NASA-CR-188734

IN-34-CR
30102
P33

DEVELOPMENT OF ITERATIVE TECHNIQUES FOR THE SOLUTION OF UNSTEADY COMPRESSIBLE VISCOUS FLOWS

Grant. NAG-1-1217

Progress Report for the Period

February 14, 1991 - August 13, 1991

Submitted to

NASA Langley Research Center
Hampton, VA 23665

Attn: Dr. John B. Malone

Prepared by

Lakshmi N. Sankar
Professor, School of Aerospace Engineering

Duane Hixon
Graduate Research Assistant

School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332

August 1991

INTRODUCTION

This research project deals with the development of efficient iterative solution methods for the numerical solution of two- and three-dimensional compressible Navier-Stokes equations. The work during the present research period (February 14 - August 13, 1991) focuses on two-dimensional applications.

Iterative time marching methods have several advantages over classical multi-step explicit time marching schemes, and non-iterative implicit time marching schemes. Iterative schemes have better stability characteristics than non-iterative explicit and implicit schemes. Thus, the extra work required by iterative schemes per time step per node may usually be offset by the use of a larger time step. Iterative schemes can also be designed to perform efficiently on current and future generation scalable, massively parallel machines.

An obvious candidate for iteratively solving the system of coupled non-linear algebraic equations arising in CFD applications is the Newton method. Many investigators have implemented Newton's method in existing finite difference and finite volume methods. Depending on the complexity of the problem, the number of Newton iterations needed per step to solve the discretized system of equations can, however, vary dramatically from a few (3 to 5) to several hundred.

In this work, another popular approach based on the classical conjugate gradient method, known as the GMRES (Generalized Minimum Residual) algorithm is investigated . The GMRES algorithm has been used in the past by a number of researchers for solving steady viscous and inviscid flow problems with considerable success. Here, we investigate the suitability of this algorithm for solving the system of non-linear equations that arise in unsteady Navier-Stokes solvers at each time step.

Unlike the Newton's method which attempts to drive the error in the solution at each and every node down to zero, the GMRES algorithm only seeks to minimize the L2 norm of the error. In the GMRES algorithm the changes in the flow properties from one time step to the next are assumed to be the sum of a set of orthogonal vectors. By choosing the number of vectors to a reasonably small value N (between 5 and 20) the work required for advancing the solution from one time step to the next may be kept to (N+1) times that of a non-iterative scheme. Many of the operations required by the GMRES algorithm such as matrix-vector multiplies, matrix additions and subtractions can all be vectorized and parallelized efficiently.

Progress During the Reporting Period

During the reporting period, the following tasks were completed:

a) Addition of GMRES solver to an existing code

The GMRES solver was added to an existing time-accurate 2-D ADI Navier-Stokes code, which optionally utilizes Newton iteration to ensure accuracy at each time step. The GMRES solver was coded such that it can solve both time accurate and steady state flow problems. The numerical and mathematical formulation is given in Appendix A. In order to validate the solver and gain experience, it was applied to a variety of steady cases before being used for unsteady calculations.

b) Steady Calculations

The GMRES code was tested on several subsonic and transonic, viscous and inviscid cases.

The first case was an inviscid subsonic problem. The airfoil was a NACA 0012 section at a 2 degree angle of attack. The freestream Mach number was 0.63. Figure 1 shows the L2 norm of the residual plotted against the CPU time used. For a given level of convergence, GMRES (with N=10) required only 50% of the CPU time that the original ADI solver used. Figure 2 shows the C_l histories of the two solvers. It may be seen that the GMRES solver does not oscillate nearly as much about the final result as does the ADI solver.

The second case was more challenging. In this calculation, a NACA 0012 airfoil is in an inviscid, transonic (M = 0.8) flow at a 1.25 degree angle of attack.

This problem was chosen to evaluate the ability of the GMRES solver to capture strong shock waves. Figures 3 and 4 give the residual and lift coefficient history comparisons between the original ADI solver and the GMRES (N=40) code. Again, the GMRES (N=40) solver requires only 50-55% of the CPU time necessary for the ADI code. Also, the lift coefficient converges much more rapidly.

The interesting part of this problem was in choosing the number of GMRES directions to use. Figure 5 shows a comparison of the global residuals for runs where the number of conjugate directions N was varied. Notice how the N=10 and N=20 runs converge very quickly initially, but completely stall after a certain level of residual is attained. Only the N=40 run gave a reasonably low residual before stalling at a global residual of 10^{-8}. A run with N=80 proved that there was a limit to the speedup and accuracy obtainable before the cost of the GMRES routine outweighed the benefits. Figure 6 shows the C_l histories of these runs. Note the inaccurate results from the N=10 and N=20 runs. This shows that the GMRES scheme with very few directions can actually perform worse than a non-iterative ADI method. However, the 40 direction run locks on to the final C_l result very quickly. Figure 7 is the correlation between the lift coefficient and the global residual for the 40 direction run. From this graph, it appears that a residual of 10^{-7} or less is necessary to get accurate lift valuesfrom GMRES for an inviscid case.

The last steady case was a NACA 0012 airfoil at a 5 degree angle of attack. This was a viscous run, with a Reynolds number of 3,450,000. This case compared N=10 with N=40. Figure 8 shows the residual histories of the two runs. This plot shows that, up to the point where it stalls out N=10 run takes 67% of the CPU time compared to a N=40 run. Figure 9 shows the C_l histories. Both solvers reach the same lift coefficient, with the two solvers taking about the same CPU time to reach a steady lift. Comparison of Figures 8 and 9 indicates that a residual of 10^{-8} is necessary for the lift coefficient to stabilize. This is the only steady viscous run performed, so it may not be a good rule of thumb. The C_p distribution on the airfoil is compared to the original ADI result in Figure 10. Excellent agreement was obtained.

c) Unsteady Calculations

Two cases were studied using the time-accurate GMRES method: a plunging NACA 64-A010 airfoil in inviscid transonic flow, and a pitching NACA 0012 airfoil in subsonic flow.

The first case to be studied was a sinusoidally plunging NACA 64-A010 airfoil in transonic flow, previously studied by Yoshihara and Magnus, and by Steger. The freestream Mach number was 0.8, and the reduced frequency was 0.2. This case was run in the Euler (inviscid) mode.

At first, a time step 20 times that of the original ADI scheme was used. The lift coefficients correlated well for both 10 and 20 directions compared to the ADI solver. Problems became apparent when the moment coefficients were plotted. The GMRES (20/20) [# of directions/time step multiplier] run gave the correct magnitude of the C_m, but the phase was shifted by 30 degrees. The 10/20 run gave even worse results: the magnitude was extremely bad, and the phase shift was also large. Figures 11 and 12 show the lift and moment coefficient histories for these runs.

At this point, reducing the time step was tried. Since a time factor of 20 meant that one GMRES step corresponds to 3.5 degrees of phase angle, it was thought that a smaller time step would help resolve the shock motion. A GMRES (10/10) run gave much better results for the phase of the moment, but overpredicted the magnitude. When a GMRES (5/5) run was tried, it was found that 5 directions were not enough to ensure stability, and the solver blew up.

The next run was a GMRES (5:5/10) (two 5 direction iterations per step with 10 times the ADI time step). This run was performed to see if the nonlinearities of the transonic flow could be causing some of the difficulties (in other words, trying to let the GMRES have a chance to correct itself). This is apparently not the case, as the results for the (10/10) and (5:5/10) run are almost identical. Figures 13 and 14 compare these results with the ADI and the GMRES (20/20) results.

To test finally whether the time step was too large, a GMRES (10/5) run was performed. Note that this run takes twice as long as the original ADI code. The results were greatly improved over the previous calculations. Figures 15 and 16 give the lift and moment coefficients results. From these runs, it is seen that a time step of 5 times the ADI step is small enough to capture the physics of the flow, while a time step 10 times as large is not.

From the above studies, it appears that a time step which is very large can give very poor results, particularly for unsteady transonic applications, where the pitching moments are governed by shock speeds and shock locations. A very large time step, which requires the shock to traverse several mesh widths can give incorrect shock speeds and shock locations, even when a temporally and spatially conservative scheme is used.

The above difficulty in using large time steps relative to an ADI method may, however, be peculiar only to inviscid transonic calculations. In viscous flows, the time steps for the ADI scheme are small. Even when a time step 20 to 40 times that of an ADI scheme is used, the shock is not likely traverse more than one or two streamwise cells per time step. Thus, the GMRES method may give good results in unsteady transonic, viscous flows, and permit use of very large time steps relative to an ADI scheme. This hypothesis remains to be tested using an unsteady transonic viscous flow case.

The dynamic stall of a NACA 0012 airfoil was the last case studied to evaluate the time-accurate GMRES method. The airfoil was pitched about the quarter chord point from 5 degrees to 25 degrees, at a reduced frequency of 0.151. Freestream Mach number is 0.283, and Reynolds number is 3,450,000. Many runs were performed on this case to evaluate the effects of changing the time step and the number of directions.

A time step 20 times that of the original ADI scheme was tried initially. To get a comparison, 20 directions were run (20/20). Note that this takes slightly longer than the original ADI code to run, mainly due to a 20x20 matrix inversion required by the algorithm. Figures 17,18 and 19 compare the GMRES results with the experiment. Figure 20 shows the L2 norm residual variation with time for the GMRES (20/20) run. The 20/20 run is seen to give good qualitative agreement with the experiments. For this reason, the GMRES (20/20) run was chosen as the baseline for the later runs.

The next series of runs were performed to see what sort of speedups were likely from GMRES. For this set, a time step of 20 times the ADI time step was used (i.e., GMRES (x/20)). The number of directions were set at 10 and 5. Results for lift, moment, and residual are shown in Figures 21, 22, and 23. These are plotted against time as it is easier to judge results in this way. The output shows that GMRES (10/20) is very nearly as good as (20/20), while accuracy falls off in the (5/20) run.

The last series of runs were done to see the effect of the time step on the GMRES solver. From the results of the last series, GMRES (x/2x) was chosen (number of directions equal to half of the time step factor). These results are shown in Figures 24, 25, 26, 27, 28, and 29. The results were split into two groups to keep the graphs legible. From these graphs, it can be seen that there is a tradeoff between accuracy of the GMRES iteration (which goes up with number of directions) and the time step necessary to resolve flow phenomena. From this series of runs, it appears that a time factor of 20 is the best choice in this case.

Multigrid Unsteady Runs:

The GMRES algorithm requires storage of the conjugate correction vectors at every time step. For a N direction scheme, 4 N additional words must be stored per node. The amount of storage can be reduced if some of the correction vectors are computed and stored on a coarse grid, and only the rest of the vectors are stored on a fine grid. This requires a multi-grid method, where the original non-linear system of equations on a fine grid are transferred to a coarse grid in the "Full Approximation Scheme (FAS)" sense.

Two algorithms were tried: a Fine-Coarse pattern, and a Fine-Coarse-Fine pattern. In Figures 30, 31, and 32, a (20/20) run is compared to: a normal, fine-grid-only (10/20) run, a F-C (10/20) run (10 directions on both fine and coarse grids), and a F-C-F (5:5/20) run (5 fine, then 5 coarse, then 5 more fine). No gain due to multigrid is apparent; in fact, the multigrid solver made the GMRES solver less stable, and both multigrid runs blew up halfway around the cycle.

CONCLUDING REMARKS

The GMRES algorithm has been implemented in an existing unsteady 2-D compressible Navier-Stokes solver. Encouraging preliminary results for steady and unsteady, viscous and inviscid calculations have been obtained. Our attempts to reduce the memory requirements of the GMRES scheme through multigrid techniques have not been successful to date.

The above results, and additional dynamic stall calculations on a fine grid, will be presented at the forthcoming AIAA Aerospace Sciences Conference in Reno, Nevada, in January 1992.

Appendix A

Mathematical and Numerical Formulation

Underlying Newton Based Formulation

For the sake of simplicity, the Newton iteration time marching scheme is discussed here for the 2-D compressible Navier-Stokes equations on a Cartesian coordinate system. The scheme is , however, applicable to 3-D flows on curvilinear body-fitted coordinate systems.

The governing equations may be written formally as:

$$\vec{q}_t + \vec{F}_x + \vec{G}_y = \vec{R}_x + \vec{S}_y \qquad (1)$$

Here $\vec{q}$ is the vector containing the flow properties such as density, u- and v- momentum per unit volume, and total energy per unit volume. The terms $\vec{F}$ and $\vec{G}$ represent the transport of mass, momentum, and energy by convection, and also include pressure effects. The terms $\vec{R}$ and $\vec{S}$ represent viscous stress effects, heat conduction, and the friction-generated heat.

