes the grid levels used for these cases.

Table 1.— Summary of grid sizes for grids around NACA 0012 airfoil for inviscid flow.

Grid level	Total nodes	Nodes on surface
0	14,269	256
1	3796	128
2	1081	65
3	424	36

Figure 16 shows convergence histories with and without multigrid. The multigrid algorithm has clearly provided a substantial improvement in the rate of convergence.

A comparison of surface pressure coefficient is presented in figure 17 along with an analytical solution[10]. The solution agrees well with the analytical data.

IV. VISCOUS FLOW SOLVER

The solver used in the present work was developed by W. Kyle Anderson at the NASA Langley Research Center. A summary of the salient features of the code are presented in the following chapter for completeness. A more detailed description of the solver is presented in appendix B, and further information can be found in reference 11.

The governing equations are the time-dependent two-dimensional Reynolds-averaged Navier-Stokes equations in conservation-law form, which are integrated in time to obtain a steady-state solution. Ideal-gas assumptions are made, and either of the one-equation turbulence models of Spalart and Allmaras[12] or Baldwin and Barth[13] may be used for calculating turbulent flows.

The temporal formulation consists of a backward-Euler time difference, with the resulting linear system of discrete equations being solved iteratively using red-black relaxation. The result is that at each iteration of the nonlinear system, a prescribed number of “subiterations” are performed to obtain an approximate solution to the linear system. A finite-volume formulation is used to discretize the governing equations in space at each node. For the convective and pressure terms, the upwind scheme of Roe[14] is used, while a simple central difference is used for the viscous terms.

The same scheme is used to solve the turbulence equation; however, this calculation is carried out separately. At each iteration, a prescribed number of subiterations are performed on the flow equations while holding the turbulence fixed, followed by an update of the turbulence equation holding the flow quantities fixed.

V. RESULTS

The following chapter presents results in the form of histories of the temporal convergence of both the L_2 norm of the residual for the continuity equation and the lift coefficient, C_l . These quantities are shown versus computer time. Unless otherwise noted, all cases were run on a Cray-YMP, and grids were generated using the advancing-front method described in reference 15.

First, cases of inviscid and laminar flow over a NACA 0012 airfoil are presented, followed by several cases of turbulent flow over both the RAE 2822 airfoil and a 3-element airfoil. Finally, results are presented for a turbulent case on a grid for which a calculation without multigrid is impractical. All turbulent cases used the Spalart-Allmaras turbulence model.

Euler Solution

The first case presented is that of inviscid, transonic flow over a NACA 0012 airfoil. The free stream Mach number is 0.8, the angle of attack is 1.25° , and the same grids that were used for the potential-flow case presented earlier are used (see table 1). Figure 18 shows the convergence histories for several V cycles in comparison to the original scheme. Note that the multigrid scheme substantially improves the convergence rate, and that the improvement increases as more grid levels are used. Results for the W cycle are shown in figure 19. Note again the substantial improvement in convergence. Figure 20 shows the best V-cycle result and the best W-cycle result together with the results for the scheme without multigrid. The V cycle and W cycle perform similarly versus computer time; however, the relaxation parameters used were those found to work well for the base scheme (i.e. without multigrid), specifically, 20 subiterations were used.

If the scheme is used without multigrid, the tunable parameters (i.e. CFL number and number of subiterations) must be chosen to give the fastest convergence. With a multigrid method, only the high-frequency error components need to be damped quickly on all but the coarsest grid. This would seem to imply that with the present scheme, a further reduction in computer time could be achieved by reducing the number of subiterations. Figure 21 shows the effect of reducing the number of subiterations for a four-level W cycle. While convergence per cycle is slightly compromised, convergence versus computer time is improved due to the decrease in computational work per cycle.

Residual and lift histories for the V and W cycles using five subiterations at each grid level are shown in figure 22. In this case, the W cycle slightly outperforms the V cycle, while both obtain a steady value for the lift approximately four times faster than the base scheme alone.

Laminar Navier-Stokes Solution

Figure 23 shows convergence histories for a case of laminar flow over a NACA 0012 airfoil at an angle of attack of 3° , a free-stream Mach number of 0.5, and a Reynolds number of 5000. The grids used in the multigrid cycle are shown in figure 24 and are summarized in table 2. The two multigrid cycles achieve a steady lift coefficient in a

Table 2.— Summary of grid sizes for grids around a NACA 0012 airfoil for laminar flow.

Grid level	Total nodes	Nodes on surface
0	16,116	256
1	5004	128
2	1891	65
3	1237	40

fraction of the time taken by the original solver, and the W cycle has a slight edge over the V cycle, particularly in the convergence of the lift coefficient.

Turbulent Navier-Stokes Solutions

After presenting a case of transonic flow over an RAE 2822 airfoil, several cases of flow past a 3–element airfoil are shown, including a case previously impractical to calculate. For all 3–element cases presented, the free-stream Mach number is 0.2 and the Reynolds number is 9 million.

Figure 25 shows convergence histories for flow past an RAE 2822 airfoil at 2.81° angle of attack, free-stream Mach number of 0.75, and a Reynolds number of 6.2 million. The grids were generated using the method described in reference 16, and a summary of their characteristics is presented in table 3. The residual for both multigrid cases

Table 3.— Summary of grid sizes for grids around RAE 2822 airfoil for turbulent flow.

Grid level	Total nodes	Nodes on surface
0	13,385	208
1	3359	104
2	847	52
3	219	26

converges a few orders of magnitude before cycling about a nearly constant level. Other runs have shown that this phenomena is a result of an adverse coupling of multigrid and the turbulence model, as holding the turbulence quantity constant after some level of convergence has been reached causes the residual to continue decreasing. Steady lift for both multigrid cases is still achieved prior to the cycling of the residual and in significantly less time than for the original scheme.

Figures 26 and 27 show the distributions of the surface pressure coefficient and skin friction coefficient, respectively, for the 4–level W cycle along with experimental data[17]. The computed results are in good agreement with the experimental data, and are virtually identical to results obtained with the base scheme.

A case of turbulent flow over the 3–element airfoil shown in figure 28 at 16.21° is shown in figure 29, and the characteristics of the grids are summarized in table 4. The multigrid cases again cycle about some level after a certain level of convergence is

Table 4.— Summary of grid sizes for grids around a 3–element airfoil for turbulent flow.

Grid level	Total nodes	Nodes on surfaces
0	97,088	1340
1	34,987	671
2	14,278	340
3	6657	178

reached. The lift, however, converges for all three cases and does so much more rapidly for the multigrid cases, with the W cycle having a significant edge.

The same configuration at a higher angle of attack is shown in figure 30. The precise angle of attack is 21.34° and is near maximum lift as determined by experiment[18]. The spikes in the residual histories are a result of restarting the code. Specifically, a point-vortex is applied at the outer boundary whose strength depends on the lift, which is not available during the first iteration since it is presently calculated after the residual. This is easily cured by computing the lift before computing the residual. Note that for this run, the V-cycle case continues converging while the W-cycle residual again cycles after less than two orders of magnitude of convergence. The multigrid scheme again shows considerable improvement over the base scheme.

To further demonstrate the advantages of multigrid, the 3–element airfoil was run at an angle of attack of 16.21° on a grid consisting of 309,770 nodes. The characteristics of the full set of grids is given in table 5. This case had been considered impractical

Table 5.— Summary of grid sizes for grids around a 3–element airfoil for turbulent flow.

Grid level	Total nodes	Nodes on surfaces
0	309,770	2679
1	97,088	1340
2	34,987	671
3	14,278	340

with the original solver due to the nonlinear increase in computer time required with the increase in the number of grid points. The convergence histories are shown in figure 31. Computer restrictions dictated that only 50 cycles could be calculated in a single run. As explained earlier, the spikes in the convergence histories are a result of restarting the code. The W cycle exhibits oscillatory behavior in the residual, while the V cycle continues converging. The lift seems nearly steady for the W cycle, but when viewed on a smaller scale, it exhibits small-scale oscillations. The lift for the V cycle, however, is steady, and the surface pressure distributions for this case are presented in figure 32.

VI. CONCLUSIONS

A multigrid algorithm has been implemented in an existing code for solving turbulent flows on triangular meshes. Intergrid transfer operators have been used that ensure conservation of the residual and preserve smoothness of the solution near solid surfaces. Once coded, the multigrid algorithm and intergrid transfer operators were used to solve Laplace's equation to verify correct operation.

The Laplace solver with the red-black relaxation scheme and multigrid algorithm is very efficient for solving nonlifting potential flow on unstructured grids, and was indispensable for validating intergrid transfer operators and the multigrid cycle itself.

The multigrid algorithm has improved convergence significantly for both inviscid and laminar viscous flows. For the turbulent flows, the improvement with multigrid can be quite dramatic, with increasing improvement with grid refinement.

