NASA Conference Publication 2202

Multigrid Methods

Proceedings of a symposium held at
Ames Research Center
Moffett Field, California
October 21-22, 1981

NASA

NASA Conference Publication 2202

Multigrid Methods

Proceedings of a symposium held at
Ames Research Center
Moffett Field, California
October 21-22, 1981

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1982

PREFACE

This publication is a compilation of the papers presented at the Symposium on Multigrid Methods held at Ames Research Center on October 21-22, 1981. The papers represent an international sampling of the most recent developments in numerical solution of certain types of partial differential equations by rapidly converging sequences of operations on supporting grids that range from very fine to very coarse.

The symposium was organized for the purpose of bringing together scientists having individual experience with, and a common interest in, multigrid processes. For the most part the common ground of these processes is an underlying matrix that is either precisely, or "close to," one which is positive definite, diagonally dominant, and similar to the Laplacian. Considerable progress has been made in identifying processes that have this common ground, in standardizing techniques best suited for optimizing their solution, and in extending these techniques to processes that have slight deviations from the standard.

At present, published material that has shown the most dramatic success in providing rapid convergence is limited to physical problems related to the incompressible Navier-Stokes equations or the irrotational forms of the Euler equations (potential or Cauchy-Riemann formulations). It is hoped that this publication will provide further knowledge and information that can be applied to the solution of compressible Navier-Stokes equations.

Harvard Lomax

CONFERENCE COMMITTEE

General Chairman:	Harvard Lomax
Program Committee:	Thomas H. Pulliam Marshal L. Merriam Dennis C. Jespersen
Administrative Chairman:	Walter A. Reinhardt
Publications Chairman:	Mamoru Inouye

CONTENTS

Page

MULTILEVEL TECHNIQUES FOR NONELLIPTIC PROBLEMS*

Dennis C. Jespersen

Department of Mathematics, Oregon State University

SUMMARY

Multigrid and multilevel methods have gained much attention lately and show great promise for the solution of elliptic problems. This paper sets up a framework for analyzing these methods with a view to extending their applicability to nonelliptic problems. A simple nonelliptic problem is given, and it is shown how a multilevel technique can be used for its solution. Emphasis is on "smoothness" properties of eigenvectors and attention is drawn to the possibility of "conditioning" the eigensystem so that eigenvectors have the desired smoothness properties.

1. INTRODUCTION

The purpose of this paper is to investigate the applicability of multigrid and multilevel methods for the numerical solution of partial differential equations that are not of the elliptic type. First an exposition of one means of analyzing "classical" multigrid techniques will be presented and then extended to deal with some simple nonelliptic problems. Throughout, the emphasis will be on analyzing and understanding how the various components of a successful multilevel process fit together, and not on the solution of practical problems. In particular, all the model problems presented here will be set in one space dimension. The analysis will concentrate on linear problems, though some remarks will be made on nonlinear problems; a thorough understanding of the linear case is an essential prerequisite for tackling nonlinear problems. The focus of the analysis will be on eigenvalues and eigenvectors.

A brief description of a standard multigrid procedure is as follows. We are to approximate the solution of some partial differential equation. Discretize the problem on a grid Γ^1 using (say) finite differences, giving a (large) linear system

$$A_b u_b = f_b \tag{1}$$

(Here the subscript b stands for "basic.") Some preconditioning procedure may be applied to this system, for example premultiplication of both sides of (1) by a matrix C, giving the system

$$A^1 u^1 = f^1 \tag{2}$$

where $A^1 := C\,A_b$, $f^1 := C\,f_b$, and $u^1 := u_b$. On the grid Γ^1, let $u_0{}^1$ be some initial guess at the solution $u_*{}^1$. Perform a few steps of some relaxation process, say

$$u_0{}^1 \rightarrow u_n{}^1 := G(u_0{}^1, f^1) \tag{3}$$

*Funds for the support of this study have been allocated by the Ames Research Center, NASA, Moffett Field, California, under Interchange No. NCA2-OR586-001.

where G denotes the relaxation process. Let r^1, the residual, be defined by $r^1 := f^1 - A^1 u_n^1$. Now, (1) is equivalent to

$$A^1 e^1 = r^1 \tag{4}$$

for the solution u_*^1 is then given by $u_*^1 = u_n^1 + e^1$. In order to solve (4), somehow transfer the problem to a coarser grid Γ^2, solve

$$A^2 e^2 = r^2 \tag{5}$$

on the coarser grid, transfer e^2 back to the fine grid, and replace u_n^1 by u_n^1 plus the transferred e^2. These transfers can be formalized by denoting by R^1 and $\tilde{R}^1$ "restriction" operators from grid functions on Γ^1 to grid functions on Γ^2, and I^1 and $\tilde{I}^1$ "interpolation" operators from grid functions on Γ^2 to grid functions on Γ^1. The transfer operators are then used to define $A^2 := \tilde{R}^1 A^1 \tilde{I}^1$ and $r^2 := R^1 r^1$; the problem on the coarser grid is then $A^2 e^2 = r^2$, and e^1 is defined as $e^1 := I^1 e^2$ once e^2 has been found. In a true multilevel process e^2 would be approximated by using still coarser grids, but the basic ideas can be seen in the analysis of the two-level process. The keys to constructing a successful process are the relaxation process G, the restriction operators R^1 and $\tilde{R}^1$, and the interpolation operators I^1 and $\tilde{I}^1$.

The framework of the analysis is similar to that of Brandt (1977), Hackbusch (1978), McCormick (1979), Wesseling (1980), and Frederickson (1975). The operator here called "restriction" (from functions on the fine grid to functions on the coarse grid) is called an "averaging operator" by McCormick, an "interpolation operator" by Brandt, a "collection operator" by Frederickson, and a "restriction operator" by Hackbusch and Wesseling. The operator here called "interpolation" is called "interpolation" by McCormick and Brandt, and is called a "prolongation operator" by Hackbusch and Wesseling. Previous authors have taken $R^1 = \tilde{R}^1$, $I^1 = \tilde{I}^1$, and either $R^1 = (I^1)^T$ or $R^1 = (\text{constant})*(I^1)^T$.

It should be noted that the difference between the computed solution and the exact solution, that is, the error, is not the focus of this work. It will always be assumed that an appropriate discretization of the partial differential equation has been derived and a discrete set of equations obtained. The job then is to solve the discrete set of equations very efficiently.

SYMBOLS

$B(a,b,c)$	tridiagonal matrix with b on main diagonal, a on subdiagonal, and c on superdiagonal
e	error (exact solution of discrete equations minus some approximation)
G	relaxation operator
I^i	interpolation operator from grid $i + 1$ to grid i
P	permutation matrix (a matrix whose entries are either 0 or 1, and such that each row and column has exactly one 1)
R^i	restriction operator from grid i to grid $i + 1$

r residual (f - Au)

:= quantity on left is defined as quantity on the right

Superscript:

i relates to the ith grid (grid 1 = finest grid)

Subscript:

n nth step of an iteration process on a fixed grid

2. A RELAXATION PROCESS

In this section we will be working on a fixed grid Γ and will therefore omit the superscripts denoting the grid level. A basic component of a multilevel scheme is the relaxation process. The relaxation process we will consider is the simple and general Richardson process. Given the linear system

$$Au = f \tag{6}$$

and an initial guess u_0, one step of the Richardson process with step-size h is defined by

$$u_1 := u_0 + h(Au_0 - f) \tag{7}$$

The common relaxation processes, such as the Jacobi, Gauss-Seidel, SOR, ADI, and their block variants, can all be written as the Richardson process applied to equation (6) or to a preconditioned form of equation (6). For example, the Gauss-Seidel process is the Richardson procedure with $h = 1$ applied to

$$(L + D)^{-1}Au = (L + D)^{-1}f$$

where L and D are the lower triangular and diagonal parts of A, respectively.

A few elementary remarks about the algorithm given in equation (7) are in order. If equation (7) is iterated, that is,

$$u_{n+1} := u_n + h(Au_n - f) \quad \text{for} \quad n \geq 0$$

it is a standard result that u_n converges to the exact solution u_* of equation (6) for any initial guess u_0, if and only if the spectral radius of the iteration matrix $I + hA$ is less than 1. Suppose now we allow the step sizes to vary, that is,

$$u_{n+1} := u_n + h_n(Au_n - f) \quad \text{for} \quad n \geq 0 \tag{8}$$

If A has a complete set of eigenvectors $\{v_m\}$, say $Av_m = \lambda_m v_m$, then the error $e_n := u_* - u_n$ satisfies $e_{n+1} = (I + h_n A)e_n$ and so, if $e_0 = \sum_m \alpha_m v_m$, then

$$e_n = \sum_m \left(\prod_{j=0}^{n-1} (1 + h_j \lambda_m) \right) \alpha_m v_m \tag{9}$$

From equation (9) it is evident that the error can (in theory) be reduced to zero in a finite number of steps by choosing a sequence of step sizes such that each h is -1 divided by an eigenvalue of A. This is usually quite impractical. We might consider trying to choose the step sizes to be approximately -1 divided by an eigenvalue of A. This is more practical; indeed, if $h_0 \simeq -1/\lambda_m$, then the component of e_1 in the direction of v_m has been greatly reduced. Pursuing this idea further leads one to study the possibility of choosing a set of step sizes $\{h_j, 0 \le j \le n-1\}$ for some given n such that the polynomial

$$p(z) := \prod_{j=0}^{n-1} (1 + h_j z)$$

has minimal modulus for z in some set in the complex plane that contains the negative reciprocals of the eigenvalues of A. This leads to the study of Chebyshev iterative methods, a topic that has been explored by many authors (for recent work in this area see Manteuffel, 1977; McDonald, 1980). Instead of carrying out a full Chebyshev process, the plan here is to use the nonstationary Richardson process to reduce the components of the error in some directions v_m and then to proceed with another idea.

It will be important that no error component be magnified by the Richardson process. From equation (9), we see that this requires $|1 + h\lambda_m| \le 1$ for all eigenvalues λ_m of A, so that h is in the intersection of the disks in the complex plane with centers $-1/\lambda_m$ and radii $|1/\lambda_m|$. Alternatively, given h we ask that all eigenvalues of A lie in the disk with center $-1/h$ and radius $|1/h|$. We might call this disk the stability region for step size h (see fig. 1).

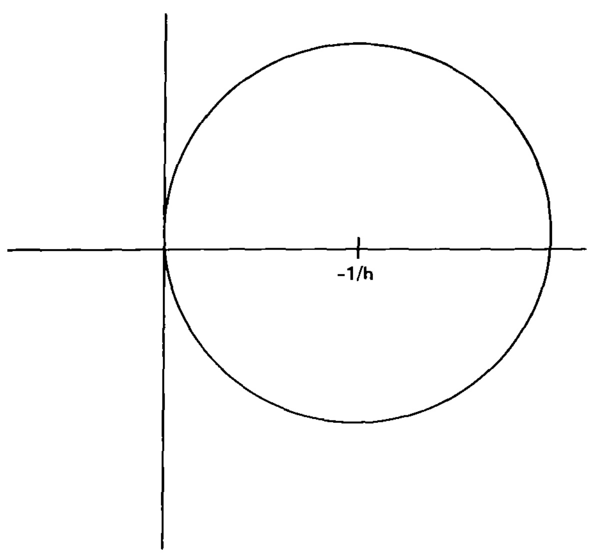


Figure 1.- Stability region for step-size h.

To have a large stability region (and thereby have the stability region contain all the eigenvalues of A, if the eigenvalues of A are widely separated), we see that h will have to be small. If h is small, the eigenvectors that are substantially diminished by one Richardson sweep with step size h are the eigenvectors associated with eigenvalues of modulus $|1/h|$, that is, eigenvalues of large modulus. Eigenvectors associated with eigenvalues of small modulus are hardly diminished by a Richardson step with small $|h|$. Loosely speaking, we might say that large eigenvalues are easy to eliminate and that small eigenvalues are difficult to eliminate (stably). The first part of the overall process will be the diminishing of (the eigenvectors associated with) the eigenvalues of large modulus by one or more Richardson steps with appropriately chosen step sizes.

If the original partial differential equation is not self-adjoint, the matrix A will probably have nonreal eigenvalues. To annihilate (nearly) the error component in the direction of an eigenvector associated with a nonreal eigenvalue, the step size h would have to be complex. In the interests of avoiding complex arithmetic, however, we note that, if A is a real matrix (as it is in most applications), then its eigenvalues come in complex conjugate pairs.

This leads one to consider a variant of the Richardson process in which a step with complex h is immediately followed by a step with the complex conjugate $\bar{h}$. The equations are thus

$$u_{1/2} := u_0 + h(Au_0 - f) \tag{10a}$$

$$u_1 := u_{1/2} + \bar{h}(Au_{1/2} - f) \tag{10b}$$

Substituting equation (10b) into (10a) gives

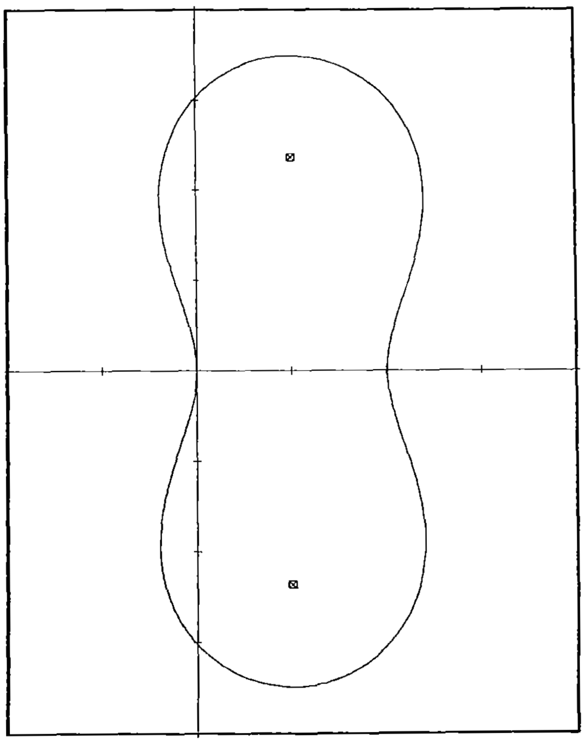
$$u_1 = u_0 + 2(\mathrm{Re}\ h)(Au_0 - f) + |h|^2 A(Au_0 - f) \tag{11}$$

where $\mathrm{Re}\ h$ denotes the real part of h. One notes that equation (11) can be carried out in real arithmetic and still annihilate the error components in the directions of the complex eigenvectors associated with $\lambda := -1/h$ and $\bar{\lambda}$. If h is a good approximation to $-1/\lambda$ for some eigenvalue λ, then the error component in the direction of the eigenvectors associated with λ and $\bar{\lambda}$ will be substantially reduced (reduced by a factor of $|1 + \lambda h|^2$).

It is again important to inquire into the stability properties of the iteration. Given a (complex) step size h, the set of eigenvectors that are not amplified by equation (11) is the set of eigenvectors belonging to eigenvalues λ such that

$$|1 + \lambda h|\ |1 + \lambda\bar{h}| \leq 1, \text{ that is, } |\lambda + 1/h|\ |\lambda + 1/\bar{h}| < 1/|h|^2$$

This defines a region in the complex plane whose boundary is called an oval of Cassini; in the special case when $\mathrm{Re}\ h = 0$, this reduces to a lemniscate (two-leaved rose; fig. 2).



FOCI = 1.0034 ± 2.36171

Figure 2.- Oval of Cassini.

Again we see that to have a large stability region, $|h|$ will have to be small, so that the (eigenvectors associated with) eigenvalues of large modulus will be easy to diminish, and the (eigenvectors associated with) eigenvalues of small modulus will be difficult to diminish. Also note that for a small $|h|$ the Richardson step $u_{n+1} = u_n + h(Au_n - f)$ is the explicit Euler method applied to the time integration of $du/dt = Au - f$. Time-like methods will reach steady-state solutions, but will require a large amount of computation to do so. If we are interested in only the steady-state solution, we are free to use methods that are not time accurate.

To conclude this section, the generality of the Richardson process should again be emphasized. As noted above, the usual iterative methods can be written in the form of a Richardson process. The considerations of eigenvector decomposition lead us to use the following as a rule of thumb: The large eigenvalues are easy to eliminate. The difficult problem is to stably and rapidly eliminate small eigenvalues. We will attempt to do this by combining a Richardson process with a multilevel procedure.

3. A MODEL ELLIPTIC PROBLEM

The goal of this paper is the study of multilevel methods as applied to nonelliptic problems. Nevertheless, in this section we will look at a model one-dimensional elliptic problem and show how the analysis proceeds. The hope is that looking at this familiar problem will give confidence in the analytical techniques when they are used to study nonelliptic problems. Consider then the simple problem (which was also used as a model problem by Hackbusch, 1978):

$$\left.\begin{aligned} u''(x) &= f(x), \quad 0 \le x \le 1 \\ u(0) &= 0, \quad u(1) = 0 \end{aligned}\right\} \tag{12}$$

We discretize equation (12) on a uniform grid with points $x_j := j\Delta x$, $0 \le j \le M + 1$ where M is an odd integer and $\Delta x := 1/(M + 1)$. The standard second-order centered-difference scheme for (12) gives the linear system $A^1u^1 = f^1$, where A^1 is the M by M tridiagonal matrix $B(1,-2,1)$, and f is the vector with entries $f(x_j)(\Delta x)^2, 1 \le j \le M$. (Recall that $B(1,-2,1)$ denotes the tridiagonal matrix with -2 on the main diagonal and 1's on the sub- and superdiagonals.) The eigenvalues of A^1 are

$$\lambda_m := -2\{1 - \cos[m\pi/(M + 1)]\} = -4 \sin^2[m\pi/(2M + 2)], \ 1 \le m \le M \tag{13}$$

with corresponding eigenvectors v_m where

$$(v_m)_j = \sin[jm\pi/(M + 1)], \quad 1 \le j,m \le M \tag{14}$$

Note that the eigenvectors associated with eigenvalues of large modulus are highly oscillatory, and the eigenvectors associated with eigenvalues of small modulus are "smooth." Thus, given some initial guess, a few Richardson sweeps with appropriate step sizes will substantially reduce the "high-frequency" component of the error. This idea seems to be one of the motivating ideas for the multigrid method for elliptic equations; high-frequency components of the error correspond to eigenvalues of large modulus and are easy to diminish by an appropriate relaxation technique which is thought of as "smoothing." (We will see that use of the term "smoothing" may not be appropriate for nonelliptic problems; the term "relaxation" will be used in this paper.)

After a few relaxation steps the residual $r^1 = f^1 - A^1G(u_0{}^1,f^1)$ should consist mostly of "low frequencies" and be well representable on a coarser grid. Letting Γ^1 denote the initial fine grid and Γ^2 denote the grid with mesh spacing $2\Delta x$, a restriction operator from functions on the fine grid to functions on the coarse grid is representable as a mapping from R^M to $R^{(M-1)/2}$, Euclidean M-space to Euclidean $(M - 1)/2$-space.

For example, if we simply transfer the even-subscripted components of the residual r^1 to the grid Γ^2, then R^1 is the $(M - 1)/2$ by M matrix

$$R^1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & . & . & . \\ 0 & 0 & 0 & 1 & 0 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & & & \\ . & & & & & & & & \\ . & & & & & & & & \\ . & & & & & & & & \end{bmatrix} \tag{15}$$

Another possibility is a linear averaging procedure, which would give

$$R^1 = \begin{bmatrix} 1/4 & 1/2 & 1/4 & 0 & 0 & 0 & 0 & . & . & . \\ 0 & 0 & 1/4 & 1/2 & 1/4 & 0 & 0 & & & \\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 1/4 & & & \\ . & & & & & & & & & \\ . & & & & & & & & & \\ . & & & & & & & & & \end{bmatrix} \tag{16}$$

The interpolation operator I^1 from functions on the coarse grid to functions on the fine grid is a mapping from $R^{(M-1)/2}$ to R^M. One possibility is linear interpolation from the even-numbered grid points to the odd-numbered grid points. In this case, the matrix representation of I^1 would be

$$I^1 = \begin{bmatrix} 1/2 & 0 & 0 & . & . & . \\ 1 & 0 & 0 & & & \\ 1/2 & 1/2 & 0 & & & \\ 0 & 1 & 0 & & & \\ 0 & 1/2 & 1/2 & & & \\ 0 & 0 & 1 & & & \\ 0 & 0 & 1/2 & & & \\ . & & & & & \\ . & & & & & \\ . & & & & & \end{bmatrix} \tag{17}$$

Another possibility is an implicit cubic polynomial interpolation, defined as follows. Consider a set of data $\{(x_j,y_j):1 \le j \le M\}$, where we assume y_j is known for j even and unknown for j odd (and $x_j := j\Delta x$). For $3 \le k \le M - 2$ and k odd, let $p_k(t)$ be the polynomial of degree at most 3 which satisfies $p_k(x_j) = y_j$ for

$j = k-2, k-1, k+1, k+2$. Define $y_k := p_k(x_k)$. Thus y_k is linearly related to the known values y_{k-1}, y_{k+1} and the unknown values y_{k-2}, y_{k+2}. For $k = 1$, let $p_1(t)$ be the polynomial of degree at most 3 which satisfies $p_1(0) = 0$, $p_1(x_2) = y_2$, $p_1(x_3) = y_3$, $p_1(x_4) = y_4$; define $y_1 := p_1(x_1)$. Define y_{M-1} in a similar manner. (Note that the definition of y_1 would have to be modified if the left-hand boundary condition were of the Neumann type; if the boundary condition for the differential equation were $u'(0) = 0$ then p_1 would be required to satisfy $p_1'(0) = 0$ instead of $p_1(0) = 0$.) The result is a linear system

$$By_{odd} = Cy_{even} \tag{18}$$

where $y_{odd} := (y_1, y_3, \ldots, y_M)^T$, $y_{even} := (y_2, y_4, \ldots, y_{M-1})^T$, B is the $(M+1)/2$ by $(M+1)/2$ tridiagonal matrix:

$$B = \begin{bmatrix} 1 & 1 & & & & \\ 1 & 6 & 1 & & & \\ & 1 & 6 & 1 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & 1 & 6 & 1 \\ & & & & & 1 & 1 \end{bmatrix} \tag{19}$$

and C is the $(M+1)/2$ by $(M-1)/2$ tridiagonal matrix:

$$C = \begin{bmatrix} 3/2 & 1/4 & & & & \\ 4 & 4 & & & & \\ & 4 & 4 & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & 4 & 4 \\ & & & & 1/4 & 3/2 \end{bmatrix} \tag{20}$$

For the cubic interpolation, one would then define

$$I^1 := P^T \left[\frac{I}{B^{-1}C}\right]$$

where P is the "even-odd" permutation matrix, which satisfies

$$P(v_1, \ldots, v_M)^T = (v_2, v_4, \ldots, v_{M-1}, v_1, v_3, \ldots, v_M)^T$$

and I is the identity matrix of order $(M-1)/2$.

In either case, the coarse-grid problem is defined to be $R^1A^1I^1e^2 = R^1r^1$, and then e^1 is approximated by I^1e^2. One can calculate that with A^1 as above, R^1 given by equation (15), and I^1 given by (17), the matrix $R^1A^1I^1$ is the $(M - 1)/2$ by $(M - 1)/2$ matrix:

$$R^1A^1I^1 = \frac{1}{2}B(-1,2,-1) \tag{21}$$

This is (up to a scaling factor) the difference operator that would have resulted from discretizing the original differential operator on the coarser grid Γ^2.

One complete step of the two-level process is defined by $u_1{}^1 := I^1e^2 + G^1u_0{}^1$. This can be written as a matrix iterative process as follows. Let P be the even-odd permutation matrix defined above. The problem

$$A^1e^1 = r^1$$

is equivalent to the problem

$$(PA^1P^T)(Pe^1) = Pr^1 \tag{22}$$

which is, in block form,

$$\begin{bmatrix} A_1{}^1 & A_2{}^1 \\ A_3{}^1 & A_4{}^1 \end{bmatrix}\begin{bmatrix} e_a{}^1 \\ e_b{}^1 \end{bmatrix} = \begin{bmatrix} r_a{}^1 \\ r_b{}^1 \end{bmatrix} \tag{23}$$

where subscript a refers to the even unknowns and the subscript b refers to the odd unknowns.

The first block equation of equation (23) reads

$$A_1{}^1e_a{}^1 + A_2{}^1e_b{}^1 = r_a{}^1 \tag{24}$$

If we make the approximation $e_b{}^1 \simeq I^1_{a,b}e_a{}^1$ for some operator $I^1_{a,b}$, then (24) becomes

$$(A_1{}^1 + A_2{}^1I^1_{a,b})e_a{}^1 = r_a{}^1 \tag{25}$$

which is equivalent to $(R^1A^1I^1)e^2 = R^1r^1$ in case the operator R^1 is given by (15); the matrix $I^1_{a,b}$ can be given by (for example)

$$I^1_{a,b} = \begin{bmatrix} 1/2 & & & & \\ 1/2 & 1/2 & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & 1/2 & 1/2 \\ & & & & 1/2 \end{bmatrix} \tag{26}$$

which corresponds to linear interpolation as in (17), or by

$$I^1_{a,b} = B^{-1}C \tag{27}$$

where B and C are the matrices from the implicit cubic interpolation process defined above.

After solving (25), we have

$$\begin{aligned} e^1 = P^T\begin{bmatrix} e_a{}^1 \\ e_b{}^1 \end{bmatrix} &= P^T\begin{bmatrix} (A_1{}^1 + A_2{}^1 I^1_{a,b})^{-1} \\ I^1_{a,b}(A_1{}^1 + A_2{}^1 I^1_{a,b})^{-1} \end{bmatrix} r_a{}^1 \\ &= P^T\begin{bmatrix} (A_1{}^1 + A_2{}^1 I^1_{a,b})^{-1} & 0 \\ I^1_{a,b}(A_1{}^1 + A_2{}^1 I^1_{a,b})^{-1} & 0 \end{bmatrix} Pr^1 \\ &=: P^T S P r^1 \end{aligned} \tag{28}$$

and thus

$$\begin{aligned} u_1{}^1 &= P^T SP(f^1 - A^1 G^1 u_0{}^1) + G^1 u_0{}^1 \\ &= (I - P^T SPA^1)G^1 u_0{}^1 + P^T SPf^1 \end{aligned} \tag{29}$$

One step of the iteration process is thus one step of a stationary iterative process with iteration matrix

$$T := (I - P^T SPA^1)G^1 \tag{30}$$

The rate of convergence of the process is controlled by the spectral radius of the matrix T.

The operators I^1 and R^1 can be identified from (28) as

$$R^1 = [I|0]P \tag{31}$$

and

$$I^1 = P^T\left[\frac{I}{I^1_{a,b}}\right] \tag{32}$$

Other authors (McCormick, 1977; Wesseling, 1980) have suggested that the condition $I^1 = (R^1)^T$ or $I^1 = (\text{constant})*(R^1)^T$ be enforced; an obvious way to do this is to replace (31) by

$$R^1 = \frac{1}{2}\,[I|(I^1_{a,b})^T]P \tag{33}$$

Numerical tests with both possibilities are reported below.

Some numerical experiments were carried out in which the eigenvalues of the overall iteration matrix T were computed for different numbers of grid points, different interpolation and restriction operators, and varying relaxation procedures. The relaxation procedure G^1 was of the form

$$G^1 = \prod_{j=1}^{k} (I + h_j A^1)$$

(i.e., the relaxation procedure consisted of k Richardson sweeps), where $k \in \{0,1,2,3\}$ and $h \in \{1/2,1/3,1/4\}$ (note that the stability condition of section 2, $|1 + \lambda h| \leq 1$, becomes in this case $0 \leq h \leq 2/\{4 \sin^2[m\pi/(2M + 2)]\}$, $1 \leq m \leq M$, so 1/2 is effectively an upper bound for h. Some results for the cases $M \in \{7,11\}$, I^1 given by (32) and (26) ("linear interpolation") or by (32) and (27) ("cubic interpolation"), and R^1 given by (15) or by (33). A selection of results from these computations is given in table 1.

From table 1 we see that the total process will not converge if $k = 0$, that is, if no relaxation sweeps are used. (This is also clear from the definition of T and from the fact that S is a rank-deficient matrix; see eq. (30).) With linear interpolation, the results with R given by (15) are identical with the results when R is given by (33), but we have no formal proof of this. From the output, there seems little point in using more than one relaxation sweep with linear interpolation. In two of the cases presented, the spectral radius of T is 0; this does not imply that the iterative process converges in one step, because, for these cases, it turns out that T has nontrivial Jordan blocks in its Jordan canonical form, that is, n by n blocks of the form

$$\begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \cdot & \cdot & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}$$

with $n > 1$. In these cases, the total process would converge in n steps.

With cubic interpolation it appears there can be a substantial gain by performing more than one relaxation sweep on the fine level. Using the R^1 from (33) is not as favorable as using R^1 from (15). A full investigation of this has yet to be undertaken.

Although not shown in table 1, computations with higher values of M revealed that the spectral radius of T was virtually independent of the number of grid points. This is an encouraging sign, and is the typical situation in this classical multigrid situation (Brandt, 1977; Hackbusch, 1978). Also, the idea of eigensystem mixing (Lomax, 1981, unpublished notes) may serve to further decrease the spectral radius of T, leading to even faster convergence of the overall process. Finally, a similar analysis can be carried out in the case when the differential equation has a boundary condition of the Neumann type; appropriate modifications must be made in the definition of the interpolation operator at a Neumann boundary.

TABLE 1.- SPECTRAL RADIUS OF TOTAL PROCESS: MODEL ELLIPTIC PROBLEM

Linear interpolation (The results with R^1 given by (33) are identical with those when R^1 is given by (15).)

M	k	h	Spectral radius of T
7	0	-	1.0
	1	1/2	0
	1	1/4	.5
	2	1/2, 1/2	.5
	2	1/4, 1/4	.46
	2	1/2, 1/4	.43
	3	1/4, 1/4, 1/4	.45
	3	1/2, 1/2, 1/2	0
	3	1/2, 1/3, 1/4	.43
11	0	-	1.0
	1	1/4	.5
	1	1/2	0
	3	1/2, 1/3, 1/4	.47
	3	1/4, 1/4, 1/4	.47

(In the cases when T has spectral radius 0, T has nonlinear elementary divisors, i.e., the Jordan canonical form of T is not diagonal.)

Cubic interpolation

			Spectral radius of T	
M	k	h	R^1 from (15)	R^1 from (33)
7	0	-	1.0	1.0
	1	1/2	1.0	.91
	1	1/4	.5	.5
	2	1/4, 1/4	.25	.46
	2	1/2, 1/4	.25	.44
	3	1/4, 1/4, 1/4	.13	.44
	3	1/2, 1/3, 1/4	.08	.42
11	0	-	1.0	1.0
	1	1/2	1.0	.96
	1	1/4	.5	.5
	2	1/4, 1/4	.25	.48
	2	1/2, 1/4	.25	.47
	3	1/4, 1/4, 1/4	.13	.47
	3	1/2, 1/3, 1/4	.08	.46

4. A MODEL NONELLIPTIC PROBLEM

The preceding section introduced some ideas and set up a framework for analysis of a multilevel method for a model elliptic problem. In this section, we will begin to investigate the applicability of multilevel methods to nonelliptic problems. The model problem to be used is the steady-state version of $u_t + u_x = f$,

$$\begin{aligned} u_x &= f , \quad 0 < x < 1 \\ u(0) &= g_0 \end{aligned} \tag{34}$$

Introduce grid points $x_j := j\Delta x = j/(M + 1)$, $0 \le j \le M + 1$, M odd, and consider second-order centered differences (leave aside for now the fact that centered differences are probably inappropriate for this particular problem; we have in mind systems of equations describing subsonic flow for which centered differencing may be very appropriate). The finite-difference equations are

$$u_{j+1} - u_{j-1} = 2\Delta x f(x_j) , \quad 1 \le j \le M - 1 \tag{35}$$

(where $u_0 := g_0$). At the right-hand boundary let us use a linear extrapolation $u_{M+1} := 2u_M - u_{M-1}$ to derive from $(u_{M+1} - u_{M-1})/2\Delta x = f(x_M)$ the equation

$$u_M - u_{M-1} = \Delta x f(x_M) \tag{36}$$

We get then the linear system $A_b u_b = f_b$, where

$$A_b = \begin{bmatrix} 0 & 1 & & & & & \\ -1 & 0 & 1 & & & & \\ & -1 & 0 & 1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & -1 & 0 & 1 \\ & & & & & -1 & 1 \end{bmatrix} \tag{37}$$

and $f_b := 2\Delta x[f(x_1), \ldots, f(x_{M-1}), f(x_M)/2]^T$.

Most of the eigenvalues of A are complex; it is easy to show that all the eigenvalues lie in the half-plane $\mathrm{Re}(z) > 0$. Computations with $M = 15$ revealed that, for this case, the eigenvalues lie on a curve from about $0.0026 \pm 1.96i$ to approximately 0.195. One possible way of treating the linear system $A_b u_b = f_b$ is to premultiply by $(-A_b{}^T)$, giving the system $(-A_b{}^T A_b)u_b = -A_b{}^T f$, with a coefficient matrix that is symmetric and negative definite, thus analogous to the matrices that arise when discretizing elliptic equations (see Lomax et al., 1981). A possible problem with this idea is that the condition number of the matrix of the new linear system is the square of the condition number of the original matrix, which may lead to slow convergence of an iterative technique. In this paper, we wish to stay in the spirit of nonelliptic problems, however, and so we do not consider this possibility further.

Another possibility for treating the linear system is to use some relaxation method, such as the complex Richardson technique of section 2, to approximately eliminate the components of the error along the directions of the eigenvectors associated with eigenvalues of large modulus, and then to transfer the error equation to a coarser grid. For this idea to succeed, it is presumably the case that the eigenvectors associated with eigenvalues of small modulus should not fluctuate rapidly on the fine grid, so that they are well representable on the coarse grid. Such is not the case for the matrix A_b; indeed, A_b can be viewed as a perturbation of the matrix $B(-1,0,1)$, which has eigenvalues

$$\lambda_k = 2i \cos \pi k/(M+1) \ , \quad 1 \le k \le M$$

and eigenvectors $x^{(k)}$ with

$$x_j^{(k)} = i^j \sin j\pi k/(M+1) \ , \quad 1 \le j,k \le M$$

all of which oscillate rapidly on the fine grid. Some of the eigenvectors of A_b are shown in figure 3; in the figure, the horizontal axis is j, and the vertical axis is v_j (where $A_b v = \lambda v$, $v = (v_1, \ . \ . \ ., v_M)^T$).

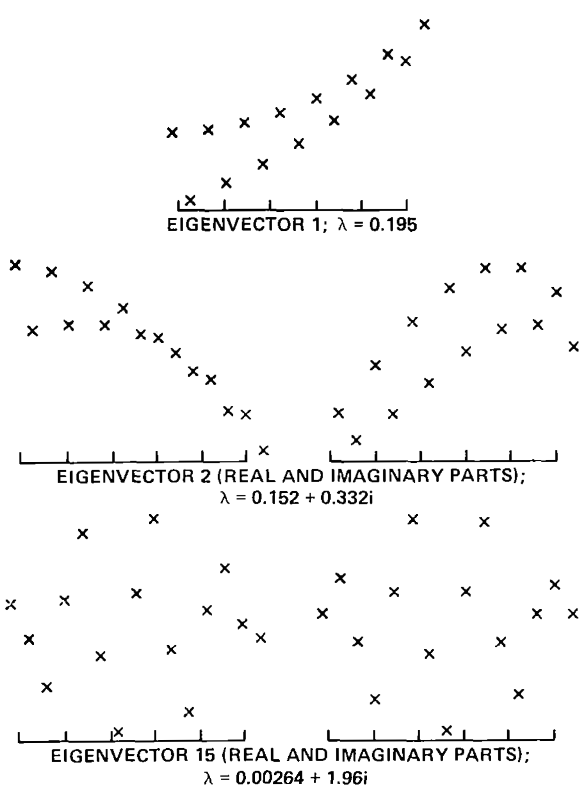


Figure 3.- Eigenvectors of A_b (see (37)); $M = 15$.

Although the eigenvectors oscillate, there is evidently a regularity about them which we can try to exploit. Let us try to find a simple transformation that will change the eigenvectors associated with small eigenvalues from rapidly oscillating to slowly oscillating. First, let P_2 be the "even-odd reversed" permutation which is defined by

$$P_2(v_1, \ . \ . \ ., v_M)^T$$

$$= (v_2, v_4, \ . \ . \ ., v_{M-1}, v_M, v_{M-2}, \ . \ . \ ., v_1)^T$$

The result of applying P to the eigenvectors of figure 3 is shown in figure 4; the eigenvectors associated with eigenvalues of small modulus have become much smoother, though there is still a pronounced kink. To remove this kink, let F be the matrix that satisfies

$$(Fv)_j = \begin{cases} v_j & \text{if} \quad j \le (M+1)/2 \\ 2v_{(M+1)/2} - v_j & \text{if} \quad j > (M+1)/2 \end{cases} \tag{38}$$

For example, if $M = 7$, then F is the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & -1 \end{bmatrix}$$

(Note that $F^{-1} = F$.) The result of applying F to the vectors of figure 4 is shown in figure 5. The eigenvectors associated with eigenvalues of small modulus are now slowly varying, and the eigenvectors associated with eigenvalues of large modulus (which we do not care about anyway, since their contribution to the error is destined to be removed by a few Richardson sweeps) are still rapidly varying. We might call F a "reflection" matrix.

Now, the basic linear system $A_bu_b = f_b$ is equivalent to

$$(FP_2A_bP_2^TF^{-1})(FP_2u_b) = FP_2f_b \tag{39}$$

or

$$A^1u^1 = f^1 \tag{40}$$

where $A^1 := FP_2A_bP_2^TF^{-1} = FP_2A_bP_2^TF$ (since $F^{-1} = F$). For the case $M = 7$, it turns out that A^1 is given by

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -3 & 2 & 0 & 0 & 0 \\ 1 & -1 & -2 & 2 & 0 & 0 & 0 \\ -1 & 0 & -2 & 2 & 0 & 0 & 0 \end{bmatrix} \tag{41}$$

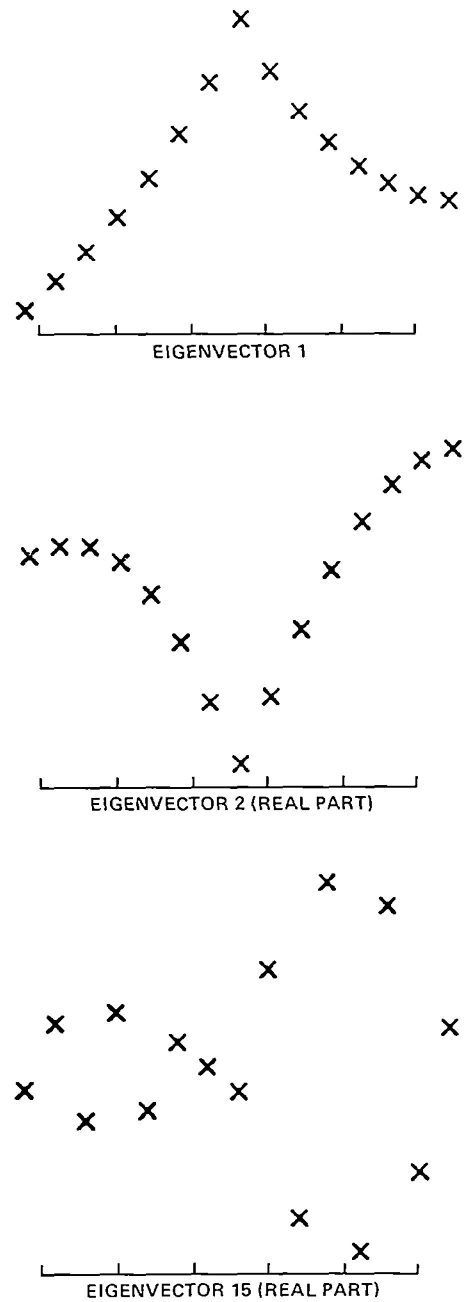


Figure 4.- Eigenvectors of A_b after permutation.

The matrix A^1 has the same eigenvalues as the matrix A_b, and the eigenvectors of A^1 are the eigenvectors of A_b premultiplied by FP_2; hence the eigenvectors of A^1 associated with eigenvalues of small modulus are slowly varying and are thus good candidates for accurate transfer to a coarser grid. This whole process might be described as a "conditioning" of the eigensystem of A_b, where the goal of the conditioning is to transform the eigenvectors associated with eigenvalues of small modulus to vectors that are smooth when considered as grid functions.

Some numerical experiments were performed along the lines of those in section 3. Beginning with the matrix A^1 of (40) the matrix $T := (I - P^T SPA^1)G^1$ was computed for $M \varepsilon \{7,11\}$, linear and cubic interpolation and restriction operators, and varying relaxation procedures. The relaxation procedure (note that "smoothing" no longer seems appropriate) was taken to be of the form

$$G^1 = \prod_{j=1}^{k} [I + 2(\text{Re } h_j)A^1 + |h_j|^2 (A^1)^2]$$

(i.e., the complex Richardson technique was used), with $k \in \{0,1,2,3\}$. The interpolation and restriction processes at the "boundaries" treated the left-hand boundary as a Dirichlet boundary and the right-hand boundary as a Neumann boundary. This was done because the transformed eigenvectors have nearly zero slope at their right-hand ends and can be thought of as vanishing at their left-hand ends. Some of the results are shown in table 2.

From table 2 we see that when no relaxation sweeps are performed on the fine level, the process will not converge. In all other cases (when using the R^1 from (15)), a spectral radius of 0.54 or less was achievable. There seems to be no particular advantage in forcing the restriction operator to satisfy $R^1 = c(I^1)^T$. The three h values picked for the case $k = 3$ turned out to be not very good choices, as the spectral radius of the overall process was greater than for $k = 2$ and a different choice of h's. A better choice of h's would have made the spectral radius of the overall process for $k = 3$ less than that for $k = 2$. There seems to be little advantage in using cubic interpolation; the spectral radius of the overall process does not seem to be significantly reduced by the use of cubic interpolation.

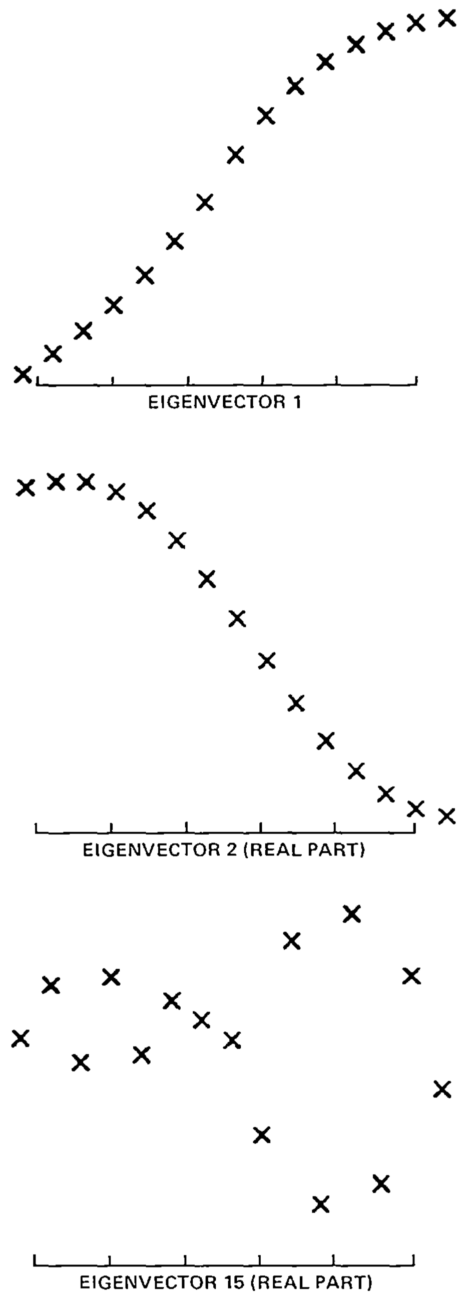


Figure 5.- Eigenvectors of A_b after permutation and reflection.

5. EXTENSIONS

In this section, we begin extending the ideas of the previous sections to actually solving some (simple) problems. The first problem to look at is the simple one-dimensional linear variable-coefficient problem:

TABLE 2.- SPECTRAL RADIUS OF TOTAL PROCESS: MODEL NONELLIPTIC PROBLEM

M	k	Re h	$\|h\|^2$	Spectral radius of T	
				R^1 from (15)	R^1 from (33)
Linear interpolation					
7	0	-	-	1	1
	1	0	0.25	.50	.53
	1	-0.05	.25	.50	.49
	2	0,-0.05	0.25,0.30	.25	.26
	2	-0.025,-0.075	.25, .50	.31	.47
	3	-0.0017,-0.0088,-0.031	0.268,0.342,0.532	.40	.57
11	0	-	-	1	1
	1	0	0.25	.50	.51
	1	-0.05	.25	.54	.51
	2	0.-0.05	0.25,0.30	.24	.26
	2	-0.025,-0.075	.25, .50	.24	.48
	3	-0.0017,-0.0088,-0.031	0.268,0.342,0.532	.40	.57
Cubic interpolation					
7	0	-	-	1	1
	1	0	0.25	.48	.43
	1	-0.05	.25	.54	.57
	2	0,-0.05	0.25,0.30	.26	.34
	2	-0.025,-0.075	.25, .50	.42	.70
	3	-0.0017,-0.0088,-0.031	0.268,0.342,0.532	.35	.83
11	0	-	-	1	1
	1	0	0.25	.50	.46
	1	-0.05	.25	.48	.53
	2	0,-0.05	0.25,0.30	.24	.32
	2	-0.025,-0.075	.25, .50	.27	.59
	3	-0.0017,-0.0088,-0.031	0.268,0.342,0.532	.25	.63

$$\begin{aligned} [c(x)u]_x &= f(x) , \quad 0 < x < 1 \\ u(0) &= g_0 \end{aligned} \tag{42}$$

where we assume $c(x) \geq c_0 > 0$. This is the steady-state version of the advection equation $u_t + [c(x)u]_x = f(x)$. Suppose (42) is discretized with second-order centered differences. On the right-hand boundary, we again use linear extrapolation, as in (36). The system of equations that arises is $A_b' u_b = f_b'$, where

$$A_b' = \begin{bmatrix} 0 & c_2 & & & & & \\ -c_1 & 0 & c_3 & & & & \\ 0 & -c_2 & 0 & c_4 & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & -c_{M-2} & 0 & c_M \\ & & & & -c_{M-1} & c_M \end{bmatrix} \tag{43}$$

Here $c_j := c(j\Delta x)$ and

$$f_b' := 2\Delta x[f(x_1), \ldots, f(x_{M-1}), f(x_M)/2]^T + [c(0)g_0, 0, \ldots, 0]^T$$

In order for the investigations of the previous section to apply, it is probably necessary that whatever matrix we work with be "close" to the matrix A_b of section 4. In an attempt to satisfy this condition, premultiply the system $A_b'u_b = f_b$ by $C := \mathrm{Diag}(1/c_1, \ldots, 1/c_M)$ (where Diag denotes a diagonal matrix), giving a new equivalent system:

$$A_b u_b = f_b \qquad (A_b := CA_b', \; f_b := Cf_b') \tag{44}$$

Now the process to follow is the one outlined in previous sections: replace equation (44) by the equivalent system

$$(FP_2 A_b P_2{}^T F^{-1})(FP_2 u_b) = FP_2 f_b \quad \text{or} \quad A^1u^1 = f^1$$

Now let $u_0{}^1$ be an initial guess; perform one or more relaxation sweeps starting with $u_0{}^1$; form the residual; transfer the error equation to a coarser level; solve the error equation on the coarser level, either exactly or via further multilevel cycles; transfer the error on the coarser level back to the finer level; update the guess on the finer level; and repeat the whole process until some convergence criterion is satisfied.

This process was programmed and some results will be presented below. Two features of the whole process are important to note. First, in order to carry out the relaxation sweeps on the coarser level one must be able to form the matrix-vector product A^1u^1. Instead of explicitly forming the matrix A^1 and then computing A^1u^1, what one can do is form the matrix-vector product $FP_2CA_b'P_2{}^TFu^1$, where each matrix-vector product (starting from the right) is easily carried out. For the relaxation sweeps the complex Richardson technique can be used; since the matrix CA_b' is in some sense close to the matrix A_b of section 4, the step sizes used can be based on our knowledge of the eigenvalues of that matrix.

Secondly, to perform the process beginning on the coarser level, one must be able to form $A^2(= R^1A^1I^1)$ times a vector; this can be (inefficiently) done by using code to compute A^1 times a vector along with code to compute R^1 times a vector and I^1 times a vector. This is very inefficient, because each matrix-vector product on a coarse level then requires computing a matrix-vector product on the finest level in addition to the work of restriction and interpolation. Since this was intended to be a pilot study, such inefficiency was judged acceptable. It is important to be able to form the matrix-vector products on the coarser levels efficiently. With regard to this, it is interesting and encouraging that with $M = 7$, linear interpolation given by (26), and $c(x) = 1$, the matrix A^2 turns out to be

$$\frac{1}{2}\begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ -3 & 2 & 0 \end{bmatrix}$$

which becomes, if it is "unreflected" and "unpermuted,"

$$\frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix}$$

which is (up to a scaling factor) the matrix one would get by writing a difference scheme for the same differential equation on a three-point mesh. The same holds for larger values of M. Thus the computations with A^2 could be performed as if one were working on a mesh with $(M - 1)/2$ points. Even in the variable coefficient case it may turn out that the matrix $R^1A^1I^1$ can be sufficiently well approximated by a matrix that comes from a difference scheme on the coarse mesh that the overall process will still converge. This problem has not yet been investigated.

In any event, the process was programmed for some test problems; it worked quite well, with an average error reduction in the L_2-norm of about 0.5 to 0.6 per step (see table 3 for details).

To conclude this section, let us describe how the whole procedure could be applied to a nonlinear system of the form

$$\begin{aligned} [A(\underset{\sim}{u})\underset{\sim}{u}]_x &= \underset{\sim}{f}(x,\underset{\sim}{u}) \;, \quad 0 < x < 1 \\ \underset{\sim}{u}(0) &= \underset{\sim}{g} \end{aligned} \tag{45}$$

In brief, the idea is to apply Newton's method, solving the linear systems at each stage of Newton's method with a multilevel technique (this has been called the Newton-multigrid technique). Suppose the system is discretized using centered differences; a nonlinear system of equations of the form

$$\mathscr{F}(\underset{\sim}{u}) = 0 \;, \quad \underset{\sim}{u} = \begin{bmatrix} \underset{\sim}{u}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \underset{\sim}{u}_M \end{bmatrix} \;, \quad \mathscr{F}(\underset{\sim}{u}) = \begin{bmatrix} A(\underset{\sim}{u}_2)\underset{\sim}{u}_2 - A(\underset{\sim}{g})\underset{\sim}{g} - 2\Delta x \underset{\sim}{f}(x_1,\underset{\sim}{u}_1) \\ A(\underset{\sim}{u}_3)\underset{\sim}{u}_3 - A(\underset{\sim}{u}_1)\underset{\sim}{u}_1 - 2\Delta x \underset{\sim}{f}(x_2,\underset{\sim}{u}_2) \\ \cdot \\ \cdot \\ \cdot \\ A(\underset{\sim}{u}_M)\underset{\sim}{u}_M - A(\underset{\sim}{u}_{M-2})\underset{\sim}{u}_{M-2} - 2\Delta x \underset{\sim}{f}(x_{M-1},\underset{\sim}{u}_{M-1}) \\ A(\underset{\sim}{u}_M)\underset{\sim}{u}_M - A(\underset{\sim}{u}_{M-1})\underset{\sim}{u}_{M-1} - \Delta x \underset{\sim}{f}(x_M,\underset{\sim}{u}_M) \end{bmatrix} \tag{46}$$

arises. Newton's method is then

$$\underset{\sim}{u}_0 \text{ given;}$$

$$\left\{ \begin{aligned} D\mathscr{F}(\underset{\sim}{u}_n)\Delta\underset{\sim}{u}_n &= -\mathscr{F}(\underset{\sim}{u}_n) \\ \underset{\sim}{u}_{n+1} &:= \underset{\sim}{u}_n + \Delta\underset{\sim}{u}_n \quad \text{for} \quad n \ge 0 \end{aligned} \right. \tag{47}$$

where $D\mathscr{F}$ is the Jacobian matrix of $\mathscr{F}$. In this case, the Jacobian matrix will have the form

TABLE 3.- RESULTS FOR $[c(x)u]_x = f(x)$, $0 < x < 1$, $u(0) = g_0$

$c(x) = 1$, $f(x) = 0$, $g_0 = 0$

1. Two levels, 7 grid points on fine level, 1 Richardson sweep on fine level; average convergence rate 0.499.

2. Three levels, 15 grid points on fine level, 1 Richardson sweep on fine level.

No. of Richardson sweeps on level 2	Average convergence rate
1	0.591
2	.566
3	.557
5	.551
10	.549
20	.549

3. Six levels, 63 points on fine level, number of Richardson sweeps: 1,3,3,3,3 on levels 1(=fine),2,3,4,5 respectively, average convergence rate 0.557.

4. Four levels, 31 grid points on fine level, 1 Richardson sweep on fine level.

No. of sweeps on level 2	No. of sweeps on level 3	Average convergence rate
1	1	0.603
1	2	.604
1	3	.599
2	1	.574
2	2	.568
2	3	.566
3	1	.561
3	2	.558
3	3	.557

$c(x) = 1 + x$, $f(x) = 1$, $g_0 = 1$

Two levels, 7 grid points on fine level, 1 Richardson sweep on fine level; average convergence rate 0.519.

$c(x) = 1 + x$, $f(x) = 2 + 2x$, $g_0 = 1$

Two levels, 7 grid points on fine level, 1 Richardson sweep on fine level; average convergence rate 0.452.

All average convergence rates measured as $(\|e_{20}\|/\|e_1\|)^{1/20}$ where norms are ℓ_2 norms and e_1, e_{20} are the error vectors at steps 1 and 20, respectively. The parameters for the implicit complex Richardson sweeps on level k were $\mathrm{Re}(h) = -0.05\cdot 2^{k-1}$, $|h|^2 = 0.25\cdot 4^{k-1}$ (level 1 = finest level).

$$
D = \begin{bmatrix}
-2\Delta x \underset{\sim}{f}'_1 & J_2 & & & & \\
-J_1 & -2\Delta x \underset{\sim}{f}'_2 & J_3 & & & \\
 & \cdot & \cdot & \cdot & & \\
 & & \cdot & \cdot & \cdot & \\
 & & & -J_{M-2} & -2\Delta x \underset{\sim}{f}'_{M-1} & J_M \\
 & & & & -J_{M-1} & -\Delta x \underset{\sim}{f}'_M + J_M
\end{bmatrix}
$$

where J_i is the Jacobian matrix of $A(\underset{\sim}{u}_i)\underset{\sim}{u}_i$ and $\underset{\sim}{f}'$ is the Jacobian matrix of $\underset{\sim}{f}$. To solve the linear system given above by multilevel techniques one would, following along the lines of the previous sections, premultiply the system by

$$C := \mathrm{Diag}(J_1^{-1}, J_2^{-1}, \ldots, J_{M-1}^{-1}, J_M^{-1})$$

and then use the previous ideas in their extensions from scalar to "block" form; that is, all permutations would be done blockwise, etc. The whole process was programmed and applied to a test problem of smooth supersonic expansion around a corner (the Prandtl-Meyer problem). The method worked; Newton's method converged nicely (the analytical solution of the problem is known, so the initial guess could be chosen fairly close to the exact solution of the continuous problem), and the multilevel procedure at each stage of Newton's method also converged adequately. The next problem to investigate is one of subsonic flow.

In conclusion, an attempt has been made to give a framework for the analysis of multilevel methods that is sufficiently general to embrace both elliptic and non-elliptic problems. The key ingredients are the relaxation process, the interpolation and restriction processes, and their relation to eigenvectors of the matrix of the linear system. Emphasis has been on the smoothness of the eigenvectors associated with small eigenvalues. What would be desirable would be a way to precondition the linear system so that the small eigenvectors are smooth on the given grid. Then natural restriction and interpolation processes should work well. For some one-dimensional problems, such a preconditioning has been given. A basic problem is to find such a preconditioning for problems in more than one space dimension, for it is only in higher dimensions that the full power of multilevel techniques can make itself felt.

REFERENCES

Brandt, A.: Multilevel Solutions to Boundary Value Problems. Math. Comp., vol. 31, 1977, pp. 333-390.

Frederickson, P.: Fast Approximate Inversion of Large Sparse Linear Systems. Report 7-75, Department of Mathematical Sciences, Lakehead U., Thunder Bay, Ont., 1975.

Hackbusch, W.: On the Multi-Grid Method Applied to Difference Equations. Computing, vol. 20, 1978, pp. 291-306.

Lomax, H.; Pulliam, T.; and Jespersen, D.: Eigensystem Analysis Techniques for Finite Difference Equations. I. Multilevel Techniques. AIAA Paper 81-1027, Palo Alto, Calif., 1981.

McCormick, S.: Multigrid Methods: An Alternate Viewpoint. UCID-18487, Lawrence Livermore Laboratory, Livermore, Calif., Oct. 1979.

McDonald, B. E.: The Chebychev Method for Solving Nonself-Adjoint Equations on a Vector Computer. J. Comp. Phys., vol. 35, 1980, pp. 147-168.

Manteuffel, T. A.: The Tchebychev Iteration for Nonsymmetric Linear Systems. Numer. Math., vol. 28, 1977, pp. 307-327.

Wesseling, P.: The Rate of Convergence of a Multiple Grid Method. Lecture Notes in Mathematics, vol. 773, G. A. Watson, ed., Springer-Verlag, 1980.

ADVANTAGES OF MULTI-GRID METHODS FOR CERTIFYING THE ACCURACY OF PDE MODELING*

C. K. Forester
Boeing Military Airplane Company

SUMMARY

Application of computer-aided analysis techniques for modelling partial differential equations (PDE) requires specification of the boundary conditions, initial conditions, modelling error criteria and error tolerances that relate realistically to the physical problem of interest. To be valid, the analysis must feature numerical techniques for assessing and certifying the accuracy of the modeling of the PDE to the user's specifications. Examples of the certification process with conventional techniques (reference 1-15) are summarized for the 3-D steady full-potential and the 2-D steady Navier-Stokes equations using fixed grid methods (FG). The advantages of the Full Approximation Storage (FAS) scheme of the multi-grid (MG) technique of A. Brandt (reference 16-19) compared with the conventional certification process of modeling PDE are illustrated in 1-D with the transformed potential equation. Inferences are drawn for how MG will improve the certification process of the numerical modeling of 2-D and 3-D PDE systems. Elements of the error assessment process that are common to FG and MG include

1. generating physical domain trial grids that are useful for estimating the contamination of the results by residual and truncation errors,

2. assessing the contamination of selected trial grid solutions by the nature of the solution process (residual error effects),

3. assessing the contamination of selected trial grid solutions by the nature of the choice of the grid (truncation error effects),

4. adjust the grid until the allowable error bounds are satisfied. Error norms suitable to the application are an implied requirement.

* This work is performed under NASA Langley Research Center Contract NAS1-16408. The contract monitor is Phil Drummond who is associated with the Hypersonic Engine and Computational Methods Branches.

CONVENTIONAL CERTIFICATION PROCESS

Numerical error assessment with conventional PDE modelling techniques (references 1-14) includes the following ingredients. A solution of finite difference equations (simultaneous system of algebraic equations) for a specific discretization of the analysis domain is generated for different choices of grid density and grid distribution in the analysis domain. It is common to use a sequence of grids of the same grid distribution that differ in grid count in each independent variable direction by factors of two — 2, 4, 8, 16, 32, etc. The coarser grids can be generated by deleting every other point of the finer grids. The effects of the choice of grid distribution are examined by choosing sequences of grids which have different mesh distribution. The data from all of these solutions of the grid-related equations is organized by constructing an error difference table. Solution differences are posted in order of the coarse-to-fine grids for each grid sequence. The solution differences are generated by subtracting the values of adjoining pairs of grid solutions of the dependent variables at all physical locations in the analysis domain that correspond to the grid coordinates of a grid of a selected intermediate density. Interpolation is used to relate other grid solutions to these selected grid coordinates. As the grid density increases the differences should decay approximately according to the formal order of accuracy for some selected mesh distribution. If this occurs, extrapolation to solutions to infinite grid density may be well-behaved and reliable estimates of the maximum global error on the finest grid may result. The preceding process appears to work best on the modelling of parabolic and elliptic equations in smooth domains with smooth boundary conditions. For mixed elliptic/hyperbolic systems erratic results may occur due to unresolved singularity regions and/or poor residual error control.

A key aspect of the previous description is that grid adjustments are made in some pattern that tends toward a limiting grid configuration. A way to think about this is to define a goal-oriented reference grid ('goal grid') to which the initially selected grid sequences must evolve. The 'goal grid' serves as the host upon which the solution will be known to some resulting error bound. The goal is that this error bound will be within the accuracy desired by the analysis process. It should be understood that the 'goal grid' may not be exactly unique in pattern because of grid initialization, grid generator, and grid-equation solver properties. It is assumed that adequate control of the residual error effect have been observed in the process of assessing the truncation error effect. This is done by developing a sequence of <u>several solutions</u> on <u>each grid choice</u> with various choices of constraints on the residual toleraces that are used to terminate the computations for each solution on that grid.

Conventional techniques for developing the data that is necessary to certify the accuracy of numerical modeling procedures are limited by five factors.

1. Adequate control of residual-error effects are costly in computer time (computer resource intensive).

2. Because of (1) above, very limited numbers of solutions are available which makes for very spare information from which error estimates can be constructed.

3. Because suitable grid adjustment to control the error within desired bounds is cumbersome or impractical, arriving at proper grid configurations in mesh density and distribution is often very difficult or impractical.

4. Numerical error during grid refinement may be erratic, not monotonic. Confusion as to the grid adjustment needs can result.

5. The computer program machinery is usually not available for conveniently constructing the error table. This means that the error assessment process is manpower intensive. These factors discourage careful, complete development of the 'goal grid' solution. Without this, the accuracy of the result is unknown; the meaning of the result is undefined and useless.

MATHEMATICAL DESCRIPTION

Let $LU = 0$ (1)

represent the PDE system of interest.
In discretized operator notation equation (1) is

$$L^I U^I = \tau^I - R^I \tag{2}$$

where τ^I is the local truncation error and R^I is the local residual error for each cell of the analysis domain. The grid structure index, I, is related to choices of maximum indicies of independent variables (K,L,M) and grid density distributions for each selected trial grid. I_g is defined as the 'goal grid' index.

An ideal or perfect difference scheme for (2) is one in which the local truncation error does not contaminate the decoded variables of interest such as velocity, density, pressure, etc. Only the residual errors impact these variables. Thus, the user specifies exactly the locations in the geometry at which values of these variables are desired. With residual error control within adequate bounds, the accuracy of the result is insured within selected limits.

Nonideal difference schemes are defined as those in which the local truncation error and residual errors simultaneously influence the value of the decoded variables. Except for specialized difference schemes for model problems, difference schemes for conventional applied analysis are nonideal. Conventional steady state numerical modeling of the full-potential equation, the Euler equations, and Navier-Stokes equations in two-and three-space dimensions are subject to both of these problems with the exception of academic cases. Conventional techniques (references 11-14) primarily address the means of efficiently controlling the residual errors rather than the truncation errors. In the present account concern for the control of both error sources is present, but the emphasis is upon the approach for controlling the truncation error. This subject is closely related to the problem of proper grid adjustment from an initial state to the 'goal grid' state with proper residual control during the grid-adjustment process.

TRUNCATION (GRID RELATED) ERRORS

The local truncation error is formally defined as the magnitude that the left-hand side of (2) yields for each cell when the 'goal grid' solution is interpolated (restricted) to any other of the trial grids. It is targeted at zero in conventional representations of (2) for all values of I. The form of MG of concern in the present account has the local truncation error targeted at zero only for the I_g grid. The local residual is computed by rearranging (2) and solving for R_I. A perfect computer solution of (2) renders R_I equal to zero to within round-off errors for all values of I for FG and MG approaches. R_I is targeted at zero for all values of I in FG and MG approaches. Fortunately the MG solution process does not require any knowledge of the 'goal grid' solution in order to generate useful estimates of the local truncation error. It is a deferred correction process in which the relative local truncation error estimates between grid pairs are corrected as solutions on the finer grid levels become available. These corrections are not major at the coarser levels as the finer and finer grid levels evolve. This is one of the powerful aspects of MG.

To clarify the nature of the grid structure index for conventional conformal analysis, the nomenclature I equals IG_{JG} is introduced where IG refers to the grid density and JG refers to the grid configuration. For example, nonorthogonal conformal-grid analysis methods require the user to choose the stretch factor related parameters that control the physical domain mesh interval rate changes in length and twist for the various independent variable directions ξ, η, σ. One choice of these parameters coincides with a value of JG. The selection of the number of grid cells in the ξ, η, σ directions relates to the selected value of IG for a selected value of JG. The maximum values of K,L,M are therefore defined by IG for conventional computer program index controls on the problem size. Symbolically these notions can be expressed as

$$I \equiv IG_{JG}^{*}$$

$$IG = IG\,(n_1\delta\xi,\ n_2\delta\eta,\ n_3\delta\sigma)$$

* This definition is incomplete for composite grids which feature grid nesting, grid overlays or coupled conformal regions with discontinuous interfaces in the logic or transformed space.

(n_1, n_2, n_3) = no. of grid intervals in ξ, η, σ directions

$K_{max} = n_1$, $L_{max} = n_2$, $M_{max} = n_3$

JG = JG (O,P,Q)

O,P,Q = physical space grid compression functions along ξ, η, σ directions

Figure 1 shows a typical distorted hexahedral type of computational cell. Hybred finite difference/finite element analysis techniques employ ordered arrays of these cells. The resulting aggregate of cells can be viewed as nonothogonal conformal-type analysis grids. A research V/STOL inlet photograph is shown in figure 2. Front and elevation views sketches of this inlet are shown in figure 3. A 3-D grid of this inlet has been generated for full-potential flow analysis (reference 10). Figure 4a shows a longitudinal slice of this grid through the crown and keel lines of the inlet. Figure 4b shows typical grid detail near the hilite of the inlet.

Conventional certification of the accuracy of full-potential flow analysis with these type of grids has been performed (references 7, 10) in which variations in IG and JG were made to develop the 'goal grid' shape. Additionally direct experience with the conventional certification process of the accuracy of Navier-Stokes analysis with conformal grid techniques (references 2-4, 6, 9) and with composite conformal grid techniques (reference 4) has been accumulated. This experience has lead to an understanding of practical problems of defining where grid adjustment is needed and how to make the proper grid adjustments with conventional techniques in order to obtain the required 'goal grid' solutions. The conclusion is that vast improvements on the conventional techniques are needed. The critical areas that require improvements for efficient application of surface grid (panel methods) and field grid methods include developing

1. practical error monitors and practical error bounds that assist the grid adjustment processes efficiently,

2. discretized analysis formulations that permit more grid flexibility attendent with reduced grid-adjustment complexity,

3. more efficient and flexible approaches to reducing residual errors,

4. processes for coupling the grid adjustment and error monitor together so that the <u>bulk</u> of the computational effort is directed to the 'goal grid' configuration. In other words, develop schemes that minimize the effort expended on trial grids in which the truncation errors are out of bounds. This is the goal for the most cost effective applied analysis methods. Many intermediate steps toward this goal are required.

In order to begin the development of MG error assessment technology, model problems with exact solutions are being studied. The results of some of this work are discussed in the next section for the mass conservative transformed potential equation. Boundary conditions of the exact velocity are used to set the gradient of the velocity potential at the grid entrance and exit cell faces. Iterative adjustment of boundary velocity potential is used at every relaxation sweep. The analytical velocity solution is a function of the ratio of channel entrance area to the channel cross section at any other station of interest. The channel area variation with station position is defined by cubic functions.

RESULTS OF I-D ERROR ASSESSMENT STUDY FOR STEADY INCOMPRESSIBLE FLOW

The discretized 3-D full-potential equation is restricted to a 1-D analysis tool (by deleting the L and M indicies). The total velocity can be computed in many ways. One formulation (reference 7) yields values of the total velocity in which the truncation errors in the velocity potential do not contaminate the computation of the total velocity. This formulation is used to illustrate uses for the local truncation error estimates. Various properties of MG process are illustrated also. A 1-D incompressible flow problem for which analytical solutions are readily available is employed for error assessment. An adaptively gridded test case is presented and its implications are discussed.

The 1-D test problem involves an analytical geometry of a straight channel with a cubic function for a constriction that reverts either abruptly step-wise or smoothly to a straight channel. Figure 5 shows the channel section shape distribution with respect to the flow direction. Figure 6 shows the analytical solution restricted to 65 grid coordinates (64 cells) with the grid intervals constant. Eleven trial fine-grid sets (JG_{max} = 11) were used to examine the 1-D potential solution properties for a) grid with uniform mesh intervals, b) grid with uniform mesh intervals in the region of cross sectional area variation but with a stretch factor of two in the straight sections, and c) grids with uniform mesh intervals in the straight sections but with a stretch factor of .80, .85, .90, .95, 1.0, 1.05, 1.1, 1.15, and 1.2 in the constricted region where the finest grid is near the abrupt enlargement of the channel cross sectional area for stretch factors less than unity. The total number of grid intervals for each set of trial grids are 4, 8, 16, 32, and 64 (IG_{max} = 5), where the number of grid intervals in the constriction region are respectively 2, 4, 8, 16, and 32. FG and MG methods have been applied to generate solutions for these sets of trial grids. The general character of these solutions is shown in Figure 6 by the solid line for the finest grid. Also shown in Figure 6 is the FG and MG solutions with very nonstringent residual error tolerances. Using point relaxation and sweeping the grid in the flow direction, MG yields a maximum global error of less than 4% in the equivalent of twenty-five sweeps of the 64 node grid whereas FG requires over one thousand sweeps of the 64 node grid to achieve the same accuracy. The maximum error occurs at the geometric discontinuity. Increasing the accuracy by an order of magnitude requires less than a factor of three increase in the work for the MG and the FG. The process of solving the problem to greater accuracy can be continued until the maximum global error satisfies desired constraints up to round-off error effects. The boundary conditions are imposed both on the FG and MG as set mass rates of equal magnitude at the entrance and exit cross sections.

Control of the contamination of the total velocity output is correlated with the computer work expended in solving the grid equations. The data shows that the residual error control efficiency increasingly favors MG over FG as the number of grid points is increased. This result is in keeping with Brandt's (references 16-19) results. For a simple elliptic problem, this result establishes one type advantage of MG over FG procedures: MG is asymototically more efficient than the FG strategy in controlling residual error. Hence the number of grid points that can be considered in an analysis with MG is greater than FG for a given computer budget. The inference of this advantage is summarized as: the potential for control of truncation error is greater with MG than FG strategy for nonideal difference schemes.

Residual errors and maximum global errors are observed to be directly linked. This can be examined by computing the discrete continuity balance (local mass balance) on each cell. By dividing the local mass balance by the local channel cross sectional area, a delta velocity results which if added to the local velocity is the correction necessary to remove the local residual error. The maximum global error is reduced to round-off error (below ten to the minus ten) when the residual velocity correction is applied successively from the entrance region point-by-point through the grid to the exit region. Alternatively the maximum global error can be computed directly from the sum of the residuals of the same sign divided by the channel cross section at which the sign in the residual changes. The channel entrance area has been set to unity.

The form of MG that is used for the computations involves a nonzero right-hand side term. With this formulation the discretized continuity equation is viewed as having a mass source right-hand side term which is constructed from the estimate of the local truncation error. Fine grid velocity potential data are interpolated (restricted) to coarse-grid continuity balances to obtain estimates of the local truncation error where global integral is zero for mass conservation. Total velocity output that is decoded from solutions of these coarse-grid Poisson-type equations are not directly useful (with an academic exception). This is a key point about MG output: the total velocity output on coarsest grids are useless in themselves. This point is illustrated in Figure 7 for three grid levels. Note that the results near the geometric discontinuity are always badly in error. In the coarsest grid the local truncation error from the geometric discontinuity contaminates the total velocities at three cell faces where the solution is developed. The extent of the contamination is reduced dramatically as the grid is refined but it is only eliminated on the finest grid level where it is exactly zero by choice. Any other choice for the finest grid solution would generate worse results than that shown; the maximum global error would be larger near the discontinuity than occurs in the present example. Therefore the truncation error extrapolation, τ-extrapolation, (ref. 15) cannot be inserted at the finest grid level, only at next to the finest grid levels. As shown in reference 19, it can be used as a method for accelerating solution convergence or for generating still finer grid solutions (finer than 64 cell cases in the present example) at lower cost. Alternatively, a finest grid selection of 32 cells could be used with τ-extrapolation to get the solution that is shown in Figure 6.