The objective of the calculation is to determine $\vec{q}$ at a time level 'n+1' given the values of $\vec{q}$ at a previous time level 'n'. On a stretched Cartesian grid, at a typical node (i,j), this equation may be discretized as:

$$\frac{\left(q_{i,j}^{n+1} - q_{i,j}^{n}\right)}{\Delta t} + \delta_x F^{n+m} + \delta_y G^{n+m} = \delta_x R^{n+m} + \delta_y S^{n+m} \qquad (2)$$

The above discretization is first order accurate in time if 'm' is set to zero or one, and second order accurate if 'm' is set to 1/2. The operators δ_x and δ_y represent second order accurate or fourth order accurate spatial differences. The terms F and G are numerical fluxes that differ from the physical fluxes F and G in that they incorporate artificial viscosity terms, or changes to F and G needed to make the scheme upwinded. In the present studies, which primarily deal with subsonic and transonic applications, the numerical viscosity model proposed by Jameson, Turkel, and Schmidt and modified by Swanson and Turkel is used [Ref. 15].

In the past, equation set (2) was solved by non-iterative time marching schemes (e.g., Ref. 10).

A variant of the non-iterative time marching schemes is an iterative time marching scheme. Several researchers have used Newton-iteration schemes in steady and unsteady Navier-Stokes calculations [e.g., Ref. 16]. In this approach, a sequence of sub-iterations (k = 0,1,2,...) are used within each time step. Equation (2) is rewritten as follows:

$$\frac{\left(q_{i,j}^{n+1,k} - q_{i,j}^{n}\right)}{\Delta t} + \delta_x F^{n+m,k} + \delta_y G^{n+m,k} = \delta_x R^{n+m,k} + \delta_y S^{n+m,k} \qquad (3)$$

The terms F, G, R, and S at time-iteration level (n+m,k) are expanded about their values at the time level 'n+m' and at the previous iteration level 'k-1'. This leads to a system of coupled, linear equations for the changes in q between two successive iterations:

$$[M]\{\Delta q\} = \{R\} \qquad (4)$$

where

$$\Delta q = q^{n+1,k} - q^{n+1,k-1} \tag{5}$$

and {R} is the residual:

$$\frac{\left(q_{i,j}^{n+1,k-1} - q_{i,j}^{n}\right)}{\Delta t} + \delta_x F^{n+m,k-1} + \delta_y G^{n+m,k-1} = \delta_x R^{n+m,k-1} + \delta_y S^{n+m,k-1} \tag{6}$$

The objective of the Newton iteration scheme is to solve equation set (3) by repeated application of equation set (4). The matrix [M] is a banded 5- or 9-diagonal matrix whose individual elements are 4x4 matrices. This matrix is usually approximately factored into tri-diagonal matrices and inverted. Equation set (4) is solved until the residual R is driven to zero. In a full Newton iteration scheme, the elements of the coefficient matrix will be recomputed every iteration, based on $q^{n+1,k-1}$. When R approaches zero, equation (2) is exactly satisfied.

The advantage of a Newton iteration scheme, particularly in the context of approximate factorization schemes, is that the errors associated with the factorization method can be reduced or removed. That is, as Δq goes to zero, the errors associated with the approximate factorization of [M] do not affect the solution. By specifying a convergence criteria for Δq, one can also ensure that equation set (2) is satisfied at every time step to within a user-specified tolerance. The disadvantage of the above type of Newton iteration schemes is that each Newton iteration requires approximately the same amount of CPU time as a single step using a non-iterative time marching scheme. To be cost-effective, a Newton-iteration based scheme that uses, say, 5 iterations per time step should use a CFL number that is, on the average, 5 times larger than the CFL number associated with a non-iterative scheme.

GMRES Formulation

The objective of the GMRES method is to accelerate the convergence rate of the existing Newton iterative solver.

In each Newton iteration, the Newton solver takes an approximation to the correct solution and uses it to obtain an improved approximate solution:

$$\vec{q}^{\,n+1,k} = A\left(\vec{q}^{\,n+1,k-1}\right) \qquad (7)$$

where $\vec{q}^{\,n+1,k}$ is the vector containing the all of the flow properties at the 'n+1' time level and the new ('k') iteration level. This vector is, in 2-D, (4 x imax x jmax) long.

The solution is converged when

$$\vec{q}^{\,n+1,k} = \vec{q}^{\,n+1,k-1} \qquad (8)$$

or

$$\vec{q}^{\,n+1,k-1} - A\left(\vec{q}^{\,n+1,k-1}\right) = 0 \qquad (9)$$

GMRES solves the system of linear equations:

$$F(\vec{q}) = \vec{q} - A(\vec{q}) = 0 \qquad (10)$$

by minimizing the L2 norm of the residual **F**. The original Newton iterative scheme is used to evaluate **F** given a value of $\vec{\mathbf{q}}$.

In order to accomplish this, GMRES computes J orthonormal search directions and finds the gradient of the residual in each direction. With this, a least squares problem is solved to minimize the residual in the Krylov subspace defined by the J orthonormal direction vectors.

The GMRES algorithm works as follows:

First, the initial direction is computed by the Newton solver from the initial guess for **q** at the 'n+1' time level:

$$\vec{\mathbf{d}}_1 = \mathbf{F}\left(\vec{\mathbf{q}}^{n+1,0}\right) \qquad (11)$$

and normalized as

$$\vec{\mathbf{d}}_1 = \frac{\vec{\mathbf{d}}_1}{\left\|\vec{\mathbf{d}}_1\right\|}. \qquad (12)$$

To compute the remaining search directions (j = 1,2,...,J-1), take:

$$\vec{\mathbf{d}}_{j+1} = \vec{\mathbf{F}}\left(\vec{\mathbf{q}}^{n+1,0}; \vec{\mathbf{d}}_j\right) - \sum_{i=1}^{j} \mathbf{b}_{ij}\vec{\mathbf{d}}_i \quad , \qquad (13)$$

where

$$\mathbf{b}_{ij} = \left(\overline{\mathbf{F}}\left(\vec{\mathbf{q}}^{\,n+1,0};\vec{\mathbf{d}}_j\right), \vec{\mathbf{d}}_i\right) \qquad (14)$$

and

$$\overline{\mathbf{F}}\left(\vec{\mathbf{q}};\vec{\mathbf{d}}\right) = \frac{\mathbf{F}\left(\vec{\mathbf{q}}+\varepsilon\vec{\mathbf{d}}\right) - \mathbf{F}\left(\vec{\mathbf{q}}\right)}{\varepsilon} . \qquad (15)$$

Here, e is taken to be some small number.

The new direction $\vec{d}_{j+1}$ is normalized before the next direction is computed:

$$\vec{\mathbf{d}}_{j+1} = \frac{\vec{\mathbf{d}}_{j+1}}{\left\|\vec{\mathbf{d}}_{j+1}\right\|}. \qquad (16)$$

After the search directions are known, the solution vector is updated using

$$\vec{\mathbf{q}}^{\,n+1,\mathrm{new}} = \vec{\mathbf{q}}^{\,n+1,0} + \sum_{j=1}^{J} a_j\vec{\mathbf{d}}_j , \qquad (17)$$

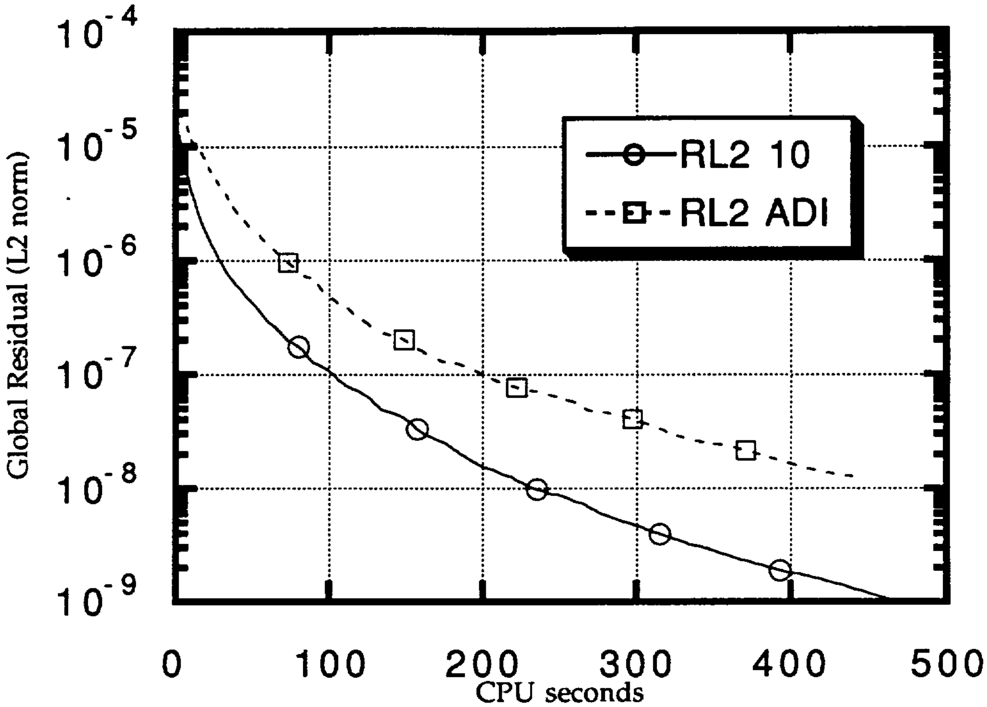
where the coefficients a_j are chosen to minimize:

$$\left\| \mathbf{F}\left(\vec{\mathbf{q}}^{\,n+1,\mathrm{new}}\right) \right\|^2 \cong \left\| \mathbf{F}\left(\vec{\mathbf{q}}^{\,n+1,0}\right) + \sum_{j=1}^{J} \mathbf{a}_j \overline{\mathbf{F}}\left(\vec{\mathbf{q}}^{\,n+1,0}; \vec{\mathbf{d}}_j\right) \right\|^2 . \tag{18}$$

One of the most important features of the GMRES method is its portability between existing flow solvers. In this formulation, the original Newton solver is used only to evaluate the residual **F**, and does not directly affect the correction applied to the flow variables. Therefore, this procedure is applicable to any iterative flow solver that can compute **F** for any given **q** and send it to the GMRES routine. This is a significant advantage over other methods such as multigrid analyses which are closely tied to the flow solver.

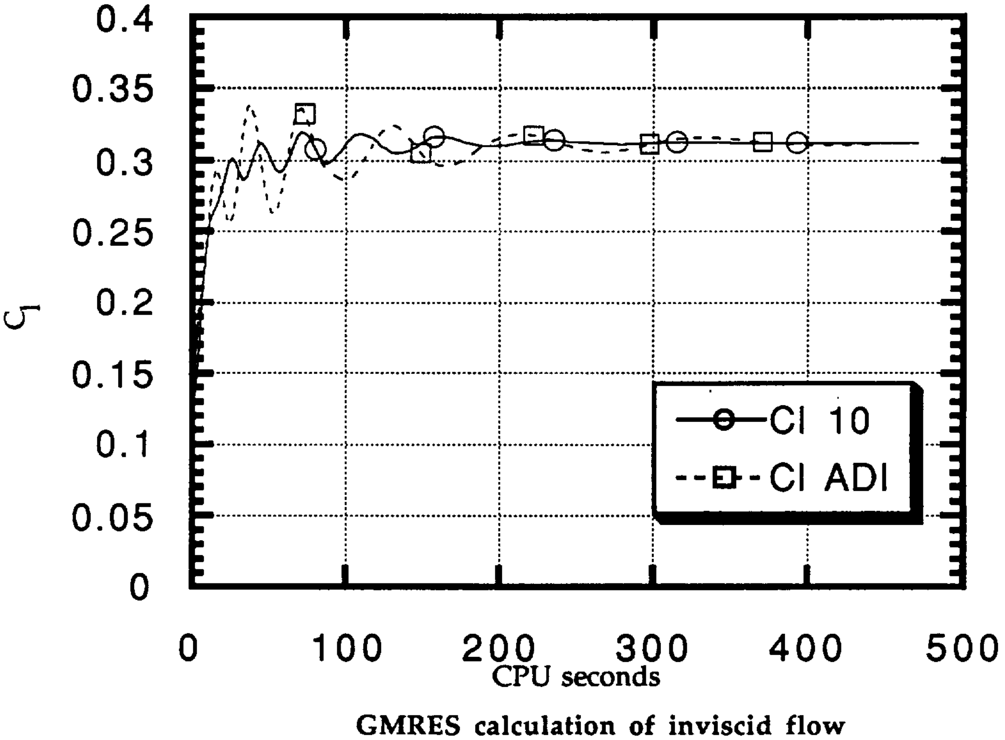
GMRES has similar advantages and disadvantages as the Newton scheme over the original ADI code. For one step using 'J' directions, GMRES calls the Newton solver J+1 times, and also must invert a full JxJ matrix. Thus, to gain an improvement in CPU time, the time step must be at least a factor of J+1 times larger than the original ADI time step.