Several avenues of future study exist as a result of this work. The apparent adverse interaction between the W cycle and the turbulence model will require a significant effort to resolve. The method can also be extended to three-dimensions, or to higher-order methods.

APPENDIX A: CODING OF MULTIGRID CYCLE USING RECURSION

Since the main driver of the flow solver is written in C, which allows a function to call itself recursively, implementation of the μ cycle described in the text is very straightforward, and can be nearly literally translated into C-code. Following is the code fragment representing the multigrid cycle:

```
mucyc(mu,ifine,igrid1,igrid2,grid,miter)
  GRID *grid;
  int mu,ifine,igrid1,igrid2;
  int *miter;
  {
  int i,j;

  relax(miter[igrid1],grid[igrid1]);
  if (igrid1 == ifine)
    {
    f77L2NORM();
    f77FORCE();
    }

  if (igrid1 < igrid2)
    {
    restricter(grid[igrid1],grid[igrid1+1]);

    for (i = 0; i < mu; ++i)
      {
      mucyc(mu,ifine,igrid1+1,igrid2,grid,miter);
      }

    prolong(grid[igrid1],grid[igrid1+1]);
    }
  }
```

In this routine, `f77L2NORM` and `f77FORCE` are FORTRAN routines that calculate quantities used to monitor convergence on the fine grid. These two routines have many arguments, but they are omitted here for clarity. Note that grids are denoted by index numbers (0, 1, 2, ...) rather than characteristic spacings ($h, 2h, 4h, \dots$). The parameter `mu`

is the cycle index, while `igrid1` and `igrid2` are the finest and coarsest grid levels in the cycle, respectively. The parameter `ifine` is a copy of the initial value of `igrid1`. The argument `grid` is an array of structures having one entry for each grid level. Each structure contains parameters indicating the size of the corresponding grid, as well as pointers to arrays containing connectivity and field information. The argument `miter` is an array containing the number of relaxation sweeps to be performed at each grid level.

APPENDIX B: DESCRIPTION OF ORIGINAL VISCOUS FLOW SOLVER

Governing Equations

The relaxation scheme solves the Reynolds-Averaged Navier-Stokes (RANS) equations in conservation-law form. These equations are given in vector form by

$$A \frac{\partial \mathbf{Q}}{\partial t} + \oint_{\partial\Omega} \vec{\mathbf{F}}_i \cdot \hat{\mathbf{n}} dl - \oint_{\partial\Omega} \vec{\mathbf{F}}_v \cdot \hat{\mathbf{n}} dl = 0 \quad (35)$$

where $\hat{\mathbf{n}}$ is the outward-pointing unit normal to the surface of the control volume $\partial\Omega$. \mathbf{Q} is the vector of conserved state variables given by

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad (36)$$

and $\vec{\mathbf{F}}_i$ and $\vec{\mathbf{F}}_v$ are the inviscid and viscous fluxes, respectively, through the surface of the control volume $\partial\Omega$ and are given by

$$\vec{\mathbf{F}}_i = \mathbf{f} \hat{i} + \mathbf{g} \hat{j} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{bmatrix} \hat{i} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ (E+p)v \end{bmatrix} \hat{j} \quad (37)$$

$$\vec{\mathbf{F}}_v = \mathbf{f}_v \hat{i} + \mathbf{g}_v \hat{j} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix} \hat{i} + \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{bmatrix} \hat{j} \quad (38)$$

The shear stress and heat conduction terms in the viscous fluxes are given by

$$\tau_{xx} = (\mu + \mu_t) \frac{M_\infty}{Re} \frac{2}{3} [2u_x - v_y] \quad (39)$$

$$\tau_{yy} = (\mu + \mu_t) \frac{M_\infty}{Re} \frac{2}{3} [2v_y - u_x] \quad (40)$$

$$\tau_{xy} = (\mu + \mu_t) \frac{M_\infty}{Re} [u_y + v_x] \quad (41)$$

$$q_x = \frac{-M_\infty}{Re(\gamma - 1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial a^2}{\partial x} \quad (42)$$

$$q_x = \frac{-M_\infty}{Re(\gamma - 1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{\partial a^2}{\partial y} \quad (43)$$

The perfect-gas equation of state is used to define the pressure p and is given by

$$p = (\gamma - 1) [E - \rho(u^2 + v^2)/2] \quad (44)$$

and the laminar viscosity μ is given by Sutherland's law

$$\frac{\mu}{\mu_\infty} = \frac{(1 + C^*)}{(T/T_\infty + C^*)} (T/T_\infty)^{3/2} \quad (45)$$

where $C^* = \frac{198.6}{460.0}$ is Sutherland's constant divided by a free-stream reference temperature assumed to be 460° Rankine.

The eddy viscosity μ_t is obtained by either of two one-equation turbulence closure models. The first, developed by Baldwin and Barth[13], is derived from the k - ϵ equations. The second, developed by Spalart and Allmaras[12], relies more heavily on empiricism and dimensional analysis. The turbulence model is solved separately from the rest of the system, but uses the same solution scheme, and, although multigrid is also used with the turbulence model, it remains decoupled from the rest of the system. The Spalart-Allmaras model is used for all turbulent calculations in this study.

Time Integration

The governing equations are integrated in time to the steady-state solution using a linearized backward-Euler time-differencing scheme. The resulting system of linear equations can be expressed as

$$[\mathbf{A}]^n \{\Delta \mathbf{Q}\}^n = \{\mathbf{R}\}^n \quad (46)$$

where

$$[\mathbf{A}]^n = \frac{A}{\Delta t} \mathbf{I} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}} \quad (47)$$

The solution of this linear system is obtained iteratively via a classic relaxation procedure. To differentiate between the nonlinear and linear systems, the term “iteration” is used to refer to the nonlinear system, while “subiteration” is used to refer to the linear system.

To illustrate the scheme used, let the matrix $[\mathbf{A}]^n$ be written as the sum of two matrices representing the diagonal and off-diagonal terms

$$[\mathbf{A}]^n = [\mathbf{D}]^n + [\mathbf{O}]^n \quad (48)$$

The simplest method for solving the linear system is commonly referred to as Jacobi iteration and consists of moving all off-diagonal terms to the right-hand side and evaluating them at the previous subiteration i . This can be written as

$$[\mathbf{D}]^n \{\Delta \mathbf{Q}\}^{i+1} = \left[\{\mathbf{R}\}^n - [\mathbf{O}]^n \{\Delta \mathbf{Q}\}^i \right] \quad (49)$$

The convergence of this method is accelerated somewhat using a red-black scheme where even-numbered nodes are update using the Jacobi scheme just described, followed by the odd numbered nodes using the update values at the even-numbered nodes. This scheme can be written as

$$[\mathbf{D}]^n \{\Delta \mathbf{Q}\}^{i+1} = \left[\{\mathbf{R}\}^n - [\mathbf{O}]^n \{\Delta \mathbf{Q}\}_i^{i+1} \right] \quad (50)$$

where $\{\Delta \mathbf{Q}\}_i^{i+1}$ is the most recent value of $\{\Delta \mathbf{Q}\}$ and will be at subiteration $i + 1$ for the even-numbered nodes and at subiteration i for the odd-numbered nodes.

To further accelerate convergence, local time-stepping is used. A separate time step is calculated at each node using the inviscid stability limit.

Spatial Formulation

The spatial discretization is a finite-volume formulation in which the inviscid and viscous fluxes are integrated over the median-dual control volume surrounding each node (see figure 8). Green's theorem is used to change the volume integrals to surface integrals over the edges of the dual mesh. These surface integrals can be calculated using edge formulas as described in reference 1.

The inviscid fluxes, \vec{F}_i , are obtained on the edges of the control volume using Roe's approximate Riemann solver[14]. The viscous fluxes, \vec{F}_v , are computed using a simple central difference.

REFERENCES

1. BARTH, TIMOTHY J. “Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes”. AIAA 91-0721, January 1991.
2. VENKATAKRISHNAN, V. AND MAVRIPLIS, DIMITRI J. “Implicit Solvers for Unstructured Meshes”. AIAA 91-1537-CP, June 1991.
3. MAVRIPLIS, DIMITRI J., JAMESON, ANTONY, AND MARTINELLI, LUIGI. “Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes”. NASA CR-181786, February 1989.
4. DAVIS, WARREN H. AND MATUS, RICHARD J. “High Lift Multiple Element Aifoil Analysis with Unstructured Grids”. AIAA 93-3478-CP, 1993.
5. ANDERSON, W. KYLE, THOMAS, JAMES L., AND WHITFIELD, DAVID L. “Three-Dimensional Multigrid Algorithms for the Flux-Split Euler Equations”. NASA TP 2829, November 1988.
6. VATSA, VEER N. AND WEDAN, BRUCE W. “Development of an Efficient Multigrid Code for 3-D Navier-Stokes Equations”. AIAA 89-1791, June 1989.
7. SOUTH, JR., JERRY C. AND BRANDT, ACHI. “Application of a Multilevel Grid Method to Transonic Flow Calculations”. In Adamson, T. C. and Platzer, M. C., editors, *Transonic Flow Calculations in Turbomachinery*. Hemisphere Publications, 1977.
8. BRIGGS, WILLIAM L. *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, 1987.
9. MAVRIPLIS, DIMITRI J. “Multigrid Solution of the 2-D Euler Equations on Unstructured Triangular Meshes”. *AIAA Journal*, 26(7):824-831, July 1988.