Figure 8 shows the truncation error spectrum for the peak values of the local truncation error asymptotically approach nearly the same values including τ-extrapolation on the next-to-the-finest grid solution. The magnitude of these terms are substantial near the discontinuity and, because they form the right-hand side of the cell-wise flux balance equations, induce large errors in the total velocity profiles that are shown in Figure 7. The coarse-to-fine grid correction equation of Brandt very effectively interpolates the Poisson type solutions on coarser grids so that the coarser grid solutions mimic the finer grid solutions. Standard interpolation, $\emptyset^{I+1}_{new} = I^{I+1}_{I}(\emptyset^{I}_{new})$ cannot account on the next finer grid, I+1, for the fact that the right-hand side term is significant in the coarser grid solutions. For this reason standard interpolation is not useful and must be replaced by a more elaborate interpolation. Brandt recommends

$$\emptyset^{I+1}_{new} = I^{I+1}_{I} \left(\emptyset^{I}_{new} - I^{I}_{I+1}\, \emptyset^{I+1}_{old}\right) + \emptyset^{I+1}_{old}$$

where I^{I}_{I+1} is the fine-to-coarse grid interpolation operator and I^{I+1}_{I} is the coarse-to-fine grid interpolation operator. This expression functions well as illustrated in Figure 9. Linear interpolation is used for these operators with weightings of 1/4 and 3/4 for I^{I+1}_{I} and weightings of 1/2 and 1/2 for I^{I}_{I+1}. No modification of these weights is used for stretched grid cases whose principle effect is to retard the convergence rate by up to about one-third for cases with stretch factors of .80 and 1.2.

In the following discussion uses of the local truncation error estimates for grid adjustment are discussed. A simple example of semi-adaptive grid refinement is shown in Figure 10 in which grid compression toward the region of high local truncation error is used. Iterative grid compression is continued until a condition of the maximum normalized local truncation error is less than .08. Semi-adaptive grid compression is implemented in the interval $0 \leq z/L \leq 1$ by iteratively decrementing the grid stretch factor from an initial value of 1.2 in steps of .05. As expected no satisfaction of the tolerance on the maximum local truncation error is found as long as an exact step-wise discontinuity is enforced at a z/L equal to unity. With a cubic transition function in the interval $31/32 \leq z/L \leq 32/32$ which has slope continuity with the remaining channel geometry, local truncation error reduction results with grid refinement. Figure 10 shows the results of the analytical solution and solution with a grid contracted toward z/L equal to unity. Over an order of magnitude reduction in the local truncation is readily achieved with a contraction ratio of .85. Obviously the selection of τ_{max} at .08 as the criteria for stopping the computation is arbitrary. The lower the magnitude selected for the stopping criteria the more grid is compressed into the region of the abrupt geometry change. Eventually this approach starves the remaining domain of the analysis of sufficient mesh to satisfy the selected maximum local truncation error tolerance. Therefore a preferred strategy involves sub-dividing the region of small length scale, $31/32 \leq z/L \leq 32/32$, with a uniform grid of varying number of grid points. It is easy to implement. It is regarded also as semi-adaptive. A 'fully' adaptive strategy requires labeling each cell of a trial grid with a special flag that designates cells with a local truncation error that exceeds a selected threshold value. Cells so flagged may be subdivided by nesting compressed grids or by uniform interval grid embedding. 'Fully' adaptive MG strategy only requires that iterative work to reduce the truncation error be applied to the flagged cells. This approach may be more efficient, 'fully' adaptive and more computer programming intensive than the semi-adaptive strategies. This approach appears to be practical to program for machine computations.

It is clear in the preceding simple problem that the local truncation error estimates indicate the proper region in which grid adjustment (mesh density or distribution) should occur or the proper region in which the geometric representation of the boundary of the analysis domain may need modifications. It is expected that shock wave or unresolved shear layer singularities would likewise produce normalized local truncation error estimates of the order of unity. A list of six causes of large truncation error includes

1. mesh density
2. mesh distribution
3. shock singularity
4. unresolved shear layer singularity
5. boundary condition discontinuity
6. improperly controlled residual errors

It is certain that the local truncation error estimates in themselves cannot distinguish among these six causes of large local error or whether the desired results of the analysis output are adversely affected. Therefore additional information must be associated with the local truncation error estimates to make them useful in certifying the accuracy of a numerical PDE modeling. For this purpose, error norms must be developed that assist identifying the 'goal grid' solution. Adaptive grid computations are defined as those that utilize, in an automated fashion, a link between the error norms and an adjustment of the analysis strategy. This is the essence of MLAT-FAS MG of reference 19. It is obvious that such an approach is designed with the problem of certifying the accuracy of the PDE modeling in mind. FG technology simply does not connect the error assessment difficulties with the solution algorithm design.

SUMMARY OF THE I-D INCOMPRESSIBLE FLOW STUDY RESULTS

1. The FAS-MG process accelerated the reduction of the residual errors in a manner which increasingly favors MG over FG as the number of grid points is increased.

2. The FAS-MG process is straight-forward to implement with standard FG grid-equation solution processes. Provided the FG solution process is convergent, FAS-MG is also convergent.

3. Sums of the same-sign residual error are analytically related to the maximum global error whether or not a geometric discontinuity exists.

4. Standard grid-equation formulations are modified for FAS-MG by the inclusion of an additional right-hand side term. This term is related to the local truncation error.

5. Solutions of the grid equations on all but the finest grid cannot be used directly for estimating the PDE solution. This is due to the non-zero right-hand side term of the coarse grid solutions for which proper account must be made before the coarse grid solutions are used.

6. Standard interpolation fails to be useful for prolongating coarser grid MG solutions to finer grid levels. Brandt's FAS-MG formula is effective for this purpose.

7. Estimates of the local truncation error are a direct consequence of the FAS-MG process.

8. The sign of the local truncation error oscillates at the highest possible frequency of two mesh intervals for an ideal difference scheme. This produces a cancellation of the local truncation error in the solution. Control of residual errors are all important for satisfying desired global error bounds, local truncation error is of no consequence in an ideal difference scheme.

9. Regions of the trial-grid solution in which the normalized local truncation error is of the order of unity are indicative of some potential problem with the analysis.

10. It is conjectured that nonideal difference schemes will exhibit two-mesh-interval sign oscillations in the local truncation error estimates only at singularities or at locations which have grid-related problems. Otherwise the local truncation error estimates will persist at longer wavelengths.

11. It is conjectured that sums of the same-sign local truncation errors are significant to estimating the maximum global error for nonideal difference schemes. Useful sums may or may not include the regions of large local truncation error depending on the purpose for the error norm.

OBSERVATIONS AND RECOMMENDATIONS

No simplifying assumptions have been made in the use of the Brandt FAS-MG scheme and it is expected to be generally applicable to and straightforward to apply to existing 2-D and 3-D codes. The following comments are supplied from that viewpoint.

As a practitioner of conventional applied analysis techniques for modeling PDE systems, the following goals of future work appear to be desirable.

1. Modify conventional applied analysis codes with the Brandt FAS scheme so that local truncation error estimates are a routine output. This will aid in quickly identifying regions of the analysis domain where one or more of six large truncation error problems exist. Concern over full MG optimality is not the issue for the short term, primarily it is desirable to reduce the labor involved in determining where in an analysis serious potential numerical error problems reside.

2. Develop error norms that properly exploit the local truncation error estimates of MG so that conventional, semi-adaptive and adaptive composite grid technology can achieve high efficiency in the PDE modeling certification process.

3. To be effective the grid generation process and the grid-equation solution process must be drawn together. Composite grid technology in the context of the PDE modeling certification process should be encouraged. Composite grids refer to the broadest definition of grid design, coupled conformal grids in which nested grids or grid overlays are permitted by the analysis approach.

4. It is customary to compare conventional analysis results with experimental data for validation. This practice should eventually yield to the more precise requirement that the PDE modelling error assessment and numerical accuracy certification should be an independent function of high standing. The comparison with experimental data could then assume the proper role of checking that the PDE system is appropriate to the goals of the analysis application. Such a practice will offer the

advantage that the PDE modelling errors will be distinct from the PDE formulation errors. This advantage is not commonly exploited.

REFERENCES

1. Forester, C. K.: Nonisothermal Storage and Expulsion of Single Phase Cryogens, Part I 1-D, Advances in Cryogenic Engineering, Vol. 12, Plenum Press, 1967.

2. Forester, C. K.: Nonequilibrium Storage and Expulsion of Single Phase Cryogens, Part II 2-D, Preprint 9A, presented at the Symposium on Cryogenic Advances in the Space Program Second Joint AICHE-IIQPR Meeting, Tampa, Florida, May 19-22, 1968.

3. Forester, C. K. and Emery, A. F.: A Computational Method for Low Mach Number Unsteady Compressible Free Convective Flow, J. of Computational Physics 10, 487-502, 1972.

4. Forester, C. K.: Task Summary Report - Flight Certification Navier-Stokes Analysis of the Redesigned Apollo 13 Oxygen Storage System - Math. Model Improvements and Accuracy Assessment, Boeing Report (D2-118407-1A, -2) prepared for NASA Manned Spacecraft Center, Houston, Texas, 1971, Contract No. NAS9-11576. Post-Flight Apollo 12, 14 and 15 Navier-Stokes Analysis of the Redesigned Apollo 13 Oxygen Supply System, requested for publication by Editor of the Advances in Cryogenic Engr. Navier-Stokes Analysis of the Space Shuttle Cryogenic Supply System, Boeing Report (D2-118462-1, -2) prepared for NASA MSC, Houston, Texas, 1973, Contract No. NAS9-12977.

5. Forester, C. K.: Higher Order Monotonic Convective Difference Schemes, J. of Computational Physics, 23, 1977, pp 1-22.

6. Forester, C.K.: Numerical Simulation of the Interaction of Jet and Freestream Flows in Engine Exhaust Systems, AIAA paper 78-144, AIAA 16th Aerospace Sciences Meeting, Huntsville, Alabama, January 16-18, 1978.

7. Forester, C.K.: Numerical Prediction of Compressible Potential Flow for Arbitrary Geometries, Winter Annual Meeting ASME, December 1979. Also accepted for publication in J. of Fluids Engineering.

8. Roberts, D. W. and Forester, C. K.: A Parabolic Procedure for Ducts with Aribtrary Cross Sections, AIAA Journal, January 1979, pp. 33-40.

9. Peery, K. M. and Forester, C. K.: Numerical Simulation of Multi-Stream Nozzle Flows, AIAA Journal, July 1980.

10. Forester, C. K.: Body-Fitted 3-D Full-Potential Flow Analysis of Complex Ducts and Inlets, AIAA paper 81-0002, AIAA 19th Aerospace Sciences Meeting, January 12-15, 1981, St. Louis, Missouri.

11. Jameson, A.: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multigrid Method, AIAA paper 79-1458, 1979.

12. MacCormack, R. W.: An Efficient Explicit-Implicit Characteristic Method for Solving the Compressible Navier-Stokes Equations, SIAM-AMS Proceedings, Vol. 11, 1978, pp. 130-155.

13. Pulliam, T. H. and Steger, J. L.: On Implicit Finite Difference Simulations of Three-Dimensional Flow, AIAA Journal, Vol. 18, No. 2, February 1980, pp. 159-167.

14. MacCormack, R. W.: A Numerical Method for Solving the Equations of Compresside Viscous Flow, AIAA paper 81-0110, St. Louis, Missouri, January 1981.

15. Rizzetta, D. P.; Burggraf, O. R.; and Jenson, Richard: Triple-deck Solutions for Viscous Supersonic and Hypersonic Flow Past Corners, J. Fluid Mech, Vol. 89, 1978, pp. 535-552.

16. Brandt, Achi: Multi-Level Adaptive Solution to Boundary Value Problems, Math. Comp., Vol. 31, 1977, pp. 333-390.

17. Brandt, Achi: Multi-Level Adaptive Solutions to Partial Differential Equations, Ideas and Software III, in Proceedings of Symposium on Mathematical Software (J. Rice, ed.) Academic Press, New York, 1977, pp. 277-317.

18. Brandt, Achi: Multi-Level Adaptive Computations in Fluid Dynamics, Proceedings AIAA Computational Fluid Dyamics Conference, July 23-25, 1979, Williamsburg, VA.

19. Brandt, Achi: Multi-Level Adaptive Techniques (MLAT) for Singular Perturbation Problems, in Numerical Analysis of Singular Perturbation Problems, P. W. Hemker and J. J. H. Miller, eds., Academic Pres, London, 1979.

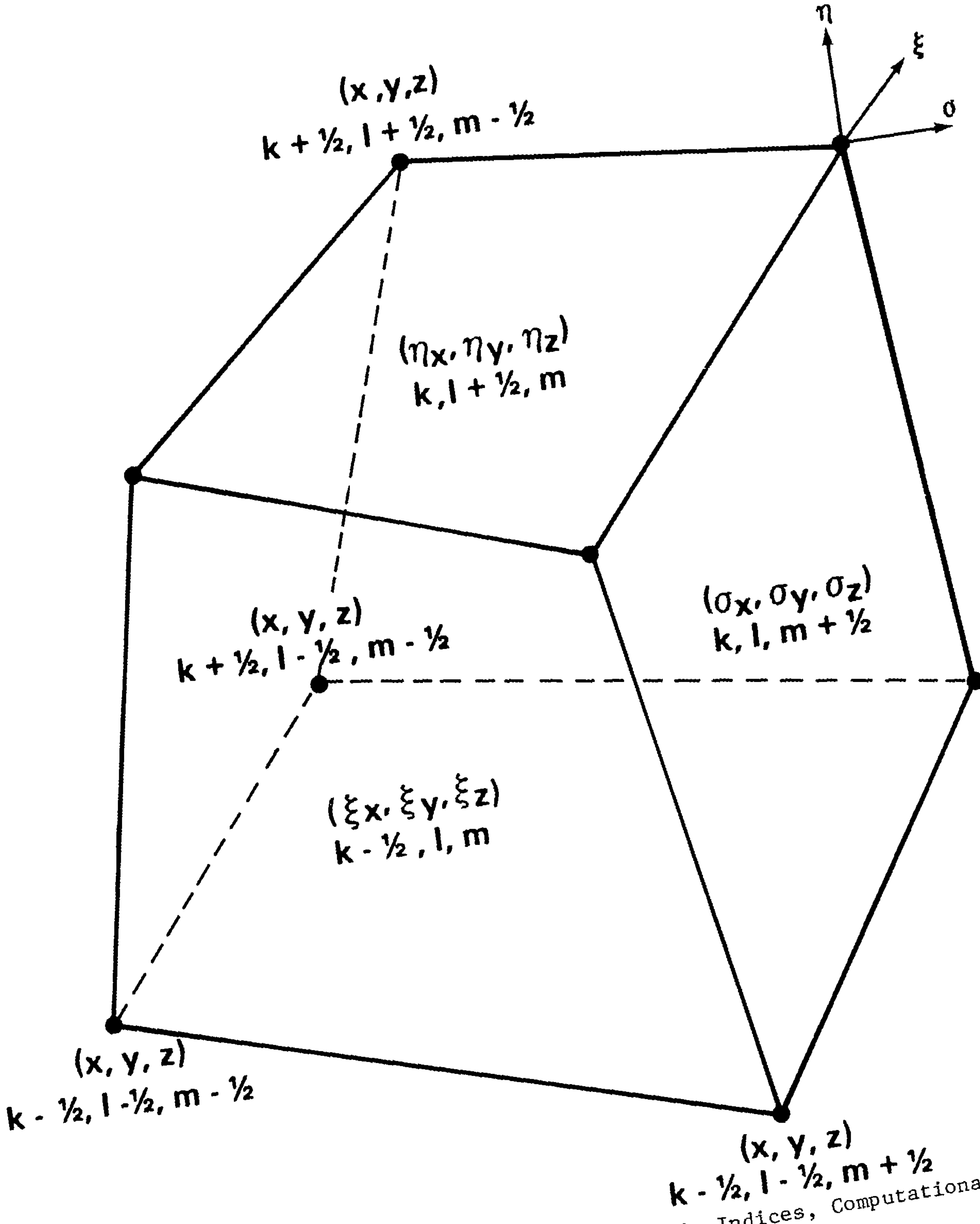


Figure 1. The Relationship of Computer Logic Indices, Computational Cell Vertices and Cell Face Metrics

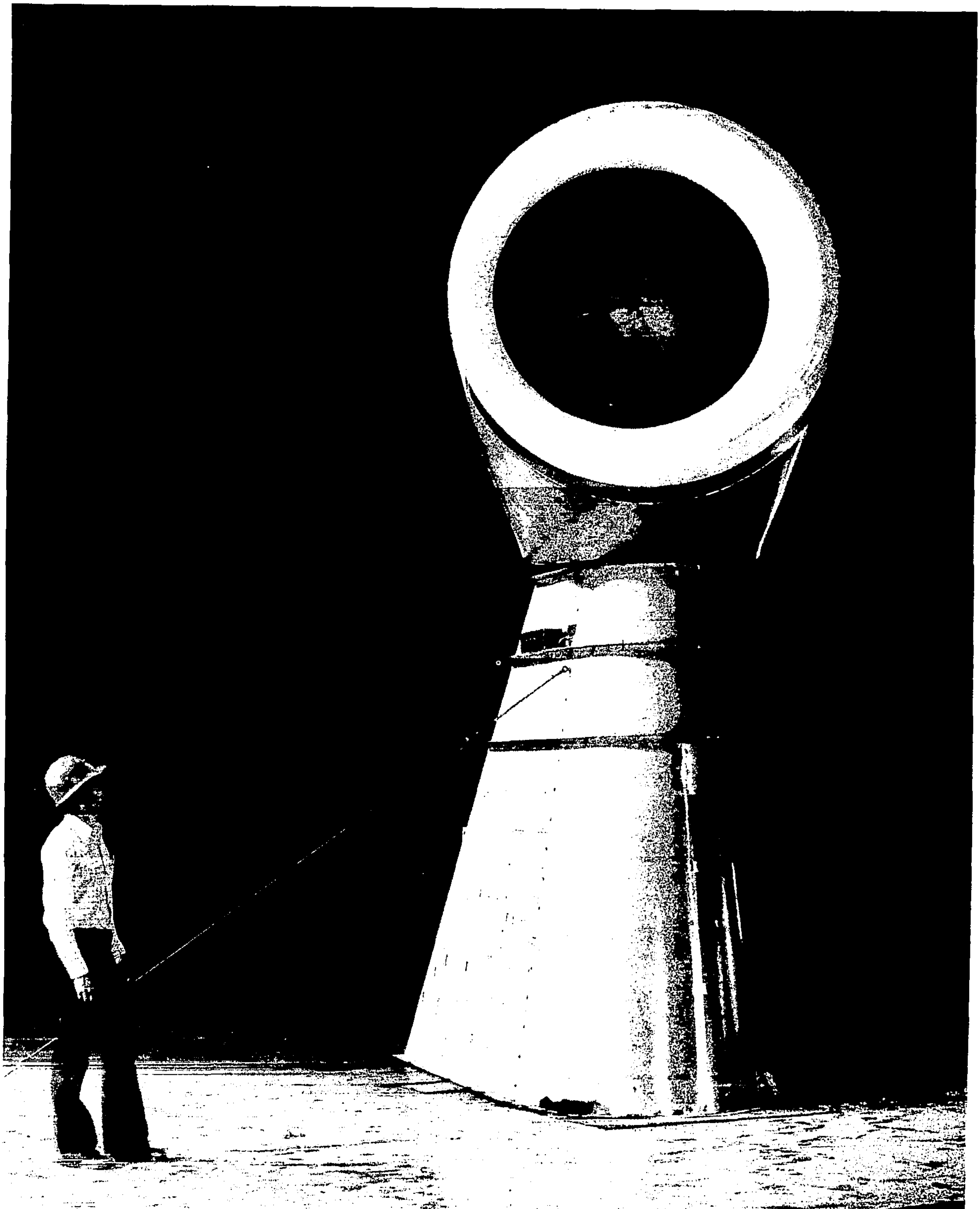


Figure 2. Asymmetric V/STOL Research Inlet

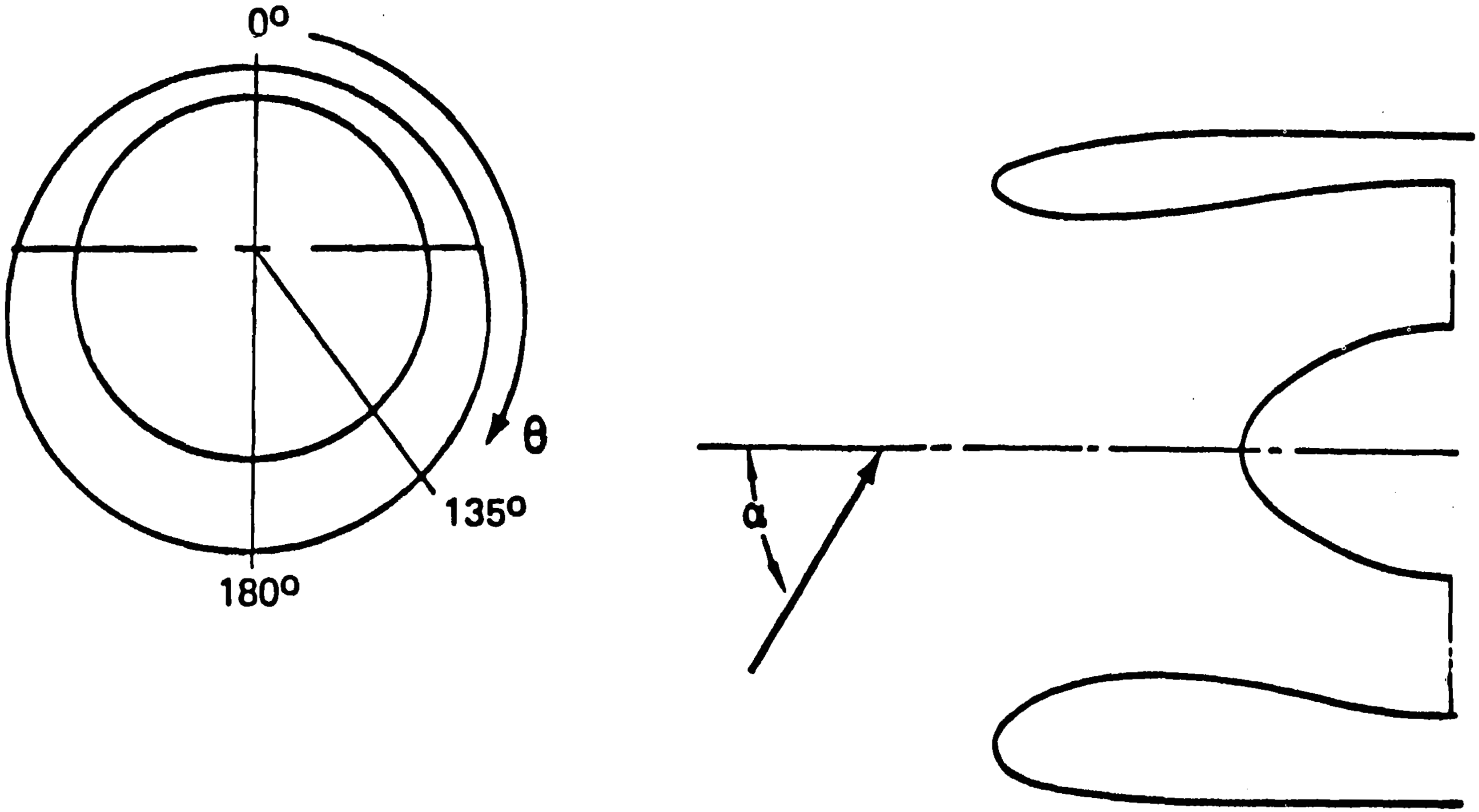


Figure 3. Sketch of Asymmetrical V/STOL Inlet

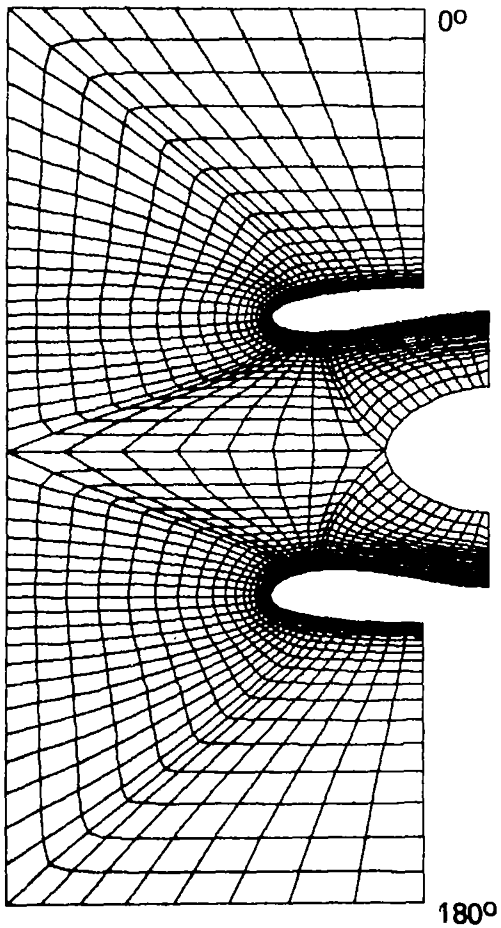


Figure 4a. Elevation View

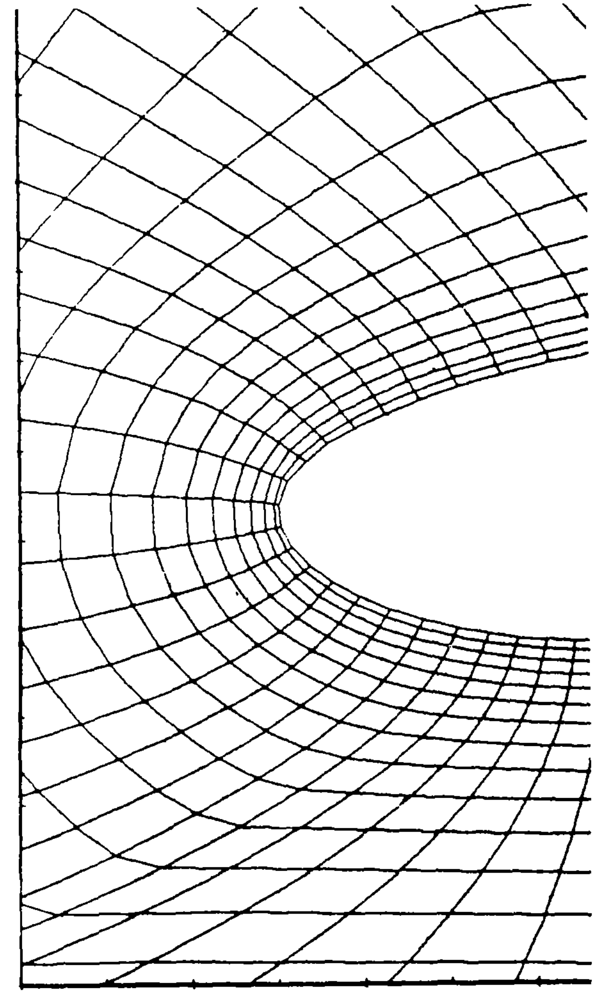


Figure 4b. Hilite Grid Detail

Figure 4. Grid at Symmetry Planes of V/STOL Inlet

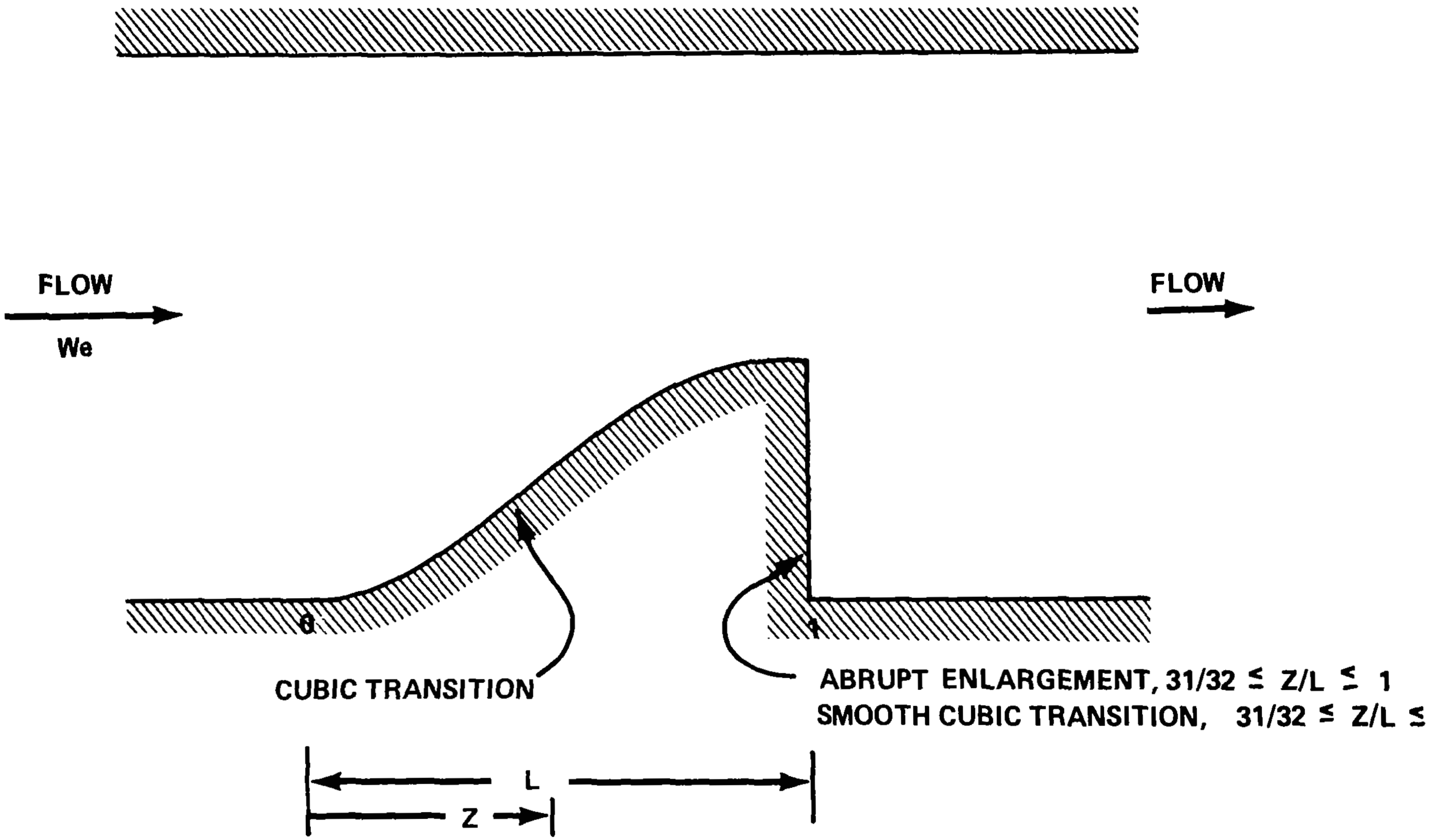


Figure 5. One-dimensional Incompressible Flow Channel

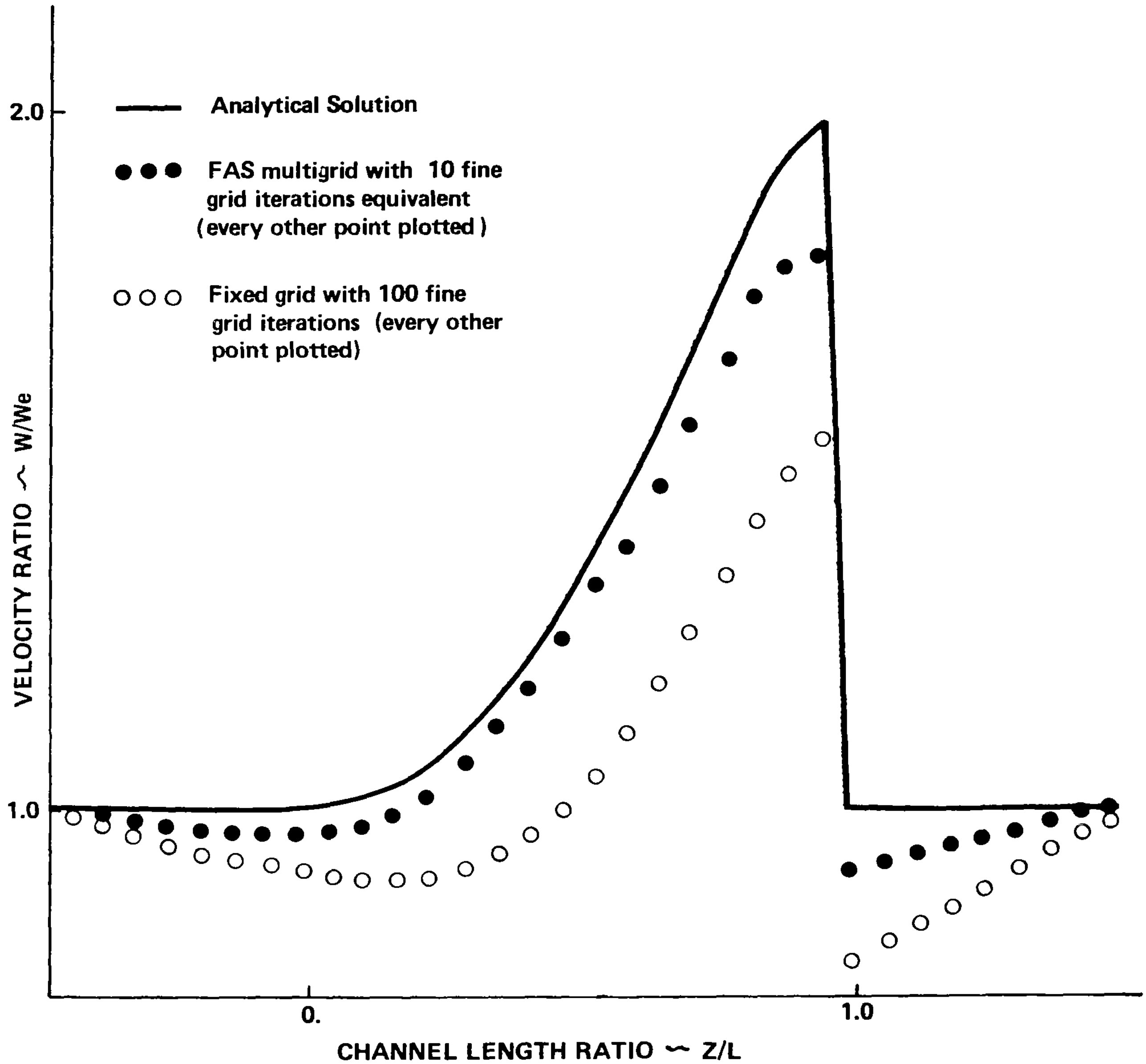


Figure 6. Residual Error Effects on Fixed Grid and Multigrid Solutions

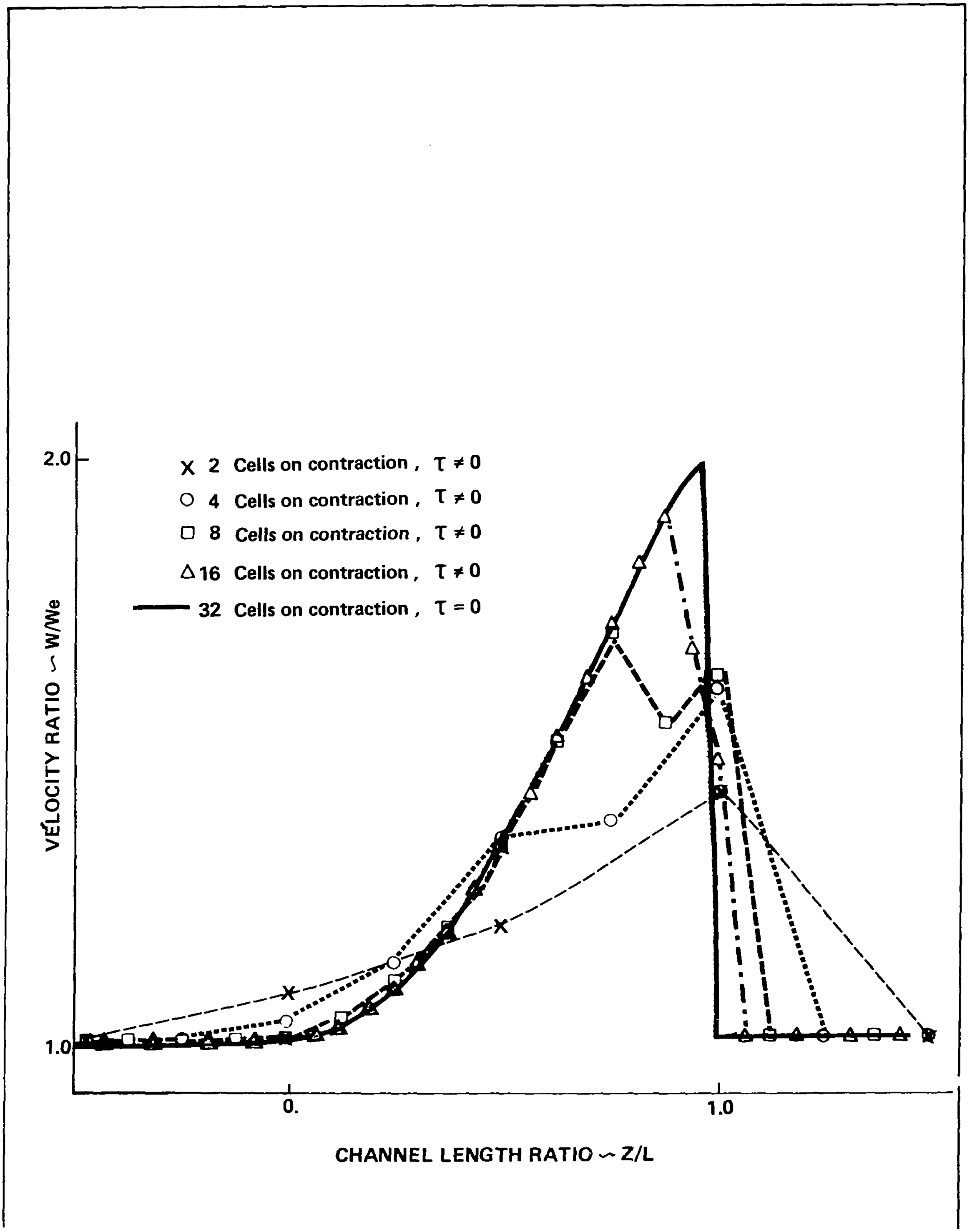


Figure 7. Truncation Error Effects on Intermediate Grid Level Solutions

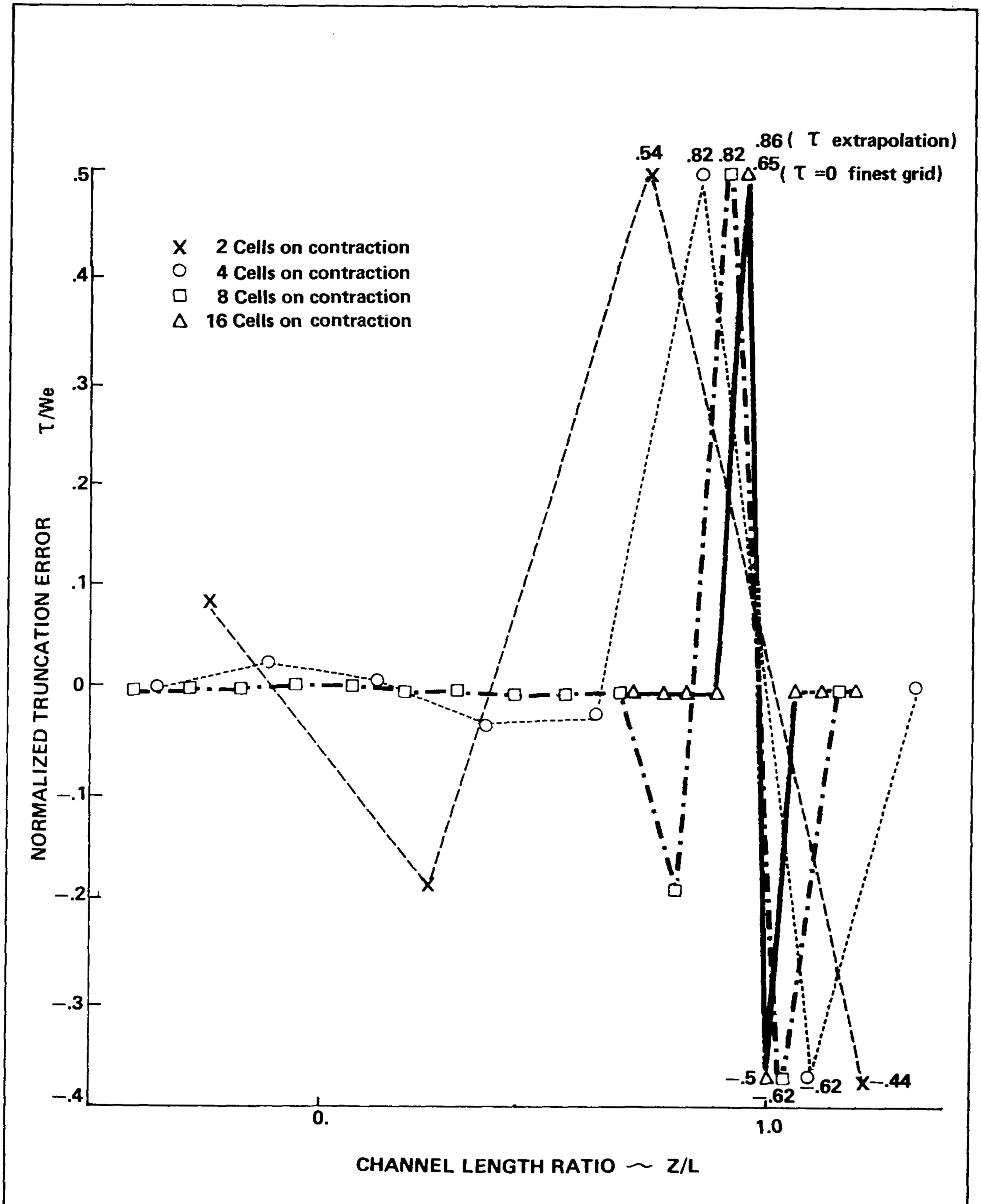


Figure 8. Truncation Error Spectrum as a Function of Grid Density

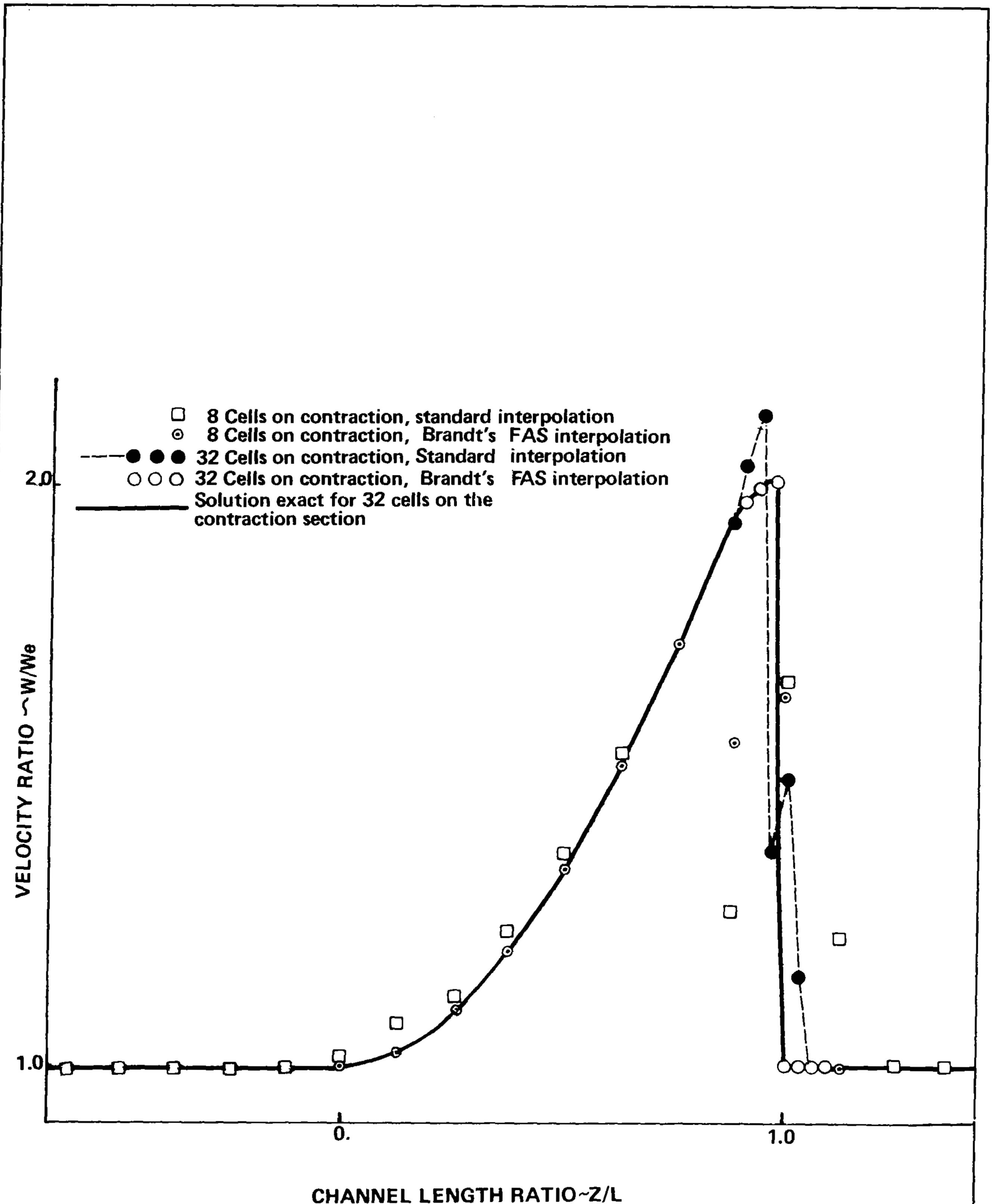


Figure 9. The Need of Brandt's Coarse-to-Fine Grid Correction Equation

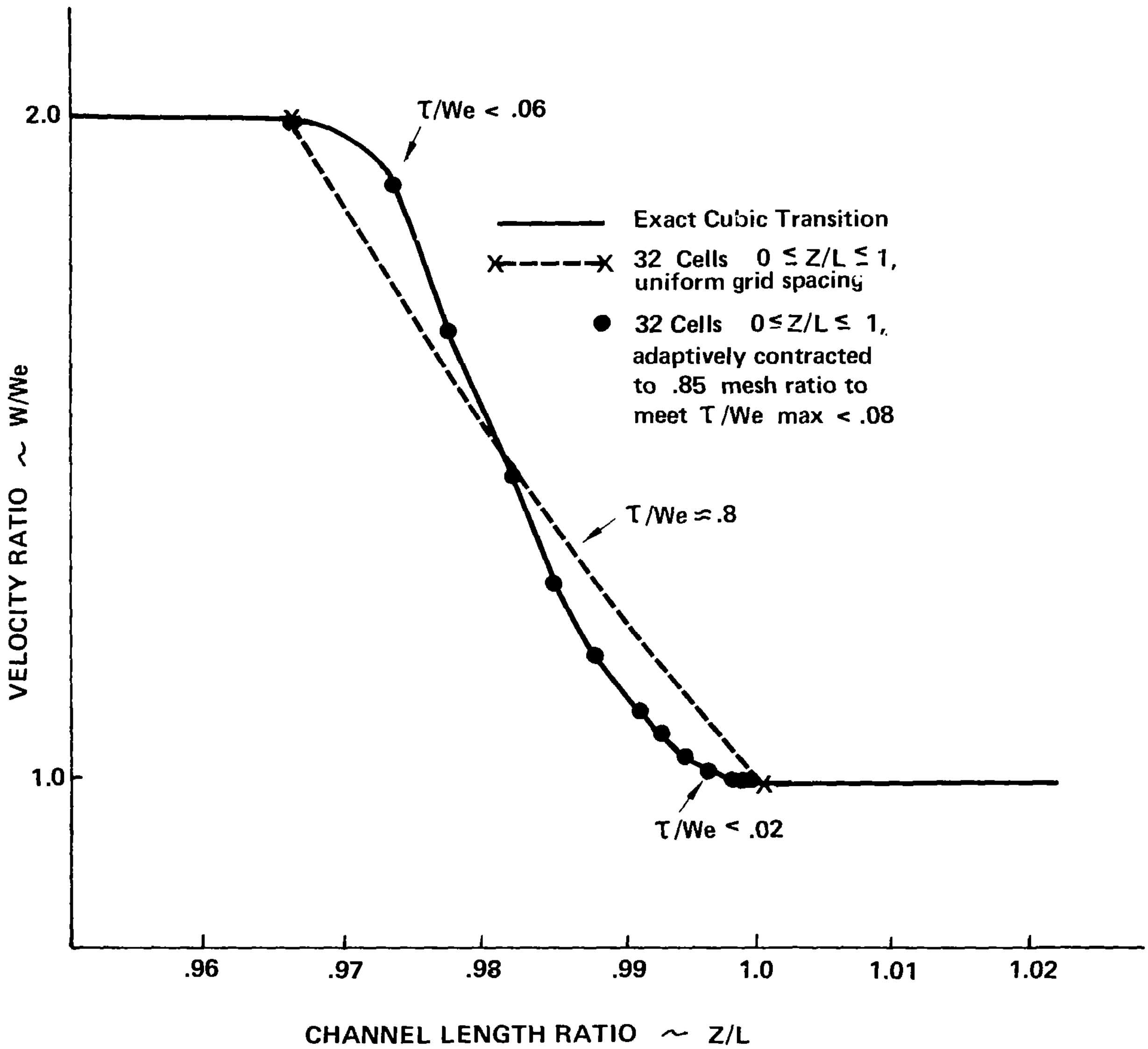


Figure 10. Adaptive Grid Control to Selected Truncation Error Tolerance

MULTIGRID METHOD WITH WEIGHTED MEAN SCHEME

Liviu R. Lustman
Systems and Applied Sciences Corporation, Hampton, VA

Quyen Huynh
Systems and Applied Sciences Corporation, Hampton, VA

M. Y. Hussaini
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center, Hampton, VA

SUMMARY

Multigrid algorithms based on the weighted mean scheme are developed for the solution of the two-dimensional incompressible Navier-Stokes equations. They are applied to two typical problems encountered in engineering applications, namely, the convection-diffusion problem of the Benard convection cell, and the driven cavity problem. An analysis of the smoothing rates and stability is given. The efficiency of the multigrid method is investigated.

The Governing Equations and Solution Technique

The two-dimensional steady-state incompressible Navier-Stokes equations are considered with vorticity, stream function and temperature as the dependent variables. They are written in the convective form:

$$u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} = \frac{1}{\alpha} \nabla^2 \phi \quad , \tag{1}$$

where ϕ represents appropriately nondimensionalized vorticity ξ or temperature T, and α correspondingly represents the Reynolds number or the Peclet number; u and v are the nondimensional velocity components in the x- and y- directions. The stream function ψ, of course, satisfies the Poisson equation

$$\nabla^2 \psi = -\zeta. \tag{2}$$

As a simple test case, we consider the "Benard cell" problem, where we solve for the temperature T, with a given velocity field

$$u = -\cos(y)\sin(x) \quad ,$$
$$v = \cos(x)\sin(y) \quad , \quad 0 \leq x,y \leq \pi$$

Research of LRL and QH was supported by NASA Contract No. NAS1-16572; research of MYH was supported by NASA Contract No. NAS1-16394.

and boundary conditions

$$T = 0 \text{ on } y = \pi ,$$
$$T = 1 \text{ on } y = 0 ,$$
$$T_x = 0 \text{ on } x = 0, \pi$$

The Numerical Method

We use the weighted mean scheme for the finite-difference approximation to Equation (1). The concept of the weighted mean scheme appears to be originally due to Allen and Southwell (ref. 1) and it has been rediscovered several times since then.

As recently pointed out by Gresho and Lee (ref. 2) this scheme was rediscovered by Spalding (ref. 3) by a more intuitive approach. Raithby and Torrance found the scheme in the two-dimensional problems they considered particularly when the grid line and velocity direction were closely aligned. Later, Raithby (ref. 4) appears to have improved upon it for cases where the flow was not aligned closely with one of the coordinate lines. Fiadeiro and Veronis found the method again, called it the weighted mean scheme, and generalized it to two- and three-dimensions. The history of this scheme in the finite element literature is briefly reviewed in reference 2. All such schemes become identical in the case of the one-dimensional steady-state advection-diffusion equation with constant coefficients: they tend to the pure central difference scheme for small Reynolds number (or Peclet number), and to the pure upwind scheme for large Reynolds number (or Peclet number); the numerical solution of the discretized equation agrees identically at the nodes with the exact solution, and thus resolves boundary layers.

Following Fiadeiro and Veronis (ref. 5), we discretize Equation (1) as follows:

$$C_{ij}\phi_{i,j} + N_{ij}\phi_{i,j+1} + S_{ij}\phi_{i,j-1} + E_{ij}\phi_{i+1,j} + W_{ij}\phi_{i-1,j} = 0. \tag{3}$$

For uniform grid, the coefficients C, N, S, E, and W are defined in terms of the local velocity components:

$$\begin{aligned} N_{ij} &= \frac{v_{j+\frac{1}{2}}}{2\Delta y}\left[\coth\left(\frac{\alpha\Delta y\, v_{j+\frac{1}{2}}}{2}\right) - 1\right] , \\ S_{ij} &= \frac{v_{j-\frac{1}{2}}}{2\Delta y}\left[\coth\left(\frac{\alpha\Delta y\, v_{j-\frac{1}{2}}}{2}\right) + 1\right] , \\ E_{ij} &= \frac{u_{i+\frac{1}{2}}}{2\Delta x}\left[\coth\left(\frac{\alpha\Delta x\, u_{i+\frac{1}{2}}}{2}\right) - 1\right] , \end{aligned} \tag{4}$$

$$W_{ij} = \frac{u_{i-\frac{1}{2}}}{2\Delta x} \left[\coth \left(\frac{\alpha\Delta x \; u_{i-\frac{1}{2}}}{2}\right) + 1\right]$$

and $C = -N-E-S-W$.

Extension to a nonuniform grid is straightforward. Analysis and the numerical results show that the method is second order accurate. It is easily established that the coefficient matrix is diagonally dominant and positive definite, and hence any reasonable iterative method will be stable and convergent on any grid irrespective of the grid Reynolds number (or grid Peclet number). However, the solution thus obtained on a coarse grid may involve gross inaccuracies owing to discretization errors. But in a multi-grid context the coarse grids are used only for smoothing the high frequency component of the residual on the fine grid. Specifically, we use the correction scheme algorithm, and generalize it to the full approximation algorithm in the terminology of Brandt (ref. 3). Gauss-Seidel relaxation or successive line relaxation is used to reduce the high frequency errors.

A quantitative measure of the relaxation efficiency is the smoothing rate defined to be the eigenvalue, largest in magnitude, of the relaxation amplification matrix for high frequency components. To evaluate the smoothing rate, the usual local analysis is employed by assuming

$$\phi_{kl} = \exp\left[i\,(k\xi + l\eta)\right] \tag{5}$$

and applying the relaxation scheme. The new values of ϕ_{kl} will have amplitudes different from unity. Ideally, the amplitudes of the high frequency modes should be reduced. This provides the definition of the asymptotic smoothing rate μ

$$\mu = \max_{\frac{\pi}{2} \le \xi,\eta \le \pi} \left| \phi_{kl}^{\;new} \right| \tag{6}$$

Consider now the Gauss-Seidel relaxation

$$N_{kl}\,\phi_{kl}^{old} + E_{kl}\,\phi_{kl}^{old} + S_{kl}\,\phi_{kl}^{new} + W_{kl}\,\phi_{kl}^{new} + C_{kl}\,\phi_{kl}^{new} = 0 \tag{7}$$

Using (5) in (7) we obtain the asymptotic smoothing rate μ_{gs} for Gauss-Seidel relaxation

$$\mu_{gs} = \max_{\frac{\pi}{2} \le \xi,\eta \le \pi} \left| \frac{N\,e^{i\eta} + E\,e^{i\xi}}{C + S\,e^{-i\eta} + W\,e^{-i\xi}} \right| \tag{8}$$

Obviously, the relaxation becomes inefficient if N >> S and E >> W. Examining explicitly the relations (4), it is found that N >> S if $v < 0$ and $|v|>1/(\alpha\Delta x)$. The inefficiency is remedied by performing the relaxation sweep in the direction of y decreasing when $v < 0$, i.e., following the flow. This is validated by the numerical results for the Benard cell problem summarized in Table 1 and the smoothing factors presented in Diagram 1. For column relaxation with increasing x, the smoothing rate is found to be

$$\bar{\mu} = \max_{\frac{\pi}{2} \leq \xi,\eta \leq \pi} \left| \frac{E}{N e^{i\eta} + S e^{-i\eta} + W e^{i\xi} + C} \right|$$

The Diagram 2 shows the region where each of N, E, S, W dominates the rest. Thus, one expects $\tilde{\mu} \approx 1$ in the upper region, and this is verified in Diagram 3. When two relaxations are performed one with x increasing and the other with x decreasing, the errors are effectively reduced all over the region. In fact, alternating row and column relaxation with changing directions is quite efficient, reducing error by 20-30%. Table 2 presents a comparison of SLOR and multigrid methods for the Benard cell problem.

TABLE 1. Efficiency of correction scheme (Benard cell problem)

Pe	Tol.	Work for plain Gauss Seidel	Work following flow
10	.01	17	22
50	.01	45	24
400	.005	200	172
400	.001	313	208

TABLE 2. Comparison of SLOR and multigrid methods (Benard cell problem)

		MULTIGRID	
Peclet No.	SLOR iterations*	Point relaxation WU**	Line relaxation WU***
10	85	38	20
20	110	35	27
40	140	44	38
80	205	83	51
160	-	101	96

GRID 25 × 25

* one SLOR iteration is a sweep in x direction, followed by a sweep in y direction.

** one WU is one sweep over whole field, following flow.

*** one WU is one sweep in the x increasing direction, followed by one sweep with x decreasing.

The Full Approximation Scheme (FAS)

The FAS is not really required for solutions of linear problems, but just as a test case it was applied to the Benard cell problem. Speed-ups of 100% or so were obtained when the relaxation was performed following the flow. Table 3 summarizes the results for this problem. The calculations were done on a finest grid of 25×25 with 4 levels, using $T = y/\pi$ as the initial data.

TABLE 3. Acceleration of FAS convergence (relaxation following flow)

Max. abs. residual	Peclet no.	Iteration WU	Iteration following flow
.001	10	48	36
	20	50	31
	40	65	36
	80	136	61
	160	211	98
.0001	10	59	47
	20	66	41
	40	84	47
	80	176	83
	100	208	64
	160	252	127

Instability

There is a trade-off in the multigrid method between the accelerated convergence usually obtained and the possible divergence of the process. In fact, simple one-level relaxation procedures for elliptic problems are always convergent--and the weighted means scheme extends this property to convection-diffusion problems. However, the multigrid algorithm introduces the possibility of divergence, which is actually met in practice. We mention here briefly some of our experience with this phenomenon.

Even in a linear problem, divergence may occur. In the Benard cell problem, at Pe = 400 the algorithm became "trapped" in the two coarsest grids. This implies that the relaxation on the coarsest grid was always convergent (considering the convergence criteria of the algorithm), while the relaxation on the next coarsest grid was never efficient. Eventually the calculation diverged to infinity. This was easily remedied by enforcing a few extra iterations on every grid--even when the algorithm would consider them inefficient--or by merely deleting the troublesome coarsest grid. The linear instability served, however, as a warning about possible complications in nonlinear cases.

For the driven cavity problem we implemented a check to guard against divergence. Namely, we introduced apriori bounds for ζ,ψ and the speeds ψ_x,ψ_y. All these bounds are easily computed using the maximum principle (see Appendix A). When such a bound is violated, there is clear indication of divergence, which we treated by returning to the finer grid, without interpolation from the coarser "incorrect" level. At this point we also changed from the correction scheme (CS) to the full approximation scheme (FAS), in order to be able to check the solution at any level and ensure it stays within bounds.

We also tried a more theoretical investigation of the divergence phenomenon. An elliptic problem--and even a convection diffusion problem with the weighted mean scheme--produces a relaxation formula with positive weights. Thus, repeated relaxation at any level is convergent, and roundoff errors are not magnified. The only point where divergence may evolve is the level change. Since we used linear interpolation from coarse to fine grids--again positive weights--it is just the fine to coarse level change which may cause instability.

Consider solving a homogeneous problem, using a two-grid scheme. We shall identify the levels by subscripts c and f for coarse and fine grid respectively. Let the discretized operators be L_c, L_f, and the relaxation operators R_c, R_f. Also, denote by I_a^b the transfer of data from level a to b. Suppose that m relaxations are performed on the fine level, and n relaxations on the coarse level. Then, an initial value u_f becomes:

$$u_f^{new} = I_c^f \; (I_f^c - (I-R_c)^n \, L_c^{-1} \, I_f^c \, L_f) \, R_f^m \, u_f$$

The operator appearing on the right hand side should have a spectral radius less than one to ensure convergence. It is obvious that by taking m large enough, this will be achieved, since

$$||R_f||<1 \text{ (also } ||R_c||<1, \; ||I_c^f||=1, \; ||I_f^c||=1).$$

But this means performing more relaxations on the fine grid, thereby losing efficiency.

Another possibility is to have the operator

$$I_f^c - (I - L_c^n) \, L_c^{-1} \, I_f^c \;\; L_f$$

small, in some suitable sense. If $n\to\infty$, i.e., coarse grid relaxations are repeated to convergence, this expression may be simplified to:

$$I_f^c - L_c^{-1} \, I_f^c \, L_f^c \, L_f = \; L_c^{-1} \; (L_c \, I_f^c - I_f^c \, L_f)$$

Now, L_c^{-1} is bounded, but the bracketed term is usually large. It is true that it involves mainly the high frequencies (where L_c and L_f differ significantly) and thus will "behave" for properly smoothed data, but smoothing involves more applications of R_f, resulting again in loss of efficiency. These results depend on the detailed expressions for I_a^b, L_c, L_f. We use injection and linear interpolation for I_a^b -- this preserves the positive weights property, and a high order of accuracy in I_f^c. If another I_f^c with positive weights is used, it may be only first order accurate, and we thought that this would conflict with the second order weighted mean scheme. At first we assumed that the very unbalanced coefficients generated by the weighted mean scheme may be a cause of instability, but the same overall behavior obtains in pure diffusion problems, also.

We also considered a one-dimensional, advection-diffusion equation with constant coefficient, expecting to obtain more insight into the interplay of various parameters, by explicitly computing spectral radii and norms (see Appendix B).

A simple Fourier mode analysis-setting $\phi \sim e^{i\omega x}$ - shows many virtually increasing amplitudes. These are reduced only by repeated relaxation on the fine grid. However, the analysis may be misleading, because unrealistic periodic boundary conditions are assumed. If Dirichlet conditions are imposed, then the spectral radii stay below 1, for all m, n, speed and diffusivity values (the spectral radii had to be obtained numerically). This implies stability and convergence of the multigrid process.

Some of these results are presented in Appendix B. One general property that may be deduced is that relaxation against the flow is always inefficient and possibly destabilizing. One broad conclusion may be stated since there are initially growing modes, proper care is needed in nonlinear problems.

CONCLUSIONS

Efficient and accurate results are obtained by the correction scheme and full approximation scheme algorithms for the problem of Benard convection in a square cell. For the driven cavity problem, preliminary results have been obtained for Reynolds numbers of the order of 1000. The multigrid technique for this problem requires further refinement, with particular reference to interpolation procedures and grid switching criteria.

Diagram 1. Gauss-Seidel Smoothing Rate* 100

57	84	95	96	97	97	97	97	97	96	96	95	95	94	93	91	88	84	77	55
43	87	97	98	98	98	98	97	97	96	96	96	96	96	96	95	95	94	92	81
44	88	98	99	99	99	99	98	97	96	96	97	97	98	98	99	99	99	98	91
44	86	98	99	99	99	99	99	98	96	96	97	98	99	99	99	99	99	98	94
44	84	97	99	99	99	99	99	98	96	96	98	99	99	99	99	99	99	98	95
44	81	95	99	99	99	99	99	98	96	96	98	99	99	99	99	99	99	98	95
44	77	93	98	99	99	99	99	98	95	95	98	99	99	99	99	99	99	98	96
44	70	87	94	97	99	99	99	98	93	94	98	99	99	99	99	99	98	98	96
44	61	75	85	91	94	96	97	96	89	92	98	99	99	99	98	98	97	97	96
44	50	56	61	66	70	73	75	74	67	79	92	95	96	96	96	96	96	96	96
44	38	33	28	24	21	19	17	16	20	67	89	93	95	95	96	96	96	96	96
44	27	15	8	5	3	2	1	1	16	75	96	98	98	98	97	97	96	96	96
44	18	6	2	0	0	0	0	1	17	75	97	99	99	99	99	98	97	95	96
44	12	2	0	0	0	0	0	2	19	73	96	99	99	99	99	99	97	95	95
44	8	1	0	0	0	0	0	3	21	70	95	99	99	99	99	99	98	95	95
44	5	0	0	0	0	0	1	6	25	67	92	98	99	99	99	99	98	94	94
44	4	0	0	0	0	1	3	10	28	62	87	96	99	99	99	99	98	93	92
43	4	1	1	1	2	4	9	17	33	56	78	90	96	98	99	99	97	89	89
41	16	8	9	10	13	16	21	29	38	50	63	74	82	88	92	93	91	79	80
55	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	42	44	57

Diagram 2. Column Relax. Increasing X Dominant Coefficient
e.g., N > 0.8 (N+E+S+W)

```
..EEEEEEEEEEEE......
...EEEEEEEEEEE......
.....EEEEEEEE.......
.......EEEEE......NN
S.......EEE.......NN
S........EE......NNN
SS........E......NNN
SSS.............NNNN
SSSS...........NNNNN
SSSSSS.......NNNNNNN
SSSSSSS.......NNNNNN
SSSSS...........NNNN
SSSS.............NNN
SSS.......W.......NN
SSS......WW........N
SS.......WWW.......N
SS......WWWWW.......
.......WWWWWWWW.....
......WWWWWWWWWWW...
......WWWWWWWWWWWW..
```

Diagram 3. Column Relax. Increasing X - Smoothing Rate* 100

51	89	98	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	96	44
44	89	98	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	95	47
44	88	98	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	95	48
44	87	98	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	94	46
44	84	97	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	92	44
44	81	95	99	99	99	99	99	99	99	99	99	99	99	99	99	99	98	88	44
44	77	93	98	99	99	99	99	99	99	99	99	99	99	99	99	99	96	82	44
44	70	87	94	97	99	99	99	99	99	99	99	99	99	99	98	96	90	74	44
44	61	75	85	91	94	96	97	97	98	98	98	97	96	95	92	87	78	63	44
44	50	56	61	66	70	73	75	77	77	78	77	75	73	70	67	62	56	50	44
44	38	33	28	24	21	19	17	16	16	16	18	20	23	25	27	30	33	38	44
44	27	15	8	5	3	2	1	1	2	3	1	2	3	4	7	12	20	30	44
44	18	6	2	0	0	0	0	0	0	0	0	0	0	0	1	4	11	25	44
44	12	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	20	44
44	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	15	44
44	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	12	46
44	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10	48
44	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9	48
44	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	18	46
44	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	33	51

APPENDIX A

Bounds for the Driven Cavity

Suppose, for simplicity, that the cavity is unit square, then

$$0.08\zeta_{min} \le \psi \le 0.08\zeta_{max}$$

$$-\ 0.08\zeta_{x\ max} \le v = -\psi_x \le -\ 0.08\zeta_{x\ min}$$

$$0.08\ (\zeta_{y\ min}-U) + U \le u = \psi_y \le 0.08\ (\zeta_{y\ max} - U) + U$$

where U is the imposed speed on the top boundary. The constant 0.08 in the maximum of ϕ, which solves:

$$\nabla^2\phi + 1 = 0$$

$\phi = 0$ on the boundary

For ζ, one uses the bound

$$\zeta_{\text{min on boundary}} \le \zeta_{\text{interior}} \le \zeta_{\text{max on boundary}}$$

APPENDIX B.

Initial Mode Amplification

We discuss here briefly the behavior of the weighted mean scheme solution of the problem:

$$U\phi_x = \sigma\phi_{xx}\ , \qquad (\sigma = 0.01)$$

$$\phi(0) = \phi(1) = 1$$

with the initial guess:

$$\phi_k = e^{i\omega k} \qquad (0<\omega<\pi,\ 0\le k<N)$$

Using N = 2M subintervals on the fine grid, we perform m fine relaxations. (Gauss-Seidel with x increasing), then m coarse relaxations, and record the largest magnitude of the resulting ϕ as a function of ω. This is maximum at $\omega=\pi$, but even at other values of ω (e.g., $\omega=\pi/4$) this quantity may exceed 1.

When a full multigrid procedure is implemented, there is always convergence to $\phi=1$ - the modes, which initially grow, decay subsequently. Comparison of results at one given speed U - columns d, e, f, (say) of Table 4 - shows that a large ratio of coarse iterations versus fine iterations produces high amplification. It is to be noted that columns d, e, f concern a pure diffusion problem; the weighted mean scheme reduces to simple central differencing, and yet there are some growing modes. Relaxation sweep against the flow (U=-1, columns a, b, c) consistently produces the worst amplification. Similar results may be derived by the Fourier mode analysis.

TABLE 4. Initial Amplification of $\phi_k = e^{i\omega k}$

Speed U	-1			0			1		
Fine/coarse relax.	5/1	1/1	1/5	5/1	1/1	1/5	5/1	1/1	1/5
$\omega=\pi$ M=5	2.4966	2.4972	3.0774	1.0000	5.2891	5.7665	1.0000	1.0004	1.0004
19	2.8570	4.8923	20.9650	1.0000	5.6666	26.9670	1.0015	3.6451	3.0393
33	1.1244	4.8289	17.9545	1.0000	5.6666	27.0000	1.0013	9.7055	19.0032
47	1.0215	4.8735	22.0000	1.0000	5.6666	27.0000	1.0000	9.0426	39.0100
$\omega=\pi/y$ M=5	2.2182	2.2186	2.5088	1.0000	1.2732	1.4704	1.0000	1.0000	1.0000
19	1.6508	2.1982	2.2339	1.0000	1.4022	1.2859	1.0015	1.0016	1.0026
33	1.0769	2.0962	2.1007	1.0000	1.4103	1.2911	1.0016	1.0000	1.0250
47	1.0193	1.9888	1.9674	1.0000	1.4122	1.2925	1.0000	1.0000	1.0564
	a	b	c	d	e	f	g	h	i

REFERENCES

1. Allen, D. N.; and Southwell, R. V.: Relaxation Methods Applied to Determine the Motion, in Two Dimensions, of a Viscous Fluid Past a Fixed Cylinder, Quart. J. Mech. and Appl. Meth., Vol VIII, Part 2, 1955, pp. 129-145.

2. Gresho, p.; and Lee, R. L.: Don't Suppress the Wiggles--They Are Telling You Something, The Winter Annual Meeting of the American Society of Mechanical Engineers, New York, N. Y., December 2-7, 1979.

3. Spalding, D. B.: A Novel Finite-Difference Formulation for Differential Expressions Involving Both First and Second Derivatives, Inter. J. Num. Meth. Engg., Vol. 4, 1972, pp. 551-559.

4. Raithby, G. D.: Skew Upstream Differencing Schemes for Problems Involving Fluid Flow, Computer Meth. Appl. Mech. Engg., Vol. 9, 1976, pp. 153-164.

5. Fiadeiro, M.; and Veronis, G.: On Weighted Mean Schemes for the Finite Difference Approximation to the Advection-Diffusion Equation, Tellus, Vol. 29, 1977, pp. 512-522.

6. Raithby, G. D.; and Torrance, K. E.: Upstream-Weighted Differencing Schemes and Their Application to Elliptic Problems Involving Fluid Flow, Computers and Fluids, Vol. 2, 1974, pp. 191-200.

7. Brandt, Achi: Multilevel Adaptive Solutions to Boundary Value Problems, Math. Comp. Vol. 31, 1977, pp. 333-390.

A MULTIGRID METHOD FOR THE TRANSONIC FULL POTENTIAL EQUATION DISCRETIZED WITH FINITE ELEMENTS ON AN ARBITRARY BODY FITTED MESH

Herman DECONINCK Charles HIRSCH
Vrije Universiteit Brussel , Department of Fluid Mechanics

ABSTRACT

A multigrid method for the acceleration of transonic potential flow calculations based on a Galerkin Finite Element approach is described. In order to allow the use of arbitrary body fitted meshes it is necessary to introduce non uniform interpolation and residual weighting. Emphasis is put on the construction of these operators consistent with the Finite Element approximation, while standard successive line overrelaxation is used as smoothing step. Substantial convergence acceleration is obtained and results are presented for different transonic flow configurations including shocks.

INTRODUCTION

The multigrid method was originally introduced for the solution of the system of equations obtained from the finite difference (F.D.) discretization of elliptic partial differential equations by Fedorenko (ref.1), extended by Bakhalov (Ref. 2) and further developed by Brandt (ref. 16). It is based on the idea that corrections for the solution on a fine grid can be effectively approximated on a coarse grid with help of the common underlying differential equation.

Finite Element (F.E.) applications were soon recognized and at the present time the mathematical foundations are even better established than in the F.D. case although practical implementations are rare. Convergence proofs under fairly general conditions for elliptic boundary value problems were obtained by Nicolaides (refs. 3,4), Hackbusch (ref. 5) and others. One of the basic conclusions of these investigations is that the convergence of the multigrid methods is independent of the step size and that the amount of computational work for solving the discrete system of n unknows is proportional to n. Practical aspects of the F.E. implementation on model problems are given in Brandt (ref. 6) and Nicolaides (ref. 7) who describes extensive numerical results obtained for a Poisson equation and another elliptic equation with variable coefficients and mixed boundary conditions, both on a uniformly discretized rectangular domain. These results confirm the convergence rates obtained with Finite Differences.

In transonic flow computations, the first multigrid solutions have been proposed by South and Brandt (ref. 8) with the transonic small perturbation equation and successive line relaxation (SLOR) as smoothing operator. Problems were encountered in the treatment of the boundary conditions and in calculations on nonuniform and curvilinear grids, probably due to a lack of smoothing on a fine grid before passing to a coarser one or due to an unsatisfactory residual weighting. Jameson (ref. 9) solved the transonic full potential equation on an arbitrary mesh and obtained very satisfying results with a generalized ADI as smoothing step. As in most other applications these multigrid methods are implemented on a rectangular (or circular) uniform computational mesh obtained from a mapping of the original physical curvilinear mesh allowing uniform interpolations. This approach is appropriate in cases where the physical problem and boundary conditions are transformed by a global coordinate transformation as in most finite difference methods. However, the classical F.E. approach handles the problem in the physical plane and uses only a local mapping of each curvilinear element to a reference parent element to facilitate the volume integrations needed in the computation.

Simple uniform interpolation is only obtained if the fine mesh elements are uniform subdivisions of a coarse grid element. This would pose a severe limit on the finest mesh that can be achieved since only the mesh points of the coarsest mesh could be chosen in an arbitrary way. Therefore non uniform interpolation and residual weighting is introduced in this paper preserving the same flexibility with respect to the geometry as the usual F.E. methods. An advantage of the F.E. treatment is that the method leads to natural choices for the interpolation and weighting, even on the boundaries of the domain.

Indeed, a simple but artificial residual injection following the lines of F.D. methods has been tried with poor results confirming the observations of Nicolaides (ref. 7) on a simple rectangular domain. It turns out that the amount of additional work due to the non uniformity is reduced due to the fact that the same numerical coefficients are needed for coarse to fine interpolations as for the fine to coarse weighting.

In the present investigation successive line relaxation with downstream sweep direction is used as smoothing component. Alternatively this smoothing operator can be replaced by the F.E. ADI method developed in the past (ref. 10).

Numerical experiments on channel, single airfoil and cascade geometries indicate a substantial convergence acceleration compared to the grid refinement technique which consists in the application of SLOR to successively finer grids with the previous coarse grid solution as initial approximation.

EQUATION AND F.E. APPROXIMATION WITH ISOPARAMETRIC ELEMENTS

A brief account of the F.E. treatment is given here. More details can be found in previous publications and the references contained there in (refs. 13, 14, 15).

The potential equation in conservative form is given by

$$\partial_x(\rho\ \phi_x) + \partial_y(\rho\ \phi_y) = 0 \tag{1}$$

where x and y are the Cartesian coordinates in the physical plane and ϕ_x, ϕ_y the velocity components.

The density ρ is obtained from the isentropic relation

$$\rho = \rho_t [1 - \frac{\gamma - 1}{\gamma\ r\ T_t} (\phi_x^2 + \phi_y^2)]^{1/\gamma-1} \tag{2}$$

where ρ_t and T_t are stagnation density and temperature and γ the ratio of specific heats.

In transonic flow regime equation (1) is mixed elliptic-hyperbolic and allows different weak solutions for a given set of boundary conditions. If proper viscosity terms are added to the equation a unique solutions is again guaranteed which is equal to the physical solution except for a small region around shocks (ref. 11).

The artificial density form of the artificial viscosity terms, due to Hafez, Murman and South (ref. 12) is particularly well suited for F.E. applications and works satisfactorily for flows with Machnumbers up to 1.5 (refs. 14, 15). It is obtained by giving an upwind bias to the density which is replaced by

$$\tilde{\rho} = \rho - \mu\ \rho_{\overleftarrow{s}}\ \Delta s \tag{3}$$

where $\rho_{\overleftarrow{s}}$ is the upwind derivative of ρ along the streamwise direction s, Δs the meshspacing and μ a switching function with cut-off Machnumber M_c which controls the amount of artifical viscosity

$$\mu = \max\ (0\ ,\ 1 - \frac{M_c^2}{M^2}) \tag{3b}$$

A Finite Element weighted residual approach is based on the weak formulation of (1) given by

$$R(\phi) = \int_S \tilde{\rho}\ \nabla W\ \nabla\phi\ dS - \oint_s W\ \tilde{\rho}\ \frac{\partial\phi}{\partial n}\ ds = 0 \tag{4}$$

for any continuous testfunction W, where S is the physical flowdomain with boundary s. The functional $R(\phi)$ is called residual. The integral over the boundary is the expression of the Neumann boundary conditions (B.C.) which are part of the problems specification. Three types of geometry are considered each giving different specific Neuman B.C. : channel geometry, single airfoil and cascade geometry : Channel walls and blade or profile boundaries require the no flux condition

$$\tilde{\rho}\ \frac{\partial\phi}{\partial n} = 0$$

Points belonging to periodic boundaries in cascade geometries are treated as interior points by letting coincide corresponding periodic points (ref. 14). At inlet and outlet boundaries either the solution is given (Dirichlet condition) or the mass flow rate $\tilde{\rho}(\partial\phi/\partial n)$ is specified directly or in an iterative way by applying a Kutta Youkowski condition at the trailing edge while the far field condition for the single airfoil geometry is also introduced by forcing the known massflow rate trough the far field boundary.

A F.E. approximation of a function $\phi(x,y)$ is obtained by defining a finite dimensional space S^h with basis functions $N^h_{ij}(x,y)$ attached to a set of meshpoints (i,j) spread over the flow domain S :

$$(5) \qquad \phi^h(x,y) = \sum_{i,j} \phi^h_{ij}\, N^h_{ij}(x,y)$$

where ϕ^h_{ij} are the meshpoint values of ϕ^h and h the typical mesh size characteristic of the space S^h. It follows from (5) that

$$(6) \qquad N^h_{ij}(x^h_{kl},y^h_{kl}) = \delta^{kl}_{ij} = \begin{cases} 1 \text{ for } (i,j) = (k,l) \\ 0 \text{ otherwise} \end{cases}$$

A discrete Galerkin approximation for the weak form (4) is found by taking a finite number of testfunctions W, namely the basisfunctions of space S^h, giving the following non linear system of equations for the meshpoint values

$$(7) \qquad R^h_{ij} = \sum_{k,l} \phi^h_{kl}\, K^{kl}_{ij}(\phi^h) - f^h_{ij} = 0$$

where $K(\phi^h)$ is the stiffness matrix and f^h the contribution of the Neuman B.C.

$$(8) \qquad K^{kl}_{ij}(\phi^h) = \int_S \tilde{\rho}(\phi^h)\, \nabla N^h_{kl}\, \nabla N^h_{ij}\, dS \quad \text{and} \quad f^h_{ij} = \int_S \tilde{\rho}\, \frac{\partial\phi}{\partial n}\, N^h_{ij}\, ds$$

It is well known that exactly the same expression for the residual is found by solving the discrete minimization problem in S^h in cases where a minimum principle equivalent to the equation can be formulated (as in the fully elliptic subsonic case).

Expression (7) for the residual is developed in the physical plane and written in physical coordinates and can be evaluated for any trial function ϕ after a choice of the type of element has been made which determines the type of basisfunctions of the space S^h. In reference 15 bilinear and biquadratic Lagrange elements have been used, the latter allowing third order accuracy and parabolic approximation of the boundaries. With these elements the integrations over an element surface (eq. 8) are usually carried out with Gauss quadrature after transformation of the arbitrarily shaped element to a unit square. In the standard F.E. treatment this transformation is the locally defined isoparametric mapping :

$$(9.a) \qquad x^h(\xi^h,\eta^h) = \sum_{i,j} x^h_{ij}\, N^h_{ij}(\xi^h,\eta^h)$$

(9.b) $$y^h(\xi^h,\eta^h) = \sum_{i,j} y^h_{ij}\, N^h_{ij}(\xi^h,\eta^h)$$

which is completely determined by the mapping of the meshpoints of the space S^h causing arbitrarily located meshpoints of the grid $S^{h/2}$ not to be mapped uniformly in the (ξ^h,η^h) plane.

The discrete non linear system (eq. 7) has been solved with the usual iterative methods such as successive line overrelaxation (SLOR) and approximate factorization (ADI) for which a F.E. version was developed (ref. 10). The simple SLOR method is reliable but extremely slow due to the fact that it elimates effectively only the errors with wavelength comparable to the meshwidth h. Substantial convergence acceleration was achieved by solving the series of N+1 problems

(10) $$R^{2^n h}_{ij} = 0 \qquad\qquad n=N\ ,\ N-1\ ,\ \dots\ ,\ 1\ ,\ 0$$

defined in the space $S^{2^n h}$ where the errors of wavelength $2^n h$ are eliminated effectively and the computational effort reduced.

In this grid refinement technique the influence of the coarse meshes is only sensible trough the initial approximation for the next finer mesh, while in the full multigrid approach described subsequently the coarse grid equations are modified in order to represent meaning full approximations of the fine grid corrections.

MULTIGRID ALGORITHM

The multigrid approach is based on a different treatment of low and high frequency errors in the approximate solution : the high frequency error components can only be resolved on a fine grid and are fortunately eliminated efficiently by existing relaxation techniques. Low frequency components on the other hand are nearly unaffected by relaxation but they are scaled with the dimensions of the physical domain and hence can be eliminated on a coarser grid where the computational effort is lower and the propagation of corrections trough the domain much more rapid.

Considering the system of non linear equations (eq. 7) constructed on the finest mesh with characteristic spacing h :

(11) $$R^h(\phi^h) = K^h(\phi^h) - f^h = 0$$

which may be written in correction form with respect to a known approximate solution ϕ^h_n :

(12) $$\hat{K}^h(\delta\phi^h) = K^h(\phi^h_n + \delta\phi^h) - K^h(\phi^h_n) = -\ R^h(\phi^h_n)$$

where the unknowns are now the corrections $\delta\phi^h$ given by :

(13) $$\phi^h = \phi^h_n + \delta\phi^h$$

Supposing that the high frequency errors have been eliminated effectively by means of a smoothing operation such as SLOR or ADI, the correction $\delta\phi^h$ and residual $R^h(\phi^h)$ may be considered as smoothly varying quantities for which an approximation on a coarser grid makes sense. This mesh with typical spacing 2h is obtained by dropping the odd numbered coordinate lines of the mesh S^h and an updated approximation ϕ^h_{n+1} can be calculated according to (eq. 13) by interpolating the coarse grid approximation $\delta\phi^{2h}$ for $\delta\phi^h$ back to the original mesh

(14) $$\phi^h_{n+1} = \phi^h_n + I^h_{2h}\ \delta\phi^{2h}$$

where I^h_{2h} is the coarse to fine grid function interpolation operator called "prolongation". The coarse grid approximation $\delta\phi^{2h}$ for the fine grid correction is the solution of the following equation on the coarse mesh :

(15) $$\hat{K}^{2h}(\delta\phi^{2h}) = - {}^{R}I^{2h}_h\ R^h(\phi^h_n)$$

The fine to coarse residual restriction operator ${}^{R}I^{2h}_h$ constructs a meaningfull approximation of the coarse grid residual R^{2h} based on the smoothly varying fine grid residuals.

By defining a coarse grid solution ϕ^{2h} as the approximation of ϕ^h on the coarse grid :

(16) $$\phi^{2h} = I^{2h}_h\ \phi^h_n + \delta\phi^{2h}$$

where I^{2h}_h is the function restriction, eq. 15 takes again the usual form of eq. 11 :

(17) $$K^{2h}(\phi^{2h}) = f^{2h}$$

where the right hand side is a known function of the fine grid approximate solution :

(18) $$f^{2h} = - {}^{R}I^{2h}_h\ R^h(\phi^h_n) + K^{2h}(I^{2h}_h\ \phi^h_n)$$

and $\delta\phi^{2h}$ can be eliminated from the updating formula (14) by means of (16) :

(19) $$\phi^h_{n+1} = \phi^h_n + I^h_{2h}(\phi^{2h} - I^{2h}_h\ \phi^h_n)$$

The solution ϕ^{2h} in turn can be approximated on the mesh S^{4h} when it is sufficiently smooth i.e. the whole procedure can be applied in a recursive way to eq. (17). This non linear algorithm (F.A.S. scheme) is due to Brandt (ref. 16) who describes an adaptive strategy for the transition to a coarser or finer grid depending on the convergence level and speed on a particular grid. A more simple fixed strategy has been used in the present work (ref. 7) : Starting on the finest mesh with spacing h one line overrelaxation sweep is performed followed by the transition to the next coarser grid by means of eqs. 17 and 18 until the coarsest grid is reached. On the coarsest grid some additional relaxation sweeps are performed and the solution of the next finer grid is updated by means of eq. 19 followed by one relaxation step until the second finest grid is reached. The cycle terminates with the updating of the finest grid approximate solution with help of eq. 19.

As distinct from F.D. approaches the interpolation operators I_h^{2h}, I_{2h}^h and ${}^R I_h^{2h}$ are not arbitrary but based on the F.E. interpolation spaces S^h and S^{2h}. They are considered in some more detail in the following sections.

THE COARSE TO FINE GRID FUNCTION INTERPOLATION : OPERATOR I_{2h}^h

The only natural choice for the interpolation of a coarse mesh function ϕ^{2h} to a fine mesh location (x_{ij}^h, y_{ij}^h) is to use the value of ϕ^{2h} in the location (x_{ij}^h, y_{ij}^h) given by the F.E. approximation in space S^{2h} :

$$[I_{2h}^h \; \phi^{2h}]_{ij} = \phi^{2h}(x_{ij}^h, y_{ij}^h) = \sum_{k,l} I_{kl}^{ij} \; \phi_{kl}^{2h} \tag{20}$$

where the matrix I_{kl}^{ij} is given by

$$I_{kl}^{ij} = N_{kl}^{2h}(x_{ij}^h, y_{ij}^h) \tag{21}$$

On an arbitrary mesh this results in non uniform interpolation coefficients I_{kl}^{ij} and for instance with bilinear elements (figure 1) uniform interpolation is only obtained if the fine grid meshpoints are situated in the middle of the coarse grid element sides and in the center giving only in this case the simple formula (figure 1a) :

$$[I_{2h}^h \; \phi^{2h}]_C = \frac{1}{4} \left(\phi_1^{2h} + \phi_2^{2h} + \phi_3^{2h} + \phi_4^{2h}\right) \quad \text{for the centernode}$$

$$[I_{2h}^h \; \phi^{2h}]_M = \frac{1}{2} \left(\phi_i^{2h} + \phi_j^{2h}\right) \quad \text{for the midside node i-j} \tag{22}$$

$$[I_{2h}^h \; \phi^{2h}]_i = \phi_i^{2h} \quad \text{for the corner nodes (identity)}$$

It follows that simple uniform interpolation is only possible for uniform refinements of the coarsest mesh which could be chosen arbitrarily.

In the general case with bilinear elements (figure 1b) four coefficients are needed for each fine grid meshpoint not coinciding with a coarse grid meshpoint. The computation of these general coefficients (eq. 21) is not trivial since $N_{kl}^{2h}(x,y)$ is not explicitly known for an arbitrarily shaped element and one has first to invert the isoparametric transformation (eq. 9) to obtain ξ_{ij}^h and η_{ij}^h from :

$$x_{ij}^h = \sum_{m,n} x_{m,n}^{2h} \; N_{m,n}^{2h}(\xi_{ij}^h, \eta_{ij}^h) \tag{23.a}$$

$$y^h_{ij} = \sum_{m,n} y^{2h}_{m,n}\, N^{2h}_{m,n}(\xi^h_{ij},\eta^h_{ij}) \tag{23.b}$$

after which the computation is carried out in the ξ-η plane where the basis-functions are simple polynomial expressions :

$$I^{ij}_{kl} = N^{2h}_{kl}(x^h_{ij},y^h_{ij}) = N^{2h}_{kl}(\xi^h_{ij},\eta^h_{ij}) \tag{24}$$

With the bilinear elements for instance $N^{2h}(\xi,\eta)$ is of the form

$$N^{2h}(\xi,\eta) = \frac{1}{4}\,(1\pm\xi)(1\pm\eta) \tag{25}$$

when the four corner points are situated at $(\xi,\eta) = (\pm1,\pm1)$

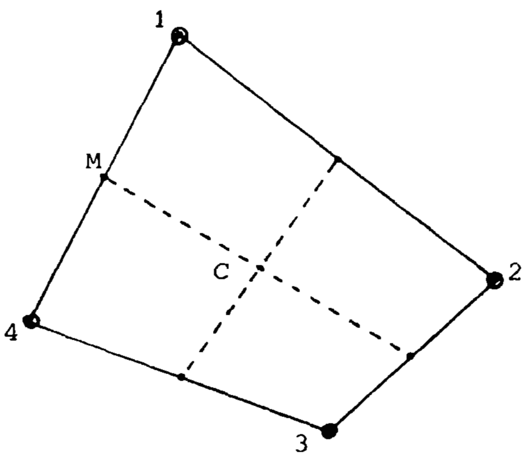


Fig. 1a : Uniform interpolation

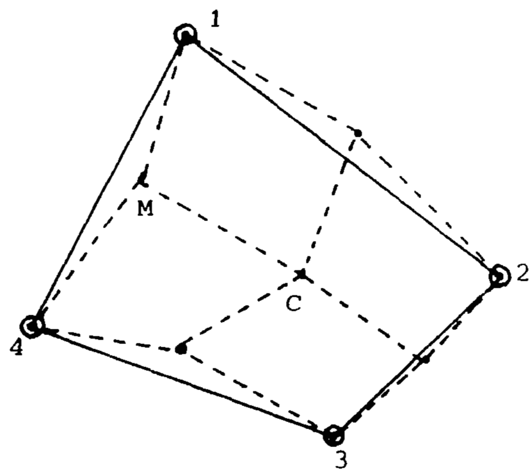


Fig. 1b : Non uniform interpolation

FINE TO COARSE GRID FUNCTION INTERPOLATION (RESTRICTION) : OPERATOR I_h^{2h}

The value of ϕ^h in the coarse mesh location calculated with the F.E. approximation in S^h leads to the identity since the coarse gridpoints belong also to the fine grid.

$$[I_h^{2h}\,\phi^h]_{ij} = \phi^h(x^{2h}_{ij},y^{2h}_{ij}) = \sum_{k,l} \phi^h_{kl}\, N^h_{kl}(x^{2h}_{ij},y^{2h}_{ij}) \tag{26}$$

which due to (eq. 6) reduces to

$$[I_h^{2h}\,\phi^h]_{ij} = \phi^h_{ij} \tag{27}$$

This type of restriction is sometimes called injection.

FINE TO COARSE GRID INTEGRAL INTERPOLATION : OPERATOR ${}^{R}I_h^{2h}$

As distinct from the F.D. case the residual is an integral quantity which is scaled differently on different grids. It cannot be represented in the spaces S^h and S^{2h} and the previous interpolation rules are inapplicable. In the Galerkin approach the volume integrals are always of the form

$$[R^{2h}(\phi^{2h})]_{ij} = \int_S N_{ij}^{2h}\, g(\phi^{2h})\, dS \tag{28}$$

For instance the residual in eq. 7 can be rewritten in this form with :

$$g(\phi^{2h}) = \nabla(\rho(\phi^{2h})\, \nabla\phi^{2h}) \tag{29}$$

A consistent representation of R_{ij}^{2h} by means of fine grid quantities is found by approximating the coarse mesh functions in the integrant of (eq. 28) with fine mesh interpolations in the space S^h, namely

$$[{}^{R}I_h^{2h}\, R^h]_{ij} = \int_{S_{ij}^{2h}} I_h^{2h}\, N_{ij}^{2h}\;\; g(I_h^{2h}\, \phi^h)\, dS \tag{30}$$

where S_{ij}^{2h} is the coarse mesh residual integration domain, i.e. the part of S where $N_{ij}^{2h} \neq 0$ (figure 2).

The interpolation of $\phi^h(x,y)$ to the coarse mesh leads again to the identity since the interpolation of meshpoint values is the identity by virtue of eq. 27 :

$$I_h^{2h}\, \phi^h(x,y) = \sum_{k,l} N_{kl}^h(x,y)\, I_h^{2h}\, \phi_{kl}^h = \sum_{k,l} N_{kl}^h(x,y)\, \phi_{kl}^h = \phi^h(x,y) \tag{31}$$

In the same way the coarse mesh basisfunction $N_{ij}^{2h}(x,y)$ is approximated in the space S^h :

$$I_h^{2h}\, N_{ij}^{2h}(x,y) = \sum_{k,l} N_{kl}^h(x,y)\, I_h^{2h}\, N_{ij}^{2h}(x_{kl}^h, y_{kl}^h) \tag{32}$$

which due to eq. 27 and 21 leads to

$$I_h^{2h}\, N_{ij}^{2h}(x,y) = \sum_{k,l} N_{kl}^h(x,y)\, I_{ij}^{kl} \tag{33}$$

The final expression for the coarse mesh residual weighting by means of fine grid quantities is obtained from eq. 30 by inserting the expressions (31) and (33) :

$$[{}^{R}I_h^{2h}\, R^h]_{ij} = \sum_{k,l} I_{ij}^{kl} \int_{S_{ij}^{2h}} N_{kl}^h(x,y)\, g(\phi^h)\, dS \tag{34}$$

On a uniformly subdivided coarse mesh (figure 2a) it is clear that this general expression reduces to

(35) $$[{}^{R}I_h^{2h}\, R^h]_{ij} = \sum_{k,l} I_{ij}^{kl}\, R_{kl}^h$$

where the summation extends only over the 9 inner points in the domain S_{ij}^{2h} since the coefficients I_{ij}^{kl} are zero on and outside the boundaries of S_{ij}^{2h}.

Comparing eqs. 20 and 35 one concludes that the coarse to fine mesh interpolation I_{2h}^h is the adjoint of the residual weighting ${}^{R}I_h^{2h}$ since they have transposed coefficient matrices :

(36) $$[I_{2h}^h\, \phi^{2h}]_{ij} = \sum_{k,l} I_{kl}^{ij}\, \phi_{kl}^{2h}$$
$$[{}^{R}I_h^{2h}\, R^h]_{ij} = \sum_{k,l} I_{ij}^{kl}\, R_{kl}^h$$

The following result is obtained for uniform subdivisions (figure 2a), which corresponds to the uniform interpolation (22)

(37) $$[I_h^{2h}\, R^h]_{ij} = R_{ij}^h + \frac{1}{2}(R_{i,j+1}^h + R_{i,j-1}^h + R_{i-1,j}^h + R_{i+1,j}^h) + \frac{1}{4}(R_{i+1,j+1}^h + R_{i+1,j-1}^h + R_{i-1,j+1}^h + R_{i-1,j-1}^h)$$

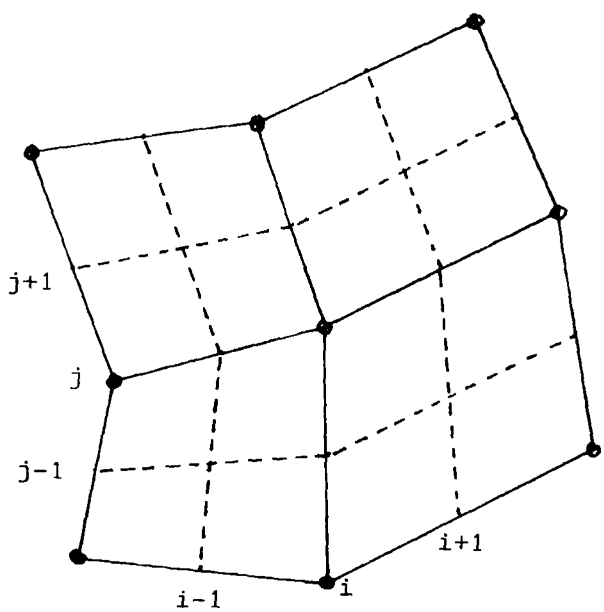


Fig. 2a : S_{ij}^{2h} uniformly subdivided mesh

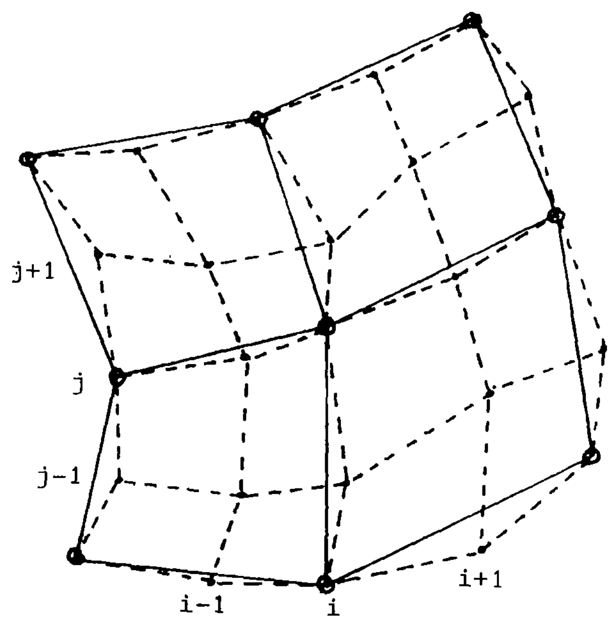


Fig. 2b : Arbitrarily subdivided mesh

On an arbitrarily subdivided mesh (figure 2b) the situation is different due to non overlapping integration domains for the coarse and fine mesh. If one is willing to apply the formula (35) with summation over the 9 innermost meshpoints in figure 2b and with the correct non uniform coefficients, two sources of errors are introduced with respect to the exact formula (34) :

- First the contributions of points $(i\pm1,j\pm2)$ and $(i\pm2,j\pm1)$ lying inside the coarse residual integration domain are omitted. For mildly distorted grids their contributions are negligible since the coefficients I_{ij}^{kl} are small near and zero on or outside the limits of S_{ij}^{2h} and also due to the fact that the integral in eq. 34 extends over only two fine mesh elements compared to 4 for the other points.
- Secondly the fine mesh integration domain for points $(i\pm1,j\pm1)$, $(i,j\pm1)$ and $(i\pm1,j)$ are not always completely contained in the coarse mesh domain S_{ij}^{2h}. Again the errors are small since the surface differences are small and more over since the integrants in eq. 34 approach zero near the limits of the fine mesh integration domain.

In conclusion, eq. 35 remains an extremely valuable approximation for eq. 34 in the arbitrary mesh case, of course only when used with the arbitrary mesh interpolation coefficient I_{ij}^{kl} already known from the non uniform interpolation.

It remains equally valid on the Neumann boundaries of the physical domain where the summation extends over 6 fine meshpoints and 4 for boundary corners.

The same expression derived here was also obtained for orthogonal meshes by Nicolaides (ref. 7) and Brandt (ref. 6) based on the minimization approach. Brandt suggests that this "natural" choice is not always better than the residual injection which is simply given by :

(38) $$[{}^R I_h^{2h}\, R^h]_{ij} = 4\, R_{ij}^h$$

in the uniform case.

This has not been confirmed by Nicolaides in his numerical experiments on a square uniform domain. On an arbitrary mesh residual injection could be constructed with some theoretical support by supposing the function $g(\phi^h)$ constant over the coarse mesh residual integration domain allowing the following approximations :

(39) $$R_{ij}^{2h} \approx g(\phi^{2h}) \int N_{ij}^{2h}\, dS$$
$$R_{ij}^{h} \approx g(\phi^{h}) \int N_{ij}^{h}\, dS$$

and hence

(40) $$[{}^R I_h^{2h}\, R^h]_{ij} = R_{ij}^h \frac{\int N_{ij}^{2h}\, dS}{\int N_{ij}^{h}\, dS}$$

giving exactly expression (38) in the uniform case.

Computational experience with eq. 40 was highly unsatisfactory and showed that it is inapplicable, at least with the simple smoothing procedure used in this paper.

COMPUTATIONAL RESULTS

The convergence of the computation is measured by the evolution of the average residual on the finest grid in terms of the work count which is defined in units representing the work needed for one line relaxation sweep on the finest mesh. For instance, the work count of one complete multigrid cycle with four grids and for the present strategy is given by

$$1 + 2\left(\frac{1}{4} + \frac{1}{16}\right) + \frac{10}{64} = 1{,}91 \text{ units}$$

plus the additional work for the residual weighting and other interpolations. The convergence rate as used below is defined as the mean reduction in the average residual per unit of work. An initial approximate solution on the finest grid for the multigrid iteration is calculated by applying the grid refinement technique with five relaxation sweeps per grid.

All computations are carried out on a fine mesh with 73 x 25 meshpoints and successive coarser meshes of 37 x 13, 19 x 7 and 10 x 4 meshpoints. Three sets of testcases are presented with different geometric boundary conditions.

The first set is the non lifting NACA 0012 single airfoil configuration for which the mesh generation method of ref. 17 was adopted, however leaving out the symmetric lower half part of the mesh since only symmetric non lifting flows can be treated with the present code which is primarily intended for cascade flow computations.

With a free stream Machnumber of .80 the standard workshop mesh (ref.17) was used. In figure 3 the pressure distribution with a shock of moderate strength is compared with the results obtained by other participants showing good agreement. The evolution of the average residual is given in figure 5 where the influence of the number of grids is apparent. The convergence rate is improved from .967 for 2 grids to .900 for 4 grids. The high speed with which the flow pattern is established is illustrated on figure 4 where the pressure distribution obtained after 1, 2, 4, 7, 10 and 13 multigrid cycles is shown. The solution is converged after 10 cycles except for a small overshoot ahead of the shock which is suppressed after 13 cycles, when the average residual is still only reduced by $2\ 10^{-2}$.

The residual evolution for the grid refining method is also plotted on figure 5, beginning at 50 work units which is the amount of work carried out on the coarse grids before passing to the final mesh. A very fast initial reduction of the mean residual is seen which corresponds to a fast suppression of the high frequency errors by the consecutive relaxation sweeps. The remaining low frequency errors are not eliminated and cause the convergence to slow down after a small number of relaxation steps.

Figure 7 shows the solution for the flow at Mach .85 containing a strong shock from M=1.4 to M=.7. For this solution the workshop mesh was used with a far field boundary at 8 chordlengths from the airfoil. The convergence history is plotted in figure 6 showing a rate of .957 with 4 grids. The convergence rate with grid refining approaches 1 after 50 SLOR iterations and in fact no further improvement of the solution could be obtained although it was not converged as is seen in figure 7, dotted line. The non converged solution has a sharp shock ahead of the correct converged shock position and is very similar to some of the non conservative solutions presented in ref. 17. As can be seen in figure 8 the correct shock position with the multigrid scheme is already obtained after 50 work units (21 multigrid cycles) with a mean residual reduction of only $5\ 10^{-2}$, showing again the extremely fast elimination of low frequency errors.

The second set of results is calculated for a channel flow with a circular bump on the lower wall with the standard workshop mesh (ref. 17) which is a sheared Cartesian system. It was given as a testcase with an isentropic inlet Machnumber of .85. This Machnumber corresponds to a choked flow for the potential solution as was confirmed by the results of Veuillot and Viviand while Jameson did not succeed to obtain a solution for a Machnumber higher that .835. On the other hand, all other potential solutions, including our grid refinement solution were far from choked namely with a peak Machnumber of ±.92 on the upper wall. Again it is clear that this solution is not converged. Indeed the multigrid solution converges at M=.849 with a choked solution as is shown in figure 10 for the pressure distribution and figure 11 for the isomach lines. In this case 300 work units were performed with an average residual reduction of $4.2\ 10^{-2}$. The pressure distribution (figure 10) is compared with the solution obtained by Veuillot and Viviand at M=.8500 with their pseudo time dependent fully conservative potential method (ref. 17). Our solution at M=.85 diverges due to the fact that at this Machnumber the imposed massflow rate is higher than the choking massflow obtained at M=.849. In figure 9 our solution at M=.835 is compared with the solution of Jameson obtained with his multigrid ADI scheme (MAD) on a 65 x 17 meshpoints grid allowing 4 or 5 different grids. The residual evolution with our method (figure 12) shows a constant convergence rate of .963 after an oscillatory behaviour during 100 work units. The rate obtained by Jameson with MAD for this case was slightly slower namely .9742 (ref. 17).

The final result is a subcritical compressor cascade flow. The mesh is generated by solving a system of elliptic partial differential equations for the curvilinear ξ-η coordinates (ref. 14). The convergence history is shown in figure 13 and compared with the grid refining. The rate obtained with four grids is .874 and illustrates that the periodic and Neumann boundary conditions have no adverse effect on the convergence speed obtained with our multigrid scheme although no special treatment of the boundaries as suggested by Brandt has been introduced.

CONCLUSION

A conceptual simple multigrid scheme has been developed consistent with the finite element method and applicable to general arbitrarily generated body fitted grids. Therefore non uniform interpolation and residual weighting operators had to be introduced. A fast and reliable method is obtained with the simple straight forward line relaxation scheme as smoothing step allowing the calculation of realistic transonic flows with about 10 to 20 multigrid cycles (30 to 50 work units).

REFERENCES

(1) Fedorenko R.P., A relaxation method for solving elliptic difference equations, Zh. Vycisl. Mat. Fiz. 1 (1961) pp. 922-927

(2) Bakhvalov N.S., On the convergence of a relaxation method with natural constraints on the elliptic operator, Zh. Vycisl. Mat. Fiz 6 (1966) pp. 861-885

(3) Nicolaides R.A., On the l^2 convergence of an Algorithm for solving Finite Element Equations, Math. of Comp., Vol. 31, 1977, pp. 892-906

(4) Nicolaides R.A., On Multi-grid Convergence in the indefinite Case, Math. of Comp., Vol. 32, 1978, pp. 1082-1087

(5) Hackbusch W., On the Convergence of a Multi-grid Iteration applied to Finite Element Equations, Report 77-8, Math. Inst. Univ. Koln, 1977

(6) Brandt A. Multi-level adaptive Finite Element Methods : I Variational problems, ICASE Res. Report 79-8, 1979

(7) Nicolaides R.A., On some Theoretical and Practical Aspects of Multigrid Methods, Math. of Comp., Vol. 33, 1979, pp. 933-952

(8) South J.C., Brandt A., Application of a Multi-level grid Method to Transonic Flow Calculations,in : Transonic Flow Problems in Turbomachinery, Ed. by Adamson & Platzer, Hemisphere, Washington, 1977

(9) Jameson A., Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method, AIAA paper 79-1458, 1979

(10) Deconinck H., Hirsch Ch., Finite Element Methods for Transonic Flow Calculations, in : Proc. III GAMM Conf. on Num. Meth. in Fluid Mech., Köln 1979 (Notes on Numerical Fluid Mechanics, Vol. 2, Vieweg 1980)

(11) Lax P.D., Weak Solutions of nonlinear hyperbolic equations and their numerical Computation, Comm. Pure & Appl. Math VII, 1954

(12) Hafez M, Murman E.M., South J.C., Artificial Compressibility Methods for Numerical Solution of Transonic Full Potential Equation", AIAA paper 78-1148, 1978

(13) Deconinck H., Hirsch Ch., Subsonic and Transonic Computation of cascade flows, in : Proc. IV Int. Symp. on Computing Methods in Applied Sciences

and Engineering, Dec. 1979, IRIA Paris

(14) Deconinck H., Hirsch Ch., Finite Element Methods for Transonic Blade-to-Blade Calculation in Turbomachines, ASME paper 81-GT-5, 1981, presented at the Int. Gas Turbine Conf. and Product Show, Houston, Texas, March 9-12, 1981

(15) Deconinck H., Hirsch Ch., Transonic Flow Calculations with Higher Order Finite Elements, Proc. of 7th Int. Conf. Num. Meth. Fl. Dyn. June 23-27, 1980, Stanford, USA

(16) Brandt A., Multi-level adaptive solutions to Boundary-value Problems, Math. Comp., Vol. 31, pp. 333-390, 1977

(17) Numerical Methods for the Computation of Inviscid Transonic Flows with shock Waves : a GAMM workshop, A. Rizzi, H. Viviand (Eds.), Vieweg 1981

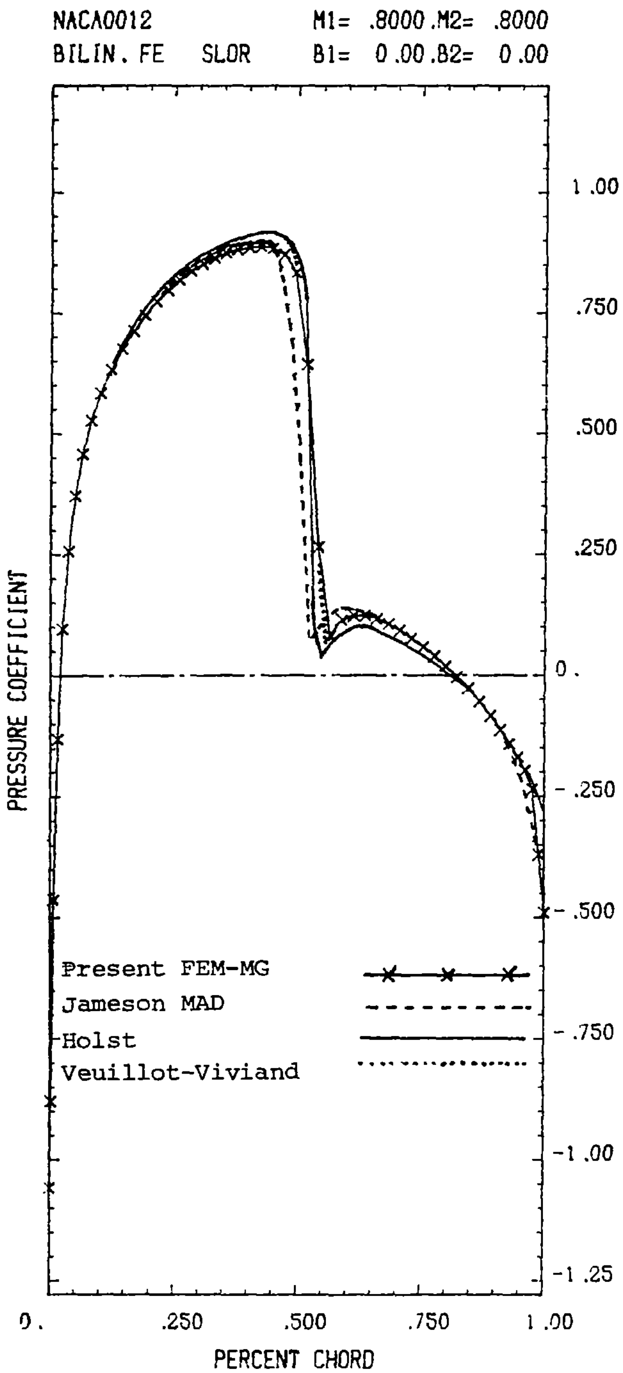


Fig. 3 : Comparison with other conservative methods

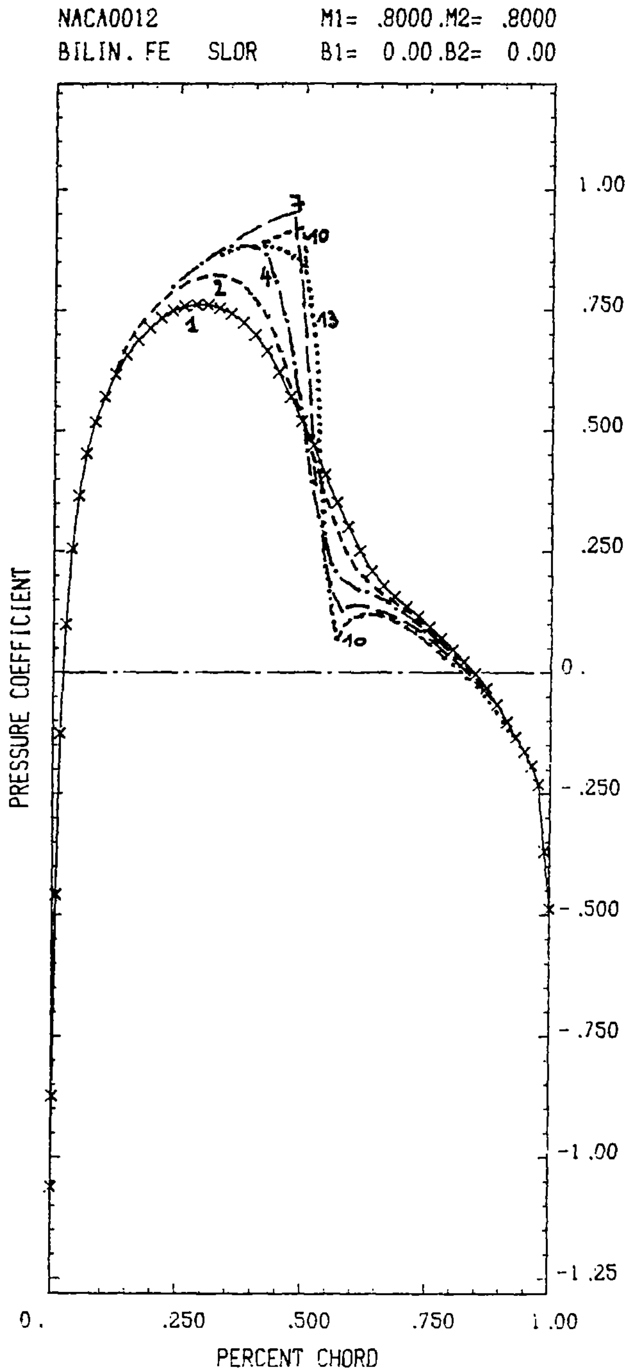


Fig. 4 : Evolution of the solution in pseudo-time; the number of multigrid cycles is indicated

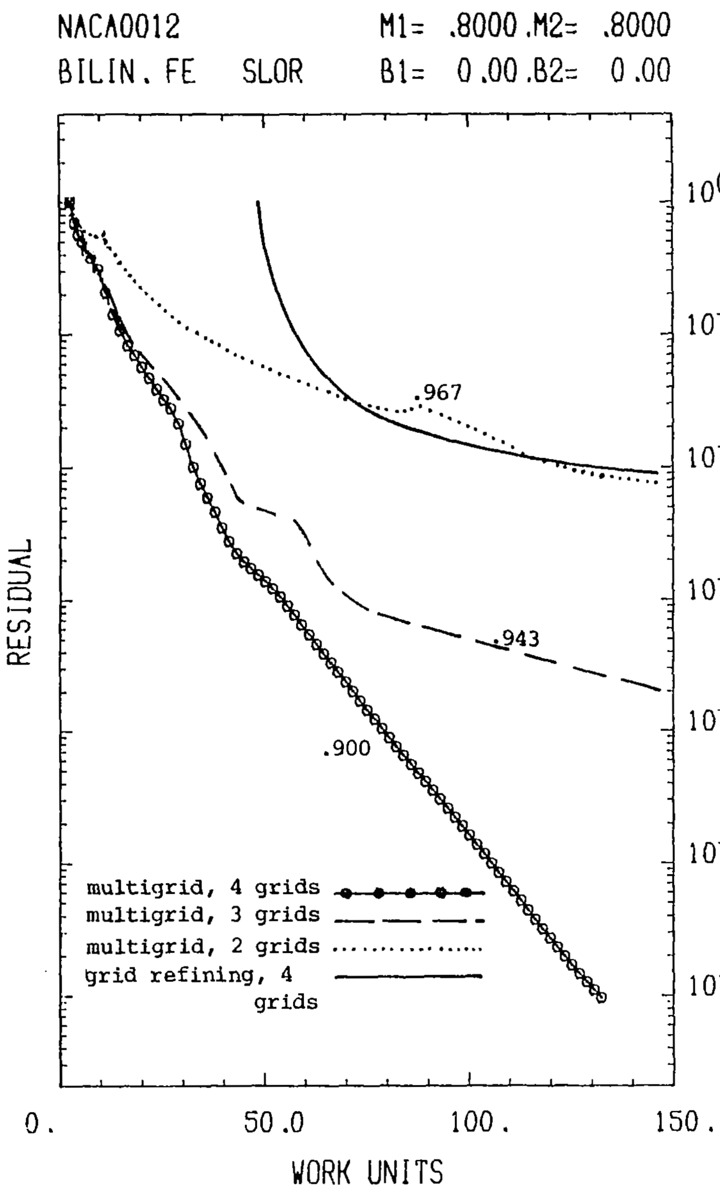


Fig. 5 : Convergence history (M=.80) convergence rate is indicated

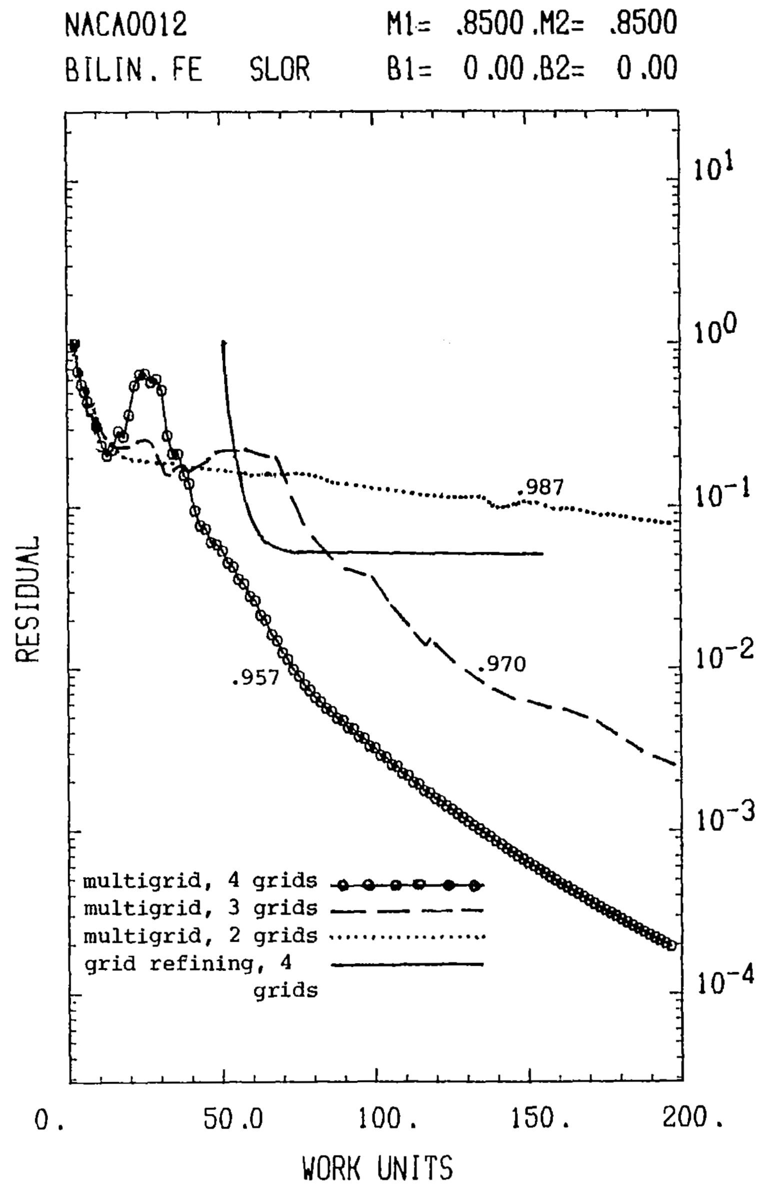


Fig. 6 : Convergence history (M=.85) convergence rate is indicated

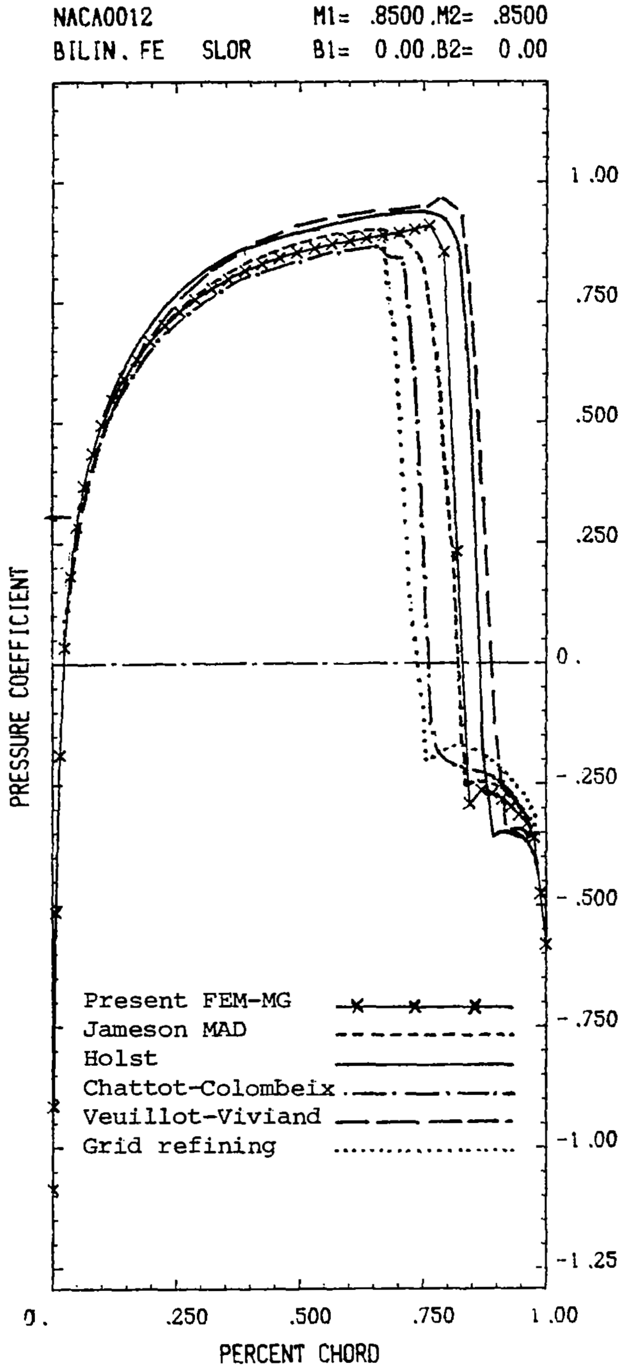


Fig. 7 : Comparison with other conservative methods

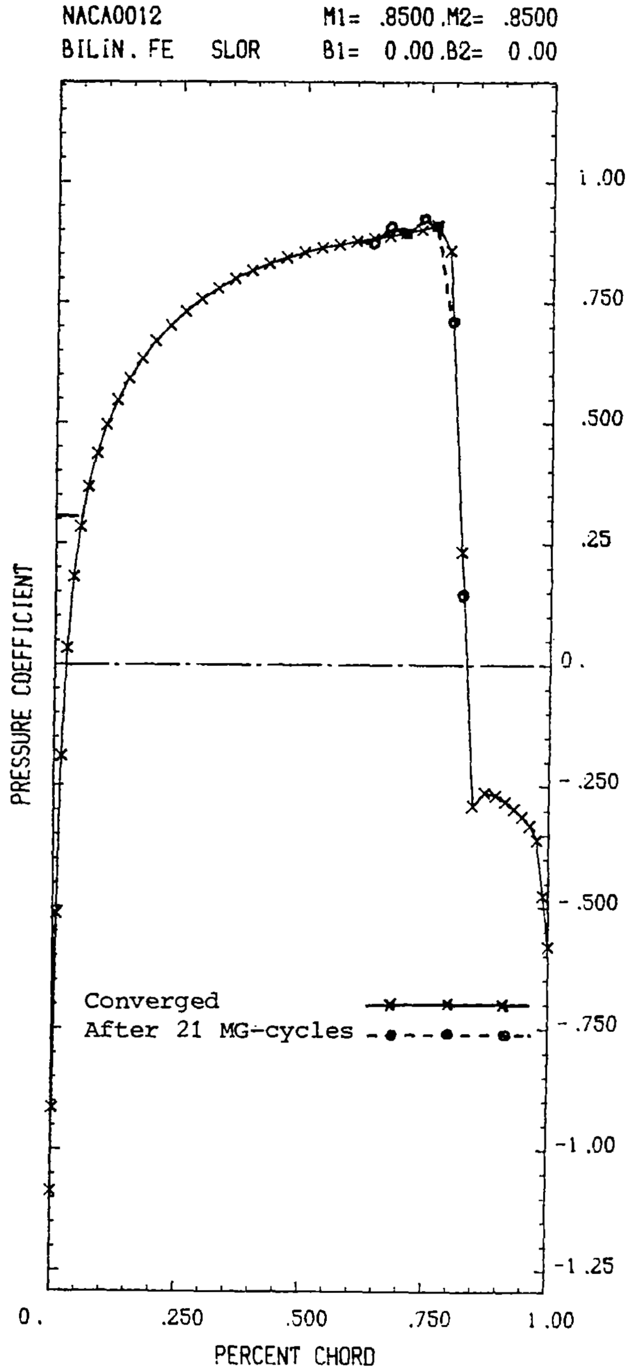


Fig. 8 : Comparison of solution after 21 Multigrid cycles (50 work units) with converged solution

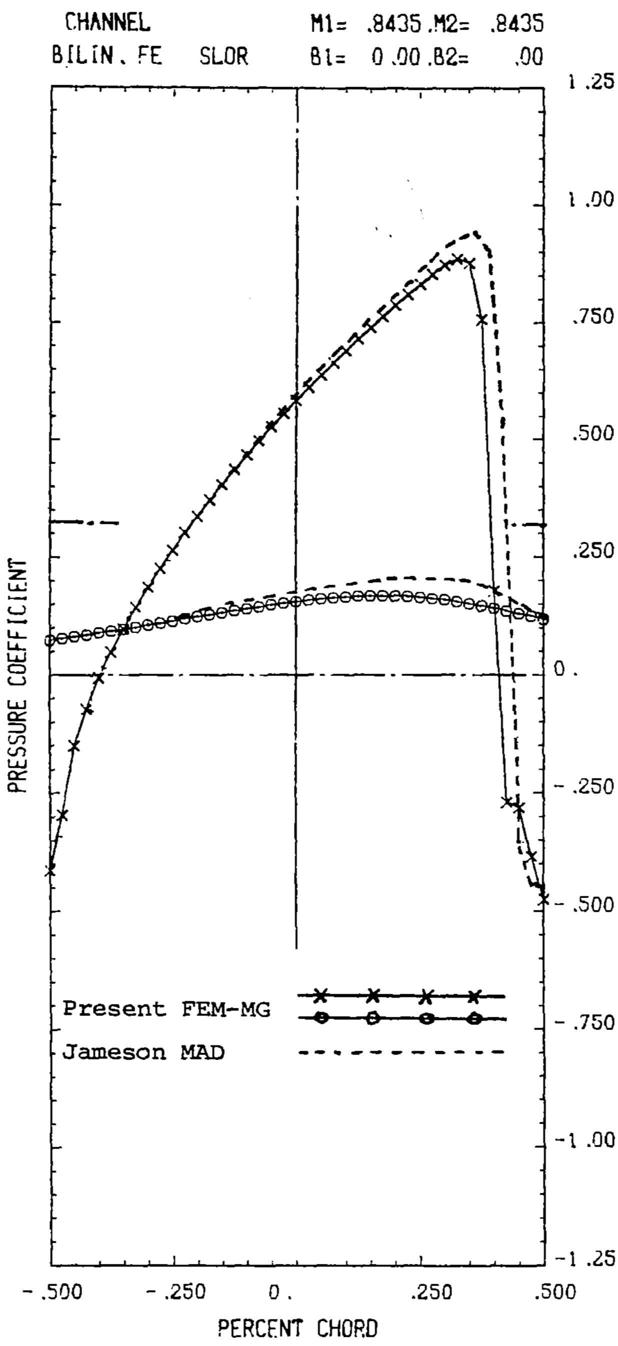


Fig. 9 : Channel flow at M=.8435

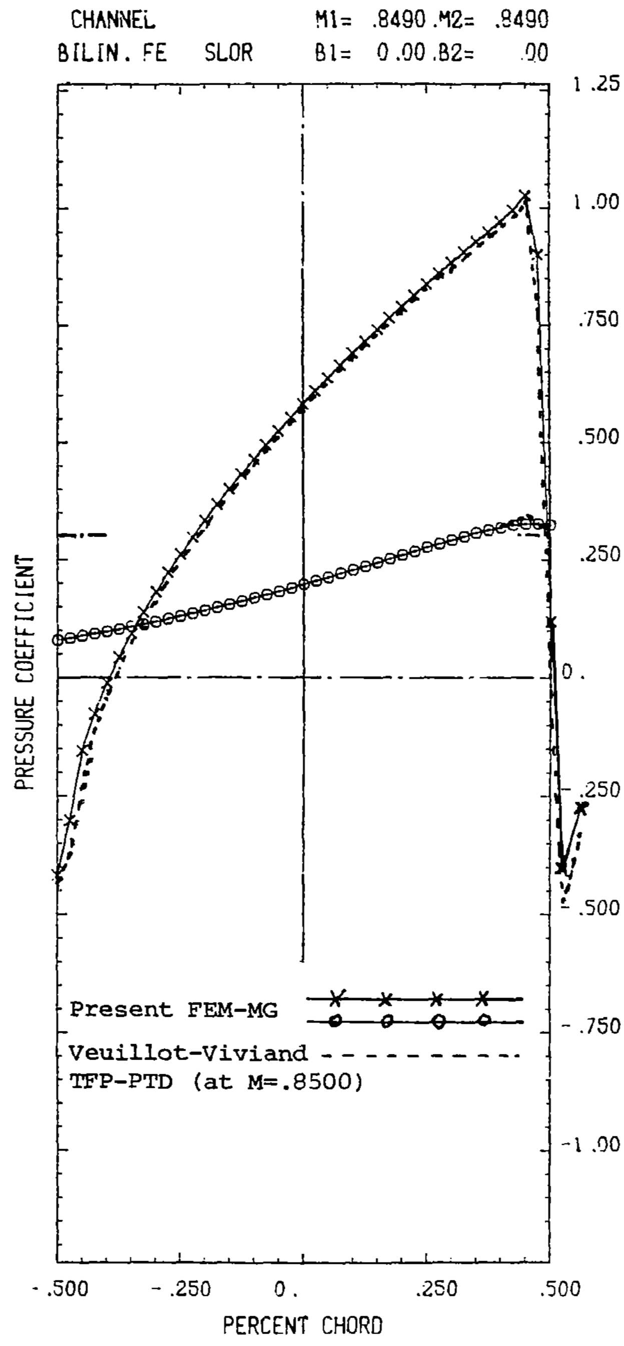


Fig. 10 : Channel flow at M=.8490 (choked)

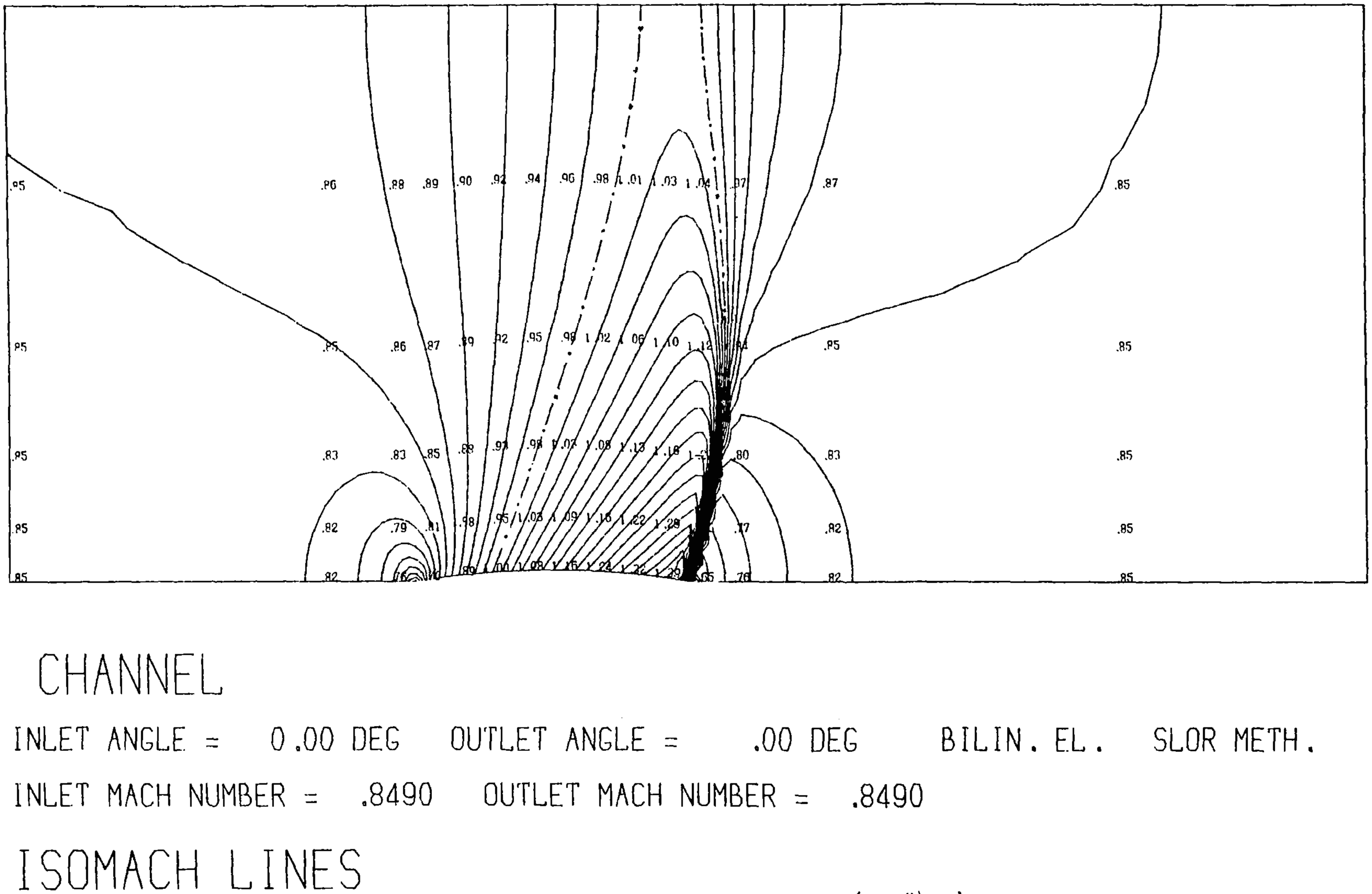
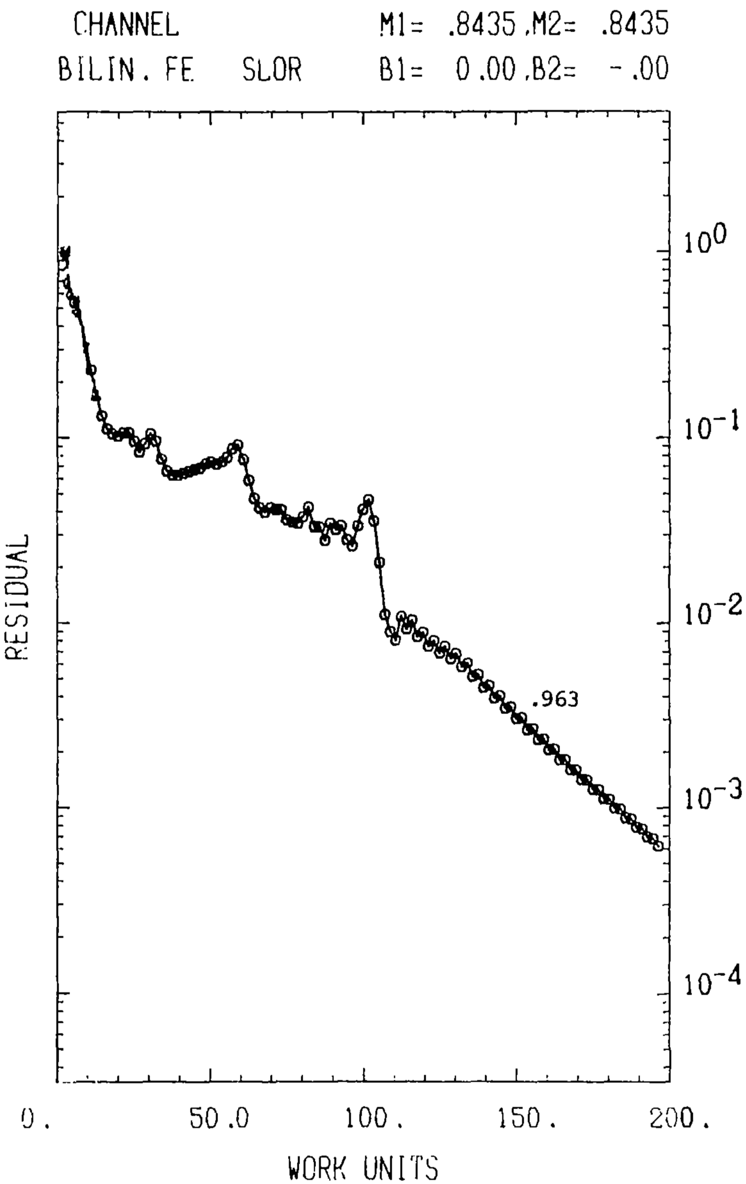
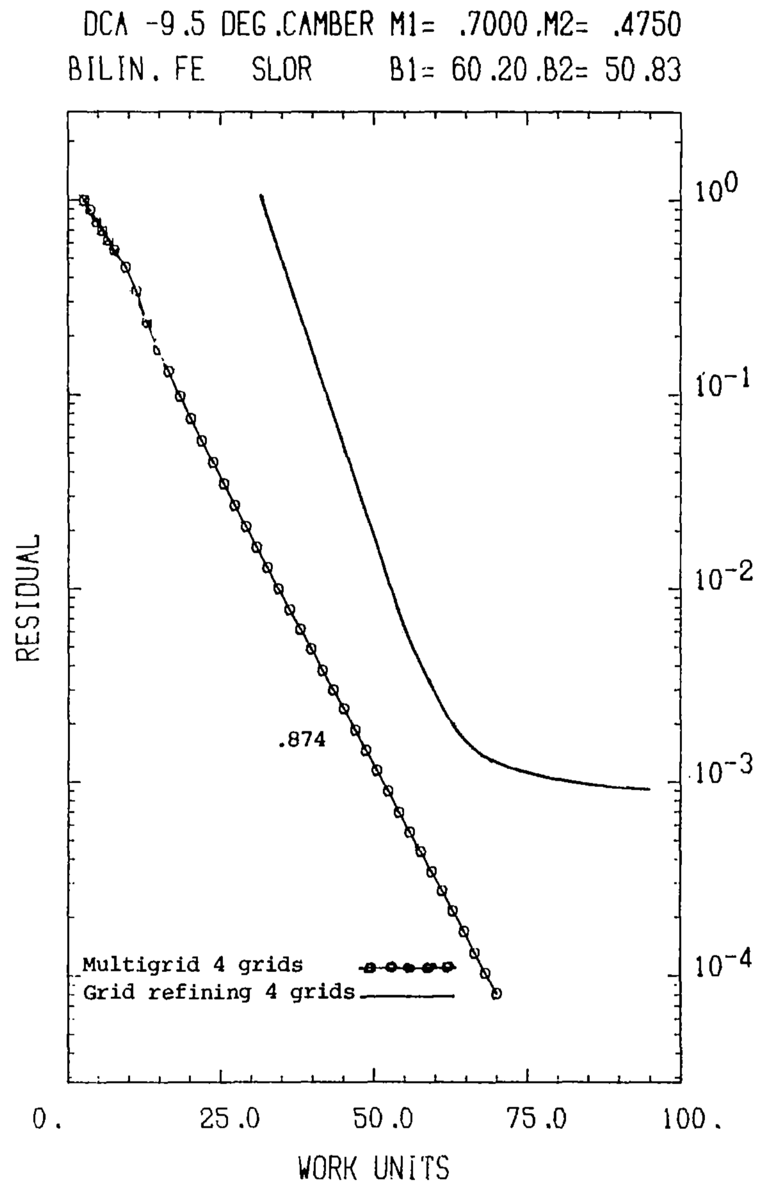


Fig. 11 : Choked channel flow (M=.8490)



<u>Fig. 12</u> : Channel flow at M=.835 Convergence history

(rate = .963)



<u>Fig. 13</u> : Subcritical cascade flow. Convergence history

(Convergence rate is indicated)

MULTI-GRID SOLUTION OF THE NAVIER-STOKES EQUATIONS ON NON-UNIFORM GRIDS

Laszlo Fuchs
The Royal Institute of Technology, Stockholm

SUMMARY

The numerical solution of the Navier-Stokes equations in general two dimensional domains is considered. A proper finite-difference approximation to the governing equations, on non-staggered grids, in a transformed plane, is formulated. Several aspects of a Multi-Grid method for the solution of the finite-difference equations are described. The emphasis in this paper is on the efficiency of some relaxation schemes and the transfer among the grids which are non-uniform in the physical plane.

INTRODUCTION

Multi-Grid (MG) methods have been applied with success to many boundary value problems. Both elliptic equations [1-4] and mixed elliptic-hyperbolic equations [5-7] have been solved with high efficiency. However, the MG method is not, yet, considered to be a general purpose method because of its relative complexity. The method is usually sensitive to basic errors which can be made even by a MG-minded user. For this and other reasons it is important to understand not only the basic principles of the MG solution procedures but also to develop, as general as possible, user oriented codes for the solution of specific equations. In this work we discuss some aspects which are of importance for the development of Navier-Stokes solvers in general two dimensional domains.

In most applications so far, the MG codes have been written for problems in cartesian coordinates. This is natural for testing the basic principles of the method. The idea of using uniform cartesian grids has been extended to include globally non-uniform grids, by using a sequence of uniform grids such that some of these are applied locally [1]. This approach can be used without too much difficulties if the boundaries of the computational domain can be approximated easily by rectangular meshes. For general geometries such an approach might be too complex and less accurate than the method of the transformation of the coordinates.

We distinguish between two cases where mesh refinements are needed in the physical plane. That is, when the mesh should be refined in order to resolve geometrical details, or when physical details of the solution are to be resolved. The purpose of the transformation of the coordinates is to make the treatment of general geometries easier, while the mesh refinements needed to resolve the solution are to be treated adaptively as part of the solution procedure. Mesh refinements can be done if and when such are needed by introducing fine uniform meshes locally, in the rectangular computational domain (e.g. in the way that is described in reference 1).

By the transformation of the coordinates, the boundaries become coordinate lines and this simplifies the application of the boundary conditions. Tha main disadvantage of such transformations is that the equations may become more complex. "New" terms turn up and the coefficients multiplying the derivatives may vary consideribly throughout the computational domain. Beside the complexity in programing, the transformation of the coordinates may reduce the computational efficiency of methods which work well on uniform cartesian grids. Thus, working with transformation of the coordinates means that additional attention must be paid to the formulation of the governing equations, the discretization of these equations, the choice of the relaxation scheme and the transfer among the grids. In this work we discuss some of these aspects with regard to the MG method for the solution of the Navier-Stokes equations in a plane. The governing equations are stated and discretized "elliptically". A proper choice for transfer among the grids is related to the governing equations. By such a transfer the continuity equation and the compatibility condition can be satisfied to the same accuracy on all grids. Some relaxation schemes, including the so called "Convective" Successive Line Relaxation (C-SLR) scheme for the momentum equations, are described. These are analysed by local mode analysis and are tested for some coordinate transformations and different values of Reynolds numbers. The C-SLR scheme is superior to standard schemes, and it may be used as a general purpose relaxation scheme in many cases. Preliminary results for the solution of the Navier-Stokes equations for the flow in rectangular and polar cavities are also given.

THE GOVERNING EQUATIONS

The steady state Navier-Stokes equations for incompressible viscous Newtonian fluids, in two dimensional cartesian coordinates, are given by

$$u_{xx} + u_{yy} - p_x - Re\ (uu_x + vu_y) = 0 \qquad (1.a)$$

$$v_{xx} + v_{yy} - p_y - Re\ (uv_x + vv_y) = 0 \qquad (1.b)$$

$$u_x + v_y = 0 \qquad (1.c)$$

where Re is the Reynolds number, u and v are the dimensionless velocity components in the x and y directions, respectively. p is the dimensionless pressure scaled by the Reynolds number.

Proper boundary conditions are defined if the velocity vector $\bar{q} = (u,v)$ is given on the boundaries, provided that the compatibility condition

$$\iint_{\Omega} (u_x + v_y)\ dx\ dy = \int_{\partial\Omega} \bar{q}\cdot\bar{n}\ dl = 0 \qquad (1.d)$$

is satisfied. ($\partial\Omega$ is the boundary of the domain Ω and $\bar{n}$ is the outward pointing unit vector normal to $\partial\Omega$).

Usually no slip boundary conditions are taken on the surface of physical bodies. Mixed boundary conditions may be used in cases of symmetry.