The major disadvantage that GMRES has compared to the Newton scheme is the required memory. Each direction is equivalent to storing the entire flow field, and GMRES requires that all J directions be stored as well as the last **F** derivative.



GMRES calculation of inviscid flow
about a NACA 0012 airfoil
(M = 0.63; α = 2.00 degrees)

Figure 1



GMRES calculation of inviscid flow
about a NACA 0012 airfoil
(M=0.63; α = 2.00 deg.)

Figure 2

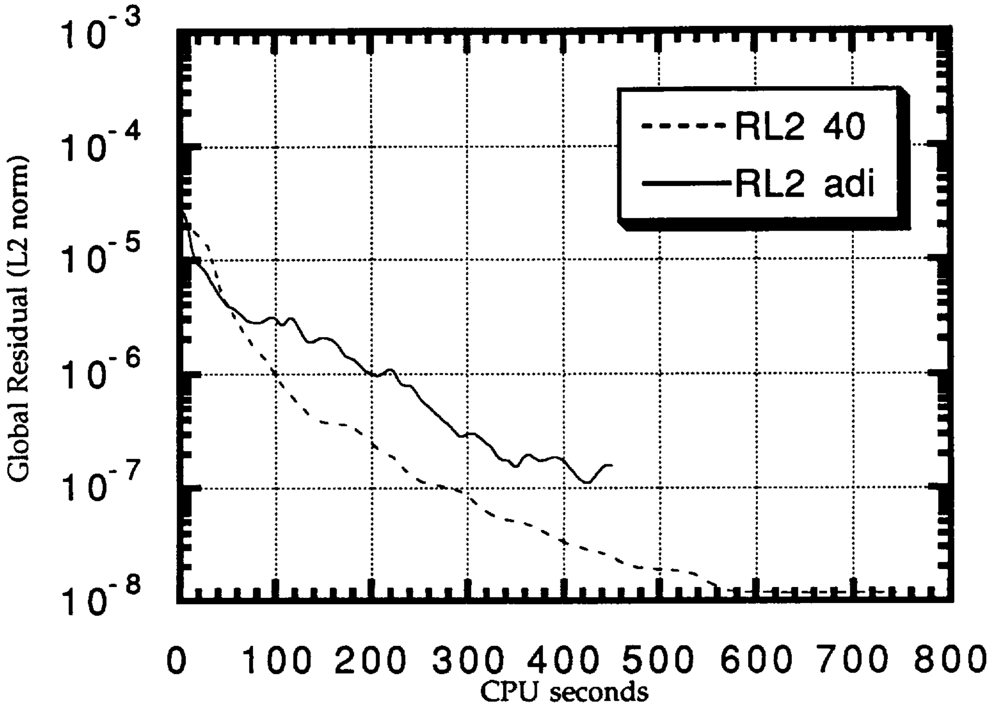


Figure 3

Residual History for a NACA 0012 Airfoil

(M=0.8; α = 1.25 deg)

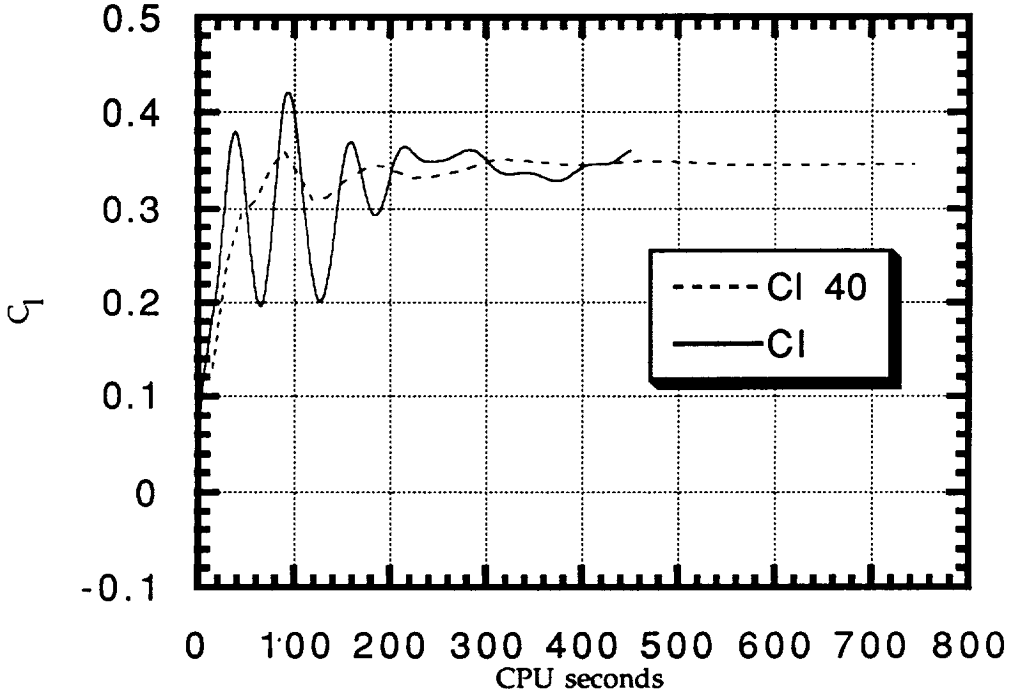


Figure 4

Lift Coefficient Histories for a NACA 0012 Airfoil

(M = 0.8; α = 1.25 deg)

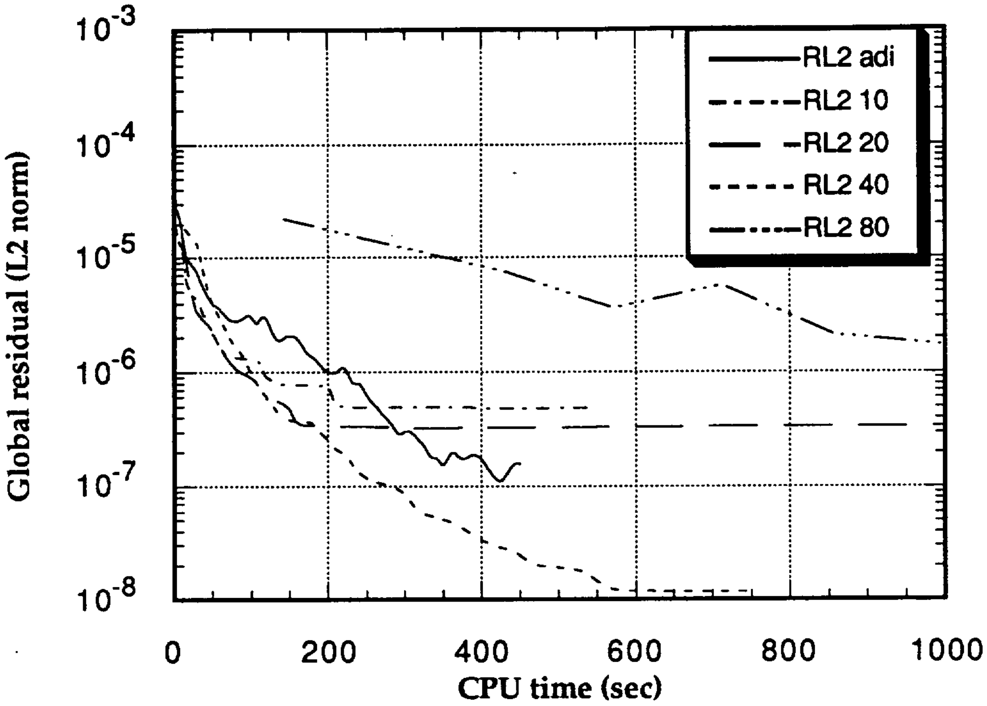


FIGURE 5

Comparison of Residual for Steady Transonic Inviscid Calculation (NACA 0012, M = 0.8, α = 1.25 deg)

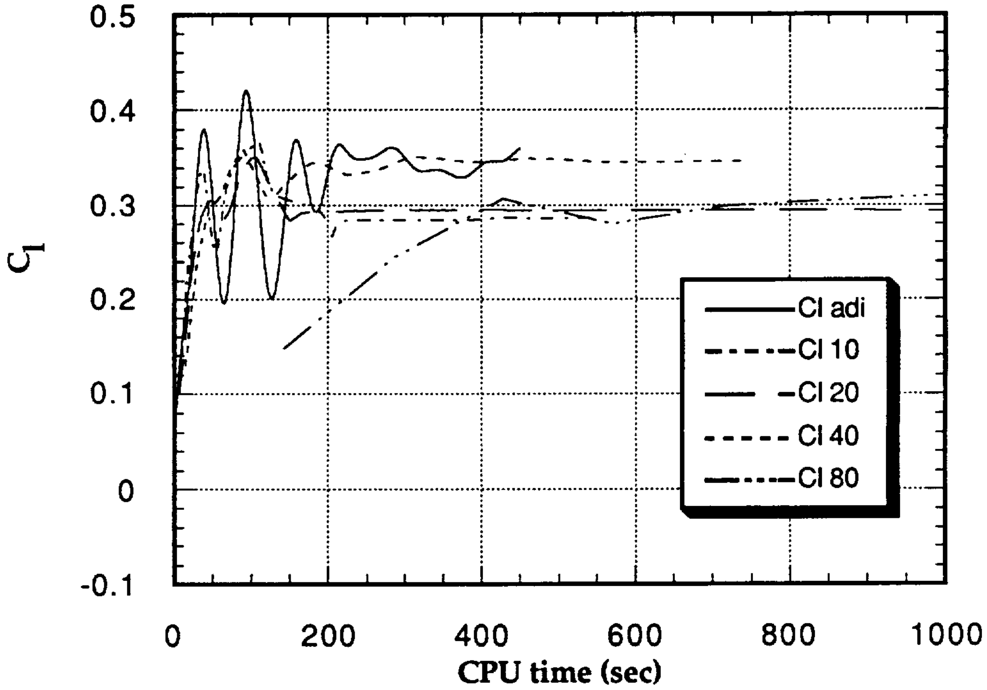
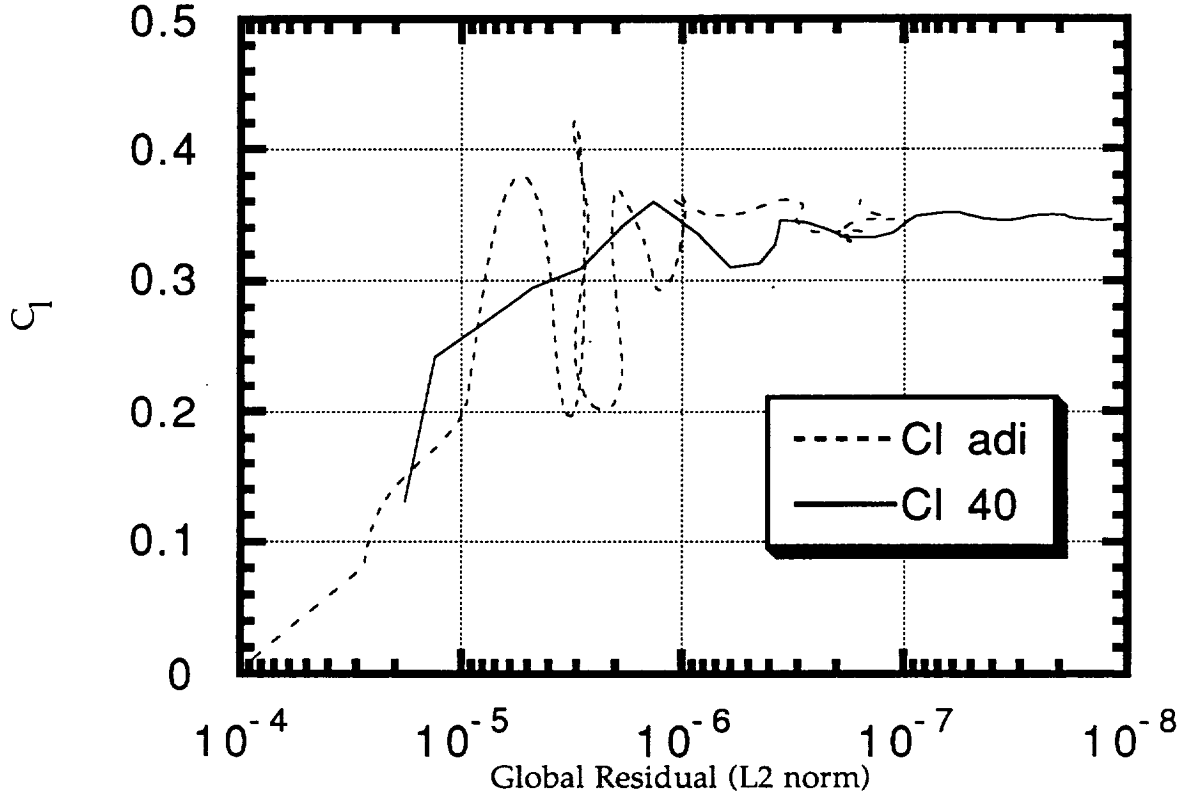


FIGURE 6

Comparison of Lift Coefficient Histories for Steady Transonic Inviscid Calculation (NACA 0012, M=0.8, α = 1.25 deg)



Relation between Global Residual and Lift Coefficient for Transonic Inviscid Test Case (NACA 0012; M = 0.8, α = 1.25 deg)

FIGURE 7

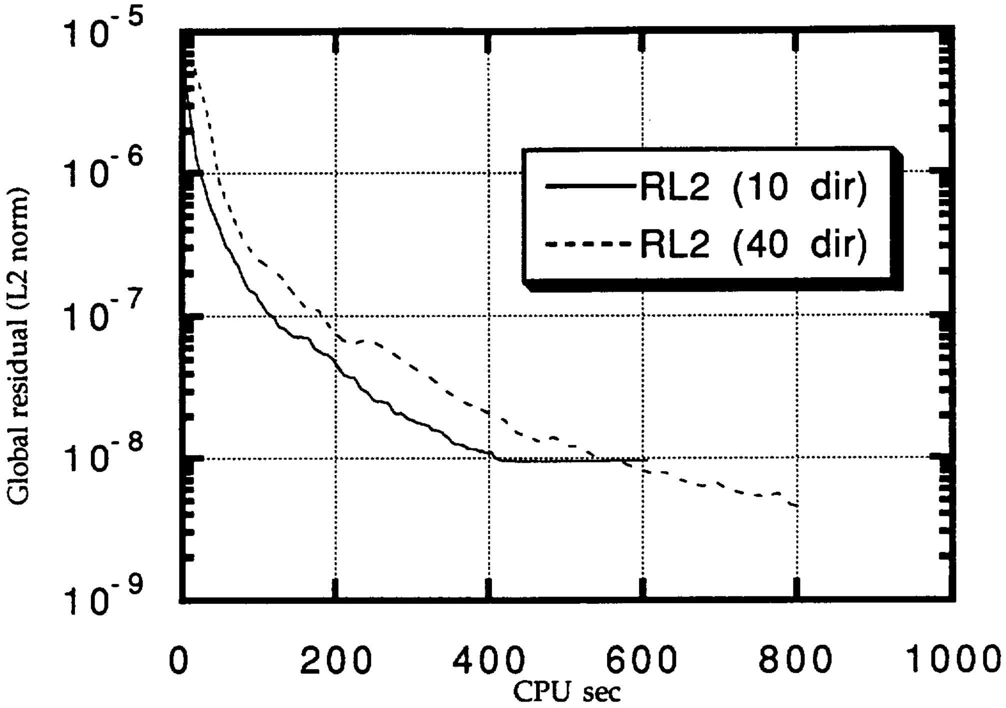


FIGURE 8

Residual History for a NACA 0012 Airfoil
(M=0.283; α=5 deg; Re=3,450,000)

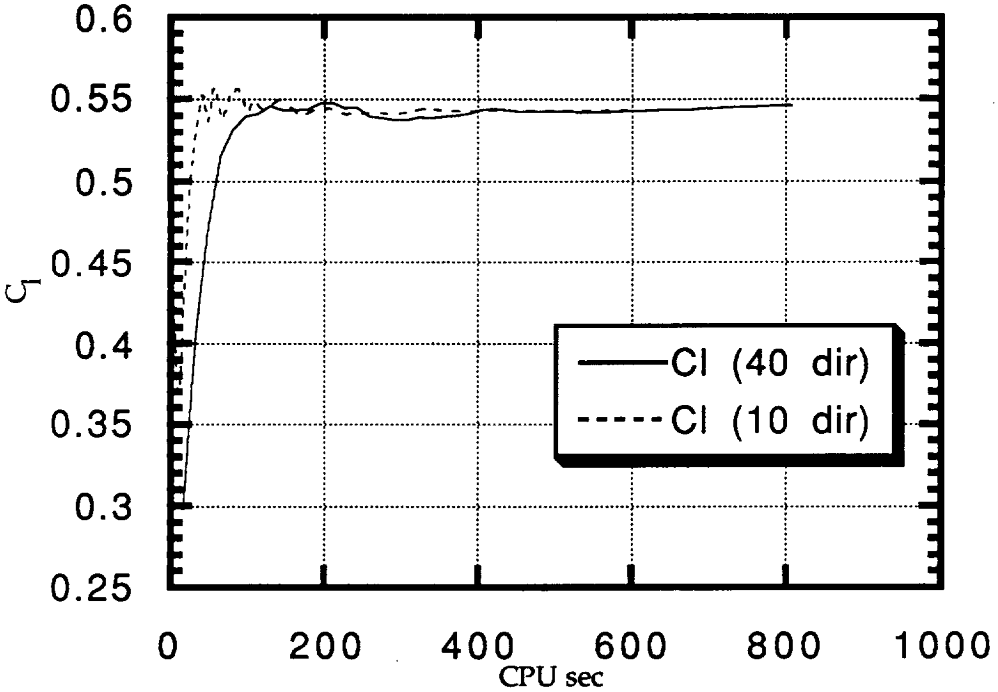


FIGURE 9

Figure 9
Cl Histories for a NACA 0012 Airfoil
(M = 0.283; α = 5 deg.; Re=3,450,000)

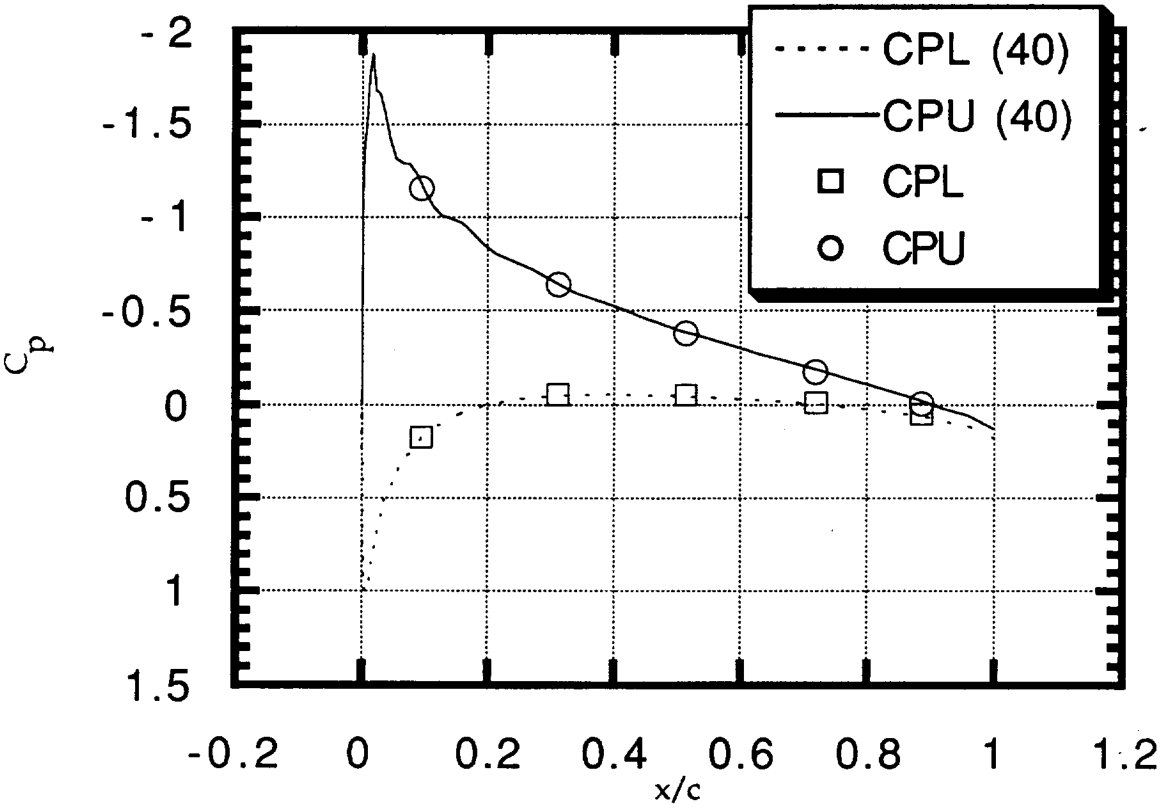
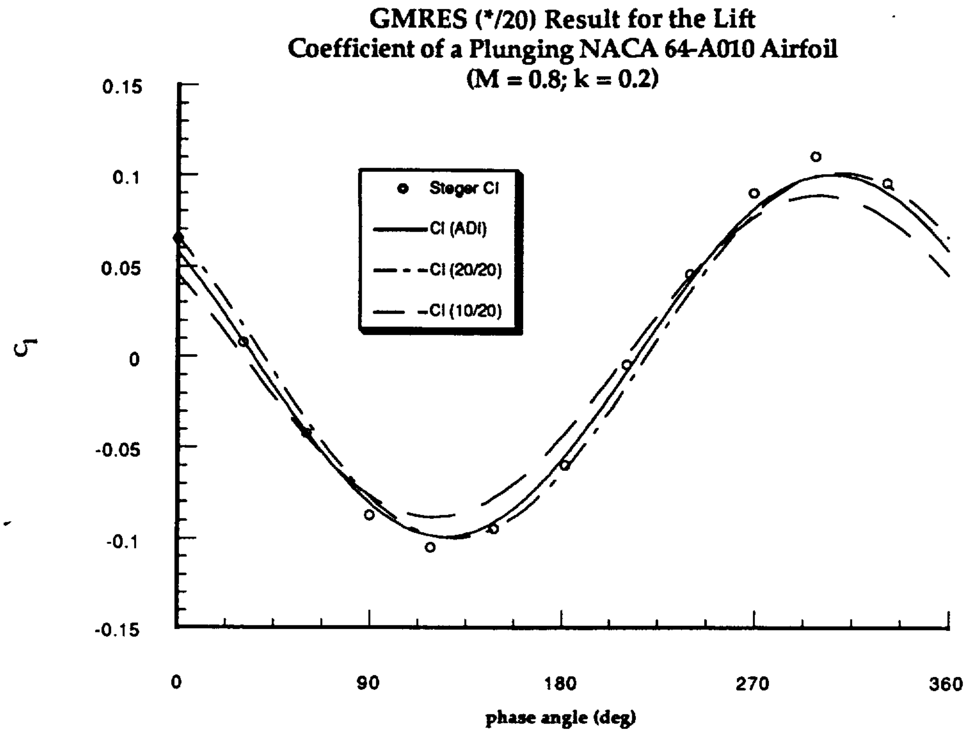


FIGURE 1C

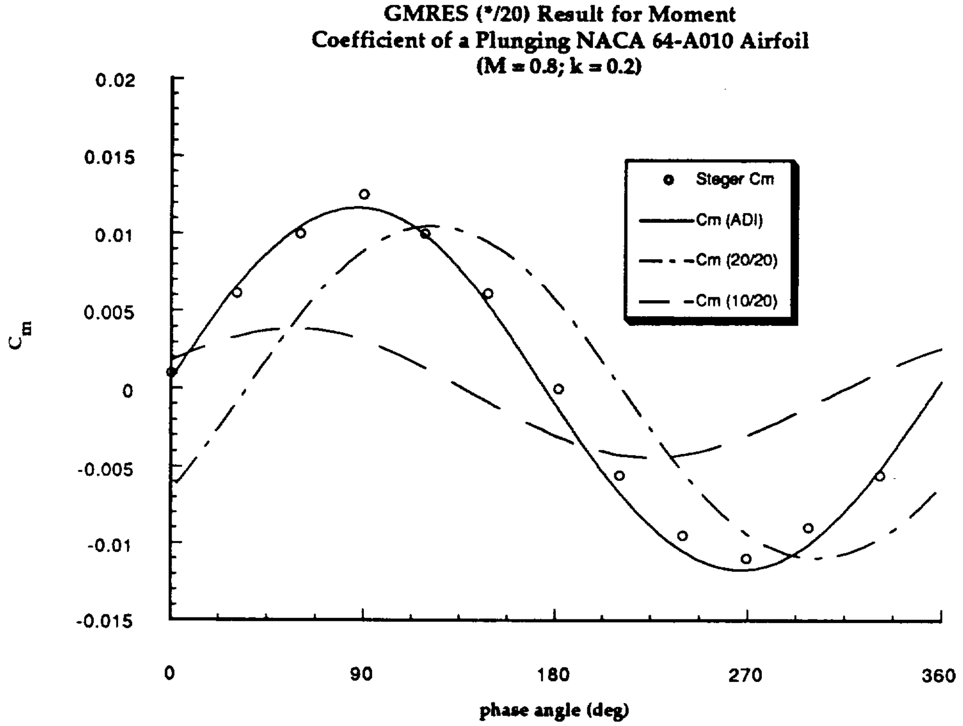
C_p distribution about a NACA 0012 Airfoil

(M = 0.283; α = 5 deg.; Re=3,450,000)



GMRES (*/20) Result for the Lift
Coefficient of a Plunging NACA 64-A010 Airfoil
(M = 0.8; k = 0.2)

FIGURE 11



GMRES (*/20) Result for Moment
Coefficient of a Plunging NACA 64-A010 Airfoil
(M = 0.8; k = 0.2)

FIGURE 12

Comparison of two step GMRES results with one step results for the Lift Coefficient of a Plunging NACA 64-A010 Airfoil (M = 0.8; k = 0.2)

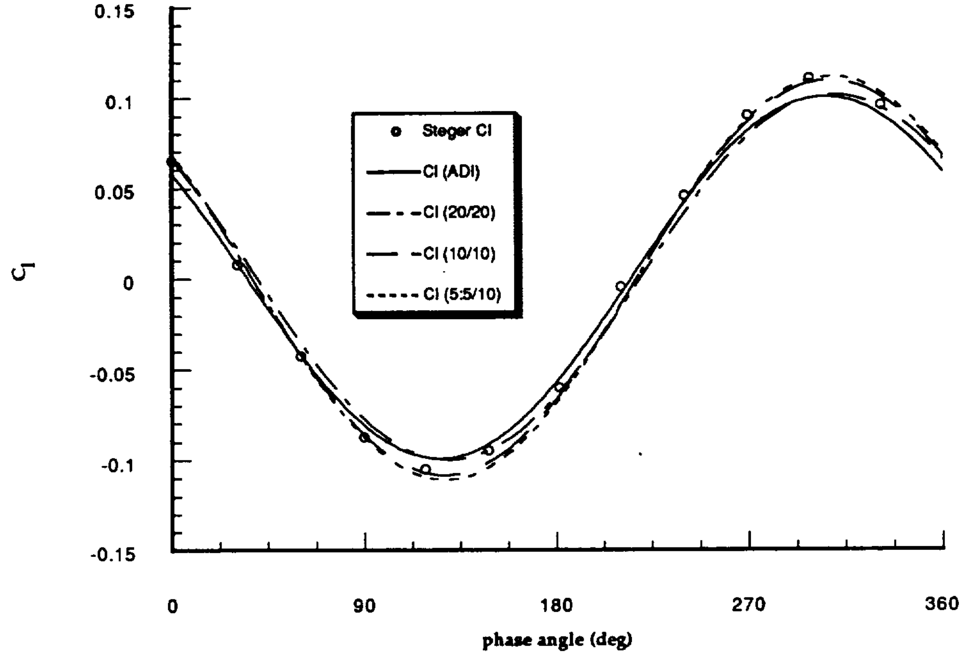


Figure 13

Comparison of two step GMRES results with one step GMRES results for the Moment Coefficient of a Plunging NACA 64-A010 Airfoil (M = 0.8; k = 0.2)

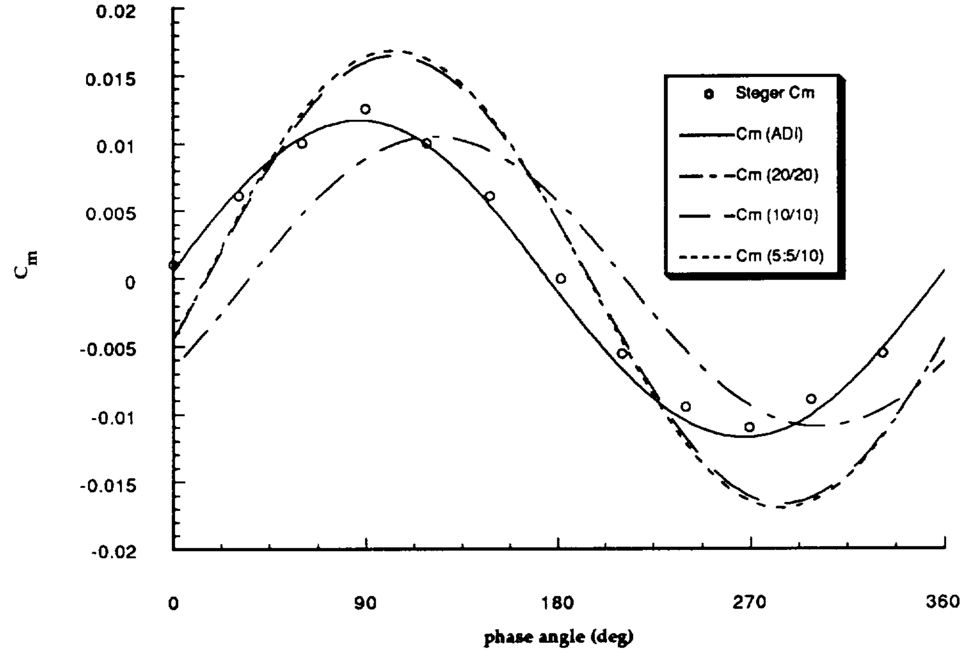


Figure 14

Effect of Time Step on GMRES Result for Lift Coefficient of a Plunging NACA 64-A010 Airfoil (M = 0.8; k = 0.2)

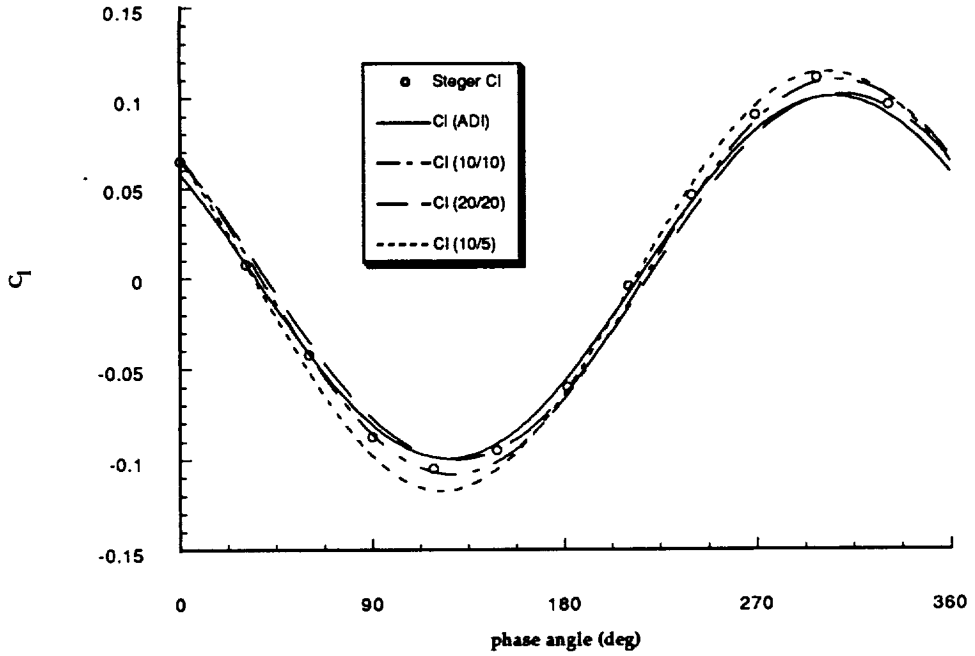


Figure 15

Effect of Time Step on GMRES Result for a Plunging NACA 64-A010 Airfoil (M = 0.8; k = 0.2)

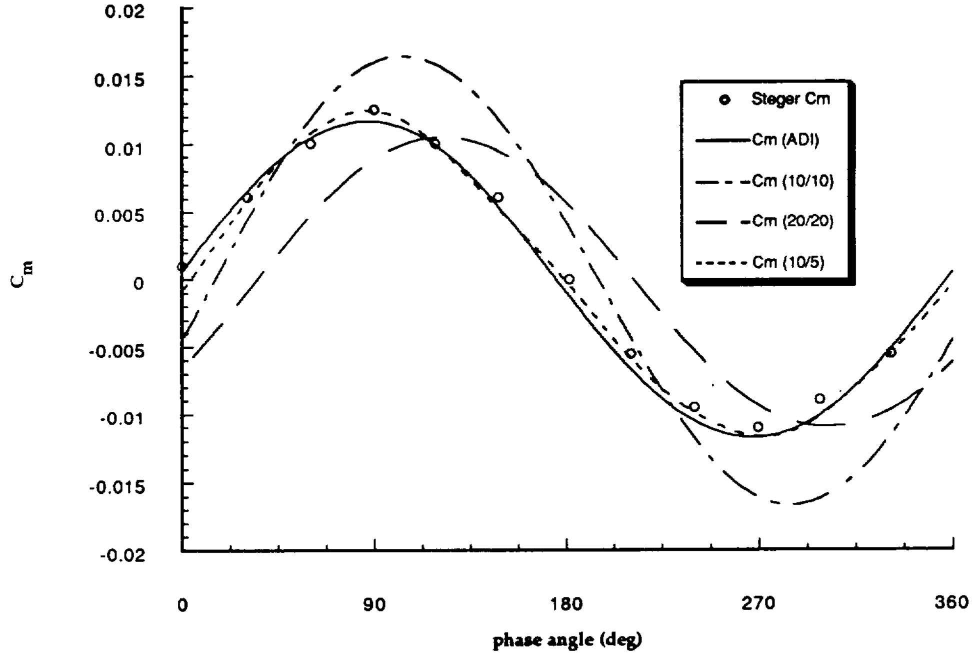


Figure 16

Comparison of GMRES (20/20) with Experimental Results for Lift Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151)

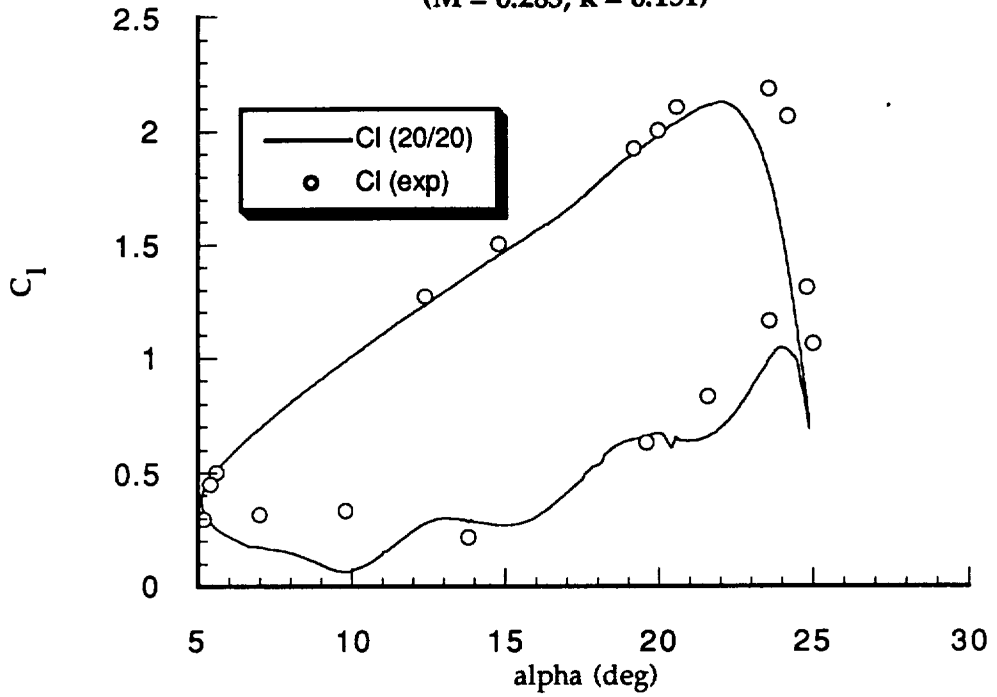


FIGURE 17

Comparison of GMRES (20/20) with Experimental Data for Coefficient of Moment (NACA 0012; M = 0.283; k = 0.151)

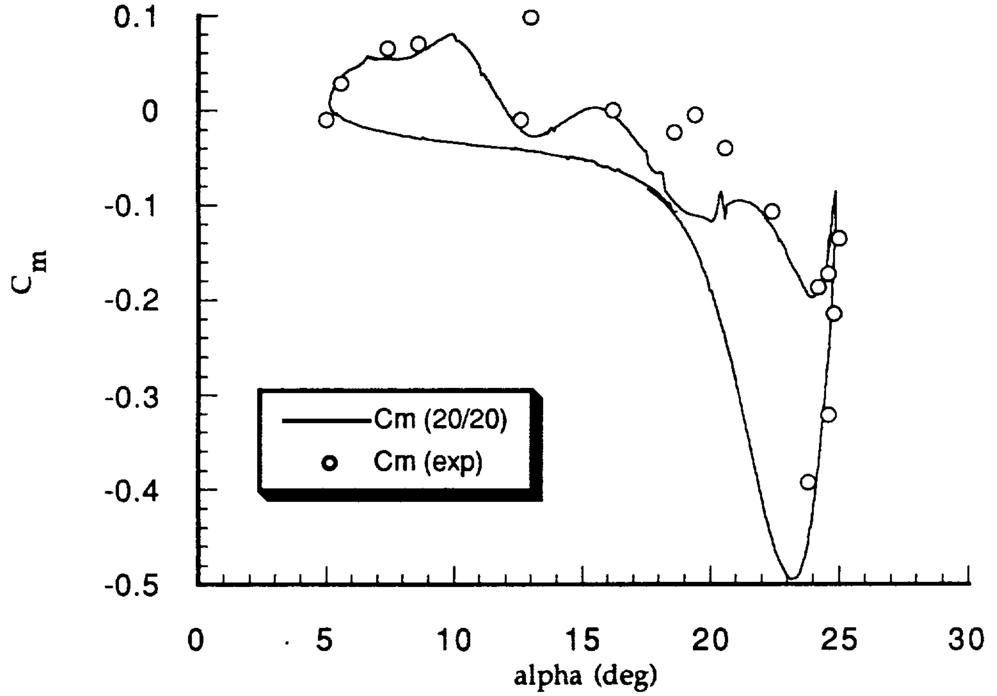
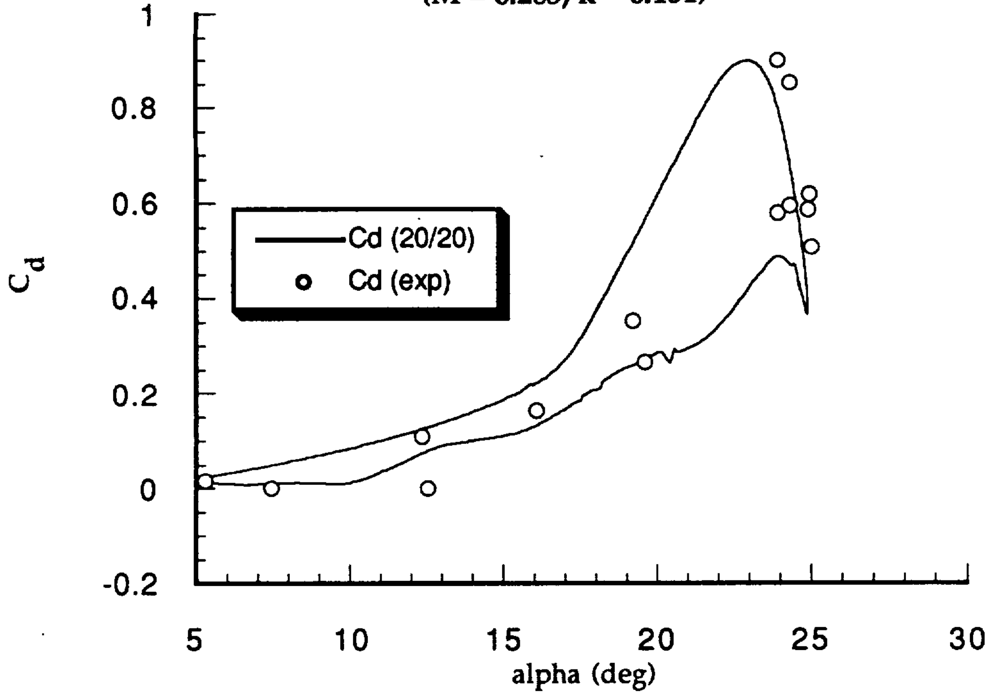
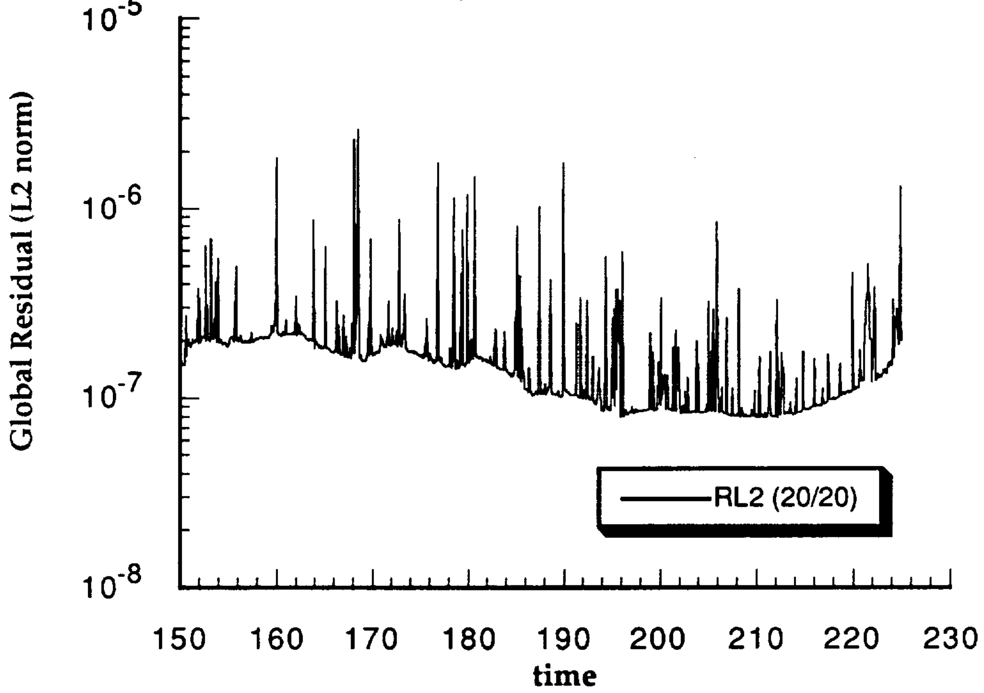


FIGURE 18



Comparison of GMRES(20/20) with Experimental Results for Drag Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151)

Figure 19



GMRES (20/20) L2 Residual results for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

Figure 20

Effect of Directions on GMRES (d/20) Results
for Lift Coefficient of a Pitching NACA 0012 Airfoil
(M = 0.283; k= 0.151; Re = 3,450,000)

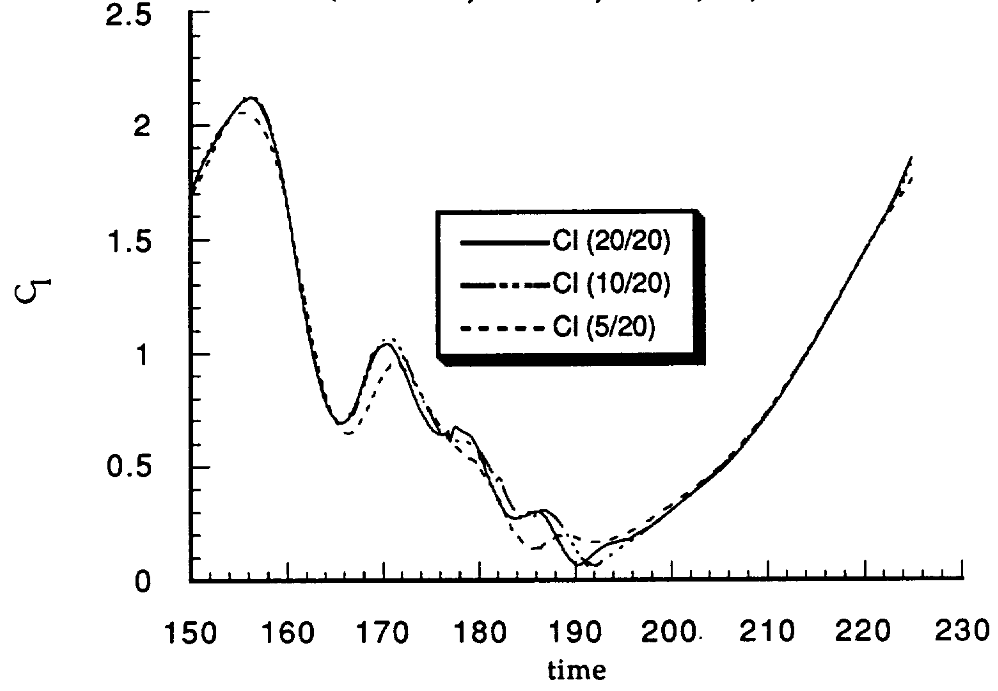


Figure 21

Effect of Directions on GMRES (d/20) Results
for Moment Coefficient of a Pitching NACA 0012 Airfoil
(M = 0.283; k = 0.151; Re = 3,450,000)

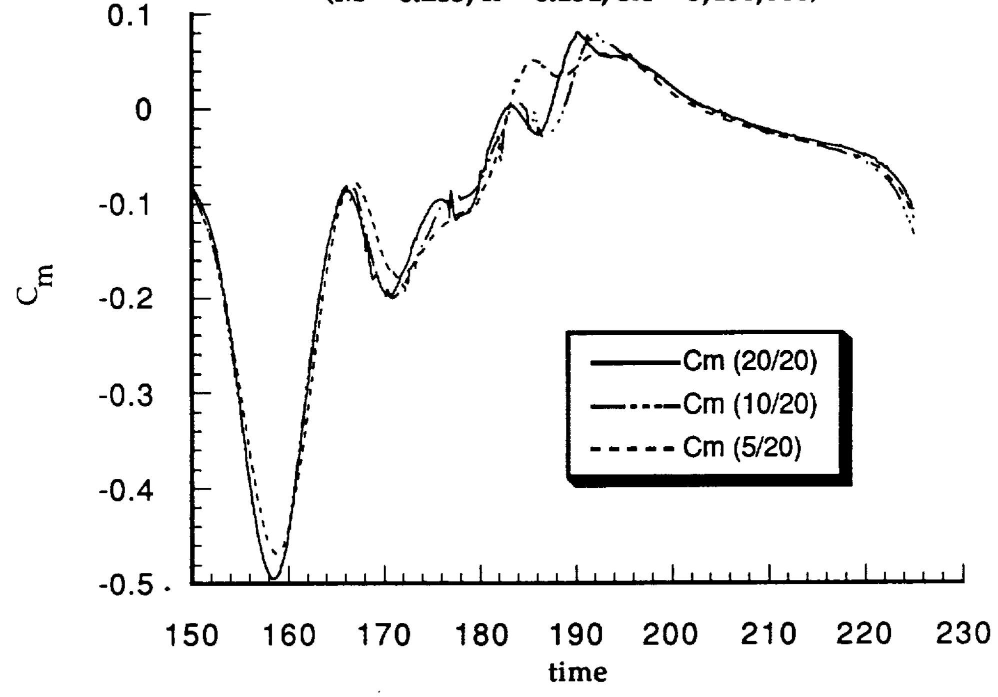


Figure 22

Effect of Directions on Residual of GMRES (d/20) for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

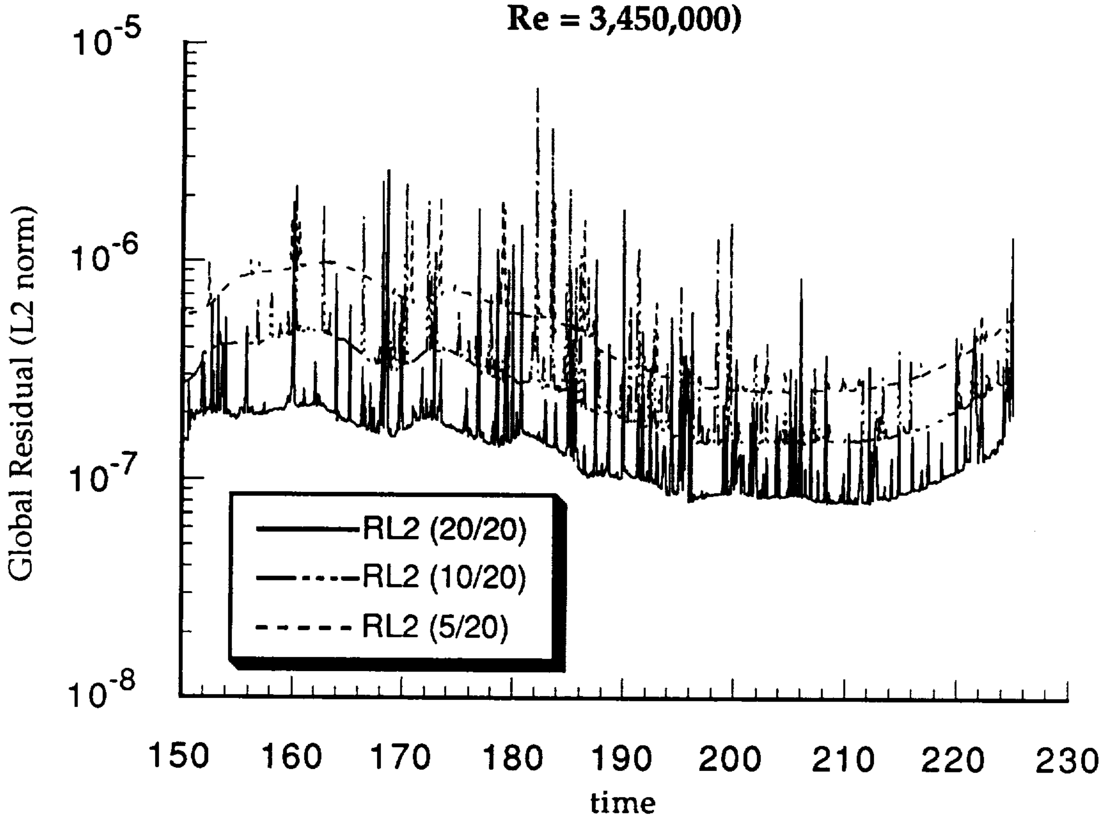


FIGURE 23

Comparison of GMRES (x/2x) results with GMRES (20/20) for Lift of a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

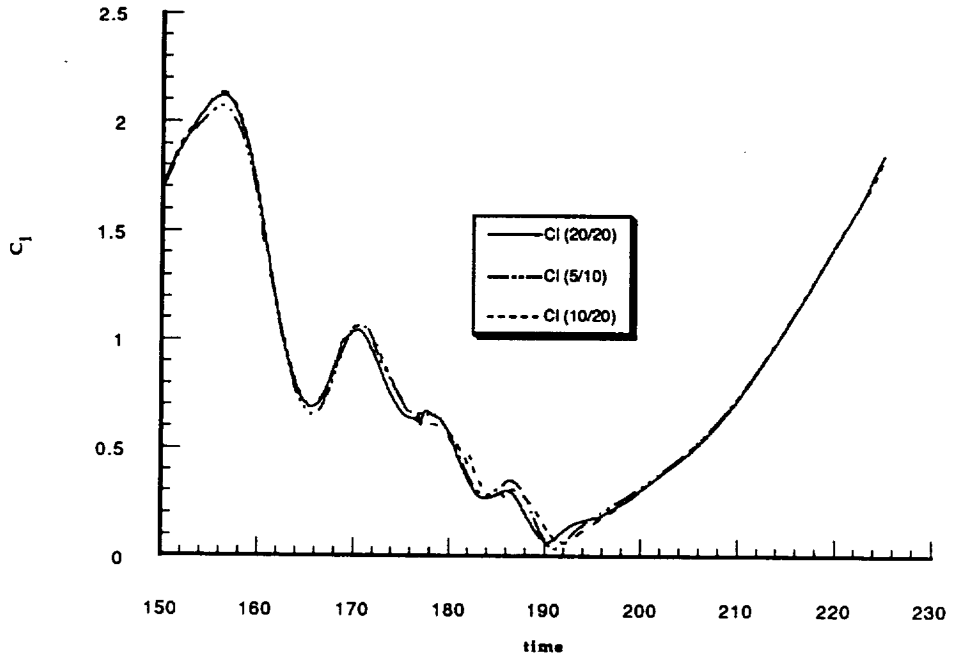


Figure 24

Comparison of GMRES (x/2x) Results with GMRES (20/20) for Lift Coefficient of a Pitching NACA 0012 Airfoil (M = 0.283; k=0.151; Re = 3,450,000)

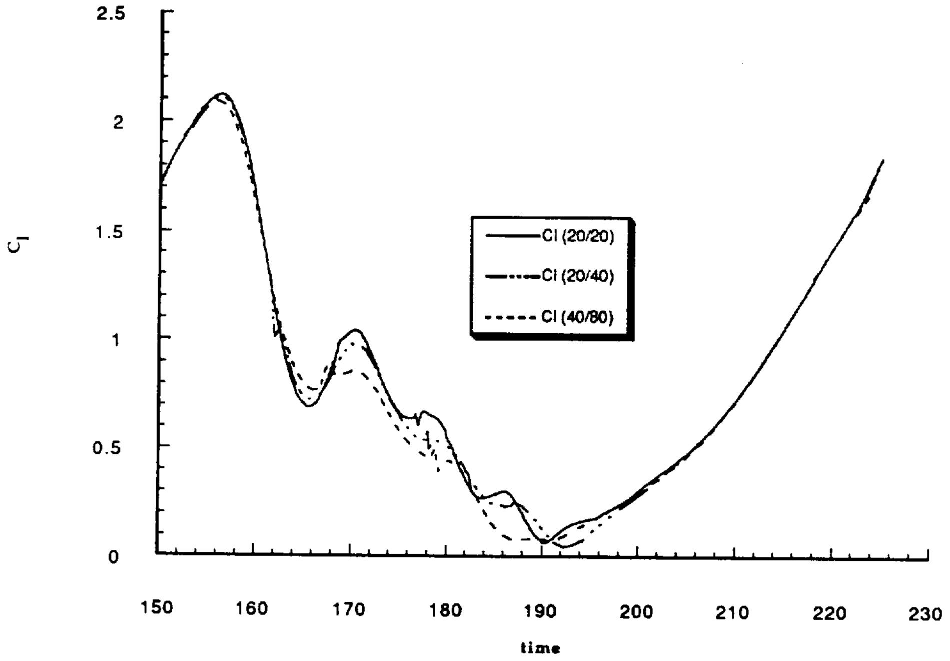


Figure 25

Comparison of GMRES (x/2x) Results with
GMRES (20/20) Results for Moment Coefficient of
a Pitching NACA 0012 Airfoil
(M = 0.283; k = 0.151; Re = 3,450,000)

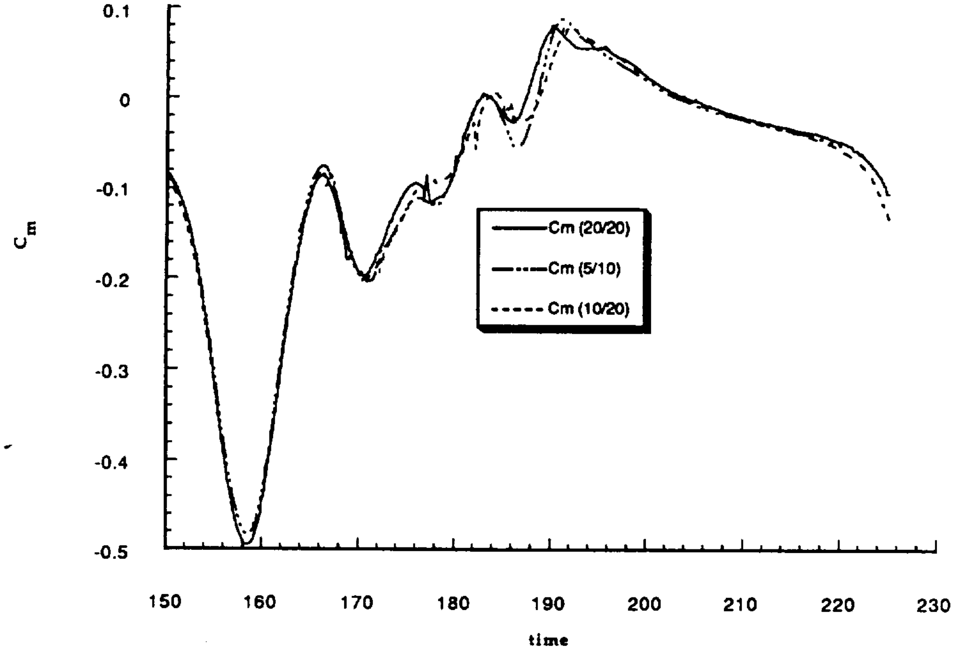


FIGURE 26

Comparison of GMRES (x/2x) Results
with GMRES (20/20) Results for Moment Coefficient
of a Pitching NACA 0012 Airfoil
(M = 0.283; k = 0.151; Re = 3,450,000)

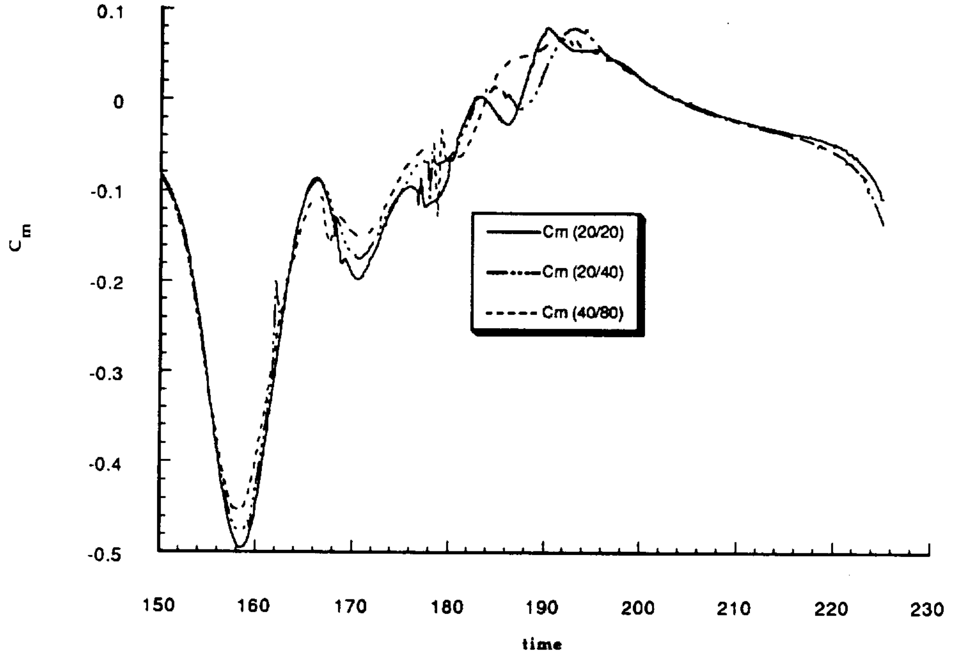


FIGURE 27

Residual History of the GMRES (x/2x) solvers Compared with GMRES (20/20) for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

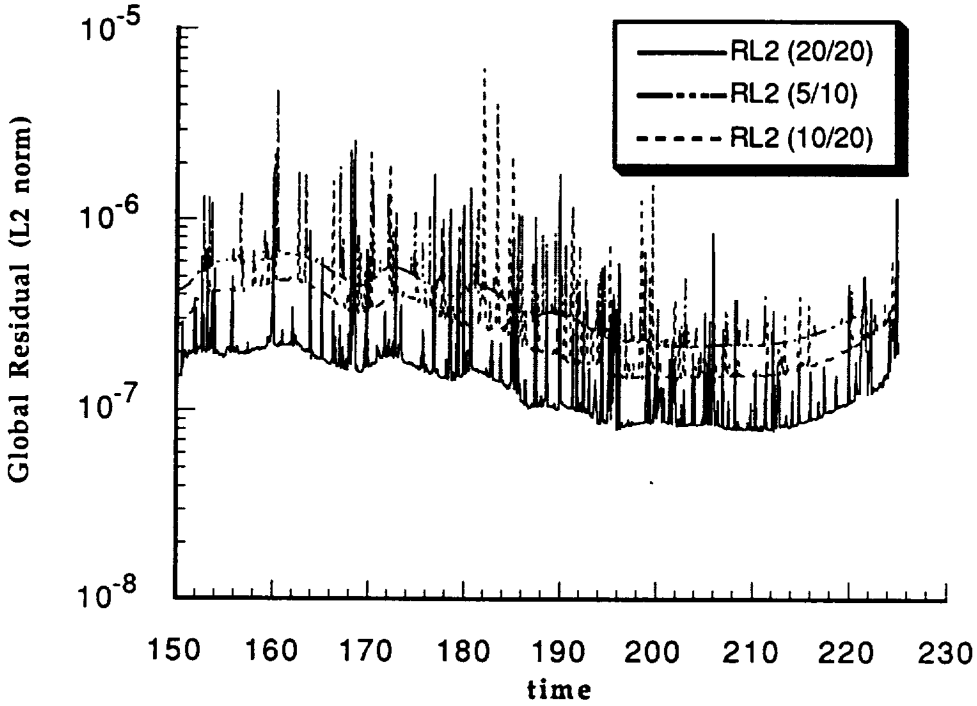


FIGURE 28

Comparison of GMRES (x/2x) with GMRES (20/20) Residual for a Pitching NACA 0012 Airfoil (M = 0.283; k = 0.151; Re = 3,450,000)

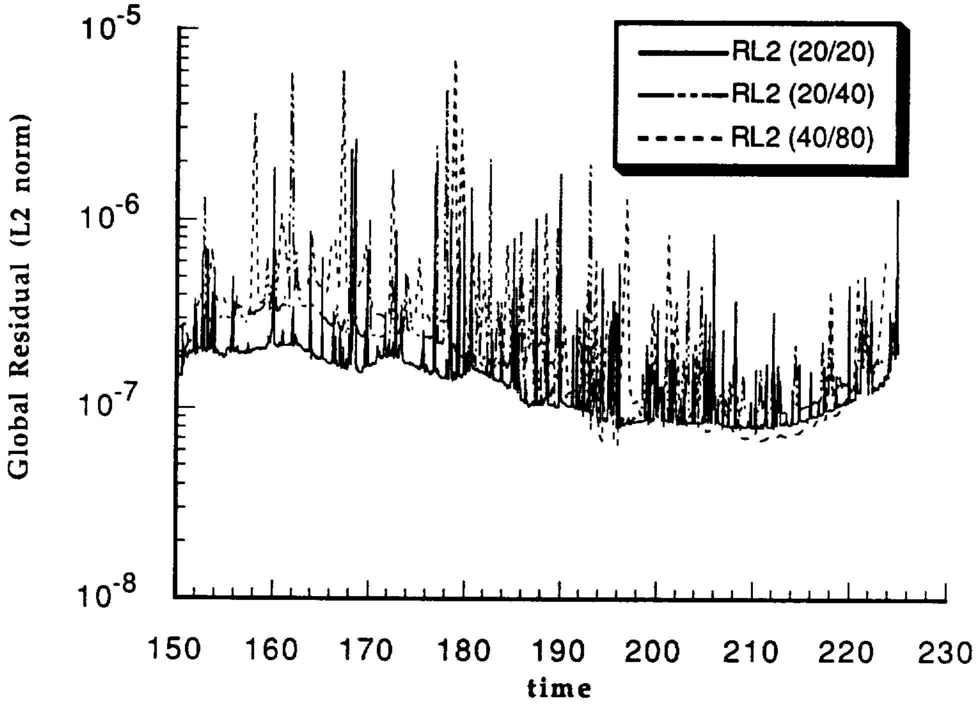


FIGURE 29

Comparison of Multigrid GMRES (10/20)
Lift Coefficients for a Pitching NACA 0012 Airfoil
(M = 0.283; k = 0.151; Re = 3,450,000)

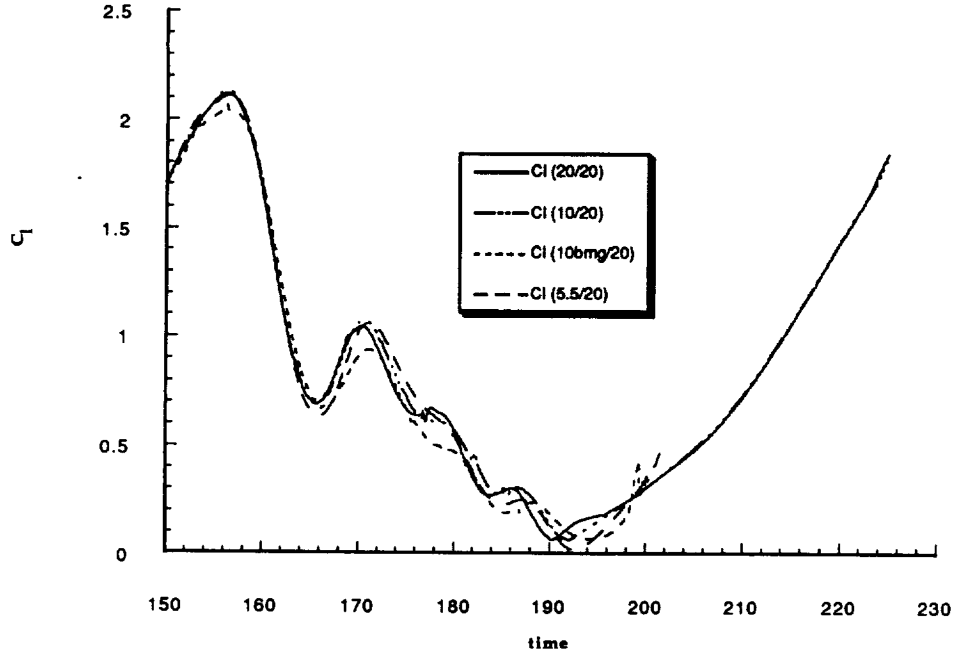


FIGURE 30

Comparison of Multigrid GMRES (10/20) Results
for Moment Coefficients of a Pitching NACA 0012 Airfoil
(M = 0.283; k = 0.151; Re = 3,450,000)

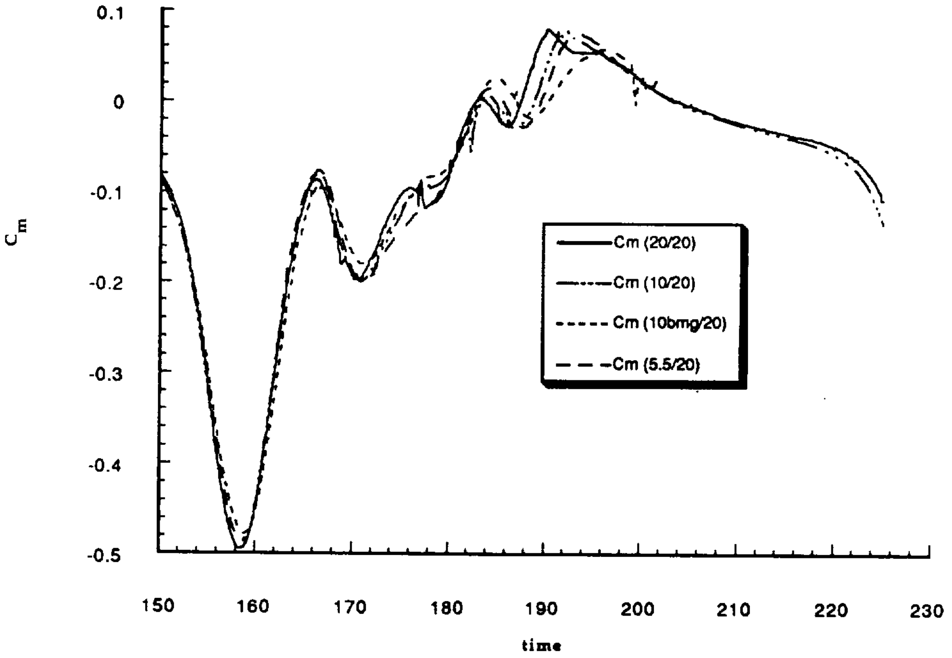


FIGURE 31

Comparison of Residuals for Multigrid
GMRES (10/20) for a Pitching NACA 0012 Airfoil
(M = 0.283; k = 0.151; Re = 3,450,000)

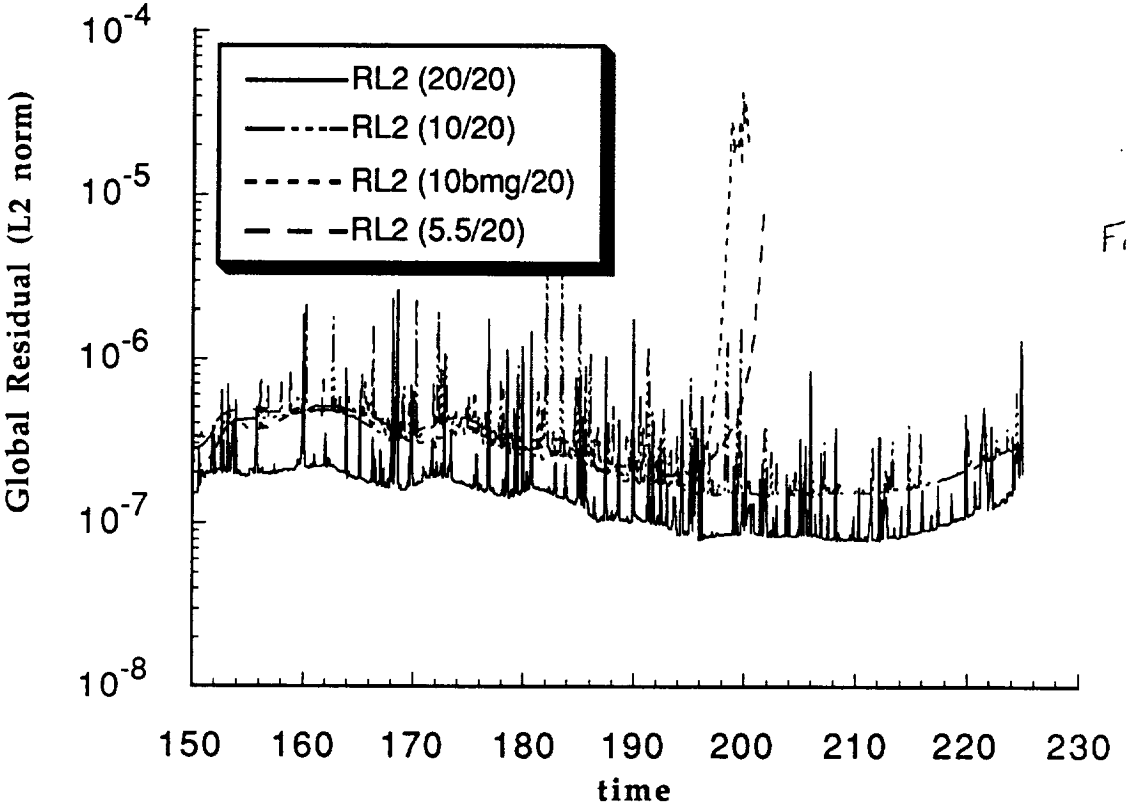


FIGURE 32