10. ABBOTT, IRA H. AND VON DOENHOFF, ALBERT E. *Theory of wing sections, including a summary of airfoil data*. Dover Publications, 1959.
11. ANDERSON, W. KYLE AND BONHAUS, DARYL L. “Navier-Stokes Computations and Experimental Comparisons for Multielement Airfoil Configurations”. AIAA 93-0645, January 1993.
12. SPALART, PHILLIPE R. AND ALLMARAS, STEVEN R. “A One-Equation Turbulence Model for Aerodynamic Flows”. AIAA 92-0439, January 1992.
13. BALDWIN, BARRET S. AND BARTH, TIMOTHY J. “A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows”. AIAA 91-0610, January 1991.
14. ROE, P. “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes”. *Journal of Computational Physics*, 43:357-372, 1981.
15. PIRZADEH, SHAHYAR. “Structured Background Grids for Generation of Unstructured Grids by Advancing Front Method”. AIAA 91-3233, September 1991.
16. MAVRIPLIS, DIMITRI J. “Unstructured and Adaptive Mesh Generation for High Reynolds Number Viscous Flows”. NASA CR-187534, February 1991.
17. COOK, P. H., MCDONALD, M. A., AND FIRMIN, M. C. P. “Aerofoil RAE 2822 — Pressure Distributions and Boundary Layer and Wake Measurements”. AGARD AR-138, 1979.
18. VALAREZO, W. O., DOMINIK, C. J., MCGHEE, R. J., GOODMAN, W. L., AND PASCHAL, K. B. “Multi-Element Airfoil Optimization for Maximum Lift at High Reynolds Numbers”. AIAA 91-3332, September 1991.

FIGURES

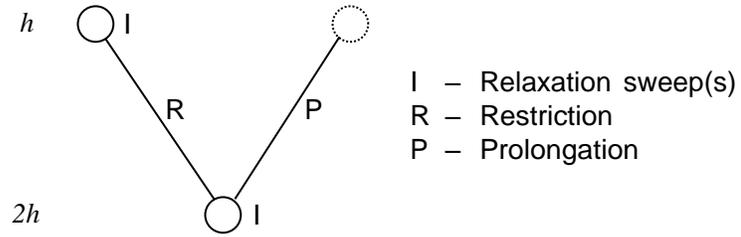


Figure 1: Schematic of two-level multigrid cycle.

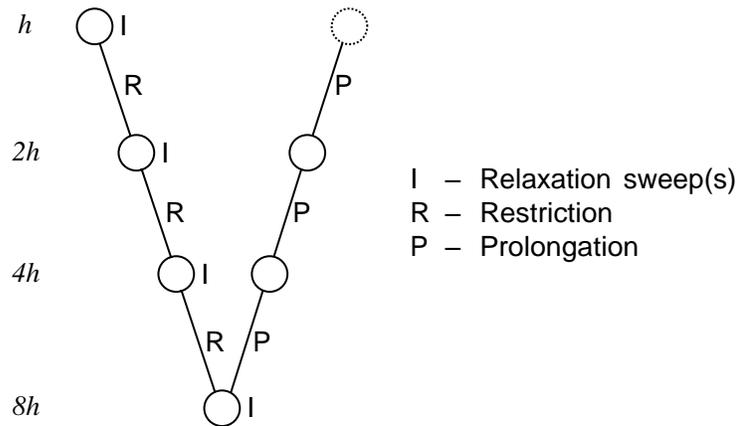


Figure 2: Schematic of V cycle for four grid levels.

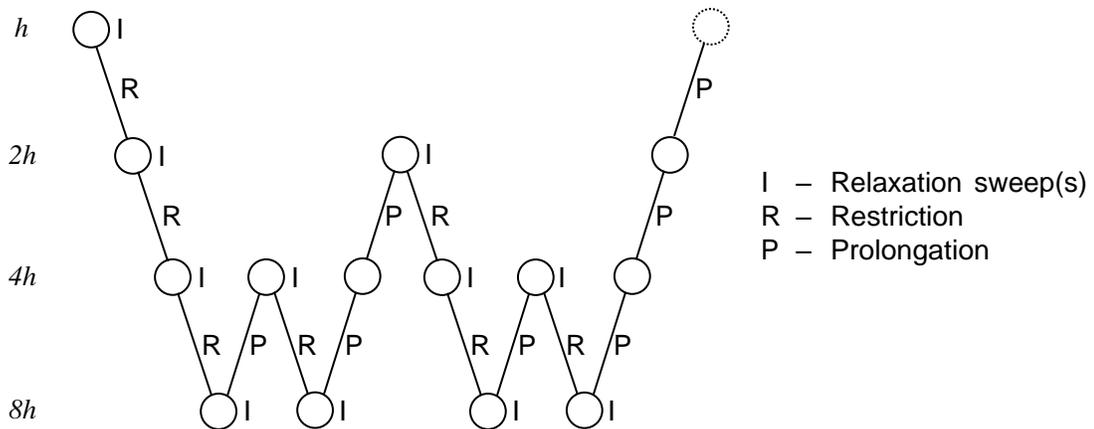


Figure 3: Schematic of W cycle for four grid levels.

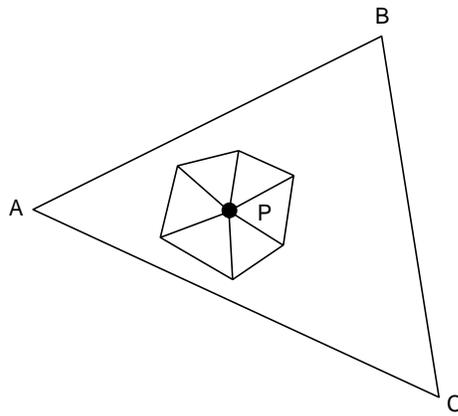


Figure 4: Example of a fine-grid node P that will not contribute information to the coarse grid if linear interpolation is used.

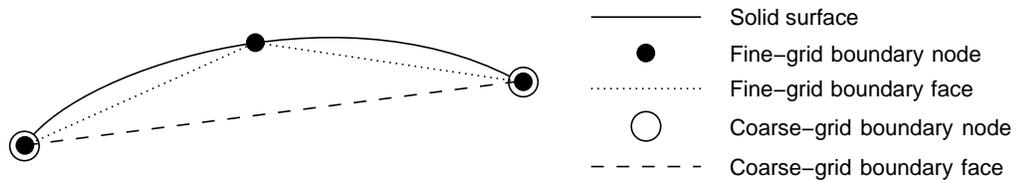


Figure 5: Discretization of a curved boundary surface for both a fine and a coarse grid.

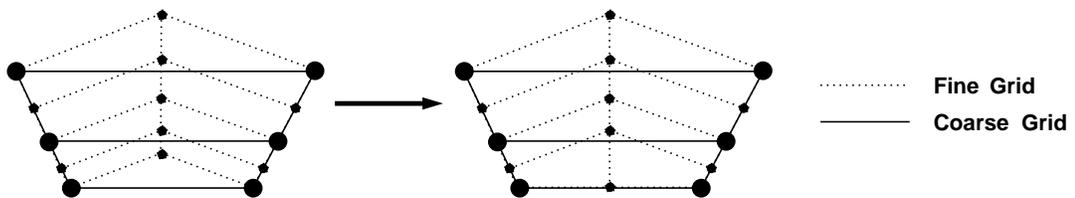


Figure 6: Effective interpolation near viscous surfaces. The diagonal edges cutting across the quadrilateral cells are omitted for visual clarity.

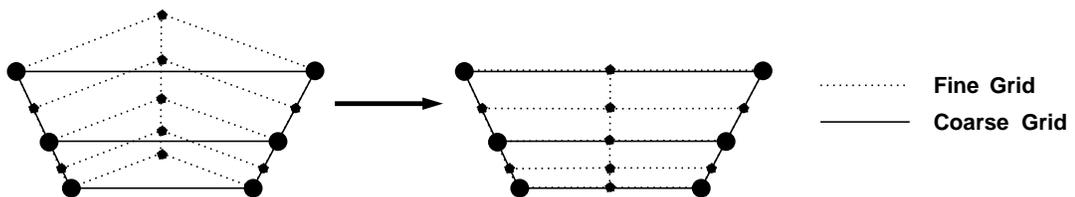


Figure 7: Effective interpolation near viscous surfaces for structured grids.

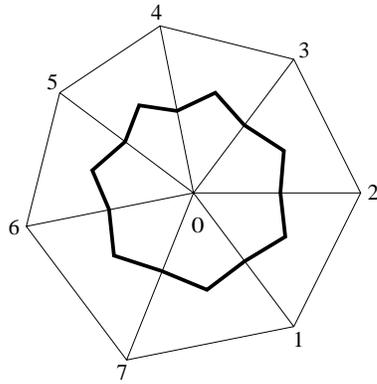


Figure 8: Median-dual control volume for node 0.

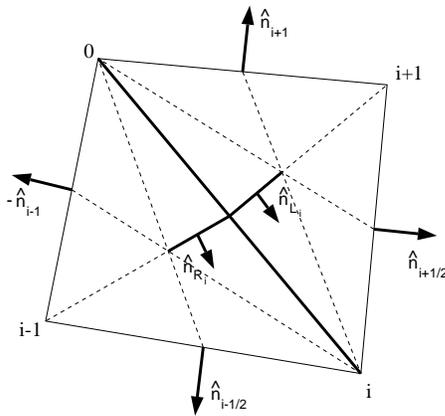


Figure 9: Contribution of an individual edge to the median-dual control volume for node 0.

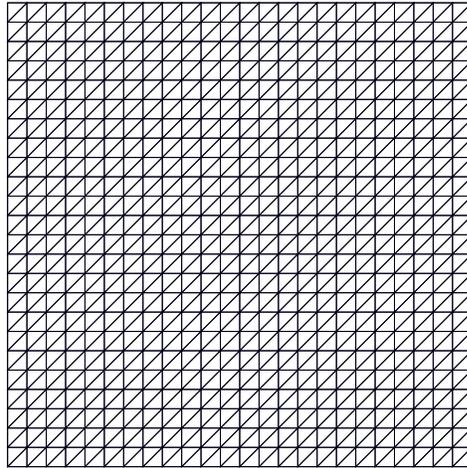


Figure 10: Sample unstructured grid for a square.

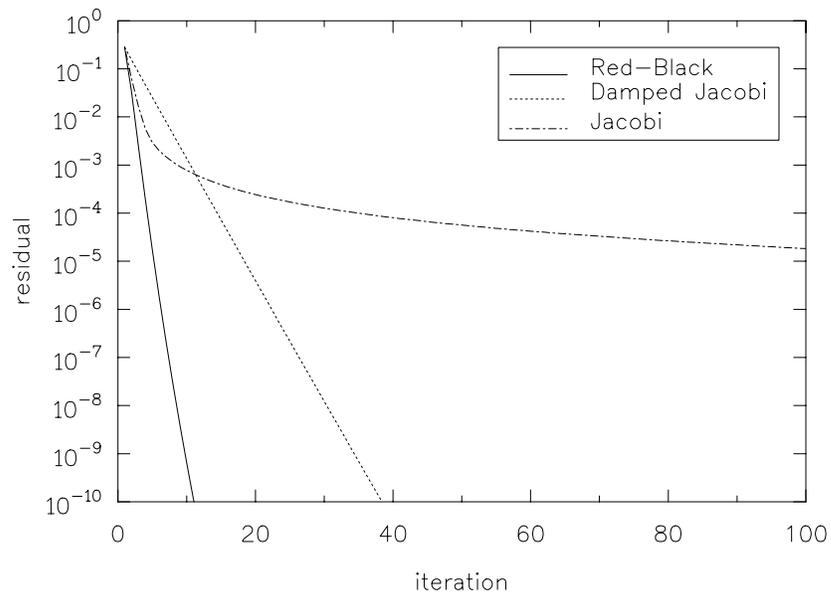


Figure 11: Effect of relaxation scheme on multigrid performance. All runs are made using a 4-level V cycle with a direct solution on the coarsest mesh.

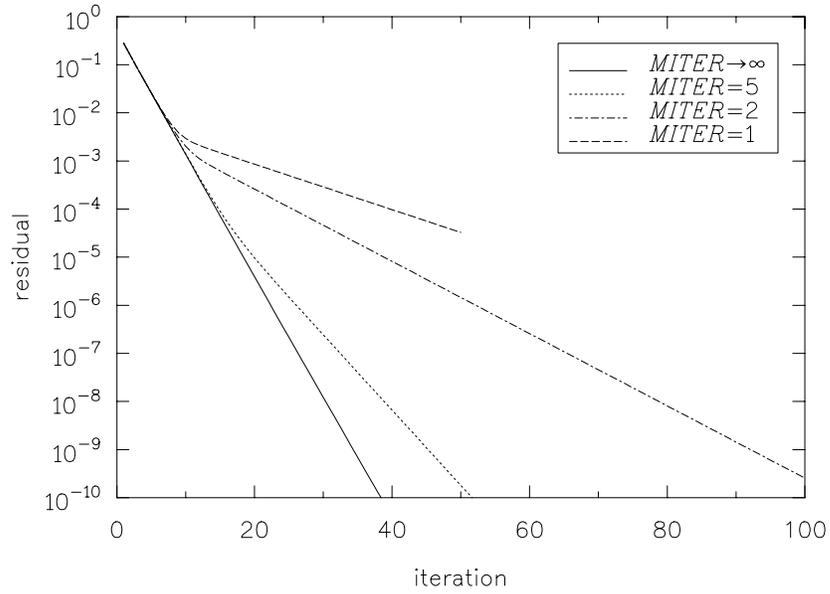


Figure 12: Effect of coarse grid convergence level on convergence of the multigrid cycle. Although not shown, convergence for $MITER = 10$ are virtually identical to those for $MITER \rightarrow \infty$.

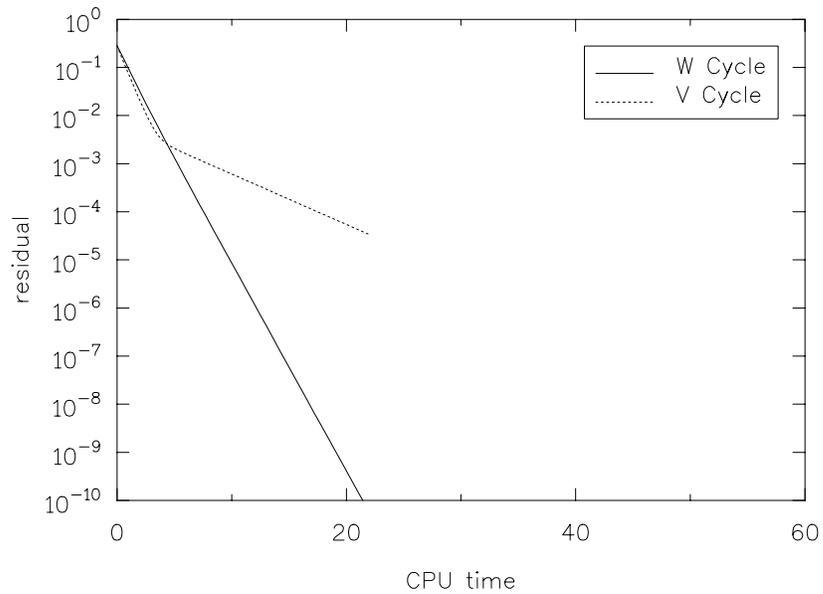


Figure 13: Effect of multigrid cycle on convergence. Both cases use 4 grid levels with 1 damped-Jacobi relaxation sweep on the coarsest grid.

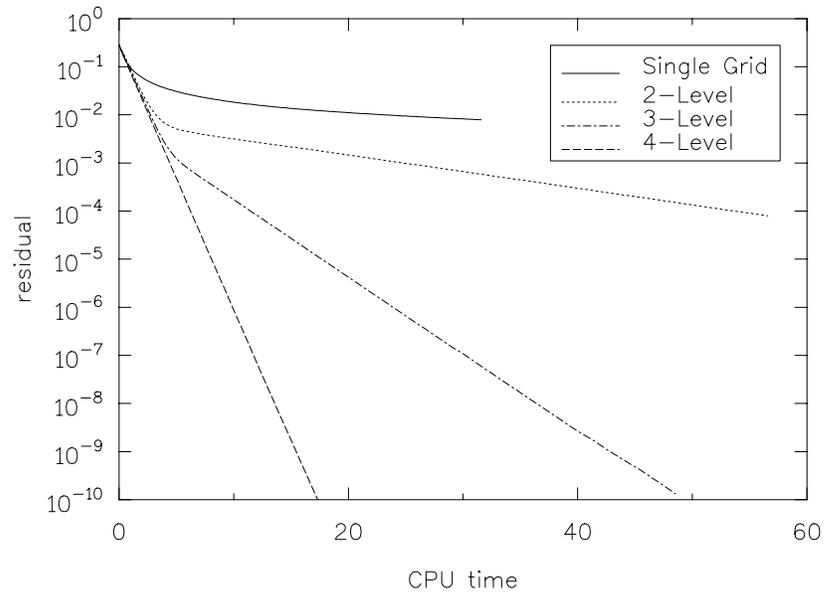


Figure 14: Effect of number of grid levels on convergence of the multigrid scheme. All cases use a V cycle with 10 damped-Jacobi relaxation sweeps on the coarsest grid.