When the physical plane is transformed into the computational plane (ξ,η) the governing equations can be written in the form:

$$J[\,(C_1u_\xi + C_3u_\eta\,)_\xi + (C_3u_\xi + C_2u_\eta\,)_\eta] - (\xi_xp_\xi + \eta_xp_\eta) - Re[u(\xi_xu_\xi + \eta_xu_\eta) + v(\xi_yu_\xi + \eta_yu_\eta)] = 0 \qquad (2.a)$$

$$J[\,(C_1v_\xi + C_3v_\eta\,)_\xi + (C_3v_\xi + C_2v_\eta\,)_\eta] - (\xi_yp_\xi + \eta_yp_\eta) - Re[u(\xi_xv_\xi + \eta_xv_\eta) + v(\xi_yv_\xi + \eta_yv_\eta)] = 0 \qquad (2.b)$$

$$(\,\xi_xu_\xi + \eta_xu_\eta + \xi_yv_\xi + \eta_yv_\eta\,) = 0 \qquad (2.c)$$

where $C_1 = (\xi_x^2 + \xi_y^2\,)/J$

$C_2 = (\eta_x^2 + \eta_y^2\,)/J$

$C_3 = (\xi_x\eta_x + \xi_y\eta_y)/J$

and $J = \xi_x\eta_y - \xi_y\,\eta_x$

u and v are the components of the velocity vector $\bar{q}$ in the x and y directions, respectively.

Due to the transformation of the coordinates the boundary conditions are simplified in the sense that u and v are given on the boundaries of a unit rectangle in the ξ,η plane.

Equations (2) are written in a non-conservative form. Conservative forms are preferable in many cases. Such a form is essential for the continuity equation (2.c). It can be written in conservative form as:

$$\hat{u}_\xi + \hat{v}_\eta = 0 \qquad (3)$$

where $\hat{u}$ and $\hat{v}$ are the velocity components of $\bar{q}$ in the ξ and η directions, respectively.

$$\hat{u} = (\xi_xu + \xi_yv)/J$$

and

$$\hat{v} = (\eta_xu + \eta_yv)/J$$

The integral of equation (3) on any closed (simply connected) domain Ω gives:

$$\iint_\Omega(\hat{u}_\xi + \hat{v}_\eta)\; d\xi\; d\eta = \int_{\partial\Omega}(u,v)\cdot\bar{n}\; dl \qquad (4)$$

The compatibility condition is satisfied if the flux through the boundaries vanishes. The numerical integral analog to equation (4) is important for the correct transfer of residuals from fine to coarse grids. The momentum equations are left in their non-conservative form (eqs. (2.a) and(2.b)).

FINITE-DIFFERENCE APPROXIMATIONS

The system of the governing equations (1) is elliptic. It is, therefore, natural that this property shall be transfered to the finite difference approximations as well. Some concepts of ellipticity of finite difference equations (as the mesh size goes to zero) are described in reference 2 and the references in that paper. For finite mesh size, Brandt and Dinar [2] use the concept of hT-ellipticity measure. This concept is directly related to the possibility of devising a proper relaxation scheme in the MG sense.

For simplicity we consider the discretization of the problem in cartesian coordinates. The following notations for the finite difference approximations are used:

$$\partial_x^F = [(\ )_{i+1,j} - (\ )_{i,j}]/h \qquad \partial_y^F = [(\ )_{i,j+1} - (\ )_{i,j}]/h$$

$$\partial_x^B = [(\ )_{i,j} - (\ )_{i-1,j}]/h \qquad \partial_y^B = [(\ )_{i,j} - (\ )_{i,j-1}]/h$$

$$\partial_x^C = (\partial_x^F + \partial_x^B)/2 \qquad \partial_y^C = (\partial_y^F + \partial_y^B)/2$$

$$\partial_x^2 = \partial_x^F\,\partial_x^B \qquad \partial_y^2 = \partial_y^F\,\partial_y^B$$

$$\nabla_h^2 = \partial_x^2 + \partial_y^2$$

where h is the mesh size in both x and y directions.

First, the Stokes problem (Re = 0) is considered. If second order accurate central differences are used (on non-staggered grids) to approximate all the terms in equations (1), then one may write the finite difference equations as:

$$L_h\ \Phi = R \tag{5.a}$$

with

$$L_h = \begin{bmatrix} \nabla_h^2 & 0 & \partial_x^C \\ 0 & \nabla_h^2 & \partial_y^C \\ \partial_x^C & \partial_y^C & 0 \end{bmatrix}$$

$$\Phi = (u,v,p)^T$$

$$R = (0,0,0)^T$$

The symbol of L_h (see Section 3.2 in reference 2), $\tilde{L}_h(\vartheta_1,\vartheta_2)$, is given by

$$\tilde{L}_h(\vartheta_1,\vartheta_2) = (\cos\vartheta_1 + \cos\vartheta_2 - 2)\ (\sin^2\vartheta_1 + \sin^2\vartheta_2)$$

This operator is not elliptic (it is quasi elliptic) and a solution of equation (5.a) approximates the solution of the differential equations only in average.

Elliptic operators may be obtained if staggered grids are used together with central differences. Such grids has also, been used for the time dependent case [8]. Brandt and Dinar [2] have used staggered grids for the solution of the two dimensional Navier-Stokes equations in cartesian coordinates. In reference 9 a version of this method has been compared with other MG solvers of the Navier-Stokes equations. An application of the three dimensional staggered grid solver is described in reference 10.

Staggered grid formulation looses its attractiveness when cross terms are to be discretized. Such cross terms may appear after the transformation of the coordinates. For this reason we use, here, non-staggered grids. An elliptic approximation to equations (1) is obtained by the following operator L_h in equation (5.a):

$$L_h = \begin{bmatrix} \nabla_h^2 & 0 & \partial_x^F \\ 0 & \nabla_h^2 & \partial_y^F \\ \partial_x^B & \partial_y^B & 0 \end{bmatrix} \qquad (6)$$

Thus $\det L_h = \nabla_h^2 (\partial_x^2 + \partial_y^2)$, with the symbol proportional to $(\cos\vartheta_1 + \cos\vartheta_2 - 2)^2$. This symbol vanishes only for $\vartheta_1 = \vartheta_2 = 0 \pmod{2\pi}$. This means that the approximation to the differential equation is elliptic. It may be noted that ∂^F and ∂^B may be interchanged without having an effect on the ellipticity. It is also clear that such an approximation is of first order accuracy. However, even if the staggered grid approximation is of second order accuracy inside the computational domain, the boundary conditions are applied with first order accuracy. The total accuracy of both approaches (staggered and non-staggered) is of first order for the velocity components [8]. Moreover, when the non-linear (convective) terms are included (Re> 0) central differences (for these terms) may be used only if the cell Reynolds number, $Re_h = \max(|Rehu|,|Rehv|)$ is less than unity. For higher Reynolds numbers the symbols of the approximations both on staggered and non-staggered grids is not elliptic. To preserve the ellipticity for all Reynolds numbers one has to use upstream approximations (usually of first order accuracy) to the convective terms (see e.g. reference 11).

A result of approximation (6) is that the standard five point approximation to the Poisson operator is satisfied if it is applied on the pressure found by solving system (5.a) with (6). That is, the pressure, in contrast to the velocity components, is computed with second order accuracy.

The generalization of approximation (6) on non-staggered grids to the transformed equations (2) is done in a straight-forward manner.

THE MULTI-GRID CYCLING PROCEDURE

For MG solution procedures one defines a sequence of M grids with mesh spacings $h_1 \dots h_M$ such that the finest grid has the spacing h_M. Usually $h_{k+1}/h_k = 2$ $(1 < k < M)$. The MG cycling procedure for the solution of the system of the algebraic equations

$$L_M \varphi = f_M$$

on a grid with spacing h_M, is as follows:

i. Relaxation sweeps are carried out (on the problem $L_k \varphi_k = f_k$), on the grid k until the convergence is too slow,

then either

ii. The problem is transfered to a coarser grid, where new relaxation sweeps are done. This transfer (for the FAS-mode) is given by:

$$f_{k-1} = L_{k-1} I_{k-1}^{k} \varphi_k + I_{k-1}^{k} (f_k - L_k \varphi_k)$$

where I_{k-1}^{k} is the transfer operator from the fine grid (k) to the coarse grid (k-1).

Or if convergence to some accuracy has been obtained, then:

iii. The correction is transfered to a finer grid. These corrections are smoothed out by relaxation sweeps (step i.).

The procedure ends when the prescribed accuracy is attained on the finest grid. In the following sections we discuss some relaxation schemes (step i.) and proper transfer operators I_{k-1}^{k} (step ii.).

THE RELAXATION PROCEDURE

The purpose of the relaxation steps in any iterative solution process is to smooth out the errors. In a MG-procedure this smoothing process may be restricted to those Fourier components of the error which can be described on a given grid but not on coarser grids (high frequency components). The efficiency of the relaxation procedure(provided that no large-amplitude low-frequency errors are generated during the transfer among the grids) determines the overall efficiency of the MG solution procedure.

When a system of difference equations approximating a system of differential equations is to be solved, the efficient relaxation of each equation (variable) does not neccessarily result in an efficient scheme for the system. This happens if by relaxaing one equation, new high frequency error components are introduced in the residuals of the other equations. A way of (almost) decoupling the relxation of the finite difference approximations to the continuity (1.c) and the momentum equations (1.a) and(1.b) has been suggested by Brandt and Dinar [2] for the staggered grid approximation. Here, a similar disributive Gauss-Seidel (DGS) relaxation scheme for the non-staggered grid approximation is described.

We consider, first, the linearized Navier-Stokes equations (with frozen coefficients) in cartesian coordinates, discretized on a uniform grid with a mesh spacing h. The finite difference approximation can be written as:

$$Qu - \partial_x^F p = 0 \tag{7.a}$$

$$Qv - \partial_y^F p = 0 \tag{7.b}$$

$$\partial_x^B u + \partial_y^B v = 0 \tag{7.c}$$

where $Q = \nabla_h^2 - Re\,(u\vec{\partial}_x + v\vec{\partial}_y)$, and $\vec{\partial}_x$ and $\vec{\partial}_y$ are the upstream approximations to the first derivatives of the convective terms.

Equations (7.a) and (7.b) can be relaxed by solving (pointwise or linewise) for u and v, respectively. Efficient relaxation procedures have been developed for these equations both in cartesian and in general curvlinear-coordinates.

The relaxation of equation (7.c) is more complex since if only u and v are changed, new (high frequency) errors are introduced in equations (7.a) and (7.b). This in turn means that the relaxation efficiency (of the high frequency error components) of the system may be very poor. If, on the other hand, for any function χ, the dependent variables are changed (Δu, Δv and Δp) according to equations (8) then the residuals of the momentum equations shall not be altered (provided that $Q\partial^F = \partial^F Q$).

$$\Delta u = \partial_x^F \chi \tag{8.a}$$

$$\Delta v = \partial_y^F \chi \tag{8.b}$$

and

$$\Delta p = Q\chi \tag{8.c}$$

A particular choice of χ, which is convenient, is to take it to be equal δ at the node point at which equation (7.c) is to satisfied, and zero elsewhere. Such a choice means that the velocity vector is changed at three node points while the pressure is changed at five node points simultaneously.

Inserting equations (8) into (7.c) gives

$$\nabla_h^2 \chi = -(\partial_x^B u + \partial_y^B v) \tag{9}$$

For our particular choice of χ,

$$\delta = (\partial_x^B u + \partial_y^B v)/(4/h^2) \tag{10}$$

If equations (8) are used then the residuals of the momentum equations are unchanged at all node points except at those adjacent to the boundaries. The reason for this is that at these points equations (8.a) and (8.b) are not valid (since u and v are specified on the boundaries, and they cannot be changed). The residual near the boundaries is changed by a factor δ/h^3 which is of the same order as the original residual in the momentum equations.

If the momentum equations are written in terms of the velocity components which are parallel to the transformed coordinates, one gets equations of the form

$$Q_1\hat{u} - \partial_\xi p = 0$$

$$Q_2\hat{v} - \partial_\eta p = 0$$

and $Q_1 \neq Q_2$ in general (e.g. cylindrical coordinates). Under such circumstances one cannot design a distributive relaxation scheme which will preserve the residuals of both momentum equations. In all such cases one has to use a common part of Q_1 and Q_2 such that the change in the residuals in both equations will be smooth and of at most the order of the errors which are generated near the boundaries. Such DGS relaxation schemes can be evaluated by local mode analysis for the system.

An important aspect which must be considered together with the distributive relaxations, is the accuracy in satisfying the compatibility condition (1.d). Again, we consider first the case of cartesian coordinates. Using the sided differences, as in (6), the compatibility condition can be written as

$$\sum_{i,j} (\partial_x^B u + \partial_y^B v)\, h^2 = \sum_j (u_1 - u_0)h + \sum_i (v_3 - v_2)h \quad (11)$$

where the subscripts 0,1,2 and 3 denote the values of the dependent variables on the sides of the computational rectangle. It is clear that the right hand side of equation (11) must vanish if the compatibility condition is to be satisfied. However, the question is what accuracy is tolerated in the numerical integration (11). Is it enough if the compatibility condition is satisfied to the truncation errors, or the accuracy must be that of the round-off errors?

To answer this question we consider the system of the algebraic equations (for δ) which is obtained from equation (9). The sum of the terms on the left had side of these equations vanishes (to round-off). This means that the equations of the system are linearly dependent. On the other hand, the right hand side equals to the left hand side of (11). This implies that in order to have a solution to the system of difference equations, the compatibility condition (11) must vanis (to round-off). A unique solution (for δ) can be obtained if the value of δ is specified at some point. The corrections in the velocity components (8) which are equal to the derivatives of the δ field, are not dependent on this prescribed value.

If non-conservative form of the continuity equation, is used then the compatibility condition can be satisfied only to a certain accuracy (the truncation error). This in turn means that the DGS relaxations can converge only to a certain level.

The numerical compatibility condition must be satisfied on all grids,and this must be taken into account when the velocity components are transfered to coarse grids.

FINE TO COARSE GRID TRANFERS

As discussed above the validity of the compatibility condition (to round-off error) is a condition for the existance of a solution. To satisfy the compatibility condition on the coarse grids to the same accuracy as on the fine one, the transfer from the fine to the coarse grids should be chosen carefully.

The discretized counter part of equation (4) is

$$S_k = \sum_{i,j} \{[\frac{1}{J}(\xi_x u + \xi_y v)]_\xi + [\frac{1}{J}(\eta_x u + \eta_y v)]_\eta\} h_k^2 \qquad (12)$$

where i,j denotes the node point (ξ_i, η_j) and h_k is the mesh size of a grid with the label k. Since equation (12) is in conservative form, one gets that S_k equals to the flux through the boundaries of the domain. That is,

$$S_k/h_k = \sum_j [\frac{1}{J}(\xi_x u + \xi_y v)]_0^1 + \sum_i [\frac{1}{J}(\eta_x u + \eta_y v)]_2^3 \quad (13)$$

To satisfy the compatibility condition, the right hand side of equation (13) must vanish on the finest grid (k □ M).

In the FAS-mode the residuals and the dependent variables are transfered to the coarse grids. The right hand side of the continuity equation on coarse grids should be compatible with the flux integral (the sum in equation (13)). If the sum in equation (12) vanishes on the grid k, then it will vanish also on the grid k-1 if

$$S_{k-1} = \Sigma\, m_{k-1}\, h_{k-1}^2 \qquad (14.a)$$

and

$$m_{k-1} = [\ \Sigma\{[\frac{1}{J}(\xi_x u + \xi_y v)]_\xi + [\frac{1}{J}(\eta_x u + \eta_y v)]_\eta\}]\frac{h_k^2}{h_{k-1}^2} \qquad (14.b)$$

where the sum in equation (14.a) is over the indices of the coarse grid (k-1) and the sum in equation (14.b) completes the first sum so that all the node points of the fine grid (k) are covered.

Equation (13) can be written as

$$S_k/h_k = \sum_j [\frac{\tilde{u}}{J}]_0^1 + \sum_i [\frac{\tilde{v}}{J}]_2^3 \qquad (15)$$

where $\tilde{u}$ and $\tilde{v}$ are the components of the velocity vector in the ξ and η directions. A natural transfer of $\tilde{u}$ and $\tilde{v}$ from the grid k to the grid k-1, is by weighted averages ,with the Jacobian as weighting function:

$$\tilde{u}_{k-1} = (\Sigma\ \tilde{u}_k/J_k)J_{k-1} \quad \text{and} \quad \tilde{v}_{k-1} = (\ \Sigma\ \tilde{v}_k/J_k)J_{k-1} \qquad (16)$$

with the sums as in equation (14.b). If relations (16) are used then the fluxes through any closed region of the domain, are the same and are independent on the grid which is used for the computation of the fluxes.

SMOOTHING FACTORS OF RELAXATION SCHEMES

The efficiency of relaxation schemes can be estimated with good accuracy by using the method of local mode analysis [1-3,5]. Fourier components which are of interest from MG point of view have wave lengths which are of the same order as the mesh size. Other error components may be considered as slowly varying and thus they do not contribute much to the variation in the residual in neighbouring points. The interesting Fourier components (of high frequencies) have short range of effect and therefore boundary effects may be neglected. By local mode analysis it is ment that one considers how the amplitude, A, of a Fourier component $\exp(i\,\vartheta_1/h + i\,\vartheta_2/h)$ of the error is reduced by one relaxation sweep (carried out on the equation with frozen coefficients). The amplification factor of the amplitude is denoted by $\mu = \mu(\vartheta_1,\vartheta_2)$ and the smoothing factor $\bar{\mu}$ is defined as

$$\bar{\mu} = \max\ |\mu(\vartheta_1,\vartheta_2)| \tag{17}$$

where the maximum is taken over the high frequency components (i.e. those components which can be described on a given grid but not on coarser ones).

In the following we derive the amplification factor for some relaxation schemes for the momentum and the continuity equations.

For the relaxation of the momentum equations we consider the following linear operator:

$$L_m = C_1\partial_\xi^2 + C_2\partial_\eta^2 - \mathrm{Re}(C_3\vec{\partial}_\xi + C_4\vec{\partial}_\eta) \tag{18}$$

where C_1, C_2, C_3 and C_4 are taken to be constants. For the Fourier component $\exp(i\vartheta_1\xi/h + i\vartheta_2\eta/h)$, let r be the amplitude of the residual before a relaxation pass, $\bar{r}$ its amplitude after the pass, $\tilde{r}$ the amplitude of the dynamic residual and A be the amplitude of the correction.

For a pointwise Gauss-Seidel (Successive Point Relaxation, SPR) relaxation scheme, the dynamical residual is given by

$$\tilde{r} = r + A[(C_1 - \mathrm{Reh}H(C_3))\,e^{-i\vartheta_1} + (C_2 - \mathrm{Reh}H(C_4))\,e^{-i\vartheta_2}]/h^2$$

where

$$H(z) = \begin{matrix} z, & z>0 \\ 0, & z<0 \end{matrix}$$

The residual after the pass, $\bar{r}$, is

$$\bar{r} = A[(C_1 - \mathrm{Reh}H(-C_3))\,e^{i\vartheta_1} + (C_2 - \mathrm{Reh}H(-C_4))\,e^{i\vartheta_2}]/h^2$$

For SPR, A is solved from:

$$\tilde{r} + A[-2(C_1 + C_2) - \mathrm{Reh}(|C_3| + |C_4|)] = 0 \tag{19}$$

which gives that

$$\mu(\vartheta_1,\vartheta_2) = \left|\frac{[C_1 + \mathrm{Reh}H(-C_3)]\,e^{i\vartheta_1} + [C_2 + \mathrm{Reh}H(-C_4)]\,e^{i\vartheta_2}}{2(C_1 + C_2) - C_1\,e^{-i\vartheta_1} - C_2\,e^{-i\vartheta_2} + \mathrm{Reh}\ NT}\right| \tag{20}$$

where

$$NT = |C_3| + |C_4| + H(C_3)\,e^{-i\vartheta_1} + H(C_4)\,e^{-i\vartheta_2}$$

For cartesian coordinates and small cell Reynold numbers, Re_h, the smoothing factor equals that of the Poisson equation, i.e. $\bar{\mu} \approx 0.5$. For large cell Reynolds numbers the smoothing factor depends strongly on the sign of C_3 and C_4 (the relative direction of the relaxation sweep and the flow direction).

If the relaxation direction and the flow direction are alligned then the smoothing factor equals 1/3 for a wide range of values of C_1 and C_2 (see Table I). If the relaxation direction is against the flow direction $\bar{\mu}$ is very close to unity. In general cases when the flow direction is not known in advance, one can make two relaxation sweeps in opposite directions. In such a way the average smoothing factor,per sweep, would be less then $(1/3)^{\frac{1}{2}}$. This type of double sweeps is called symmetrical relaxations. Difficulty arises also for small Re_h when $C_1 >> C_2$ or $C_2 >> C_1$. In these cases the relaxation efficiency is poor, but it can be improved if Successive Line Relaxations (SLR) are used. The amplification factor $\mu(\vartheta_1,\vartheta_2)$ for (ξ-) SLR applied on (18) (solving for each line ξ = constant), is:

$$\mu(\vartheta_1,\vartheta_2) = \left| \frac{C_1 - Reh\ H(-C_3)}{2(C_1 + C_2 - \cos\vartheta_2) - C_1\ e^{-i\vartheta_1} + Reh\ NT} \right| \qquad (21)$$

where $\quad NT = H(C_3)e^{-i\vartheta_1} + |C_3| + |C_4| - H(-C_4)e^{i\vartheta_2} + H(C_4)e^{-i\vartheta_2}$

The ξ-SLR scheme is efficient if $C_3>0$ (independent of C_4) or for small values of Re_h if $C_2>>C_1$ (e.g. when the mesh size in the η direction is larger than that in the ξ direction). If on the other hand, $C_1>>C_2$, one can use η-SLR instead of ξ-SLR. In general cases, when the computational field contains large variations in C_1/C_2, good smoothing factor is obtained if ξ-SLR is followed by η-SLR. This type of relaxation is called Alternating Direction SLR (AD-SLR). ξ-SLR is efficient if $C_3>0$ (independent of C_4) but as the cell Reynolds number increases and if the flow and the relaxation directions are against each other ($C_3<0$), then $\bar{\mu}$ is close to unity (see Table I). General flow fields can be relaxaed by symmetrical SLR sweeps.

To improve the simple SLR for cases of general geometry, a modified SLR has been tested. The basic idea is to introduce a term for the correction problem which simulates a high Reynolds number flow in the relaxation direction. One way to achive such a term, implicitly, is to add part of the correction to the approximation on the line just up-steam the line which is being updated. In some sense the method resembels distributive relaxations even though the purpose and the motivations are completely different. Since the term which is added to the correction probelm has a convective character, the scheme is called Convective Successive Line Relaxation (C-SLR).

In C-SLR like in SLR, each line (e.g ξ=constant) is being updated simultaneously. The correction at each point j (on the line i) is δ_j and $\alpha\delta_j$ is added to the variables at the node point i-1,j. If α = 0, then C-SLR is identical to SLR.

A local mode analysis for C-SLR (solving each line ξ=constant) gives that the amplification factor is given by:

$$\mu(\vartheta_1,\vartheta_2) = \left| \frac{C_1(1 - 2\alpha + \alpha e^{i\vartheta_1}) + 2C_2\alpha(\cos\vartheta_2 - 1) - RehNT1}{C_1(\alpha - 2) + 2C_2(\cos\vartheta_2 - 1) + C_1 e^{-i\vartheta_1} - Reh\ NT2} \right| \qquad (22)$$

where

$$
NT1 = \begin{cases} \alpha C_3 & , \text{ if } C_3>0 \\ C_3(1 - \alpha + \alpha e^{i\vartheta_1}), & \text{if } C_3<0 \end{cases} + \begin{cases} C_4\alpha(1 - e^{-i\vartheta_2}), & \text{if } C_4>0 \\ C_4\alpha(e^{i\vartheta_2} - 1) & , \text{ if } C_4<0 \end{cases}
$$

$$
NT2 = \begin{cases} C_3(1 - \alpha) - e^{-i\vartheta_1}, & \text{if } C_3>0 \\ -C_3 & , \text{ if } C_3<0 \end{cases} + \begin{cases} C_4(1 - e^{-i\vartheta_2}) & , \text{ if } C_4>0 \\ C_4(e^{i\vartheta_2} - 1) & , \text{ if } C_4<0 \end{cases}
$$

In Tables I we compare the smoothing factor of the SPR, ξ-SLR (C-SLR with $\alpha = 0$), C-SLR with near optimal value of α ($\alpha_{n.o.}$), and C-SLR with a fixed value of α ($\alpha = 0.25$). In Table I.a the relaxation direction is alligned with the flow direction ($C_3 = C_4 = 1$). In Table I.b the results are for the case when the relaxation and the flow directions are against each other.

TABLE I. - SMOOTHING FACTORS FOR OPERATOR (18)

Table I.a: $C_3 = 1$, $C_4 = 1$

Re_h	C_1/C_2	SPR	SLR α=0	C-SLR $\bar{\mu}$ α_{no}	C-SLR α=.25
	0.05	0.909	0.447	0.277,0.25	0.277
	0.10	0.835	0.447	0.277,0.25	0.277
	0.50	0.567	0.477	0.254,0.30	0.277
0	1.00	0.500	0.447	0.254,0.30	0.277
	5.00	0.721	0.714	0.333,0.50	0.565
	10.00	0.835	0.833	0.467,0.60	0.737
	50.00	0.962	0.962	0.855,0.60	0.937
	0.05	0.333	0.036	0.036,0.00	0.340
	0.10	0.333	0.068	0.068,0.00	0.323
	0.50	0.333	0.243	0.159,0.10	0.224
1	1.00	0.333	0.333	0.154,0.25	0.154
	5.00	0.525	0.620	0.256,0.40	0.420
	10.00	0.670	0.767	0.391,0.60	0.632
	50.00	0.909	0.949	0.785,0.60	0.907
	0.05	0.333	0.004	0.004,0.00	0.493
	0.10	0.333	0.008	0.008,0.00	0.488
	0.50	0.333	0.038	0.038,0.00	0.453
10	1.00	0.333	0.072	0.072,0.00	0.412
	5.00	0.333	0.254	0.119,0.20	0.188
	10.00	0.333	0.414	0.160,0.30	0.162
	50.00	0.671	0.796	0.404,0.60	0.668
	0.05	0.333	0.000	0.000,0.00	0.538
	0.10	0.333	0.001	0.001,0.00	0.538
	0.50	0.333	0.004	0.004,0.00	0.533
100	1.00	0.333	0.008	0.008,0.00	0.527
	5.00	0.333	0.038	0.038,0.00	0.481
	10.00	0.333	0.073	0.073,0.00	0.429
	50.00	0.333	0.277	0.127,0.20	0.185

Table I.b: $C_3 = -1$, $C_4 = -1$

Re_h	C_1/C_2	SPR	SLR $\alpha=0$	C-SLR $\bar{\mu}$ α_{no}	C-SLR $\alpha=.25$
	0.05	0.821	0.954	0.674,0.50	0.752
	0.10	0.807	0.914	0.644,0.50	0.717
	0.50	0.724	0.728	0.494,0.40	0.550
1	1.00	0.667	0.632	0.418,0.40	0.460
	5.00	0.718	0.663	0.347,0.50	0.513
	10.00	0.797	0.784	0.447,0.60	0.672
	50.00	0.945	0.944	0.800,0.60	0.909
	0.05	0.953	0.995	0.717,0.40	0.786
	0.10	0.951	0.990	0.714,0.40	0.782
	0.50	0.934	0.954	0.685,0.40	0.752
10	1.00	0.914	0.914	0.664,0.40	0.717
	5.00	0.839	0.728	0.530,0.40	0.558
	10.00	0.807	0.638	0.439,0.40	0.481
	50.00	0.847	0.825	0.606,0.60	0.739
	0.05	0.995	1.000	0.721,0.40	0.790
	0.10	0.995	0.999	0.720,0.40	0.790
	0.50	0.995	0.995	0.717,0.40	0.786
100	1.00	0.993	0.990	0.714,0.40	0.782
	5.00	0.970	0.954	0.718,0.50	0.752
	10.00	0.951	0.914	0.670,0.40	0.717
	50.00	0.867	0.729	0.534,0.40	0.562

From Tables I it is clear that ξ-SLR is most efficient for large values of Re_h if the flow and the relaxation directions are alligned. In general cases of coordinate transformations C-SLR with α_{no} give better results than SPR or SLR. Even with a constant, non-optimal $\alpha = 0.25$, the results are better than with SPR or SLR (especially for moderate and large Re_h and $C_3 = -1$). It is also noted that for operator (18) the C-SLR needs only marginally more computational effort compared with SLR. For this reason one can expect real, in terms of computational times, improvement in the efficiency when C-SLR is used.

The smoothing efficiency of both pointwise DGS (P-DGS) and line DGS (L-DGS) relaxation schemes are cosidered for an equation of the form:

$$(a\partial_\xi^B + b\partial_\eta^B)u + (c\partial_\xi^B + d\partial_\eta^B)v = 0 \qquad (23)$$

where a,b,c and d are constants.

Local mode analysis has been carried out for both cases.The amplification factor for the P-DGS scheme is

$$\mu(\vartheta_1,\vartheta_2) = \left| \frac{(a\beta + c\gamma)\, e^{i\vartheta_1} + (b\beta + d\gamma)\, e^{i\vartheta_2}}{\sigma - (a\beta + c\gamma)\, e^{-i\vartheta_1} - (b\beta + d\gamma)\, e^{-i\vartheta_2}} \right| \qquad (24)$$

where $\sigma = 2(a^2 + b^2 + c^2 + d^2 + ab + cd)$

and $\beta = a + b$
$\gamma = c + d$

For the case of rectangular coordinates, (with $a = b = 1, c = d = 0$), one gets back the smoothing factor for the Poisson equation. The efficiency of the P-DGS decreases somewhat when all the coefficients are of the same order. The smoothing factor is then equal to 0.632. The worse case is when either a or d are much larger than the other coefficients. In such cases improvement is obtained if L-DGS is used. The amplification factor of L-DGS (solving for a line ξ=constant) is

$$\mu(\vartheta_1,\vartheta_2) = \left| \frac{a\beta + c\gamma}{\sigma - 2(b\beta + d\gamma)\cos\vartheta_2 - (a\beta + c\gamma)\, e^{-i\vartheta_1}} \right| \qquad (25)$$

This L-DGS is most efficient when d is dominant, while L-(η)DGS is most efficient when a is dominant. Alternating direction of L-DGS relaxation sweeps can be used in those cases when the coefficients vary largely in the computational domain.

COMPUTATIONAL RESULTS AND DISCUSSION

At this stage of our work we have tested mainly the efficiency of the relaxation schemes for the momentum equations. These studies are of interest for some coordinate transformations which may be used for practicle problems. Results have also been obtained for the Navier-Stokes equation solver, but these results are of preliminary character, since the transfer among the grids has not been done properly.

In reference 3 a user oriented MG-program for the solution of the Poisson equation in general two dimensional coordinates, is presented. The relative efficiency of several relaxation schemes,applied to the Poisson equation in linearly and exponantially stretched coordinates,have been compared. In that work [3] it has been concluded that:

i. The rate of convergence of different relaxation operators is not sensitive to the number of the net points.
ii. As predicted by local mode analysis, simple η- or ξ-SLR are not efficient on grids where max $|C_1/C_2|>>1$ and min $|C_1/C_2|<<1$.
iii. The AD-SLR method has a rate of convergence which is not sensitive in variations in C_1/C_2.
iv. The C-SLR scheme is superior to usual SLR and AD-SLR schemes except, possibly, when the stretching is very large. In such cases C-AD-SLR is recomended.

These tests are valid also for the Stokes (Re = 0) momentum equations. Recently, the tests have been extented to cover the momentum equations for non-vanishing Reynolds numbers. The following coordinate transformations have been considered:

1. The identity transformation. $\xi = x, \eta = y$.
2. Mesh refinements near the boundaries of a unit square:

$$x = \alpha \mathrm{tgh}(\xi - 0.5)/\varepsilon + 0.5$$

$$y = \alpha \mathrm{tgh}(\eta - 0.5)/\varepsilon + 0.5$$

where $\alpha = 1/(2\mathrm{tgh}\ 0.5/\varepsilon)$

3. Exponentially stretched coordinates (as in reference 3):

$$x = \xi + e^{c\xi}$$

$$y = \eta + e^{c\eta}$$

4. Polar coordinates:

$$x = r \cos\vartheta$$
$$y = r \sin\vartheta$$

and $r = \xi + 1$ and $\vartheta = \eta$.

5. Polar (stretched) coordinates, as in 4, but with:

$$r = 1.5 + \alpha \mathrm{tgh}(\xi - 0.5)/\varepsilon$$
$$\vartheta = \alpha \mathrm{tgh}(\eta - 0.5)/\varepsilon + 0.5$$

Known functions,of order one, are chosen and they define a forcing term in the governing equations in such a way that the exact numerical solution equals these functions. Only MG-cycling is used (starting always with zero initial approximation) and the iterations are continued to an accuracy beyond the truncation errors. In this way the asymptotic convergence factor Θ is obtained.

In the following tables the asymptotic convergence factor Θ, the computational time, T, and the absolute error in the solution, E, are given, for the following cases:

i. Results for transformation 2 ($\varepsilon = 1/3$) are given in Table II.
ii. Results for transformation 3 ($c = 1$) are given in Table III.
iii. Results for transformation 4, are given in Table IV.

TABLE II: TRANSFORMATION 2.

1/h		8	16	32
	Re	Θ T E	Θ T E	Θ T E
SPR	0	0.81,0.12,6E-3	0.86,0.43,6E-2	0.88,1.67.2E-1
	1	0.79,0.16,5E-3	0.86,0.45,6E-2	0.88,1.71,2E-1
	10	0.70,0.10,2E-3	0.84,0.48,2E-2	0.88,1.71,1E-1
	100	0.25,0.03,9E-5	0.61,0.22,6E-4	0.80,1.50,5E-3
SLR	0	0.75,0.13,8E-3	0.83,0.56,5E-2	0.84,2.20,2E-1
	1	0.74,0.14,6E-3	0.82,0.56,5E-2	0.84,2.27,2E-1
	10	0.63,0.09,3E-3	0.80,0.22,1E-2	0.84,2.24,1E-1
	100	0.29,0.04,3E-4	0.57,0.24,3E-4	0.75,2.04,3E-3
C-SLR	0	0.56,0.07,6E-4	0.64,0.37,8E-4	0.68,1.92,1E-3
	1	0.54,0.06,3E-3	0.64,0.40,1E-3	0.68,1.92,1E-3
	10	0.34,0.03,5E-4	0.59,0.29,1E-3	0.64,1.57,9E-4
	100[a]	0.29,0.03,3E-4	0.57,0.26,3E-4	0.75,1.75,3E-3

[a]The convectivity coefficient in C-SLR is taken to be zero.

TABLE III: TRANSFORMATION 3.

1/h		8	16	32
	Re	Θ T E	Θ T E	Θ T E
SPR	0	0.56,0.05,4E-3	0.57,0.17,2E-3	0.60,0.91,3E-4
	1	0.58,0.07,1E-3	0.59,0.23,2E-3	0.61,0.89,2E-3
	10	0.48,0.03,1E-3	0.59,0.19,2E-3	0.66,1.00,1E-3
	100	0.26,0.03,9E-5	0.26,0.08,5E-4	0.50,0.65,8E-5
SLR	0	0.50,0.05,2E-3	0.60,0.33,5E-4	0.64,1.56,2E-3
	1	0.51,0.05,1E-3	0.60,0.32,1E-3	0.65,1.58,1E-3
	10	0.36,0.04,3E-4	0.52,0.21,6E-4	0.60,1.28,3E-4
	100	0.31,0.03,2E-4	0.21,0.09,5E-4	0.33,0.63,5E-5
C-SLR	0	0.41,0.04,1E-3	0.46,0.21,5E-4	0.45,0.90,8E-4
	1	0.53,0.04,1E-3	0.52,0.25,6E-4	0.52,1.33,4E-4
	10[a]	0.36,0.03,3E-4	0.52,0.21,6E-4	0.60,1.36,3E-4
	100[a]	0.31,0.04,2E-4	0.21,0.11,5E-4	0.33,0.68,5E-5

TABLE IV: TRANSFORMATION 4.

1/h		8	16	32
	Re	Θ T E	Θ T E	Θ T E
SPR	0	0.62,0.06,3E-3	0.64,0.27,1E-3	0.67,1.15,1E-3
	1	0.62,0.06,3E-3	0.63,0.23,5E-3	0.66,1.17,1E-3
	10	0.54,0.06,1E-3	0.61,0.21,2E-3	0.64,1.04,1E-3
	100	0.24,0.01,3E-3	0.45,0.12,2E-4	0.56,0.77,3E-4
SLR	0	0.56,0.06,2E-3	0.63,0.34,6E-4	0.68,1.67,1E-3
	1	0.56,0.06,4E-3	0.63,0.32,2E-3	0.68,1.78,8E-4
	10	0.54,0.06,2E-3	0.64,0.40,9E-4	0.66,1.67,9E-4
	100	0.23,0.02,1E-4	0.40,0.17,1E-4	0.54,1.08,1E-4
C-SLR	0	0.62,0.06,1E-3	0.43,0.22,1E-3	0.46,0.93,2E-3
	1	0.46,0.04,1E-3	0.46,0.22,2E-3	0.44,1.14,5E-4
	10	0.32,0.02,3E-4	0.51,0.25,3E-4	0.54,1.21,7E-4
	100[a]	0.23,0.03,1E-4	0.40,0.19,1E-4	0.54,1.12,1E-4

[a]The convectivity coefficient in C-SLR is taken to be zero.

It is noted from Table II that the SPR and the SLR schemes do not converge on the finest grid. The C-SLR reults even in this cases in an acceptable convergence factor. For the C-SLR only three different values of the parameter α are used in the actual computations, and hence the C-SLR has not been optimal. Furthermore, all the results (in Tables II-IV) have been obtained with the same values of the MG control parameters, which are nearly optimal for SPR in cartesian coordinates. Improved results for C-SLR are obtained when other MG control parameters are used. The C-SLR gives a real computational gain (not only in terms of convergence factor, but also in terms of computational times) in general, and especially in cases similar to Transformation 2. For simple cartesian cases, however, SPR results in the shortest computational times.

The Navier-Stokes solver has been tested for the computation of the flow in rectangular-and polar-driven cavities. The code does not include,yet, the correct transfer of the dependent variables to coarse grids, and hence one can expect a reduction in the total efficiency. The problem has been solved on several grids, and the convergence factor is found to be about the same for both cavities and the different grids. A comparison of these preliminary results for the (non-optimal) C-SLR and the SPR schemes is given in Table v.

TABLE V: CONVERGENCE FACTORS FOR DRIVEN CAVITIES.

Re	C-SLR	SPR
0	0.78	0.76
1	0.74	0.84
10	0.72	0.83

As expected the C-SLR gives somewhat better results than the SPR scheme. However, the results are not satisfactory and effort is being made to improve the overall effieciency of the Navier-Stokes solver.

CONCLUDING REMARKS

The progress which has been made in developing a general purpose two dimensional Navier-Stokes solver is reported here. The computational code which gives good results, is not considered to be in its final form and additional improvement is expected. Further work is being done on the extebsion of the method to problems which include planes of symmetry where mixed boundary conditions are to be applied.

REFERENCES

1. Brandt, Achi: Multi-Level Adaptive Solution to Boundary Value Problems. Math. Comp. vol. 31, 1977, pp. 333-390.
2. Brandt, Achi;Dinar, Natan: MultiGrid Solution to Elliptic Flow Problems. Numerical Methods in PDE, Ed. Bramble, J.H., Academic Press, 1979,pp. 53-147.
3. Fuchs, Laszlo: A Fast Numerical Method for the Solution of Boundary Value Problems. TRITA-GAD-2, 1980.
4. Fuchs, Laszlo: A Newton-Multi-Grid Method for the Solution of Non-linear PDE, Boundary and Interior Layers-Computational and Asymptotic Methods, Ed. Miller,J.J.H, Boole Press, 1980, pp. 291-296.
5. Fuchs, Laszlo: Finite-Difference Methods for Plane Steady Inviscid Transonic Flows. TRITA-GAD-2, 1977.
6. Fuchs, Laszlo: Transonic Flow Computation by a Multi-Grid Method. Notes on Numerical Fluid Mechanincs, Eds. Rizzi, A.W. and Viviand, H. vol. 3, Vieveg Verlag, 1981. pp. 58-65.
7. Jameson, Antony: Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method. AIAA paper 79-1458, 1979.

8. Harlow, F.H.;Welch, J.E: Numerical Calculation of Time-Dependent Viscous Incompressible Flow. Phys. of Fluids. vol. 8, 1965, pp.2182-2187.
9. Thunell, Tomas;Fuchs, Laszlo: Numerical solution of the Navier-Stokes Equations by Multi-Grid Techniques. Numerical Methods in Laminar and Turbulent Flow. Taylor, Cedrick: Schrefler, J.W. Eds. Pineridge Press, 1981, pp. 141-152.
10. Fuchs, Laszlo; Zhao, Heshu; Numerical Simulation of Three-Dimensional Flows in Ducts. Numerical Methods in Laminar and Turbulent Flow. Taylor, C.: Schrefler, J.W. Eds. Pineridge Press, 1981. pp. 167-178.
11. Roache, Patrick. J.: Computational Fluid Dynamics, Hermose Publishers, 1972.

APPLICATION OF THE MULTI-GRID METHOD TO CALCULATIONS OF TRANSONIC POTENTIAL FLOW ABOUT WING-FUSELAGE COMBINATIONS*

Arvin Shmilovich and D. A. Caughey
Cornell University

ABSTRACT

The Multi-Grid (MG) method has been applied to the calculation of transonic, potential flowfields about arbitrary, three-dimensional, wing-body combinations. Numerical results for iterative convergence rate are in good agreement with those predicted by a local mode analysis, and show substantial improvement over conventional relaxation algorithms.

NOMENCLATURE

a = speed of sound
a,b,R = normalization parameters for the angular coordinate (equation (19))
A,B,C,D,E,H = coefficients of second derivatives for the potential equation (28) in the transformed cylindrical coordinates
A_X, A_Y, A_Z = recoupling coefficients
A_η, A_r = cell aspect ratios in η and r directions, relative to the width of cell in ξ direction, respectively
c = chord length
C_p = pressure coefficients
d = dimension of the problem
F = forcing function
g^{ij} = elements of inverse of the metric tensor
G = growth factor
G^k = k^{th} MG level
$\overline{G}, \overline{\overline{G}}$ = maximum growth factor on finest grid per iteration and per MG cycle, respectively
h = mesh spacing
H,h = Jacobian of the transformation and its determinant, respectively
I = interpolation operator
k = ratio of specific heats
M = Mach number
M_c = "cut off" Mach number
p = wave number
P,Q,R = artificial viscosities fluxes
$\hat{P}, \hat{Q}, \hat{R}$ = terms used for constructing the P,Q,R fluxes, respectively
$\overline{q}$ = velocity vector
R_f, R_t = radial coordinate of fuselage surface and wing tip, respectively
s = coordinate tangent to streamline
S = width of strip in conformally mapped plane (figure 2b)
T = recoupling term
u,v,w = velocity components in x,y,z coordinates, respectively
U,V,W = contravariant velocity components in X,Y,Z coordinates, respectively
W_k = work spent of the k^{th} MG level to reduce the residuals to within the truncation error of the k^{th} grid
$\overline{W}_K$ = total work (estimated) for obtaining a solution to the level of the truncation error
x,y,z = Cartesian coordinates

* This work has been supported by the Office of Naval Research under Contract N00014-77-C-0033.

x,r,θ = cylindrical coordinates

$\overline{x},\overline{r},\overline{\theta}$ = normalized cylindrical coordinates

X,Y,Z = coordinates of the computational domain

α = modes (equation (29))

δ = central difference operator

ε = recoupling parameter

μ = artificial viscosity switching function (equation (16))

μ = averaging operator

ξ,η = coordinates in mapped cylindrical plane (equation (20))

ρ = density

$\rho_{\alpha,\beta,\gamma}$ = weighting coefficients for the residual transfer (figure 4a)

τ = truncation error

ϕ = velocity potential

ω = overrelaxation factor

Subscripts

$(\ )_\infty$ = value at upstream infinity

$(\ )_s$ = coordinates of the singular line

I. INTRODUCTION

The MG method has been shown effective in speeding up the convergence of relaxation solutions of the difference equations arising from discrete approximations to problems of elliptic type (refs. 1-5). Less attention has been focussed on non-elliptic problems. The advantages of the method for problems of mixed type have been demonstrated by South and Brandt (ref. 6), who treated the two-dimensional transonic small disturbance equation for the non-lifting flow past a parabolic arc airfoil. Substantial deterioration in performance of the MG method has been encountered when using successive line overrelaxation (SLOR) on stretched grids. Jameson (ref. 7) applied the MG method to the two-dimensional potential equation using an alternative-direction-implicit (ADI) smoothing algorithm, and obtained good rates of convergence even on highly stretched grids.

The extension of the MG method to three-dimensional calculations seems attractive, since the process of eliminating the persistent low frequency components of the error using conventional relaxation techniques is expensive, and the work required for a relaxation sweep on a coarser grid is only 1/8 of that on the preceding grid when the grid spacing is doubled in each direction. Recent work on an MG code for three-dimensional transonic flow about axisymmetric inlets has been reported by McCarthy and Reyhner in reference 8.

An existing three-dimensional transonic potential code designed by Caughey and Jameson has been modified to accommodate the MG procedure. The code is called FLO 30, and solves a fully conservative difference approximation in a boundary-conforming coordinate system.

Experience gained in two-dimensional numerical calculations, both from the programming and predictability aspects, guided us in carrying out the work reported herein. An attempt is made to predict the performance of the accelerated iterative scheme by means of the local mode analysis. By an a priori knowledge of the rate of convergence, a stopping criterion can be established.

In the following sections we describe the finite volume method and the grid generation procedure (refs. 9-11) to the extent that is needed for understanding the features associated with the incorporation of the MG technique into the existing code. The MG procedure is reviewed as well, emphasizing the direct implications by those aspects relevant to the problem under consideration. Theoretical estimates of the MG performance on rather complicated grids are discussed. Numerical calculations demonstrating the beneficial effect of the MG technique in accelerating the original relaxation scheme are presented, and the validity of the theoretical estimates is confirmed.

II. ANALYSIS

A. Finite Volume Method

A detailed exposition of the finite volume method devised by Jameson and Caughey may be found in reference 9.

The continuity equation for steady-inviscid, isentropic flow in Cartesian coordinates x,y,z reads

$$(\rho\phi_x)_x + (\rho\phi_y)_y + (\rho\phi_z)_z = 0, \tag{1}$$

where ϕ is the velocity potential. The density ρ is given by the isentropic relation

$$\rho = (1 + \frac{k-1}{2} M_\infty^2(1-q^2))^{1/k-1}, \tag{2}$$

where k is the ratio of specific heats and M_∞ is the free stream Mach number. The description of the velocity $\overline{q} = (u,v,w)$ in terms of a scalar potential

$$\overline{q} = \nabla\phi, \tag{3}$$

is a consequence of the assumption that the flowfield contains no strong shocks, so that the flow may be regarded as being irrotational.

The finite volume method does not require knowledge of the global nature of the transformation which generates the grid network, but uses only local properties of the transformation. We introduce an arbitrary transformation to a new coordinate system X,Y,Z and define the Jacobian H

$$H = \begin{Bmatrix} x_X & x_Y & x_Z \\ y_X & y_Y & y_Z \\ z_X & z_Y & z_Z \end{Bmatrix}, \tag{4}$$

with its determinant

$$h = \det(H).$$

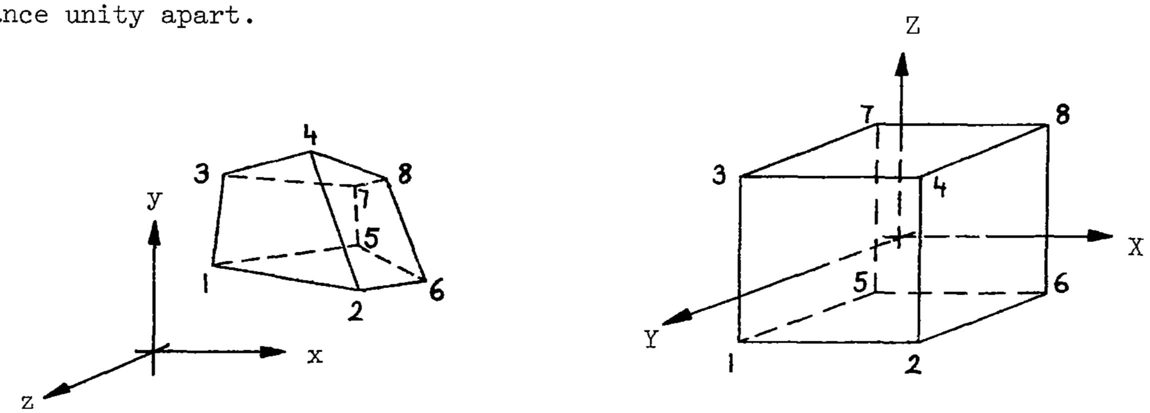
The contravariant components U,V,W of the velocity can be expressed as

$$\begin{Bmatrix} U \\ V \\ W \end{Bmatrix} = H^{-1} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = (H^T H)^{-1} \begin{Bmatrix} \phi_X \\ \phi_Y \\ \phi_Z \end{Bmatrix} , \tag{5}$$

where $H^T H$ represents the metric tensor. Under the transformation, the continuity equation becomes

$$(\rho hU)_X + (\rho hV)_Y + (\rho hW)_Z = 0. \tag{6}$$

It is convenient to utilize a transformation that locally converts an arbitrary cell in the physical space into a cube in the computational domain, such that its center is located at the origin and its vertices are at a distance unity apart.



The simplest such mapping assumes a trilinear variation of the coordinates and the potential, within each cell. Thus, the shape function for the x coordinate, say, is

$$x = 8 \sum_{i=1}^{8} x_i(\tfrac{1}{4} - XX_i)(\tfrac{1}{4} - YY_i)(\tfrac{1}{4} - ZZ_i), \tag{7}$$

i denoting the i^{th} vertex of the cell. It can be verified that such formulas yield expressions for the derivatives such as

$$x_X = \tfrac{1}{4}\,(x_2 - x_1 + x_4 - x_3 + x_6 - x_5 + x_8 - x_7) \tag{8}$$

when evaluated at the centers of the mesh cells. Thus, the Jacobian and the contravariant velocity vector may be readily calculated. For the sake of simplifying the notation, let us introduce the averaging and differencing operators

$$\mu_x f = \tfrac{1}{2}\,(f_{i+1/2,j,k} + f_{i-1/2,j,k})$$

$$\delta_X f = f_{i+1/2,j,k} - f_{i-1/2,j,k} \tag{9}$$

With this notation, the approximations for the derivatives can be written as

$$x_X = \mu_{YZ}\delta_X x \tag{10}$$

with similar expressions for derivatives of y,z,ϕ and of derivatives in the other directions. Taking a second difference of the contravariant fluxes, the continuity equation is cast in the form

$$\mu_{YZ}\delta_X(\rho hU) + \mu_{ZX}\delta_Y(\rho hV) + \mu_{XY}\delta_Z(\rho hW) = 0. \tag{11}$$

This formula can be interpreted as conserving mass fluxes in an auxiliary cell which overlaps eight primary cells, having vertices at the centers of the primary cells.

Since the fluxes are calculated by averaging the values of the cell centers rather than using values evaluated at the face centers of the auxiliary cells, this formulation tends to decouple the solution at alternate points of the grid. In order to compensate for this decoupling we effectively shift the point of evaluation of the fluxes to the center of the faces of the auxiliary cell by adding one term of their expansions about the centers of the mesh cells. The added term is of the form

$$\begin{aligned} T = -\varepsilon(&\mu_Z\delta_{XY}(A_X+A_Y)\mu_Z\delta_{XY} + \mu_X\delta_{YZ}(A_Y+A_Z)\mu_X\delta_{YZ} \\ &+ \mu_Y\delta_{ZX}(A_Z+A_X)\mu_Y\delta_{ZX} - \frac{1}{2}\,\delta_{XYZ}(A_X+A_Y+A_Z)\delta_{XYZ})\phi, \end{aligned} \tag{12}$$

where $A_X = \rho h(g^{11} - U^2/a^2)$ and similar formulas hold for A_Y, A_Z. Here a is the speed of sound. The g^{ij} are the elements of the inverse of the metric tensor, and $0 \le \varepsilon \le 1/2$. In practice $\varepsilon = 1/2$ is generally used, representing the strongest recoupling.

In order to properly reflect the correct domain of dependence in supersonic regions of the flowfield, it is necessary to introduce an artificial viscosity. Since the local flow direction is not known in advance, and we want the directional bias to be added in the streamwise direction, we make use of Jameson's rotated scheme (ref. 12). Consider the potential equation in quasilinear form written in coordinates locally aligned with the velocity vector

$$(a^2 - q^2)\phi_{ss} + a^2(\nabla^2 - \phi_{ss}) = 0 , \tag{13}$$

where s is a coordinate tangent to the streamline. The upwinding of the ϕ_{ss} term can be accomplished by explicitly adding an appropriate artificial viscosity to the central difference approximation. The addition of such a term in divergence form can be shown to be analogous to the following modified numerical scheme

$$(\rho hU+P)_X + (\rho hV+Q)_Y + (\rho hW+R)_Z + T = 0. \tag{14}$$

The added flux P is constructed from

$$\hat{P} = \mu \frac{\rho h}{a^2} (U^2\delta_{XX} + UV\mu_{XY}\delta_{XY} + WU\mu_{ZX}\delta_{ZX})\phi$$

by defining

$$P_{i+1/2,j,k} = \begin{cases} \hat{P}_{i,j,k} & \text{if } U > 0 \\ -\hat{P}_{i+1,j,k} & \text{if } U < 0 \ . \end{cases} \tag{15}$$

The fluxes Q and R are constructed in an analogous fashion. Here, a switching function μ has been introduced

$$\mu = \max(0,\ 1 - \frac{M_c^2}{q^2}), \tag{16}$$

such that the directional bias is added in regions where the local speed exceeds the value of some "cut off" Mach number M_c. It has been observed that the MG technique functions at its best when the upwinding is switched on even in a region slightly larger than the supersonic pocket. In practice $M_c^2 = .9$ is generally used.

The solution of the nonlinear algebraic equations resulting from the discretization is accomplished by the formulation of an iterative scheme, embedding the steady state equation in an artificial time-dependent equation.

B. Grid Generation

The major difficulty in treating the full potential equation rather than its small perturbation approximation is to correctly satisfy the boundary conditions. This can be done easily if the difference equations are solved in a boundary conforming coordinate system. An essential advantage of the finite volume method is the decoupling of the grid generation step from the iterative procedure, since only local properties of the transformation are used. Nevertheless, it is often convenient to generate the coordinate grid by sequences of conformal and shearing transformations for a variety of practical problems.

Consider the configuration shown in figure 1, consisting of a wing mounted upon a fuselage of varying cross sectional shape. We assume the flow is symmetrical about the vertical plane passing through the fuselage centerline so that only the flow in half space $z \geq 0$ need be considered.

Denoting the fuselage surface by $R_f(x,\theta)$, the radial coordinate is normalized out by defining

$$\bar{r} = \frac{r - R_f(x,\theta)}{r - R_t}\ , \tag{17}$$

R_t being the radial coordinate of the wing tip. The wing sweep and dihedral are normalized out by introducing the coordinates:

$$\bar{x} = \frac{x - x_{s(\bar{r})}}{c_{(\bar{r})}} + \log 2$$

$$\bar{\theta} = 2(b \mp \sqrt{R^2 - (\theta - a)^2}) \qquad (18)$$

where

$$a = \frac{2\theta_s^2 - \pi^2}{4\theta_s} = -b$$

$$R^2 = (\theta_s - a)^2 + a^2 \qquad (19)$$

and the $\mp$ sign is taken depending upon whether θ_s is positive or negative. Here $c_{(\bar{r})}$ is the local chord length, and $x_{s(\bar{r})}$, $\theta_{s(\bar{r})}$ represent the location of a singular line just inside the leading edge of the wing. The singular line is then used as a branch point in the conformal mapping

$$\bar{x} + 2i\bar{\theta} = \log(1 - \cosh(\xi + i\eta)) \qquad (20)$$

to "unwrap" the wing surface. Under this transformation a surface of constant $\bar{r}$ that intersects the wing surface, shown in figure 2a, will take the form depicted in figure 2b. A final shearing transformation

$$X = \xi$$
$$Y = \eta / S_{(\xi, \bar{r})}$$
$$Z = \bar{r} \qquad (21)$$

reduces the strip of width $S_{(\xi, \bar{r})}$ to one of constant width, as shown in figure 2c, resulting in a nearly orthogonal mesh if the location of the singular line has been carefully chosen. The computational domain shown schematically in figure 3 is rendered finite by suitable stretchings in the X and Z directions. The X stretching is set up at each spanwise station, so that the far downstream boundary remains a plane in the physical domain, even if the wing is tapered and/or swept.

The definition of $R_{f(x,\theta)}$ and $S_{(\xi,r)}$ from the input data of the fuselage and wing geometry is achieved by spline fits. A spline fit in x is applied for interpolating the coefficients of the Fourier decomposition of the fuselage cross sections. Spline fits in the spanwise and ξ directions are employed to define $S_{(\xi,\bar{r})}$. Having defined R_f and S, we proceed with calculating the physical coordinates of the grid points by reversing the mapping sequence.

C. Boundary Conditions

The treatment of boundary conditions in a boundary conforming coordinate system is quite easily accomplished since the finite difference scheme is formulated in terms of the contravariant velocity vector (U,V,W). To enforce the no flux condition across the solid surfaces (the fuselage and the wing) the normal component of the velocity vector must be zero. This is incorporated by reflecting the normal flux contributions for the cells adjacent to the boundary.

The algorithm is simplified by introducing a reduced velocity potential representing perturbations from the free stream. This potential is set to zero on the upstream and lateral far field boundaries. On the far downstream boundary, the first derivative in the streamwise direction of the perturbation potential is set to zero; this is equivalent to setting the streamwise velocity component to its free stream value. This boundary condition is a consequence of the fact that the flow becomes fully developed for downstream.

To account for lift, a vortex sheet emanating from the trailing edge of the wing must be allowed. It is assumed that the trailing vortex sheet lies along the branch cut (the dashed line in figure 2b), thus convection and roll-up are ignored. On the two sides of the sheet we require that mass be conserved. For this purpose it is convenient to introduce dummy points above the boundary, which are identified in the physical domain with corresponding interior points on the other side of the cut. To envision this, imagine rotating the left branch cut (in figure 2b) in the counter-clockwise direction by 180°, about the singular line, thus obtaining the physical plane in figure 2a. For conserving the mass in cells whose centers lie on the cut we calculate the contribution to the fluxes at the centers of the dummy mesh cells, from the values of the potential and the coordinates of the corresponding cells reflected about the origin. Points on both sides of the cut are treated as interior points by the same iterative algorithm. This procedure can also be used when the cut is a vortex sheet across which the jump in the potential is determined by the Kutta condition.

In the original program it was required that the normal velocity component be continuous across the vortex sheet: $V_Y = 0$. This condition was applied also on the branch cut outboard of the tip of the wing. (This portion of the cut has no physical meaning.) In the modified code mass is conserved on points that lie on the vortex sheet inboard of the wing tip, while $V_Y = 0$ was posed on the outboard part of the cut. This special treatment of the branch cut provided best results when using the MG algorithm.

D. Multi-Grid Technique and Implementation Aspects

An extensive discussion of the MG technique by Brandt in reference 5 is very illuminating. In addition, we suggest other relevant references (refs. 3, 4) as background to our rather succinct presentation.

The MG method relies on the fact that relaxation schemes are efficient in eliminating those components of the error whose wavelengths are comparable to the mesh spacing. However, the process of liquidating the lower frequency modes is characterized by a slow rate of convergence since the effective signal speed on the fine grid is slow. The basis of the MG method is to discretize the problem in a sequence of grids of different mesh widths, allowing simultaneous treatment of the whole spectrum of error modes. This greatly speeds up the convergence of the relaxation scheme. Moreover, solving on coarser grids requires far less computational effort, since the mesh points are fewer.

We proceed to describe the logical sequence of the MG procedure. Denote the discretization of the scheme in equation (14) on a hierarchy of grids G^0, G^1, $G^2, \dots,$ G^K of varying coarseness, by

$$L^k \phi^k = F^k, \quad k = 0,1,2,\dots,K , \tag{22}$$

K designating the finest grid, so that $F^K = 0$. We start the iteration on the finest grid with the aid of some initial estimate. When slow convergence is sensed, the relaxation process is discontinued on this grid. Exploiting the smoothness of the tentative solution obtained, we carry out relaxation on the next coarser grid. While ϕ^k is an approximate solution on G^k, it cannot be expected to be a good approximation on G^{k-1}, because of differences in the discretization errors of the two grids. The link between the grid levels is made by using a forcing term which accounts for the difference between the truncation errors of the two grids. Thus

$$L^{k-1} \phi^k = F^{k-1}, \tag{23}$$

where

$$F^{k-1} = I_k^{k-1} F^k + \tau_k^{k-1} \tag{24}$$

and τ_k^{k-1} is the truncation error of the coarse grid relative to the fine grid,

$$\tau_k^{k-1} = L^{k-1}(I_k^{k-1} \phi^k) - I_k^{k-1} L^k \phi^k. \tag{25}$$

Here I_k^{k-1} denote interpolation operators from the fine grid to the next coarsest level. It should be noted that $I_k^{k-1}(F^k - L^k \phi^k)$ is the residual left by ϕ^k. The operator for the residual interpolation is not necessarily the same as that for the solution transfer.

Having calculated an approximate solution on the coarse grid, an updated solution of the fine grid may be obtained. Simply interpolating ϕ^{k-1} to the fine grid, however, cannot be done, since this would cause the high frequency components of the solution to be lost. These components can be maintained by adding to the recent solution ϕ^k the contribution of low frequency components to the correction, namely the difference between the updated solution on the coarse grid ϕ^{k-1}, and its estimate $I_k^{k-1}\phi^k$. Thus, an improved solution on the fine grid is

$$\phi^k_{new} = \phi^k + I^k_{k-1}(\phi^{k-1} - I_k^{k-1}\phi^k). \tag{26}$$

Note that ϕ^k needs to be saved. Alternatively, equation (26) can be written as

$$\phi^k_{new} = I^k_{k-1}\phi^{k-1} + (\phi^k - I^k_{k-1} I_k^{k-1} \phi^k),$$

and the updated solution can be interpreted as follows: transfer the updated solution ϕ^k to G^{k-1} and interpolate it back to the fine grid ($I^k_{k-1} I^{k-1}_k \phi^k$ now contains only low frequency components); subtract the result from ϕ^k to form the contribution of high frequency components to the correction.

Error components of arbitrarily low wavenumbers can be diminished by extending the above sequence onto yet coarser grids.

Calculations have been performed on grids of 64×16×16 cells in X,Y,Z directions, respectively. The grid was coarsened by eliminating every other mesh point in each direction. Most often a sequence of four grid levels was employed in the MG process. The set up of the program admits the use of a fifth level (4×1×1), one that takes the wing to be infinite.

The equations resulting from the discretization in equation (14) are sequentially solved on planes of constant Z (marching from the fuselage towards the lateral boundary), each of which is swept by successive line overrelaxation along lines of constant X (XSLOR). Because of the local nature of relaxation schemes, it is convenient to store in virtual memory the coordinates and the solution of only the Z plane being swept and its two neighboring planes on either side, plus the old solution of the preceding plane. The coordinates and the solution on the entire domain are stored on a disk and information is buffered in and out of the virtual memory as needed, while calculations are being performed. Disk manipulation requires careful programming for transferring the potential and the weighted residuals to the next coarser grid, and in interpolating the corrections to the next finer level. The interpolation of the potential to the next coarser level (in equations (23) and (26)) is done by 'injection', i.e., values of the potential from the fine grid are transferred at points corresponding to both levels. Three fine grid planes contribute to the construction of the volume average of the residuals (in equation (24)) as sketched in figure 4a. Four coarse grid planes participate in forming the four point Lagrangian interpolation in the Z direction – see figure 4b. The same interpolation operator is used within each of these planes to improve the solution at each of the fine grid points that lie on them (in equation (26)). The buffering of the potential for interpolating in the Z direction is somewhat complicated in that solution from the first and third preceding planes must be available.

A fixed strategy using one sweep on each visited grid has been shown to be effective (ref. 7). The domain is swept once on each grid level until the coarsest grid is reached. Each level is swept once after coarse grid corrections are added while backing up to the second finest grid. This completes a MG cycle. Thus, the work required for one MG cycle for a problem in d-space dimensions is

$$1 + 2\left(\frac{1}{2^d} + \frac{1}{2^{2d}} + \frac{1}{2^{3d}} + \dots\right) \leq 1 + \frac{2}{2^d - 1} \quad \text{units,} \tag{27}$$

where 1 unit is the work required for a fine grid sweep (ignoring the overhead due to transferring and interpolation). For a three-dimensional problem, this amounts to $1^2/7$ work units. The use of a fixed strategy rather than an adaptive one simplifies the coding. Also, no switching criteria whose determination would have required numerous numerical experiments, need be specified.

The adaptation of the MG procedure calls for an additional storage of $2/(2^d-1)$, which amounts to 2/7 of the storage needed for the solution of the fine grid. Half of this space is needed for storing the potential of all levels except the finest, while the other half is needed for the residuals on these levels.

The need to use the buffering procedure is due to the limited virtual memory of the computer in use. Since only the potential and the coordinates of the points corresponding to the grid being relaxed need be retrieved, it follows that the buffering procedure employed is inefficient on coarse grids. This procedure was appropriate for the original code. We could have chosen one of the following three options for the modification of the retrieval procedure of the coordinates:

-Making use of a special routine that coarsens/refines the grid. This would have still required the retrieval of the coordinates of the finest grid when relaxing on any level.

-Utilizing the 'random' access mode for buffering in just the coordinates of the points of the grid under treatment, skipping all the others. This is a quite expensive operation since the mode consists of a searching operator in addition to the retrieving operator.

-Taking advantage of the fixed strategy, by constructing a disk file that contains the stored coordinates of all levels in a preset order. The arrangement is so made as to coincide with the strategy used. More specifically, the coordinates of the fine grid are put at the head of the disk, followed by the coordinates of the coarser grids in sequential order, down to the coarsest level. The coordinates of every level are stored twice, because the grid is once relaxed and then swept for the calculation of the residuals. Next, the coordinates of the grids are stored in the reverse order, up to the second finest grid. The finest level need be stored only once, since for the residual claculation (for transferring to the next coarse level) the disk can be inexpensively rewound.

Initial estimates indicated that the second option should be more efficient than the first, and it was coded. Subsequently, the third alternative has been incorporated into the code, exhibiting an additional thirty percent reduction in the cost of computation. Adopting this option implies 3/7 storage extension of the space required for storing the fine grid coordinates. This does not cause any problems since here we utilize the disk storage. The use of a computer of larger storage capacity (either real or virtual) would allow the coordinates of the fine grid and the potential of all levels to be stored in memory, eliminating the need for the buffering.

Special attention must be paid to the handling of the boundary conditions when implementing the MG procedure. As formerly described, the incorporation of the boundary conditions on solid surfaces and on the vortex sheet allows use of the same algorithm as at internal points. This treatment proves adequate for the adaptability of the MG, since it is non-disturbing to the interior smoothness. The Dirichlet boundary condition on the upstream and lateral boundaries does not affect the smoothness of the solution at interior points, either. However, difficulty was encountered when imposing the Neumann condition for the velocity potential on the downstream boundary (which was done by setting the potential on the boundary plane to its value on the third plane upstream of the boundary). This difficulty was overcome by invoking the Neumann condition directly in the following manner: fictitious cells are introduced next to the downstream boundary; their velocities in the streamwise direction (at the center of the cells) are calculated by extrapolation of the velocities at the centers of the cells of the immediate interior cells and the free stream velocity on the boundary; the standard algorithm is then applied for calculating the potential on the downstream boundary. Note the similarity of this approach to that used for the calculations at points on the solid surfaces and the vortex sheet. The special operator (for $V_Y = 0$) applied on the cut outboard of the wing tip requires careful treatment. The residuals at these points must be correctly scaled in order to make them comparable to the residuals at neighboring points.

Our recommendation is that for a well-coded MG program the solution at all points of the computational domain including boundary points is to be calculated by the standard relaxation algorithm, excluding the points whose specified conditions are of the Dirichlet type. If special operators need to be devised on some boundaries, caution is required when implementing MG, to guarantee smoothness of the solution elsewhere.

E. Predictability

A local mode analysis provides a reliable estimate of the MG performance. It is assumed that the correction can be represented by a Fourier series, and it is of interest to follow the evolution of the amplitude of one mode. It is further assumed that periodic boundary conditions are specified. Noting that equation (6) is equivalent to $\rho h/a^2$ times equation (13), the potential equation in quasilinear form is utilized to construct the iterative scheme. Under the transformation in equation (20), the potential equation in cylindrical coordinates reads

$$A\phi_{\xi\xi} + B\phi_{\eta\eta} + C\phi_{rr} + D\phi_{\xi\eta} + E\phi_{\xi r} + H\phi_{\eta r} = = F_{(\phi_\xi,\phi_\eta,\phi_r)}, \tag{28}$$

the coefficients being functions of a, u_x, u_θ, u_r, r and the derivatives of the transformation. Locally freezing the coefficients about the available solution (the Fourier analysis being useful only for linear schemes), the XSLOR scheme yields for the growth factor G (the reduction of the error amplitude during one iteration)

$$G(\alpha_1,\alpha_2,\alpha_3;A_\eta,A_r,\omega) = \left\{A_\eta^2[A_r^2(2 - \frac{2}{\omega} - e^{i\alpha_1}) + \frac{C}{A}(2 - \frac{2}{\omega} - e^{i\alpha_3})]\right.$$

$$\left. - \frac{A_\eta A_r}{A}[DA_r \sin\alpha_1 \sin\alpha_2 + EA_\eta \sin\alpha_1 \sin\alpha_3 + H\sin\alpha_2 \sin\alpha_3]\right\}$$

$$\times \left\{A_\eta^2[A_r^2(\frac{2}{\omega} - e^{-i\alpha_1}) + \frac{C}{A}(\frac{2}{\omega} - e^{-i\alpha_3})] + 2A_r^2 \frac{B}{A}(1 - \cos\alpha_2)\right\}^{-1} \quad (29)$$

where:

ω - overrelaxation factor
$\alpha_i = p_i h_i$, $i = 1,2,3$ in ξ,η,r
p_i - wave number
h_i - mesh spacing

for subsonic regions, with a similar expression for hyperbolic regions. The growth factor is thus seen to be strongly dependent on the aspect ratios of the cells: $A_\eta = \Delta\eta/\Delta\xi$, $A_r = \Delta r/\Delta\xi$.

The assumption of periodic boundary conditions implicit in these estimates can introduce substantial errors in problems of practical interest. Therefore one cannot always extract accurate predictions of rates of convergence for conventional relaxation schemes. In particular, the Fourier analysis is not accurate for low wavenumber modes, since they are affected by the boundaries of the domain. In contrast, the analysis is most reliable for those high wavenumber harmonics which interact at short distances. On this hinges the reliability of the estimates for MG; consideration need be given only to harmonics in the range $\pi/2$ to π (from the wavelength of four times the mesh spacing to the smallest discernible wavelength), since the low frequency components of the error are effectively eliminated by the relaxation processes performed on the coarser grids in much less work. Therefore

$$\overline{G} = \max G(\alpha_i;A_\eta,A_r,\omega), \quad \pi/2 \leq \alpha_i \leq \pi \quad (30)$$

provides an estimate for the rate of convergence of the MG algorithm. For the fixed strategy we have used, $\overline{\overline{G}} = \overline{G}^{7/9}$ is the growth factor per work unit.

Equation (29) is rather complicated for analytical treatment for seeking $\overline{G}$. The worst growth factor (for a general relaxation scheme) can be found by inspecting modes of possible combinations of 0, $\pi/2$ and π for extreme values of the aspect ratios. This yields comparatively simple expressions for G. In figure 5, G is plotted against A_η for extreme values of A_r (0 and ∞) for a specified subsonic uniform stream (the velocity vector has very little effect on G), the coordinate r and the overrelaxation factor ω. The choice of r is not important since it is inversely related to A_r and it can be absorbed in the definition of A_r (which is checked for 0 and ∞ anyway). These results suggest that MG is capable of significantly accelerating the original code and that regardless of the mesh in use, the upper bound for the growth factor is approximately 0.78 ($\overline{\overline{G}} \approx .824$).

Although we did not call attention to it, the derivation presented is appropriate only for parts of the domain where the grid lines of the mapped mesh ξ,η,r coincide more or less with the cylindrical coordinates x,θ,r (simplification of the expressions for the coefficients in equation (28) was permitted by setting $|\xi_x| = 1$, $\xi_\theta = 0$). It is seen in figure 6 that this might be a good approximation only in regions downstream of the midchord of the wing. A more complete analysis requires further investigation of the network, such that the shearing transformation (21) is taken into account, as well as the orientation of the cells and their true aspect ratios. This can be achieved by resorting to numerical calculations of G by scanning the entire domain. The growth factor thus obtained (at the far upstream cell, close to the fuselage) was very close to the $\overline{G}$ estimated above. (For supersonic regions in the vicinity of the wing $\overline{G}$ is smaller.) The fact that the $\overline{G}$'s are virtually the same for this case should not be taken to imply that the more complete analysis can be overlooked in all cases. Two-dimensional calculations on a parabolic grid revealed that the more complete estimate was 'slower' (it was equal to the 'rough' estimate raised to the power of 1/1.4), and insisting on attaining rates of convergence predicted by a rough analysis, would have been futile.

We remark that the calculation of the aspect ratios of the cells in two-dimensional meshes is greatly simplified by utilizing the conformal properties of the transformation (no distortion), so that aspect ratios are readily calculated from the physical coordinates. This, unfortunately, does not hold for the three-dimensional networks, and calculations must be carried out in the transformed space ξ,η,r. Also, the problem of highly elongated cells in the two-dimensional grid aforementioned, was alleviated by 'redistributing' the aspect ratios within the domain (which was most easily done by introducing a suitable stretching function), resulting in better rates of convergence — both theoretically and numerically. Such a cure cannot be prescribed for the three-dimensional networks used, since the aspect ratios of the cells are already quite uniformly distributed.

Given an estimate for the rate of convergence, it is possible to estimate the computational effort required to solve the problem to the level of its truncation errors. Suppose the problem is first solved on the coarse grid G^{K-1} to within τ^{K-1} (τ designates the truncation error). Assuming that high frequency errors are not introduced by interpolation to the finest level, the initial estimate for the G^K problem is already reduced to $O(\tau^{K-1})$. Thus, the work required to reduce them to $O(\tau^K)$ is

$$W_K = \log O(\tau^K/\tau^{K-1})/\log \overline{\overline{G}}. \qquad (31)$$

Likewise, having an initial estimate of $O(\tau^{K-2})$, the work required to solve the G^{K-1} problem (for the strategy in use) to the level of τ^{K-1} is $2 \log O(\tau^{K-1}/\tau^{K-2})/(2^d \log \overline{\overline{G}})$, since it has $1/2^d$ as many grid points as the finest grid. For the second order scheme, $\tau^k/\tau^{k-1} \sim h_K^2/h_{K-1}^2 = 1/2^2$. Therefore, the computational work for solving the problem to the level of its truncation errors, is

$$\begin{aligned}\overline{W}_K &= W_o + \dots W_k + \dots W_{K-1} + W_K \\ &= W_o + \dots \frac{2}{2^{2d}} \cdot \frac{\log(\tau^{K-2}/\tau^{K-3})}{\log \overline{\overline{G}}} \\ &\quad + \frac{2}{2^d} \cdot \frac{\log(\tau^{K-1}/\tau^{K-2})}{\log \overline{\overline{G}}} + \frac{\log(\tau^K/\tau^{K-1})}{\log \overline{\overline{G}}} \\ &\approx \frac{18}{7} \log 0.5/\log \overline{\overline{G}} \ , \end{aligned} \tag{32}$$

neglecting the work on the coarsest grid.

This, of course, should not be taken as a practical terminating criterion of the relaxation process. Even if it does not represent the state of affairs exactly, it certainly constitutes a good approximation, and one can chose a stopping criterion depending upon the requirements of accuracy. The criterion used in practice should be determined by a perceptive interpretation of experimental results. The modified program has not been devised to include a switch for terminating the computing process since our objective was to check the validity of the estimate in forming the basis for deriving a stopping criterion.

III. RESULTS

In the following we demonstrate the advantages of the MG procedure. The geometry tested is the ONERA wing M-6 mid-mounted upon a cylindrical body of radius of .25 the semispan. The wing has a 30° leading edge sweep, a taper ratio of .562 and a uniformly tapered cross-section of approximately 10% thickness ratio. A perspective view of the configuration is shown in figure 7.

Computations were carried out on a mesh of 64×16×16 cells in the X,Y,Z directions, respectively, the crudest grid (corresponding to the fifth MG level) containing just 4 cells (it is a 4×1×1 grid). The first set of results was calculated for a lifting configuration with a moderately sized supersonic zone (containing approximately 2% of the points on the grid used). The free stream Mach number was .84 and the wing and fuselage were at an angle of attack of 3.06°. The results show the effect of using different numbers of grid levels. Figure 8a displays the convergence histories of the average residuals, and indicates the beneficial effect of MG: while the convergence rate of the original scheme is .982, MG using a sequence of four grids yields a rate of .80 (more than 12 times faster than the original code). Without MG, the situation is aggravated on finer meshes (in fact, it can be predicted by expansion of G in equation (29) for low frequencies that the asymptotic convergence rate will be $1 - O(h^2)$), whereas the performance of MG is independent of the fineness of the mesh. Therefore, the relative advantage of the MG increases as the mesh gets finer. The superiority of the

MG scheme can also be seen in figures 8b and 8c, which show the convergence histories of the circulation at the root section of the wing and the number of supersonic points detected in the solution. (In these figures the continuous lines are drawn by connecting the points for the sake of clarity.)

As described in the previous section, the estimated rate of convergence is $\bar{\bar{G}} = .824$, while the experimental result is slightly better: .80. That the theoretical prediction tends to underestimate the performance of MG algorithms was also reported in reference 3. An MG code can be considered to be correctly programmed and free of bugs and flaws (which are most commonly inflicted by incorrect handling of boundary conditions), as long as the experimental rate of convergence is bounded by the predicted rate. Also, these results suggest that the choice of a fixed strategy was adequate. The rate attained in our calculations bears out Brandt's assertion that one should not settle for any convergence rate slower than that predicted by a local mode analysis.

The estimated computational work required for obtaining converged solutions which are at the level of the truncation error is $\bar{W}_K = 9.2$ work units (WU). This follows from equation (32) when inserting the theoretical value for the rate of convergence. We prefer to use this value for the rate rather than the convergence rate obtained experimentally, since this yields a more conservative criterion for stopping the relaxation process (letting $\bar{\bar{G}} = .80$ yields $\bar{W}_K = 8$ WU). This is 'safer' since the experimental $\bar{G}$ may increase (hopefully still bounded by .824) when using different meshes or treating different configurations.

Surprisingly, the solution obtained by five grid levels converges faster than the MG that employs just four grids, although the rate of convergence of the last is marginally higher. The solutions obtained in both cases are identical, even though on the fifth grid the no flux condition on the wing is extended to the region outboard of the wing tip to the lateral boundary. Referring to figure 6, this region lies between the dashed line representing the singular line of the conformal map (in equation (20)), and the dashed line leaving the trailing edge of the wing. It was reported in several publications that the coarsest possible MG level has a negligible effect in improving the performance of MG that uses a sequence of levels excluding the coarsest. To explore the difference between MG employing five grids and MG that uses four grids, we list the error in the solutions after 9.2 WU as compared to their converged values (refer to figures 8b, 8c):

	circulation	number of supersonic points
4 grids	2%	2.1%
5 grids	1%	1.6%

It appears that the four-grid MG will require about three more work units to achieve the value of the circulation obtained by the five-grid MG after 9.2 WU. The relative advantage of the scheme employing all possible coarse levels decreases as we pose requirements for higher accuracy. For example, if this were the case, we would have to continue the relaxation process up to 15.1 WU for the five-grid MG or 16.4 WU when using just four grids, in order to guarantee a converged circulation to within 0.07% and to capture all supersonic points. Note that it is only after 16 iterations that the original code starts to detect any supersonic points at all.

In figure 9 the streamwise pressure distributions are plotted for the .3, .6 and .9 semispan stations. Comparison is made between a well converged solution and the solution obtained by the four level MG after 8.7 WU. Notice the typical pattern of the leading edge shock and the trailing edge shock (which is smeared out because of the poor resolution on the aft portion of the wing caused by the parabolic nature of the mapping) that merge as we proceed outboard. The differences in the pressures are rather small and are limited to the region between the shocks. The discrepancies seem to be larger as we approach the tip. After 8.7 WU the lift, drag and moment coefficients of the wing were converged to within 2.6%, 5.0% and 2.5%, respectively. A fifth coarse level could probably provide a better converged solution.

Next, we consider a uniform free stream at a Mach number of .923 and zero angle of attack, resulting in a non-lifting flow for this symmetrical geometry. A well converged solution indicates that the flow velocity is supersonic at 6.7% of the mesh points. Convergence histories of the average residuals and the number of supersonic points are displayed in figure 10, in which comparison is made between the four-grid MG scheme and the relaxation scheme without the MG. The rates of convergence of both cases are almost identical to the corresponding rates attained for the lifting case. At the estimated computational effort required for convergence (9.2 WU), 96% of the total number of supersonic points were established. After 16.4 WU, the number of supersonic points had converged to within .6% of the total number.

The streamwise pressure distributions are presented in figure 11 for the .3, .6 and .9 semispan locations. As in the lifting case, we show the deviation of the pressure distributions achieved by the four-level MG scheme after 8.7 WU from those of a well converged solution. Differences are minor in the vicinity of the fuselage and they become more prominent as we proceed outboard. Unlike in the lifting case, the overall drag coefficient of the wing was well converged at the end of 8.7 WU having an error of 1.45% vs. 5.0% for the lifting case.

IV. CONCLUSIONS

The MG technique has been incorporated into an existing computer program that calculates the transonic potential flow past wing-fuselage combinations. The program uses a conventional SLOR/rotated-difference smoothing algorithm to calculate mixed elliptic-hyperbolic flowfields that contain discontinuities. The computational effort when solving on a rather coarse grid (64×16×16) is reduced by an order of magnitude for a given accuracy. The merit of the method becomes more prominent when calculating on finer meshes which are of engineering interest. The rates of convergence of the MG algorithm are in remarkably good agreement with theoretical estimates from a local mode analysis, even on the curved and highly stretched mesh of the present computations. We stress that it is of great importance to find the maximum growth factor by such an analysis in the early stages of developing a MG code. Although the expression for the growth factor may be quite complicated, it is worthwhile to extract from it as much information as possible so as to be aware of what might be expected from the program, and also for systematically optimizing the mesh.

The modified MG program should be subject to further study of other practical configurations. Also, the MG technique may be utilized for increasing the accuracy in various ways. For example, by sequential refinement which can be employed in regions of the flowfield where high derivatives of flow properties are likely to occur (as in the vicinity of shocks, at the leading and trailing edges of the wing and at the wing tip), or, by extrapolating the truncation errors on coarse grids (which implies the need for minor changes in the forcing term in equation (23)).

V. REFERENCES

1. Fedorenko, R. P.: The Speed of Convergence of One Iterative Process. USSR Comp. Math. and Math. Phys., Vol. 4, 1964, pp. 227-235.

2. Backvalov, N. S.: On the Convergence of a Relaxation Method with Natural Constraints on the Elliptic Operator. USSR Comp. Math. and Math. Phys., Vol. 6, 1966, pp. 101-135.

3. Brandt, Achi: Multi-Level Adaptive Solution to Boundary Value Problems. Math. Comp., Vol. 31, 1977, pp. 333-391.

4. Brandt, Achi: Multi-Level Adaptive Techniques for PDE's. Math. Software III - Proceedings of Symposium, Math. Research Center, University of Wisconsin, Vol. 39, 1977, pp. 277-318.

5. Brandt, Achi; Dinar, Natan: Multigrid Solutions to Elliptic Flow Problems. Numerical Methods for PDE's, Proceedings of Symposium, Math. Research Center, University of Wisconsin, Vol. 42, 1978, pp. 53-149.

6. South, Jerry C.; Brandt, Achi: Application of a Multi-Level Grid Method to Transonic Flow Calculations. Transonic Flow Problems in Turbomachinery, edited by T. C. Adamson and M. F. Platzer, Hemisphere, Washington, 1977, pp. 180-207.

7. Jameson, Antony: A Multi-Grid Scheme for Transonic Potential Calculations on Arbitrary Grids. Proc. of AIAA 4th Comp. Fluid Dynamics Conference, Williamsburg, VA, July 23-25, 1979, pp. 122-146.

8. McCarthy, D. R.; Reyhner, T. A.: A Multi-Grid Code for Three-Dimensional Transonic Potential Flow About Axisymmetric Inlets at Angle of Attack.

Proc. of AIAA 13th Fluid and Plasma Dynamics Conference, AIAA-80-1365, Snowmass, Colorado, July 14-16, 1980.

9. Jameson, Antony; Caughey, David A.: A Finite-Volume Method for Transonic Potential Flow Calculations. Proc. of AIAA 3rd Comp. Fluid Dynamics Conference, Albuquerque, NM, June 27-29, 1977, pp. 35-54.

10. Caughey, David A.; Jameson, Antony: Numerical Calculation of Transonic Potential Flow about Wing-Body Combinations. AIAA Journal, Vol. 17, Feb. 1979, pp. 175-181.

11. Caughey, David A.; Jameson, Antony: Progress in Finite-Volume Calculations for Wing-Fuselage Combinations. AIAA Journal, Vol. 18, Nov. 1980, pp. 1281-1288.

12. Jameson, Antony: Iterative Solution of Transonic Flows over Airfoils and Wings Including Flows at Mach 1. Comm. Pure and Appl. Math., Vol. 27, 1974, pp. 283-309.

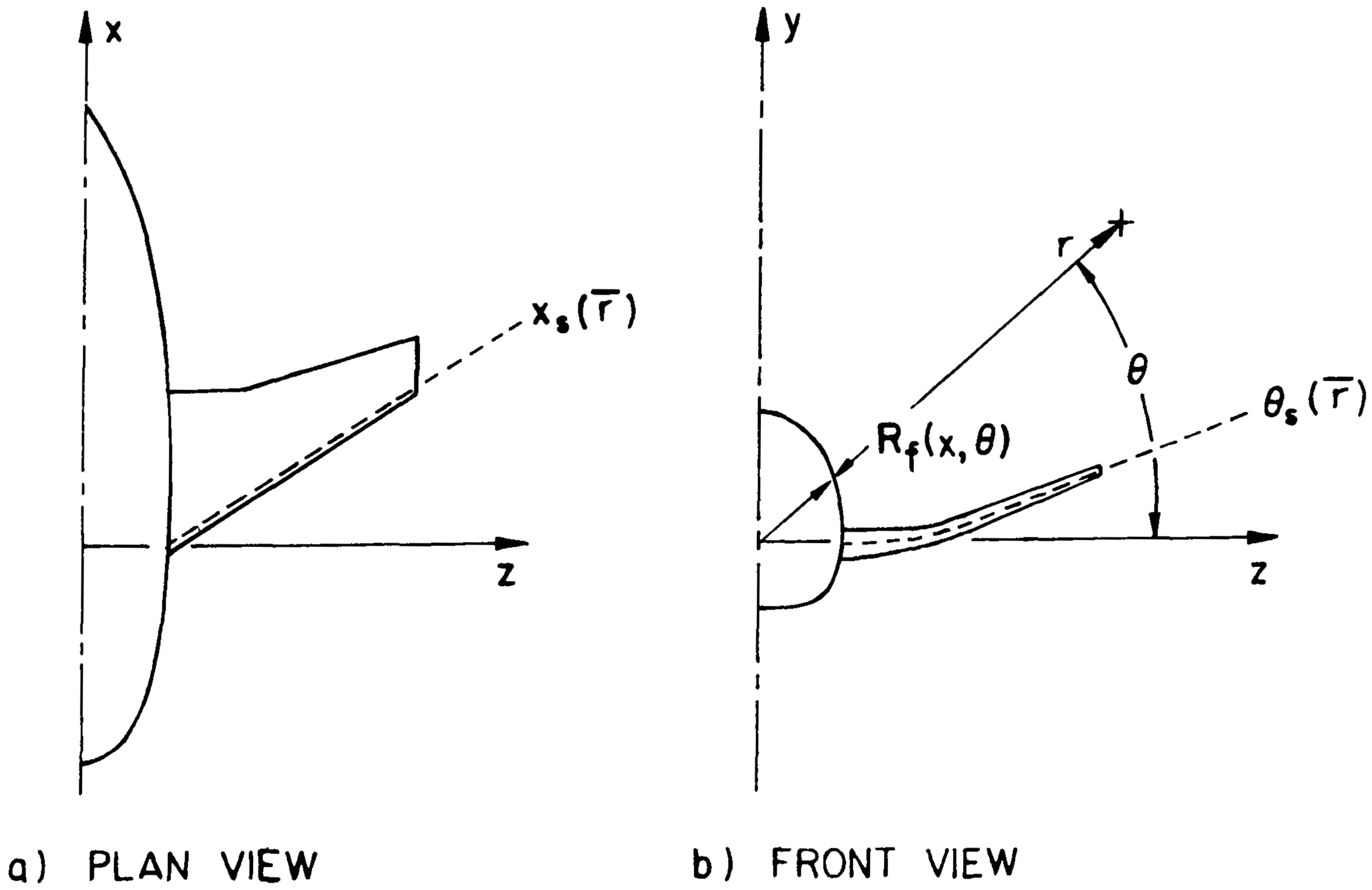


Figure 1. - Wing-fuselage geometry.

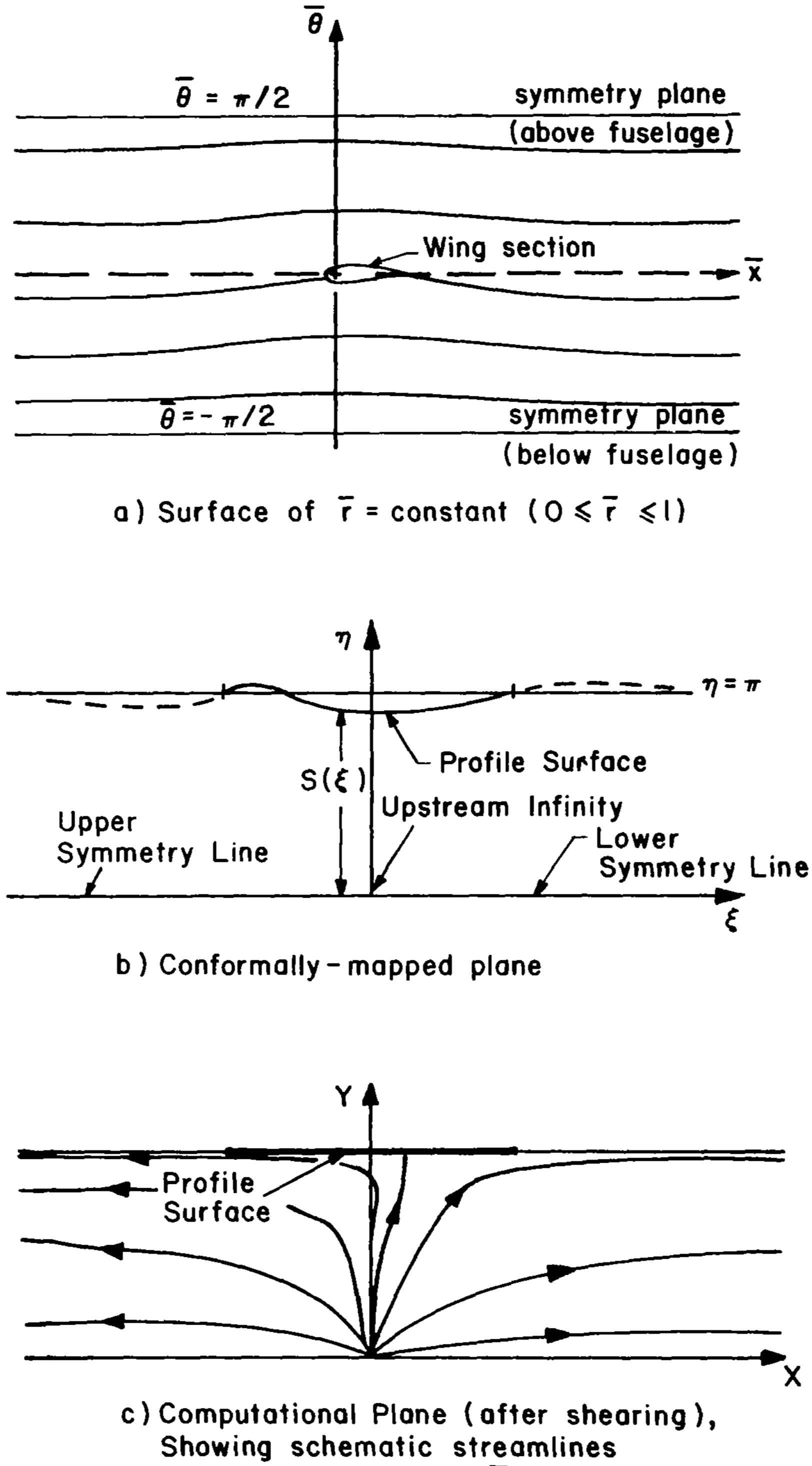


Figure 2. - Nearly-conformal mapping of $\bar{r}$ = constant surfaces ($\bar{r} \leq 1$).

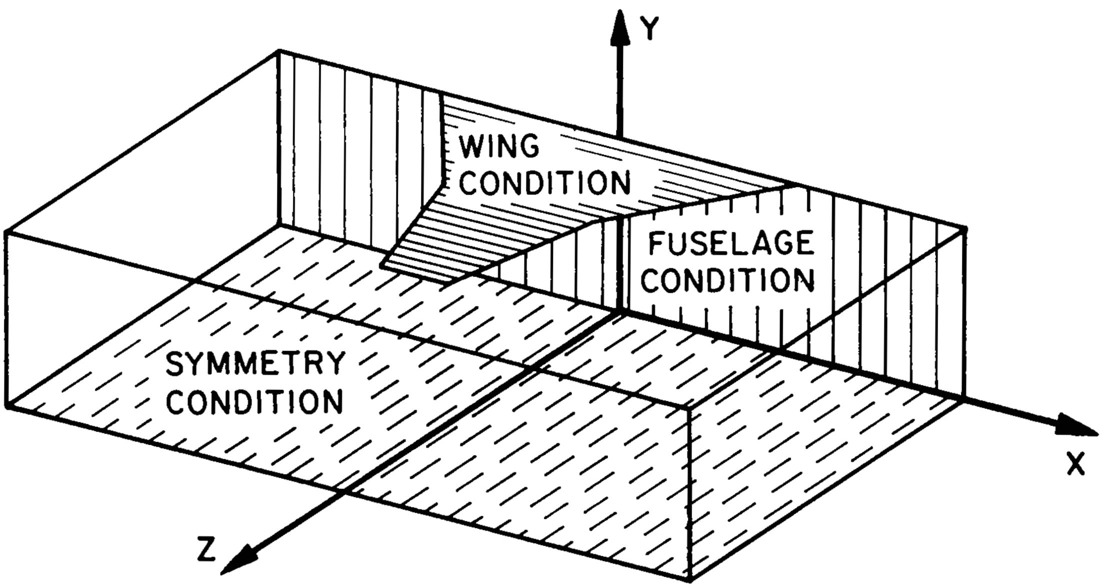
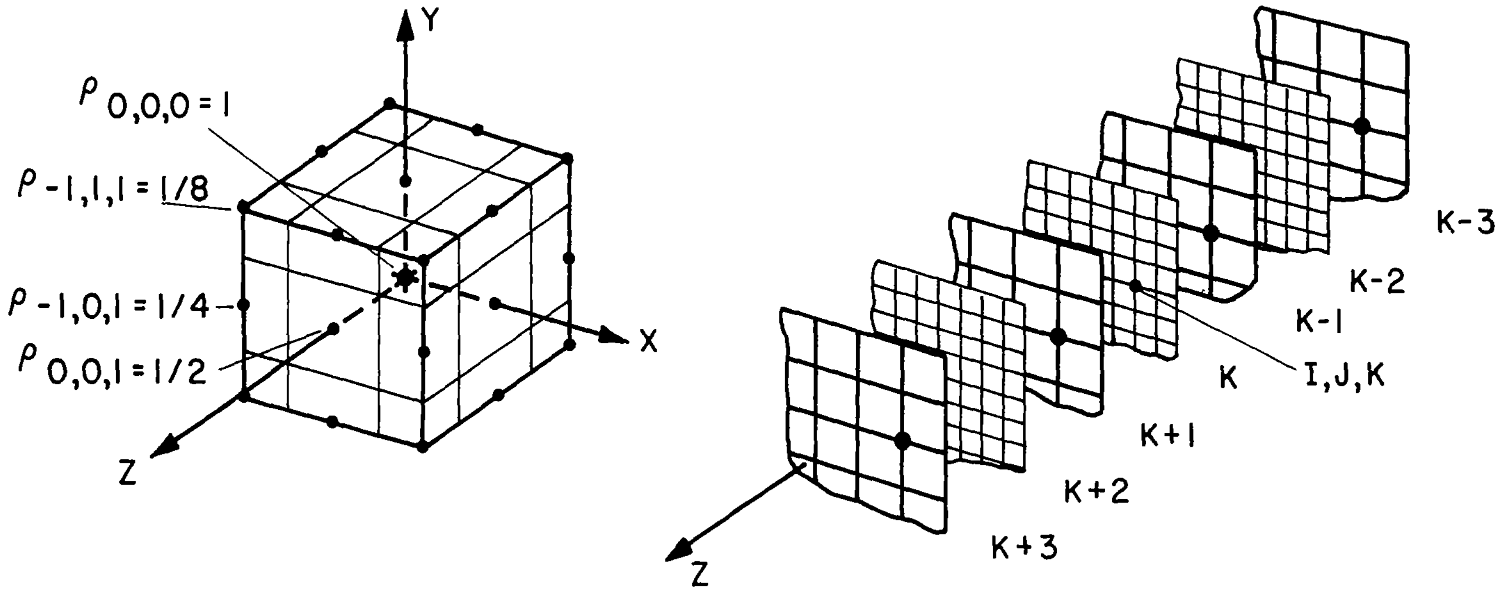


Figure 3. - Sketch of computational domain.

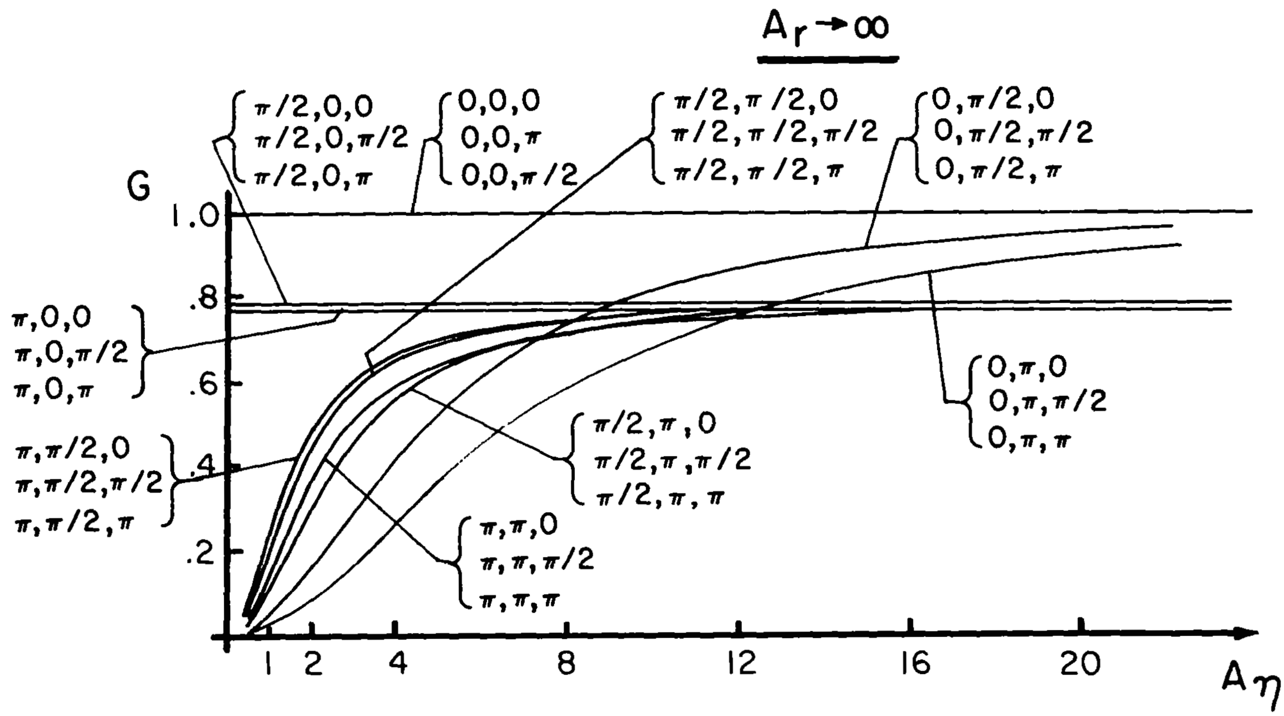


a) RESIDUAL TRANSFER

$$I_{\text{FINE}}^{\text{COARSE}} = \sum_{\alpha,\beta,\gamma} \rho_{\alpha,\beta,\gamma} \cdot \text{RES}_{(x+\alpha\cdot\Delta x,\, y+\beta\cdot\Delta y,\, z+\gamma\cdot\Delta z)}$$

b) SCHEMATIC REPRESENTATION OF PARTICIPATING COARSE GRID POINTS IN FORMING AN IMPROVED SOLUTION AT I,J,K.

Figure 4. - Schematic representation of residual weighting and correction interpolation.



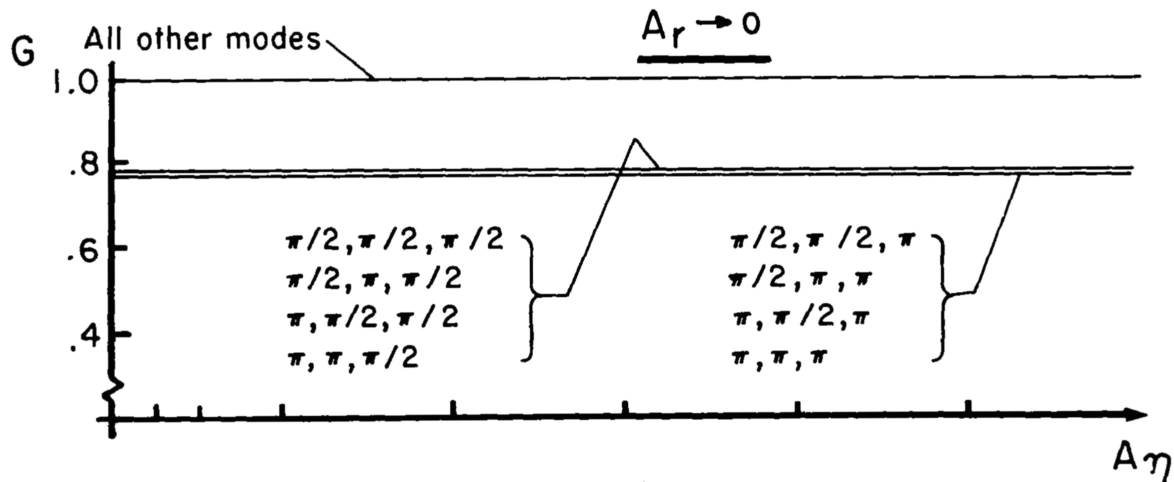


Figure 5. - Growth factor for modes of 0, π/2, π, for A_r = 0,∞, for XSLOR scheme. (M_∞ = .8, α = 0°, r = 1, ω = 1.6).

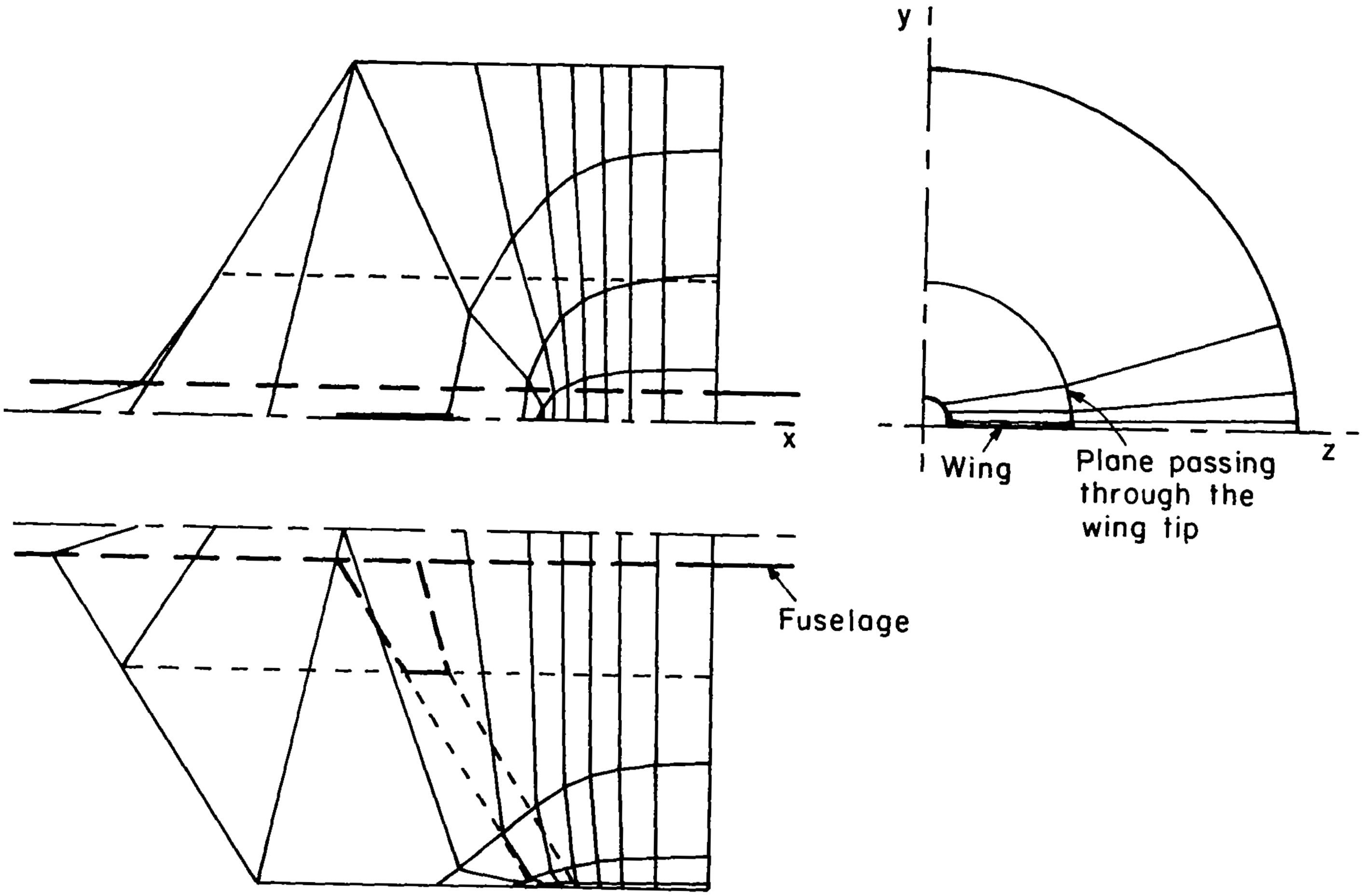


Figure 6. - Silhouettes of numerical network.

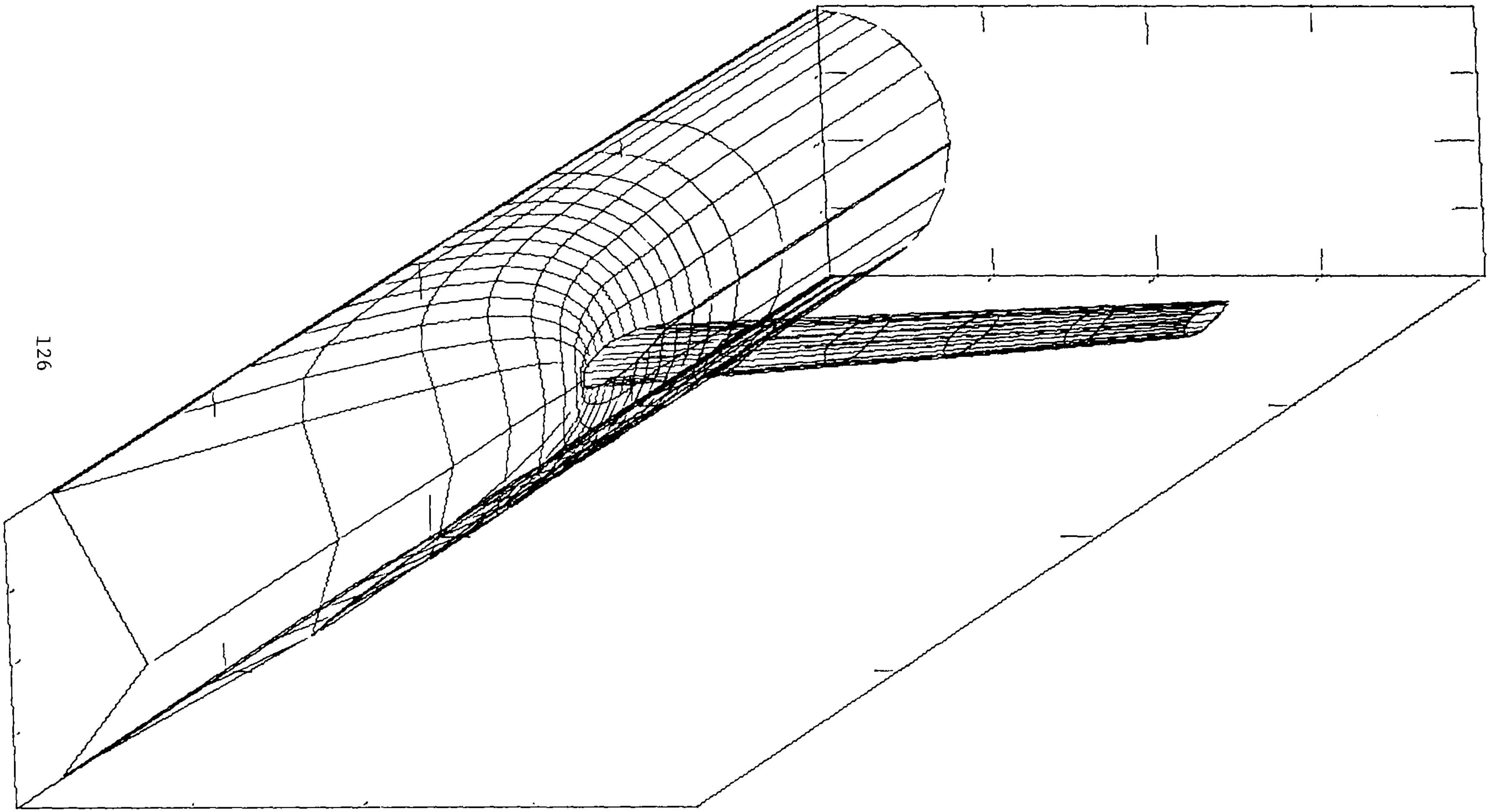


Figure 7. - Perspective view of ONERA wing M-6 mid-mounted upon cylindrical fuselage.

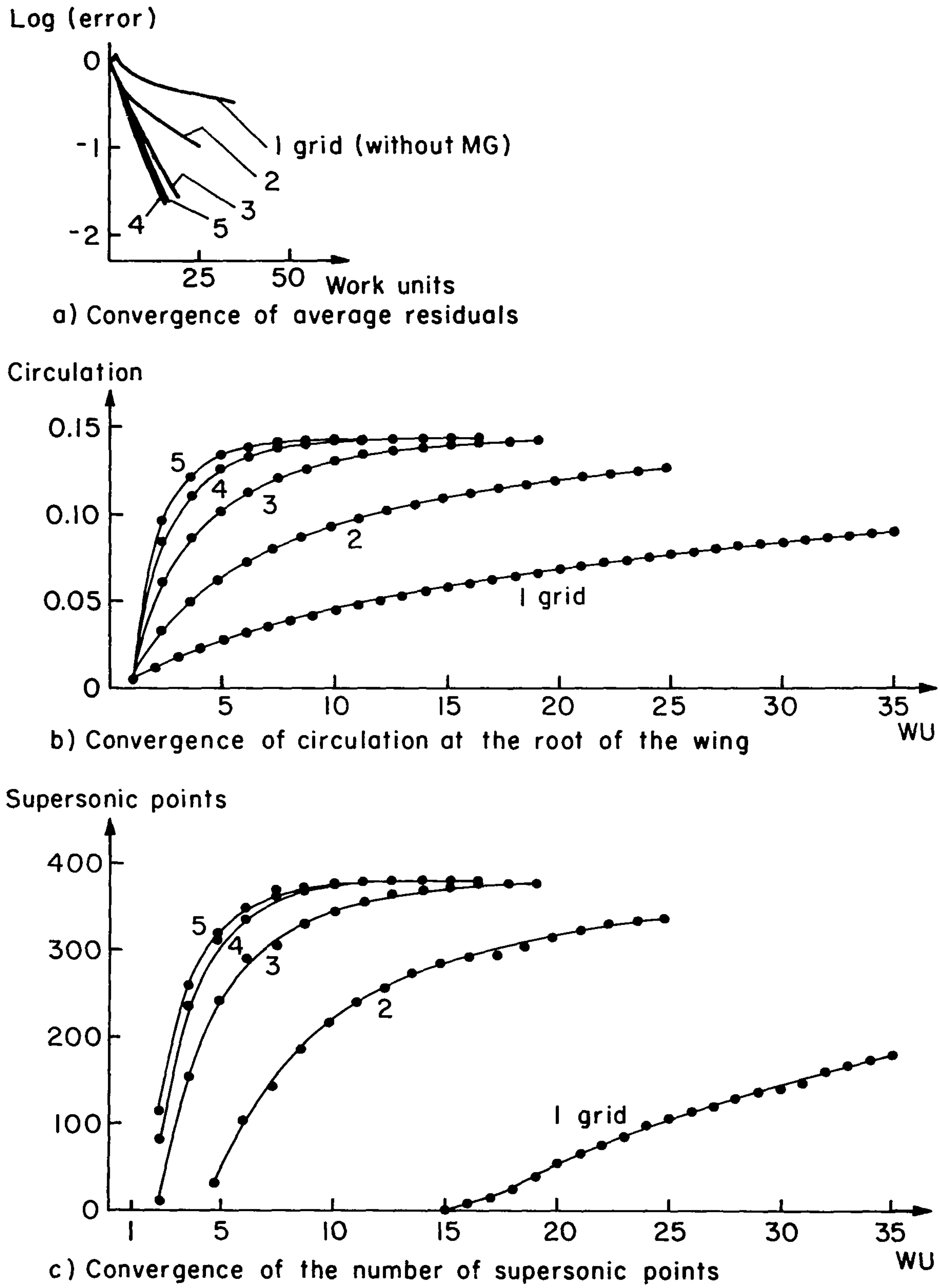


Figure 8. - Convergence histories of average residuals, circulation at root section of wing and number of supersonic points, for M_∞ = .84 and 3.06° angle of attack.

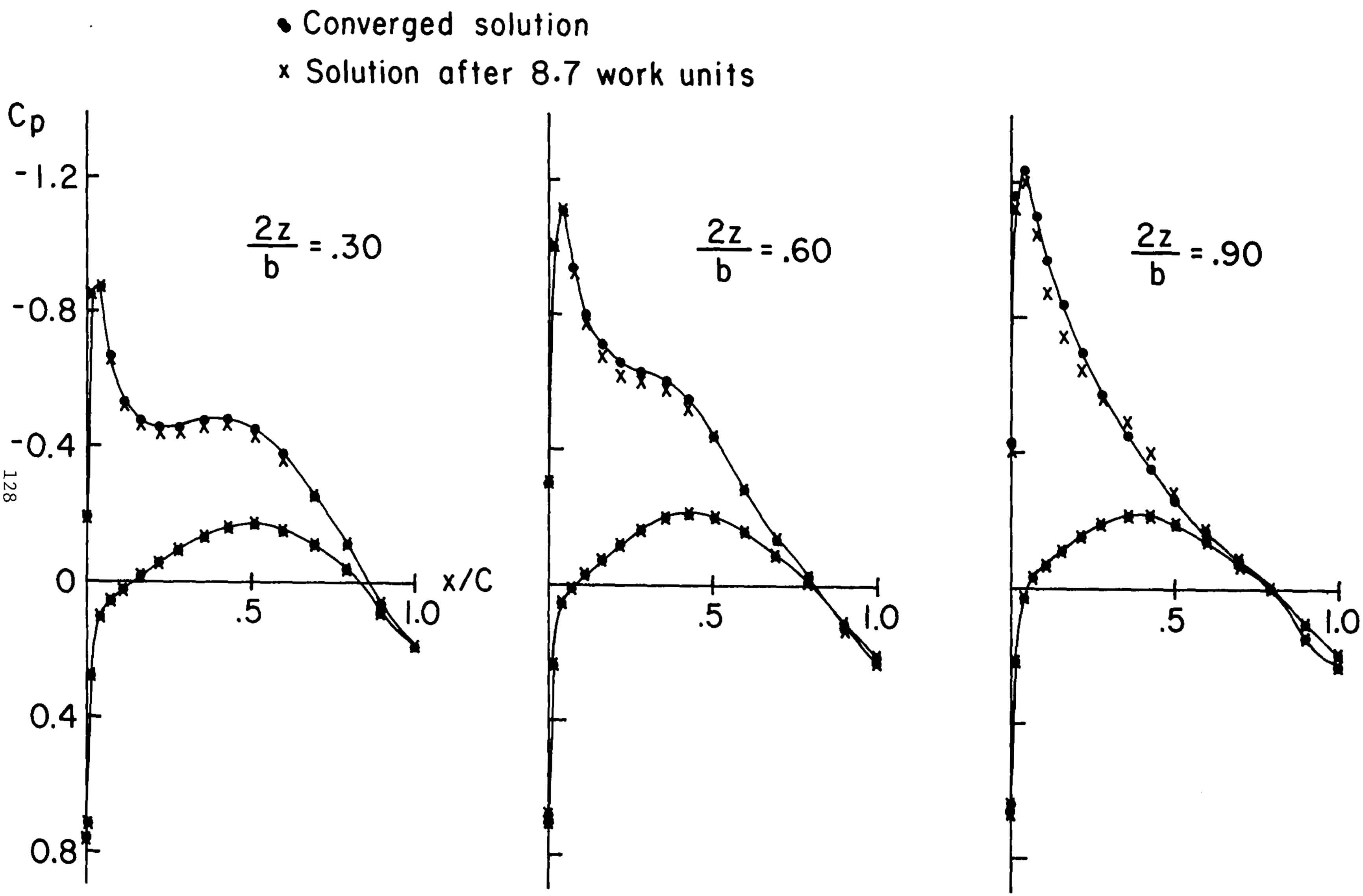


Figure 9. - Streamwise wing surface pressure distributions for M_∞ = .84 and 3.06° angle of attack.

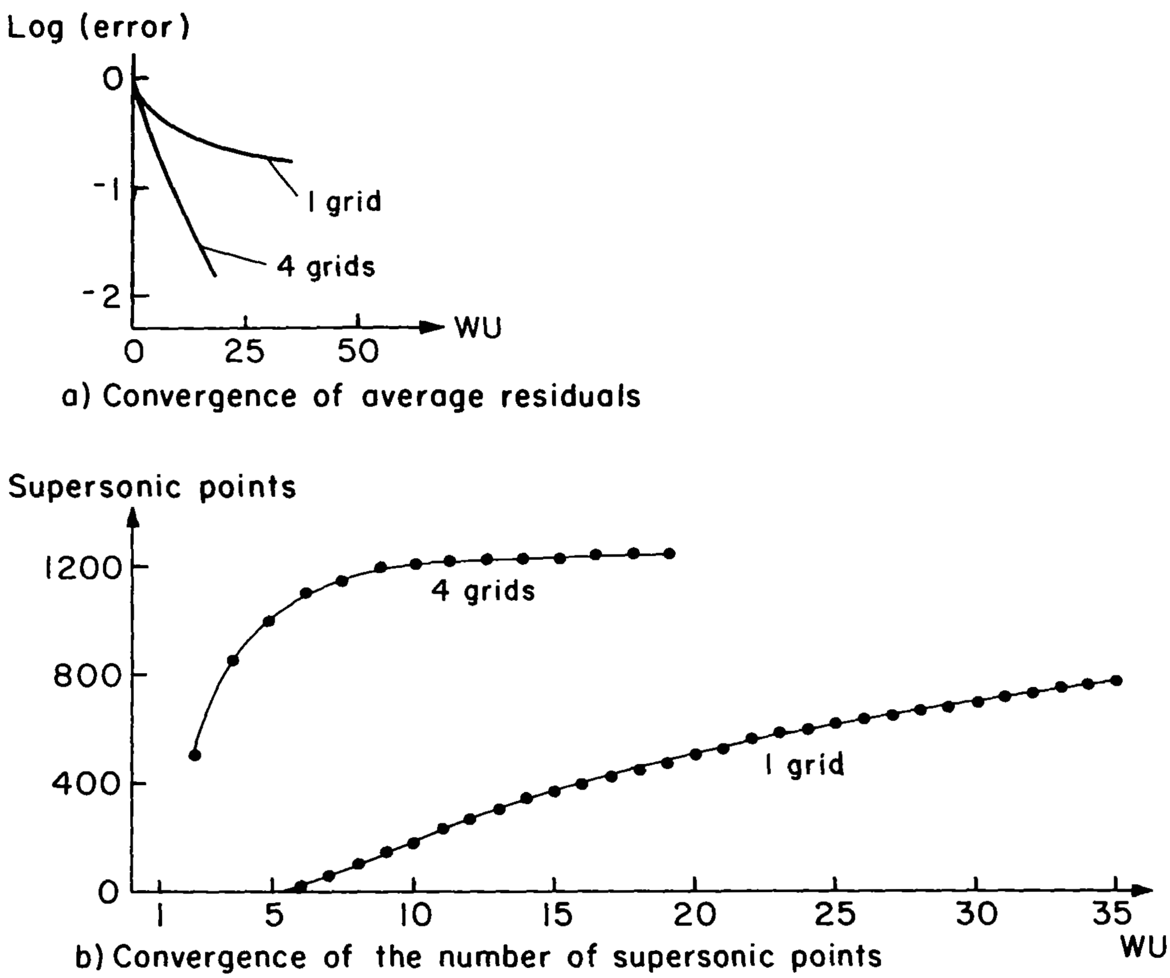


Figure 10. - Convergence histories of average residuals and number of supersonic points for M_∞ = .923 and zero angle of attack.

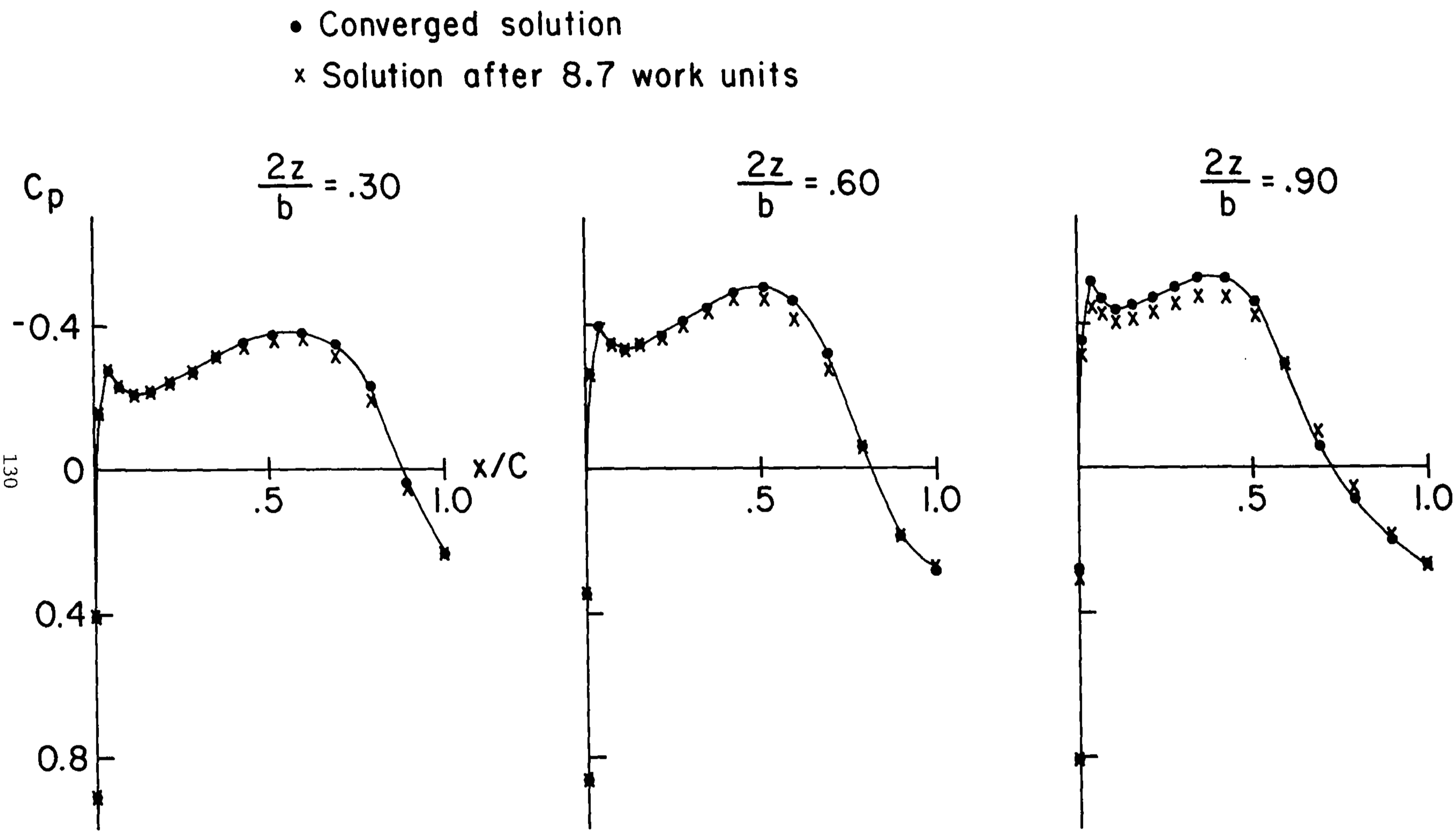


Figure 11. - Streamwise wing surface pressure distributions for $M_\infty = .923$ and zero angle of attack.

A MULTIGRID MESH-EMBEDDING TECHNIQUE

for

Three-Dimensional Transonic Potential Flow Analysis

By: Jeffrey J. Brown

Boeing Commercial Airplane Company

ABSTRACT

A method for obtaining the fine detail of a transonic flowfield is presented. The technique employs the multigrid method to embed very dense meshes in regions of interest. Accurate results are obtained on meshes of a heretofore unobtainable density with reasonable computer expenditures. Comparisons of results with data reveal accurate predictions in the supersonic bubble of a transonic inlet, an area which is incorrectly predicted by existing techniques. More accurate results are also obtained with the new method on a mesh of a density comparable to existing codes and at a lower cost.

NOMENCLATURE

English

a	speed of sound
F	forcing function
G	grid level identifier
I	interpolation operator
L	differential operator
q	freestream velocity
r	radial ccordinate
z	axial coordinate

Greek

γ	ratio of specific heats
θ	circumferential coordinate
ϕ	potential function

Superscript

k	grid level

Subscript

k	grid level
∞	freestream condition

INTRODUCTION

The standard relaxation methods used in large-scale fluid-dynamics computations are local by nature. The current solution at each grid point is influenced solely by information from neighboring points. Consequently, the global rate of information transmission is asymptotically slow and is aggravated the more dense a mesh becomes. The result of this situation is an inability to economically predict the fine details of a flowfield (i.e., peak Mach numbers, shock locations, etc.). Indeed, the computing time required to obtain the fine details of a flowfield seriously limits the usefulness of many realistic production codes. This is especially true for design applications which are, by nature, iterative processes. Consequently, methods for increasing the efficiency of the relaxation process have received much attention. While gains have been made, success has seldom been dramatic, often relying upon highly problem-dependent assumptions.

Recently, however, new mathematical techniques, referred to as multigrid methods, have been proposed by Brandt (references 1 and 2). These methods theoretically offer from one to several orders of magnitude improvement in execution time and provide greatly improved accuracy as well. Brandt has demonstrated remarkable success with two-dimensional elliptic problems of generally academic interest. The applicabilty of the multigrid method to transonic fluid dynamics computations (a mixed hyperbolic-elliptic problem) was demonstrated by McCarthy and Reyhner (reference 3). They incorporated the multigrid procedure into the Reyhner code for three-dimensional transonic potential flow around axisymmetric inlets at angle of attack (reference 4).

The McCarthy-Reyhner code is a finite-difference, non-conservative, successive-line-over-relaxation (SLOR) scheme which operates in cylindrical coordinates in the physical domain. It uses a hierarchy of four mesh densities, the finest of which (level 4) is roughly twice as dense as the finest practical mesh available with the Reyhner code. Very accurate results are obtained with the McCarthy-Reyhner code on level 4 in approximately three minutes CYBER-175 central processor (CP) time. As a comparison, it has been estimated that it would require six hours CP time to achieve similar results with the Reyhner code (modified for level 4). As dramatic as these results are, experience with the McCarthy-Reyhner code has indicated that there are regions of a flowfield (e.g., the highlight region in an engine inlet) where level 4 is not sufficiently fine to accurately determine the details of the flowfield. It would not be practical, from a computer storage requirement, to extend the McCarthy-Reyhner code to level 5 (twice as dense as level 4) to attempt to resolve the fine detail of the flow.

The present work describes an investigation of a technique for embedding very dense meshes in regions of interest. The approach involves modifying the McCarthy-Reyhner code to embed a series of dense meshes in the region of an inlet highlight. Global information is transmitted to the embedded meshes from the coarser meshes via the multigrid procedure. Likewise, the fine detail of the flowfield is conveyed to the coarser meshes during the multigrid process. Hence, the solution on each grid (embedded and full domain) is corrected by information transmitted by the multigrid process from the other grid.

The author gratefully acknowledges the work of Gary E. Shurtleff of Boeing Computer Services, Inc., who performed the computer coding in a timely and efficient manner.

MULTIGRID ALGORITHM

A brief description of the general multigrid method is presented for completeness and to introduce terminology. After the discussion of the general procedure, the mesh-embedding philosophy is discussed. Reference 1 should be consulted for a detailed description of the multigrid method.

The objective is to solve the potential flow equation

$$L\left[\phi(r,\theta,z)\right] = F(r,\theta,z) \tag{1}$$

where L is a differential operator defined as

$$\begin{aligned} L(\phi) = (a^2 - \phi_r^2)\,\phi_{rr} + \left(a^2 - \frac{\phi_\theta^2}{r^2}\right)\frac{\phi_{\theta\theta}}{r^2} + (a^2 - \phi_z^2)\,\phi_{zz} \\ -2\,\frac{\phi_r\phi_\theta}{r^2}\,\phi_{r\theta} - 2\,\phi_r\phi_z\phi_{rz} - 2\,\frac{\phi_\theta\phi_z}{r^2}\,\phi_{\theta z} + \left(a^2 + \frac{\phi_\theta^2}{r^2}\right)\frac{\phi_r}{r} \end{aligned} \tag{2}$$

where

$$a^2 = a_\infty^2 - \frac{\gamma-1}{2}\left(\phi_r^2 + \frac{\phi_\theta^2}{r^2} + \phi_z^2 - q_\infty^2\right)$$

and

$$F(r,\theta,z) = 0 \quad .$$

The (finite difference) multigrid method replaces equation (1) with a collection of finite difference equations

$$L^k \Phi^k = F^k \tag{3}$$

In equation (3), L^k represents a discretized version of the operator L, and Φ^k and F^k represent scalar fields defined on a grid G^k which is one of a hierarchy of grids of varying coarseness. (Φ^k is the exact solution to the discretized equation.)

An economical approximate soluton ϕ^k to equation (3) can be obtained by interpolation from a coarser grid G^{k-1}. Grid G^{k-1} is obtained from grid G^k by deleting every other grid line from grid G^k. On grid G^{k-1}, the discretized equation is

$$L^{k-1} \Phi^{k-1} = F^{k-1} \tag{4}$$

When the approximate solution ϕ^{k-1} to equation (4) is obtained, it can be interpolated to grid G^k as follows

$$\phi^k = I^k_{k-1} \phi^{k-1} \tag{5}$$

where I^k_{k-1} is an interpolation operator from grid G^{k-1} to grid G^k. This procedure has been used by several authors (e.g., reference 5) to obtain a solution to equation (3) as a sequence of solutions on coarser meshes (i.e., G^k, G^{k-1}, G^{k-2}, etc). The next natural step is to ask whether one can exploit the proximity between the G^k and G^{k-1} problems not only in generating a good first approximation on G^k, but also in the process of improving the first approximation. This can be done and is the crux of the multigrid philosophy.

By taking this essential step, the errors on grid G^k can be smoothed inexpensively and efficiently on grid G^{k-1}. At any point in the solution process on grid G^k, one has the approximate solution to equation (3), ϕ^k. One can formally define the error, Ψ^k, on grid G^k as:

$$\Phi^k = \phi^k + \Psi^k \tag{6}$$

where Φ^k is the exact solution to equation (3). After several relaxation sweeps on grid G^k, the error Ψ^k is smooth. Hence, a good approximation of Ψ^{k-1} can inexpensively be computed on the coarser grid G^{k-1}. For this purpose, the fine grid equation

$$L^k(\phi^k+\Psi^k) - L^k\phi^k = F^k - L^k\phi^k = r^k \tag{7}$$

is approximated by the coarse grid equation

$$L^{k-1}\left(I_k^{k-1}\phi^k + \Psi^{k-1}\right) - L^{k-1} I_k^{k-1}\phi^k = \bar{I}_k^{k-1} r^k \tag{8}$$

where $\bar{I}_k^{k-1}$ need not be the same as I_k^{k-1} .

By defining

$$\bar{\phi}^{k-1} = I_k^{k-1}\phi^k + \Psi^{k-1}$$

equation (8) can be rewritten

$$L^{k-1}\bar{\phi}^{k-1} = \bar{I}_k^{k-1} r^k + L^{k-1} I_k^{k-1}\phi^k = f^k \tag{9}$$

The new unknown $\bar{\phi}^{k-1}$ represents, on the coarse grid, the sum of the basic approximation ϕ^k and its correction error Ψ^k.

When the approximate solution $\bar{\phi}^{k-1}$ to equation (9) is obtained, it can be employed to correct the approximation on the fine grid as follows

$$\phi^k_{NEW} = \phi^k_{OLD} + I^k_{k-1}\left(\bar{\phi}^{k-1} - I_k^{k-1}\phi^k_{OLD}\right) \tag{10}$$

When this procedure is extended over several grids (four in the McCarthy-Reyhner code) it yields accurate solutions in the equivalent work of only a few sweeps of the finest level. This is because the global errors are smoothed efficiently and inexpensively on the coarse mesh.

MESH EMBEDDING

In regions of high gradient, a dense mesh is required to resolve the fine details of the flowfield. Away from the regions of high gradient, the dense mesh is not needed. Extending the dense mesh over the entire domain is actually counterproductive (particularly without multigrid) because of the extremely slow rate of convergence on a dense mesh.

Fine flow detail can be obtained by embedding a dense mesh in regions of interest. This is easily implemented in a multigrid procedure by merely redefining the domain of the finest level to be a subset of the overall computational domain (cf., Figure 1). The computations on the embedded mesh are then restricted to the embedded domain. The coarser grid then serves a dual purpose in the multigrid procedure. On the subdomain where the embedded mesh is not defined, the coarse grid has the role of the finest grid and the original difference equation (equation 3) is solved in that region. At the same time, on the subdomain which is coextensive with the embedded mesh, the coarse grid serves for calculating the coarse-grid correction, equation (10).

Understanding of this process is facilitated by letting G^{k+1} denote the embedded mesh and by defining G^k_{k+1} as the set of points of grid G^k where the G^{k+1} difference equations are defined (i.e., the points of G^k which are also points of G^{k+1}, cf., Figure 2). The difference equations on grid G^k are accordingly modified as follows

$$L^k \Phi^k = \bar{F}^k \tag{11}$$

where $\bar{F}^k = F^k \quad \text{in} \quad G^k - G^k_{k+1}$

and $\bar{F}^k = F^k_{k+1} \quad \text{in} \quad G^k_{k+1}$

where $F^k_{k+1} = I^k_{k+1}\left(\bar{F}^{k+1} - L^{k+1}\phi^{k+1}\right) + L^k\left(I^k_{k+1}\phi^{k+1}\right)$

In equation (11), $\bar{F}^k$ may be regarded as the usual G^k right-hand side (F^k), corrected to achieve G^{k+1} accuracy in the G^k solution.

Figure 2 illustrates a typical embedded mesh. On the boundaries of the embedded domain (exclusive of solid boundaries), constant potential, ϕ^{k+1}, boundary conditions are imposed. The values of ϕ^{k+1} for the constant potential boundary conditions are obtained by interpolation from the next coarser grid, G^k.

RESULTS

The present study employs a hierarchy of five mesh densities. The axial and radial step sizes in the coarsest mesh, level 1, need not be uniform. Once the level 1 mesh, G^1, is defined, all finer meshes, G^k, are obtained by halving the local axial and radial step sizes from the coarser mesh, G^{k-1}. The step size in the θ -direction is held constant for all levels. Levels 1, 2, and 3 extend over the entire computational domain while levels 4 and 5 are embedded meshes. The analyses to date have employed co-extensive domains for levels 4 and 5. Table I shows the number of points in each of these five levels for a typical test case.

The current analysis studied the flowfield around the NASA TM X-2937 1.26 contraction ratio inlet (reference 6) at angle of attack. The geometry of that inlet is axisymmetric and includes a high degree of turning in the highlight region of the lip. This turning provides a difficult test case for analysis. Figure 3 illustrates the results of several analyses of the NASA inlet. This test case analyzed a freestream Mach number of 0.13, an angle of attack of 30°, and a throat Mach number of 0.48. Figure 3 is a plot of surface Mach number verses surface arc length for the windward lip region. The negative arc length depicts the external surface while the positive arclength corresponds to the internal surface. A comparison of analytical results to experimental data is illustrated.

Figure 3a illustrates the results of an analysis of the test case with the Reyhner code. The key discrepancy in the Reyhner code results is the underprediction of the Mach numbers in the supersonic bubble. It is suspected that this underprediction is due to either a lack of resolution in the lip region (i.e., the computational mesh is not adequately dense) or to viscous effects in the data which the potential flow analysis can not determine. The mesh in the Reyhner code can not be made finer for reasons discussed above. Thus, the present study was undertaken to address this question.

Figures 3b and c illustrate the results of an analysis of the same test case with the modified McCarthy-Reyhner code. Figure 3b depicts the results with an embedded level 4 mesh. The level 4 mesh is approximately four times as dense as the mesh employed in the Reyhner code analysis. Examination of Figure 3b reveals a more nearly accurate prediction of the supersonic Mach numbers. Figure 3c shows the results with an embedded level 5 mesh (level 4 is embedded as well). The overprediction of the peak Mach number

TABLE I

TYPICAL MESH DENSITIES

LEVEL	NUMBER OF Z MESH	NUMBER OF R MESH	NUMBER OF Θ MESH	TOTAL
1	15	15	5	1125
2	29	29	5	4205
3	57	57	5	16245
4 TOTAL 4 EMBEDDED	113 41	113 70	5 5	63845 14350
5 TOTAL 5 EMBEDDED	225 81	225 139	5 5	253125 56295

can be explained by considering the flow in that region as a simple Prandtl-Meyer expansion. Apparently, either viscous interaction tends to mitigate this expansion causing the lower peak Mach number in the data or the high peak Mach number was not measured. This phenomenon is not resolved by the coarse meshes of Figures 3a and 3b. An excellent prediction of the remaining supersonic Mach numbers is obtained by the level 5 results.

Figure 3 shows that the external surface Mach numbers are underpredicted by all three analyses. The magnitude of the error is constant, thus the underprediction is apparently due to viscous effects (the mesh refinements have not yielded any improvement).

Figure 4 shows the results of an analysis with level 4 extended over the entire domain compared to an embedded level 4 analysis. The case analyzed was the NASA 1.26 contraction ratio inlet at the same flight Mach number and angle of attack but with a throat Mach number of 0.64. It is apparent from this comparison that an analysis on level 4 does not need to be extended over the entire domain. Restricting attention to an embedded domain will yield comparable accuracy in, for this test case, one-half the CP time.

Table II illustrates a comparison of the computational work and level of accuracy obtained using various mesh densities for a typical analysis. The measure of convergence employed in the present study is $\Delta\phi$, the average change in potential, ϕ, from one sweep to the next. This is used because it is not convenient to obtain the actual residual, $F^k - L^k\phi^k$. This number $(\Delta\phi)$ can be misleading when comparing the accuracy obtained on different meshes. Therefore, the maximum variation in mass flow rate from the enforced mass flow rate at the compressor face is calculated and is indicated in Table II for completeness. The mass flow error across the shock wave is constant for each mesh (0.6 percent). The work unit quoted in Table II is an amount of computational work equal to one relaxation sweep over a full level 4. An interesting observation from Table II can be made in regard to the embedded level 4 solution. That solution was obtained in one-half the computational work of the Reyhner code solution (i.e., the version of the code without multigrid). When one considers the increased accuracy of the embedded level 4 solution along with the decrease in computational effort one begins to appreciate the power of the multigrid code.

TABLE II COMPUTATIONAL WORK AND ACCURACY COMPARISONS

		REYHNER CODE	McCARTHY REYHNER CODE	MODIFIED CODE – EMBEDDED LEVEL > 3
LEVEL 3	Number of Sweeps	150	13	13
	Work Units	39	10	10
	CPU Seconds (Cyber 175)	138	50	50
	$\Delta \dot{m}$	2.6%	1.6%	1.6%
LEVEL 4	Number of Sweeps		14	14
	Work Units		33	19
	CPU Seconds (Cyber 175)		180	98
	$\Delta \dot{m}$		1.07%	1.24%
LEVEL 5	Number of Sweeps			16
	Work Units			42
	CPU Seconds (Cyber 175)			235
	$\Delta \dot{m}$			0.97%

CONCLUSIONS

A method has been developed for utilizing the multigrid heirarchy of meshes to embed very dense meshes in regions of high gradient. The new method provides accurate, economical solutions to real problems of engineering interest. The embedded dense meshes yield the inviscid fine detail of the flowfield. Without multigrid, the fine detail was not economically obtainable.

REFERENCES

1. Brandt, A., "Multi-Level Adaptive Computations in Fluid Dynamics", Proceedings AIAA Computational Fluid Dynamics Conference, July 23-25, 1979, Williamsburg, VA.

2. Brandt, A., "Multiple Level Adaptive Solutions to Boundary Value Problems", Mathematics of Computation, Vol. 31, 1977, pp 333-390.

3. McCarthy, D. M. and T. A. Reyhner, "A Multi-Grid Code for Three-Dimensional Transonic Potential Flow About Axisymmetric Inlets at Angle of Attack", AIAA Journal (to be published).

4. Reyhner, T. A., "Transonic Potential Flow Around Axisymmetric Inlets and Bodies at Angle of Attack", AIAA Journal, Vol. 15, 1977, pp 1299-1306.

5. Southwell, R. V., Relaxation Methods in Theoretical Physics, Claredon Press, Oxford, 1946.

6. Albers, J. A., "Comparison of Predicted and Measured Low Speed Performance of Two 51-Centimeter Diameter Inlets at Incidence Angle" NASA TM X-2937, 1973.

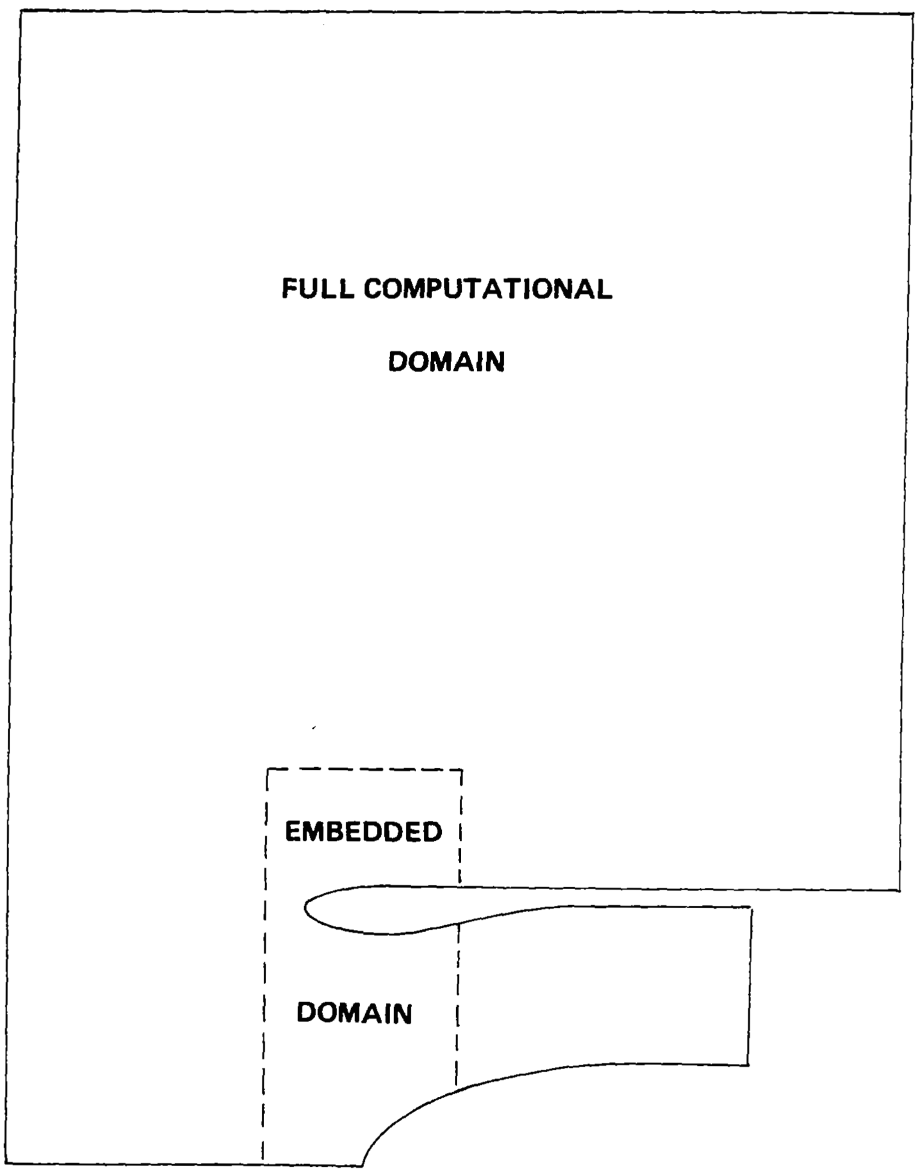


FIGURE 1. COMPUTATIONAL DOMAINS

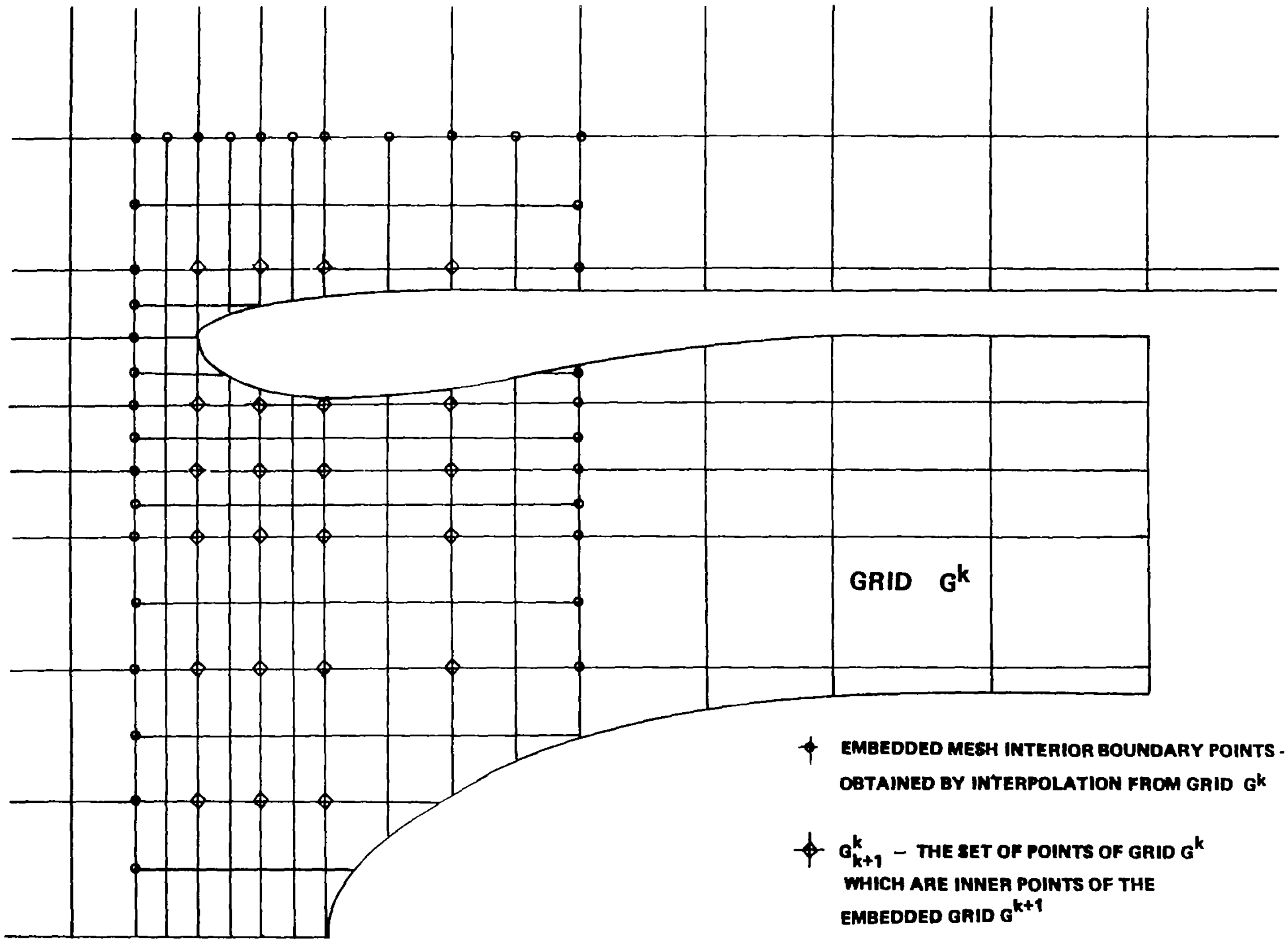


FIGURE 2. EMBEDDED GRID G^{k+1}

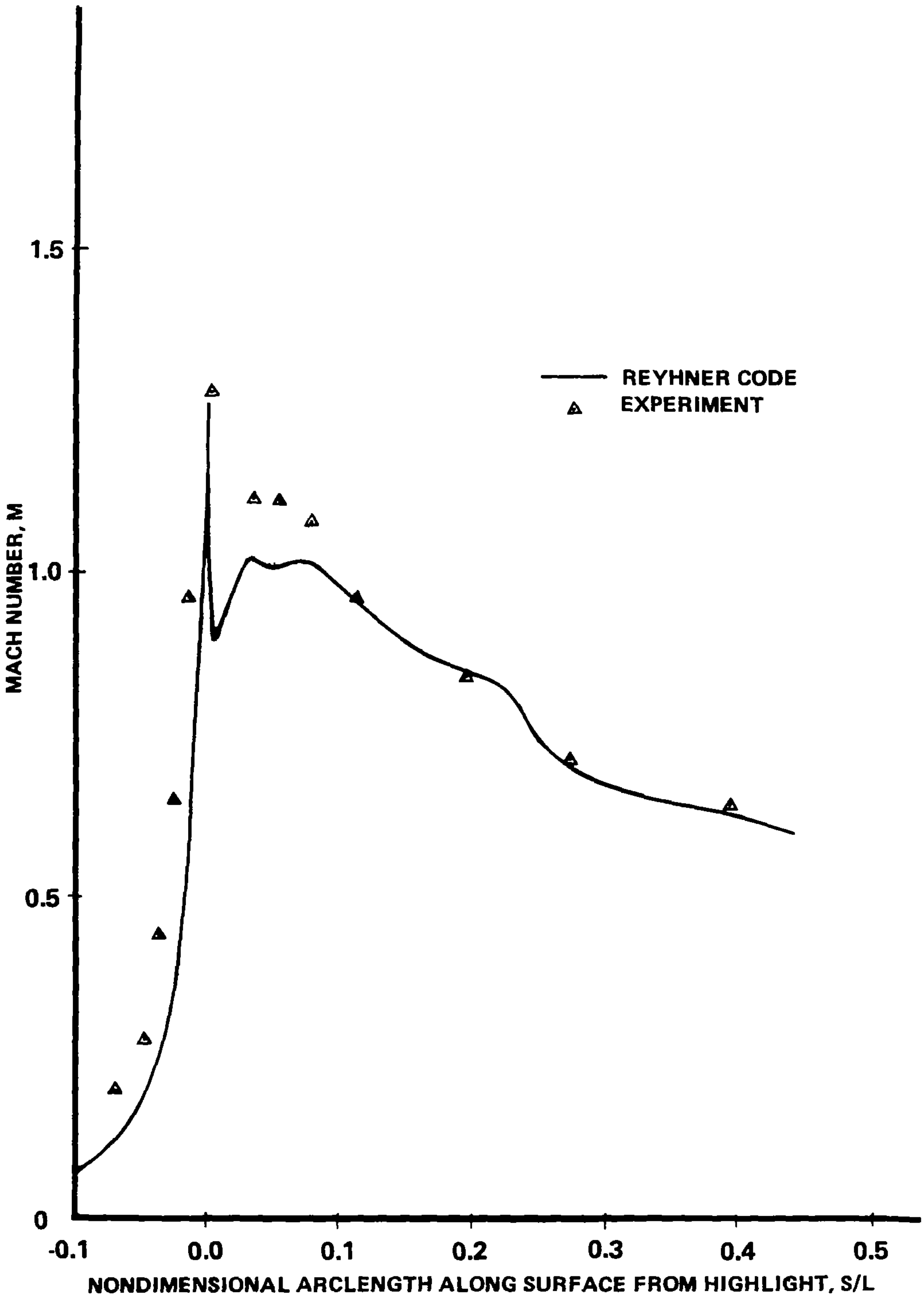


FIGURE 3A. NASA TM X - 2937 INLET WINDWARD LIP $M_{CF} = 0.452$ $M_{\infty} = 0.13$ $\alpha = 30$

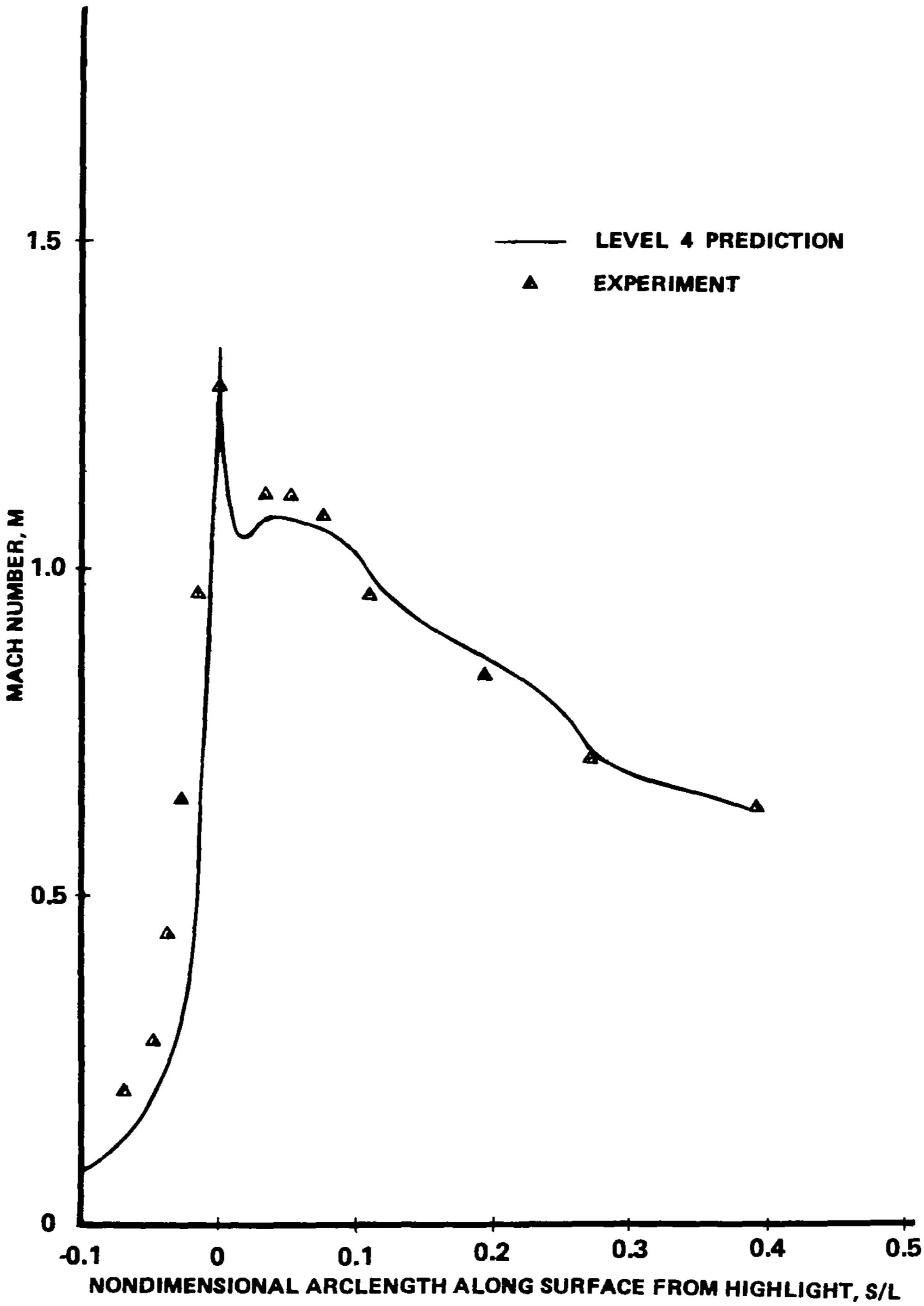


FIGURE 3B. NASA TM X - 2937 INLET WINDWARD LIP $M_{CF} = 0.452$ $M_{\infty} = 0.13$ $\alpha = 30$

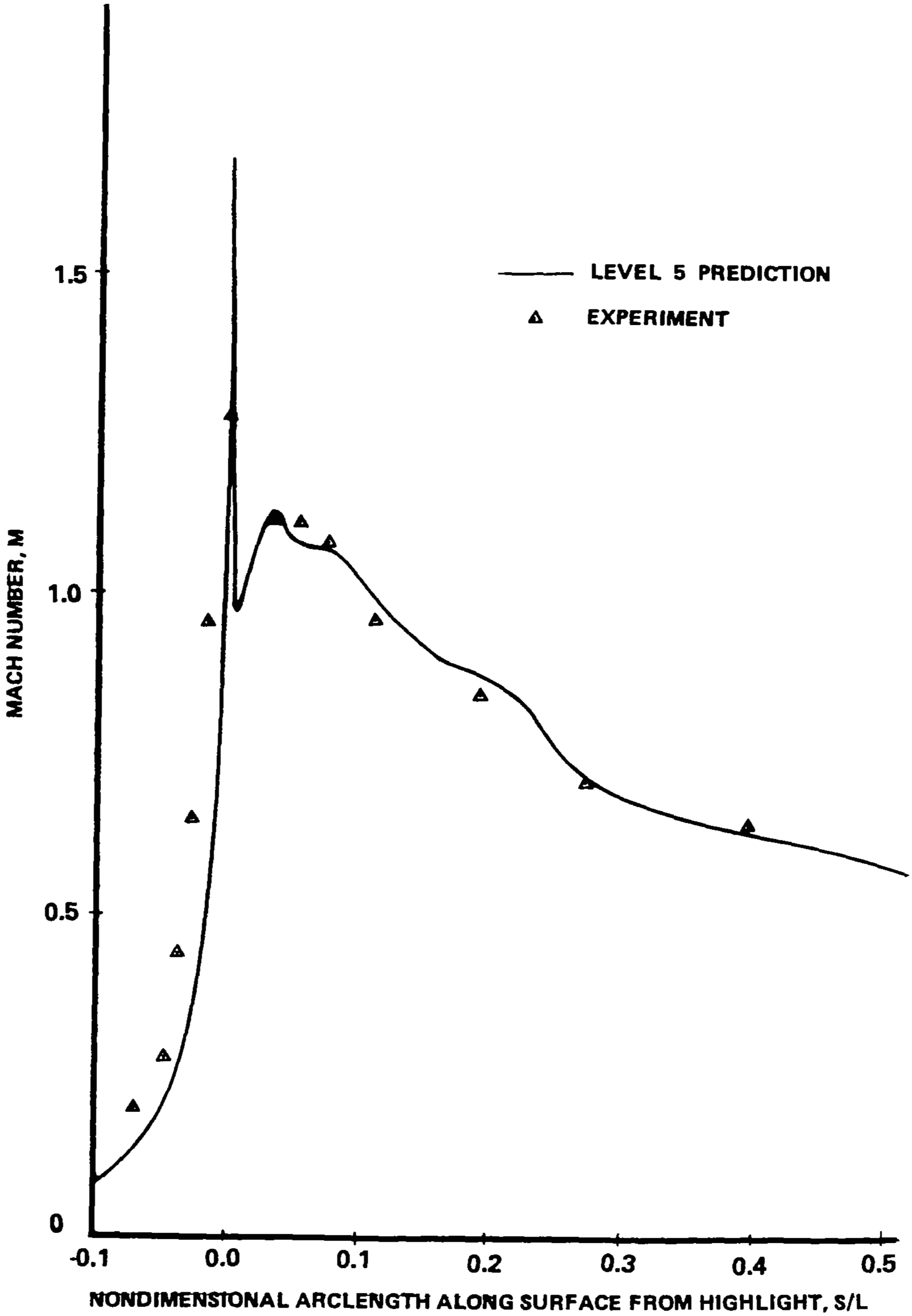


FIGURE 3C. NASA TM X - 2937 INLET WINDWARD LIP $M_{CF} = 0.452$ $M_{\infty} = 0.13$ $\alpha = 30$

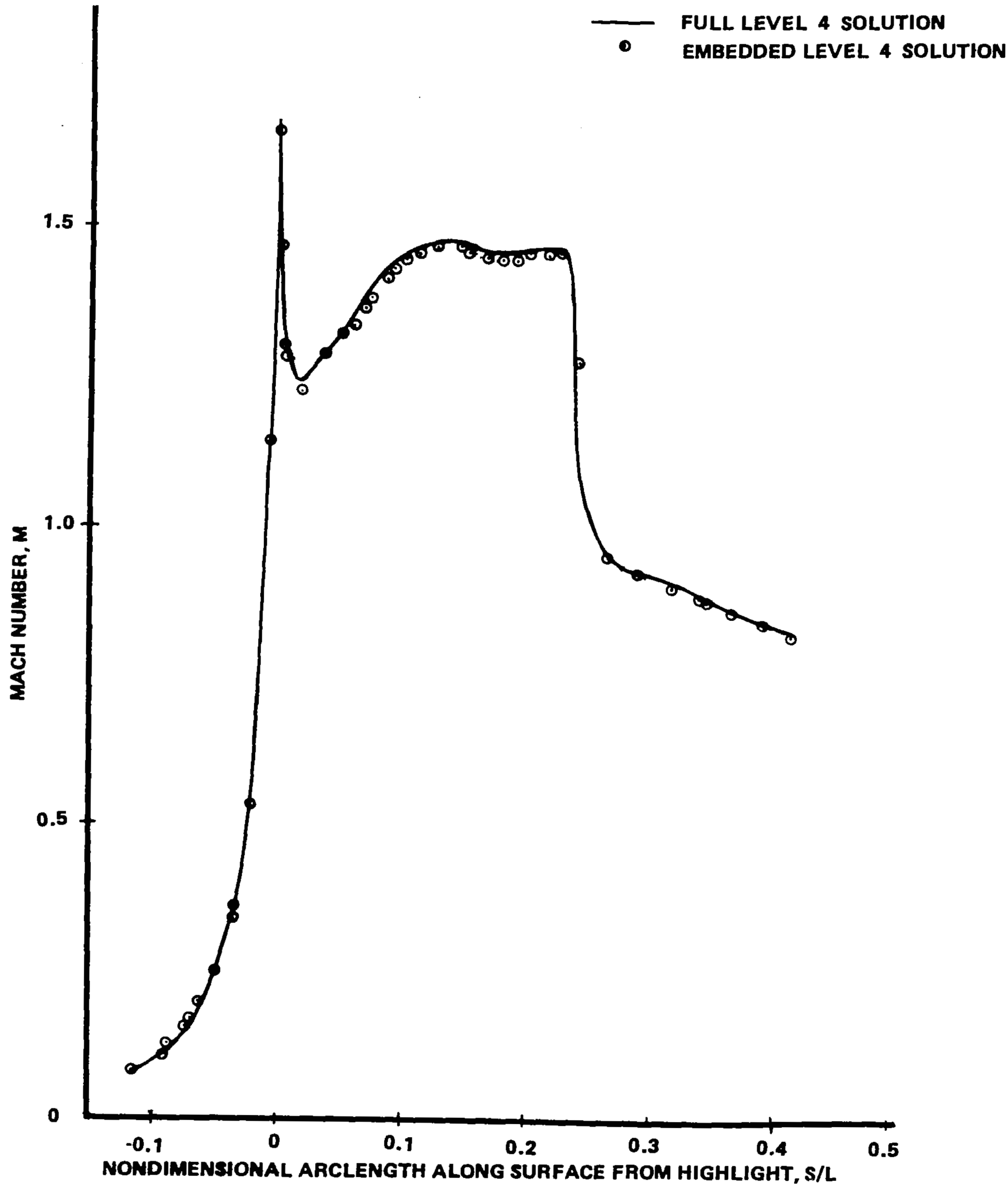


FIGURE 4. A COMPARISON OF A FULL LEVEL 4 SOLUTION WITH AN EMBEDDED LEVEL 4 SOLUTION, NASA TM X - 2937 INLET.

A MULTIGRID ALGORITHM FOR STEADY TRANSONIC POTENTIAL FLOWS AROUND AEROFOILS USING NEWTON ITERATION*

J.W. Boerstoel
National Aerospace Laboratory NLR
Amsterdam, The Netherlands

Abstract

The application of multigrid relaxation to transonic potential-flow calculation was investigated. Fully conservative potential flows around aerofoils were taken as test problems. The solution algorithm was based on Newton iteration. In each Newton iteration step, multigrid relaxation was used to calculate correction potentials. It was found that the iteration to the circulation has to be kept outside the multigrid algorithm. In order to obtain meaningful norms of residuals (to be used in termination tests of loops), difference formulas with asymptotic scaling were introduced. Nonlinear instability problems were solved by upwind differencing using mass-flux-vector splitting instead of artificial viscosity or artificial density. It was also found that the multigrid method cannot efficiently update shock positions due to the (mainly) linear character of individual multigrid relaxation cycles. For subsonic flows, the algorithm is quite efficient. For transonic flows, the algorithm was found robust; its efficiency should be increased by improving the iteration on the shock positions; this is a highly nonlinear process.

* The study was performed under contract for the Netherlands Agency for Aerospace Programs (NIVR), contract number 1853.

1. Introduction

Most computer codes for the calculation of transonic potential flows are based on the solution of a large finite-difference equation system by some nonlinear relaxation algorithm. The development of these algorithms started about a decade ago with work of Murman and Cole, who applied upwind differencing in supersonic zones to generate directional bias (Ref. 1). The most important improvements since then were the introduction of the concept of full discrete conservation (Murman, Ref. 2), the extension to the full nonlinear potential-flow equation (Jameson, Ref. 3), and the application of results of tensor theory to allow non-orthogonal curvilinear grids, so that grids can be easily aligned with complex flow boundaries (Jameson e.a., Ref. 4, Caughey e.a., Ref. 5). An impression of the state-of-art may be obtained from references 6 and 7.

During the last few years, numerical analysts have proposed various new fast-solver algorithms that perhaps may also be used to solve finite-difference equations for transonic potential flow more efficiently than nonlinear relaxation algorithms. The most interesting fast solvers are CR/FFT (cyclic reduction/ fast Fourier transformation), AF (approximate factorization), ILU (incomplete lower-upper decomposition), and MGR (multigrid relaxation). For transonic potential-flow calculations, fast solvers of wide applicability are of particular interest because of the complexity of the potential-flow equation (nonlinear, of elliptic-hyperbolic type, singular at shocks and sonic lines).

The application of multigrid methods to transonic potential-flow calculations was investigated by several authors (Fuchs, Refs 8,9; South and Brandt, Ref. 10; Jameson, Ref. 11; McCarthy e.a. Ref. 12; Arlinger, Ref. 13). Interesting results were obtained by Fuchs for two-dimensional transonic small-perturbation flow around non-lifting symmetrical aerofoils. Some combinations of the various finite-difference equation systems and various versions of multigrid relaxation algorithms tested by Fuchs turned out to be very efficient. The other authors also reported promising results. Approximate factorization techniques have also been applied with success (Holst, Ref. 14; Baker, Ref. 15; Catherall, Ref. 16). As a function of the number of grid points approximate factorization is theoretically asymptotically slower than multigrid relaxation, however. ILU methods have not yet been applied to transonic problems. The application of CR/FFT to transonic flow problems turned out to be not quite successful.

The present study concerns the design of a fast-solver algorithm for transonic potential-flow calculations using Newton iteration and multigrid relaxation.
From preliminary investigations it was known that Newton iteration (exact or approximate) was promising (Boerstoel, Ref. 17; Fuchs, Refs 8,9; Piers and Slooff, Ref. 18). The Newton iteration technique was also proposed by Hackbusch to solve other nonlinear problems than transonic problems (Ref. 19).

Within each Newton iteration step, a linear correction problem has to be solved. This is done with multigrid relaxation.

Various multigrid relaxation algorithms exist (Brandt, Refs 20,21): for example, a nonlinear version known as the FAS (full approximation storage) method, and a linear version known as the cycle C method. In this study, the linear version was applied because the convergence analysis is considerably simpler.

Two-dimensional full potential flow around an arbitrary given aerofoil was used as a test problem. The flow equations were discretized in a fully conservative manner on an approximately orthogonal grid of O-type. The grid is aligned along the aerofoil.

The nonlinear finite-difference equations are presented in section 2, the main structure of the solution algorithm in section 3, the multigrid process in section 4, and the relaxation technique applied in the multigrid process in section 6. Results of numerical experiments and a concluding discussion form the last two sections. Some stability considerations are presented in section 5.

2. Finite-difference equations

The finite-difference equations to be solved are defined on a grid of O-type. Such a grid may be generated by a mapping from an equidistant grid in a computational (ξ,η)-plane to the physical (x,y)-plane. The mapping used here consists of a sequence of a few simple transformations, illustrated in figure 1: a conformal Karman-Trefftz transformation, followed by simple correction transformations. A Karman-Trefftz aerofoil is (crudely) fitted to the aerofoil such that the aerofoil becomes a smooth near-circle under the corresponding Karman-Trefftz transformation. The trailing-edge corner is thereby removed. The subsequent transformations map the aerofoil into a circle (stretching, and shearing in radial direction), and introduce a stretching far from the aerofoil in a direction approximately normal to the streamlines, with a stretch factor $(1-M_\infty^2)^{1/2}$. Near the trailing edge, the total mapping was designed to be conformal, to first order in $(\eta-\eta_u)$;

this permits easy implementation of various forms of the Kutta condition.

An example of a grid is presented in figure 3. As shown, the grid is truncated far from the aerofoil (4 to 10 chords).

Because, in fast-solver algorithms, the corrections to given approximate solutions are in general much greater than the (usually small and smooth) corrections in nonlinear relaxation algorithms, fast-solver algorithms are considerably more prone to nonlinear instability. In the initial stages of the study it was found that various forms of artificial-viscosity terms gave often rise to expansion shocks, in particular on coarser grids, and also at tops of supersonic zones. A nonlinear finite-difference equation system with excellent stability properties was obtained by introducing directional bias with mass-flux-vector splitting; this is a generalization to full potential flow of a concept applied by Engquist and Osher to stabilize the fully-conservative difference equations for transonic small-perturbation flow (Ref. 22). Numerical experiments revealed, that it was also necessary to compute the density at cell-face centres instead of at cell corners. (Computation of the density at cell corners is usual practice in most computer codes.)

The finite-difference equation for the mass conservation equation of each cell (i,j) on the computational plane has the form ($.^T$ means transposition) (see Fig. 4)

$$\nabla^T_{i,j}\ F^d = 0 \quad , \tag{1}$$

where ∇ is a second-order accurate discretization of the gradient operator $(\partial/\partial\xi, \partial/\partial\eta)$, see below for details. F^d is a discrete mass-flux vector with three components (hence, the term mass-flux-vector splitting):

$$F^d = F - F^a + F^{ar} \quad , \tag{2}$$

with F the usual mass-flux vector:

$$F = \rho\ h\ U \quad , \tag{3}$$

$$U = G\ \nabla\varphi \quad , \tag{4}$$

$$\rho = \{1-\tfrac{1}{2}(\gamma-1)M_\infty^2(1-q^2)\}^{1/(\gamma-1)} \quad ,$$

$$q^2 = (\nabla\varphi)^T\ G\ \nabla\varphi \quad . \tag{5,6}$$

G is the contravariant metric tensor, and h the determinant of the mapping $(\xi,\eta) \to (x,y)$. Velocities q and densities ρ have been scaled by their free-stream values. The mass-flux vector F^a is nonzero in supersonic zones:

$$F^a = \text{if} \quad M \leq 1 \quad \text{then} \quad 0 \quad \text{else}$$
$$\{(\rho q - \rho^* q^*)/q\}\ h\ (U\ U^T/q^2)\ \nabla\varphi \quad , \tag{7}$$

where M is the local Mach number, ρ^* and q^* the sonic values of the density and the speed, and $U\ U^T$ is the 2*2 matrix defined by the exterior product of U with itself. The mass-flux vector F^{ar} is equal to F^a at the centre of the upstream cell-face. (F^d will be computed at centres of cell faces).

The components of the mass-flux vectors F and F^a are computed at the centres of cell faces with second-order accurate central-difference and central-averaging formulas applied to φ. The Mach number test involved in the calculation of F^a is also made at the cell-face centres; this implies that the Mach number test for F^{ar} is made at the upstream cell-face centre; it will be seen below that this has interesting consequences for the construction of sonic and shock operators. The metric data are assumed to be known at the cell-face centres.

Physical and mathematical properties of the mass-flux vectors and their discrete divergences are readily obtained by decomposing the matrices G and $(U\ U^T/q^2)$ using the orthonormal matrix

$$V = \begin{bmatrix} \varphi_x/q & -\varphi_y/q \\ \varphi_y/q & \varphi_x/q \end{bmatrix} = \begin{bmatrix} x_s & x_n \\ y_s & y_n \end{bmatrix} \qquad (8)$$

$$V\,V^T = \text{unit matrix} \quad ,$$

and the relation $G = H^{-1}\,H^{-1T}$, H the Jacobian of the mapping from the computational space to the physical space. In (8), (s,n) are the natural coordinates of the flow. In particular, F, F^a, and $F\text{-}F^a$ depend on mass-flux vectors in natural coordinates as follows if $M \geqslant 1$:

$$F = h\,H^{-1}\,V\,[\quad \rho\,\varphi_s \quad , \rho\,\varphi_n]^T \quad ,$$

$$F^a = h\,H^{-1}\,V\,[\rho\,\varphi_s - \rho^*q^*, \quad 0 \quad]^T \quad ,$$

$$F\text{-}F^a = h\,H^{-1}\,V\,[\quad \rho^*q^* \quad , \rho\,\varphi_n]^T \quad . \qquad (9\text{-}11)$$

These expressions show that, in supersonic zones and in natural coordinates, the streamline component of $F\text{-}F^a$ has a fixed sonic magnitude; the other component is zero because $\varphi_n = 0$. Because the scalar $\rho q = \rho\,\varphi_s$, as a function of the speed $q = \varphi_s$, has a maximum at the sonic speed q^*, the vector F^a measures the mass-flux excess in comparison to the sonic maximum mass flux ρ^*q^*. Although $F\text{-}F^a$ is a vector of fixed magnitude in natural coordinates, its divergence is generally nonzero (φ_{nn} is not identically zero); this divergence is a measure of the convergence of the streamlines.

It may be shown that, in smooth parts of supersonic zones, the implicit artificial viscosity generated by the divergence of $F^a - F^{ar}$ is closely related to that of Jameson (Ref. 3), and to the viscosity encountered in the artificial-density used by Eberle (Ref. 25), Hafez e.a. (Ref. 26), and Holst (Ref. 14). At sonic lines and at shocks, the relation between the vector-split-concept and the artificial-viscosity and artificial-density concepts is lost, however.

At sonic lines, expansion shocks cannot occur because the mass-flux vector F^d is forced to have a sonic magnitude:

$$F^d = h\,H^{-1}\,V\,[\rho^*q^*, \,\rho\,\varphi_n]^T \quad ,$$

$$F^a \neq 0 \quad , \quad F^{ar} = 0 \quad , \qquad (12)$$

at the first supersonic cell-face centre after the sonic line. Approximately normal shocks are allowed to become very steep, mainly because, at the first subsonic cell-face centre after the sonic line in the shock, the mass-flux vector F^d has the special value

$$F^d = F + F^{ar} \quad , \quad F = \rho\,h\,U \quad ,$$

$$F^{ar} \neq 0 \quad \text{because} \quad M^r > 1 \quad . \qquad (13)$$

It may be shown that the finite-difference equations for transonic small-perturbation flow proposed by Engquist and Osher have simular properties. They showed that their difference formulas have stable and unique solutions (Ref. 23). The concept of mass-flux-vector splitting presented here is a formal generalization to the full nonlinear flow equation of their splitting.

The precise definition of the discrete gradient operators ∇ in (1,4,7) differs from the usual ones because asymptotic scaling is applied. This has been done to obtain useful norms of residuals to be used in termination tests of iteration loops. Because of the grid stretching and the singular behaviour of the potential near free-stream infinity, the residuals of sufficiently accurate approximate solutions need not be uniformly small over the entire grid, but are allowed to have a certain growth rate when tending to free-stream infinity. Efficient residual norms should account for the permitted growth rate. On O-type grids, the permitted growth rate may be analyzed if finite-difference formulas with asymptotic scaling are applied.

Asymptotic scaling naturally emerges on O-type grids, if we require that the velocity must be approximated uniformly to $O(h^m)^2$ (h^m mesh size) for any sufficient smooth potential φ having the expected asymptotic behaviour when tending to infinity ($\eta \downarrow 0$). This requirement leads to an analysis of the relation between approximation errors of difference formulas and the asymptotic behaviour for $\eta \downarrow 0$ of all kinds of functions of (ξ,η), such as potentials, metric data, mass-flux vectors, residues, etc.

The main steps of the analysis are the following. The mapping from the computational to the physical plane is defined such that

$$x = \eta^{-1} \cos 2\pi\zeta + \ldots , \qquad \beta y = \eta^{-1} \sin 2\pi\zeta + \ldots , \tag{14}$$

$$\varphi = \eta^{-1} \bar{c}_\infty(\zeta) + \Gamma d_\infty(\zeta) + \ldots , \tag{15}$$

$$\zeta = (\xi-\xi_\ell) / (\xi_u-\xi_\ell) + \text{constant} , \qquad \beta = (1 - M_\infty^2)^{1/2} . \tag{16-17}$$

Using these formulas, it may be shown that the metric constants in the expression (6) for q^2,

$$q^2 = (\nabla\varphi)^T G \nabla\varphi = g^{11} \varphi_\xi^2 + 2 g^{12} \varphi_\xi \varphi_\eta + g^{22} \varphi_\eta^2 \tag{18}$$

and the derivatives of the potential have asymptotic magnitudes given by

$$g^{11} = O(\eta^2) , \; g^{12} = O(\eta^3) , \; g^{22} = O(\eta^4), \qquad \varphi_\xi = O(\eta^{-1}), \; \varphi_\eta = O(\eta^{-2}). \tag{19}$$

It follows that q may be approximated with an absolute accuracy of $O(h^m)^2$, indeed, if φ_ξ and φ_η are approximated by difference formulas with an absolute accuracy of the order $\eta^{-1}(h^m)^2$ and $\eta^{-2}(h^m)^2$, respectively.

In general, difference formulas for derivatives f_η of functions $f(\xi,\eta)$ having an asymptotic power series of the form

$$f(\xi,\eta) = c_0(\xi) \eta^{-q} + c_1(\xi) \eta^{-q+1} + \ldots . \tag{20}$$

may be derived from the identity

$$f_\eta = \{(\eta^{q+1} f)_\eta - (q+1) \eta^q f\} \eta^{-q-1} \tag{21}$$

by applying the usual central-difference formulas to the terms $(\eta^{q+1} f)$ and $(\eta^q f)$, because these terms are of unit order in η. The resulting difference formulas are a mixture of numerical and analytical differentiation in η, and have an absolute accuracy of order $\eta^{-q-1}(h^m)^2$.

These general considerations were used to define the discrete gradient operators ∇ as follows. Indicate the usual averaging and first-order difference operators by μ_ξ, μ_η, δ_ξ, δ_η, with

$$\mu_{\xi(i,j)} f = (f_{i+\frac{1}{2},j} + f_{i-\frac{1}{2},j}) / 2 , \qquad \delta_{\xi(i,j)} f = (f_{i+\frac{1}{2},j} - f_{i-\frac{1}{2},j}) / h^m, \tag{22}$$

then (see figure 5 for the stencils)

$$\nabla^T_{i,j} F^d = [\delta_\xi F_1^d + \{ \delta_\eta (\eta^2 F_2^d + - 2 \eta \mu_\eta (\eta F_2^d) \} \eta^{-2}]_{i,j} ,$$

$$\nabla_{i+\frac{1}{2},j} \varphi = [\delta_\xi \varphi, \{ \mu_\eta \delta_\eta (\eta^2 \eta_\xi \varphi) + - 2 \eta \mu_\xi \varphi \} \eta^{-2}]^T_{i+\frac{1}{2},j} , \tag{23-24}$$

$$\nabla_{i,j+\frac{1}{2}}\,\varphi = [\mu_\xi\,\delta_\xi\,\{\,\eta^{-1}\,\mu_\eta\,(\eta\,\varphi)\},\ \{\delta_\eta\,(\eta^2\varphi) - 2\eta\,\mu_\eta\,(\eta\varphi)\}\,\eta^{-2}]^T_{i,j+\frac{1}{2}}\ . \tag{25}$$

Asymptotic scaling is also applied in the retarded mass-flux vectors F^{ar}, when retardation in η-direction has to occur:

$$F^{ar}_{i,j+\frac{1}{2}} = [F^a_{1(i,j+\frac{1}{2}-\alpha)}\,(\eta^2_{j+\frac{1}{2}}/\eta^2_{j+\frac{1}{2}-\alpha}),\ F^a_{2(i,j+\frac{1}{2}-\alpha)}(\eta_{j+\frac{1}{2}}\ /\ \eta_{j+\frac{1}{2}-\alpha})]^T. \tag{26}$$

The Neumann boundary condition on the aerofoil surface is zero mass flux through the aerofoil surface. This condition has been implemented in two linearly independent ways in the finite-difference equation system:

- The mass-conservation equation (1,23) of each cell (i,J-1) adjacent to the aerofoil image is modified by requiring that no mass enters the cell through the cell face $(i,J-\frac{1}{2})$ on the aerofoil image:

$$F^d_{2(i,J-\frac{1}{2})} = 0\quad . \tag{27}$$

- The potential values $\varphi_{i,J}$ inside the aerofoil are coupled to the potential values in the flow field by applying the boundary condition at each cell-face centre $(i,J-\frac{1}{2})$ on the aerofoil in the form

$$\dot{F}_{2(i,J-\frac{1}{2})} = 0\quad ; \tag{28}$$

F^a and F^{ar} are thus not used in this boundary condition, so that a second row of potential values inside the aerofoil is not needed.

The Dirichlet boundary condition is applied on a large closed curve $\eta = \eta_\ell$ around the aerofoil (in calculations, 4 to 10 chords from the aerofoil):

$$\varphi_{i,\frac{3}{2}} = \bar{x}_{i,\frac{3}{2}} - (\Gamma/2\pi)\ \arctan\ (\beta\bar{y}/\bar{x})_{i,\frac{3}{2}}, \tag{29}$$

$$\begin{aligned}\bar{x} &= \ x\cos\alpha_\infty + y\sin\alpha_\infty\ ,\\ \bar{y} &= -x\sin\alpha_\infty + y\cos\alpha_\infty\ ,\end{aligned} \tag{30}$$

with α_∞ the free-stream incidence. Because the potential values are given on the free-stream boundary $j = \frac{3}{2}$ instead of at the grid points (i,1), one-sided second-order accurate difference formulas have been used at the free-stream boundary instead of the central formulas (24,25).

The circulation Γ is determined by the Kutta condition. Because near the trailing edge the grid is approximately conformal, the Kutta condition may be given the form $(\varphi_\xi)_{te} = 0$ if the flow is subsonic at the trailing edge. $(\varphi_\xi)_{te}$ is approximated by a central-difference formula.

3. Solution algorithm

The nonlinear finite-difference equation system is solved by a fast-solver algorithm based on the combined use of Newton iteration and multigrid relaxation. The main structure of this algorithm is presented in this section.

During the study it was found that due attention has to be paid to a few new problems.

- Circulation changes give in general rise to an increase of norms of residuals. In order to prevent limit cycling or divergence of nested iteration processes (here, Newton iteration and multigrid relaxation), the increase must be allowed for in termination criteria of iteration loops.
- The solution algorithm has to iterate on different types of nonlinearity: a short-wavelength nonlinearity at shocks with a length scale of the order of one mesh of the

finest grid in a current grid sequence, and a mild nonlinearity elsewhere with length scales of geometric properties of the aerofoil such as the chord or the leading-edge radius. The short-wavelength nonlinearity is encountered during shock-position improvements. Both types of nonlinearity have to be processed after each circulation improvement, because circulation changes give rise to changes far from the aerofoil (almost linear, length-scale the chord), at the leading edge (mildly nonlinear, length scale l.e. radius), and at shocks (strongly nonlinear, length scale the mesh of the finest grid).

It was also found that multigrid processes do not efficiently improve shock positions on fine grids if the shocks have to move over several (say five) meshes of such grids. (Such movements may be easily required, by circulation updates, for example.) This may be explained as follows. Multigrid relaxation is based on the assumption that a correction to an approximate solution may be decomposed into a sum of short-wavelength and long-wavelength components; the short-wavelength components are (efficiently) computed on the finest grid of a grid sequence, and the long-wavelength components are computed on coarser grids where less grid points are involved in the calculations. Such a linear decomposition of a correction grid function in short- and long-wavelength components has sense in linear problems, and also in linearizations of nonlinear problems. However, linearizations of the shock operators can at best estimate shock movements over one mesh of the finest grid. In fast-solver algorithms, the shock should be able to move over several meshes, however. The basic assumption of linearity of the multigrid relaxation process conflicts thus with the nonlinearity of the shock-movement process. (This is also true for the nonlinear FAS-multigrid relaxation method proposed by Brandt (Ref. 21), because Brandt's construction of the FAS method makes use of the linearity of the correction problem on the finest grid.) Other details concerning multigrid relaxation and shock position updates are presented in figure 6.

An iteration process in which these general considerations have been taken into account may be chosen to consist of an outer Newton iteration on the circulation Γ, and two inner iteration procedures, one for the calculation of corrections outside shocks (whereby multigrid relaxation is applied), and one for the update of shock positions. This combination of iteration procedures may be represented by the following algorithm.

initialise φ, Γ_a;

<u>until</u> Γ_a accurate enough <u>do</u>

<u>begin</u> <u>until</u> flow equations at fixed Γ_a are solved <u>do</u>

<u>begin</u> improve φ at fixed Γ_a by one multigrid cycle;

improve shock positions with partial relaxation sweeps;

<u>end</u> of iteration at fixed Γ_a;

compute error in Kutta condition;

improve circulation estimate Γ_a;

<u>end</u> of outer Newton iteration on Γ.

The outer Newton iteration on the circulation is based on a split of the finite-difference equation system of the form

$$L\{\varphi(\Gamma)\} = Q \quad , \tag{31}$$

$$\{\partial\varphi(\Gamma)/\partial\xi\}_{te} = 0 \quad , \tag{32}$$

where the last equation is the Kutta condition, and the first one represents all other equations. The solutions $\varphi(\Gamma_a)$ of the nonlinear system (31) alone define a (nonlinear) relation between Γ_a and $\{\partial\varphi(\Gamma_a)/\partial\xi\}_{te}$, see figure 7 for an illustration. The Kutta condition (32) means that we are interested in the value Γ on the horizontal axis. We may iterate to this value as shown in the figure. The slopes needed in this Newton iteration process are estimated by numerical differentiation. On a fixed grid, usually three to four steps are sufficient to fix the lift coefficient to about three significant figures.

In each step of the outer Newton iteration process to Γ, the nonlinear equation system (31) has to be solved for a fixed estimate Γ_a of the desired value of the circulation. This is done iteratively. In each iteration step, the solution is first improved outside shock layers by one multigrid-relaxation cycle (see section 4), whereby shock positions do hardly vary. This multigrid-relaxation cycle is followed by an update of shock positions with nonlinear relaxation sweeps on the finest grid in (small) subdomains around shocks (partial relaxation sweeps). In the first partial-relaxation sweep, the subdomains to be relaxed cover only shock cells. In each subsequent sweep, the subdomains are enlarged by one row of cells upstream, above, below and downstream of the subdomain relaxed in the previous sweep. This enlargement is necessary, because partial relaxation on fixed subdomains may lead to divergence due to increase of residuals at the boundaries of the subdomain.

The termination of the iteration to a solution of (31): $L\{\varphi(\Gamma_a)\} = Q$ for a fixed estimate of Γ_a of the desired circulation is based on a test combining two criteria. When the circulation is not yet accurate enough, the iteration terminates as soon as the value of $\{\varphi_\xi\}_{te}$ is so accurate that it may be reliably used to improve the circulation to a better estimate. However, when the circulation is accurate enough, the iteration to a solution is terminated when a norm of the residuals of the mass-conservation equations of the cells has become small enough. This test strategy drives the circulation as fast as possible to its final value.

A suitable norm of the residuals was found to be the maximum norm (see (1))

$$R_{max} = \max_{i,j} \{(\eta_j/\eta_u)^2 \, |\nabla^T_{i,j} \, F^d|\} \, h^m \quad . \tag{33}$$

The scaling of the residuals $\nabla^T_{i,j} \, F^d$ by η_j^2 reflects, that the residuals of sufficiently accurate approximate solutions are of order $\eta^{-2} \, O(h^m)^2$; hence, they are allowed to grow with $\eta \downarrow 0$. The scaling by the mesh size h^m makes the norm nondimensional (hence, mesh-size independent). A maximum norm is preferred over a rms-norm, because an rms-norm does not show that, in certain stages of the iteration process, large residuals may occur in very small regions. For example, when iterating at fixed Γ_a, after each multigrid-relaxation sweep, the residuals of shock cells are usually at least an order of magnitude larger than elsewhere in the flow due to velocity overshoots (or undershoots) as sketched in figure 6. Rms-norms (or average absolute-value norms) do not efficiently measure large residuals in such small regions, and cannot be safely used in termination tests of loops.

In subsonic-flow calculations, the partial-relaxation sweeps are suppressed.

The Newton iteration process on a grid is started with an initial approximation of the potential that is computed by solving the nonlinear finite-difference equation system (31,32) on a coarser grid (mesh size doubled). This is repeated on a sequence of three to four grids. On the coarsest grid of this sequence, the entire calculation is started with a uniform-flow potential having no circulation.

4. Multigrid-relaxation cycles

The multigrid relaxation cycles used in this study have a general structure closely resembling the cycle C algorithm of Brandt (Ref. 21). Its general structure is presented in figure 8a. It may be seen that the cycle starts with a linearization of the flow equations on the finest grid so that the whole cycle effectively represents one (approximate) Newton iteration step.

A few details of the implementation of this multigrid relaxation cycle are of special interest.

- The restriction operations are applied to the complete linearized conservation equations instead of residuals. This is done in such a way that the equation system on the coarser grids may be interpreted as approximations to the mass conservation equations on the finest grid.
- Certain stability properties of the linearized flow equations are transferred in a controlled way to the coarser-grid equations. See section 5 for more details.

As shown in figure 8a, each multigrid relaxation cycle starts with a linearization of the nonlinear flow-equation system (31) on the finest grid around a given approximate solution ϕ^m. The result is a linear first-variation equation system for a first-variation potential $d\varphi^m$ on the finest grid,

$$\varphi^m = \phi^m + d\varphi^m \quad , \tag{34}$$

$$C^m \, d\varphi^m = dR^m \equiv Q^m - L^m\{\phi^m\} \quad . \tag{35}$$

The long-wavelength part of the correction $d\varphi^m$ will be computed on the coarser grids of the grid sequence. This requires the definition of equation systems

$$C^n \, d\varphi^n = dR^n \quad , \qquad n = m-1(1)1 \ , \tag{36}$$

for these long-wavelength parts on the coarser grids by a restriction process. In order to obtain simple restriction rules based on the interpretation of the linearized equation system (35) as a system of mass-conservation equations and boundary conditions, the grids are chosen staggered so that four cells of a grid coincide with one cell of the next-coarser grid, see figure 8b. Then the equation systems $C^n \, d\varphi^n = dR^n$ may be defined recursively from the one on the finest grid.

At each cell of a grid H^n, the first-variation equation is assumed to be known and to have the form:

$$\nabla^{nT}_{i,j} \, dF^{dn} = dR^n_{i,j} \quad , \tag{37}$$

$$dF^{dn} = dF^n - dF^{an} + dF^{arn} \quad , \tag{38}$$

$$dF^n = P^n \, \nabla^n d\varphi^n \quad , \tag{39}$$

$$dF^{an} = A^n \, \nabla^n d\varphi^n \quad , \tag{40}$$

where, on the finest grid $(n = m)$, the $dR^n_{i,j}$ are residuals of ϕ in (1):

$$dR^m_{i,j} = - \nabla^{mT}_{i,j} \, F^{dm} \quad . \tag{41}$$

dF, dF^a, and dF^{ar} are the first-variations of F, F^a, and F^{ar} around ϕ so that, on the finest grid, P^m and A^m may be shown to be the 2*2-matrices

$$P^m = [\rho \ h \ \{G - M^2 \ U \ U^T/q^2\}]^m , \tag{42}$$

$$A^m = 0 \quad \text{if} \quad M \leqslant 1 \quad \text{else}$$
$$[\rho \ h \ (1-M^2) \ (U \ U^T/q^2)]^m \ . \tag{43}$$

Each equation

$$V_{\underline{i},\underline{j}}^{(n-1)T} \, dF^{d(n-1)} = - \, dR_{\underline{i},\underline{j}}^{(n-1)} \qquad (44)$$

of a large cell $(\underline{i},\underline{j})$ on a grid H^{n-1} (see Fig. 8) is defined from the four corresponding equations (37) on grid H^n by requiring that the coarse-grid equation should represent a mass-conservation equation for the long-wavelength content $d\varphi^{n-1}$ of a suitable class of correction grid functions $d\varphi^n$. From this requirement, restriction rules for the residues and the coefficient matrices are readily derived with mass-flux considerations. For example, the mass flux through the face $(\underline{i},\underline{j}+\frac{1}{2})$ of the large cell $(\underline{i},\underline{j})$ in figure 8 should be equal to the total mass flux of the two corresponding faces of the small cells, giving

$$(P \, \nabla d\varphi)_{\underline{i},\underline{j}+\frac{1}{2}}^{n-1} = \frac{1}{2} \left[(P \, \nabla d\varphi)_{i,j+\frac{3}{2}}^{n} + (P \, \nabla d\varphi)_{i+1,j+\frac{3}{2}}^{n} \right]. \qquad (45)$$

This should be true for the long-wavelength content in $d\varphi^n$. For the long-wavelength content, the three gradients in (45) are about equal:

$$(\nabla d\varphi)_{i,j+\frac{1}{2}}^{n-1} \cong (\nabla d\varphi)_{i,j+\frac{3}{2}}^{n} \cong (\nabla d\varphi)_{i,j+\frac{3}{2}}^{n} \quad , \qquad (46)$$

so that an equation for the coefficient matrix P on the coarse grid is found:

$$P_{\underline{i},\underline{j}+\frac{1}{2}}^{n-1} = \frac{1}{2} \left(P_{i,j+\frac{3}{2}}^{n} + P_{i+1,j+\frac{3}{2}}^{n} \right) . \qquad (47)$$

Similar arguments are used to define the other coefficient matrices at the cell-face centres on the grid H^{n-1}. The residues are readily restricted by applying a discrete version of Gauss' theorem.

From (47) and similar formulas it follows that the coefficients at the cell-face centres are determined with coefficient-weighting. Residual weighting is also applied: each residual $dR_{\underline{i},\underline{j}}^{n-1}$ on the coarse grid turns out to be a weighted average of the residuals of the four smaller cells on the next-finer grid that are covered by the coarse-grid cell $(\underline{i},\underline{j})$.

The linearized forms of the Neumann boundary conditions (27,28) are restricted to coarser grids in a similar way as illustrated by (45-47).

When the equation system $C^m \, d\varphi^m = dR^m$ for the desired correction $d\varphi^m$ of the potential has been restricted to the coarser grids, $d\varphi^m$ is estimated. This is done by a recursive process involving on each grid improvement of $d\varphi^n$ by line relaxation, and subsequent prolongation to the next-finer grid by bilinear interpolation. The process starts on the coarsest grid by putting the initial correction potential $d\varphi^1$ zero. When a $d\varphi^{m-1}$ has been prolonged to a $d\varphi^m$ on the finest grid of the grid sequence, $d\varphi^m$ is first added to the last potential ϕ^m to a new ϕ^m; this new potential is subsequently improved by nonlinear line relaxation over the entire finest grid. This nonlinear relaxation on the entire finest grid terminates the multigrid cycle.

On each grid, one line-relaxation sweep over the grid is sufficient. On the coarsest grid H^1, it is desirable to make more sweeps, however, to obtain a reasonable estimate of $d\varphi^1$. Four sweeps were found a suitable number in applications.

5. Stability

From numerical experiments, it was found that both the application of mass-flux-vector splitting as well as calculation of gradients, velocities, and densities at cell-face centres are necessary to obtain good stability properties. As far as stability at sonic lines and shocks is concerned, much insight may be obtained from (9-13).

It is also very helpful to analyze the structure of the coefficient matrices in the first-variation equation (37) of each individual nonlinear discrete mass-conservation equation (1). A necessary condition for stability of the nonlinear finite-difference equation system is stability of each individual first-variation equation (37), because (37) is an exact linearization of (1). The last property is a consequence of the computation of q and ρ from $\nabla\varphi$ at cell-face centres (in stead of at cell-face corners, as usually is done). The stability of each first-variation equation (37) depends on the eigenvalues of the matrices P^m-A^m and A^m, see (42,43). It may be shown that, in subsonic flow, P^m is positive definite and A^m is zero while, in supersonic flow, P^m-A^m and A^m both are precisely semi-definite, with A^m having one negative eigenvalue corresponding to the streamline direction U/q, and P^m-A^m having one zero eigenvalue also corresponding to the streamline direction U/q. This may be concluded from the following factorization for supersonic flow of the mass-flux vectors and matrices (the superscript m for the grid is omitted):

$$F-F^a = hH^{-1}V\begin{bmatrix} \rho^*q^*/q & 0 \\ 0 & \rho \end{bmatrix} V^T H^{-1T}\nabla\varphi ,$$

$$F^a = hH^{-1}V\begin{bmatrix} (\rho q-\rho^*q^*)/q & 0 \\ 0 & \rho \end{bmatrix} V^T H^{-1T}\nabla\varphi ,$$

$$P-A = hH^{-1}V\begin{bmatrix} 0 & 0 \\ 0 & \rho \end{bmatrix} V^T H^{-1T} ,$$

$$A = hH^{-1}V\begin{bmatrix} \rho(1-M^2) & 0 \\ 0 & 0 \end{bmatrix} V^T H^{-1T} , \qquad (48\text{-}51)$$

where V is the orthornormal matrix (8), and H is the Jacobian of the mapping from the computational to the physical plane, so that $G = H^{-1} H^{-1T}$. Mass-flux-vector splitting leads thus to an exact separation of the positive and negative eigenvalues of the matrix P associated with the first-variation $dF = P\,\nabla d\varphi$ of the mass-flux vector $F = \rho\, h\, U$. The positive eigenvalue in the part $P-A$ suggests central differencing for $F-F^a$ and its first variation $dF-dF^a$, the negative eigenvalue in the part A suggests upstream differencing for F^{ar} and dF^{ar}.

The factorizations follow directly from the relation

$$[\varphi_s \;\; 0]^T = V^T H^{-1T} \nabla\varphi \quad ,$$

(this follows from the chain rule for differentiation and from $\varphi_n = 0$), so that, together with (4,8), we obtain

$$U / q = H^{-1} V [\,1 \quad 0\,] \quad , \qquad (52)$$

$$U U^T / q^2 = H^{-1} V \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} V^T H^{-1T} . \qquad (53)$$

It will be seen from the results to be presented that approximately normal shocks are very steep. Detailed analysis of shock operators shows that this is a direct consequence of the eigenvalue of $P-A$ corresponding to the streamline direction being zero, while $F-F^a$ has a constant sonic streamline component; see (11), and (48,52). The last property means that the central-difference part of the mass-conservation equation (1) of cells just ahead of approximately normal shocks are independent of the large φ_{ss} in the shock. On fine grids, the Jameson artificial viscosity has also this property, if the coefficients in this viscosity are evaluated at cell centres.

Mass-flux-vector splitting as presented here has been extensively tested numerically, and was found useful for approximately normal shocks because the matrix $U\,U^T/q^2$ in the mass-flux vector F^a is the image in the computational plane of a unit vector along the streamline in the physical plane, see (52). It may be expected that steep oblique shocks will at least require replacement of $U\,U^T/q^2$ by some matrix $\tilde{U}\,\tilde{U}^T$, where $\tilde{U}$ is the image in the computational plane of a unit vector approximately normal to the oblique shock.

The definiteness properties of the matrices P^m-A^m and A^m are used for the design of diagonally-dominant tridiagonal matrices to be applied in line-relaxations, see section 6 below. The definiteness properties are transferred to coarser grids by the simple restriction rules of the form (47), so that on coarser grids these properties are easily traced. This is important for the convergence of relaxations on the coarser grids.

6. Line relaxation

In each multigrid relaxation sweep and on each grid H^n of the grid sequence, an approximation of the solution the first-variation equation system (36): $C^n\,d\varphi^n = dR^n$ or of the nonlinear equation system (31): $L^n(\varphi^n(\Gamma_a)) = Q^n$ is improved by one or a few line-relaxation sweeps over the entire grid. Relaxation in downstream direction is applied; a sweep over the lower part of the aerofoil is followed by a sweep over the upper part.

Tridiagonal equation systems to be used in line relaxation sweeps may be derived in various ways. For example, Jameson used an analysis of a pseudo-time-dependent process to derive these relaxation equations (Ref. 23). However, the relaxation equations may also be derived directly from the first-variation equations by considering relaxation on each individual grid line as a crude Newton iteration step at that grid line.

The derivation of the relaxation equation system of each i-line starts thus with the assumption that for the potential values on the i-line a correction problem has to be solved. We may put on the entire grid:

$$dC^n = d\varphi^n - d\phi^n \quad \text{or} \quad dC^n = \varphi^n - \phi^n \quad , \tag{54}$$

where $d\phi^n$ or ϕ^n are given estimates of potentials, and dC^n is the correction to be computed from a relaxation equation system. Initially, this system has the same form as the first-variation equation (37-40), with the mass-flux vectors dF^n, dF^{an}, and dF^{arn} now depending on dC^n in stead of on $d\varphi^n$:

$$dF^n = P^n\,\nabla^n dC^n \quad , \tag{55}$$

$$dF^{an} = A^n\,\nabla^n dC^n \quad , \tag{56}$$

while the right-hand side is replaced by the residual of $d\phi^n$ or ϕ^n in (37) or in (1). This equation system is subsequently crudely approximated to a simple relaxation system for the calculation of $dC_{i,j}$-values on line i, with a tridiagonal diagonally-dominant coefficient matrix. The approximation process consists of the following steps.

- Finite-difference formulas with asymptotic scaling are replaced by the usual difference formulas.
- Second-order cross-differences $dC^n_{\xi\eta}$ are removed by zeroing the off-diagonal elements in the matrices P^n-A^n and A^n. The only differences that remain are those of $dC^n_{\xi\xi}$ and $dC^n_{\eta\eta}$ multiplied by diagonal elements of P^n-A^n and A^n with a known sign (see the discussion in section 5 about the definiteness properties of P^n-A^n and A^n).
- Retarded fluxes dF^{arn} representing inflow into a cell are zeroed. This simulates, for each cell in the supersonic zone, a zero initial condition if the calculation of $dC^n_{i,j}$-values on

line i is considered to be an isolated subproblem.
- Corrections $dC^n_{i+\alpha,j+\beta}$ at points $(i+\alpha,j+\beta)$ not yet updated in the current sweep are zeroed. This simulates a Dirichlet boundary condition in the subproblem.

The resulting tridiagonal system is augmented by a formula for the improvement of the Neumann boundary condition, derived by linearizing the nonlinear condition (28), and (crudely) approximated by one-sided differences at the point (i,J) in such a way that diagonal-dominance is preserved.

The tridiagonal equation system derived in this way from the linear first-variation equations turns out to be practical identical to that of Jameson (Refs 23,3,5) if the flow is subsonic or supersonic, however, without Jameson's subsonic or supersonic relaxation factors. At sonic lines and shocks, a comparison with Jameson's formulas was not possible because of lack of published results. The relaxation equation of sonic or shock cells turns out to be different from those elsewhere in the flow if they are derived from the first-variation equation.

Relaxation factors were not used in the calculation results presented below, except at sonic cells where underrelaxation was applied.

7. Results of numerical experiments

From a large number of numerical experiments, a number of cases have been selected. This selection permits a separate analysis of the effect of circulation changes, grid changes, shock-position variations, etc.

All results presented were produced by calculations made on three successive grids of size 34*10, 66*18, 130*34, which are numbered 2, 3, and 4. Each 130*34 grid is similar to that of figure 3. Multigrid sweeps for the calculations on grid 2 used two grids levels, multigrid sweeps for grid 3 three, and multigrid sweeps for grid 4 used four grid levels.

Figure 9 concerns the flow around a symmetrical 12.8 % thick Karman-Trefftz aerofoil at $M_\infty = 0$, $\alpha_\infty = 0$ (linear problem, no circulation). Two conclusions follow from the convergence history.
- The multigrid convergence rate is good (about 0.6 per multigrid cycle).
- The residual norm increases when a solution is prolonged to a next-finer grid to serve as starting solution. This is due to a poor resolution of the coarser grids at the leading edge, as will be evident from figure 3 when 4 or 16 cells of the finest grid are grouped together to one cell of a coarser grid.

Results of the incompressible flow around the same Karman-Trefftz aerofoil, at an incidence α_∞ of 10^o, are presented in figure 10. It is seen that:
- changes in circulation, in particular large changes, may lead to large increases of residual norms. This is due to large changes of the solution at the leading edge.
- the grids 2 and 3 are too coarse at the leading edge to permit a reasonably accurate calculation of the expansion of the flow at the leading edge.
- the multigrid convergence rate is good (about 0.6 per multigrid cycle).

Results for a high-subsonic flow are presented in figure 11 (NACA0012, $M_\infty = 0.63$, $\alpha_\infty = 2^o$). The addition of subsonic nonlinearity does not lead to new conclusions.

The added complication of a shock in the calculation process is first considered for a nonlifting case with a moderate shock (NACA0012, $M_\infty = 0.8$, $\alpha_\infty = 0$). Results are presented in figure 12. New conclusions are the following.
- After each multigrid-relaxation cycle, and on all but the coarsest grid 2, the maximum norm of the

residuals, (33), is usually found to be increased by an order of magnitude. This annoying behaviour of the maximum norm is due to velocity overshoots or undershoots at shocks, as discussed in section 3, and illustrated in figure 6. A typical example of a velocity overshoot is presented in figure 12d. (This figure is the result of a somewhat different algorithm not presented in this report; the velocity overshoot effect is representative, however.) As discussed, the velocity overshoot is due to a tendency of multigrid-relaxation cycles to keep the shock position fixed. See also Jameson's remark in reference 11, page 125 about "the appearance ahead of the shock of a temporary overshoot", and the corresponding flat segment in the convergence history in his figure 2b. This implies that there must be large residuals in small zones keeping his average-absolute-value norm temporarily about constant.

- The velocity overshoots can be reliably transformed with partial relaxation sweeps on the current finest grid to appropriate shock displacements: partial relaxation usually reduces the residual norm considerably.
- The lack of resolution at the leading edge on the coarser grids leads to too small flow expansion over the leading edge and to too forward shock positions. It may be expected that improvement of the resolution at the leading edge on grids 2 and 3 will lead to better pressure distributions so that a smaller calculation effort to improve shock positions is required (see Holst and Brown, Ref. 24).

A transonic case with lift is presented in figure 13 (NACA0012, $M_\infty = 0.75$, $\alpha_\infty = 2^o$). This case has also been computed in references 7 and 11. There are no important new points to be observed. The peak Mach number ahead of the shock is 1.37. Hence, for practical purposes, the shock is fairly strong. The shock covers obviously two cells; this is always true.

Central processor times of the research code used for the numerical experiments are presented in table 1. These times were measured on the NLR Cyber 73-28 computer. The table illustrates that the algorithm is efficient for subsonic flows. For transonic flows, the computation

(M_∞,α_∞)	GRID	NBR OF MGR CYCLES	NBR OF PART. RELAX. SWEEPS	CP-S PER GRID	TOTAL CP-S
(.63,0)	2	14	0	31	
	3	10	0	74	
	4	8	0	219	324
(.8,0)	2	8	14	23	
	3	5	30	64	
	4	10	122	680	767
$(.63,2^o)$	2	21	29	58	
	3	23	68	231	
	4	25	278	1544	1833

times are too long. This is primarily due to the partial relaxation sweeps used to update shock positions. A continued search for improved shock-position update algorithms will be required.

Results of computations with various forms of artificial-viscosity terms instead of split mass-flux-vectors are omitted. We found all artificial viscosity-terms tested to have poor stability properties at sonic lines and/or shocks, in particular on coarse grids, and also at the tops of supersonic zones when corrections were large. This was due to the fact that the viscosity terms did not deliberately exclude expansion shocks.

8. Conclusions

From the results presented in this study it is evident that the introduction of multigrid methods in transonic potential-flow calculations is not a simple matter. A number of conclusions are clear from the

present study, however.
- Circulation changes give usually rise to an increase of residual norms (section 7).
- Fast-solver algorithms are considerably more sensitive to nonlinear instability at shocks and sonic lines. This requires difference formulas with excellent stability properties. Such formulas are obtained with mass-flux-vector splitting (sections 2,5).
- It will be hard to improve shock-wave positions by multigrid relaxation processes, because multigrid relaxation is a linear or weakly nonlinear (FAS) correction process, while shock-wave displacement processes are highly nonlinear (section 3, figure 6, section 7). Usually, velocity overshoots or undershoots at shocks have to be eliminated during the calculation process.
- In order to obtain a useful residual norm to be used in termination test of loops, finite-difference formulas with asymptotic scaling have been used (section 2, equation (33)). Compared to the usual finite-difference formulas, these are more expensive, however. There is a possibility that asymptotic scaling may be avoided if termination tests not using residual norms can be found.
- The algorithm was found to be reasonably robust in numerical experiments (section 7). This conclusion is supported by theoretical results concerning the stability of the equation system (section 5; section 2, eqs (12-13)).

Concluding, it can be said that, for subsonic-flow calculations, the algorithm was found to be quite efficient. For transonic-flow calculations with shocks, the algorithm was found to be reliable. However, more efficient procedures for the update of shock-positions are required; these should be at least as robust (convergence guaranteed) as the preliminary procedure (partial relaxation near shocks) investigated in this study.

9. References

[1]Murman, E.M.; Cole, J.D.: Calculation of plane steady transonic flows, AIAA Journ., 9,1 (Jan. 1971) pp. 114-121.

[2]Murman, E.M.: Analysis of embedded shock waves calculated by relaxation methods, AIAA Journ., 12, 5 (May 1974) pp. 626-633.

[3]Jameson, A.: Transonic potential flow calculations using conservation form, Proc. AIAA 2nd Comp. Fl. Dyn. Conf. (June 1975) pp. 148-174.

[4]Jameson, A.; Caughey, D.A.: A finite-volume method for transonic potential flow calculations, Proc. 3rd Comp. Fl. Dyn. Conf. (June 1977) pp. 35-54.

[5]Caughey, D.A.; Jameson, A.: Numerical calculation of transonic potential flow about wing-body combinations, AIAA Journ. 17,2 (Feb. 1979) pp. 175-181.

[6]Van der Kolk, J. Th.; Slooff, J.W.: A comparison of computational results for transonic flow around the ONERA-M6 wing, NLR-report prepared for GARTEUR AD(AG01) (1981) (to be published).

[7]Rizzi, A.: Numerical methods for the computation of inviscid transonic flow with shock waves, FFA-report prepared for GAMM-workshop (Sept. 1979).

[8]Fuchs, L.J.: Finite-difference methods for plane steady inviscid transonic flows, rep. TRITA-GAD-2, Stockholm (May 1977).

[9]Fuchs, L.J.: A Newton-multigrid method for the solution of nonlinear partial differential equations, Proc. 1st Conf. Bound. Interior Layers/ Comp. Asympt. Methods, Dublin (1980).

[10] South, J.C. Jr; Brandt, A.: Application of a multi-level grid method to transonic flow calculations, ICASE rep. 76-8 (1976).

[11] Jameson, A.: Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple grid method, AIAA 4th Comp. Fl. Dyn. Conf. (July 1979) pp. 122-146.

[12] McCarthy, D.A.; Rheyner, T.A.: A multi-grid code for threedimensional transonic potential flow about axisymmetric inlets at angle of attack, AIAA paper 80-1365 (1980).

[13] Arlinger, B.: Multigrid technique applied to lifting transonic flow using full potential flow equation, SAAB-Scania rep. L-0-1 B439 (Dec. 1977).

[14] Holst, T.L.: Implicit algorithm for the conservative transonic full-potential equation using an arbitrary mesh, AIAA Journ., 17, 10 (Oct. 1979) pp. 1038-1045.

[15] Baker, T.J.: A fast implicit algorithm for the nonconservative potential equation, Open forum presentation, AIAA 4th Comp. Fl. Dyn. Conf. (July 1979).

[16] Catherall, D.: Optimum approximate-factorisation schemes for 2D steady potential flows, AIAA paper 81-1018 (July 1981).

[17] Boerstoel, J.W.; Simons, L.J.: On the approximate solution of the transonic small perturbation equation with a Galerkin/least-squares method and bicubic Hermite functions, NLR-rep. TR 78134 C (1978) (Confidential).

[18] Piers, W.J.; Slooff, J.W.: Calculation of transonic flow by means of a shock-capturing field panel method, AIAA Comp. Fl. Dyn. Conf. (July 1979) pp. 147-156.

[19] Hackbush, W.: On the fast solution of nonlinear elliptic equations, Num. Math., 32 (1979) pp. 83-95.

[20] Brandt, A., Dinar, N.: Multigrid solutions to elliptic flow problems, ICASE rep. 79-15 (July 1979).

[21] Brandt, A.: Multi-level adaptive solutions to boundary-value problems, Math. Comp., 31, 138 (Apr. 1977) pp. 333-390.

[22] Engquist, B.; Osher, S.: Stable and entropy satisfying approximations for transonic flow calculations, Math. Comp., 34, 149 (Jan. 1980) pp. 45-75.

[23] Jameson, A.: Iterative solution of transonic flows over airfoils and wings, including flows at Mach 1, Comm. Pure Appl. Math., 27 (1974) pp. 283-309.

[24] Holst, T.L.; Brown, D.: Transonic airfoil calculations using solution-adaptive grids, AIAA paper 81-1010 (1981).

[25] Eberle, A.: Eine Methode finiter Elemente zur Berechnung der transsonischen Potential-Strömung um Profile, MBB-rep. UFE1352(Ö) (Sept. 1977).

[26] Hafez, M.M.; South, J.; Murman, E.: Artificial compressibility methods for numerical solution of transonic full potential equation, AIAA Journ. 17,8 (Aug. 1979) pp. 838-844.

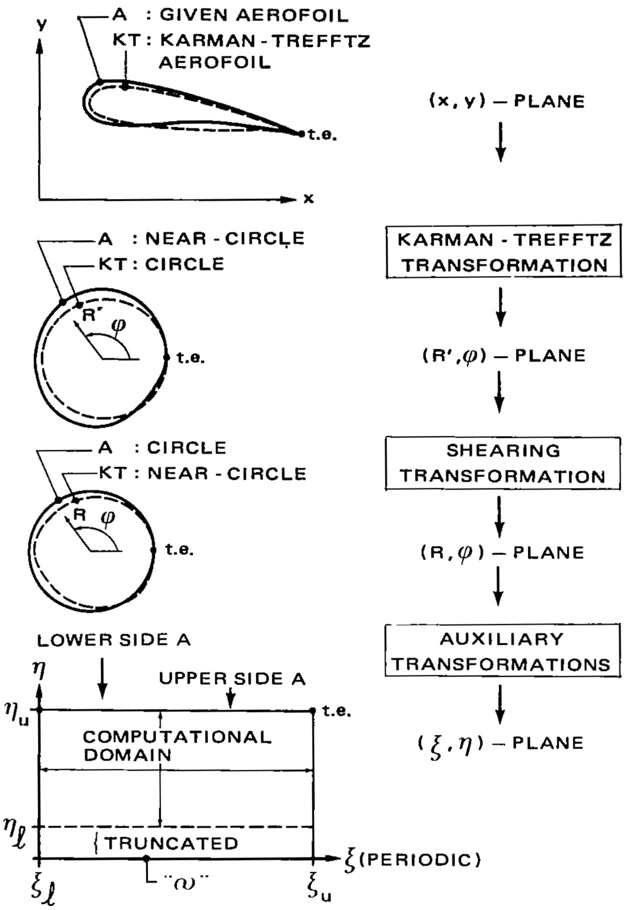


Fig. 1 Mapping of grid generator

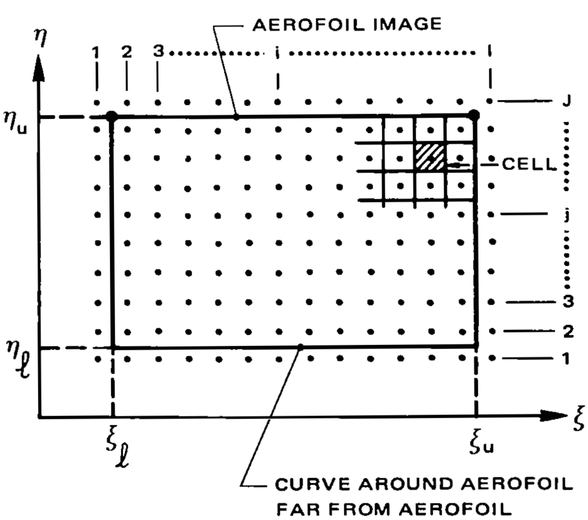


Fig. 2 Grid point and cell arrangement on computational plane

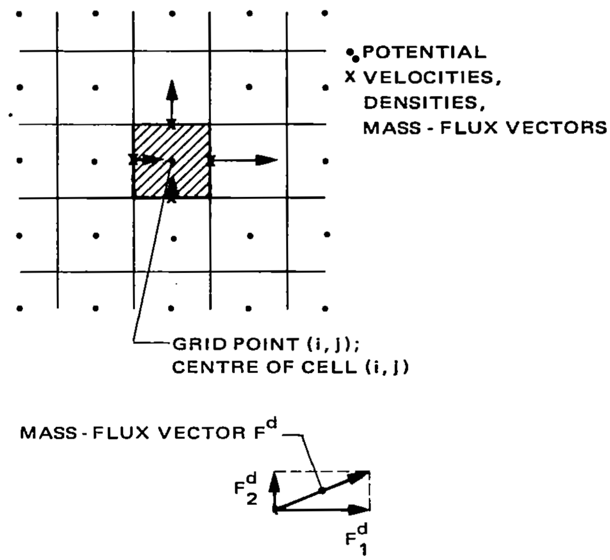


Fig. 4 Discrete divergence of mass-flux vector over a cell

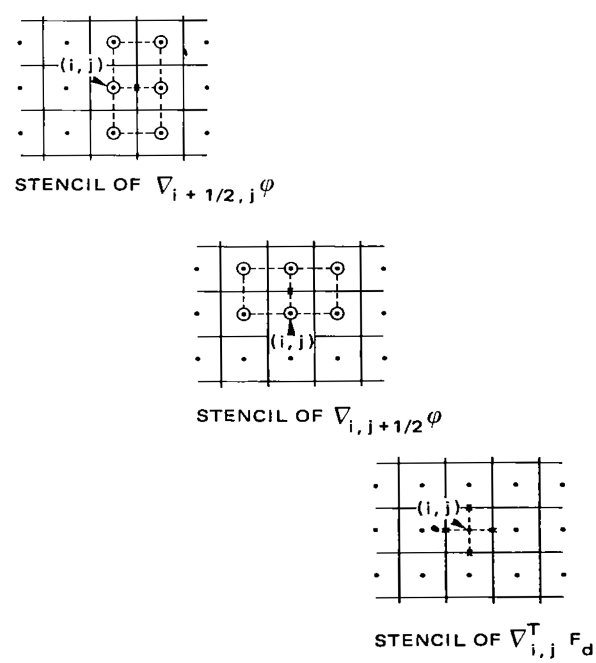


Fig. 5 Stencils of $\nabla = (\partial/\partial\xi, \partial/\partial\eta)$

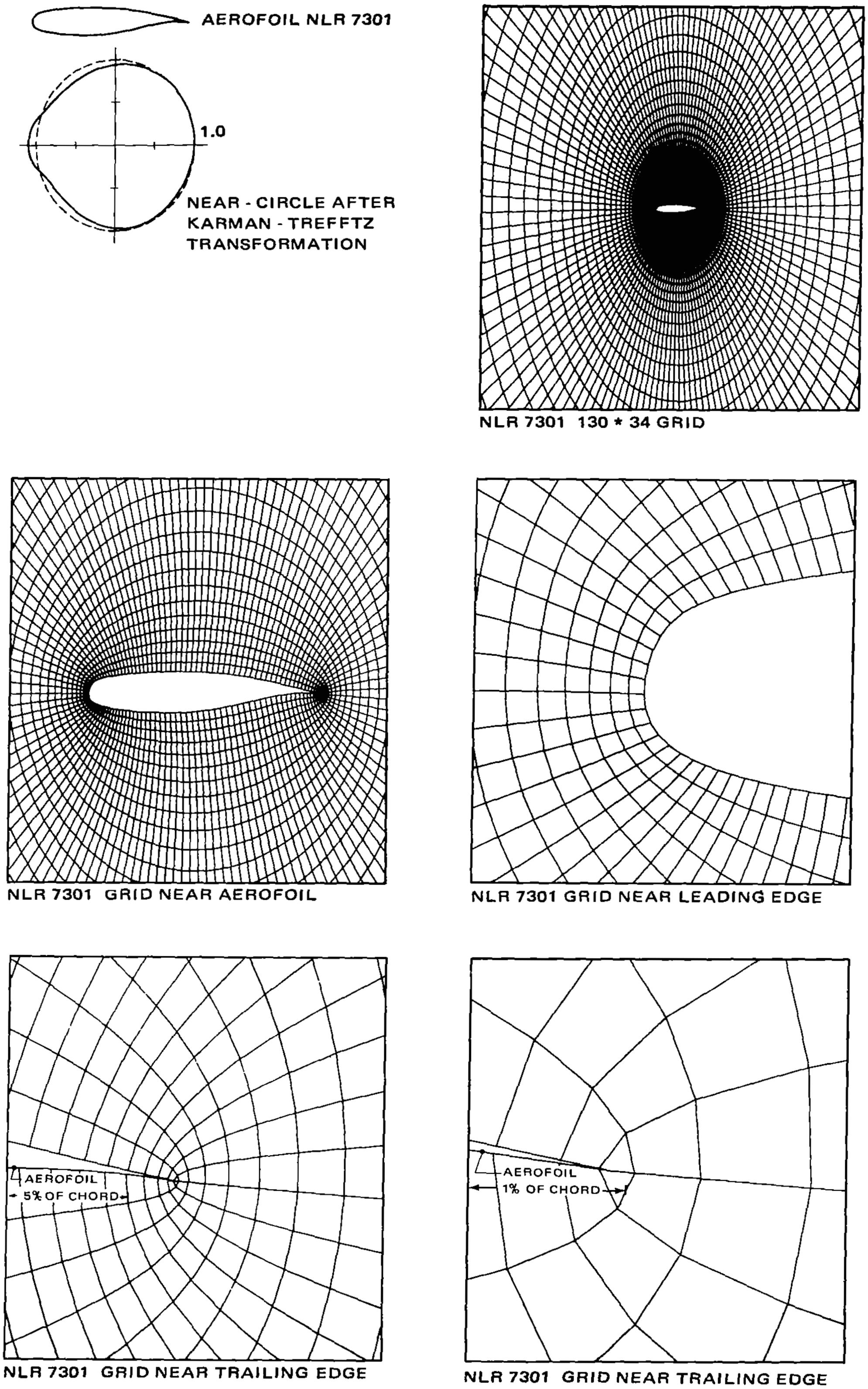


Fig. 3 Example of a grid

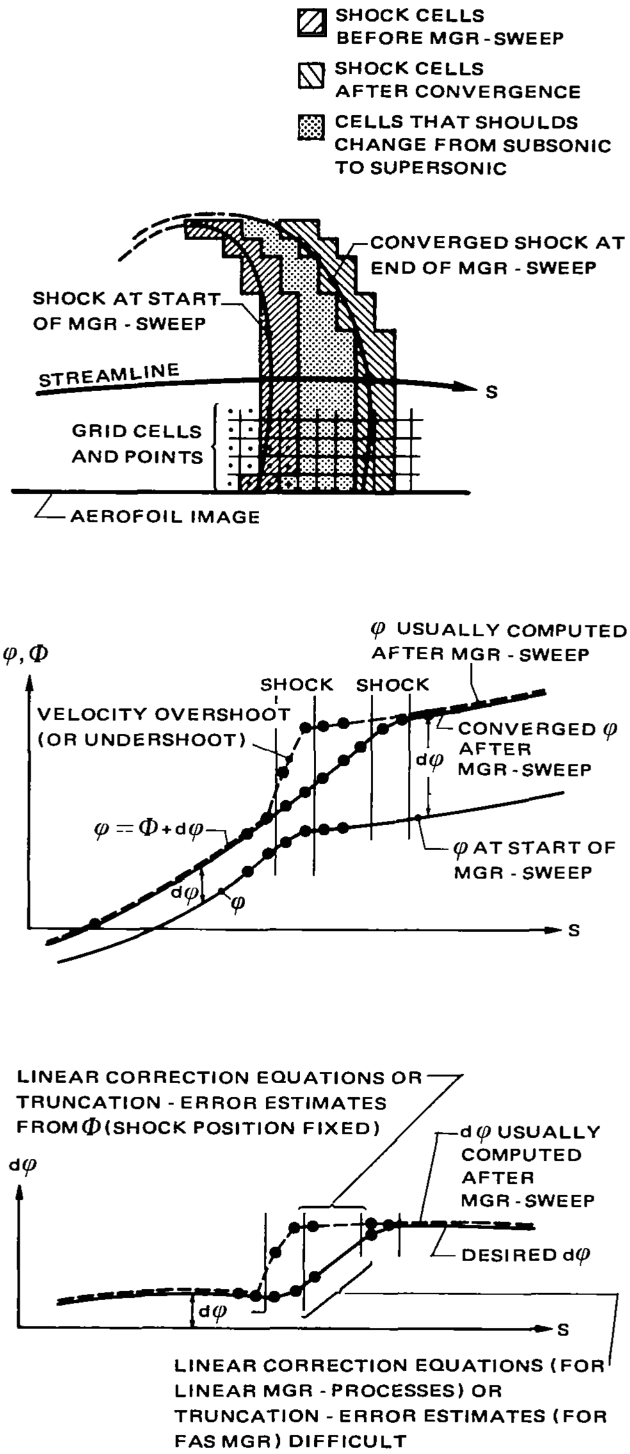


Fig. 6 Usual main effects of one MGR (multigrid relaxation)-cycle near shocks

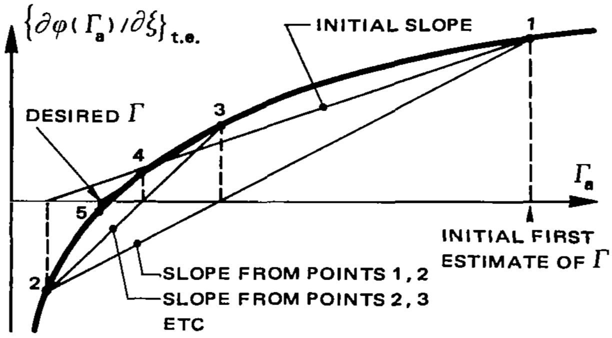


Fig. 7 Newton iteration on Γ

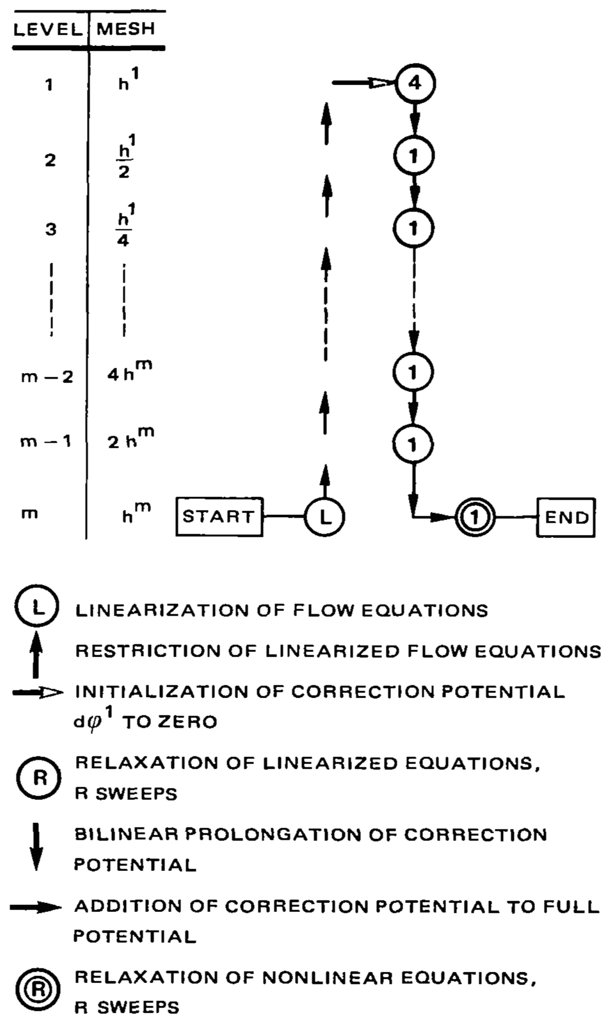


Fig. 8a Flow diagram of multigrid cycle

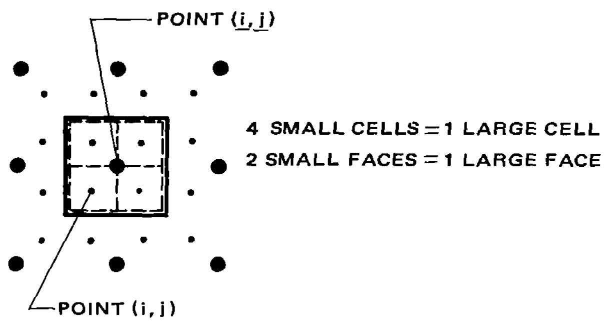


Fig. 8b Arrangement of grid points, cells, and cell faces on two successive grids

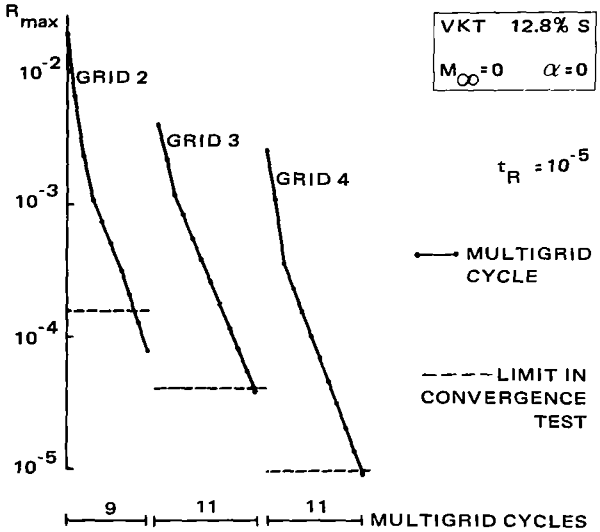


Fig. 9a Multigrid convergence, linear case, no circulation

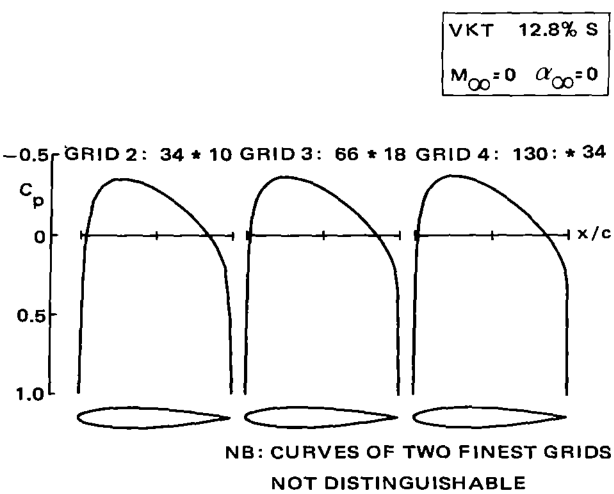


Fig. 9b Convergence of pressure distributions

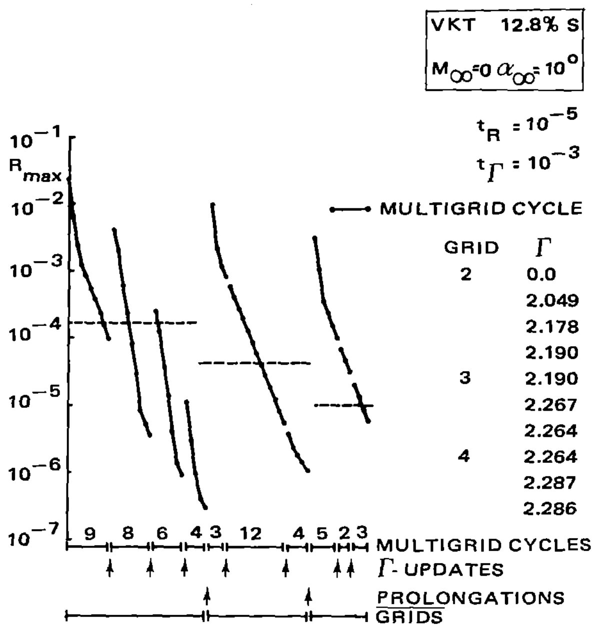


Fig. 10a Convergence analysis (linear case, with circulation)

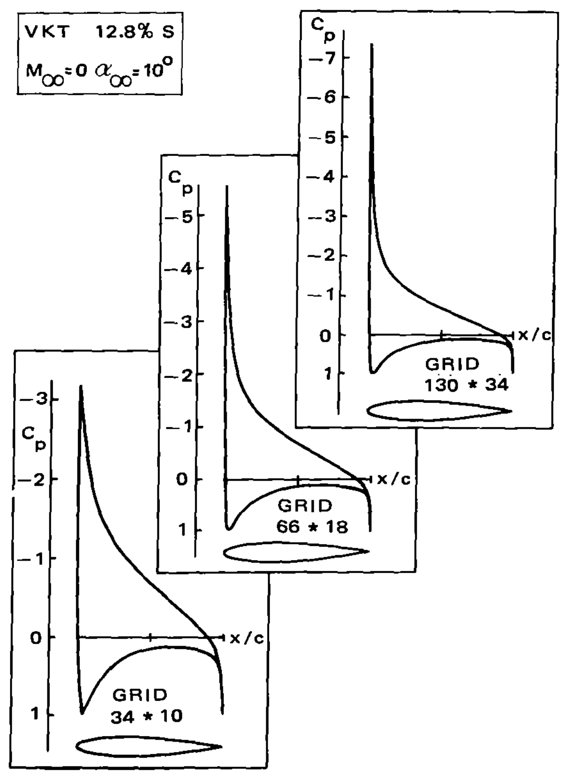


Fig. 10b Convergence of pressure distributions

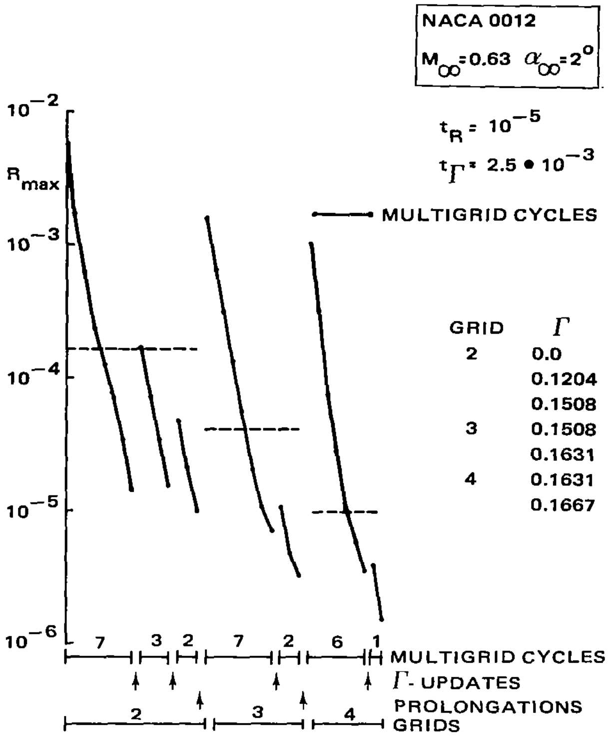


Fig. 11a Convergence analysis

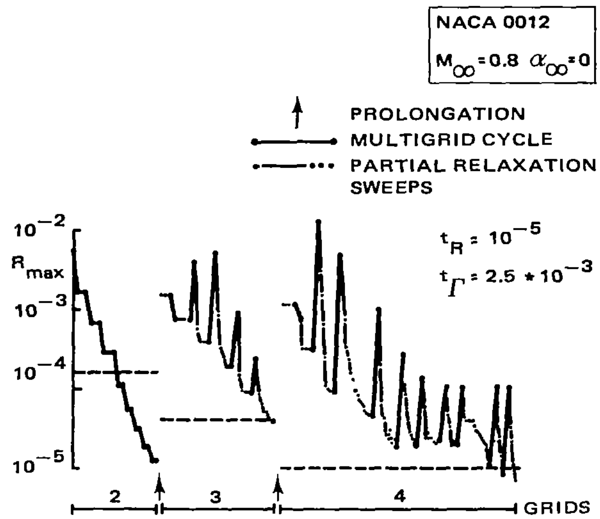


Fig. 12a Convergence analysis

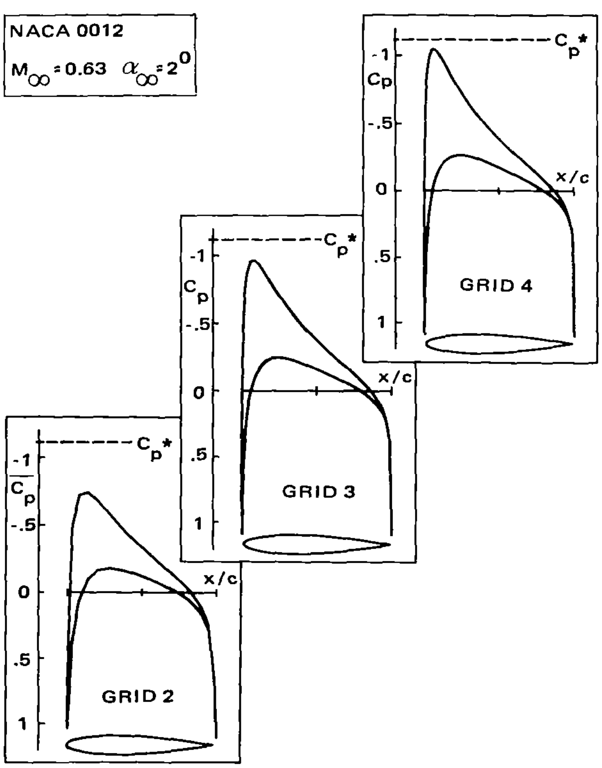


Fig. 11b Convergence of pressure distributions

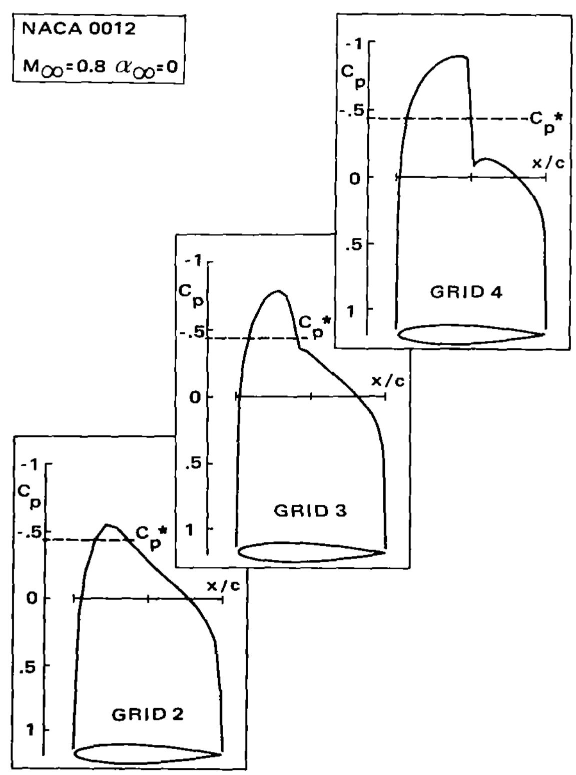


Fig. 12b Convergence of pressure distributions

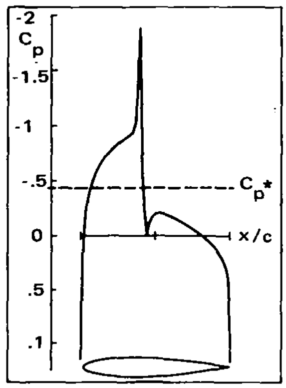


Fig. 12c Example of nonconverged pressure distribution

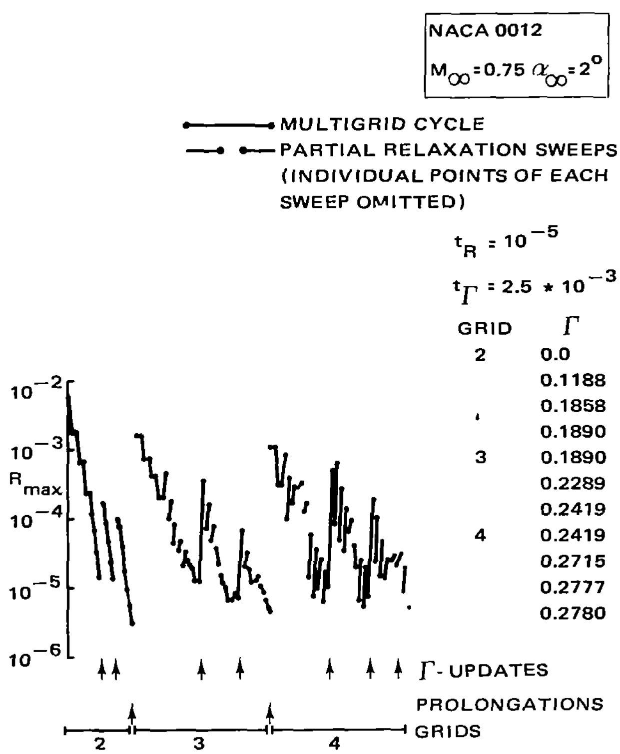


Fig. 13a Convergence analysis

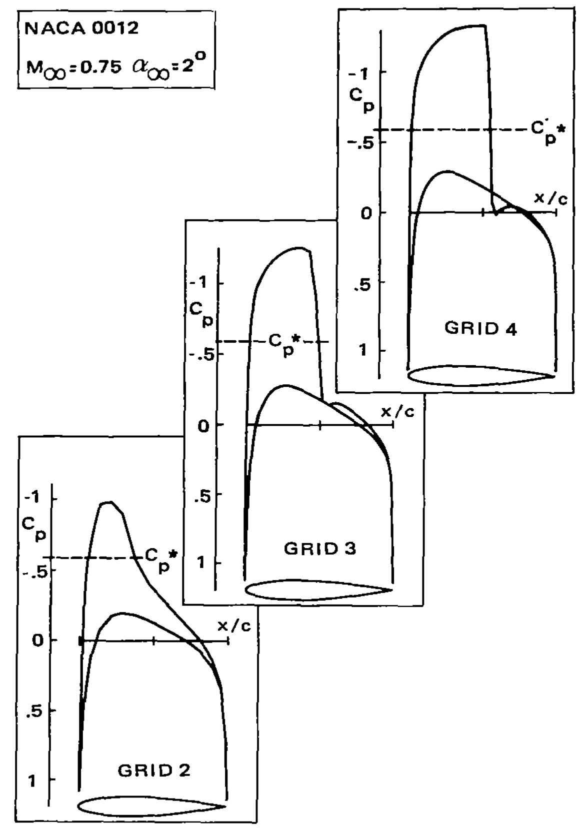


Fig. 13b Convergence of pressure distributions

SPECTRAL MULTI-GRID METHODS FOR ELLIPTIC EQUATIONS

Thomas A. Zang
College of William and Mary

Yau Shu Wong
Institute for Computer Applications in Science and Engineering

M. Y. Hussaini
Institute for Computer Applications in Science and Engineering

INTRODUCTION

Two computational approaches which achieved substantial popularity during the past decade are spectral methods and multi-grid techniques. The former have proven highly efficient for time-dependent, smooth flows in simple geometries (refs. 1-3). The latter have been remarkably successful for elliptic equations and some steady-state calculations (refs. 4-7). The principal advantage of spectral methods lies in their ability to achieve accurate results with substantially fewer grid points than required by typical finite difference methods. Despite the fact that spectral methods are represented by full matrices, explicit time-stepping algorithms can be implemented nearly as efficiently for them as for finite difference methods on a comparable grid. Transform methods (ref. 8) are often the key to this efficiency. For implicit methods or for steady-state equations, direct solution of the spectral equations is not practical in general. Iterative schemes for such equations are essential. Orszag (ref. 9) has described several attractive methods.

This paper examines an alternative approach which employs multi-grid concepts in the iterative solution of spectral equations. In particular, spectral multi-grid methods are described for self-adjoint elliptic equations with either periodic or Dirichlet boundary conditions. For realistic fluid calculations the relevant boundary conditions are likely to be periodic in at least one (angular) coordinate and Dirichlet (or Neumann) in the remaining coordinates. Spectral methods may not always be effective for flows in strictly rectangular geometries since corners generally introduce singularities into the solution. These singularities can seriously degrade the accuracy of a spectral method. If the boundary is smooth, then mapping techniques (ref. 9) can often be used to transform the problem into one with a combination of periodic and Dirichlet boundary conditions. Spectral multi-grid methods in these geometries can be devised by combining the techniques presented separately here.

Research of TAZ was supported by NASA Grant NAG1-109; research of YSW and MYH was supported by NASA Contract Nos. NAS1-15810 and NAS1-16394, respectively, while they were in residence at ICASE, NASA Langley Research Center.

SYMBOLS

A	diagonal matrix of PDE coefficients at collocation points
a	variable coefficient in PDE
B	diagonal matrix of PDE coefficients at collocation points
b	variable coefficient in PDE
C	matrix representing Fourier transform
c	constants used in describing the discrete cosine transform
D	matrix representing first derivative operator in transform space
E	matrix describing trigonometric interpolation in transform space
F	right-hand-side terms of PDE at collocation points
f	right-hand-side term of PDE
G	grid on level k
H	pre-conditioning matrix
K	finest level of the multiple grids
k	any grid (or level) k of multiple grids
L	matrix representing spectral approximation to PDE operator
$\mathcal{L}$	lower-triangular matrix
M	matrix representing first derivative operator in physical space
M	vector used for describing M
N	number of collocation points (in one coordinate direction)
n	number of distinct relaxation parameters
P	matrix representing coarse-to-fine grid interpolation
R	matrix representing fine-to-coarse grid interpolation
S	matrix representing finite difference approximation to PDE

T	Chebyshev polynomial of degree n
U	vector of solution at collocation points
$\mathcal{U}$	upper-triangular matrix
u	solution to PDE
u	Fourier transform of solution to PDE
V	vector of corrections in multi-grid scheme
v	approximate solution to V
x,y	physical space coordinates
$\delta_{j,1}$	Kronecker delta function
ε	amplitude in variable coefficient term
λ	eigenvalue
ξ	eigenvector
ρ	spectral radius
ω	relaxation parameter
μ	smoothing rate
$\bar{\mu}$	average smoothing rate

PERIODIC PROBLEMS

Fourier Spectral Approximations

Several types of spectral approximations can be employed. The specific method used here is often termed collocation or pseudo-spectral approximation. In many cases this method is easier to implement and is more efficient than the alternative Galerkin and tau approximations. A thorough discussion of all these methods can be found in reference 3.

For a periodic problem spectral approximations should be based upon Fourier series. In the collocation approach the fundamental representation of the solution remains in physical space. The Fourier coefficients are only employed as an intermediate result in the approximate evaluation of derivatives. Consider a function u(x) which is periodic over the interval

$[0,2\pi]$. Use N evenly spaced collocation points

$$x_j = 2\pi j/N \qquad j=0,1,\ldots,N-1 \tag{1}$$

and denote $u(x_j)$ by u_j . The first step in the evaluation of du/dx is the computation of the approximate Fourier coefficients $\hat{u}_p$ via

$$\hat{u}_p = (1/\sqrt{N}) \sum_{j=0}^{N-1} u_j \, e^{-ipx_j} \qquad p=-N/2,-N/2+1,\ldots,N/2-1 \, . \tag{2}$$

Since the $u(x_j)$ are real, $\hat{u}_{-N/2}$ is real and $\hat{u}_{-p} = \hat{u}_p^*$ for $|p| < N/2$ where the * denotes complex conjugation. The derivative is then computed via

$$du/dx(x_j) = (1/\sqrt{N}) \sum_{=-\frac{N}{2}+1}^{\frac{N}{2}-1} ip\hat{u}_p \, e^{ipx_j} \qquad j=0,1,\ldots,N-1 \, . \tag{3}$$

Both sums can be evaluated in $O(N \ln N)$ operations by the Fast Fourier Transform (ref. 10). This algorithm is most commonly employed with N chosen as a power of 2 .

Note that the lower limit on the sum in equation (3) is not $p = -N/2$ but $p = -N/2 + 1$. This change is equivalent to setting $\hat{u}_{-N/2} = 0$. The right-hand-side of equation (3) is necessarily real. The neglected term

$$-i \; (N/2) \; e^{-i(\frac{N}{2})x_j}$$

is purely imaginary and cannot contribute to $du/dx(x_j)$. This neglected term represents the familiar "two-point oscillation" in $u(x)$. (Finite difference schemes which use central differences for first derivatives also remove the two-point oscillation.)

The spectral evaluation of derivatives has a convenient matrix representation. Let U denote the vector of the solution at the grid, or collocation, points, i.e.,

$$U = (u_0, u_1, \; . \; . \; . \; , u_{N-1}) \quad , \tag{4}$$

let C represent the discrete Fourier transform, i.e.,

$$C_{jl} = (1/\sqrt{N}) \; e^{-2\pi il\,(j-\frac{N}{2})/N} \qquad j,l=0,1,\ldots,N-1 \tag{5}$$

and let D be the diagonal matrix which represents the first derivative in Fourier space, i.e.,

$$D_{jj} = \begin{cases} i(j - N/2) & \text{for } j = 1,2,\ldots,N-1 \\ 0 & \text{for } j = 0 . \end{cases} \tag{6}$$

Note that $C^{-1} = C^*$, the Hermitian transpose of C . Then the matrix

$$M = C^{-1}DC \tag{7}$$

represents (in physical space) the spectral evaluation of a first derivative. This matrix is given explicitly by

$$M_{jl} = \tilde{M}_{j-l} , \tag{8}$$

where

$$\tilde{M}_j = \begin{cases} 0 & j=0, \pm N, \pm 2N, \ldots \\ \cos\, (1-1/N)\pi j/(2 \sin(\pi j/N)) & \text{otherwise} . \end{cases} \tag{9}$$

A spectral approximation to the ordinary differential equation

$$(d/dx)\{a(x)\ du/dx\} = f(x) \tag{10}$$

on $[0,2\pi]$ with periodic boundary conditions, and with $a(x)$ and $f(x)$ infinitely differentiable as well as periodic, satisfies the discrete equation

$$L\ U = F , \tag{11}$$

where

$$L = MAM , \tag{12}$$

$$A_{jl} = a(x_j)\, \delta_{j,l} \tag{13}$$

and

$$F = (f_0, f_1, \ldots, f_{N-1}) . \tag{14}$$

Equation (11) may be inverted to yield

$$U = (C^{-1}D^{-1}CA^{-1}C^{-1}D^{-1}C)\ F . \tag{15}$$

Although the matrix D is technically singular, this merely reflects the usual non-uniqueness of the solution of equation (10). All of the matrix multiplies required by the right-hand-side of equation (15) may be implemented efficiently. There are three diagonal matrices and four Fourier transforms. Thus, the solution to equation (11) can be obtained directly in

$O(N \ln N)$ operations, even though the matrix L is full.

Unfortunately, efficient direct solutions are not available in higher dimensions. Consider the self-adjoint elliptic equation

$$\frac{\partial}{\partial x}\{a(x,y)\frac{\partial u}{\partial x}\} + \frac{\partial}{\partial y}\{b(x,y)\frac{\partial u}{\partial y}\} = f \tag{16}$$

on the square $[0,2\pi] \times [0,2\pi]$. Again impose periodic boundary conditions and assume that the functions a , b and f are also periodic as well as infinitely differentiable. A spectral approximation to equation (16) will exhibit exponential convergence, i.e., the error will ultimately decrease faster than any finite inverse power of the number of collocation points.

For simplicity, suppose that an $N \times N$ mesh is employed. Define the approximate solution

$$U_{jl} = u(x_j, y_l) \qquad \text{for } j,l=0,1, \ldots, N-1 . \tag{17}$$

Define F in a similar fashion and let A and B be the diagonal matrices representing $a(x,y)$ and $b(x,y)$, respectively, in the manner of equation (13). The discrete approximation to equation (16) is

$$L\, U = F , \tag{18}$$

where the fourth-order tensor

$$L = (M \otimes I)A(M \otimes I) + (I \otimes M)B(I \otimes M) , \tag{19}$$

with $\otimes$ denoting a tensor product and I representing the identity matrix of order N .

The authors are unaware of any efficient method for solving equation (18) directly. The iterative methods described in reference 9 are one possible solution scheme. A different sort of iterative method -- one involving the use of multiple grids -- is described below.

Euler Iteration on a Single Grid

The direct solution of the $N^2 \times N^2$ system represented by equation (18) would require $O(N^4)$ storage locations and $O(N^6)$ operations. Many iterative schemes require only $O(N^2)$ storage locations and $O(N^2 \ln N)$ operations per step. Perhaps the simplest iterative scheme is the Euler method

$$U \longleftarrow U - \omega (F - L\, U) , \tag{20}$$

where ω is a relaxation parameter. Aside from the coefficients $a(x_j,y_l)$ and $b(x_j,y_l)$, the only substantial storage required is for the residual [the term in parentheses in eq. (20)], which is clearly $O(N^2)$. The tensor L is never explicitly required. The residual itself costs $O(N^2 \ln N)$ operations to compute. Jacobi's method (see below) is also economical in storage and cost per step. Not all iterative schemes used to solve finite difference equations are practical for the spectral equations, however. Gauss-Seidel is an obvious example. The term $L\,U$ can only be evaluated efficiently if it is done all at once.

It is instructive to consider the application of the Euler iteration to the constant coefficient case $a(x,y) = b(x,y) = 1$. The tensor L simplifies to

$$L = M^2 \otimes I + I \otimes M^2 \,. \tag{21}$$

The eigenvalues and eigenvectors of L are

$$\lambda_{pq} = -\,(p^2 + q^2) \tag{22}$$

$$\xi_{jl}(p,q) = e^{\frac{2\pi i}{N}(pj + ql)} \,, \tag{23}$$

where the eigenvalues and eigenvectors are labelled by p and q which lie in the range $p,q = -N/2, -N/2+1, \ldots, N/2-1$. In equation (22), if either p or $q = -N/2$, then that term should be replaced by 0 on the right-hand-side. A single iteration by equation (20) replaces the error component $\xi(p,q)$ with $(1 + \omega\lambda_{pq})\ \xi(p,q)$. There are two eigenvectors which are unaffected by the iteration. One of these -- for $p = q = 0$ -- represents the mean level of the solution. It must be specified for the partial differential equation to have a unique answer. The other term -- for $p = q = -N/2$ -- represents the high-frequency component that is ignored by the discretization. This component should be filtered out of the right-hand-side F.

This scheme is convergent if

$$\omega < -2/\lambda_{\frac{N}{2}-1,\frac{N}{2}-1} = 4/(N-2)^2 \,. \tag{24}$$

The smallest spectral radius

$$\rho = (N^2 - 4N + 2)/(N^2 - 4N + 6) \cong 1 - 4/N^2 \,, \tag{25}$$

is obtained when

$$\omega = 4/(N^2 - 4N + 6) \,. \tag{26}$$

According to the usual reasoning, equation (25) implies $O(N^2)$ iterations are required. This means a total of $O(N^4 \ln N)$ operations are required in order to solve equation (18) in this fashion.

Euler Iteration Using Multiple Grids

Multi-grid methods have become a standard means of accelerating convergence for finite difference and finite element discretizations of elliptic equations. The basic processes are the relaxation scheme and the transfer of residuals and corrections between the various grids. In addition to specific choices of relaxation and interpolation procedures, a multi-grid algorithm must give rules governing the transfer between grids. A variety of control structures for this latter process have been employed. For examples of some of the control structures, see the flow charts in reference 5. The present discussion will focus on the relaxation and interpolation procedures, since they are less arbitrary than the control structure. Moreover, the description will be given for the spectral discretization of the one-dimensional problem [eqs. (11) and (12)]. This is done simply for notational convenience. The performance will be assessed, and numerical examples given, however, for the two-dimensional case.

Define a series of grids (or levels) G_k, for $k = 2, 3, \ldots, K$ covering the interval $[0,2\pi]$. Let G_k consist of N_k uniformly spaced points, where $N_k = 2^k$. The solution to equation (11) is obtained by combining Euler iterations on level K with Euler iterations for related problems on the coarser levels $k < K$. Denote the relevant discrete problem at any level k by

$$L_k V_k = F_k . \tag{27}$$

On the finest level K, $L_K = L$, $F_K = F$ and the solution $V_K = U$, the solution to equation (11). At any stage in the iterative solution process for equation (27), only an approximation v_k to the exact answer V_k is available. If this approximation is deemed adequate, then the approximation on the next-finer level $k+1$ is corrected via

$$v_{k+1} \longleftarrow v_{k+1} + P_{k+1} v_k . \tag{28}$$

The matrix P_k represents the coarse-to-fine transfer of corrections from level $k-1$ to level k. On the other hand, if the approximation v_k is deemed inadequate, either another relaxation is performed, via

$$v_k \longleftarrow v_k - \omega_k (F_k - L_k v_k) , \tag{29}$$

or else control shifts to a problem on the next-coarser level $k-1$. The relaxation parameter ω_k on level k is chosen to damp preferentially those error components which are not represented on coarser grids. The right-hand-side of the coarser grid problem is obtained from

$$F_{k-1} = R_k (F_k - L_k v_k) . \tag{30}$$

The matrix R_k represents the fine-to-coarse residual transfer from level k to level $k-1$.

For the spectral multi-grid method the natural interpolation operators represent trigonometric rather than polynomial interpolation. For the one-dimensional case,

$$R_k = C_{k-1}^{-1} E_{k-1} C_k \ , \tag{31}$$

$$P_k = C_k^{-1} E_{k-1}^T C_{k-1} \ , \tag{32}$$

where the $N_k \times N_{k+1}$ matrix

$$E_k = (0 \mid I_k \mid 0) \tag{33}$$

(with I_k the identity matrix of order N_k), E_k^T is its transpose, and C_k is the matrix given in equation (5) for $N = N_k$. The matrix E_k represents the dropping of the high-frequency Fourier coefficients in the trigonometric interpolation from the fine grid to the coarse grid. Note that $P_k = R_k^*$. The generalization to higher dimensions is straightforward.

For the constant coefficient, one-dimensional case, the finest grid relaxation operator

$$L_K = C_K^{-1} D^2 C_K \ , \tag{34}$$

and it is natural to use

$$L_k = C_k^{-1} D_k^2 C_k \tag{35}$$

for $k < K$. It is easy to show that

$$L_{k-1} = R_k L_k P_k \ . \tag{36}$$

The description of the variable coefficient relaxation operator is more complicated and the details will be published elsewhere. The procedure used in the numerical experiments reported below amounts to performing the collocation operations in an alias-free fashion.

For the two-dimensional Poisson equation discussed in the previous section the level k relaxation parameter ω_k is chosen to maximize the smoothing of all the modes except those for which $|p|, |q| < N_k/4$:

$$\omega_k = 2/((9/16)N_k^2 - 2N_k + 2) \ . \tag{37}$$

This choice produces a smoothing rate for the high-frequency modes of

$$\mu_k = 1 - 2N_k^2/(9N_k^2 - 32N_k + 32) \ . \tag{38}$$

This smoothing rate is listed in Table 1 alongside the spectral radius for the single grid Euler method. The advantage of multiple-gridding is apparent. For large N_k , $\mu_k \cong 7/9$. Thus, according to the usual multi-grid argument, the number of iterations needed to obtain a given reduction in the residual should be independent of the number of grid points on the finest grid. This assumes, of course, no untoward effects of the interpolation process. But the trigonometric interpolation procedure used here is ideally suited to minimize the spurious generation of high-frequency components at these stages.

TABLE 1. Convergence Rates for Euler Iteration in Two-dimensions

N	single grid spectral radius	multi-grid smoothing rate
4	0.3333	0.3333
8	0.8947	0.6364
16	0.9798	0.7193
32	0.9956	0.7510
64	0.9990	0.7649
∞	1.0000	0.7778

Alternatives to Euler Relaxation

A straightforward improvement upon the simple relaxation scheme described in the preceding sub-section is to make it non-stationary. This approach has been used for accelerating point-Jacobi iterations for finite difference multi-grid algorithms (see ref. 11). The non-stationary Euler iteration consists of using n relaxation parameters $\omega_{k,1}, \omega_{k,2}, \cdots , \omega_{k,n}$ in a cyclic fashion on each level k . These parameters are determined from the solution of a standard minimax problem over the interval covered by the high-frequency eigenvalues.

For the two-dimensional Poisson equation, this eigenvalue range is from $-(N_k/4)^2$ to $-(N_k/2-1)^2$. The results are only changed slightly if the upper limit of this range is changed to $-(N_k/2)^2$. Then the optimal parameters are given by

$$\omega_{k,j} = (32/N_k^2)/(7 \cos(j-1/2)\pi/n + 9) \tag{39}$$

and the total smoothing of the high-frequencies after the full n relaxations is $1/|T_n(-9/7)|$, where $T_n(x)$ is the Chebyshev polynomial of degree n . Then the effective smoothing rate

$$\bar{\mu}_k = 1/|T_n(-9/7)|^{1/n} , \tag{40}$$

which is the average smoothing per single step in the cyclic relaxation. The values are given in Table 2 along with the corresponding effective smoothing rates for a finite difference multi-grid method which also is relaxed with Euler iteration. The spectral smoothing rates are larger than the finite difference ones because the ratio of the largest high-frequency eigenvalue to the smallest high-frequency eigenvalue is 8 in the former case and only 4 in the latter. This ratio may be termed the multi-grid condition number. The higher smoothing rate for the spectral method suggests that a larger number of distinct relaxation parameters should be used here than for the finite difference case.

TABLE 2. Smoothing Rates for Euler Iteration on Poisson's Equation

number of parameters	spectral smoothing rate	finite difference smoothing rate
1	0.7778	0.6000
2	0.6585	0.4685
3	0.5995	0.4198
4	0.5676	0.3964
5	0.5485	0.3749

It should be kept in mind that this larger eigenvalue ratio for the spectral method occurs because this method represents the larger eigenvalues of the partial differential equation much better than finite difference methods. Indeed, it is just this property which is responsible for the exponential convergence rate of spectral methods as N is increased and for their low phase-error in time-dependent calculations.

Another obvious relaxation scheme is point-Jacobi. The actual implementation of this method requires that the diagonal elements of the matrix L be known explicitly. Consider the one-dimensional situation, where L is given be equation (12) for the general case. It would appear that the evaluation of the elements L_{jj} requires $O(N^2)$ operations. This would be impractical since the results of the previous section suggested that only O(N ln N) operations are needed to get the solution itself.

Nonetheless, Jacobi relaxation is worth considering since transform methods may be employed to compute the requisite diagonal elements in only O(N ln N) operations. It is clear from equation (9) that $\tilde{M}_j$ is odd in j. Thus,

$$L_{jj} = - \sum_{l=0}^{N-1} \tilde{M}_{j-l} a_l \tag{41}$$

But this is a convolution sum and may be evaluated efficiently by the transform methods described in reference 8. Therefore, even for non-linear problems, Jacobi relaxation may be implemented efficiently.

Numerical Examples

The spectral multi-grid method was implemented for the two-dimensional problem [eq. (16)] for which the coefficients

$$a(x,y) = b(x,y) = 1 + \varepsilon\, e^{\cos(x+y)} \tag{42}$$

and the solution itself

$$u(x,y) = \sin(\pi \cos x + \pi/4) \sin(\pi \cos y + \pi/4) \; . \tag{43}$$

The Fourier coefficients of this function may be expressed in terms of Bessel functions. Reference 3 (pp. 35-37) uses this function to illustrate exponential convergence. The term $\pi/4$ serves to make all the Fourier coefficients non-zero. The constant ε in equation (42) measures the departure of the equation from the strictly Poisson form.

A simple control structure was selected for the multi-grid algorithm: start on the finest level and relax once on each level in turn until the coarsest level $k=2$; there iterate until convergence; then work back up to the finest level, relaxing once more on each intermediate level. This process is repeated until the desired accuracy is achieved. This algorithm requires more frequent interpolation but is less arbitrary than many alternatives. Despite the necessity for employing the Fast Fourier Transform in the trigonometric interpolations, this portion of the computations takes less than 10% of the total computation time.

TABLE 3. RMS Residuals for Fourier Spectral Multi-grid Using Stationary Euler Iteration

relaxation number	ε = 0.0	ε = 0.1	ε = 0.2
3	2.92 (1)	3.23 (0)	3.72 (0)
6	2.27 (-1)	2.49 (-1)	3.12 (-1)
9	3.24 (-2)	3.52 (-2)	4.40 (-2)
12	1.02 (-2)	1.11 (-2)	1.37 (-2)
15	4.00 (-3)	4.37 (-3)	5.55 (-3)

The results of calculations for which the finest level $K = 5$ are shown in Tables 3 and 4. The non-stationary Euler iteration used 3 distinct parameters. The transfer between grids does not occur until all 3 relaxations have been performed. The residuals are listed in the tables after every 3 relaxations on the finest grid. The number in parentheses is the exponent of the residual. For comparison purposes note that Euler iteration on a single grid exhibits a residual of about 10 after 15 relaxations. The multi-grid results are a marked improvement.

TABLE 4. RMS Residuals for Fourier Spectral Multi-grid Using Non-stationary Euler Iteration

relaxation number	$\varepsilon = 0.0$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
3	2.82 (1)	3.12 (1)	3.47 (1)
6	2.42 (-1)	2.10 (-1)	2.56 (-1)
9	6.35 (-3)	3.68 (-3)	5.57 (-3)
12	4.56 (-4)	3.19 (-4)	6.30 (-4)
15	8.30 (-5)	5.36 (-5)	1.17 (-5)

On a 32 x 32 grid the true solution of the Fourier collocation equation (18) has an RMS error of 5.08 (-10) compared with the exact solution of equation (43) for $\varepsilon = 0.0$. The RMS error of the non-stationary iteration after 15 fine-grid relaxations is 2.20 (-7). To get the full accuracy out of a spectral method it may be necessary to reduce the residual by many orders of magnitude. By contrast a second-order finite difference approximation on a 32 x 32 grid gives an RMS error of 7.64 (-2) for the $\varepsilon = 0.0$ problem. Even a fourth-order method gives only 5.04 (-3). For this problem, at least, it seems worthwhile to accept the less advantageous smoothing rate of the spectral multi-grid method (see Table 2), since a far smaller grid can be used than for a finite difference method.

DIRICHLET PROBLEMS

Chebyshev Spectral Approximations

For problems with Dirichlet (or Neumann) boundary conditions, spectral approximations should be based upon Chebyshev series. The standard interval is [-1,1] and the collocation points are

$$x_j = \cos(2\pi j/N) \qquad j=0,1, \ldots ,N . \tag{44}$$

The analog to equation (7) with Dirichlet boundary conditions may be written in the form of equations (11)-(14) where now

$$C_{jl} = (2/N\bar{c}_j\bar{c}_l) \cos(\pi jl/N) \qquad j,l=0,1, \ldots , N , \tag{45}$$

$$\bar{c}_j = \begin{cases} 2 & j=0 \text{ or } j=N \\ 1 & 1\leq j \leq N \ , \end{cases} \tag{46}$$

$$D_{jl} = \begin{cases} 2l/c_j & l \geq j+1 \text{ and } l \equiv j+1 \pmod 2 \\ 0 & \text{otherwise} \ , \end{cases} \tag{47}$$

and

$$c_j = \begin{cases} 2 & j=0 \\ 1 & j\geq 1 \ . \end{cases} \tag{48}$$

Reference 3 is a good source for many details about Chebyshev collocation. The matrix M which represents the Chebyshev approximation to a first derivative is again given by equation (7) where now

$$\begin{aligned} M_{jl} &= \bar{c}_j(\tilde{M}_{j+l} + \tilde{M}_{j-l})/(\bar{c}_l \sin(\pi j/N)) \quad \text{for} \quad 1\leq j \leq N-1 \\ M_{00} &= -M_{NN} = (2N^2 + 1)/6 \\ M_{0l} &= -M_{Nl} = 2(-1)^l/(1 - \cos(\pi l/N)) \quad \text{for} \quad 1 \leq l \leq N-1, \end{aligned} \tag{49}$$

where

$$\tilde{M}_j = \begin{cases} 0 & j=0, \pm 2N, \pm 4N, \ . \ . \ . \\ (1/2)(-1)^{j+1} \cot(\pi j/N) & \text{otherwise} \ . \end{cases} \tag{50}$$

Once more M is a full matrix but the product M U can be evaluated in O(N ln N) operations.

Pre-conditioned Euler Iteration Using Multiple Grids

The direct analog of the Euler iteration method described in the preceding section is not practical for the Dirichlet problem. The difficulty is that for the Chebyshev second derivative operator the multi-grid condition number grows as N^2. In the one-dimensional case Gershgorin's Theorem can be used to show that the largest eigenvalue grows as N^4 (ref. 3). All but the several largest eigenvalues are good approximations to the eigenvalues of the continuous problem. Thus, the smallest high-frequency eigenvalue grows as N^2. (Direct numerical computation of the eigenvalues supports these conclusions.) Since the ratio of these two eigenvalues (the multi-grid condition number) is N^2, the smoothing rate of a straightforward Chebyshev Euler multi-grid method is of the same order as the spectral radius of the Fourier Euler iteration on a single grid (see Table 1). The non-stationary Chebyshev Euler multi-grid method has the same problem.

Clearly, pre-conditioning is essential for an effective Chebyshev spectral multi-grid algorithm based on Euler iteration. Thus, in place of equation (20) the relaxation scheme is

$$U \longleftarrow U - \omega H^{-1}(F - L U) , \qquad (51)$$

where the pre-conditioning matrix is denoted by H. An effective pre-conditioning matrix has been devised by Orszag (ref. 9) for finding solutions iteratively on a single grid to Chebyshev spectral approximations. That pre-conditioning matrix, denoted here by S, is a full finite difference approximation to the spectral matrix L. Orszag noted that the conventional condition number of the matrix $S^{-1}L$ should be about 2.4 regardless of N.

The pre-conditioning matrix employed in the present spectral multi-grid calculations is a cheaper but less precise version of S. Instead of using S itself an approximate lower-triangular/upper-triangular decomposition of S is used as H, i.e.,

$$H = \mathcal{L}\mathcal{U} , \qquad (52)$$

where script letters are used to denote the lower-triangular ($\mathcal{L}$) and upper-triangular ($\mathcal{U}$) factors. This matrix H is cheaper to employ than S because H^{-1} can be found by simple forward- and back-substitutions, whereas finding S^{-1} amounts to computing the solution to a finite difference discretization of the problem.

To determine H one starts with S as a standard finite difference approximation to equation (16) on the non-uniform grid of the Chebyshev collocation points. The matrices $\mathcal{L}$ and $\mathcal{U}$ are determined by the row sum agreement factorization which enforces the following conditions:

(1) $\mathcal{L}$ and $\mathcal{U}$ have non-zero elements only on those positions which correspond to the non-zero elements in the lower- and upper-triangular part of S itself.

(2) Whenever $S_{jl} \neq 0$ and $j \neq l$, then $H_{jl} = S_{jl}$. (The off-diagonal elements of H whose locations correspond to the non-zero off-diagonal elements of S are set to those values.)

(3) The row sums of H are the same as those of S.

For further details on this sort of pre-conditioning see reference 12.

TABLE 5. Extreme Eigenvalues of the Pre-conditioned Matrices

N	$S^{-1}L$ smallest	$S^{-1}L$ largest	$H^{-1}L$ smallest	$H^{-1}L$ largest
4	1.000	1.757	1.037	1.781
8	1.000	2.131	1.061	2.877
16	1.000	2.305	1.043	4.241
32	1.000	2.363	1.034	5.379

The decreased accuracy of the matrix H is indicated in Table 5, which lists the smallest and largest eigenvalues of the pre-conditioned matrix $H^{-1}L$. In contrast to the matrix $S^{-1}L$, for which the largest eigenvalue is roughly 2.4, the largest eigenvalue here shows a slow growth with N, evidently increasing as $\sqrt{N}$. Both matrices yield essentially the same value for the smallest eigenvalue. Moreover, the smallest high-frequency eigenvalue of $H^{-1}L$ stays roughly constant -- at about 1.45 -- as N increases. Thus, the multi-grid condition number of this pre-conditioned Euler method increases slowly with N.

The eigenvalue results given above suggest that an Euler iteration scheme using the approximate Lu factorization form of pre-conditioning will have the convergence rates listed in Table 6. The advantage of using multiple grids here is not as great as in the periodic case. The basic problem is the slow growth of the multi-grid condition number with N. Clearly, better forms of pre-conditioning are needed.

TABLE 6. Convergence Rates for Euler Iteration in Two-dimensions

N	single grid spectral radius	multi-grid smoothing rate
4	0.264	0.264
8	0.462	0.330
16	0.605	0.490
32	0.725	0.630

The interpolation for this multi-grid schemd can be based upon the Chebyshev polynomial expansions of the solution. Expressions analogous to equations (31) to (33) can be employed, where equation (45) is now used for the matrix C and the expression for the matrix E is altered accordingly. If the boundary conditions are homogeneous, then C can easily be manipulated into a self-adjoint form.

Non-stationary Euler iteration will, of course, improve the multi-grid smoothing rates. The use of 4 distinct parameters reduces the smoothing rates of the $N = 16$ and $N = 32$ cases to 0.30 and 0.40, respectively.

Point-Jacobi is a viable alternative here as well. The present form of the matrix [eq. (49)] also permits the diagonal elements of variable coefficient (or non-linear) problems to be computed efficiently by transform methods. Two convolution sums now appear in the analog of equation (41). The portion involving $\tilde{M}_{j-1}$ can be evaluated in the usual manner after allowing for special treatment of the terms for which $j = 0$ and $j = N$. The portion involving $\tilde{M}_{j+1}$ appears in transform space as the product of the transform of $\tilde{M}_j$ and the complex conjugate of the transform of the variable coefficient term $\tilde{a}_1$.

Numerical Example

The test problem for the Chebyshev multi-grid method has the coefficients

$$a(x,y) = b(x,y) = 1 + \varepsilon (x^2 + y^2) \tag{53}$$

for the exact solution

$$u(x,y) = \sin(\pi \cos x) \sin(\pi \cos y) . \tag{54}$$

Some of the results using the finest level $K = 5$ are listed in Table 7. On a single grid the residual for the $\varepsilon = 0.0$ case is 8.39 (-1) after 15 relaxations. The exact solution to the discrete equations for this case has an error that is essentially round-off error. There is relatively little content in the high-frequency component. The multi-grid approach to this problem makes its biggest gains by using the coarser grids to damp out the low-frequency components.

TABLE 7. RMS Residuals for Chebyshev Spectral Multi-grid Using Stationary Euler Iteration

relaxation number	$\varepsilon = 0.0$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
3	1.25 (0)	1.29 (0)	1.32 (0)
6	2.14 (-1)	1.89 (-1)	1.67 (-1)
9	4.68 (-2)	3.81 (-2)	3.16 (-2)
12	1.18 (-2)	9.14 (-3)	7.34 (-3)
15	3.32 (-3)	2.47 (-3)	1.93 (-3)

An example similar to equation (54) was examined in reference 13, where two schemes were given for solving the constant coefficient Chebyshev equations exactly. The results of a recent note (ref. 14) suggest that greater accuracy can be achieved, especially on problems with singularities, by sub-dividing the original domain and patching the individual Chebyshev spectral solutions together along the internal boundaries. The spectral multi-grid method can be applied to patched collocation approximations as well. Moreover, the multi-grid approach would appear to present a noticeable improvement over the admittedly inefficient schemes used in reference 14.

CONCLUSION

The spectral multi-grid methods described here exhibited a substantial improvement over the simplest iterative schemes. It has not yet been checked whether this specific algorithm is more efficient than the best available iterative methods. There, of course, is still room for improvement in the spectral multi-grid methods. This is especially true for the Chebyshev methods, for which better pre-conditioning procedures would help considerably.

It is technically straight-forward to extend this solution technique to two-dimensional incompressible Navier-Stokes equations, particularly in the vorticity-streamfunction formulation, since the problem addressed in this paper is representative of the advection-diffusion equation. Present efforts are directed towards using the spectral multi-grid method to compute the classical problem of flow past a circular cylinder. The appropriate method for this geometry combines a Fourier approximation in angle and a Chebyshev approximation in radius.

REFERENCES

1. Orszag, Steven A.: Numerical Simulation of Incompressible Flows Within Simple Boundaries: Accuracy, J. Fluid. Mech., vol. 49, pt. 1, September 13, 1971, pp. 75-112.

2.. Orszag, Steven A. ; and Israeli, Moshe: Numerical Simulation of Viscous Incompressible Flows, Ann. Rev. of Fluid Mechanics, vol. 6, 1974, pp. 281-318.

3. Gottlieb, David; and Orszag, Steven A.: Numerical Analysis of Spectral Methods: Theory and Applications, CBMS-NSF Regional Confererrnce Series in Applied Mathematics, Society for Industrial and Applied Mathematics, 1977.

4. Fedorenko, R. P.: A Relaxation Method for Solving Elliptic Difference Equations, Z. Vycisl. Mat. i Mat. Fiz., vol. 1- 1961, pp.922-927.

5. Brandt, Achi: Multi-Level Adaptive Solutions to Boundary-Value Problems, Math. Comp., vol. 31, no. 138, April 1977, pp. 333-390.

6. Nicolaides, R. A.: On Multiple Grid and Related Techniques for Solving Discrete Elliptic Systems, J. Comp. Phys., vol. 19, no. 4, December 1975, pp. 418-431.

7. Hackbush, W.: Ein Iteratives Verfahren zur Schnellen Auflosung Elliptischer Randwertprobleme, Math. Inst., Universitat zu Koln, Report 76-12, November, 1976.

8. Orszag, Steven A.: Numerical Simulation of Incompressible Flows Within Simple Boundaries: Galerkin (Spectral) Representation, Studies in Appl. Math., vol. 50, no. 4, December 1971, pp. 293-327.

9. Orszag, Steven A.: Spectral Methods for Problems in Complex Geometries, J. Comp. Phys., vol. 37, no. 1, August 1980, pp. 70-92.

10. Cooley, J. W.; and Tukey, J. W.: An Algorithm for the Machine Computation of Complex Fourier Series., Math. Comp., vol. 19, no. 90, April 1965, pp 297-301.

11. Lomax, Harvard: Some Prospects for the Future of Computational Fluid Dynamics, AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, CA, June 22-23, 1981 (Paper No. 81-0994-CP).

12. Wong, Y. S.: Conjugate Gradient Type Methods for Unsyummetric Matrix Problems, Inst. of Appl. Math. & Stat., Univ. of British Columbia, Technical Report No. 79-36, 1979.

13. Haidvogel, Dale B.; and Zang, Thomas: The Accurate Solution of Poisson's Equation by Expansion in Chebyshev Polynomials, J. Comp. Phys., vol. 30, no. 2, February, 1979, pp. 167-180.

14. McKerrell, A.; Phillips, C.; and Delves, L. M.: Chebyshev Expansion Methods for the Solution of Elliptic Partial Differential Equations, J. Comp. Phys., vol. 41, no. 2, June 1981, pp. 444-452.

APPLICATION OF MULTIGRID METHODS FOR INTEGRAL EQUATIONS TO TWO PROBLEMS FROM FLUID DYNAMICS.

H. Schippers,
Mathematisch Centrum, 1009 AB Amsterdam.

INTRODUCTION

Multigrid methods have been advocated by Brandt (ref.1) for solving sparse systems of equations that arise from discretization of partial differential equations. Convergence and computational complexity of such multigrid techniques have been studied since. In reference 2 we have shown that these techniques can also be used advantageously for the non-sparse systems that occur in the numerical solution of Fredholm integral equations of the second kind

(1) $f = Kf + g,$

where g belongs to a Banach space X and the integral operator K is compact on X. Theoretical and numerical investigations show that multigrid methods give the solution of (1) in $\mathcal{O}(N^2)$ operations as $N \to \infty$, whereas other iterative schemes take $\mathcal{O}(N^2 \log N)$ operations (N: the dimension of the finest grid). In practice this results in algorithms for the solution of these integral equations that are significantly more efficient than the other schemes. In the present paper we apply multigrid methods to the following problems from fluid dynamics.

Calculation of potential flow around bodies - The total velocity potential ϕ is assumed to be the superposition of the potential ϕ_∞ , due to a uniform onset flow and a perturbation potential ϕ_d, due to a doublet distribution at the body surface. This approach leads to a Fredholm equation of the second kind for the unknown doublet distribution. We introduce a multigrid method which makes use of a sequence of grids, that are generated by dividing the body surface into an increasing number of smaller and smaller panels. On these grids the doublet distribution is assumed to be constant over each panel. For a two-dimensional (2-D) aerofoil we have applied the multigrid method to the calculation of circulatory flow around Kārmān-Trefftz aerofoils. The use of multigrid techniques becomes more preferable for 3-D problems because the number of panels is much larger than for 2-D ones. The calculations have been performed for the flow around an ellipsoid. From numerical investigations it follows that ± 3 multigrid cycles are sufficient to obtain the approximate solution.

Calculation of oscillating disk flow - This application deals with the rotating flow due to an oscillating disk at an angular velocity $\Omega \sin \omega\tau$. The Navier-Stokes and continuity equations are reduced by means of the von Kārmān similarity transformations to

$$\frac{\omega}{\Omega} f_t = \frac{\Omega}{2\omega} f_{zz} + 2hf_z - f^2 + g^2, \tag{2}$$

$$\frac{\omega}{\Omega} g_t = \frac{\Omega}{2\omega} g_{zz} + 2hg_z - 2fg, \tag{3}$$

$$h_z = f, \tag{4}$$

where (f,g,h) is a measure of the velocity vector in a cylindrical polar coordinate system (r,ϕ,z). For a single disk problem the boundary conditions are:

$$f = h = 0,\ g = \sin t \text{ at } z = 0;\ f = g = 0 \text{ for } z \to \infty. \tag{5}$$

In reference 3 the author has shown that the periodic solution:

$$h(z,0) = h(z,2\pi);\ g(z,0) = g(z,2\pi) \tag{6}$$

can be obtained by implicit finite difference schemes taking the state of rest as an initial condition. The transient effects have been eliminated by calculating a sufficiently large number of periods. Using the multigrid method we do not simulate the physical process, but reformulate the problem (2)-(6) as

$$(f,g,h) = K(f,g,h), \tag{7}$$

where K is a non-linear integral operator. The multigrid method for integral equations is used to solve (7). For $\Omega = 0.1\ \omega$ the computational work has been reduced by a factor 0.1.

The present paper is based on parts of the Doctor's Thesis of the author prepared under the guidance of Prof. P. Wesseling of Delft University of Technology.

CALCULATION OF POTENTIAL FLOW AROUND BODIES.

For potential flow around a two- or three-dimensional body there exists a velocity potential ϕ satisfying Laplace's equation

(8) $$\Delta \phi = 0$$

with boundary conditions,

(9) $$\frac{\partial \phi}{\partial n_e} = 0 \text{ along the boundary } S,$$

where $\frac{\partial}{\partial n_e}$ denotes differentiation in the direction of the outward normal to S and

(10) $$\phi(\zeta) \to \phi_\infty(\zeta) \quad \text{for } |\zeta| \to \infty ,$$

with ϕ_∞ the velocity potential due to a uniform onset flow. If the flow is non-circulatory, we have $\phi_\infty(\zeta) = U\cdot\zeta$, with U the velocity vector of the undisturbed flow. Here $U\cdot\zeta$ denotes the usual innerproduct in $\mathbb{R}^2$ or in $\mathbb{R}^3$. We represent the velocity potential ϕ as follows

$$\phi(\zeta) = \phi_\infty(\zeta) + \phi_d(\zeta),$$

with ϕ_d the double layer potential given by

(11) $$\phi_d(\zeta) = \frac{2^{1-m}}{\pi} \int_S \mu(z) \frac{\cos(n_z, z-\zeta)}{|z-\zeta|^{m-1}} dS_z \ , \ \zeta \notin S ,$$

where m = 2,3 for the two- and three-dimensional case, respectively and n_z the outward normal to the boundary S at the point z. The doublet distribution μ is such that ϕ satisfies the boundary condition

(12) $$\phi^-(\zeta) = 0,$$

where ϕ^- denotes the limit from the inner side to S. Using the Plemelj-Privalov formulae (see reference 4) we obtain the following integral equation

$$\mu(\zeta) + \frac{2^{2-m}}{\pi} \int_S \mu(z) \frac{\cos(n_z, z-\zeta)}{|z-\zeta|^{m-1}} dS_z = -2\phi_\infty(\zeta) , \qquad \zeta \in S. \tag{13}$$

Assuming the boundary S to be sufficiently smooth it can be proven that the solution of the interior Dirichlet-problem (12) also satisfies the Neumann-problem (8)-(10) for the exterior of the boundary S.

Calculation of Circulatory Flow around an Aerofoil.

For circulatory flow around an aerofoil one must introduce a cut to make the velocity potential single valued. The Kutta condition of smooth flow at the trailing edge can be satisfied if we construct the cut from the trailing edge to infinity.

We denote the upper and lower side of the cut by S^+ and S^-, respectively. The contour composed of the aerofoil S and the cut is denoted by $S^- + S + S^+$. Along the cut there exists a constant discontinuity in velocity potential. The jump is represented by a constant double layer potential with strength μ^+ and μ^- along S^+ and S^-, respectively. The difference $\mu^- - \mu^+$ is equal to the circulation which is taken positive in clockwise direction.

We can represent the velocity potential by

$$\phi(\zeta) = U \cdot \zeta + \frac{1}{2\pi} \int_{S^- + S + S^+} \mu(z) \frac{\cos(n_z, z-\zeta)}{|z-\zeta|} dS_z$$

or rewritten

$$\phi(\zeta) = U \cdot \zeta + \phi_d(\zeta) + \frac{1}{2\pi} (\mu^+ - \mu^-) \arg(z_t - \zeta) , \tag{14}$$

where ϕ_d is defined by (11) with $m = 2$ and z_t is the trailing edge. In this section we denote by $\arg(z_1/z_2)$ with $z_1, z_2 \in \mathbb{R}^2$ the real value of the usual function defined by the complex numbers corresponding to z_1 and z_2. The doublet strength along S follows from (12). So far we did not say anything about μ^+ and μ^-, but we still have to satisfy the Kutta condition. In the present paper we only consider aerofoils with non-zero trailing edge angle. For these cases the Kutta condition states that the flow speed must be zero at both sides of the trailing edge. Let ζ^+ and ζ^- be points at the upper and lower part of the trailing edge. The Kutta condition is satisfied if:

$$(15) \qquad \begin{cases} D\mu(\zeta^+) \to 0 & \text{for } |\zeta^+ - z_t| \to 0 , \\ D\mu(\zeta^-) \to 0 & \text{for } |\zeta^- - z_t| \to 0 , \end{cases}$$

where D denotes differentiation in the tangential direction. Application of conditions (12) and (15) to (14) yields the following integral equation

$$(16) \qquad (I-K)\mu + \beta(\mu^+ - \mu^-) = g,$$

with

$$(17) \qquad K\mu(\zeta) = \frac{-1}{\pi} \int_S \mu(z) \frac{\cos(n_z, z-\zeta)}{|z-\zeta|} \, dS_z ,$$

$$\beta(\zeta) = \frac{1}{\pi} \arg(z_t - \zeta) ,$$

$$g(\zeta) = -\, 2\, U \cdot \zeta.$$

Numerical approach - The contour S is divided into N segments S_i such that $S = \bigcup_{i=1}^{N} S_i$ and $S_i \cap S_j = \emptyset$, $i \neq j$. The begin- and end-points of the i^{th} segment are z_{i-1} and z_i and are called nodal points. On this grid μ is approximated by a piecewise constant function μ_N and the resulting equation is solved by a collocation method. The collocation points ζ_i, $i = 1,2,\ldots, N$, are taken to be the mid-points of the segments S_i. By means of projection at the collocation points we get N equations. However, we have N + 2 unknowns $\mu_{N,1}, \mu_{N,2}, \ldots, \mu_{N,N}, \mu_N^+$ and μ_N^- with $\mu_{N,i} = \mu_N(\zeta_i)$ and $\mu_N^{\pm} = \mu_N(\zeta; \zeta \in S^{\pm})$, so that we need two extra equations. Following condition (15) we replace μ_N^+ and μ_N^- by $\mu_{N,N}$ and $\mu_{N,1}$, i.e.

$$(18) \qquad \begin{cases} \mu_N^+ = \mu_N(\zeta_N), \\ \mu_N^- = \mu_N(\zeta_1), \end{cases}$$

where ζ_1 and ζ_N are the collocation points which are closest to the trailing edge at the lower and upper part of S. Let T_N be the projection operator defined by piecewise constant interpolation at the collocation points. We have to solve the following equation

$$(19) \qquad (I-T_N K)\mu_N + T_N \beta (\mu_{N,N}-\mu_{N,1}) = T_N g.$$

In aerodynamics the above numerical approach is called a first order panel method. In reference 5 we have put it in a functional analytic framework. Assuming the contour S to be sufficiently smooth (except for a small region near the trailing edge) it was shown that a once continuously differentiable numerical solution can be obtained by a single iteration

$$(20) \qquad \tilde{\mu}_N = g + (\mu_{N,N} - \mu_{N,1})\beta + K\mu_N .$$

Furthermore, it was proven that the operator K is compact on the space of essentially bounded functions, provided the boundary is sufficiently smooth. Since aerofoils (inclusive the trailing edge) are not smooth this property of K does not hold for our application.

Multigrid method - The principal aim of this section is to show that equation (19) can be solved efficiently by a multigrid iterative process. In reference 2 we introduced multigrid methods for integral equation (1). The Jacobi-relaxation was used to smooth the high-frequency errors. Assuming the integral operator to be compact we were able to prove that the reduction factors of these multigrid methods decrease as N increases. For our application this nice property is completely destroyed (see table 1) because K is not compact. Problems with respect to the convergence of the iterative process arise in the neighbourhood of the trailing edge. Here the high-frequency errors are not removed by the Jacobi-relaxation:

$$(21) \qquad \mu_N^{(\nu+1)} = T_N g + T_N K\, \mu_N^{(\nu)} - T_N\, \beta(\mu_{N,N}^{(\nu)} - \mu_{N,1}^{(\nu)}).$$

Inspection of the matrix corresponding to $T_N K T_N$ reveals that the cross-diagonal contains elements of magnitude $k-1 + \mathcal{O}(1/N)$ as $N \to \infty$ with k = (exterior trailing edge angle)/π. This occurrence of off-diagonal elements of about the same size as diagonal elements explains why Jacobi-relaxation does not work well. Therefore we apply another relaxation scheme, which we call *paired Gauss-Seidel relaxation*. In order to explain this scheme we first rewrite (21) as follows:

$$\mu_{N,i}^{(\nu+1)} = g_i + \sum_{\ell=1}^{N} k_{i\ell}\, \mu_{N,\ell}^{(\nu)} - \beta_i(\mu_{N,N}^{(\nu)} - \mu_{N,1}^{(\nu)}) \text{ for } i = 1(1)N.$$

We obtain the *paired Jacobi relaxation* (PJ) scheme by removing the cross-diagonal to the left-hand side:

$$\mu_{N,i}^{(\nu+1)} - k_{ij}\, \mu_{N,j}^{(\nu+1)} = g_i + \sum_{\substack{\ell=1 \\ \ell\neq j}}^{N} k_{i\ell}\, \mu_{N,\ell}^{(\nu)} - \beta_i(\mu_{N,N}^{(\nu)} - \mu_{N,1}^{(\nu)}),$$

for $i = 1,2,\ldots, N/2$ and $j = N+1-i$. A similar expression is obtained for $i=j$. As a result we have to solve $\frac{1}{2}N$ systems of equations of dimension 2. Substituting the new values of $\mu_{N,i}$ and $\mu_{N,j}$ as soon as they are available we obtain the paired Gauss-Seidel (PGS) relaxation scheme. For $i = 1,2,\ldots, N/2$ and $j = N+1-i$ we define

$$\nu_{\ell i} = \begin{cases} \nu & \text{for } i \le \ell \le j, \\ \nu+1 & \text{for } \ell < i \text{ and } \ell > j. \end{cases}$$

We solve simultaneously the following equations

$$\mu_{N,i}^{(\nu+1)} - k_{ij}\, \mu_{N,j}^{(\nu+1)} = g_i + \sum_{\substack{\ell=1 \\ \ell\neq j}}^{N} k_{i\ell}\, \mu_{N,\ell}^{(\nu_{\ell i})} - \beta_i(\mu_{N,N}^{(\bar\nu)} - \mu_{N,1}^{(\bar\nu)})$$

and

$$\mu_{N,j}^{(\nu+1)} - k_{ji}\, \mu_{N,i}^{(\nu+1)} = g_j + \sum_{\substack{\ell=1 \\ \ell\neq i}}^{N} k_{j\ell}\, \mu_{N,\ell}^{(\nu_{\ell i})} - \beta_j(\mu_{N,N}^{(\bar\nu)} - \mu_{N,1}^{(\bar\nu)}),$$

for $i = 1,2,\ldots, N/2$ and $j = N+1-i$, with $\bar{\nu} = \nu$ for $i=1$ and $\bar{\nu} = \nu+1$ for $1 < i \le N/2$. The matrix elements k_{ij} can be easily calculated. Let

$$\phi_{ij} = \frac{1}{\pi} \arg \frac{z_j - \zeta_i}{z_{j-1} - \zeta_i} ,$$

then

$$k_{ij} = \begin{cases} \phi_{ij} & \text{for } i \neq j , \\ \phi_{ii} + \begin{cases} 1 & \text{if } \phi_{ii} < 0 , \\ -1 & \text{if } \phi_{ii} > 0 . \end{cases} \end{cases}$$

Let X_p be a short notation for the space X_{N_p} of piecewise constant functions of dimension N_p. We introduce a sequence of spaces $\{X_p | p = 0,1,\ldots,\ell\}$ with $N_p = 32 * 2^p$ such that

$$X_0 \subset X_1 \subset \ldots \subset X_\ell .$$

The corresponding projection operators are denoted by T_p. In the context of multigrid iteration the subscript p is called level.

The calculations have been performed for several Kármán-Trefftz aerofoils with thickness $\delta = 0.05$ and length $\ell = 1.0$. These aerofoils are obtained from the circle in the χ-plane, $\chi = c\, e^{i\theta}$, by means of the mapping

$$z = f(\chi) = (\chi - \chi_t)^k / (\chi - c(\delta - i\gamma))^{k-1} ,$$

where γ measures the camber and k the exterior trailing edge angle;

$$c = 2\ell\,(\delta + (1-\gamma^2)^{\frac{1}{2}})^{k-1} / (2(1-\gamma^2)^{\frac{1}{2}})^k ,$$

$$\chi_t = c\,((1-\gamma^2)^{\frac{1}{2}} - i\gamma) .$$

Partition of the boundary on level p: Let the interval $[0,2\pi]$ be divided into N_p uniform segments with nodal points $\{\Theta_j | j = 0(1)N_p\}$. The nodal- and collocation-points in the z- plane follow from $f(ce^{i\Theta_j})$ and $f(ce^{i\Theta_{j+\frac{1}{2}}})$, respectively, $\Theta_{j+\frac{1}{2}}$ being the midpoint of subinterval $[\Theta_j, \Theta_{j+1}]$. The collocation points defined in this way are situated at the boundary and do not coincide with the collocation points of the other levels. Therefore, the elements of the matrix K_p , $p = 0,1,\ldots, \ell$, corresponding to $T_p K T_p$ have to be computed for all levels. Asymptotically for $\ell \to \infty$, the number of kernel evaluations is $\frac{4}{3} N_\ell^2$, when the values are computed once and stored. We have taken the following testcases:

I. $k = 1.90$ and $\gamma = 0$,

II. $k = 1.90$ and $\gamma = \sin 0.05$,

III. $k = 1.99$ and $\gamma = 0$,

IV. $k = 1.99$ and $\gamma = \sin 0.05$.

The velocity U of the undisturbed flow is taken to be $(\cos\tau, \sin\tau)$ with τ the angle of incidence. For the above testcases we give numerical results for $\tau = 0$ and $\tau = \pi/2$.

Algorithm: The approximate solution of (19) is obtained by the multigrid method defined in the ALGOL-68 program given in TEXT 1:

```
PROC mulgrid = (INT p,σ, VEC u,g) VOID:
IF p = 0
THEN solve directly (u,g)
ELSE FOR  i TO σ
     DO   relax (u,g); INT n = UPB u ;
          VEC residu = g-u+K_p*u-β_p*(u[n] - u [1]);
          VEC um := 0_{p-1}, gm := restrict (residu);
          mulgrid (p-1, ν, um, gm );
          u := u + interpolate (um);
          relax (u,g)
     OD
FI
```

TEXT 1. Multigrid algorithm.

Because of reasons of efficiency the number of coarse grid corrections (integer ν) must be less than 4. For $\nu = 1$ and $\nu = 2$ we obtain the so-called V- and W-cycle, respectively. Here we choose $\nu = 2$. For the 3-D problem of flow around an ellipsoid we take $\nu = 1$. The interaction between the grids is defined by the procedures *restrict* and *interpolate* which are specified as follows. Let n be the upper bound of *VEC u*, then:

$$restrict\ (u)\ [i] := 0.5 * (u\ [2*i-1] + u[2*i]),\ i = 1(1)n/2,$$

$$interpolate\ (u)[2*i] := interpolate\ (u)\ [2*i-1] := u\ [i],\ i = 1(1)n.$$

On level 0 the system of equations is solved by Gaussian elimination. For *relax* we take: Jacobi - , paired Jacobi - and paired Gauss-Seidel relaxation, respectively. We start our algorithm on level 0. The interpolation to level p ($p \geq 1$) of the approximate solution from level p-1 is used as initial guess of the multigrid process at level p; truncation occurs when the residual is less than 10^{-6}. Let *VEC* g_p denote the restriction of g to the collocation points of level p. In ALGOL-68 notation this algorithm reads:

```
solve directly (u0,g0);
FOR p TO 3
DO     up := interpolate (u0);
       FOR i TO 25 WHILE residual > 10^-6
       DO mulgrid (p, 1, up, gp) OD;
       u0 := COPY up
OD;
```

TEXT 2. Implementation of the full multi-grid algorithm.

In the following table we compare the performance of the multigrid processes using various relaxation schemes.
From this table we conclude that the multigrid method defined by Jacobi-relaxation is not acceptable (it converges too slowly). The process defined by PGS-relaxation turns out to be the most efficient. Furthermore, we draw the following conclusions: 1. the number of iterations decreases as N increases and 2. on the highest level (N=256) only a few iterations are necessary.

TABLE 1 - NUMBER OF ITERATIONS

Testcase	τ	N = 64 J	PJ	PGS	N = 128 J	PJ	PGS	N = 256 J	PJ	PGS
I	0	15	4	3	13	3	2	11	3	2
	$\pi/2$	15	9	9	4	7	5	2	4	2
II	0	15	8	8	13	5	5	11	3	2
	$\pi/2$	15	10	9	9	7	6	6	4	2
III	0	>25	4	3	>25	3	3	>25	3	2
	$\pi/2$	> 25	12	9	19	9	6	9	6	3
IV	0	> 25	11	8	>25	7	4	>25	6	4
	$\pi/2$	> 25	13	10	>25	10	8	>25	7	3

J - Jacobi , PJ - Paired Jacobi, PGS - Paired Gauss-Seidel.

Calculation of Potential Flow around an Ellipsoid.

The numerical approach to find the solution of (13) is connected with the shape of the kernel-function. Application of the collocation method in the space of piecewise constant functions leads to moment-integrals, which consist of the calculation of solid angles. We consider the ellipsoid defined by

$$\frac{x^2}{4} + y^2 + z^2 = 1 .$$

The velocity of the undisturbed flow is given by $U = (1,0,0)$. The partition of the ellipsoid into panels is carried out as follows. First we divide the surface into N rings by planes orthogonal to the z-axis. Next each ring is divided into N^* trapeziform segments. The spherical caps are divided into N^* triangle-form segments. We denote these segments by S_{ij} , $i = 1(1)N$ and $j = 1(1) N^*$. The collocation points are chosen to be the "midpoints" of these segments and are situated at the surface. The solid angle subtended at ζ by S_{ij} with $\zeta \notin S_{ij}$ is given by

$$\int_{S_{ij}} \frac{\cos(n_z, z-\zeta)}{|z-\zeta|^2} \, dS_z \, .$$

In contrast with 2-D in general these integrals cannot be evaluated analytically. We approximate S_{ij} by one or two flat planes. The solid angles subtended by such planes can be evaluated analytically.

Multigrid method - The different grids are related by $N_p = 4 * 2^p$ and $N_p^* = 4 * 2^p$. Putting $\beta_p = 0$ we use the algorithm given in TEXT 1 with $\nu = 1$. Analogously to 2-D we define the procedures *solve directly*, *restrict* and *interpolate* by Gaussian elimination, weighted injection and piecewise constant interpolation, respectively. For *relax* we take the Jacobi-relaxation scheme. Assuming the surface to be smooth Wolff (ref.6) has analysed this multigrid method. He has proven that the reduction-factor of the multigrid process is less than ch^α for $h \to 0$, where h and α are a measure for the mesh-size and the smoothness of the surface, respectively. For the ellipsoid $\alpha = 1$.

Numerical results - In table 2 we give the residuals and the observed reduction factors

$$\eta_i = \| \mu_N^{(i+1)} - \mu_N^{(i)} \| / \| \mu_N^{(i)} - \mu_N^{(i-1)} \| ,$$

with $\| . \|$ the supremum norm. We also give the mean reduction factor

$$\bar{\eta} = \{ \prod_{i=1}^{k} \eta_i \}^{1/k}$$

and the operation count expressed in work-units. One work-unit is defined by (total number of multiplications)/$(N_\ell * N_\ell^*)^2$ with ℓ the highest level. We only take into account matrix-vector multiplications and the direct solution on the coarsest grid for which we count $\frac{1}{3}(N_0 * N_0^*)^3$ multiplications. Table 2 enables us to draw the following conclusions: 1. Comparing the results obtained with $\ell = 2$ and $\ell = 3$ we see that the mean reduction factor of the multigrid

TABLE 2 - POTENTIAL FLOW AROUND AN ELLIPSOID *

MULTIGRID METHOD ($N_0 = 4$, $N_0^* = 4$)					
$\ell = 2$; $N_2 = 16, N_2^* = 16$			$\ell = 3$; $N_3 = 32$, $N_3^* = 32$		
iter	residual	red. factor	iter	residual	red. factor
1	$1.17\ 10^{-1}$		1	$4.56\ 10^{-2}$	
2	$2.04\ 10^{-3}$	$4.13\ 10^{-2}$	2	$4.38\ 10^{-4}$	$1.67\ 10^{-2}$
3	$7.75\ 10^{-5}$	$1.40\ 10^{-2}$	3	$8.48\ 10^{-6}$	$6.98\ 10^{-3}$
4	$1.89\ 10^{-6}$	$4.63\ 10^{-2}$	4	$9.93\ 10^{-8}$	$2.56\ 10^{-2}$
5	$6.54\ 10^{-8}$	$2.36\ 10^{-2}$			
mean red. factor: $2.83\ 10^{-2}$			mean red. factor: $1.44\ 10^{-2}$		
operation count : 10.68			operation count : 8.53		
JACOBI ITERATIVE PROCESS					
$N = 16$, $N^* = 16$			$N = 32$, $N^* = 32$		
iter	residual	red. factor	iter	residual	red. factor
1	1.73		1	2.15	
2	$8.05\ 10^{-1}$	$4.51\ 10^{-1}$	2	1.20	$5.44\ 10^{-1}$
3	$3.82\ 10^{-1}$	$4.68\ 10^{-1}$	3	$6.72\ 10^{-1}$	$5.57\ 10^{-1}$
4	$1.83\ 10^{-1}$	$4.75\ 10^{-1}$	4	$3.79\ 10^{-1}$	$5.62\ 10^{-1}$
5	$8.75\ 10^{-2}$	$4.78\ 10^{-1}$	5	$2.14\ 10^{-1}$	$5.64\ 10^{-1}$
6	$4.20\ 10^{-2}$	$4.79\ 10^{-1}$	6	$1.21\ 10^{-1}$	$5.65\ 10^{-1}$
7	$2.01\ 10^{-2}$	$4.80\ 10^{-1}$	7	$6.85\ 10^{-2}$	$5.65\ 10^{-1}$
⋮	⋮	⋮	8	$3.88\ 10^{-2}$	$5.66\ 10^{-1}$
			⋮	⋮	⋮
21	$6.94\ 10^{-7}$	$4.80\ 10^{-1}$	27	$7.79\ 10^{-7}$	$5.66\ 10^{-1}$
mean red. factor: $4.77\ 10^{-1}$			mean. red. factor: $5.64\ 10^{-1}$		
operation count : 21			operation count : 27		

* $\frac{x}{4} + y^2 + z^2 = 1$, $\mathcal{U}$ parallel to the x - axis.

method has been decreased by a factor 2, which is in agreement with the theoretical results of Wolff (ref.6) and 2. the multigrid method is much cheaper than the Jacobi-iterative process.

CALCULATION OF OSCILLATING DISK FLOW.

The rotating flow due to an infinite disk performing torsional oscillations at an angular velocity $\Omega \sin \omega \tau$ in a viscous fluid otherwise at rest involves two relevant length scales : 1. the Von Kármán layer thickness $(\nu/\Omega)^{1/2}$, where ν is the kinematic viscosity and 2. the Stokes layer thickness $(\nu/\omega)^{1/2}$. By means of the Von Kármán similarity transformations the velocities (u,v,w) in a cylindrical coordinate system (r,ϕ,x) can be written as:

$$u = \Omega r f(z,t) \ , \ v = \Omega r g(z,t) \ , w = - 2(2\nu\omega)^{1/2} h(z,t),$$

where $z = (\frac{\Omega^2}{2\nu\omega})^{1/2} x$ and $t = \omega\tau$. In that case the Navier-Stokes equations reduce to the partial differential equations (2) - (4). Apparently the oscillating disk flow is characterized by the parameter $\varepsilon = \Omega/\omega$, which determines the ratio of the Stokes layer thickness to the Von Kármán layer thickness.

For the high-frequency flow ($\varepsilon << 1$) analytical solutions are found in the literature in the form of series expansions in terms of ε. This type of flow consists of an oscillatory inner layer (i.e. Stokes layer) near the rotating disk and a secondary outer layer (i.e. Von Kármán layer). Using a multiple scaling technique Benney (ref.7) was able to find series expansions valid throughout the region of flow. The first order terms of the solution are given by:

$$(22) \qquad g(z,t) = e^{-z/\varepsilon} \sin(t-z/\varepsilon) \ , \ f(z,t) \sim \varepsilon \, e^{-4az} \text{ for } z \to \infty \ ,$$

with $a = 0.265$. In reference 3 we used this technique to determine the axial inflow at infinity up to the term with ε^3 :

$$(23) \qquad h(\infty,0) = a\,\varepsilon + \{ a\,b + \frac{1}{16}(\sqrt{2} - 1) \}\,\varepsilon^2 + O(\varepsilon^3),$$

with $b = -0.207$. Inspection of (22) reveals that problem (2) - (6) is singularly perturbed and for a fixed t the solution contains more and more high frequency components as $\varepsilon \to 0$.

In this paper we discuss two computational methods to find the periodic solution satisfying (6). The first method is based on simulation of the physical process by taking the state of rest as an initial condition and eliminating the transient effects by integration in time. In mathematical terminology this process can be interpreted as Picard's method for computing a fixed point. Let the velocity vector be:

$$v = (f,g,h).$$

Denote by $(v(z,t); v_0)$ the solution of the usual initial-value problem (2)-(5) with initial data:

$$(24) \qquad v(z,0) = v_0(z).$$

Assume that the initial data v_0 belong to a suitable class L. Define a map of L into itself by the equation

$$(25) \qquad K_\varepsilon(v_0) := (v(\cdot,2\pi); v_0),$$

being the solution of (2) - (5) and (24) at $t = 2\pi$. Since (2)-(4) is a parabolic system K_ε may be expected to have a smoothing influence, just as the integral operators of the Fredholm equations studied in reference 2. In operator notation simulation of the physical process is written as the Picard sequence

$$(26) \qquad v_{i+1} = K_\varepsilon(v_i) \text{ with } v_0 = 0.$$

The periodic condition (6) rewrites as

(27) $\qquad v = K_\varepsilon(v), \quad v \in L.$

We remark that K_ε is a non-linear operator. For $\varepsilon < 1$ (26) converges slowly. Therefore we have devised another method. Since equation (27) has a superficial resemblance with a Fredholm equation of the second kind we have applied a multigrid method to (27).

Numerical Approach

This section is divided into two parts: 1. the numerical solution of the initial-boundary value problem (2) - (5) with the initial data (24) and 2. numerical methods for finding periodic solutions satisfying (6).

Discretization of the initial-boundary value problem - Consider the partial differential equations (2) -(4) with the boundary conditions (5) and the initial data (24). To this problem we apply implicit finite difference techniques in combination with an appropriate stretching function for the construction of the computational grid. In calculations the boundary conditions at infinity are applied at a finite value $z = \ell$:

(28) $\qquad f(\ell,t) = g(\ell,t) = 0.$

We want to resolve the flow structure near the disk with a limited number of mesh points. Therefore, taking into account (22) we transform the z-coordinate by:

(29) $\qquad z(x) = \ell(\varepsilon x + (1-\varepsilon)x^3), \quad x \in [0,1],$

and we take the mesh covering of the new range $0 \le x \le 1$ uniform with stepsize $\Delta x = {}^1/N$. Integration in time is done by the Euler-backward formula:

$$g_t = \frac{g_{k+1} - g_k}{\Delta t} \quad \text{,with } \Delta t = 2\pi/T .$$

The right-hand sides of (2) - (3) are discretized by central differences at $t = t_{k+1}$. The left- and right- hand side of (4) are integrated by means of the mid-point and trapezoidal rule, respectively. The resulting non-linear system of finite difference equations is solved by means of Newton iteration, which is terminated if the residual is less than 10^{-6}. For further details see reference 3.

Numerical methods for computing periodic solutions - Using the above finite difference approach we define the discrete counterpart of the operator K_ε and the velocity vector υ by $K_{\varepsilon;\, N,T,\ell}$ and υ_N respectively. In discrete operator notation the periodic condition reads:

$$\upsilon_N = K_{\varepsilon;\, N,T,\ell}\,(\upsilon_N). \tag{30}$$

In the present paper we propose two computational methods to solve (30) : A. simulation of the physical process by Picard iteration and B. a multigrid method. In the first method the parameters ε, N, T and ℓ are fixed. In the second method the parameters N and T are taken from a sequence $\{(N_p, T_p)\}$, $p = 0,1,\ldots, L$ such that with $p = L$ we have $N_L = N$, $T_L = T$ and with $p < q \le L$ we have $N_p \le N_q$, $T_p \le T_q$ (i.e. a smaller p corresponds with a coarser discretization).

A. Simulation of the physical process: We take the state of rest ($\upsilon_N^{(0)} \equiv 0$) as an initial condition. The transient effects are eliminated by Picard's method:

$$\upsilon_N^{(i+1)} = K_{\varepsilon;N,T,\ell}\,(\upsilon_N^{(i)}). \tag{31}$$

The iteration index i counts the number of periods that is calculated. This process is truncated if the residual $\| \upsilon_N^{(i)} - K_{\varepsilon;N,T,\ell}\,(\upsilon_N^{(i)})\|$ is less than $0.5\ \ 10^{-4}$. Here

$$\|\upsilon_N\| = \max_{0\le j\le N} |\, g_j \,| + \max_{0\le j\le N} |\, h_j \,| .$$

B. Multigrid method: We introduce a sequence of grids with $N_p = 20 * 2^p$ and $T_p = 8 * 2^p$. The integer p is called level. We replace the subscript N_p by p:

$$\upsilon_{N_p} = \upsilon_p \text{ and } K_{\varepsilon;\, N_p,\, T_p,\ell} = K_{\varepsilon;\, p} .$$

Denote the velocity at grid point x_j on level p by $\upsilon_p[j] = (f_j, g_j, h_j)$. The addition v_p [j] + v_p [k] and the multiplication c * v_p [j] are defined as usual (element by element). The interaction between the grids is defined by piecewise-linear interpolation:

$$\textit{interpolate}\ (U)\ [j] = \begin{cases} U\,[\,j/2], & j = 0,2,\ldots,\ 2N, \\ 0.5 * (U\,[\frac{j+1}{2}] + U\,[\frac{j-1}{2}\,]), & j = 1,3,\ldots,\ 2N-1, \end{cases}$$

and by injection:

$$\textit{restrict}\quad (U)\ [j] = U\ [2j\,]\ ,\ j = 0,1,\ldots,\ \frac{N}{2}\ ,$$

where N is the upper-bound of the velocity vector U.

We use a multigrid method that starts on level 0 with simulation of the physical process (method A). For small values of ε we apply continuation. Suppose we have the following ε-sequence $\{\varepsilon_\ell \mid \varepsilon_0 > \varepsilon_1 > \ldots > \varepsilon_m$ with $\varepsilon_0 = 1\}$. At each stage of this continuation process we approximately solve the equation $v_0 = K_{\varepsilon_\ell\,;\,0}\ (v_0)$ by (31) until the residual is less than $0.5\ 10^{-3}$. As initial guess of (31) we take the solution of the previous stage ($\varepsilon = \varepsilon_{\ell-1}$). For $\varepsilon = \varepsilon_0$ we take the state of rest. Denote the solution of this continuation method by $v_0\ (\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_m)$.

Since (30) is a non-linear equation it is only solved approximately. Let U_p be an approximation to the solution v_p of (30) on level p. We define the defect of U_p by

$$d_p = U_p - K_{\varepsilon;p}\ (U_p)\ .$$

The multigrid method is given by the ALGOL -68 program in TEXT 3, where *VELO* is a mode for the vector of unknowns:

```
MODE VELO = STRUCT (VEC f,g, h) .

PROC compute periodic solution = ( # to level # INT ℓ) VOID:
(U0 := v0(ε0,ε1,...,εm)·
 FOR j TO ℓ
 DO    d j-1 := U j-1 - K ε; j-1 (U j-1) ;
       U j := interpolate (U j-1);
       multigrid (j,1,U j,0 j)
 OD
 );
```

```
PROC multigrid = (INT m, σ, REF VELO U, VELO y) VOID:
(IF m = 0
 THEN FOR k TO 50 WHILE residual > δ_ε
      DO VELO r = y - U + K_{ε;m} (U);
         residual := ‖r‖;
         U := U + ω_k * r
      OD
 ELSE FOR i TO σ
      DO  U := y + K_{ε;m} (U);
          VELO d = d_{m-1}-restrict (y-U + K_{ε;m}(U));
          VELO v :=  COPY U_{m-1};
          multigrid (m-1, 2, v, d);
          U := U + interpolate (U_{m-1} - v)
      OD
 FI
);
```

TEXT 3 Multigrid algorithm for the computation of periodic solutions of parabolic equations.

The structure of this multigrid algorithm has been proposed by Hackbusch (ref.8) for the numerical solution of general time-periodic parabolic problems. Here we apply it to the particular problem of oscillating disk flow.

On level 0 of *multigrid* we use overrelaxation for extremely small values of ε. The parameter ω_k takes the values 1,2 and 4. Initially we put $\omega_k = 1$. If the axial inflow converges slowly it is multiplied by a factor 2. As soon as the residual increases the value $\omega_k = 1$ is restored.

Numerical results - From Zandbergen and Dijkstra (ref.9) it is known that Von Kármán's rotating disk solution can be represented sufficiently accurate with $\ell = 12$, hence we fix infinity at this value. We give numerical results for the following values of ε :

$$\varepsilon_0 = 1 \ , \ \varepsilon_1 = 0.5 \ , \ \varepsilon_2 = 0.1 \ , \ \varepsilon_3 = 0.05.$$

This sequence is also applied in the continuation process that is used to find an approximation U_C of the multigrid method, e.g. for $\varepsilon = 0.1$ we have $U_0 := v_0\ (1, 0.5, 0.1)$. For N = 160 and T = 64 we compare the performance of simulation of the physical process (method A) and the multigrid method (B). On the coarsest grid the latter method needs 20 stepsizes in space and 8 stepsizes in time; hence it uses four levels: 0,1, 2 and 3.

Let a work unit be defined by the computational work needed for calculating one Picard iterate with N = 160 and T = 64. In table 3 we compare the computed axial inflow at infinity with the value of its asymptotic approximation (23) for $\varepsilon \to 0$. Between parentheses we give the number of work units and the iteration error $\| U_N - K_{\varepsilon; N,T,\ell}\ (U_N) \|$, where U_N is the final solution.

On level 0 of the multigrid method we used Picard iteration (i.e. $\omega_k \equiv 1$) for $\varepsilon \geq 0.1$. The iterative process was terminated when the residual was less than $\delta_\varepsilon = 0.5\ \ 10^{-4}$. For $\varepsilon = 0.05$ we have applied overrelaxation ($1 \leq \omega_k \leq 4$) and we have put $\delta_{0.05} = 10^{-7}$. That is the reason why the computational work increased for this case.

TABLE 3 AXIAL INFLOW*

ε	method A	method B	(23)
1.0	0.2014 (8, 4.4 10^{-5})	0.2014 (6.8, 9.3 10^{-7})	0.2360
0.5	0.1177 (17, 4.7 10^{-5})	0.1178 (7.0, 3.9 10^{-6})	0.1253
0.1	0.0236 (74, 4.9 10^{-5})	0.0271 (7.4, 1.6 10^{-5})	0.0262
0.05	0.0083 (72, 4.9 10^{-5})	0.0137 (12.5, 3.3 10^{-6})	0.0132

* Between parentheses : number of work units, residual.

From table 3 we conclude that the multigrid method becomes more efficient as ε decreases. For $\varepsilon = 0.1$ the computational work has been reduced by a factor 1/10. For $\varepsilon = 0.1$ and $\varepsilon = 0.05$ the numerical results of method A still contain a low-frequency error. In this case the test for termination of the physical process is not adequate. The process converges slowly, as can be seen from figure 1, in which we have displayed the axial inflow as a function of the number of periods. For $\varepsilon = 0.05$ the axial inflow is still increasing after 72 periods. The same phenomenon occurs on the coarsest grid of the multigrid method. Therefore we have applied overrelaxation.

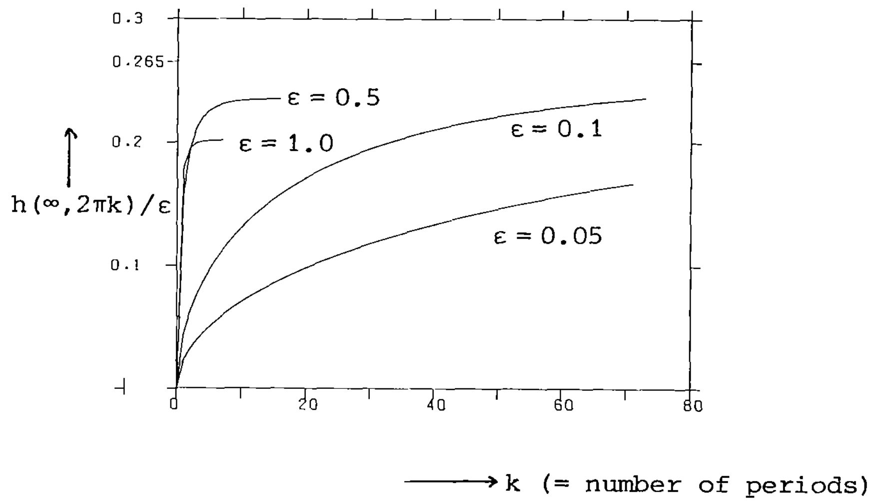


FIGURE 1. Dependence of the axial inflow on the number of periods

The results of our analysis are given in figures 2-3. The profiles of the variables f/ε , g and h/ε are displayed in figure 2. We see that there is an oscillatory boundary layer. For smaller values of ε (see figures 2 (c-d)) the azimuthal component of velocity (g) is confined to this boundary layer and the radial and axial component of velocity (resp. f and h) persist outside this layer. The results for the quantities $\varepsilon\, g_z(0,t)$, $f_z(0,t)$ and $h(\infty,t)/\varepsilon$ are displayed in figure 3. Comparing these figures we see that the fluctuations in $h(\infty,t)$ decrease as $\varepsilon \to 0$. This means that outside the boundary layer the fluid motion becomes stationary (i.e. the outer flow does not depend on t). These numerical results are in agreement with the analytical solutions of Benney (ref.7).

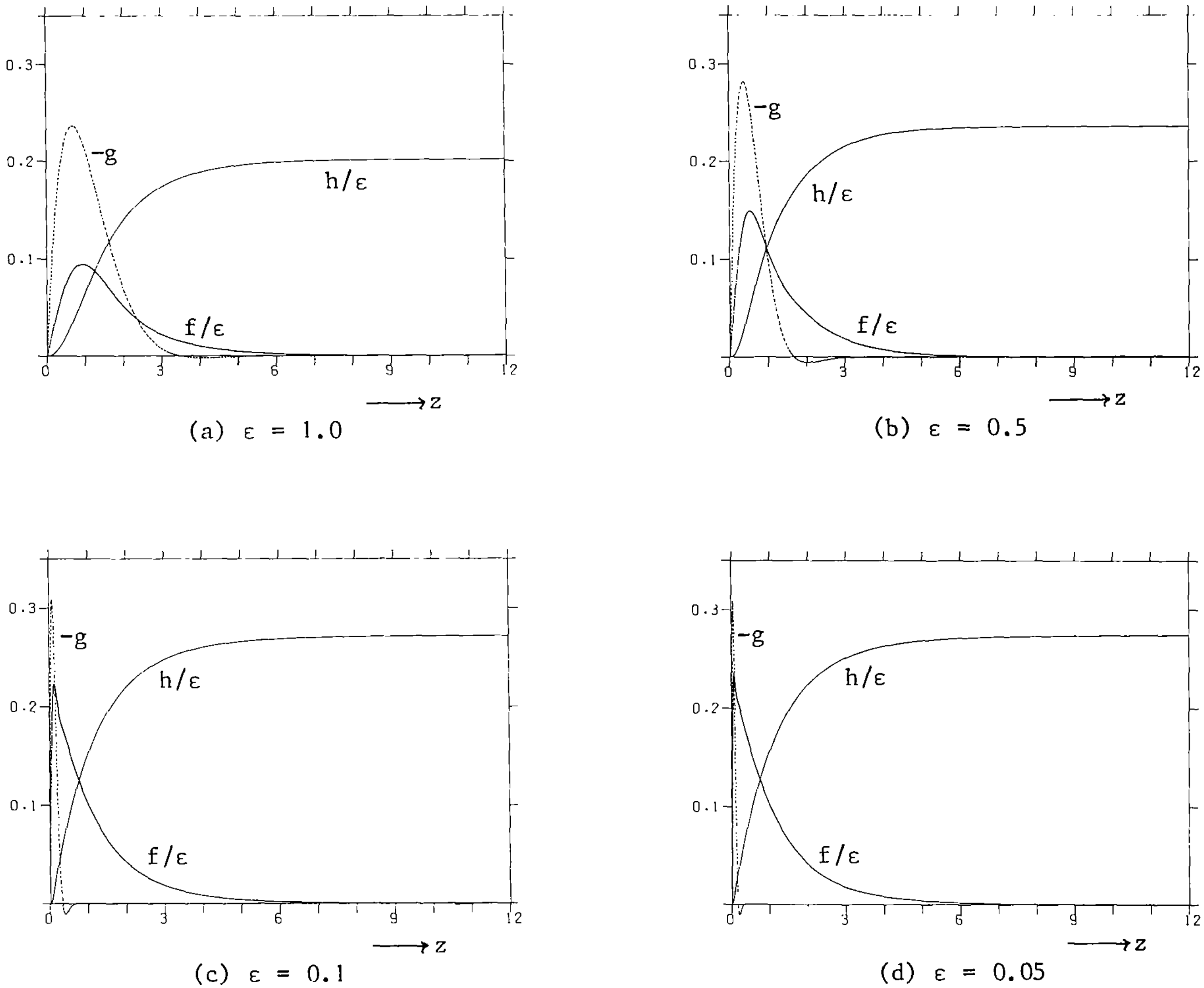


FIGURE 2. Velocity profiles

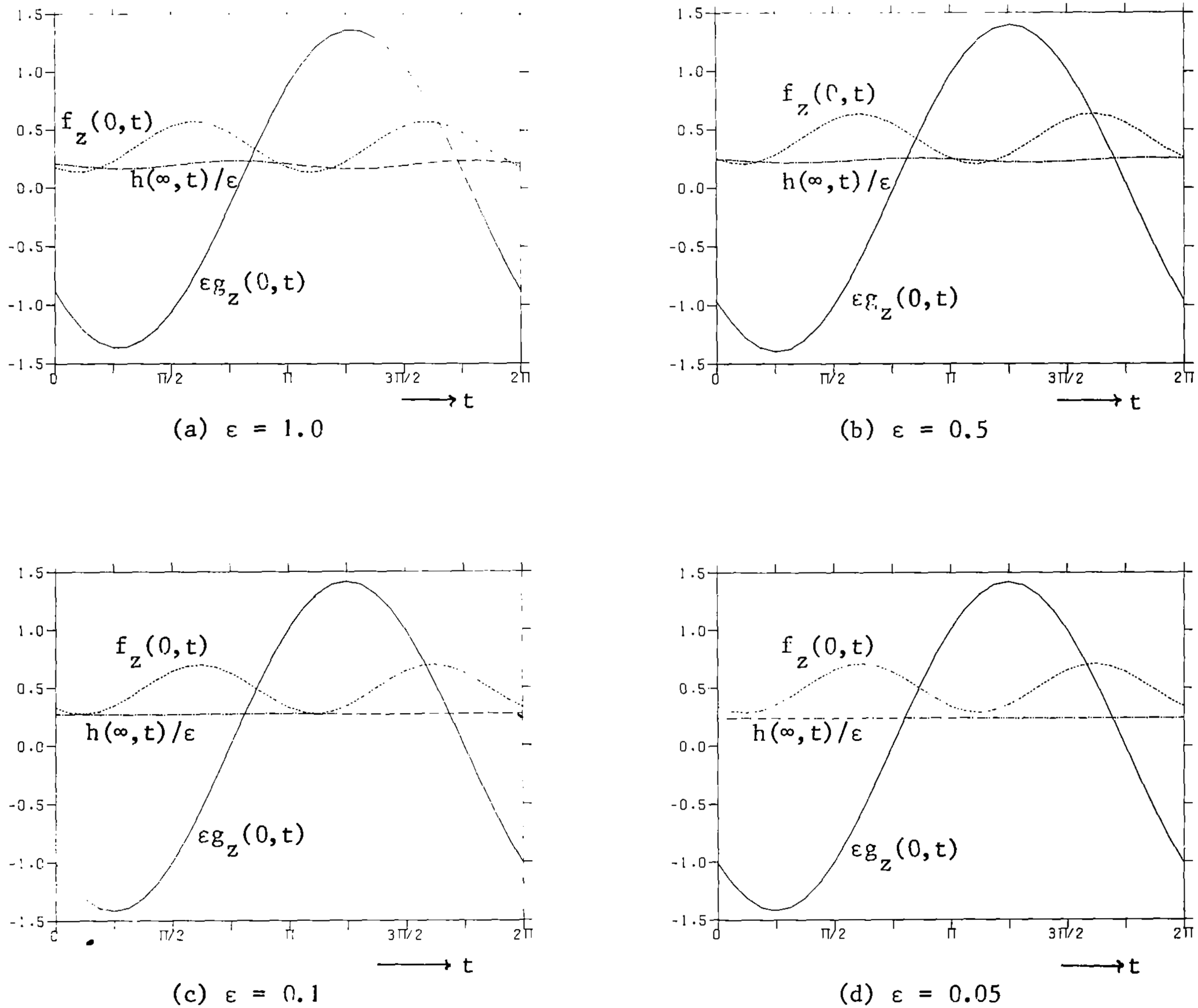


FIGURE 3. Axial inflow and shear stresses

Finally, from the results just presented we conclude that for the computation of periodic solutions of the single disk problem for $\varepsilon \leq 1$ the multigrid method is preferable, whereas for $\varepsilon > 1$ simulation of the physical process may be employed.

REFERENCES

1. Brandt, A.: Multi-Level Adaptive Solutions to Boundary-Value Problems. Math. Comp., vol. 31, no. 138, Apr. 1977, pp. 333-390.
2. Hemker, P.W.; and Schippers, H.: Multiple Grid Methods for the Solution of Fredholm Integral Equations of the Second Kind. Math. Comp., vol. 36, no. 153, Jan. 1981, pp. 215-232.
3. Schippers, H.: Analytical and numerical results for the nonstationary rotating disk flow. J. Engng. Math., vol. 13, no. 2, Apr. 1979, pp. 173-191.
4. Muschelischwili, N.I.: Singuläre Integralgleichungen. Akademie-Verlag, Berlin, 1965.
5. Schippers, H.: On the regularity of the principal value of the double layer potential, to be published in Journal of Engineering Mathematics.
6. Wolff, H.: Multigrid method for the calculation of potential flow around 3-D bodies, Report NW xxx/81, Mathematisch Centrum, Amsterdam, 1981 (Preprint).
7. Benney, D.J.: The flow induced by a disk oscillating in its own plane. J. Fluid Mech., vol. 18, no. 3, March 1964, pp. 385-391.
8. Hackbusch, W.: The fast numerical solution of time-periodic problems, Report 79-14, Mathematisch Institut, Universität zu Köln, 1979.
9. Zandbergen, P.J.: and Dijkstra, D.: Non-unique solutions of the Navier-Stokes equations for the Kārmān swirling flow. J. Engng. Math., vol.11, no. 2, April 1977, pp. 167-188.

GENERAL RELAXATION SCHEMES IN MULTIGRID ALGORITHMS

FOR HIGHER ORDER SINGULARITY METHODS

B. Oskam and J.M.J. Fray
National Aerospace Laboratory, NLR

SUMMARY

This paper describes relaxation schemes based on an approximate and incomplete factorization technique (AF). These AF schemes allow one to construct a fast multigrid method for solving integral equations of the second as well as integral equations of the first kind. Novel items are the smoothing factors found for integral equations of the first kind, and the comparison with similar results for equations of the second kind. Application of the MG algorithm shows convergence to the level of the truncation error, of a second order accurate panel method, within 2 multigrid cycles.

INTRODUCTION

Most effort going into the application of multigrid techniques seems to be directed to solving the sparse systems of difference equations associated with partial differential equations. However, the multigrid technique can also be used advantageously to solve the nonsparse systems of equations that arise from integral equations, as shown in references 1 and 2.

In the present paper we study the application of multigrid techniques to the solution of integral equations associated with potential flow problems. This effort fits into the larger framework of the development, at NLR, of a next generation singularity or "panel" method. A question associated with this development is whether singularity methods do have a future, particularly in view of the current progress in finite difference methods. Reference 3 contains several arguments for a positive answer to this question, but at the same time presents the rather stringent requirement of high computational efficiency. The scope of the present investigation is limited to the analysis of multigrid (MG) techniques and the subsequent application to some model problems in two dimensions. Various relaxation schemes, which are used as smoothing operators in multigridding, are evaluated.For some particular geometries, such as an unbounded flat plate and a circular cylinder, this smoothing problem is analyzed by the local mode analysis of reference 4. For more complicated geometries, such as an airfoil, it is found that the finite-dimensional, discrete Fourier transform can be used to define a global smoothing factor which represents an upper bound of the actual convergence factor of the high frequency components of the residual vector. A general multigrid algorithm is described and applied to solve the potential flow problem of multicomponent airfoils.

Before starting the discussion of the integral equations it is important to realize that the asymptotic operation counts remain of the order of n^2 if nothing is done to reduce the work associated with the residue evaluations which involve a full matrix times vector multiplication. Multigrid methods to lower the computational work involved with these residue evaluations are currently being studied at NLR, see reference 3. The basic concept is to lower the asymptotic operation counts by treating the far field connections on a sequence of coarser grids without compromizing the truncation error. These aspects of a next generation panel method are however outside the scope of the present paper.

INTEGRAL EQUATIONS

Most panel methods use the boundary condition of zero-normal-velocity on the surface of the contour to derive an integral equation for a distribution of surface sigularity, source or doublet, over the body surface. Let us denote the source and doublet strength by σ and μ, respectively, and let $\vec{x}_p$ and $\vec{x}_q$ be the positions of the points p and q. The normal velocity at the point p induced by distributions of these singularities may be represented as

$$v_n^s\ (\vec{x}_p) = \frac{1}{2\pi} \int \sigma(\vec{x}_q)\ \frac{\partial}{\partial n_p} (\ln|\vec{r}_{pq}|) ds_q, \qquad (1)$$

and

$$v_n^d\ (\vec{x}_p) = \frac{1}{2\pi} \int \mu\ (\vec{x}_q)\ \frac{\partial}{\partial n_p}\ \frac{\partial}{\partial n_q} (\ln|\vec{r}_{pq}|) ds_q, \qquad (2)$$

where $\vec{r}_{pq} = \vec{x}_p - \vec{x}_q$ and n_q is the outward normal, and direction of the doublet axis, at the point q. The normal at $\vec{x}_p$ is denoted by n_p.
The integration variable s is the distance measured along the contour.

Attention is directed to two particular panel methods which may be formulated by employing eqs. (1) and/or (2). The first is the surface source method having an unknown source distribution on the body surface and an auxiliary doublet distribution of known shape but unknown magnitude, also on the body surface, to produce the lift, see reference 5. The second panel method considered employs equation (2) only and is called the doublet method, e.g. see reference 6. The reason these two methods have been employed in the present paper is that they produce quite different integral equations, being of the first and second kind for the doublet and source method respectively.

To facilitate the discussion of various discretization schemes we rewrite equation (2) for the particular case of an unbounded flat plate as

$$v_n^d\ (x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{d^2\mu}{d\xi^2} \ln|x-\xi|\ d\xi$$
$$- \frac{1}{2\pi} \int_{-\infty}^{+\infty} d\left(\frac{d\mu}{d\xi} \ln|x-\xi|\right) - \frac{1}{2\pi}\int_{-\infty}^{+\infty} d\left(\frac{\mu(\xi)}{x-\xi}\right) \qquad (3)$$

where x, ξ is the distance measured along the plate.

DISCRETIZATION OF INTEGRAL EQS

The aerodynamic influence coefficients are evaluated using a consistent, small curvature expansion of the integrals that remain after discretization, see Hess (ref. 7). Specifically, the profile curve that defines a two-dimensional body is approximated by a piecewise quadratic representation and the source, doublet distributions are approximated by piecewise linear, quadratic representations, respectively. These choices result in aerodynamic influence coefficients (AIC's) of second order accuracy in h, where h is the panel size.

Let the doublet representation for the case of an unbounded flat plate be given by

$$\tilde{\mu}(\xi) = \mu_i + \left(\frac{d\mu}{d\xi}\right)_i (\xi-\xi_i) + \left(\frac{d^2\mu}{d\xi^2}\right)_i \frac{1}{2} (\xi-\xi_i)^2, \qquad |\xi-\xi_i| \leq \frac{h}{2} . \tag{4}$$

For the purpose of studying the dependency of the smoothing factor on the discretization scheme the derivatives in equation (4) have been discretized by 3-point differences

$$\left(\frac{d\mu}{d\xi}\right)_i = (\mu_{i+1}-\mu_{i-1})/2h \qquad \text{and} \qquad \left(\frac{d^2\mu}{d\xi^2}\right)_i = (\mu_{i+1}-2\mu_i + \mu_{i-1})/h^2, \tag{5}$$

and by 5-point differences, resulting from a continuity requirement of $\tilde{\mu}$ across panel edges,

$$\left(\frac{d\mu}{d\xi}\right)_i = (-\mu_{i+2} + 10\mu_{i+1} - 10\mu_{i-1} + \mu_{i-2})/16\ h \tag{6a}$$

and

$$\left(\frac{d^2\mu}{d\xi^2}\right)_i = (-\mu_{i+2} + 8\mu_{i+1} - 14\mu_i + 8\mu_{i-1} - \mu_{i-2})/4\ h^2, \tag{6b}$$

where μ_i is the value of the doublet representation $\tilde{\mu}$ at ξ_i, which is the midpoint of panel with index i. For the case of the flat plate all panels have equal size h. The difference between the 3-point and 5-point representations, $\tilde{\mu}_{3-p}$ and $\tilde{\mu}_{5-p}$ respectively, turns out to be

$$\tilde{\mu}_{5-p} - \tilde{\mu}_{3-p} = \frac{h^2}{8}\left(\frac{d^3\mu}{d\xi^3}\right)_i (\xi-\xi_i) + \frac{h^2}{8} \left(\frac{d^4\mu}{d\xi^4}\right)_i (\xi-\xi_i)^2, \tag{7}$$

which is of the same order in h as higher order terms neglected in equation (4). Thus the 3- and 5-point differences both result in a doublet representation of third order accuracy in h. Both representations have sufficient continuity at the panel edges such that the contributions of the second and third integral in equation (3) may be neglected, being not larger than the basic truncation error of the first integral in eq. (3).

Evaluating eq. (3) at panel control points, after substituting eq. (4), results in a system of algebraic equations

$$v_n^d(x_i) = \frac{1}{2\pi h} \sum_{k=-\infty}^{+\infty} a_k \, \mu_{k+i}, \qquad i = \infty, \dots \infty, \tag{8}$$

where

$$a_k = k \ln \left|1+32\, k/(2k-3)(2k+1)^3\right| + \frac{3}{2} \ln\left|1-8/(4k^2-1)\right|, \tag{9}$$

or

$$a_k = \frac{9}{16} \ln\left|1+16/(4\,k^2-25)\right| + \frac{21}{8} \ln \left|1-8/(4k^2-1)\right|$$

$$+ \frac{1}{4} k \ln\left|1-8k/(4k^2+k-15)\right| + 2\, k \ln\left| 1+8\, k/(4k^2-4k-3)\right|$$

$$+ \frac{7}{2} k \ln\left|1-2/(2k+1)\right|. \tag{10}$$

Equation (9) represents the AIC's resulting from the 3-point differences in eq. (5) and the AIC's of eq. (10) correspond to 5-point differences (eq. (6b)).

For the case of a curved contour, such as an airfoil, we need a small curvature expansion of the integrals as mentioned before. Moreover we will take a nonuniform panel distribution. The resulting expressions of the AIC's will not be presented here for the sake of brevity. However it should be mentioned that the AIC's of the doublet distributions are based on a third order accurate representation $\tilde{\mu}$, requiring continuity of $\tilde{\mu}$ across panels edges, which involves a generalization of the 5-point differences in equation (6) to nonuniform panels.

The resultant linear system of algebraic equations may be written as

$$\sum_{j=1}^{n+n_c} a_{ij}\, u_j = f_i, \qquad i=1,2,\dots,(n+n_c), \tag{11}$$

where n is the total number of surface panels and n_c the number of components of a multicomponent airfoil. The unknown parameters u_j for $j = 1,\dots,(n+n_c)$ denote $\sigma_j(j = 1,2,\dots,n)$, $c_j(j = 1,\dots,n_c)$ for the source method, where σ_j is the value of the source representation at control point $\vec{x}_j$ and c_j the magnitude of the auxiliary doublet distributions of component j. In case of the doublet method $u_j(j = 1,2,\dots, n+n_c)$ denotes $\mu_j(j = 1,2,\dots,n)$, $\mu_j^c(j = 1,\dots,n_c)$, where μ_j is the value of the doublet representation at control point $\vec{x}_j$ and μ_j^c the value of the doublet representation at the endpoint of the integration interval of component j. The value of the doublet distribution at the beginning of each integration interval is equated to zero without any loss of generality. The ordering of the equations (11) is such that the diagonal elements a_{ii} of the first n equations express the influence of a parameter u_i at the control point $\vec{x}_i$. The last n_c equations of system (11) express the Kutta conditions at the trailing edges of the airfoil components $j(j=1,\dots,n_c)$.

An example solution of the source method applied to a 12 percent thick von Kármán-Trefftz airfoil with a trailing edge angle of 15 degrees (KT0012) is shown in figure 1. Second order convergence of the n-dimensional vector norms of the error in the tangential velocity component of this solution is found, see figure 1.

RELAXATION SCHEMES

The relaxation schemes, also called smoothing operators in MG algorithms, exploit the behavior of the kernels of equations (1) and (2), being like $1/r$ and $1/r^2$, respectively, where r denotes the distance. This behavior tells us that the high frequency components of the singularity distribution have a short coupling range. Neglecting the far field connections between parameters and control points should, therefore, be a sound basis for constructing effective smoothing operators.

On the basis of this consideration we will present two basic classes of relaxation schemes. The first class of schemes is based on incomplete LU factorization (ref. 8) of the approximate system of linear equations that remains after omitting the far field connections, resulting in an approximate factorization (AF). The factors L and U of the LU factorization are forced to have an extensive zero pattern by omitting the nonzero entries which may arise outside of the intended nonzero pattern in the factors L and U during factorization. The present AF scheme is different from the incomplete factorization of algebraic equations associated with the discretization of partial differential equations because there is no need to omit any far field connections in the latter. Moreover the extensive zero pattern in the lower and upper triangular factors need not be the same as the zero pattern of the approximate system of linear equations that remains after omitting the far field connections, although we have chosen these two patterns identical in the present examples.

A second class of relaxation schemes is based on the direct construction of a sparse, approximate inverse. We may construct such an inverse if we approximately satisfy each individual equation of system (11) in its turn by directly solving a very small system of equations, comprising a subset of the entries of the full system, for every unknown parameter. These small systems should be chosen such that they include the coupling range of high frequencies. Thus we relax each equation individually, distributing changes to its neighboring parameters. This second class, which we will call natural relaxation schemes (NRS), is also a general technique. An example of this technique is given in the next section.

FOURIER ANALYSIS

Let a relaxation scheme based on approximate factorization of equation (8) be defined by

$$\sum_{k=-n_a}^{n_a} a_k \mu_{k+i}^{(\nu+1)} = f_i - \sum_{k=-\infty}^{-n_a-1} a_k \mu_{k+i}^{(\nu)} - \sum_{k=n_a+1}^{\infty} a_k \mu_{k+i}^{(\nu)}, \tag{12}$$

where the superscript ν is the iteration index and f_i the right hand side which is given. The pattern of far field connections which are neglected in the approximate equation on the left hand side of equation (12) is denoted by the integers k satisfying $|k| > n_a$. This zero pattern also applies to the factorization. The convergence factor ρ of the θ component, defined in reference 4, of the error in the solution during the iteration procedure (12) is found to be

$$\rho(\theta) = \left|2\sum_{k=1+n_a}^{\infty} a_k \cos(k\theta)\right| \Big/ \left|a_0 + 2\sum_{k=1}^{n_a} a_k \cos(k\theta)\right|, \qquad (13)$$

where the second summation term is to be omitted for $n_a = 0$. This convergence factor as function of the frequency θ is shown in figures 2 and 3 for $n_a = 0,1,$ 2 and 4. It is seen that the convergence of the high frequencies, i.e. $\theta > \frac{\pi}{2}$, is of the order of 10^{-2} for $n_a \geqslant 1$.

The second relaxation scheme (NRS) for equation (8) is based on a sparse inverse, $\tilde{a}_k$, which is defined by

$$\tilde{a}_k = 0 \text{ for } |k| > n_a \text{ and } \tilde{a}_k = g_k \text{ for } |k| \leqslant n_a \qquad (14)$$

where g_k is the solution of

$$\sum_{j=-n_a}^{n_a} a_k\, g_j = \delta_{i0} \quad \text{for} \quad i=-n_a,\dots, n_a, \qquad (15)$$

with $k=|j-i|$ and δ_{i0} = Kronecker delta. Applying this approximate inverse $\tilde{a}_k$ in a residual correction iteration process, see appendix, results in an error amplification matrix given by $I-\tilde{A}A$. The matrix $\tilde{A}A$, denoted by B, is an infinite, symmetric Toeplitz matrix because A and $\tilde{A}$ are infinite, symmetric Toeplitz matrices. This observation allows one to obtain the convergence factor implied by this NRS scheme, similarly to equation (13). One finds:

$$\rho(\theta) = \left|\, 1-b_0 -2\sum_{k=1}^{\infty} b_k \cos(k\theta)\, \right|, \qquad (16)$$

where b_k are the elements of $B = \tilde{A}A$. It may be verified that equations (13) and (16) are identical for $n_a = 0$. The local smoothing factor $\bar{\rho}$ is defined in reference 4 by

$$\bar{\rho} = \max_{\pi/2 \leqslant |\theta| \leqslant \pi} \left\{\rho(\theta)\right\}. \qquad (17)$$

It is a significant measure by which the relative merits of equations (13) and (16) may be judged for $n_a \geqslant 1$. Values of $\bar{\rho}$ for $n_a = 0$, 1, 2, 4 and 7 are given in table 1, for both the 3- and 5-point difference schemes. It may be seen that the smoothing factor of the AF scheme is considerably better than that of the NRS scheme. Comparing the 3- and 5-point differences shows that in case of the AF scheme the 5-point differences result in a lower smoothing factor for $n_a \geqslant 1$.

Applying the source or doublet method to the parallel flow around a circular cylinder (no lift) results in a symmetric, circulant matrix eq. (11), which is denoted by c_k ($k=0, 1, 2, \ldots, (n-1)$), provided we use uniform paneling. The convergence factor of the errors in the solution during the iteration procedure (12) applied to these circulants is found to be

$$\rho(\theta_i) = \left| c_{n/2} + 2 \sum_{k=1+n_a}^{n/2-1} c_k \cos(k\theta_i) \right| \Big/ \left| c_0 + 2 \sum_{k=1}^{n_a} c_k \cos(k\theta_i) \right|, \quad (18)$$

where the discrete frequencies θ_i extend over $2\pi i/n$, $i=0, 1, \ldots, n/2$. This equation (18) turns out to be identical with equation (13) in the limit of $n \to \infty$. The smoothing factors obtained from eq. (18) for the doublet method, as shown in table 1, reflect this observation. For the source method the smoothing factors obtained from eq. (18) are given in table 2. These factors tend to zero in the limit of n going to infinity, which is characteristic for "MG algorithms of the second kind".

Although the results obtained above do give valuable insight in the smoothing properties of relaxation schemes, the local mode analysis cannot take the effects of such practical things as surface slope discontinuities and/or nonuniform paneling of the surface into account. A n-dimensional, discrete Fourier transform of the residue amplification matrix $I-A\tilde{A}$, see appendix, given by

$$G = F\,(I-A\tilde{A})\,F^{-1}, \quad (19a)$$

$$F = f_{kl} = \sqrt{\frac{1}{n}} \exp\,[i\; 2\;\pi\; kl/n], \qquad k,l = 0, 1, \ldots, (n-1), \quad (19b)$$

$$F^{-1} = F^{*} \text{ (the complex conjugate)}, \quad (19c)$$

is more suitable to study these aspects of the smoothing problem. The matrix $\tilde{A}$ in equation (19a) may either be an actual inverse (NRS) or the implied inverse of an AF scheme. Let the row sum of G be defined by

$$\lambda_k = \sum_{l=0}^{n-1} |\,G_{kl}|. \qquad k = 0, \ldots, (n-1), \quad (20)$$

This row sum can be shown to be an upper bound of the convergence factor of the θ_k component of the residue vector in a residual correction iteration process, where $\theta_k = 2\pi k/n$ occupies the unique part of the frequency range for $k = 0, 1, \ldots, n/2$. These considerations allow us to define a global smoothing factor $\bar{\lambda}$ by

$$\bar{\lambda} = \max_{n/4 \leqslant k \leqslant n/2} \left\{\lambda_k\right\}, \quad (21)$$

analogous to the local smoothing factor (eq. (17)). It should be noted however that this global smoothing factor is only an upper bound of the convergence of the high-frequency components because the transformation in eqs. (19b) and (19c) results in a matrix G which is not diagonal, the off-diagonal elements representing the coupling of differing frequencies.

The global smoothing factor of the AF scheme applied to the source and doublet method for the KT0012 profile (see fig. 1) has been determined using a Fast Fourier Transform algorithm. The particular AF scheme used is characterized as before by the far field connections omitted from equation (11) and the subsequent zero pattern forced onto the incomplete LU factorization of the resultant sparse matrix. These two sets, the far field connections a_{ij} and the zero pattern, are defined by the pairs of integers (i,j) satisfying

$$|i-j| > n_a \text{ and } |i+j-n-1| > n_a \text{ and } i \leqslant n \text{ and } j \leqslant n. \qquad (22)$$

Table 3 gives the computed global smoothing factors for n = 32, 64, 128 and 256 and for n_a = 0, 1, 2 and 4. From this table the following conclusions are drawn. The smoothing improves as the dimension of the nonzero pattern n_a is increased. There is no qualitative difference between the source and doublet method, the smoothing factors being approximately independent of the number of panels. This is expected of the doublet method, but the source method results are qualitatively different from those of table 2. Numerical experiments suggest that this qualitative difference is a direct result of the surface slope discontinuity at the trailing edge.

The similarity between the source and doublet method may also be observed from the results plotted in figures 4 and 5 where the row sum λ_k is shown as function of frequency, for n_a = 0 and 1. In case of n_a = 1 a typical smoothing character is observed, i.e. the convergence bound λ_k decreases with increasing frequency.

MULTIGRID ALGORITHM

The multigrid algorithm is described by the following quasi-FORTRAN 77 program, see also reference 9:

```
      SUBROUTINE MG (i, ℓ, u^ℓ, r^ℓ, p, m, q)
      INTEGER p, q
      it (ℓ) = i $ k=ℓ
 1    IF (k. EQ.1) GOTO 4
 2    CALL SMOOTHING (r^k, u^k, p)
      r^(k-1) = RESTRICTION (r^k)
      k=k-1 $ u^k=0 $ it(k)=m
      GOTO 1
 4    CALL DIRECTSOLVER (r^1, u^1)
 5    IF (k.EQ.ℓ) RETURN
      k=k+1
      δu^k = PROLONGATION (u^(k-1))
      r^k=r^k- A^k δu^k    $ u^k =u^k + δu^k
      CALL SMOOTHING (r^k, u^k, q)
      it (k) = it (k) -1
      IF (it (k). EQ. zero) GOTO 5
      GOTO 2
      END 'OF MG'
      SUBROUTINE SMOOTHING (r^k, u^k, pq)
```

```
      INTEGER pq
      DO 1 I=1, pq
      δu^k = RELAXATION SCHEME (r^k)
      r^k = r^k - A^k δu^k    $  u^k = u^k + δu^k
    1 CONTINUE
      RETURN $ END 'OF SMOOTHING'
```

One call to subroutine MG $(i, \ell, u^\ell, r^\ell, p, m, q)$ performs i iterations of the basic multigrid cycle where ℓ is the number of levels, $k\ (=\ell, \ldots, 1)$, u^ℓ the initial solution at level ℓ (taken equal to zero in the present examples) and r^ℓ the corresponding residue at level ℓ. The parameters p, q and m specify the multigrid strategy, m being the number of times the coarse level correction is entered consecutively.

The only operators that remain to complete the description of this MG algorithm are the restriction, prolongation and coarse level equations A^k.

Let a panel distribution on level ℓ be denoted by h_i^ℓ $(i=1, \ldots, n^\ell)$, where h is the panel length and n^ℓ the number of panels. Define the coarse levels recursively by

$$n^{k-1} = \frac{1}{2} n^k \text{ and } h_i^{k-1} = h_{2i-1}^k + h_{2i}^k \quad (i=1, \ldots, n^{k-1}).$$

Let the restriction operator R_{ij}^k $(i=1,\ldots, (n^{k-1}+n_c),\ j=1,\ldots, (n^k+n_c))$ and the prolongation P_{ij}^k $(i=1,\ldots, (n^k+n_c),\ j=1,\ldots, (n^{k-1}+n_c))$ be defined:

$$R_{ij}^k = h_j^k / h_i^{k-1} \quad \text{for } j=1,\ldots,n^k \text{ with } i=\text{IFIX}\left(\frac{j+1}{2}\right) \tag{23a}$$

$$R_{ij}^k = 1 \quad \text{for } j=(n^k+1),\ldots,(n^k+n_c) \text{ with } i=j-n^{k-1} \tag{23b}$$

$$P_{ij}^k = 1 \quad \text{for } i=1,\ldots,n^k \text{ with } j=\text{IFIX}\left(\frac{i+1}{2}\right) \tag{23c}$$

$$P_{ij}^k = 1 \quad \text{for } i=(n^k+1),\ldots,(n^k+n_c) \text{ with } j=i-n^{k-1} \tag{23d}$$

$$R_{ij}^k = P_{ij}^k = 0 \quad \text{for all other pairs of integers } (i,j) \tag{23e}$$

If we let the fine level equations be given by equation (11) and be denoted by A^ℓ, then we may define the coarse level equations recursively by (see ref. 10)

$$A^{k-1} = R^k A^k P^k, \tag{24}$$

which choice has been motivated by the results of Wesseling (Ref. 11), who found the Galerkin coarse grid approximation (eq. 24) to be better than coarse grid discretization of the continuous problem.

MULTIGRID CONVERGENCE

In this section we illustrate the convergence characteristics of the MG algorithm described above by applying it to a number of potential flow problems, restricting ourselves to the source method and the AF scheme.

The first example pertains to the KT0012 profile, shown in figure 1, placed in a uniform flow and at an angle of attack of 20 degrees. The AF scheme used is characterized by the logical expression (22). The convergence history, the norm L_∞ of the residue as function of the number of fine grid residue evaluations, is shown in figure 6 where we have chosen $n = 32$, $\ell = 2$, $p = 1$, $q = 1$, $m = 1$. Here we have defined

$$L_\infty(r^\ell) = \| r^\ell(\nu) \|_\infty \Big/ \| r^\ell\ (\nu = 0) \|_\infty , \tag{25}$$

being the ratio of the maximum norm of the current (ν) residual vector r^ℓ and the maximum norm of the initial (ν=0) residue, i.e. the right hand side of equation (11). The observed convergence factor in figure 6 is about twice as good as the global smoothing factor, which represents an upper bound of the convergence factor of the high-frequencies obtained from a one-level analysis. These findings indicate that $\bar{\lambda}$ is a rather conservative estimate, although it is very realistic with respect to the effect of the nonzero pattern n_a.

Increasing the number of levels to 5, see figure 7, does not change the asymptotic MG convergence rate, although the initial convergence improves somewhat. Using the AF scheme as a classical iteration procedure (fig. 7), i.e. omitting the coarse level corrections, is found to be quite ineffective as should have been expected from the Fourier analysis.

Let us define the computational work associated with one residue evaluation at the finest level as work unit, in order to be able to compare various multigrid strategies. Results are given in table 4, indicate costs ranging from 2.0 to 2.7 work units per 10^{-1} reduction in the maximum norm of the residual vector over a range of strategies p, q with m = 1. Convergence to the level of the truncation error is obtained within 2 MG cycles. MG strategies with m = 2, i.e. entering the coarse level correction two times consecutively, are found to be computationally less efficient, see table 4.

The second example illustrates the convergence of the MG algorithm when applied to the problem of a wing plus flap configuration shown in figure 8. The convergence history is shown in figure 9. It is observed that the AF smoothing is quite effective for $n_a \geqslant 1$, although we have just repeated the nonzero pattern for single-component airfoils given in expression (22) by applying it to each submatrix corresponding to the wing alone and flap alone for $(i,j) \leqslant n$. No zero entries for $i > n$ or $j > n$ have been created. This zero pattern, which neglects all connections in the matrix between wing and flap for $(i,j) \leqslant n$, results in acceptable AF smoothing for a gap of 2.6 percent. However in the limit of vanishing gap size for a fixed paneling we would, of course, have to take some nonzero entries representing the flap/wing connections into account if acceptable AF smoothing is to be retained in this limit, as has been confirmed by numerical experiments. These observations clearly lead to the requirement that all near field connections have to be taken into account during the construction of a particular AF smoothing scheme.

CONCLUSION

Approximate factorization (AF) relaxation schemes provide a smoothing capability that allows one to construct a fast multigrid method for solving integral equations of the second as well as equations of the first kind.

The local mode analysis of Brandt (Ref. 4) is applied for the special cases of an unbounded flat plate and circular cylinder and predicts the qualitative difference between multigrid problems of the first and second kind, where the former has a smoothing factor independent of h and the latter a smoothing factor proportional to h. For more realistic geometries, having surface slope discontinuities such as airfoils, Fourier analysis predicts no qualitative difference between smoothing factors obtained with the AF scheme when applied to integral equations of either the first or second kind.

Numerical experiments show that convergence to the level of the truncation error of a second order accurate integral method can be obtained within 2 MG cycles.

REFERENCES

1. Hackbusch, W.; Die schnelle Auflösung der Fredholmschen Integralgleichung zweiter Art, Report 78-4, Universität zu Köln, 1978.

2. Schippers, H.; The automatic solution of Fredholm equations of the second kind, Report NW 99/80, Mathematisch Centrum Amsterdam, 1980.

3. Slooff, J.W.; Requirements and developments shaping a next generation of integral methods, Paper presented at IMA Conference on numerical methods in aeronautical fluid dynamics, University of Reading, UK, 1981.

4. Brandt, A.; Multi-level adaptive solutions to boundary-value problems, Math. of Comp. Vol. 31, No. 138, April 1977, pp. 333-390.

5. Hess, J.L.; The use of higher-order surface singularity distributions to obtain improved potential flow solutions for two-dimensional lifting airfoils, Computer Methods in Appl. Mech. and Eng., Vol.5, 1975, pp. 11-35.

6. Johnson, F.T. and Rubbert, P.E.; Advanced panel-type influence coefficient methods applied to subsonic flows, AIAA Paper No. 75-50, 1975.

7. Hess, J.L.; Consistent velocity and potential expansions for higher order surface singularity methods, MDCJ6911, 1975.

8. Meijerink, J.A. and Van der Vorst, H.A.; An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. of Comp., Vol. 31, No. 137, 1977, pp. 148-162.

9. Hackbush, W.; Introduction to multigrid methods for the numerical solution of boundary value problems, Von Karman Institute for Fluid Dynamics, Lecture Series 1981-5, 1981.

10. Frederickson, P.O.; Fast approximate inversion of large sparse linear systems, Lakehead University, Mathematics Report 7-75, 1975.

11. Wesseling, P.; Theoretical and practical aspects of a multigrid method, Delft University of Technology, Department of Math. Report NA-37, 1980.

APPENDIX

Let a residual correction iterative scheme to solve the matrix equation, Au = f, be given by

$$\delta u^{(\nu+1)} = \tilde{A}\, r^{(\nu)}, \tag{A-1}$$

$$u^{(\nu+1)} = u^{(\nu)} + \delta u^{(\nu+1)}, \tag{A-2}$$

and

$$r^{(\nu+1)} = r^{(\nu)} - A\, \delta u^{(\nu+1)}, \tag{A-3}$$

with the iteration index ν (=0, 1, 2,...). Setting the initial solution $u^{(0)}$ equal to zero results in an initial residue $r^{(0)}$ equal to the right hand side f. The column vector $\delta u^{(\nu+1)}$ is the correction to the approximate solution $u^{(\nu)}$ and $\tilde{A}$ is an approximate inverse of A. This inverse is either constructed (NRS scheme) or implied by the AF scheme. For the latter case we have $\tilde{A} = (LU)^{-1}$.

This iterative scheme results in an error amplification matrix, M_e, defined by

$$u^{(\nu+1)} - u = M_e\,(u^{(\nu)} - u), \quad \nu=0,\ 1,\ 2,\ldots, \tag{A-4}$$

which reads: $M_e = I - \tilde{A}A$. The corresponding residue amplification matrix, defined by

$$r^{(\nu+1)} = M_r\, r^{(\nu)}, \quad \nu = 0,\ 1,\ 2,\ldots, \tag{A-5}$$

is equal to: $M_r = I - A\tilde{A}$. In case A and $\tilde{A}$ are either circulant matrices or inifinite Toeplitz matrices one finds $\tilde{A}A = A\tilde{A}$ and $M_e = M_r$. For a more general matrix A resulting from the airfoil problem, equation (11), we have chosen to analyze the residue amplification matrix M_r. The choice of the zero pattern (22) in the AF scheme when applied to the $(n + n_c)$-dimensional matrix equation (11) results in a n-dimensional matrix $M_r = I - A\tilde{A}$.

TABLE 1.- THEORETICAL SMOOTHING FACTORS FOR DOUBLET METHOD WITH NORMAL-VELOCITY BOUNDARY CONDITIONS

	Unbounded flat plate				Circ. cylinder	
	3-p[a] NRS[c] eq.(16)	5-p[b] NRS eq.(16)	3-p AF[d] eq.(13)	5-p AF eq.(13)	5-p AF eq.(18) n=512	5-p AF eq.(18) n=128
n_a =0	0.415	0.797	0.415	0.797	0.795	0.790
n_a =1	0.406	0.534	0.163	0.0227	0.0232	0.0249
n_a =2	0.321	0.421	0.0422	0.0251	0.0252	0.0255
n_a =4	0.261	0.326	0.0156	0.0102	0.0102	0.0103
n_a =7	0.202	0.256	0.0052	0.0037	0.0037	0.0037

[a] 3-point differences
[b] 5-point differences
[c] