

(NASA-CR-174326) VECTORIZABLE MULTIGRID  
ALGORITHMS FOR TRANSONIC FLOW CALCULATIONS  
M.S. Thesis (George Washington Univ.) 100 p  
HC A05/MF A01 CSCL 01A

N85-16827

Unclas

G3/02 13754

VECTORIZABLE MULTIGRID ALGORITHMS FOR  
TRANSONIC FLOW CALCULATIONS

by

Norman Duane Melson  
B.S. in Aerospace and Ocean Engineering  
Virginia Polytechnic Institute and State University  
1979

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  
January 1985



## ABSTRACT

The analysis and incorporation into a multigrid scheme of several vectorizable algorithms are discussed. von Neumann analyses of vertical line, horizontal line, and alternating direction ZEBRA algorithms were performed; and the results were used to predict their multigrid damping rates. The algorithms were then successfully implemented in a transonic conservative full-potential computer program. The convergence acceleration effect of multiple grids is shown and the convergence rates of the vectorizable algorithms are compared to the convergence rates of standard successive line overrelaxation (SLOR) algorithms.

## ACKNOWLEDGMENTS

I would like to express my appreciation to the National Aeronautics and Space Administration, Langley Research Center, for the time given to me and for the use of the facilities needed to accomplish the necessary research for this thesis. I would also like to thank my faculty advisor, Dr. Douglas L. Dwoyer, and Dr. Frank C. Thames and Mr. Jerry C. South, Jr., for all their advice and suggestions.

TABLE OF CONTENTS

ABSTRACT.....i  
ACKNOWLEDGMENTS.....ii  
TABLE OF CONTENTS.....iii  
LIST OF FIGURES.....iv  
LIST OF SYMBOLS.....vi  
INTRODUCTION.....1  
THE POTENTIAL MODEL FOR TRANSONIC FLOW.....4  
    Continuum Formulation.....4  
    Discrete Formulation.....6  
THE MULTIGRID METHOD.....8  
    Linear Equations.....9  
    Nonlinear Equations.....12  
    Restriction Operators.....14  
    Prolongation Operators.....15  
SMOOTHING ALGORITHMS.....15  
    Algorithms.....15  
    Smoothing Analysis.....18  
    A von Neumann Analysis of VLOR.....21  
    A von Neumann Analysis of VZEB1.....27  
RESULTS.....32  
CONCLUSIONS.....38  
APPENDICES.....40  
    Appendix A.....40  
    Appendix B.....44  
REFERENCES.....49  
TABLE 1.....52  
FIGURES.....62

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.	Coordinate system and schematic of stretched grid.....	62
2.	Prolongation operator.....	63
3.	Schematic of VLOR algorithm.....	64
4.	Schematic of HLOR algorithm.....	65
5.	Schematic of VZEB1 algorithm.....	66
6.	von Neumann damping rate contours for VLOR.....	67
7.	Multigrid smoothing factor contours for VLOR.....	68
8.	von Neumann damping rate and multigrid smoothing factor contours for VZEB1.....	69
	(a) von Neumann damping rate contours.	
	(b) Multigrid smoothing factor contours.	
9.	Multigrid smoothing factor versus $M$ and $\lambda$ for VZEB1.....	70
10.	Multigrid smoothing factor versus $M$ and $\lambda$ for HZEB1.....	71
11.	Multigrid smoothing factor versus $M$ and $\lambda$ for ADZEB1.....	72
12.	Potential on finest grid of first cycle using VZEB1.....	73
13.	Residual on finest grid using VZEB1 ( $M_\infty = 0.1$ ).....	74
14.	End view of residual on finest grid using VZEB1 ( $M_\infty = 0.1$ ).....	75

<u>Figure</u>	<u>Page</u>
15. Forcing function on second finest grid using VZEB1 and no residual weighting ( $M_\infty = 0.1$ ).....	76
16. Forcing function on second finest grid using VZEB1 and residual weighting ( $M_\infty = 0.1$ ).....	77
17. Effect of residual weighting on VZEB1 ( $M_\infty = 0.1$ ).....	78
18. Restricted potential from finest grid using VZEB1 ( $M_\infty = 0.1$ ).....	79
19. Effect of multigrid on VZEB1 ( $M_\infty = 0.1$ ).....	80
20. Effect of $M_\infty$ and grid stretching on VZEB1.....	81
21. Effect of multigrid on VZEB1 ( $M_\infty = 0.8$ ).....	82
22. Comparison of VZEB1 and VLOR in multigrid ( $M_\infty = 0.8$ , 5 grids).....	83
23. Effect of multigrid on HZEB1 ( $M_\infty = 0.1$ ).....	84
24. Effect of $M_\infty$ and grid stretching on HZEB1.....	85
25. Effect of multigrid on HZEB1 ( $M_\infty = 0.8$ ).....	86
26. Comparison of HZEB1 and HLOR in multigrid ( $M_\infty = 0.8$ , 5 grids).....	87
27. Effect of $M_\infty$ and grid stretching on ADZEB1.....	88
28. Comparison of ADZEB1 and ADLOR in multigrid ( $M_\infty = 0.8$ , 5 grids).....	89

## LIST OF SYMBOLS

a	nondimensional speed of sound or as defined in equations (33), (B3), and (B6)
A	defined in figure 3
b	defined in equations (33), (B3), and (B6)
c	defined in equations (33), (B3), and (B6)
C	defined in figure 2
d	defined in equations (33) and (B3)
D	defined in figure 2
e	error
E	displacement operator
f	forcing function
$F_x$	x-direction grid stretching parameter
$F_y$	y-direction grid stretching parameter
g	von Neumann damping rate or amplification factor
G	grid
h	cell size of a grid

$i$  indicates value at  $i^{\text{th}}$  vertical line in a grid  
 $I$  grid transfer operator or identity operator (if no subscript or superscript)  
 $j$  indicates value at  $j^{\text{th}}$  horizontal line in a grid  
 $k$  dummy variable in equation (A7)  
 $L$  operator  
 $M$  Mach number  
 $M_s$  switch Mach number  
 $n$  number of unknowns in set of finite-difference equations  
 $\vec{n}$  unit normal vector  
 $R$  residual  
 $u$  approximation to  $U$  or nondimensional velocity in  $x$ -direction as context indicates  
 $U$  solution  
 $v$  nondimensional velocity in  $y$ -direction  
 $V$  error  
 $V_\infty$  dimensional free-stream velocity  
 $\vec{V}$  dimensional velocity vector



x	coordinate direction aligned with free stream
y	coordinate direction normal to free stream
Y(x)	function defining airfoil shape
z	$\sqrt{q}$
$\beta$	free parameter, coefficient of $\phi_{xt}$
$\gamma$	ratio of specific heats or as otherwise defined in equation (B5)
$\delta$	finite-difference operator
$\delta_{xx}^+$	upwind finite difference
$\partial$	partial derivative
$\Delta$	finite change in quantity
$\eta$	defined in equation (B5)
$\theta$	phase angle (in degrees or radians)
$\lambda$	grid aspect ratio
$\mu$	multigrid smoothing factor
$\pi$	3.14159 . . .
$\rho$	density
$\bar{\rho}$	artificial density

$\tau$	artificial density method switch function
$\phi$	potential
$\hat{\phi}$	exact potential solution
$\tilde{\phi}$	potential solution which makes residual zero
$\psi$	generic function
$\omega$	overrelaxation factor
$\cap$	intersection of two sets
$\in$	is an element of set . . .

Subscripts

$i$	value at the $i^{\text{th}}$ vertical line in the grid
$j$	value at the $j^{\text{th}}$ horizontal line in the grid
new	value at new time level
$o$	value at arbitrary location
old	value at old time level
$t$	derivative in time
$x$	derivative in x-direction
$y$	derivative in y-direction
$\infty$	value based on free-stream conditions

### Superscripts

h            quantity on grid with spacing h

n            quantity at n<sup>th</sup> time level

+            defined in equation (23)

-            defined in equation (23)

## INTRODUCTION

Research in the computation of transonic flows in recent years has been devoted to improving the accuracy, speed, and geometric capability of computational tools. Most three-dimensional transonic codes use the potential approximation to model the flow-field physics and the successive line overrelaxation algorithm (SLOR) to solve the resulting system of equations. These three-dimensional transonic codes, however, still require large amounts of computer resources and are therefore expensive to use extensively. Thus, there is a need for improving the computational efficiency and reducing the cost of these codes.

Two ways to increase the computational efficiency of a program are to improve its computation rate and the convergence rate of its algorithm. The computation rate of a program can be improved through efficient program coding and the use of faster computers (ref. 1). The efficiency of the coding of a program is programmer dependent, and it is not truly an area for research as much as a topic for education. Improvements in computer speed are available but usually through specialized computer architecture such as the Control Data Corporation<sup>®</sup> CYBER 203 and CYBER 205 computer systems. Vector architecture places certain restrictions on the algorithm in the program, and significant research has been devoted to the development of algorithms which can satisfy these restrictions. Algorithms which can make use of

the vector nature of the machines are referred to as vectorizable. Unfortunately, the workhorse algorithm of transonic potential codes, SLOR, is not completely vectorizable. Even if SLOR were completely vectorizable, it still has a very slow convergence rate for transonic flow problems.

The ZEBRA algorithms of South, Keller, and Hafez (ref. 2) are a promising new class of vectorizable algorithms. Although vectorizable, the convergence rate for the ZEBRA algorithms is approximately the same as that for SLOR. The number of operations required to obtain a solution is proportional to the square of the number of points ( $O(n^2)$ ) in the flow field, where  $n$  is the number of points in the flow field for potential flow calculations. Hence, for very fine grids, programs using the SLOR and ZEBRA algorithms are too expensive for extensive use.

The concept of using multiple grids to accelerate the iterative solution of a set of finite difference equations was first proposed by Federenko (refs. 3 and 4) in 1961. The multiple grid concept was further analyzed by Bakhvalov (ref. 5); but it was not until Brandt (ref. 6) extended the technique and applied it to an elliptic set of equations that multigrid started to gain acceptance.

Federenko showed that the solution for a set of equations could be obtained in  $O(n)$  operations. Convergence in  $O(n)$  operations using multigrid was proven for fairly general conditions by Nicolaides (ref. 7) and Hackbush (ref. 8). SLOR algorithms take  $O(n^2)$  operations to obtain a solution, so it

is quite obvious that multiple grid (multigrid) techniques are quite attractive.

In recent years, multigrid has gained wide acceptance as an acceleration technique. It was first applied to two-dimensional nonlifting transonic potential flow calculations in 1977 by South and Brandt (ref. 9) to accelerate an SLOR algorithm. This work was extended by Jameson (ref. 10) in 1979 to lifting airfoils but with an alternating direction implicit (ADI) type algorithm. Multigrid was applied to three-dimensional transonic potential flow calculations by Caughey (ref. 11) in 1983 using a pentadiagonal form of an SLOR-type algorithm. Although tremendous improvements in convergence rates were observed in each of these studies, the algorithms used with the multigrid limit the vectorizability of the programs.

In an attempt to take advantage of the fast convergence rate obtainable with multigrid and yet retain a high degree of vectorizability, multigrid is used to accelerate the vectorizable ZEBRA algorithms in the present study. Details of multigrid, the ZEBRA algorithms and their smoothing rates as compared to SLOR, and the results of the incorporation of ZEBRA algorithms in multigrid are given below. To reduce run costs, a two-dimensional problem is considered rather than a three-dimensional problem.

## THE POTENTIAL MODEL FOR TRANSONIC FLOW

### Continuum Formulation

Consider the flow of a perfect gas past an airfoil with a shape defined by the function  $Y(x)$ . If irrotational flow is assumed, then the dimensional velocity,  $\vec{V}$ , satisfies

$$\nabla \times \vec{V} = 0$$

This condition is satisfied by the introduction of a reduced (or disturbance) potential function,  $\phi$ , such that

$$\vec{V} = V_\infty \mathbf{x} + V_\infty \vec{\nabla} \phi$$

In conservation form, the two-dimensional potential equation is:

$$(\rho u)_x + (\rho v)_y = 0 \quad (1)$$

where  $\rho$  is the isentropic density:

$$\rho = (M_\infty^2 a^2)^{1/(\gamma - 1)}$$

and

$$\begin{aligned} a^2 &= 1/M_\infty^2 + (\gamma - 1) (1 - u^2 - v^2) / 2 \\ u &= 1 + \phi_x \\ v &= \phi_y \end{aligned} \quad (2)$$

where  $\rho$  and  $a$  have been normalized by their free-stream values and  $u$  and  $v$  have been normalized by  $V_\infty$ . Equation (1) is nonlinear and is strongly affected by the Mach number. This effect is more clearly seen if equation (1) is rewritten in nonconservation form:

$$(a^2 - u^2) \phi_{xx} + (a^2 - v^2) \phi_{yy} - 2uv\phi_{xy} = 0$$

Note that as  $u \rightarrow a$  (or  $M \rightarrow 1$ ), the coefficient of  $\phi_{xx}$  approaches 0. If  $u > a$  ( $M > 1$ ), the equation changes type and becomes hyperbolic instead of elliptic.

In the present study, subcritical and supercritical flows over a symmetric, nonlifting 12-percent-thick parabolic arc airfoil are considered. The airfoil boundary condition is:

$$\frac{\partial \psi}{\partial x} = \frac{v}{u}$$

This boundary condition was applied at the airfoil chord line ( $y = 0$ ). Since only nonlifting cases are considered, symmetry boundary conditions ( $\phi_y = 0$ ) are used fore and aft of the airfoil section. Zero-disturbance boundary conditions ( $\phi = 0$ ) are used at the upstream and downstream extents of the grid as well as the top boundary of the grid. (See fig. 1.)



### Discrete Formulation

Equation (1) may be solved at discrete points by rewriting it in a finite-difference formulation. The artificial density method is used in the present work to introduce dissipation. The discrete form of equation (1) with artificial density is:

$$\frac{\delta(\bar{\rho}u)}{\delta x} + \frac{\delta(\bar{\rho}v)}{\delta y} = 0$$

where

$$\bar{\rho} = \rho - \tau \Delta x \rho_x, \quad (3)$$

$\rho$  is the isentropic density, and

$$\tau = \max [0, 1 - (1/M)^2] \quad (4)$$

is the switching factor.

In the present work, a modification to the switching factor,  $\tau$ , was used of the form:

$$\tau = \max [0, 1 - (M_s/M)^2] \quad (5)$$

where  $M_s$  is an input switch Mach number. By setting  $M_s < 1$ , the additional damping of the artificial density method can be introduced at subsonic points near the sonic region. This has been found to aid convergence (ref. 10).

Boundary conditions as noted in figure 1 were used. Ghost points were used to allow central differences for the Neumann boundary conditions at the airfoil section and the symmetry line.

A Cartesian grid with "interest formula" grid stretching is used in the present study. In the y-direction (see fig. 1), grid stretching begins at the airfoil where:

$$\Delta y_{j+1} = F_y * \Delta y_j \quad (6)$$

The parameter,  $F_y$  is an input to the program and varies from 1.0 for no grid stretching (uniform grid) to higher numbers such as 1.1 for 10-percent grid stretching. In the x-direction (see fig. 1), a uniform grid is used above the airfoil. Forward of the leading edge,

$$\Delta x_{i-1} = F_x * \Delta x_i \quad (7)$$

Aft of the trailing edge,

$$\Delta x_{i+1} = F_x * \Delta x_i \quad (8)$$

The parameter,  $F_x$  is an input to the program and is similar to  $F_y$  except it is used for the x-direction stretching. The local grid aspect ratio is defined as

$$\lambda = \Delta x / \Delta y \quad (9)$$

where  $\Delta x$  and  $\Delta y$  are the grid spacing in the x- and y-directions, respectively, as shown in figure 1.

For the subcritical test cases run in the present study ( $M_\infty = 0.1$ ), a nonstretched uniform grid with  $\lambda = 1.0$  was used. In the supercritical cases ( $M_\infty = 0.8$ ), grid stretching was necessary to keep the outer boundaries far enough from the airfoil to eliminate their effect on the results. Therefore, 8 percent stretching was used in the x-direction ( $F_x = 1.08$ ) and 12 percent stretching was used in the y-direction ( $F_y = 1.12$ ). This gave a range of aspect ratios on the fine grid of  $0.298 < \lambda < 3.426$ . This same stretched grid was also used in the compressible subcritical cases ( $M_\infty = 0.5$ ). All cases were run using a fine grid with 65 points in the x-direction and 33 points in the y-direction unless otherwise noted.

#### THE MULTIGRID METHOD

The basis for multigrid is the use of successively coarser grids to calculate corrections to a fine grid solution. Excellent developments of the multigrid technique are given in references 4, 12, and 13. For completeness, brief developments of multigrid are given below for both linear and nonlinear operators.

## Linear Equations

Consider the problem

$$L^h U^h = f^h \quad (10)$$

where  $L^h$  is a linear, finite-difference operator on a grid,  $G^h$ , with spacing  $h$ . The forcing function,  $f$ , is known and  $U^h$  is the solution to the problem on the grid with spacing  $h$ . If we take  $u^h$  as an approximation to  $U^h$  with an error of

$$v^h = U^h - u^h,$$

equation (10) can be written as

$$L^h(u^h + v^h) = f^h \quad (11)$$

Since  $L$  is a linear operator, this can be written as

$$L^h(u^h) + L^h(v^h) = f^h \quad (12)$$

If  $v^h$  is a smooth function, it can be represented on a coarser grid,  $G^{2h}$ , with spacing  $2h$ , twice the spacing between the points as the grid with spacing  $h$ . The grid  $G^{2h}$  is formed by deleting every other point in  $G^h$ . Therefore,  $G^{2h} \subseteq G^h$ . Points are eliminated from  $G^{2h}$  to form  $G^{4h}$  and so forth to form  $G^{8h}$ ,  $G^{16h}$ , etc. Each subsequent grid is a subset of the previous

grid. In this context, a function is considered smooth if it contains no high-frequency components which will cause aliasing of the function when it is transferred to a coarser grid.