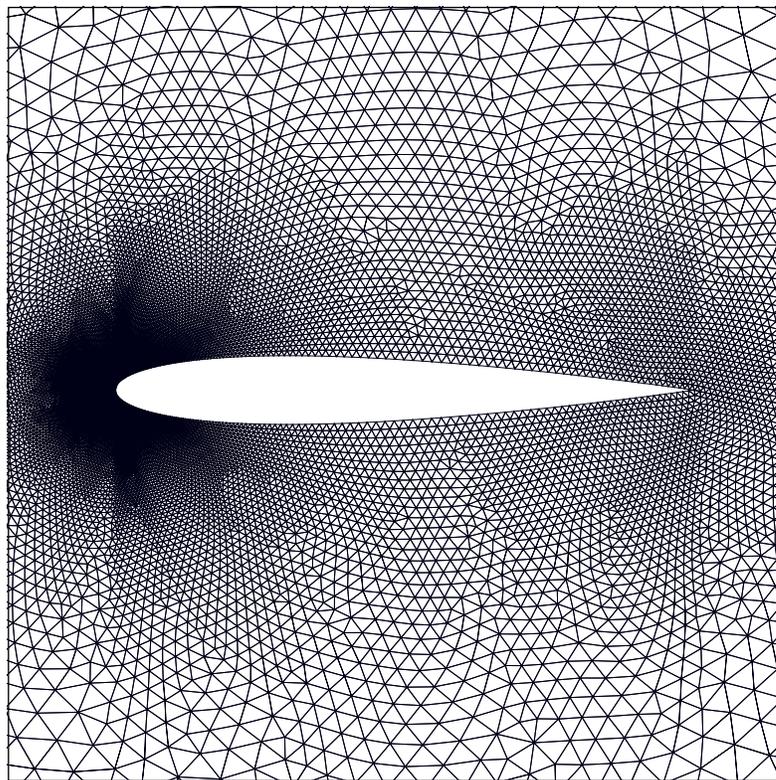


Figure 15: Portion of grid around a NACA 0012 airfoil.

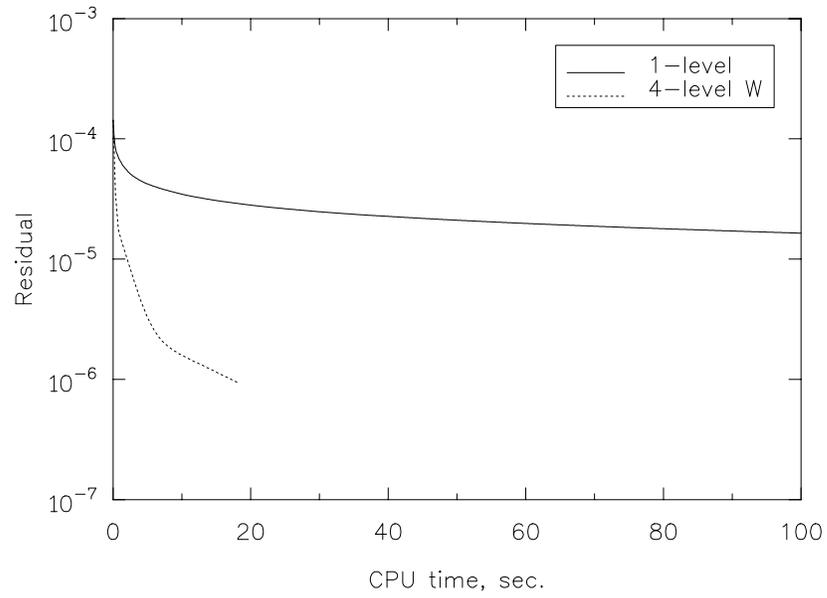


Figure 16: Convergence history for nonlifting potential flow over a NACA 0012 airfoil at zero incidence.

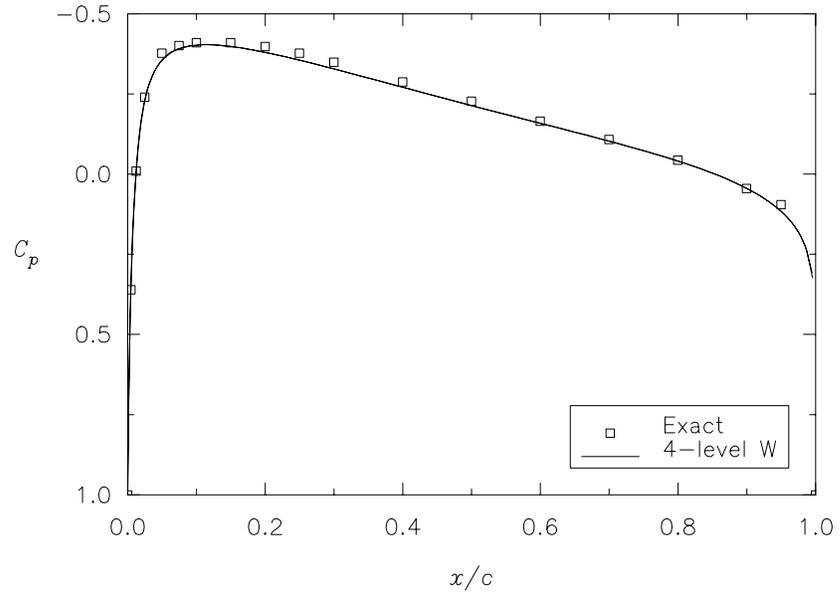


Figure 17: Surface pressure coefficient distribution on a NACA 0012 airfoil at zero incidence.

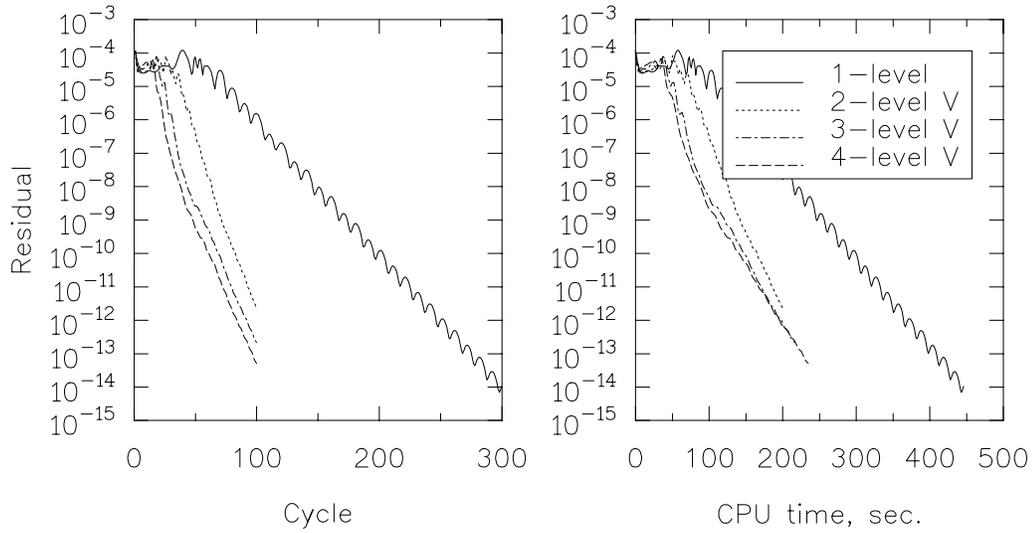


Figure 18: Convergence histories for several V cycles versus both cycle number and computer time for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).

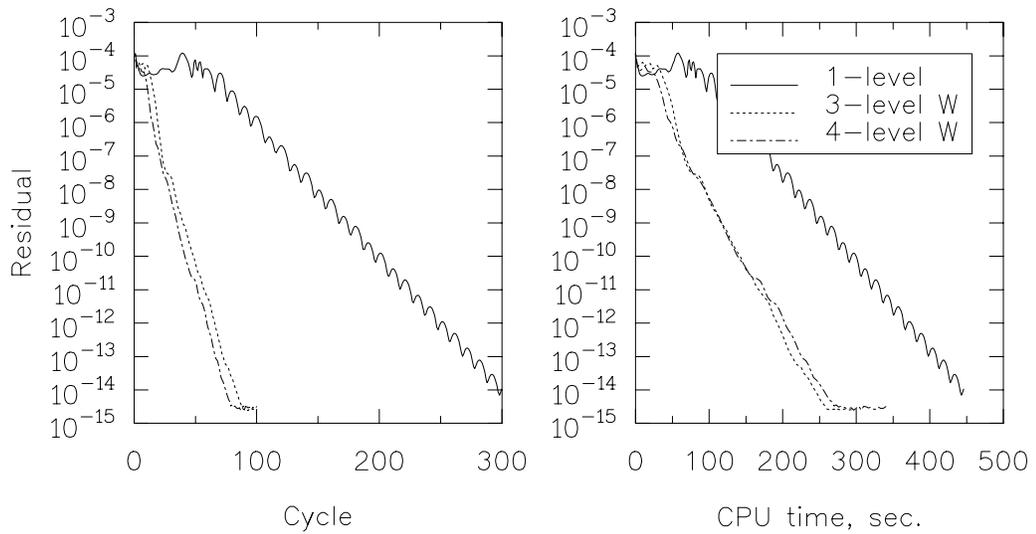


Figure 19: Convergence histories for several V cycles versus both cycle number and computer time for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).

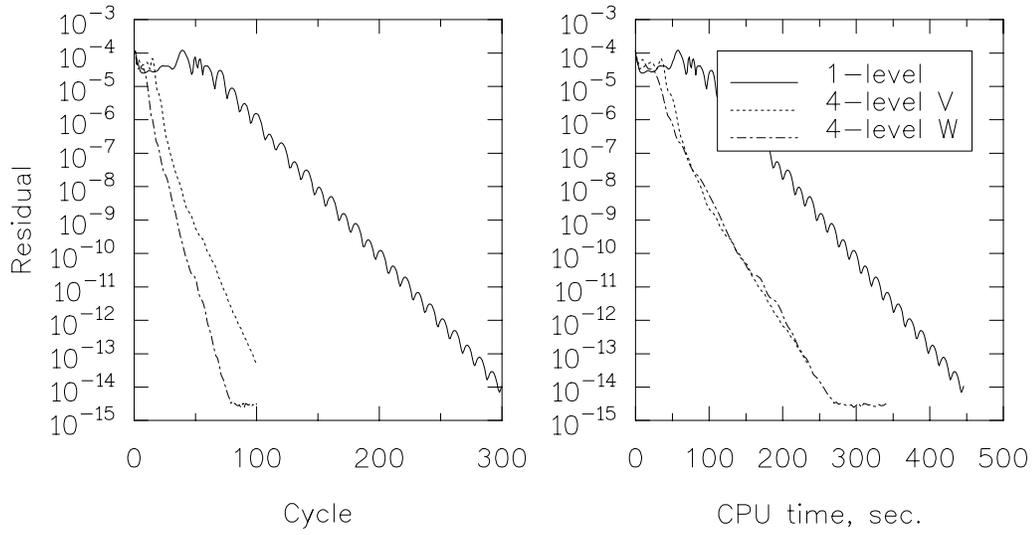


Figure 20: Comparison of performance of V and W cycles versus both cycle number and computer time for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).

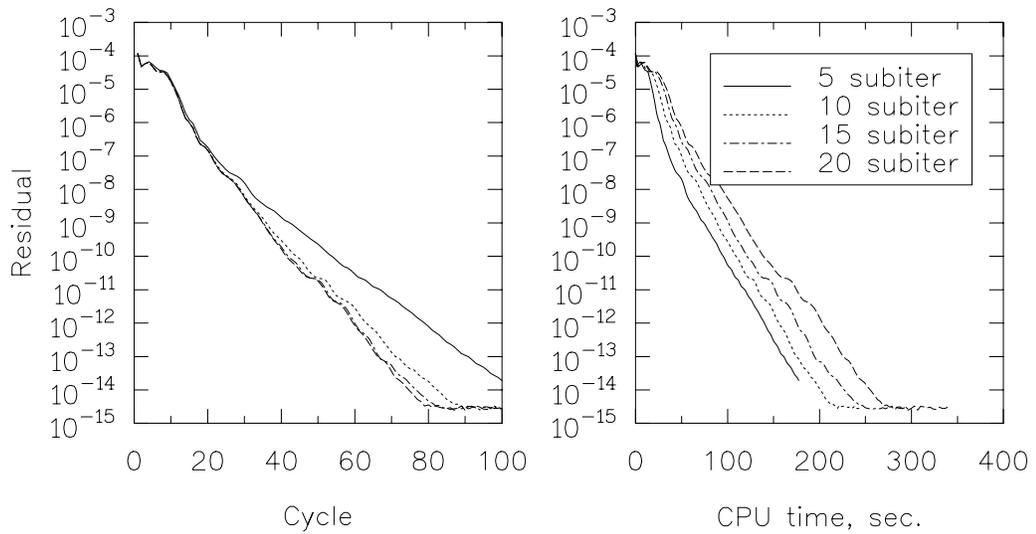


Figure 21: Effect of number of subiterations on performance of a 4-level W cycle for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).

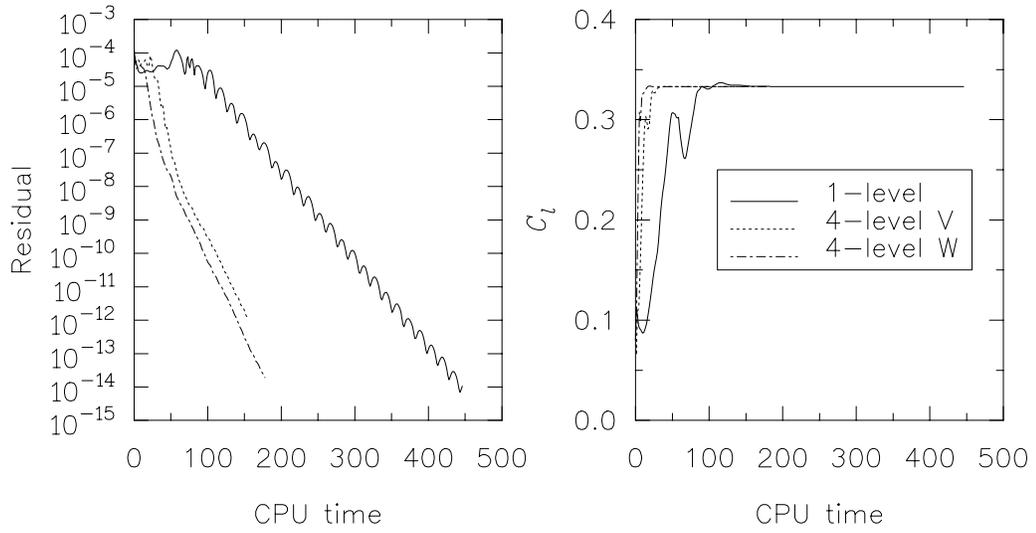


Figure 22: Comparison of performance of V and W cycles for inviscid flow over a NACA 0012 airfoil ($M_\infty = 0.8$, $\alpha = 1.25^\circ$).

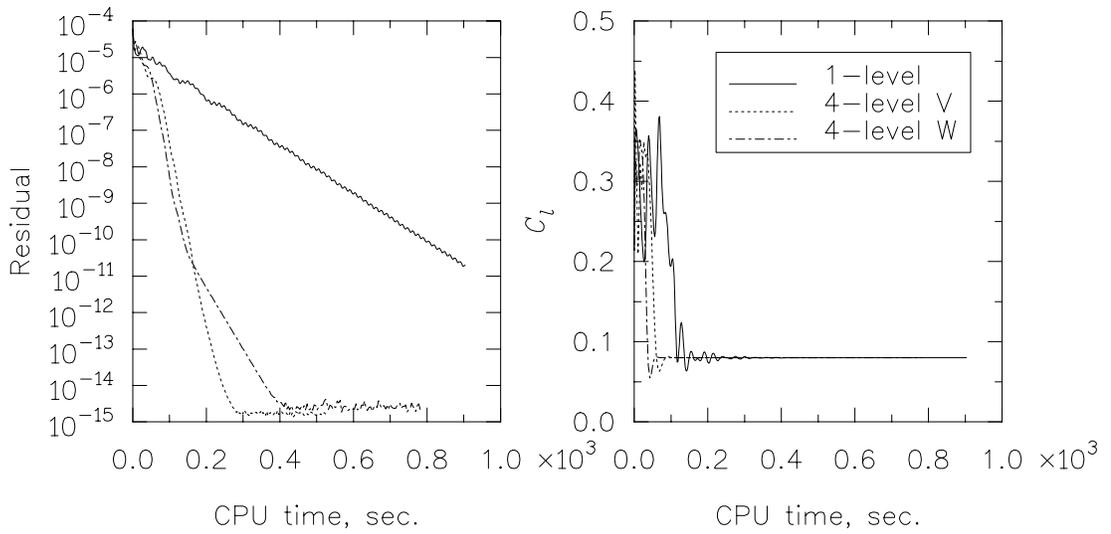
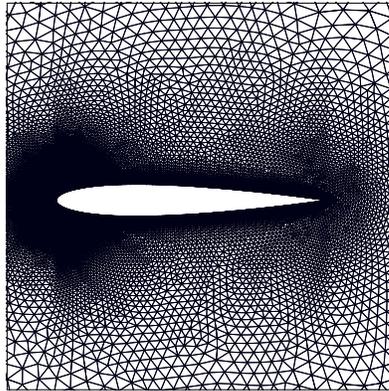
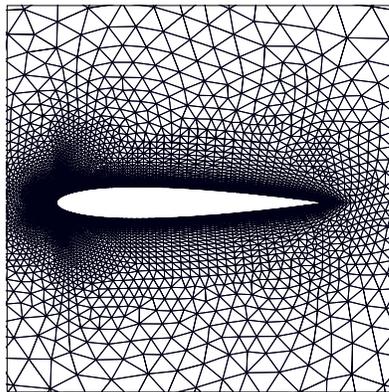