It is possible to solve for an approximation to  $v^h$  on the coarser grid,  $G^{2h}$ , using equation (12) written for the coarser grid

$$L^{2h}(I_h^{2h}v^h) = I_h^{2h}(f^h - L^h u^h) \quad (13)$$

$I_h^{2h}$  is known as the restriction operator, and it simply transfers the values of a function from the fine grid to the coarse grid. Details of restriction operators are given later. For simplicity, define  $f^{2h} = I_h^{2h}(f^h - L^h u^h)$  as a forcing function on the coarse grid. Taking  $v^{2h} = I_h^{2h}v^h$ , equation (13) becomes

$$L^{2h}v^{2h} = f^{2h} \quad (14)$$

Since equation (14) is for a coarser grid than equation (12), the numerical solution for  $v^{2h}$  is much cheaper to obtain because fewer points are involved. Once  $v^{2h}$  is obtained, it is used to correct the fine grid iterative solution,  $u^h$  using

$$(u^h)_{\text{new}} = (u^h)_{\text{old}} + I_{2h}^h v^{2h} \quad (15)$$

The coarse grid to fine grid transfer operator,  $I_{2h}^h$ , in equation (15) is called the prolongation operator. This operator is discussed later.

Since the form of equation (14) is identical to the form of equation (10), it is obvious that a grid with spacing  $4h$  can be used to find corrections to the "solution" of the problem on the grid with spacing  $2h$ . Successively coarser grids may be used until a grid is reached which is so coarse that the "solution" is obtained very, very quickly and cheaply. (The limit is a grid so coarse that only a single unknown remains and is obtainable by direct solution.) The correction from the coarsest grid is then used to correct the correction on the next finer grid; and this is continued through successively finer levels until the finest grid is reached and the approximate solution is updated.

The usefulness of corrections obtained on a coarser grid is dependent on the smoothness of the fine grid error passed to the coarse grid. Hence, it is absolutely necessary that the high-frequency components of the error on the fine grid are reduced, if not completely eliminated. It is the responsibility of the smoother (usually a relaxation algorithm) to damp the high-frequency components of the error. The low-frequency components of the error are unimportant for all but the coarsest grid since the grids which are coarser than the current grid will see these low-frequency components as high frequencies and they will be damped there. If the high-frequency components are not damped, then the restriction operator will pass aliased information to

the coarser grid and the entire multigrid scheme will cease to converge. (An example of this is given in the Results section.) The choice of a smoother is critical to the use of multigrid. A subsequent section is devoted to a discussion of smoothers.

The cycle of work performed starting on the finest grid, then visiting the coarser grids, and then returning to the finest grid is called one multigrid cycle. The cycles are repeated until sufficient convergence is obtained on the finest grid. In the present study, a fixed cycle known as the V-cycle was used. In the V-cycle, a prescribed number of iterations are performed on increasingly coarser grids, starting on the fine grid and proceeding to the coarsest grid, and then proceeding on increasingly finer grids back to the finest grid.

#### Nonlinear Equations

The previous development of the multigrid scheme was for linear operators. Since the potential equation used to model transonic flow is nonlinear, it is necessary to use the Full Approximation Storage (FAS) multigrid scheme (ref. 9) which is applicable to both linear and nonlinear problems. A development of the FAS scheme is given below in equations (16)-(20).

If the  $L^h$  operator is taken to be nonlinear, the step taken between equations (11) and (12) in the previous development is no longer valid. Instead,  $L^{h_u^h}$  is subtracted from both sides of equation (11) to give:

$$L^h(u^h + v^h) - L^h(u^h) = f^h - L^h(u^h) \quad (16)$$

For the coarse grid, this becomes:

$$L^{2h}(I_h^{2h}u^h + v^{2h}) - L^{2h}(I_h^{2h}u^h) = I_h^{2h}(f^h - L^h u^h) \quad (17)$$

If the second term on the left-hand side is moved to the right-hand side, equation (17) can be written as:

$$L^{2h}(u^{2h}) = f^{2h} \quad (18)$$

where

$$f^{2h} = I_h^{2h}(f^h - L^h u^h) + L^{2h}(I_h^{2h}u^h) \quad (19)$$

Once values of  $u^{2h}$  are obtained, the fine grid iterative solution is updated using the following equation:

$$(u^h)_{\text{new}} = (u^h)_{\text{old}} + I_{2h}^h [u^{2h} - I_h^{2h}(u^h)_{\text{old}}] \quad (20)$$

Two points should be noted. First, the prolonged term on the right-hand side is a correction for the fine grid. Second, the operator used on the coarse grid (eq. (18)) has the same form as the fine grid operator, the grid spacing ( $h$  and  $2h$ ) being the only difference.



### Restriction Operators

As mentioned previously, the restriction operator,  $I_h^{2h}$ , transfers the values of some discrete function from a fine grid to a coarser grid. The simplest restriction operator is injection. Here, the value of the function at each point on the coarse grid is equated to the value of the function at the coincident point on the underlying fine grid. In equation form for a generic function,  $\psi$ , injection is:

$$\psi^{2h}(x_0, y_0) = \psi^h(x_0, y_0)$$

for all  $(x_0, y_0) \in (G^h \cap G^{2h})$ .

For some problems, it is necessary to use more complex restriction operators to transfer more "global" information from the fine grid. Normally, this is to eliminate small oscillations in the function on the fine grid. In the present work, a nine-point weighted average was used.

$$\begin{aligned} \psi^{2h}(x_0, y_0) = & 1/4 \psi^h(x_0, y_0) \\ & + 1/8 [\psi^h(x_0 + h, y_0) + \psi^h(x_0 - h, y_0) \\ & + \psi^h(x_0, y_0 + h) + \psi^h(x_0, y_0 - h)] \\ & + 1/16 [\psi^h(x_0 + h, y_0 + h) + \psi^h(x_0 + h, y_0 - h) \\ & + \psi^h(x_0 - h, y_0 + h) + \psi^h(x_0 - h, y_0 - h)] \end{aligned}$$

Note that  $\psi^{2h}(x_0, y_0)$  is now influenced by the values of the function at nine points on  $G^h$  rather than just  $(x_0, y_0)$ .

#### Prolongation Operators

The prolongation operator,  $I_{2h}^h$ , is the coarse- to fine-grid transfer operator. In the present study, direct transfer of the values of the function is used for fine grid points coincident with coarse grid points (Type A in fig. 2). Linear interpolation of coarse grid values is used for those fine grid points not coincident with coarse grid points (Types B, C, and D in fig. 2). It is obvious that four steps are necessary to complete the prolongation.

#### SMOOTHING ALGORITHMS

As mentioned in the previous section, the smoother is used in a multigrid scheme to eliminate the high-frequency components of the error in the solution. The proper choice of a smoother is critical to the success of multigrid. Fortunately, it is usually possible to predict the performance of a smoother before it is incorporated in a multigrid context.

#### Algorithms

Six algorithms were considered as multigrid smoothers in the present work - vertical, horizontal, and alternating-direction SLOR and ZEBRA I. The three SLOR algorithms have been studied by other researchers and were included in the present work for

comparison with the ZEBRA I algorithms which are the focus of the current study. A description of each of the six algorithms is given below.

VLOR - Vertical line overrelaxation.- When updating the point  $(i,j)$ , VLOR uses updated values at three adjacent points;  $(i-1,j)$ ,  $(i,j+1)$ , and  $(i,j-1)$ , if the solution proceeds in the increasing  $i$  direction. (See fig. 3.) The values at  $(i,j)$  are updated when the previous  $i = \text{constant}$  vertical line is updated. The values at  $(i,j+1)$  and  $(i,j-1)$  are updated at the same time as the  $(i,j)$  point. The implicit set of tridiagonal equations generated by this scheme is readily solved using the Thomas algorithm. Since the solution of the implicit lines is order dependent and the Thomas algorithm is recursive, VLOR is not fully vectorizable.

HLOR - Horizontal line overrelaxation.- The HLOR scheme is basically the same as the VLOR scheme except that the implicit lines are in the horizontal direction. When updating the point  $(i,j)$ , HLOR uses updated values at three adjacent points;  $(i,j-1)$ ,  $(i+1,j)$ , and  $(i-1,j)$ , if the solution proceeds in the increasing  $j$  direction. (See fig. 4.) The values at  $(i,j-1)$  were updated when the previous  $j = \text{constant}$ , horizontal line was updated. The values at  $(i+1,j)$  and  $(i-1,j)$  are updated at the same time as the  $(i,j)$  point. The implicit set of tridiagonal equations generated by this scheme is readily solved by the Thomas algorithm and, as with VLOR, HLOR is not fully vectorizable.

ADLOR - Alternating direction line overrelaxation.- ADLOR is just the alternating application of VLOR and HLOR sweeps. Either VLOR or HLOR is used to update all of the points in the field and then the other is used to further update all of the points in the field. The ADLOR algorithm tends to be less sensitive to grid stretching.

VZEB1 - Vertical line ZEBRA I.- VZEB1 is a two-color, implicit, vertical line scheme. For the first color, say  $i = \text{even}$  points, only the "new" values at the points above and below  $((i,j+1)$  and  $(i,j-1))$  the point to be updated  $((i,j))$  are used. (See fig. 5.) This gives a set of tridiagonal equations which can be solved using the Thomas algorithm. Once the points in the first color are updated, their new values are used to update the points in the second color, odd values of  $i$  in this example. Hence, for the second color, updated values of the points  $(i-1,j)$ ,  $(i+1,j)$ ,  $(i,j-1)$ , and  $(i,j+1)$  are used to update the values at the point  $(i,j)$ . (See fig. 5.) The values at the points above and below the point at  $(i,j)$  and the values at  $(i,j)$  are updated simultaneously. This results in another set of tridiagonal equations which is solved using the Thomas algorithm. Since the implicit lines of a given color are decoupled from one another, all the lines of a given color may be solved simultaneously. Recall that the solutions of the tridiagonal lines in VLOR were order dependent so that the lines had to be solved one at a time. Hence, they were not vectorizable. The tridiagonal lines of a given color in VZEB1

are independent and can be solved simultaneously. This means that vector instructions can be used. Specifically, vector instructions are used to calculate the tridiagonal coefficients for all the implicit lines of a given color at each value of  $j$ . After all the coefficients are calculated, the back-substitutions are performed for all the implicit lines, one value of  $j$  at a time. Further information about the vectorization of the ZEBRA algorithms is contained in references 1 and 2.

HZEB1 - Horizontal line ZEBRA I.- HZEB1 is identical to VZEB1 except that the implicit lines are in the horizontal direction.

ADZEB1 - Alternating direction line ZEBRA I.- ADZEB1 is just the alternate application of VZEB1 and HZEB1 sweeps. One of the two is used to update all of the points in the field and then the other is used to further update the points in the field.

#### Smoothing Analysis

The von Neumann stability analysis was developed by John von Neumann at Los Alamos around 1944. The method was circulated privately for several years (ref. 14). A short description of the approach was first published in 1947 by Crank and Nicolson (ref. 15) and the first thorough explanation was given by O'Brien, Hyman, and Kaplan (ref. 16). An excellent textbook explanation of the method was presented by Roache (ref. 17) in 1972.

In the von Neumann analysis, the approximate solution to a model equation generated by the algorithm of interest is expanded using a finite Fourier series. The exact solution is then subtracted - leaving the error. The decay or amplification of each frequency of the error is considered to determine stability or instability (ref. 17). The decay of the error is expressed by the von Neumann damping or amplification factor,  $g$ , as a function of frequency. The maximum value of  $g$  over the high-frequency range of the error frequencies is defined as the smoothing factor (ref. 9),  $\mu$ .

$$\mu \equiv \max \{g(\theta_x, \theta_y)\}, \quad \pi/2 < |\theta_x, \theta_y| < \pi \quad (21)$$

where  $\theta_x$  and  $\theta_y$  are the phase angles associated with the  $x$ - and  $y$ -coordinate directions, respectively. The full range of each  $\theta$  in a general solution is  $-\pi < \theta < \pi$ . The range from  $-\pi$  to  $\pi$  may be divided into two groups, the high frequencies from  $-\pi$  to  $-\pi/2$  and  $\pi/2$  to  $\pi$  and the low frequencies from  $-\pi/2$  to  $\pi/2$ . The high-frequency components of the errors must be eliminated on the fine grid because they cannot be resolved on a coarser grid. The low-frequency components of the error are reduced on a coarser grid where one-half of them become "high"-frequency errors due to the change in grid spacing. Coarser and coarser grids are visited to eliminate lower frequency components of the error as one-half of the low-frequency range of a given grid becomes the high-frequency range

on the next coarser grid. Therefore, the smoother only needs to eliminate the error frequencies from  $\pi/2 < |\theta| < \pi$  and this is the range examined for the smoothing factor. (One additional restriction is that the error not grow in the low-frequency range, or  $g < 1$  for  $0 < |\theta| < \pi/2$ .)

A von Neumann analysis was performed for each of the six different algorithms considered in this study. In these analyses, generalizations for varying grid aspect ratio and Mach number were taken into account. Grid stretching was not directly modeled in the von Neumann analyses, but one of its effects was modeled by the inclusion of grid aspect ratio,  $\lambda$ .

From the description of the conservative transonic potential flow problem given previously, it is clear that Mach number effects must be included in the von Neumann analyses of the smoothers. Recall that the density appears as a coefficient in the conservative transonic potential equation. Variations in density due to compressibility effects cause nonlinearities. The type of the equation also changes with Mach number, elliptic for  $M < 1$  and hyperbolic for  $M > 1$ .

The conservative full-potential equation with artificial density is difficult to analyze using the von Neumann analysis. Considerably simpler is the analysis of the nonconservative small-disturbance potential equation  $(1 - M^2) \phi_{xx} + \phi_{yy} = 0$ . Results of the von Neumann analyses using the simpler equation are expected to be consistent with the full-potential equation due to the validity of the substitution as shown in Appendix A.

Hence, the performance of the smoothers operating on the transonic small-disturbance equation was analyzed, rather than the conservative full-potential equation. Two separate analyses were performed for each of the algorithms: One for  $M < 1$  with central differencing and one for  $M > 1$  with upwind differencing. Over- or underrelaxation was accounted for in each subsonic analysis with the inclusion of the relaxation parameter,  $\omega$ . Additional damping in the form of  $\beta \phi_{xt}$ , where  $\beta$  is a free parameter, was included in the supersonic analyses.

The von Neumann method for VLOR is straightforward and is included here to demonstrate the analysis for a simpler case before the more complicated details of VZEB1 are given. The analyses for the ZEBRA algorithms are more complex since they are two-color schemes. The development for HLOR roughly follows the analysis of VLOR and so is not included. The analysis of HZEB1 is given in Appendix B. The six algorithms analyzed and their amplification factors are summarized in Table 1.

#### A von Neumann Analysis of VLOR

The following analysis is for the nonconservative small-disturbance potential equation using central differencing in subsonic regions and upwind differencing in supersonic flow regions. Since the finite differencing operator is different for subcritical (central differencing) and supercritical (upwind differencing) flow regions, it is necessary to analyze the two



flow conditions separately. The subcritical case is analyzed first.

Subsonic Flow  $M < 1$

The small-disturbance equation in operator form is:

$$\left[ (1 - M^2) \delta_{xx} + \lambda^2 \delta_{yy} \right] \phi_{ij} = 0 \quad \text{for } M < 1. \quad (22)$$

where

$M$  is the local Mach number

$\delta_{xx}$  is an undivided, central difference operator in the  $x$ -direction

$\delta_{yy}$  is an undivided, central difference operator in the  $y$ -direction

$\lambda$  is the grid aspect ratio,  $\Delta x / \Delta y$

and

$\phi_{ij}$  is the potential at the point  $i, j$ .

VLOR can be written as

$$(1 - M^2) \left[ E_x^- \phi_{ij}^{n+1} + (-2I) \tilde{\phi}_{ij}^{n+1} + E_x^+ \phi_{ij}^n \right] + \lambda^2 (E_y^+ - 2I + E_y^-) \tilde{\phi}_{ij}^{n+1} = 0 \quad (23)$$

where

$E$  is a displacement operator in the  $x$  ( $E_x$ ) or  $y$  ( $E_y$ ) direction in the positive ( $E^+$ ) or negative ( $E^-$ ) direction.

That is,

$$E_x^\pm \phi_{ij} \equiv \phi_{i\pm 1, j}$$

and

$$E_y^\pm \phi_{ij} \equiv \phi_{i, j\pm 1}$$

$\phi_{ij}^n$  is a value of the potential at the  $n^{\text{th}}$  time level previous to the update, and  $\tilde{\phi}_{ij}^{n+1}$  is an updated value of the potential which satisfies equation (23).

Fourier coefficients are substituted for the error,  $e$ , which also satisfies the homogeneous difference equation given by equation (23). Therefore,

$$e_{ij}^n = g^n e^{\sqrt{-1}(\theta_x + \theta_y)} \quad (24)$$

Substitution of the Fourier components (eq. (24)) into equation (23) for  $\phi_{ij}^n$  yields:

$$g \equiv \frac{g^{n+1}}{g^n} = \frac{(-1) \left[ (1 - M^2) \left( e^{\sqrt{-1} \theta_x} - 2 \frac{\omega-1}{\omega} \right) + \lambda^2 \frac{\omega-1}{\omega} (2 \cos \theta_y - 2) \right]}{(1 - M^2) \left( e^{-\sqrt{-1} \theta_x} - 2/\omega \right) + \lambda^2/\omega (2 \cos \theta_y - 2)} \quad (25)$$

where  $g$  is the amplification factor;  $\theta_x$  and  $\theta_y$  are the phase angles associated with the  $x$ - and  $y$ -coordinate directions, respectively; and  $\omega$  is the overrelaxation factor such that

$$(\phi_{ij}^{n+1} - \phi_{ij}^n) = \omega (\tilde{\phi}_{ij}^{n+1} - \phi_{ij}^n).$$

#### Supersonic Flow $M > 1$

For a Mach number greater than or equal to 1.0, the small-disturbance potential equation is:

$$\left[ (1 - M^2) \delta_{xx} + \lambda^2 \delta_{yy} - \beta \delta_{xt} \right] \phi_{ij} = 0 \quad \text{for } M > 1 \quad (26)$$

where

$M$ ,  $\lambda$ ,  $\delta_{yy}$ , and  $\phi_{ij}$  are as defined below equation (22)

$\delta_{xx}^{\dagger}$  is an undivided upwind difference operator in the x-direction

$\beta$  is a free parameter

and

$\delta_{xt}$  is an undivided second difference operator in the x- and time-directions which is added for stability. The difference expression is:

$$\delta_{xt} \phi_{ij} = (\phi_{ij}^{n+1} - \phi_{ij}^n) - (\phi_{i-1,j}^{n+1} - \phi_{i-1,j}^n)$$

Using the operators defined with equation (23), equation (26) may be written as:

$$(1 - M^2) \left[ (E_x^-)^2 - 2E_x^- + I \right] \phi_{ij}^{n+1} + \lambda^2 (E_y^- - 2I + E_y^+) \phi_{ij}^{n+1} + \beta \left[ (I - E_x^-) \phi_{ij}^{n+1} - (I - E_x^-) \phi_{ij}^n \right] = 0 \quad (27)$$

Substitution of the Fourier coefficients and simplification yields:

$$\begin{aligned}
g = & \beta [1 - (\cos \theta_x - \sqrt{-1} \sin \theta_x)] / \{ (1 - M^2) [(\cos \theta_x - \sqrt{-1} \sin \theta_x)^2 \\
& - 2 (\cos \theta_x - \sqrt{-1} \sin \theta_x) + 1] + \lambda^2 (2 \cos \theta_y - 2) \\
& + \beta [1 - (\cos \theta_x - \sqrt{-1} \sin \theta_x)] \} \quad (28)
\end{aligned}$$

From equations (27) and (28) it is clear that VLOR is a marching scheme in purely supersonic flow and, if  $\beta = 0$ ,  $g = 0$  for all frequencies.