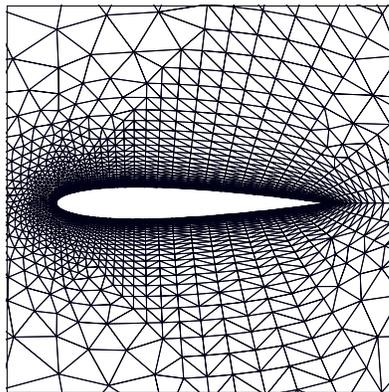
Figure 23: Comparison of performance of V and W cycles for laminar flow over a NACA 0012 airfoil ($M_\infty = 0.5$, $\alpha = 3^\circ$, $Re = 5000$).



a. Grid level 0.

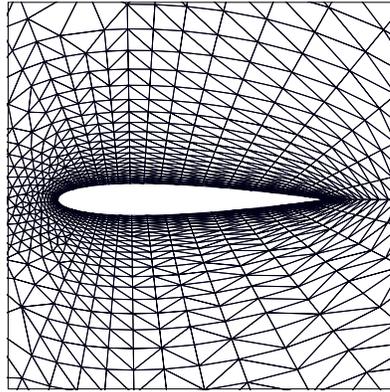


b. Grid level 1.



c. Grid level 2.

Figure 24: Grids around NACA 0012 airfoil used for laminar-flow case. (Continued) . . .



d. Grid level 3.

Figure 24: Grids around NACA 0012 airfoil used for laminar-flow case.

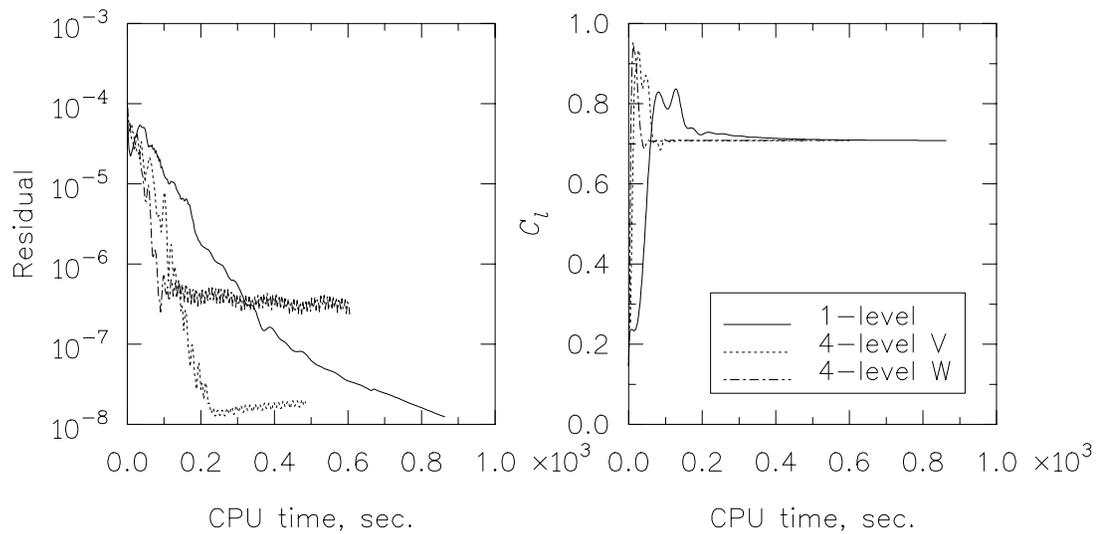


Figure 25: Comparison of performance of V and W cycles for turbulent flow over an RAE 2822 airfoil ($M_\infty = 0.75$, $\alpha = 2.81^\circ$, $Re = 6.2 \times 10^6$).

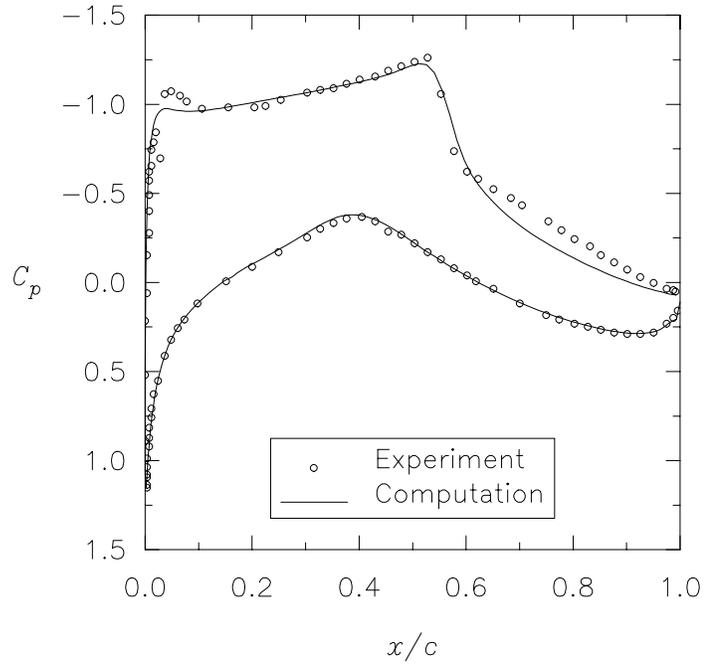


Figure 26: Comparison of surface pressure coefficient distribution with experiment on RAE 2822 airfoil ($M_\infty = 0.75$, $\alpha = 2.81^\circ$, $Re = 6.2 \times 10^6$).

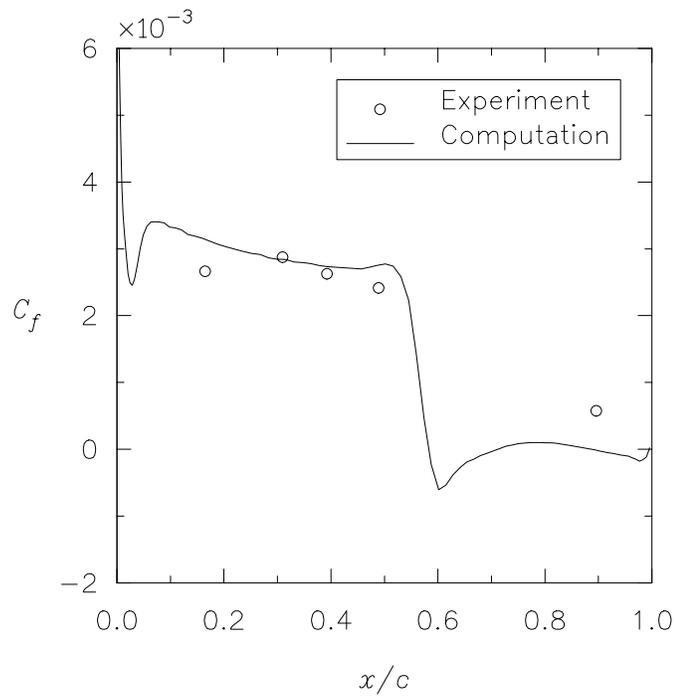


Figure 27: Comparison of skin friction coefficient distribution with experiment on RAE 2822 airfoil ($M_\infty = 0.75$, $\alpha = 2.81^\circ$, $Re = 6.2 \times 10^6$).

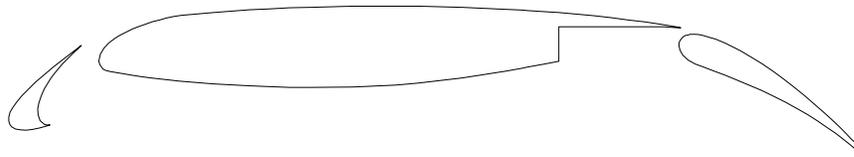


Figure 28: Geometry of 3-element airfoil.

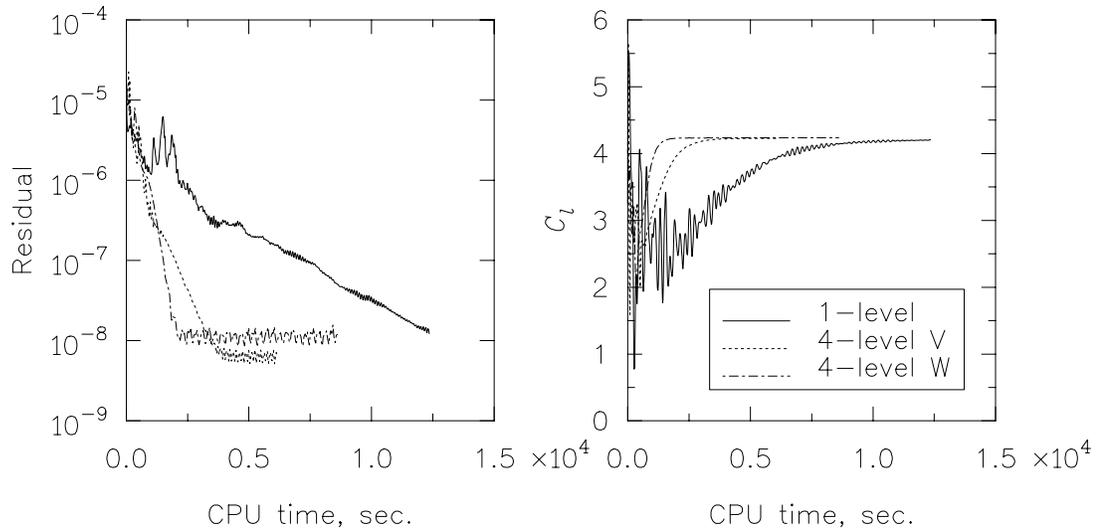


Figure 29: Comparison of performance of V and W cycles for turbulent flow over a 3-element airfoil ($M_\infty = 0.2$, $\alpha = 16.21^\circ$, $Re = 9 \times 10^6$).

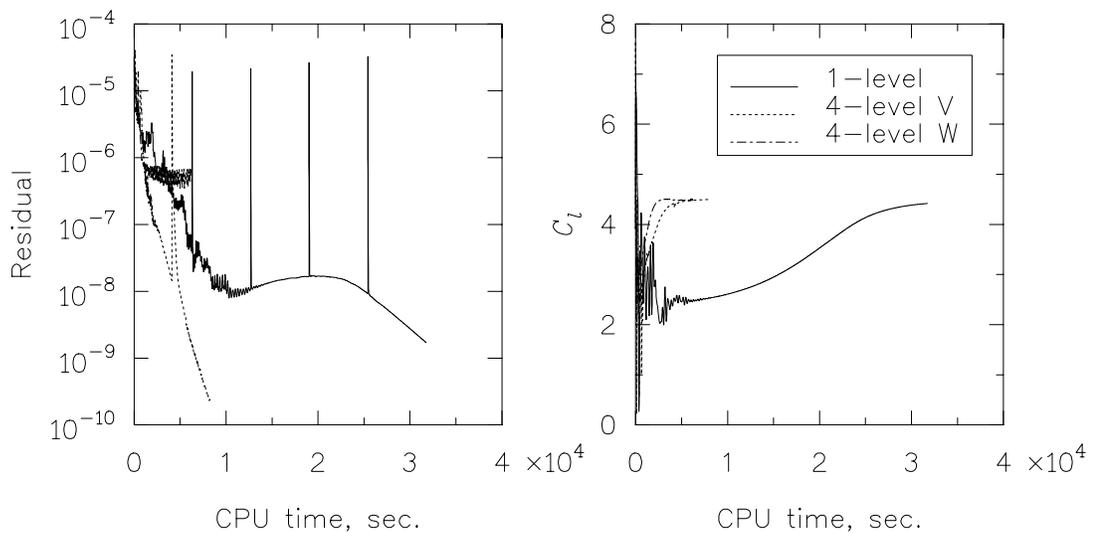


Figure 30: Comparison of performance of V and W cycles for turbulent flow over a 3-element airfoil ($M_\infty = 0.2$, $\alpha = 21.34^\circ$, $Re = 9 \times 10^6$).

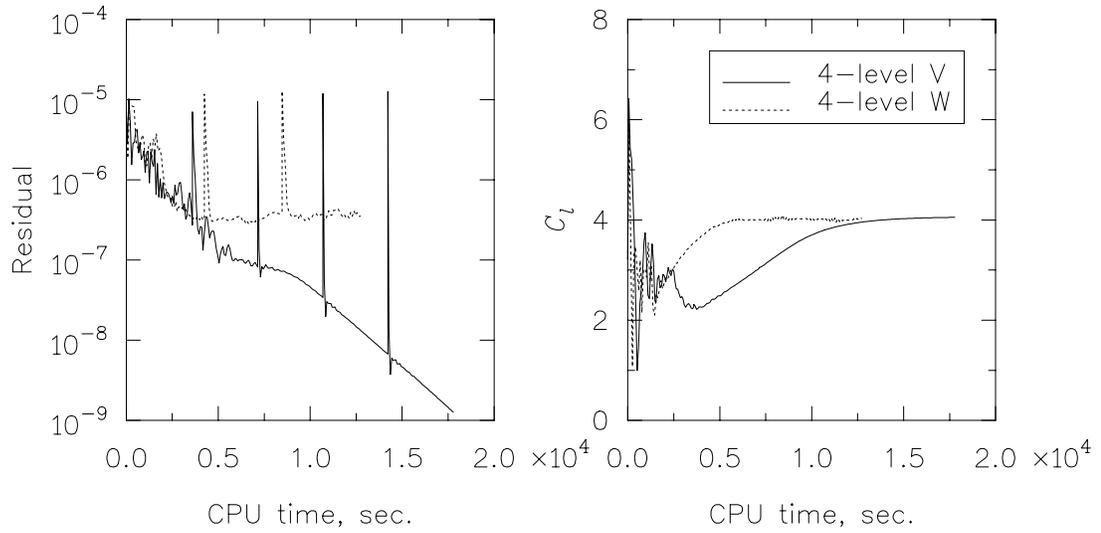


Figure 31: Comparison of performance of V and W cycles for turbulent flow over a 3-element airfoil ($M_\infty = 0.2$, $\alpha = 16.21^\circ$, $Re = 9 \times 10^6$) on a grid of 309,000 nodes.

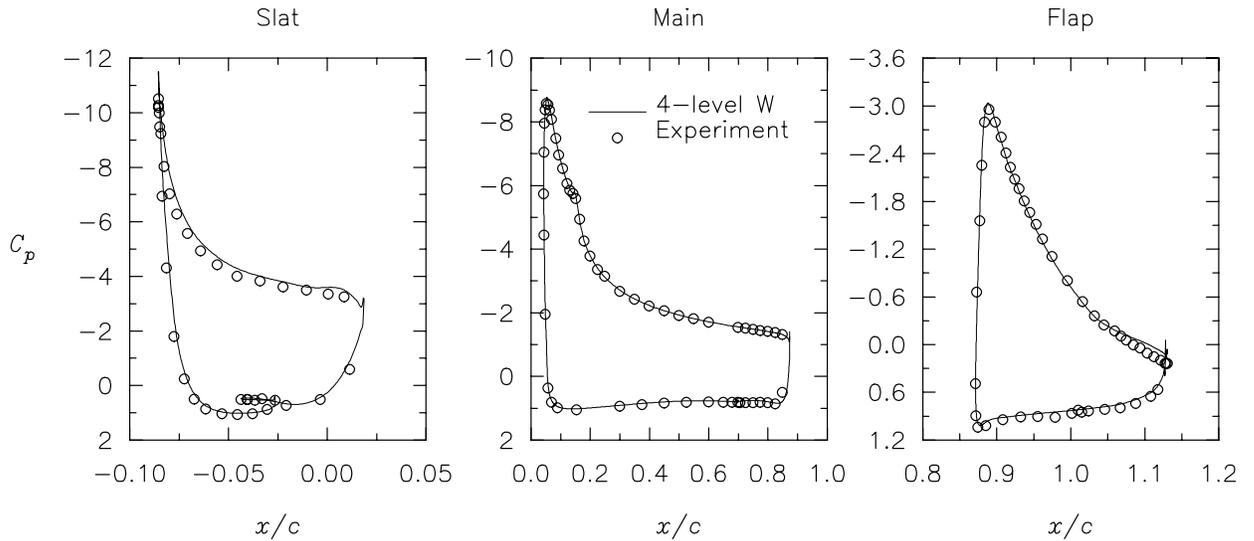


Figure 32: Comparison of distributions of surface pressure coefficients for V cycle with experiment for turbulent flow over a 3-element airfoil ($M_\infty = 0.2$, $\alpha = 16.21^\circ$, $Re = 9 \times 10^6$) on a grid of 309,000 nodes.