For a given method, Mach number (M), and values of  $\beta$ ,  $\omega$ , and  $\lambda$ ; values of  $g$  may be found as a function of  $\theta_x$  and  $\theta_y$  where  $\theta_x$  and  $\theta_y$  vary from  $-180^\circ$  to  $180^\circ$ . It is useful to look at contour plots of  $g$  for varying  $\theta_x$  and  $\theta_y$  as shown in figure 6. The frequency range for each of the  $\theta$ 's can be divided into two segments - the high-frequency range ( $-180^\circ$  to  $-90^\circ$  and  $90^\circ$  to  $180^\circ$ ) and the low-frequency range ( $-90^\circ$  to  $90^\circ$ ). Note that the maximum values of  $g$  in figure 6 are in the low-frequency ranges of  $\theta_x$  and  $\theta_y$  ( $g_{\max} = 0.97$ ).

With multigrid, only the high-frequency region is important and the low-frequency region may be ignored as reflected in figure 7. The maximum value of  $g$  in the high-frequency range is the smoothing factor,  $\mu$ . Notice that the elimination of the low-frequency region significantly lowers the maximum value of  $g$  obtainable and so  $\mu$  is significantly less than  $g_{\max}$  for this case ( $\mu = 0.44$  versus  $g_{\max} = 0.97$ ).

A von Neumann Analysis of VZEB1

Subsonic Flow  $M < 1$

Using the previously defined displacement operators, the VZEB1 algorithm applied to equation (22) yields:

$$\begin{aligned}
 (1 - M^2) & \left[ E_x^- \phi_{ij}^n - 2I \phi_{ij}^{\sim n+1/2} + E_x^+ \phi_{ij}^n \right] \\
 & + \lambda^2 \left[ E_y^- - 2I + E_y^+ \right] \phi_{ij}^{\sim n+1/2} \\
 & - \beta \left[ \phi_{ij}^{\sim n+1/2} - \phi_{ij}^{n-1/2} - E_x^- (\phi_{ij}^n - \phi_{ij}^{n-1}) \right] = 0
 \end{aligned} \tag{29}$$

where:

$\phi_{ij}^n$ : a new value of the potential at the "other" color at the  $n^{\text{th}}$  time level. The two colors are considered to be one-half time step apart in the analysis.

$\phi_{ij}^{n-1/2}$ : an old value of the potential on the color being updated.

$\phi_{ij}^{n+1/2}$ : a new value of the potential on the color being updated (overrelaxed).

$\phi_{ij}^{\sim n+1/2}$ : a value of the potential on the color being updated which makes the residual zero.

The three time levels in equation (29) are related by

$$\tilde{\phi}_{ij}^{n+1/2} = 1/\omega \phi_{ij}^{n+1/2} + \frac{\omega-1}{\omega} \phi_{ij}^{n-1/2} \quad (30)$$

where  $\omega$  is the relaxation parameter. Substitution of the Fourier coefficients for the potential and simplification gives a quadratic equation for  $\sqrt{g}$ .

$$\begin{aligned} & \left[ (1 - M^2)(-2)(1/\omega) + \lambda^2 (2 \cos \theta_y - 2)(1/\omega) \right] (\sqrt{g})^2 \\ & + \left[ (1 - M^2)(2 \cos \theta_x) \right] \sqrt{g} \\ & + \left[ (1 - M^2)(-2) \left( \frac{\omega-1}{\omega} \right) + \lambda^2 (2 \cos \theta_y - 2) \left( \frac{\omega-1}{\omega} \right) \right] = 0 \quad (31) \end{aligned}$$

This may be solved using the quadratic equation to find the two roots. The root which represents  $g$  in the physical problem cannot be determined. The purpose of the von Neumann analysis is to predict the multigrid smoothing rate  $\mu$ , for VZEB1. Since the value of  $\mu$  is governed by the maximum value of  $g$ , it is prudent to pick the root of  $g$  which has the maximum absolute value.

#### Supersonic Flow $M > 1$

In operator form, VZEB1 applied to equation (26) can be written as:

$$\begin{aligned}
(1 - M^2) & \left[ (E_x^-)^2 \phi_{ij}^{n-1/2} - 2 (E_x^-) \phi_{ij}^n + I \phi_{ij}^{n+1/2} \right] \\
& + \lambda^2 \left[ E_y^- + 2I + E_y^+ \right] \phi_{ij}^{n+1/2} \\
& - \beta \left[ \phi_{ij}^{n+1/2} - \phi_{ij}^{n-1/2} - (E_x^-) (\phi_{ij}^n - \phi_{ij}^{n-1}) \right] = 0 \quad (32)
\end{aligned}$$

There are some interesting things to note here. First, an "old" value is used at  $i-2$  to maintain the independence of the vertical implicit lines. Second, note that the terms representing  $\delta_{xt}$  are split between the two time steps. This equation may be rewritten as a cubic equation in  $\sqrt{g}$ .

$$a \sqrt{g}^3 + b (\sqrt{g})^2 + c \sqrt{g} + d = 0 \quad (33)$$

where

$$a = (1 - M^2) + \lambda^2 (2 \cos \theta_y - 2) - \beta$$

$$b = [(1 - M^2)(-2) + \beta] (\cos \theta_x - \sqrt{-1} \sin \theta_x)$$

$$c = (1 - M^2) (\cos \theta_x - \sqrt{-1} \sin \theta_x)^2 + \beta$$

$$d = -\beta (\cos \theta_x - \sqrt{-1} \sin \theta_x)$$



The maximum magnitude of the three roots is taken as the magnitude of  $g$ .

In figure 8(a), a contour plot of  $g$  for VZEB1 at the conditions of  $M_\infty = 0.1$ ,  $\beta = 0$ ,  $\lambda = 1.0$ , and  $\omega = 1.0$  is given. These parameter values are the same as figures 6 and 7 for VLOR. Note that there are three areas of  $g \approx 1.0$  in this contour plot. There is one at  $(\theta_x, \theta_y)$  equal to  $(0^\circ, 0^\circ)$  as with VLOR and there are two at  $(-180^\circ, 0^\circ)$  and  $(+180^\circ, 0^\circ)$ . The latter two areas of  $g \approx 1$  are not in the low-frequency regions and so are not discarded when  $\mu$  is examined, figure 8(b). Fortunately, since the values of  $\theta$  equal to  $0^\circ$  and  $\pm 180^\circ$  are not obtainable on a discrete grid, only a limited effect of this region should be felt in a multigrid application. However, since  $\mu$  for VZEB1 is 0.97 and  $\mu$  for VLOR is 0.44 for the given conditions, it is anticipated that the VLOR may give a better convergence rate as a multigrid smoother.

It is useful to consider only  $\mu$  rather than the full range of values of  $g$ . In figure 9,  $\mu$  for VZEB1 is plotted versus Mach number for three values of aspect ratio.  $\lambda = 1.0$  is the uniform grid aspect ratio used for the entire grid for the incompressible test case mentioned below in the Results section.  $\lambda = 3.426$  and  $0.0298$  are the upper and lower limits of aspect ratio on the fine grid of the stretched grid in the compressible subsonic and supersonic test cases also discussed in the Results section. It is clear from figure 9 that the performance of VZEB1 is strongly affected by aspect ratio over

the entire Mach number range presented. The predicted performance is poor for  $\lambda < 1$  but better for  $\lambda > 1$ , as expected for VZEB1.

A von Neumann analysis for HZEB1 is given in the Appendix B. Contour plots of  $g$  for HZEB1 are quite similar to those of VZEB1 except that they are rotated  $90^\circ$ . This is reflected in figure 10 where  $\mu$  versus  $M$  is plotted for various values of aspect ratio. Note that HZEB1 has good performance for small aspect ratio and poor performance at high aspect ratio, the opposite of VZEB1.

The von Neumann analysis of ADZEB1 is just a combination of the analyses of VZEB1 and HZEB1. At each frequency, the damping rate for ADZEB1 is the square root of the product of the individual damping rates of VZEB1 and HZEB1. This reduces the sensitivity of ADZEB1 to aspect ratio effects since frequencies that either VZEB1 or HZEB1 has problems with are compensated for by good performance by the opposite smoother (HZEB1 or VZEB1, respectively; i.e., for small-aspect-ratio grid cell, HZEB1 does well and for large-aspect-ratio grid cells, VZEB1 does well. The improved performance of ADZEB1 for varying aspect ratio is reflected in figure 11.

## RESULTS

A computer code was written to test the performance of the ZEBRA algorithms as multigrid smoothers. The transonic full-potential equation was modeled using central differencing with the artificial density method used to provide upwind bias in supersonic regions. Small-disturbance boundary conditions were used to model the presence of an airfoil.

The flow over a symmetric, 12-percent-thick parabolic arc airfoil was calculated. Free-stream Mach numbers of 0.1, 0.5, and 0.8 were considered. Stretched grids (8 percent in the x-direction and 12 percent in the y-direction) were used at all three Mach numbers. A uniform grid (no stretching,  $\Delta x = \Delta y$ ) was also used at  $M_\infty = 0.1$ . Only nonlifting cases were considered.

The performance of VZEB1 was examined at the conditions of  $M_\infty = 0.1$  on a uniform unstretched grid. Simple injection of the potential and residual was used in a 5-grid multigrid scheme. The solution did not converge. To understand why this occurred, surface plots of the potential, residual, and forcing function for the various grids were made.

A surface plot of the potential on the fine grid (grid 1) just before injection to grid 2 is shown in figure 12. The coordinate system for the grid is as shown in the figure and will apply to all other surface plots. Note that the positive direction for the function shown is down. Note also that the function is reasonably smooth.

A surface plot of the residual on grid 1 is shown in figure 13. Note the jagged nature of the residual function. If the x-y plane is rotated up and toward the viewer, so that the line of sight is directly down the y-axis (see fig. 14), it can be seen that every other "x = constant" line has a residual of zero. This is to be expected with the decoupled-line, two-color nature of the ZEBRA I schemes.

The high-frequency oscillations in the residual on the first grid caused aliasing of the forcing function on the second grid. (The forcing function,  $f$ , is the difference of the fine and coarse grid operators; see eq. (19).) With no residual weighting (see Restriction Operator section above), the forcing function on the second grid had the shape shown in figure 15. With residual weighting, the forcing function on the second grid had the shape shown in figure 16.

The effect of residual weighting on the convergence history is shown in figure 17 where the logarithm of the ratio of the fine grid residual to the initial fine grid residual is plotted versus work. One unit of work is defined as the number of operations required to perform one fine grid ( $G^h$ ) iteration. One iteration on grid  $G^{2h}$  requires 0.25 work units for a two-dimensional problem, etc.

From the above discussion, it is obvious that residual weighting must be used with VZEB1 to eliminate aliasing of the coarse grid forcing function and allow convergence. (Further research showed that, for these same reasons, residual weighting is necessary for HZEB1 also.)

Weighted averaging of the potential function was also examined. It was found that this was not necessary since the potential is relatively smooth on the fine grid (fig. 12) and the injected values of the potential on the coarse grid are not aliased without the weighting. The "shape" of the coarse grid potential is shown in the surface plot of figure 18. This same shape is obtained independent of potential weighting. Injection was used for the potential in all subsequent results discussed.

With residual weighting, VZEB1 was greatly accelerated by the use of multigrid. In the most benign case considered in this study ( $M_\infty = 0.1$ , uniform grid), the acceleration of VZEB1 by multigrid is illustrated in figure 19 using 2, 3, 4, and 5 grids. All runs were conducted with an overrelaxation factor of 1.0. Note that 4- and 5-grid multigrid give identical convergence and are only slightly faster than 3-grid multigrid for these conditions. At a Mach number of 0.1, the flow is completely subsonic and effectively incompressible. The potential equation governing the flow is elliptic; and, since the aspect ratio is unity, it effectively reduces to Laplace's equation. Thus, the excellent speedup obtained using multigrid was expected even though it was better than the von Neumann analysis suggests. Evidently, the frequencies giving the smoothing factor ( $\mu$ ) as plotted in figure 9 were not obtained. Figure 8(b) shows that the maximum values of  $g$  for VZEB1 at  $M = 0.1$  and  $\lambda = 1.0$  are concentrated at the two points of high frequency in  $\theta_x$  and the low frequency in  $\theta_y$ . If these frequencies do not appear in the error of the solution,

then they do not need to be damped and the effective  $\mu$  is reduced. This evidently occurred at  $M_\infty = 0.1$  and  $\lambda = 1.0$  for VZEB1. From this example, it is clear that the results from a von Neumann analysis must be carefully examined to determine validity. Predictions of  $\mu$  can be biased by frequencies not found in the coded problem.

The performance of VZEB1 multigrid is dependent on the grid aspect ratio. This dependence is reflected in figure 20. Starting with the most benign conditions,  $M_\infty = 0.1$  and an unstretched grid, the performance of VZEB1 multigrid is excellent. If the Mach number is held the same but grid stretching is included (8 percent in the x-direction and 12 percent in the y-direction) the performance is significantly degraded. (Compare  $M_\infty = 0.1$  curves.) Note that not only are the absolute levels of the residual higher but also the asymptotic rate of convergence is not nearly as good with the grid stretching. Recall that the von Neumann analysis did predict a sensitivity to aspect ratio.

If the Mach number is increased slightly to  $M_\infty = 0.5$  so that compressibility effects are in the solution, the performance is slightly worse than the incompressible case on the same stretched grid. Most of the difference is in the absolute levels of the residual and not in the asymptotic convergence rate. A further increase in Mach number to  $M_\infty = 0.8$  so that the flow field becomes transonic further degrades the performance of VZEB1 multigrid but again mostly in absolute levels of residual and not asymptotic rate of convergence. (Compare stretched grid curves

in fig. 20.) From this discussion it is clear that the performance of VZEB1 multigrid is affected more strongly by grid aspect ratio than by compressibility (nonlinearity) or supercritical flow regions (equation type changes). This behavior was supported by the von Neumann analysis as summarized in figure 9.

The acceleration afforded VZEB1 by various numbers of grids in the multigrid scheme at  $M_\infty = 0.8$  is shown in figure 21. The speedup at  $M_\infty = 0.8$  is not as dramatic as at  $M_\infty = 0.1$ . Note that 4- and 5-grid multigrid give nearly identical performance and are only slightly better than the 3-grid multigrid. The asymptotic rate for 5 grid is only slightly better than for 1 grid, but the absolute level of residual is significantly less.

Since many others have used VLOR as a multigrid smoother, it is worthwhile to compare the performance of VZEB1 and VLOR in multigrid. At  $M_\infty = 0.8$  on a stretched grid, their performance is compared in figure 22. Note that their rate of convergence is nearly identical, however the VZEB1 can be more fully vectorized and so should give better overall performance (less computer time to obtain a converged solution). Similar comparison plots for HZEB1 multigrid are shown in figures 23-26. The results follow those for VZEB1. It should be noted that for the test problem in the present work, HZEB1 gives better performance than VZEB1. This is not a general result, but is obtained here only because of the specific grid used and the difference in percentage of total points included in the update implicit lines of the two algorithms.

In the von Neumann analysis of ADZEB1, it was found that the alternating direction scheme should give better performance on varying aspect ratio grids. (See fig. 11.) The prediction is confirmed in figure 27 where the effects of  $M_\infty$  and aspect ratio on ADZEB1 multigrid are examined. Note that the incompressible (unstretched and stretched grid) and the compressible cases all give roughly the same convergence rates. With the effects of grid stretching reduced, the effect of supersonic regions is seen to adversely affect the convergence rate. The performance of ADZEB1 in the transonic flow calculation is quite good and is compared to ADLOR in figure 28. Unfortunately, because of the alternating direction nature of ADZEB1, vectorization is not as straightforward as with VZEB1 and HZEB1.



## CONCLUSIONS

Three vectorizable multigrid smoothers were examined; vertical, horizontal, and alternating direction ZEBRA. The smoothing performance of each was predicted using von Neumann stability analyses. The effects of Mach number, grid aspect ratio and damping factor were included. As expected, the analyses predicted that VZEB1 performs best on grids with  $\lambda > 1$  and HZEB1 performs best on grids with  $\lambda < 1$ . The analysis of ADZEB1 predicted that it is less sensitive to grid aspect ratio than VZEB1 or HZEB1. All three ZEBRA methods were predicted to have poorer performance at supercritical Mach numbers than at subcritical Mach numbers.

The actual performance of each of the ZEBRA algorithms was then assessed by incorporation into a two-dimensional full-potential code. The performance of the vectorizable algorithms was compared to three well-known SLOR algorithms. In each comparison (VZEB1 to VLOR, HZEB1 to HLOR, and ADZEB1 to ADLOR), the ZEBRA scheme had a convergence rate comparable with the respective SLOR scheme. These comparisons indicated a reasonable level of success of the multigrid acceleration of each ZEBRA scheme. Using very powerful smoothers, it is possible to have faster multigrid convergence rates than were obtained in the present study. However, the purpose of the present work was to study vectorizable ZEBRA multigrid algorithms, so slightly less than optimal convergence rates were acceptable as a compromise to obtain vectorizable code.

It is unfortunate that the von Neumann analyses did not predict quantitatively the performance of the ZEBRA schemes. In a discrete problem, only a finite number of frequencies are present in the error. These frequencies are dictated, in part, by the grid. A von Neumann analysis models a continuum problem and so considers all frequencies of the error. The value of  $\mu$  obtained from the analysis can be biased if it is predicted based on frequencies which are not present in the discrete problem, particularly when  $\theta_x$  or  $\theta_y$  equals or is close to 0 or  $\pm\pi$ .

Finally, it should be noted that only a nonlifting airfoil was considered in the present study. Convergence rates of calculations for lifting airfoils using the current methods would probably not be as fast.

APPENDIX A  
SIMPLIFICATION OF CONSERVATIVE FULL-POTENTIAL  
EQUATION WITH ARTIFICIAL DENSITY

Consider the one-dimensional form of the full-potential equation using the artificial density method:

$$(\bar{\rho}u)_x = 0 \tag{A1}$$

where

$$\bar{\rho} = \rho - \tau \Delta x \rho_x \tag{A2}$$

$$\rho = (M_\infty^2 a^2)^{1/(\gamma - 1)} \tag{A3}$$

and

$$a^2 = 1/M_\infty^2 + (\gamma - 1)(1 - u^2)/2 \tag{A4}$$

(Note that equations (A1)-(A4) are the same as equations (1)-(4) in the main paper with  $v = 0$ .) The finite difference form of equation (A1) is:

$$\frac{\bar{\rho}_{i+1/2} u_{i+1/2} - \bar{\rho}_{i-1/2} u_{i-1/2}}{\Delta x} = R_i(\phi) \tag{A5}$$

where

$$\begin{aligned}\bar{\rho}_{i+1/2} &= \rho_{i+1/2} - \tau_i (\rho_{i+1/2} - \rho_{i-1/2}) \\ &= (1 - \tau_i) \rho_{i+1/2} + \tau_i \rho_{i-1/2}\end{aligned}\tag{A6}$$

and  $\tau$  is the switching factor as described in equation (5). The residual,  $R_i^n(\phi)$ , can be expanded about the exact solution  $\hat{\phi}$ , such that  $R_i(\hat{\phi}) = 0$ ,

$$R_i^n(\phi) = R_i(\hat{\phi}) + \sum_{k=i-2}^{i+1} \frac{\partial R_i^n}{\partial \phi_k} (\phi_k^n - \hat{\phi}_k)\tag{A7}$$

$$= 0 + \sum_{k=i-2}^{i+1} \frac{\partial R_i^n}{\partial \phi_k} e_k^n$$

where  $e_k^n$  is the error in the solution at the  $n^{\text{th}}$  time level. Substituting equation (A6) into equation (A5) and taking the partial derivative of  $\Delta x^2 R_i$ , we obtain:

$$\begin{aligned}\delta(\Delta x^2 R_i) &= (1 - \tau) u \Delta x \delta \rho_{i+1/2} + \tau \Delta x u \delta \rho_{i-1/2} + \rho \Delta x \delta u_{i+1/2} \\ &- (1 - \tau) u \Delta x \delta \rho_{i-1/2} - \tau \Delta x u \delta \rho_{i-3/2} - \rho \Delta x \delta u_{i-1/2} \\ &- (\rho_{i+1/2} - \rho_{i-1/2}) \Delta x \delta (\tau_i u_{i+1/2}) \\ &+ (\rho_{i-1/2} - \rho_{i-3/2}) \Delta x \delta (\tau_{i-1} u_{i-1/2})\end{aligned}\tag{A8}$$

Since  $(\rho_{i+1/2} - \rho_{i-1/2}) \Delta x$  and  $(\rho_{i-1/2} - \rho_{i-3/2}) \Delta x$  are  $O(\Delta x^2)$ , the last two terms in equation (A8) may be dropped because they are higher order terms than the first six terms. The values of the derivatives in equation (A8) are:

$$\begin{aligned}
 u \Delta x \frac{\partial \rho_{i+1/2}}{\partial \phi_{i+1}} &= \rho M^2 & u \Delta x \frac{\partial \rho_{i+1/2}}{\partial \phi_i} &= \rho M^2 \\
 u \Delta x \frac{\partial \rho_{i-1/2}}{\partial \phi_i} &= -\rho M^2 & u \Delta x \frac{\partial \rho_{i-1/2}}{\partial \phi_{i-1}} &= \rho M^2 \\
 u \Delta x \frac{\partial \rho_{i-3/2}}{\partial \phi_{i-1}} &= -\rho M^2 & u \Delta x \frac{\partial \rho_{i-3/2}}{\partial \phi_{i-2}} &= \rho M^2 & (A9) \\
 \rho \Delta x \frac{\partial u_{i+1/2}}{\partial \phi_{i+1}} &= \rho & \rho \Delta x \frac{\partial u_{i+1/2}}{\partial \phi_i} &= -\rho \\
 \rho \Delta x \frac{\partial u_{i-1/2}}{\partial \phi_i} &= \rho & \rho \Delta x \frac{\partial u_{i-1/2}}{\partial \phi_{i-1}} &= -\rho
 \end{aligned}$$

Substituting equations (A9) into equations (A7) and (A8) and simplifying gives:

$$\begin{aligned}
 \frac{\Delta x^2 R_1^n}{\rho} &= [1 - (1 - \tau) M^2] e_{i+1}^n [(2 - 3\tau) M^2 - 2] e_i^n \\
 &\quad - [(1 - 3\tau) M^2 - 1] e_{i-1}^n - \tau M^2 e_{i-2}^n & (A10)
 \end{aligned}$$

For  $M < 1$ , equation (4) gives  $\tau = 0$ . Substitution of this into equation (A10) gives:

$$\Delta x^2 R_i^n = \rho (1 - M^2) [e_{i+1}^n - 2e_i^n + e_{i-1}^n] \quad (\text{A11})$$

For  $M > 1$ , equation (4) gives  $\tau = 1 - 1/M^2$ . Substitution of this into equation (A10) gives:

$$\Delta x^2 R_i^n = \rho (1 - M^2) [e_i^n - 2e_{i-1}^n + e_{i-2}^n] \quad (\text{A12})$$

From equations (A11) and (A12) we see that the conservative full-potential equation with artificial density may be simplified to the nonconservative small-disturbance potential equation with upwind differencing in the supercritical regions for the purposes of analysis.\*

---

\*The development showing this simplification is based on work originally done by Jerry C. South, Jr., and is contained in his personal notes.

APPENDIX B  
ANALYSIS OF HZEB1

Subsonic Flow -  $M < 1$

The small-disturbance equation in operator form is

$$[(1 - M^2) \delta_{xx} + \lambda^2 \delta_{yy}] \phi_{ij} = 0 \quad (B1)$$

where each term is defined after equation (22) in the main body of this text. Using the displacement operators defined in the text, the HZEB1 algorithm applied to equation (B1) yields:

$$(1 - M^2) [E_x^- - 2I + E_x^+] \tilde{\phi}_{ij}^{n+1/2} + \lambda^2 (-2I) \tilde{\phi}_{ij}^{n+1/2} \\ + \lambda^2 (E_y^- + E_y^+) \phi_{ij}^n = 0 \quad (B2)$$

The time levels of the potential are defined with equation (29) in the text. Substitution of the Fourier coefficients for the potential and simplification give a quadratic equation for  $\sqrt{g}$ . This can be solved for the two roots of  $\sqrt{g}$  which are squared to get two roots of  $g$ . Which of the roots represents  $g$  in the physical problem cannot be determined. The purpose of the von Neumann analysis is to predict the multigrid smoothing rate,  $\mu$ , for HZEB1. Since  $\mu$  is governed by the maximum value of  $g$ ,

it is prudent to pick the root of  $g$  from the quadratic equation which has the maximum absolute value. The two roots are:

$$g = (\sqrt{g})^2 = \left[ \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right]^2 \quad (B3)$$

where:

$$a = \frac{1}{\omega} [(1 - M^2) (2 \cos \theta_x - 2) - 2\lambda^2]$$

$$b = 2\lambda^2 \cos \theta_y$$

and

$$c = (\omega - 1) a$$

#### Supersonic Flow - $M > 1$

HZEB1 applied to the transonic small-disturbance equation for supersonic flow can be written as:

$$\left[ (1 - M^2) \delta_{xx} + \frac{2\lambda^2}{\omega} \right] \Delta \phi_{ij} = R_{ij}^{n, n+1/2} \quad (B4)$$

where

$$R_{ij}^{n, n+1/2} = (1 - M^2) \delta_{xy} \phi_{ij}^n + \lambda^2 (\phi_{ij+1}^{n+1/2} - 2\phi_{ij}^n + \phi_{ij-1}^{n+1/2})$$



and the time levels of  $\phi$  are defined above. The inclusion of "new" values at  $i - 2$  would result in a pentadiagonal set of equations. To eliminate this, a  $\phi_{xt}$  type term is added. For stability, additional  $\phi_{xt}$  is also added. The following  $\phi_{xt}$  terms are used:

$$- (1 - M^2) (\Delta\phi_{ij} - \Delta\phi_{i-2,j}) + \beta (\Delta\phi_{ij} - \Delta\phi_{i-2,j})$$

Adding these terms to the left-hand side of equation (B4) gives:

$$\begin{aligned} & [-2 (1 - M^2) + \beta] (\Delta\phi_{ij} - \Delta\phi_{i-1,j}) + \lambda^2 \Delta\phi_{ij} = \\ & (1 - M^2) (\phi_{ij}^n - 2\phi_{i-1,j}^n + \phi_{i-2,j}^n) \\ & + \lambda^2 (\phi_{i,j+1}^{n+1/2} - 2\phi_{ij}^n + \phi_{i,j-1}^{n+1/2}) \end{aligned} \quad (B5)$$

Substitution of the Fourier coefficients for the potential followed by algebraic manipulation of the equation results in a quadratic equation for  $\sqrt{g}$ .

$$a \sqrt{g}^2 + b \sqrt{g} + c = 0 \quad (B6)$$

where:

$$a = \lambda^2 + [-2 (1 - M^2) + \beta] (1 - e^{-\sqrt{-1}\theta x})$$

$$b = -2\lambda^2 \cos \theta_y$$

and

$$c = \lambda^2 + (1 - M^2) + (1 - M^2) e^{-2\sqrt{-1} \theta_x}$$

Although the above represents a complete von Neumann analysis, HZEB1 was studied in greater detail in the present work. Values of  $g$  can easily be found from equation (B6), but it is not immediately obvious that the algorithm is stable,  $g < 1$  for all  $(\theta_x, \theta_y)$ . It is possible to show this analytically. Equation (B4) can be rewritten in the form:

$$z^2 + \eta z + \gamma = 0 \tag{B7}$$

where

$$z = \sqrt{g}$$

$$\eta = b/a$$

and

$$\gamma = c/a$$

Since  $\eta$  and  $\gamma$  are complex, the conditions that must be satisfied for  $z$  (and so  $g$ ) to be less than or equal to 1 are:

$$|\gamma| < 1$$

and

$$|\eta - \gamma\bar{\eta}| < 1 - |\gamma|^2$$

(B8)

where  $\bar{\eta}$  is the complex conjugate of  $\eta$ . The inequalities in equation (B6) lead to the restriction:

$$\beta > 2 - 2M^2 + \sqrt{(1 - M^2)(1 - b - M^2)} \quad \text{for } M > 1 \quad (\text{B9})$$

A numerical experiment was performed to determine the sharpness of the restriction in equation (B7). A slightly supersonic free-stream case ( $M_\infty = 1.01$ ) was run with a very thin airfoil (0.10-percent thick parabolic arc) to produce a nearly uniform supersonic flow. Various percentages of the "minimum"  $\beta$  prescribed in equation (B7) were used as the coefficient of the additional  $\phi_{xt}$  term. It was found that the  $\beta$  required by equation (B9) was not exact. As little as 80 percent of the  $\beta$  required by equation (B9) was found to provide stability. Although the von Neumann analysis does not exactly predict the performance of HZEB1, it does provide a qualitative indication of the performance.

## REFERENCES

1. Melson, N. Duane; and Keller, James D.: Use of CYBER 203 and CYBER 205 Computers for Three-Dimensional Transonic Flow Calculations. NASA TM-84604, April 1983.
2. South, Jerry C., Jr.; Keller, James D.; and Hafez, Mohamed M.: Vector Processor Algorithms for Transonic Flow Calculations. AIAA Paper 79-1457, July 1979.
3. Brandt, Achi: Multi-Level Adaptive Solutions to Boundary Value Problems. Mathematics of Computation, vol. 31, no. 138, April 1977, pp. 333-390. (Primary Source - Fedorenko, R. P.: A Relaxation Method for Solving Elliptic Difference Equations, USSR Comp. Math. and Math Phys., vol. 1, 1961, pp. 922-927.) (In Russian)
4. Jameson, Antony: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method, AIAA Paper 79-1458, July 1979. (Primary Source - Fedorenko, R. P.: The Speed of Convergence of One Iteration Process, USSR Comp. Math. and Math Phys., vol. 4, 1964, pp. 227-235.) (In Russian)
5. Jameson, Antony: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method, AIAA Paper 79-1458, July 1979. (Primary Source - Bakhvalov, N. S.: On the Convergence of Relaxation Method With Natural Constraints on the Elliptic Operator, USSR Comp Math. and Math Phys., vol. 6, 1966, pp. 101-135.) (In Russian)

6. Brandt, Achi: Multi-Level Adaptive Solutions to Boundary Value Problems. Mathematics of Computation, vol. 31, no. 138, April 1977, pp. 333-390.
7. Jameson, Antony: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method, AIAA Paper 79-1458, July 1979. (Primary Source - Nicholiades, R. A.: On the  $l^2$  Convergence of an Algorithm for Solving Finite Element Systems, Math. Comp., vol. 31, 1977, pp. 892-906.)
8. Jameson, Antony: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method, AIAA Paper 79-1458, July 1979. (Primary Source - Hackbusch, Wolfgang: Convergence of Multi-Grid Iteration Applied to Difference Equations, Koln University Mathematics Institute Report 79-5, April 1979.)
9. South, Jerry C., Jr.; and Brandt, Achi: Application of a Multilevel Grid Method to Transonic Flow Calculations, in Transonic Flow Calculations in Turbomachinery, T. C. Adamson and M. C. Platzer, eds., Hemisphere Publications, 1977.
10. Jameson, Antony: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method, AIAA Paper 79-1458, July 1979.
11. Caughey, D. A.: Multi-Grid Calculation of Three-Dimensional Transonic Potential Flows. AIAA Paper 83-0374, Jan. 1983.
12. Sankar, N. L.: A Multigrid Strongly Implicit Procedure for Two-Dimensional Transonic Potential Flow Problems. AIAA Paper 82-0931, June 1982.

13. Thames, F. C.: Multigrid Applications to Three-Dimensional Elliptic Coordinate Generation. Applied Mathematics and Computation, vol. 15, no. 4, 1984, pp. 325-342.
14. Eddy, E. P.: Stability in the Numerical Solution of Initial Value Problems in Partial Differential Equations. NOLM 102, Naval Ordnance Laboratory, White Oak, Silver Spring, Maryland, 1949.
15. Crank, J.; and Nicholson, P.: A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type. Proceedings of the Cambridge Philosophical Society, vol. 43, no. 50, 1947, pp. 50-67.
16. O'Brien, G. G.; Hyman, M. A.; and Kaplan, S.: A Study of the Numerical Solution of Partial Differential Equations, J. Math. and Phys., vol. 29, 1950, pp. 223-251.
17. Roach, Patrick J.: Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, New Mexico, 1972.

TABLE 1

VLOR

$$M < 1$$

Algorithm:

$$(1 - M^2) E_x^{n+1} + (1 - M^2)(-2I) \phi_{1j}^{n+1} + (1 - M^2)(E_x^+) \phi_{1j}^n + \lambda^2 (E_y^+ - 2I + E_y^-) \phi_{1j}^{n+1} = 0$$

52

Amplification factor:

$$g = \frac{(-1) \left[ (1 - M^2) (\cos \theta_x + \sqrt{-1} \sin \theta_x - 2 \left(\frac{\omega-1}{\omega}\right)) + \lambda^2 \left(\frac{\omega-1}{\omega}\right) (2 \cos \theta_y - 2) \right]}{(1 - M^2) \left( -\frac{2}{\omega} + \cos \theta_x - \sqrt{-1} \sin \theta_x \right) + \frac{\lambda^2}{\omega} (2 \cos \theta_y - 2)}$$

VLOR

$M > 1$

Algorithm:

$$(1 - M^2) \left[ (E_x^-)^2 - 2E_x^- + I \right] \phi_{ij}^{n+1} + \lambda^2 (E_y^- - 2I + E_y^+) \phi_{ij}^{n+1} + \beta \left[ (I - E_x^-) \phi_{ij}^{n+1} - (I - E_x^-) \phi_{ij}^n \right] = 0$$

Amplification factor:

$$g = \beta \left[ 1 - (\cos \theta_x - \sqrt{-1} \sin \theta_x) \right] / \left\{ (1 - M^2) \left[ (\cos \theta_x - \sqrt{-1} \sin \theta_x)^2 - 2 (\cos \theta_x - \sqrt{-1} \sin \theta_x) + 1 \right] \right. \\ \left. + \lambda^2 (2 \cos \theta_y - 2) + \beta \left[ 1 - (\cos \theta_x - \sqrt{-1} \sin \theta_x) \right] \right\}$$



HLOR

M < 1

Algorithm:

$$\left\{ (1 - M^2) (E_x^- - 2I + E_x^+) + \lambda^2 (-2) I \right\} \tilde{p}_{1j}^{n+1} + \lambda^2 E_y^- \phi_{1j}^{n+1} + \lambda^2 E_x^+ \phi_{1j}^n$$

56

Amplification factor:

$$g = \frac{-\lambda^2 (\cos \theta_y + \sqrt{-1} \sin \theta_y) - \frac{\omega-1}{\omega} [(1 - M^2)(2 \cos \theta_x - 2) - 2\lambda^2]}{\frac{1}{\omega} (1 - M)^2 (2 \cos \theta_x - 2) + \lambda^2 (\cos \theta_y - \sqrt{-1} \sin \theta_y - \frac{2}{\omega})}$$

HLOR

M > 1

Algorithm:

$$\begin{aligned} & [(1 - M^2)(E_x^-)^2] \phi_{ij}^n + [(1 - M^2) (-2E_x^- + I)] \phi_{ij}^{n+1} + \lambda^2 [E_y^+ - I] \phi_{ij}^n + \lambda^2 [-I + E_y^-] \phi_{ij}^{n+1} \\ & + (1 - M^2) (\phi_{ij}^{n+1} - \phi_{ij}^n) - \beta [(I - E_x^-) \phi_{ij}^{n+1} - (I - E_x^-) \phi_{ij}^n] = 0 \end{aligned}$$

Amplification factor:

$$g = \frac{(-1) [(1 - M^2) (e^{\sqrt{-1}\theta} x)^2 + \lambda^2 (e^{\sqrt{-1}\theta} y - 1) - (1 - M^2) + \beta (1 - e^{-\sqrt{-1}\theta} x)]}{(1 - M)^2 (-2e^{-\sqrt{-1}\theta} x + 1) + \lambda^2 (e^{-\sqrt{-1}\theta} y - 1) + (1 - M^2) - \beta (1 - e^{-\sqrt{-1}\theta} x)}$$

ADLOR

Algorithm:

A sweep of VLOR followed by a sweep of HLOR or a sweep of HLOR followed by a sweep of VLOR

Amplification factor:

$$g_{ADLOR} = \sqrt{g_{VLOR} \cdot g_{HLOR}}$$

VZEB1

M < 1

Algorithm:

$$(1 - M^2) [E_x^- \phi_{ij}^n - 2I \tilde{\phi}_{ij}^{n+1/2} + E_x^+ \phi_{ij}^n] + \lambda^2 [E_y^- - 2I + E_y^+] \tilde{\phi}_{ij}^{n+1/2} - \beta [\tilde{\phi}_{ij}^{n+1/2} - \phi_{ij}^{n-1/2} - E_x^- (\phi_{ij}^n - \phi_{ij}^{n-1})] = 0$$

57

Amplification factor equation:

$$(\sqrt{g})^2 [(1 - M^2) (-2/\omega) + (\lambda^2/\omega) (2 \cos \theta_y - 2)] + \sqrt{g} [(1 - M^2) (2 \cos \theta_x)] + [(\frac{\omega-1}{\omega}) (-2) (1 - M^2) + \lambda^2 (\frac{\omega-1}{\omega}) (2 \cos \theta_y - 2)] = 0$$

VZEB1

M > 1

Algorithm:

$$(1 - M^2) \left[ (E_x^-)^2 \phi_{ij}^{n-1/2} - 2E_x^- \phi_{ij}^n + I \phi_{ij}^{n+1/2} \right] + \lambda^2 \left[ E_y^- - 2I + E_y^+ \right] \phi_{ij}^{n+1/2} - \beta \left[ \phi_{ij}^{n+1/2} - \phi_{ij}^{n-1/2} - E_x^- (\phi_{ij}^n - \phi_{ij}^{n-1}) \right] = 0$$

58

Amplification factor equation:

$$(\sqrt{g})^3 \left[ (1 - M^2) + \lambda^2 (2 \cos \theta_y - 2) - \beta \right] + (\sqrt{g})^2 \left[ (1 - M^2) (-2e^{\sqrt{-1} \theta} x) + \beta e^{\sqrt{-1} \theta} x \right] + \sqrt{g} \left[ (1 - M^2) (e^{-2 \sqrt{-1} \theta} x + \beta) \right] - \beta e^{\sqrt{-1} \theta} x = 0$$

HZEB1

M < 1

Algorithm:

$$[(1 - M^2) (E_x^- - 2I + E_x^+) - 2\lambda^2 \tau] \left(\frac{1}{\omega} \phi_{ij}^{n+1/2}\right) + \frac{\omega-1}{\omega} \phi_{ij}^{n-1/2} + \lambda^2 (E_y^- + E_y^+) \phi_{ij}^n = 0$$

59

Amplification factor equation:

$$(\sqrt{g})^2 \left\{ [(1 - M^2) (2 \cos \theta_x - 2) - 2\lambda^2] \left(\frac{1}{\omega}\right) \right\} + \sqrt{g} \left[ \lambda^2 (2 \cos \theta_y) \right] + \frac{\omega-1}{\omega} [(1 - M^2) (2 \cos \theta - 2) - 2\lambda^2] = 0$$

HZEBI

$M > 1$

Algorithm:

$$\begin{aligned} & [-2(1 - M^2) + \beta] (\Delta\phi_{ij} - \Delta\phi_{i-1,j}) + \lambda^2 \Delta\phi_{ij} = (1 - M^2) (\phi_{ij}^n - 2\phi_{i-1,j}^n + \phi_{i-2,j}^n) \\ & + \lambda^2 (\phi_{i,j+1}^{n+1/2} - 2\phi_{ij}^n + \phi_{i,j-1}^{n+1/2}) \end{aligned}$$

60

Amplification factor equation:

$$\begin{aligned} & (\sqrt{g})^2 [(2e^{-\sqrt{-1}\theta} x - 2)(1 - M^2) + \lambda^2 + \beta (1 - e^{-\sqrt{-1}\theta} x)] \\ & - \sqrt{g} [\lambda^2 (e^{\sqrt{-1}\theta} y + e^{-\sqrt{-1}\theta} y)] + [(-1)(1 - M^2) e^{-2\sqrt{-1}\theta} x + \lambda^2 + (1 - M^2) \\ & - \beta (1 - e^{-\sqrt{-1}\theta} x)] = 0 \end{aligned}$$

ADZEB1

Algorithm:

A sweep of VZEB1 followed by a sweep of HZEB1 or a sweep of HZEB1 followed by a sweep of VZEB1

Amplification factor:

$$9ADZEB1 = \sqrt{9VZEB1 \cdot 9HZEB1}$$



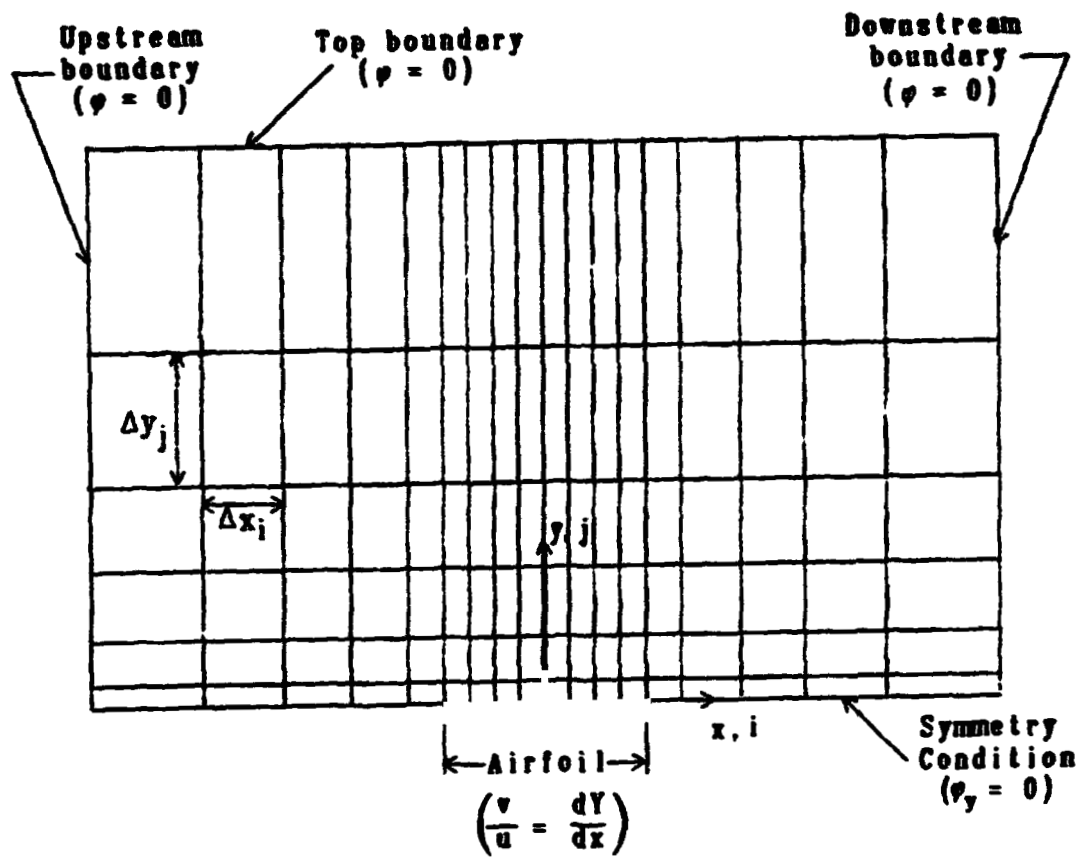
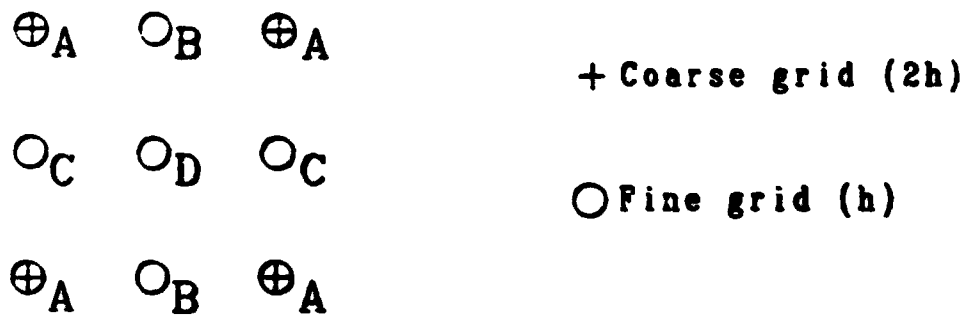


Figure 1.- Coordinate system and schematic of stretched grid.



**Type A.**  $\psi^h(x_0, y_0) = \psi^{2h}(x_0, y_0)$

**Type B.**  $\psi^h(x_0, y_0) = \frac{1}{2} \{ \psi^{2h}(x_0 + h, y_0) + \psi^{2h}(x_0 - h, y_0) \}$

**Type C.**  $\psi^h(x_0, y_0) = \frac{1}{2} \{ \psi^{2h}(x_0, y_0 + h) + \psi^{2h}(x_0, y_0 - h) \}$

**Type D.**  $\psi^h(x_0, y_0) = \frac{1}{4} \{ \psi^{2h}(x_0 + h, y_0 + h) + \psi^{2h}(x_0 + h, y_0 - h) + \psi^{2h}(x_0 - h, y_0 - h) + \psi^{2h}(x_0 - h, y_0 + h) \}$

Figure 2.- Prolongation operator.

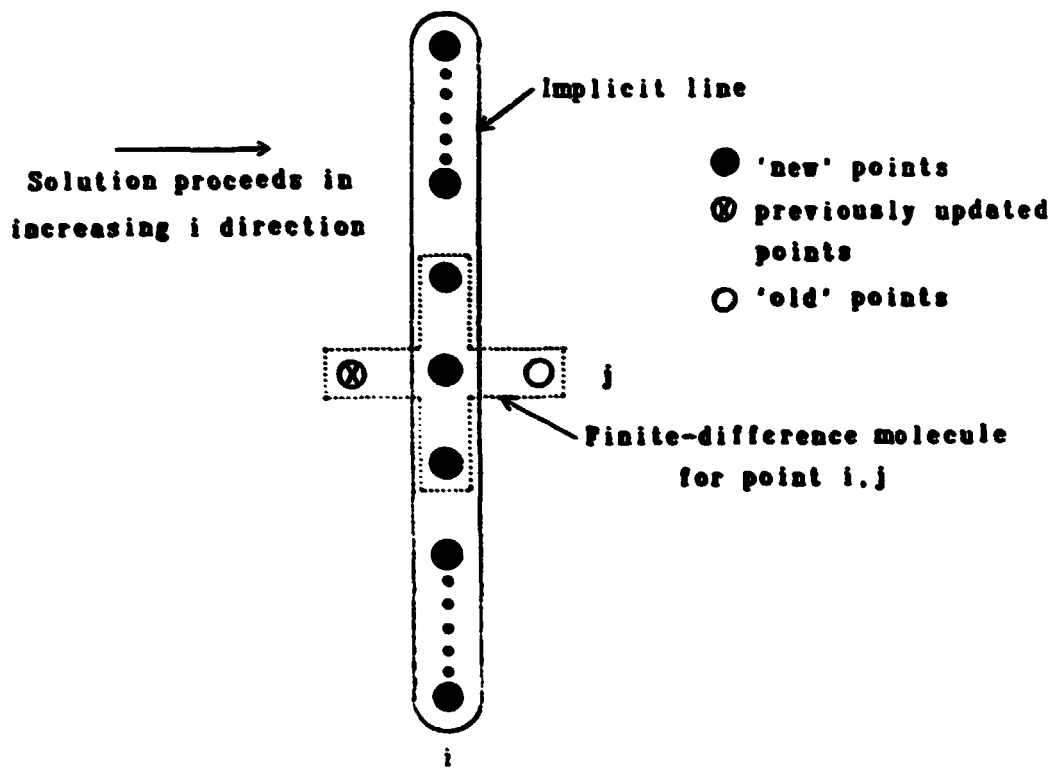


Figure 3.- Schematic of VLOR algorithm.

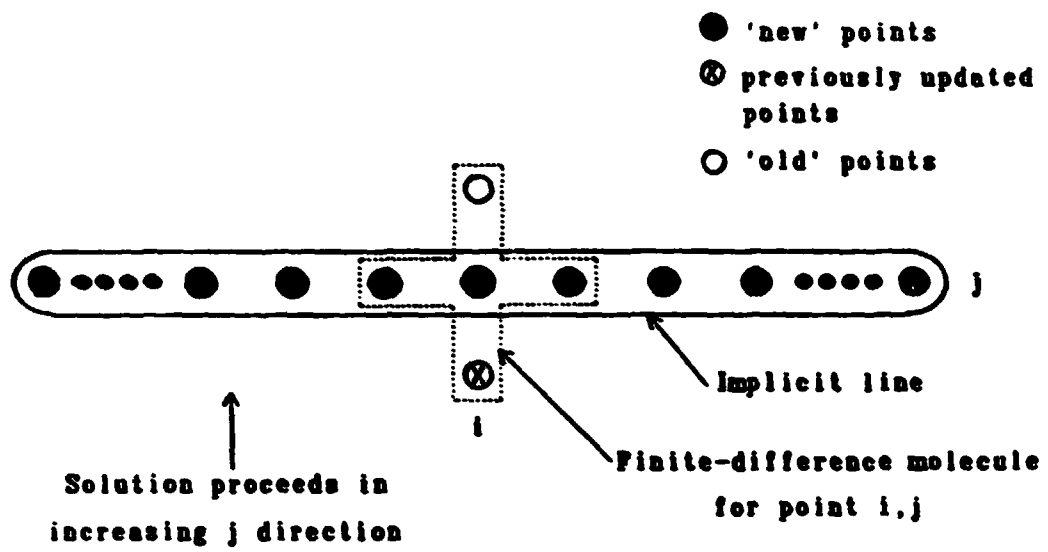


Figure 4.- Schematic of HLOR algorithm.

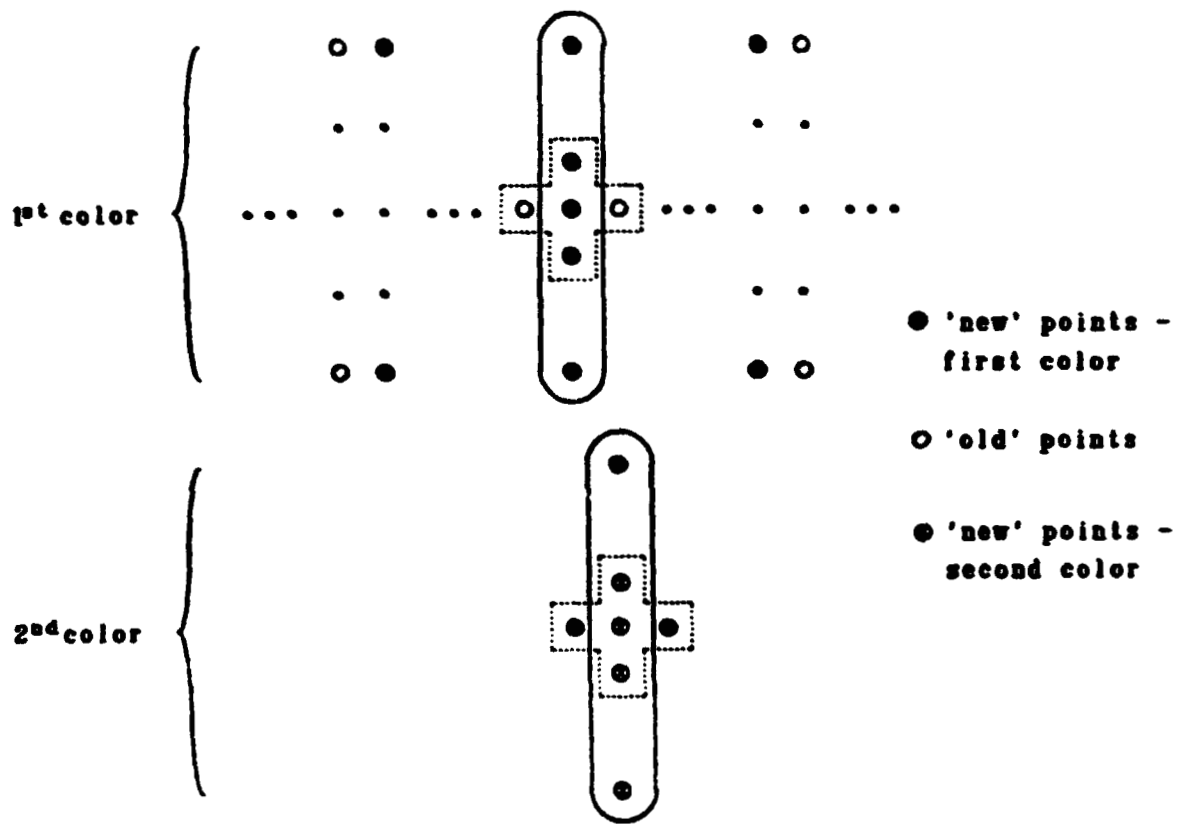
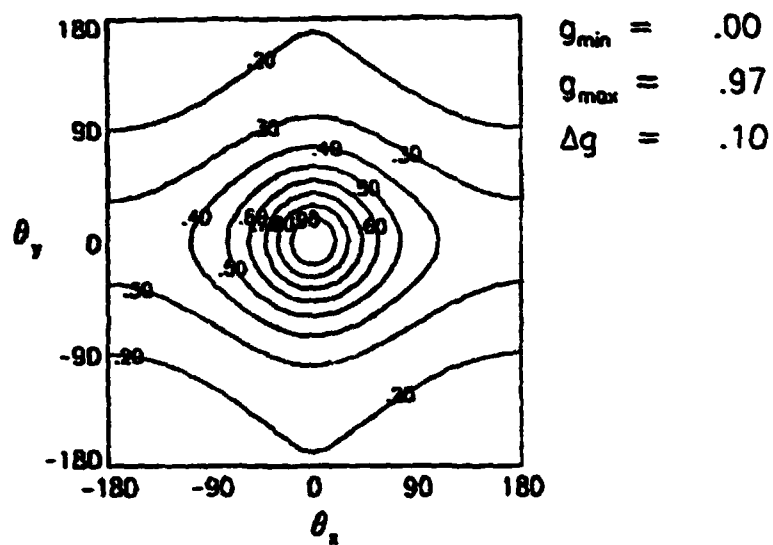
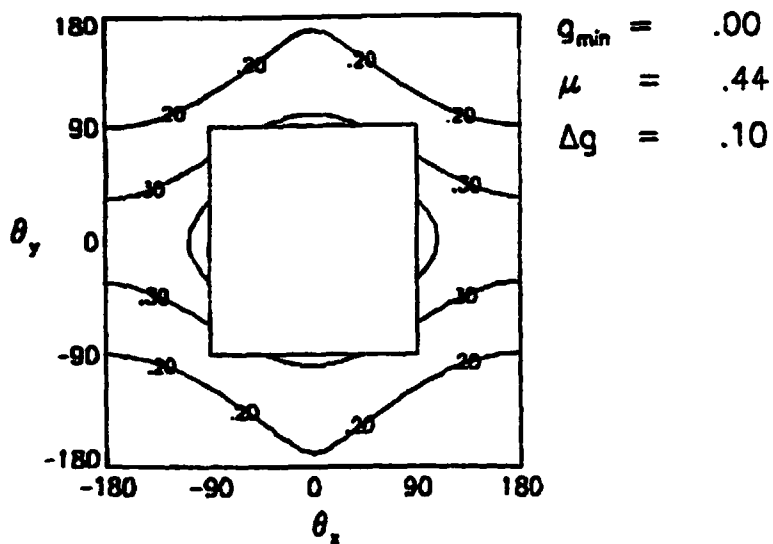


Figure 5.- Schematic of VZEB1 algorithm.



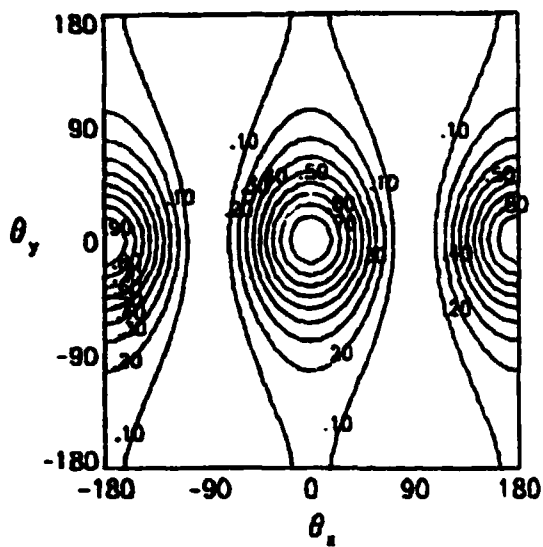
von Neumann Damping Rate Contours  
 Equation:  $(1 - M^2) \phi_{xx} + \phi_{yy} = 0$   
 Method: VLOR  
 Parameters:  
 $\omega = 1.000$   
 $\lambda = 1.000$   
 $M = .1000$   
 $\beta = 0.$

Figure 6.- von Neumann damping rate contours for VLOR.



**Multigrid Smoothing Factor Contours**  
 Equation:  $(1 - M^2) \phi_{xx} + \phi_{yy} = 0$   
 Method: VLOR  
 Parameters:  
 $\omega = 1.000$   
 $\lambda = 1.000$   
 $M = .1000$   
 $\beta = 0.$

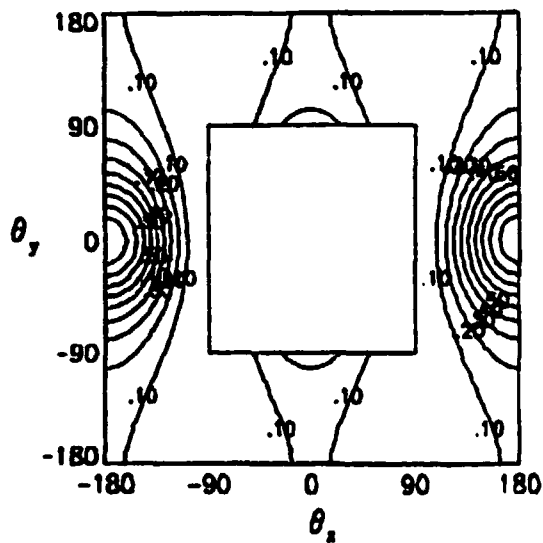
Figure 7.- Multigrid smoothing factor contours for VLOR.



$g_{\min} = .00$   
 $g_{\max} = .97$   
 $\Delta g = .10$

**Parameters:**  
 $\omega = 1.000$   
 $\lambda = 1.000$   
 $M = .1000$   
 $\beta = 0.$

(a) von Neumann damping rate contours.



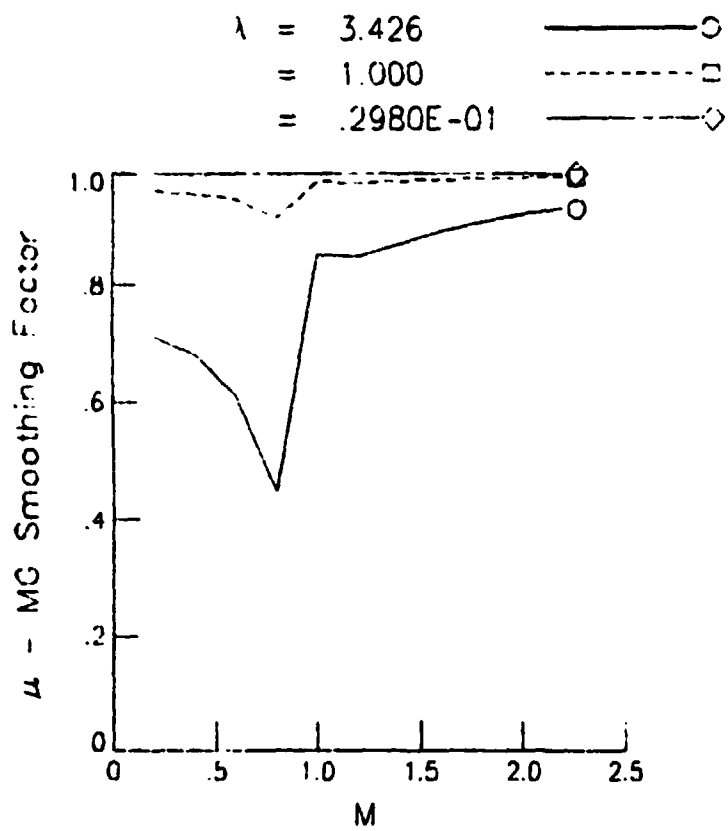
$\mu_{\min} = .00$   
 $\mu = .97$   
 $\Delta \mu = .10$

**Parameters:**  
 $\omega = 1.000$   
 $\lambda = 1.000$   
 $M = .1000$   
 $\beta = 0.$

(b) Multigrid smoothing factor contours.

Figure 8.- von Neumann damping rate and multigrid smoothing factor contours for VZEB1.





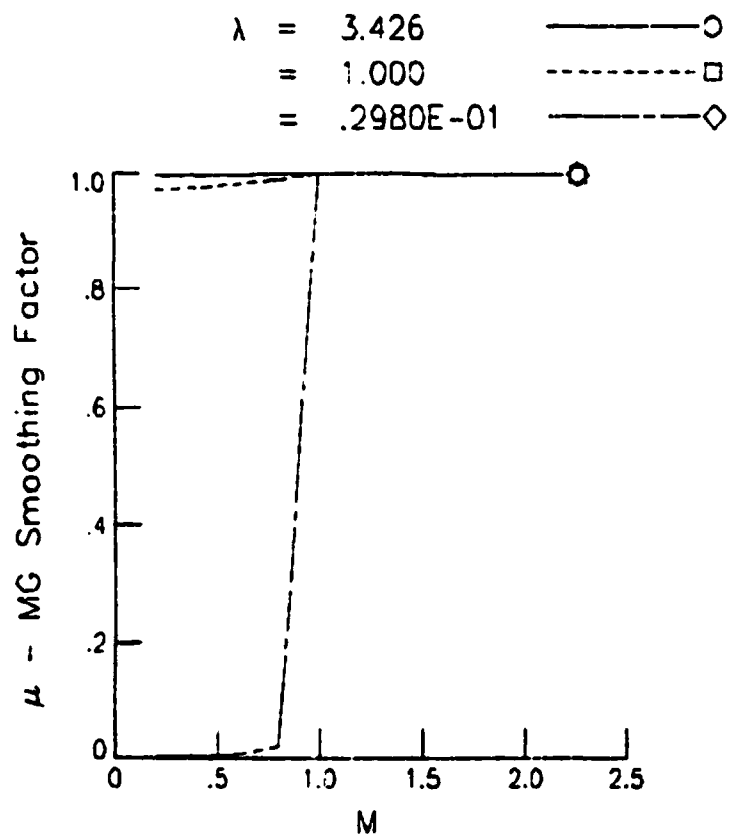
Equation:  $(1 - M^2) \phi_{xx} + \phi_{yy} = 0$

Method: VZEB1

$\omega = 1.000$

$\beta = 1.000$

Figure 9.- Multigrid smoothing factor versus M and λ for VZEB1.



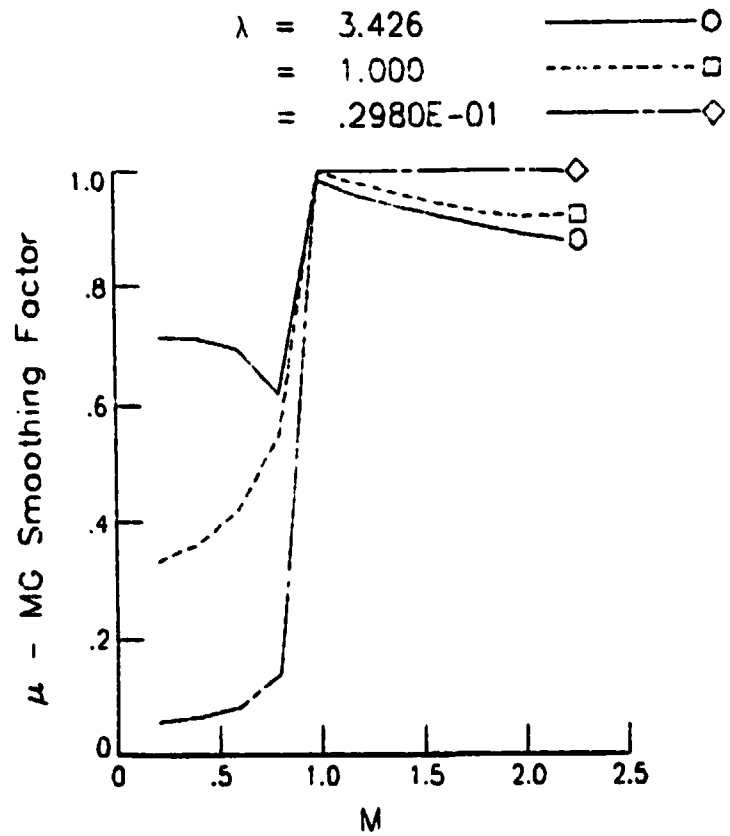
Equation:  $(1 - M^2) \phi_{xx} + \phi_{yy} = 0$

Method: HZEB1

$\omega = 1.000$

$\beta = 10.00$

Figure 10.- Multigrid smoothing factor versus M and λ for HZEB1.



Equation:  $(1 - M^2) \phi_{xx} + \phi_{yy} = 0$

Method: ADZEB1

$\omega = 1.000$

$\beta = 10.00$

Figure 11.- Multigrid smoothing factor versus M and λ for ADZEB1.

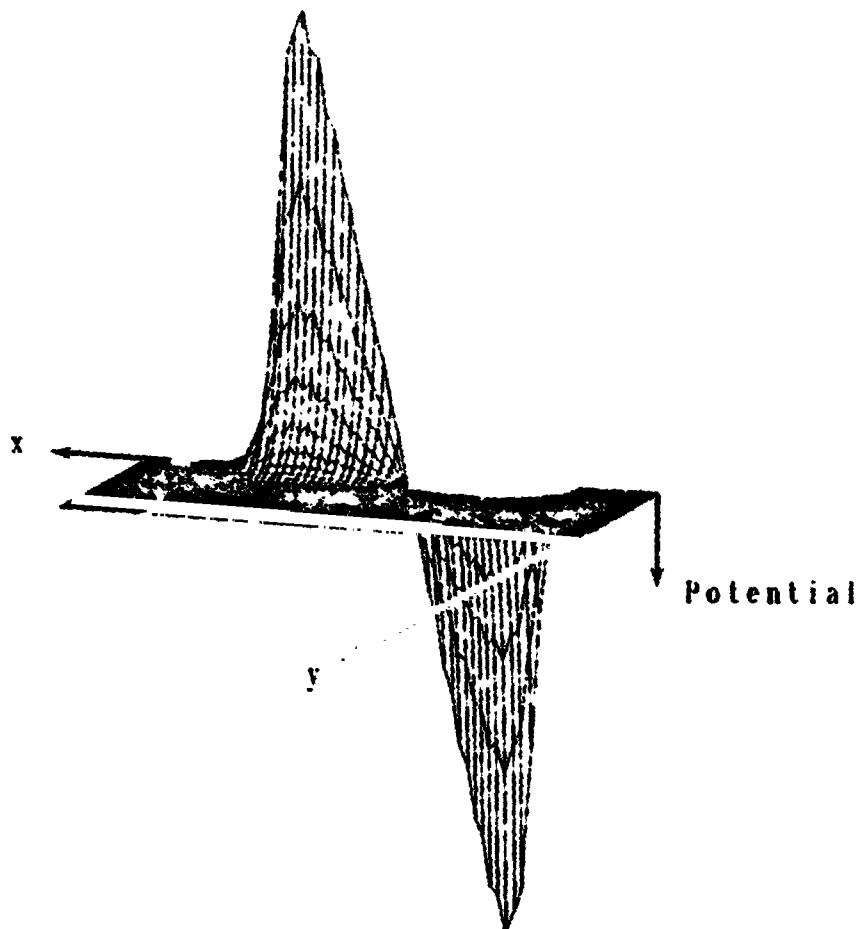


Figure 12.- Potential on finest grid of first cycle using 32B1.

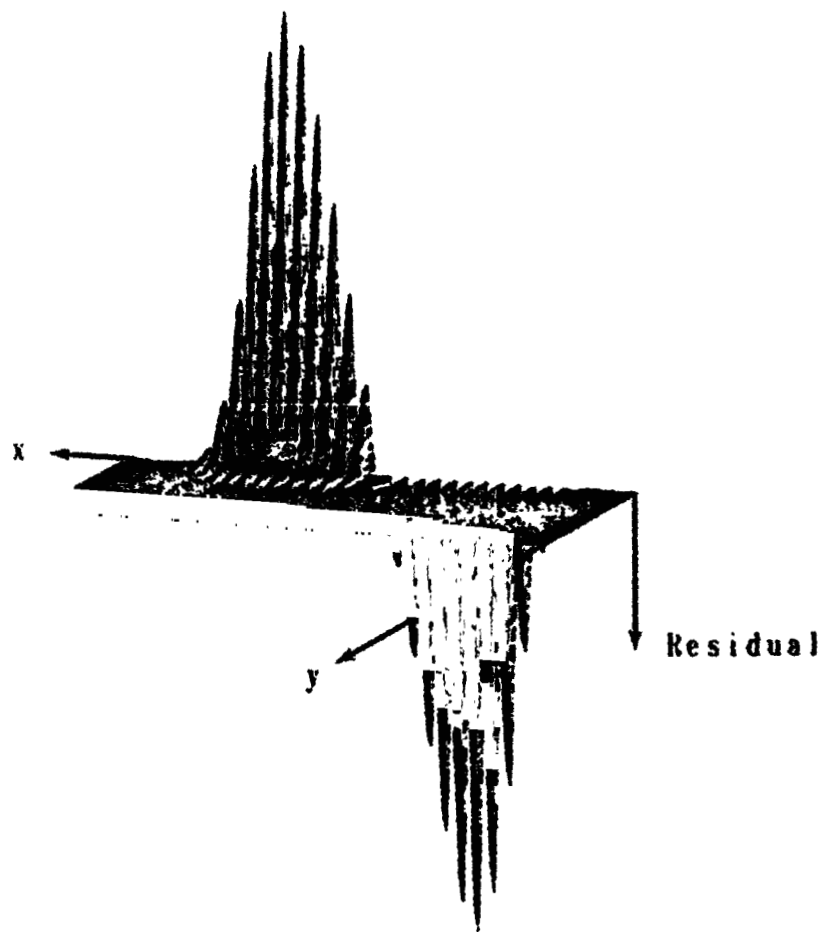


Figure 13.- Residual on finest grid using ZEB1 ( $M_\infty = 0.1$ ).

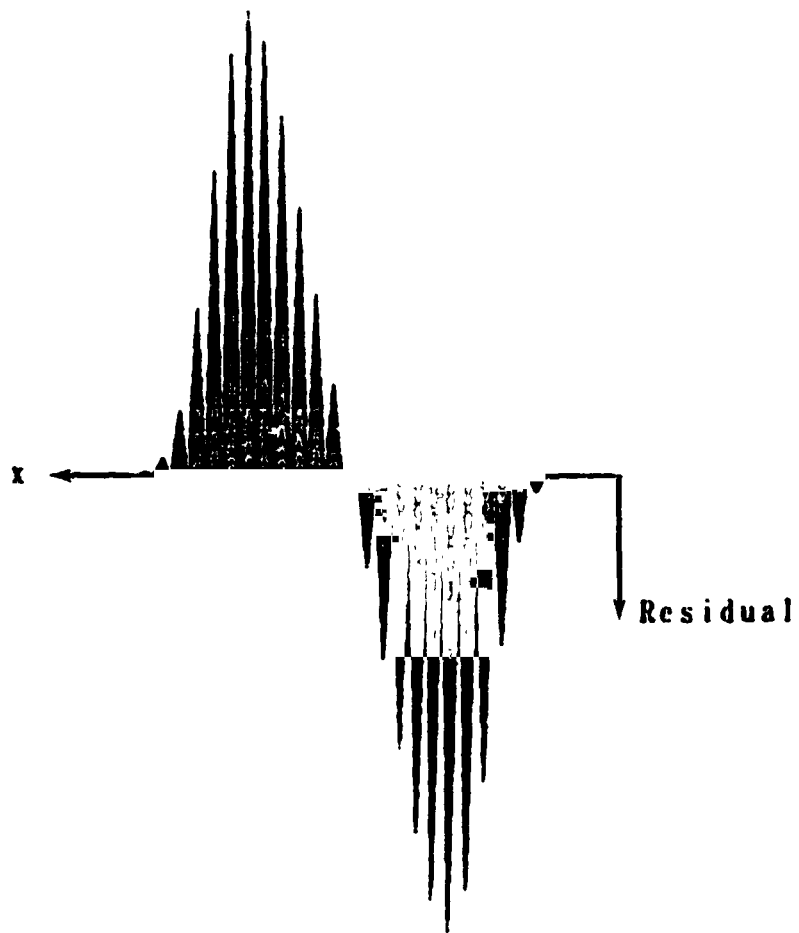


Figure 14.- End view of residual on finest grid using EB1  
( $M_\infty = 0.1$ ).

MAXIMUM  $|f| = 1.7007$

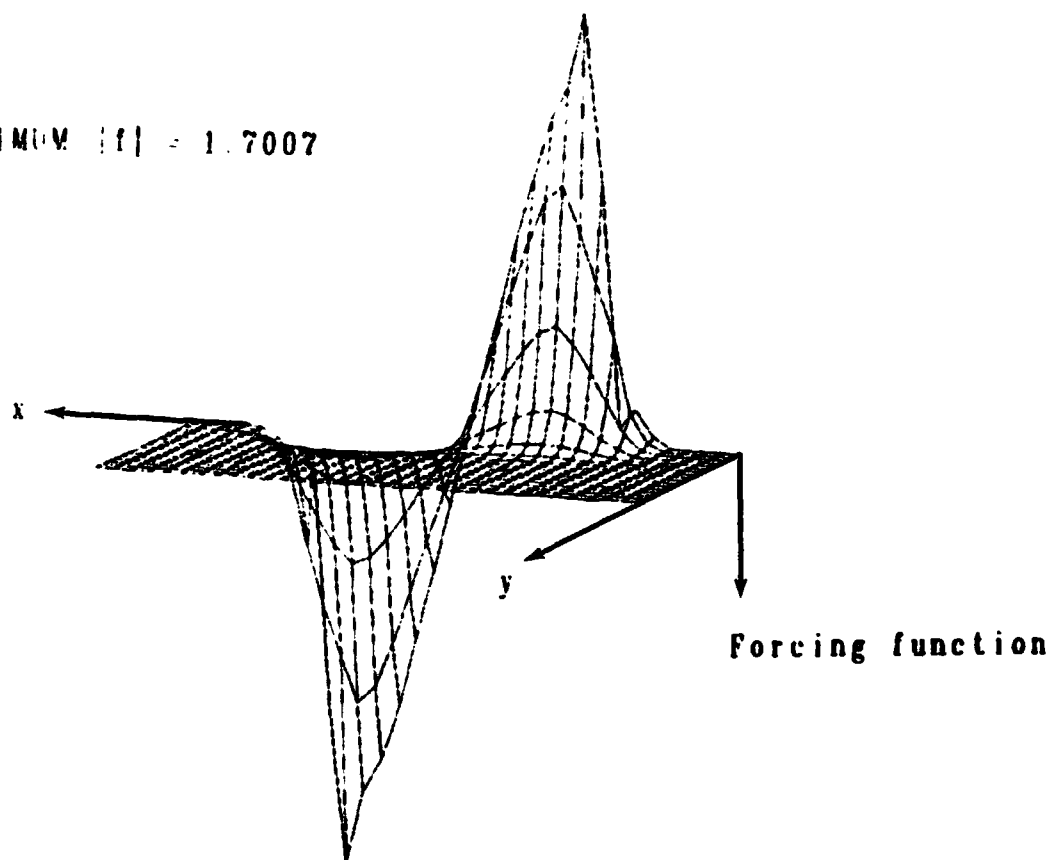


Figure 15.- Forcing function on second finest grid using VZEB1 and no residual weighting ( $M_\infty = 0.1$ ).

MAXIMUM  $|f| = 0.4323$

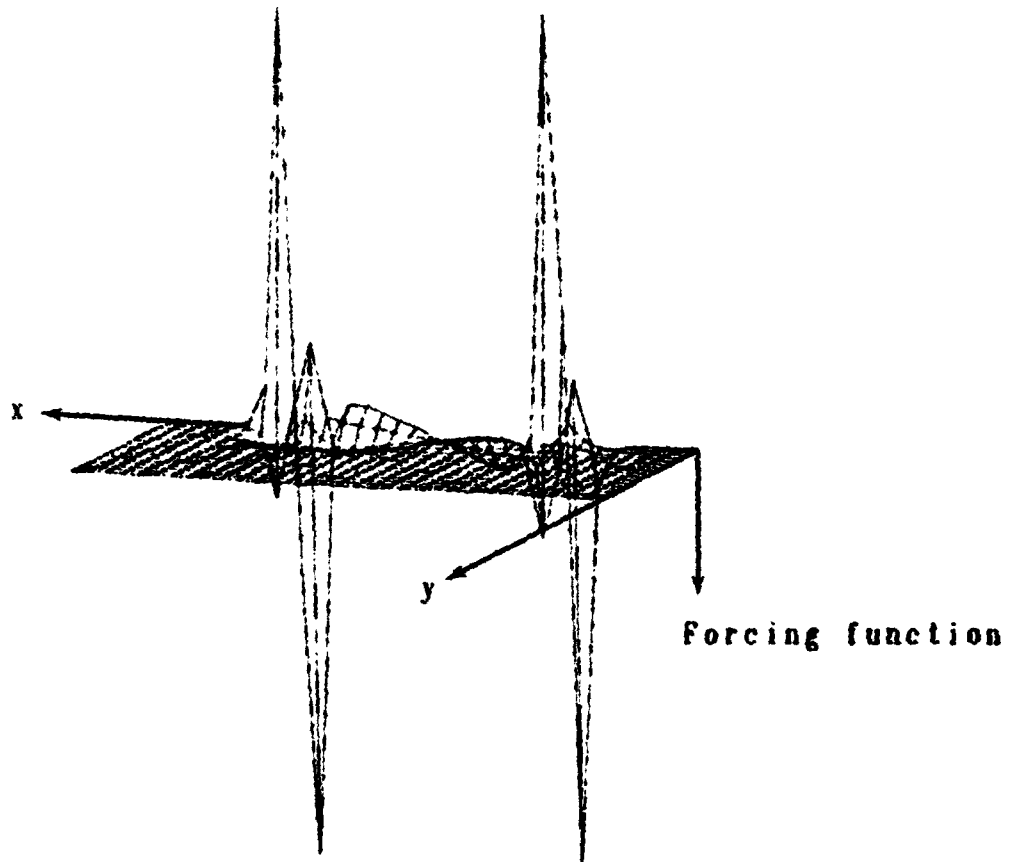


Figure 16.- Forcing function on second finest grid using VZEB1 and residual weighting ( $M_{\infty} = 0.1$ ).

*[Handwritten signature]*



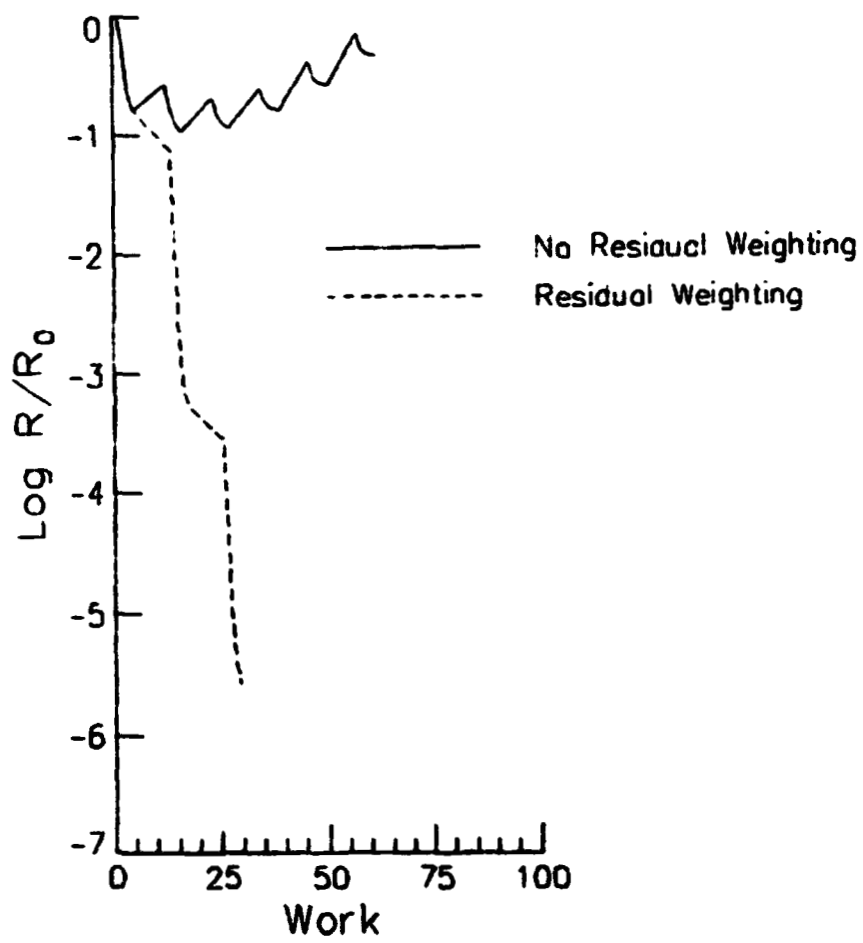


Figure 17.- Effect of residual weighting on VZEB1 ( $M_\infty = 0.1$ ).

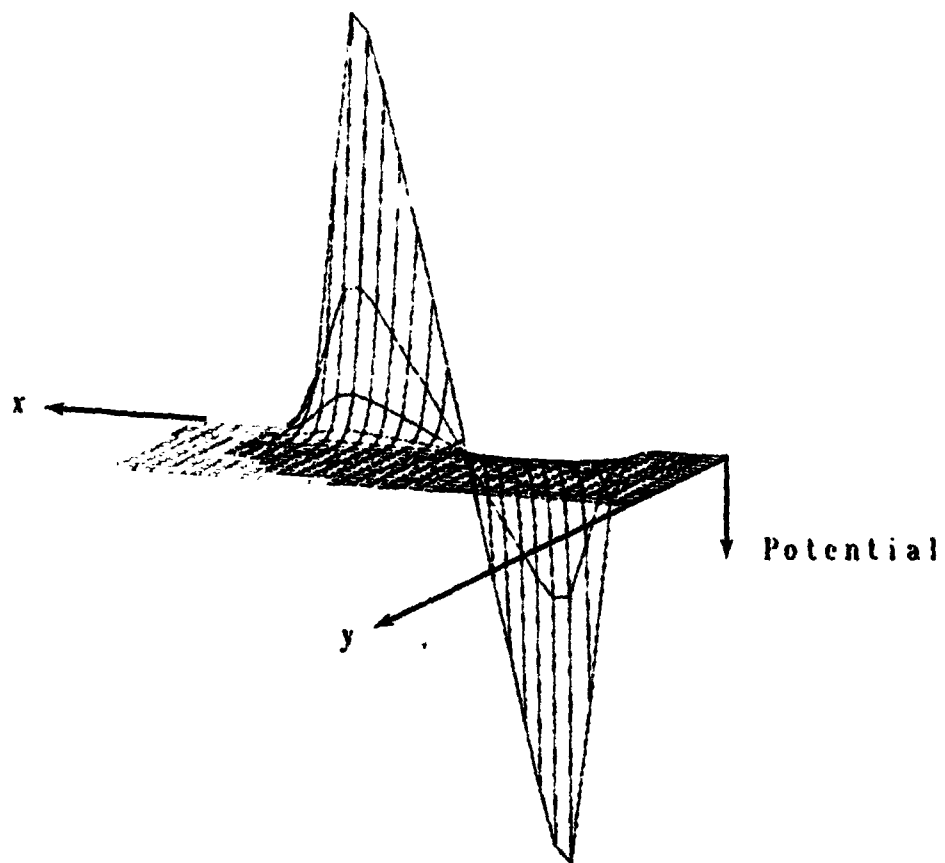


Figure 18.- Restricted potential from finest grid using VZEB1  
( $M_\infty = 0.1$ ).

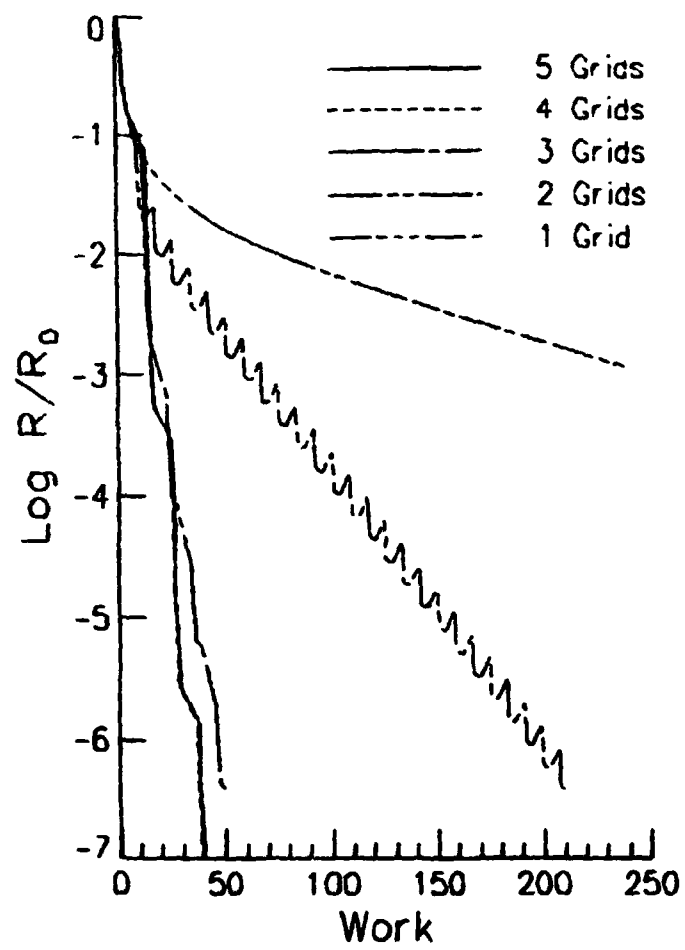


Figure .9.- Effect of multigrid on VZEB1 ( $N_{\infty} = 0.1$ ).

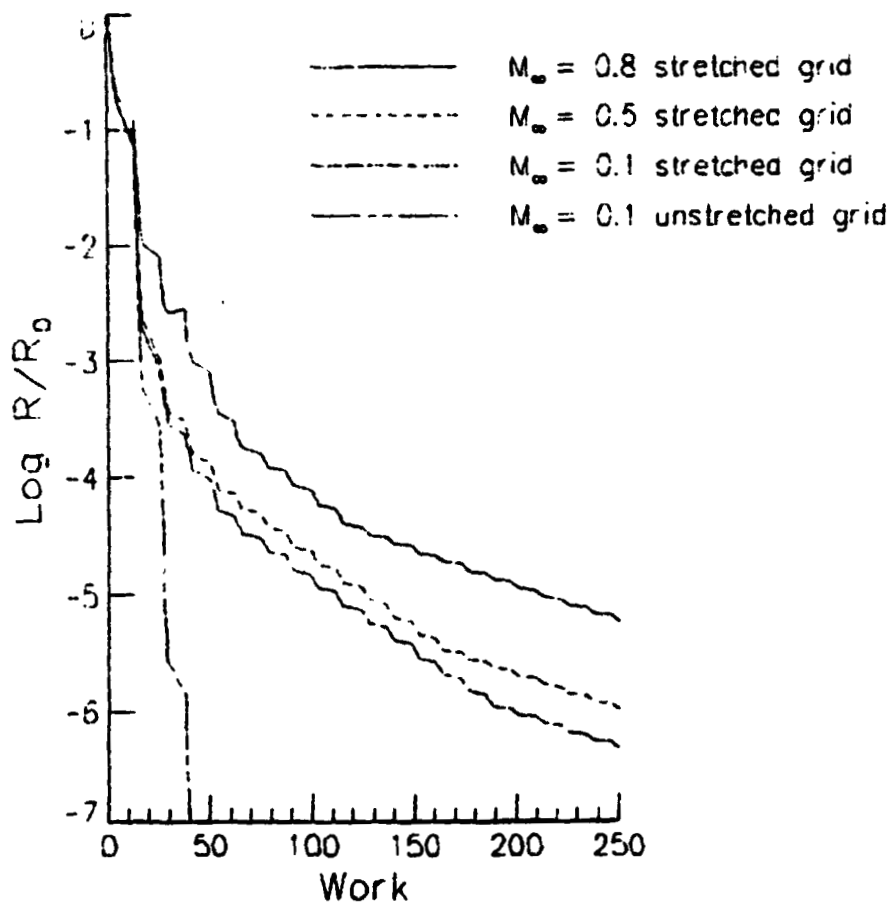


Figure 20.- Effect of  $M_\infty$  and grid stretching on VZEB1.

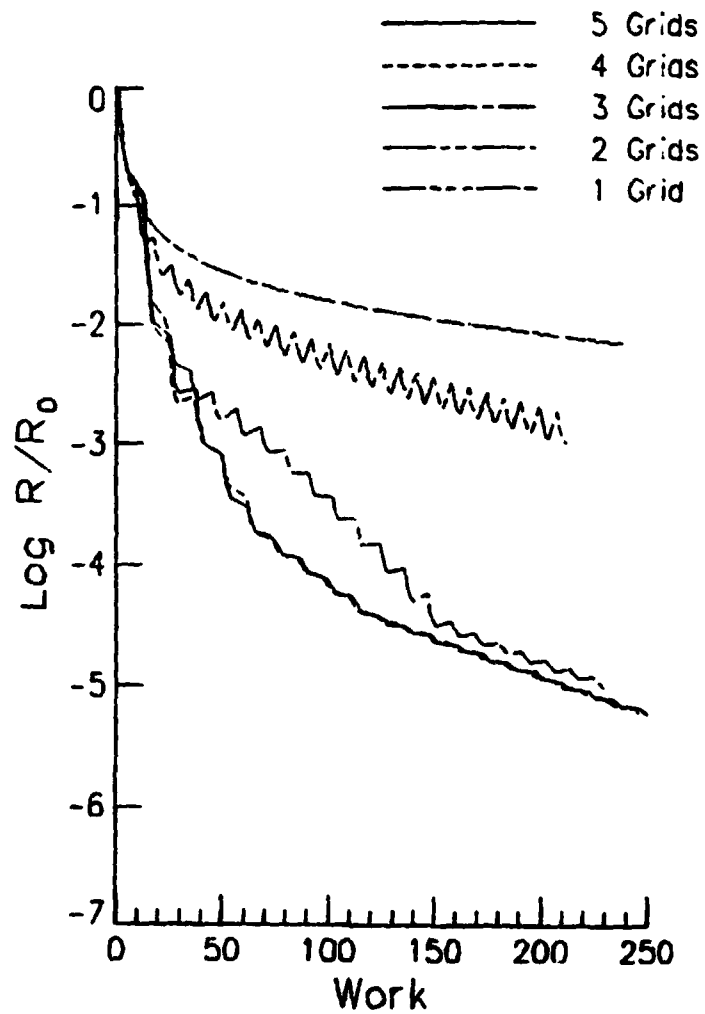


Figure 21.- Effect of multigrid on VZEB1 ( $M_\infty = 0.8$ ).

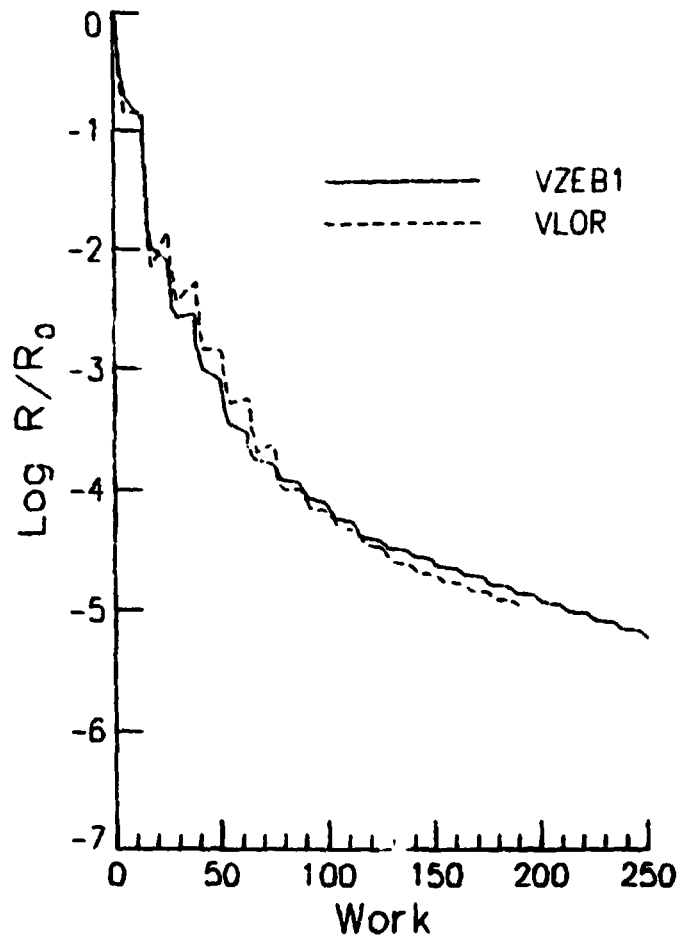


Figure 22.- Comparison of VZEB1 and VLOR in multigrid ( $M_\infty = 0.8$ , 5 grids).

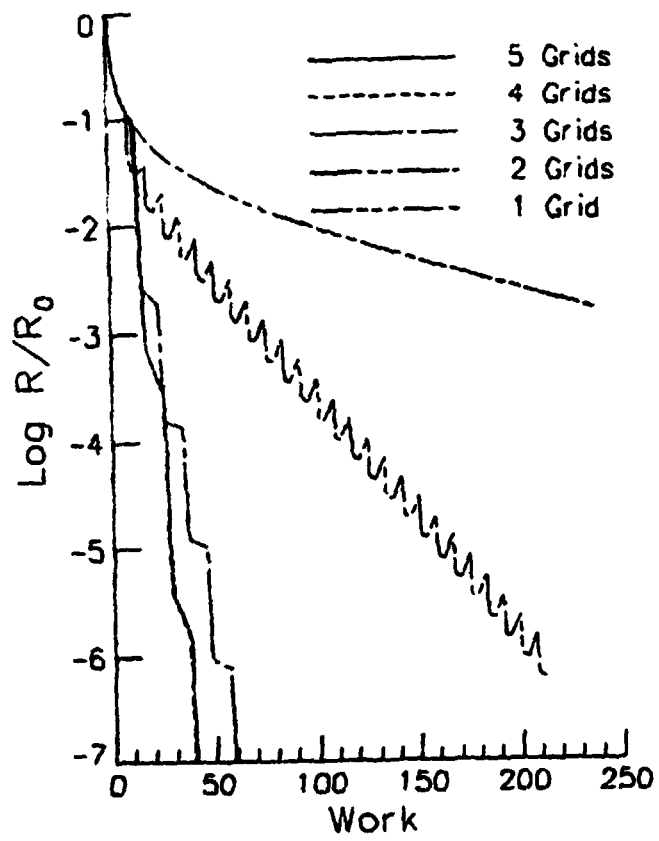


Figure 23.- Effect of multigrid on HZEB1 ( $M_\infty = 0.1$ ).

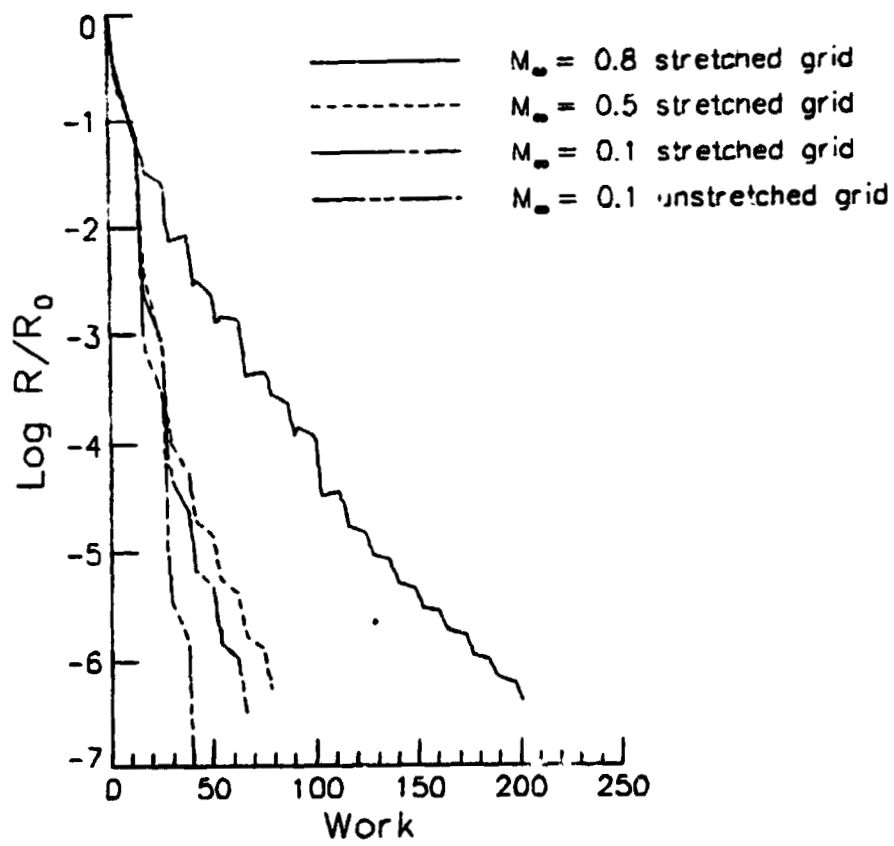


Figure 24.- Effect of  $M_\infty$  and grid stretching on HZEB1.



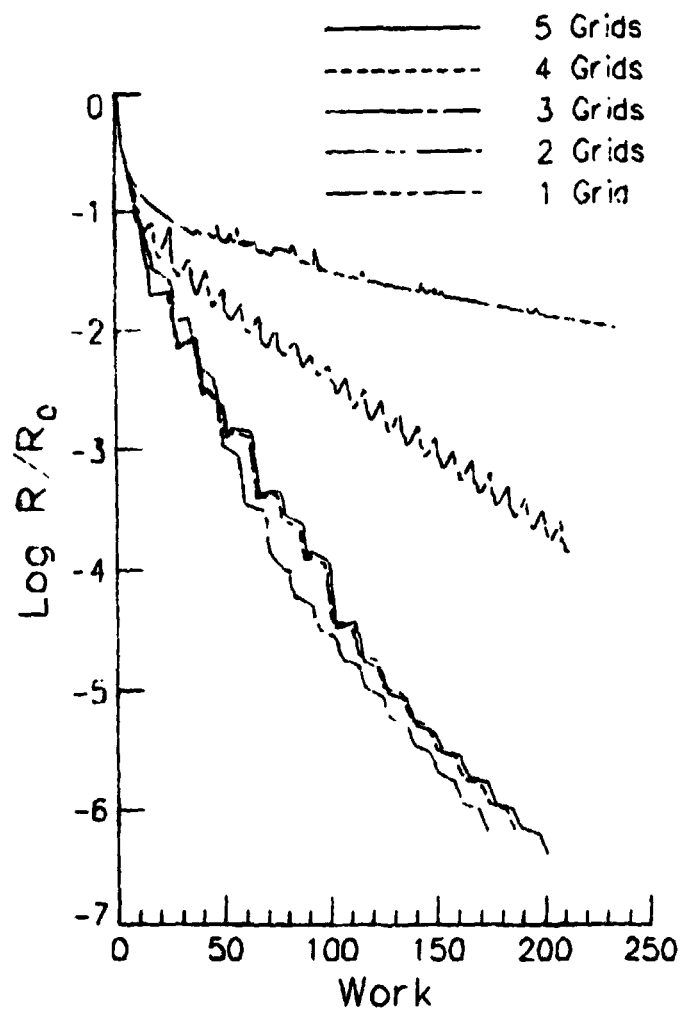


Figure 25.- Effect of multigrid on HZEB1 ( $M_{\infty} = 0.8$ ).

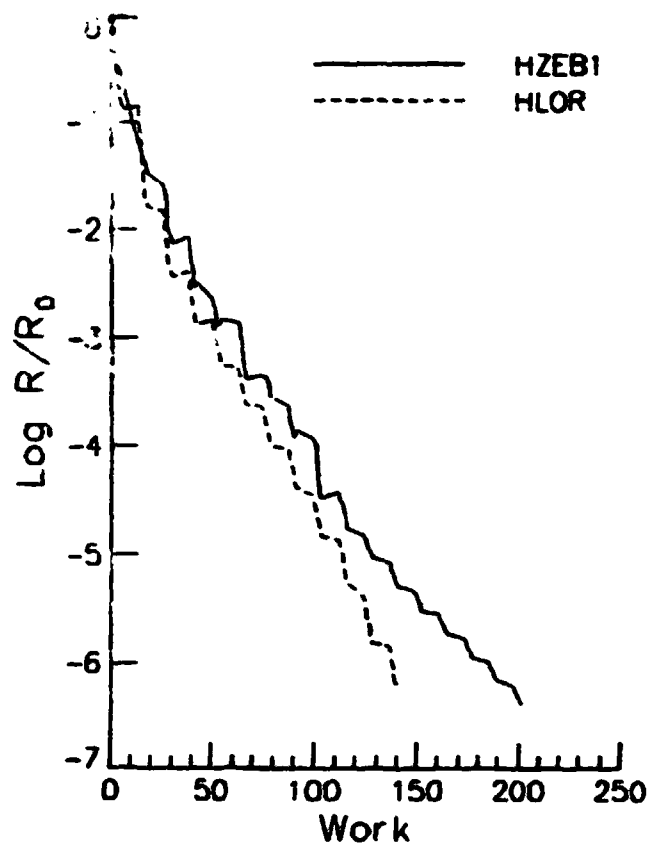


Figure 26.- Comparison of HZEB1 and HLOR in multigrid ( $M_\infty = 0.8$ , 5 grids).

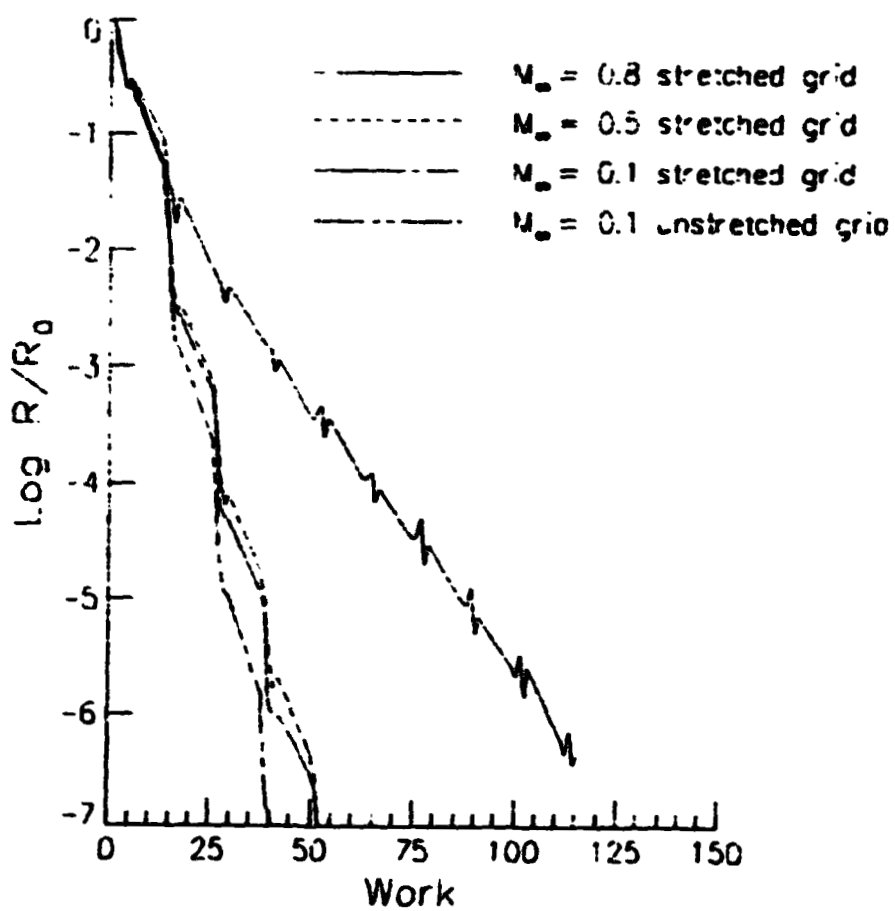


Figure 27.- Effect of  $M_0$  and grid stretching on ADZEB1.

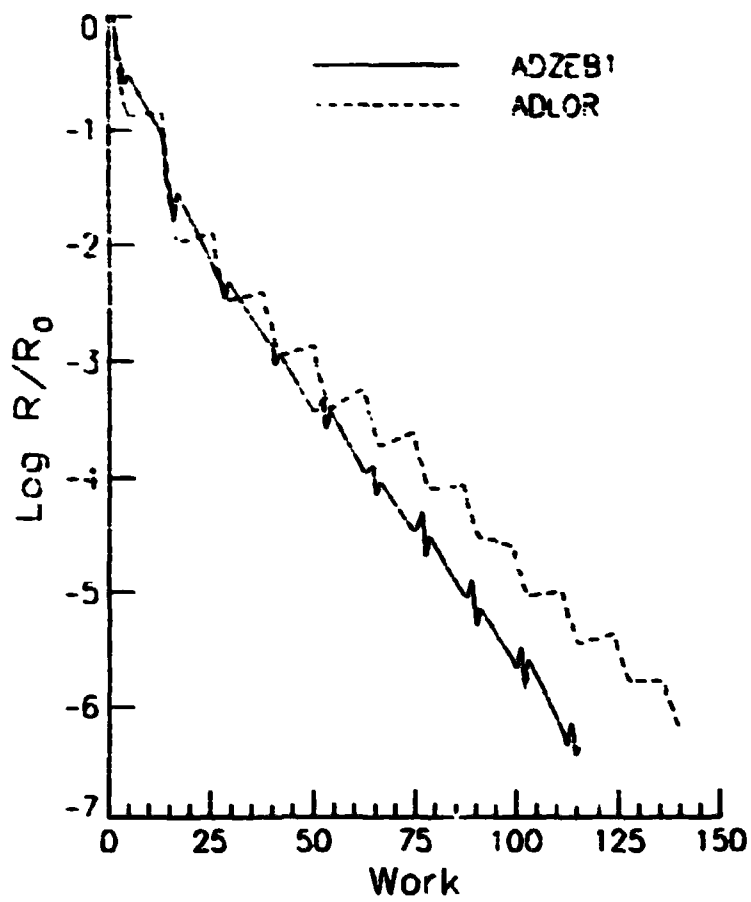


Figure 28.- Comparison of ADZEB1 and ADLOR in multigrid ( $M_\infty = 0.8$ , 5 grids).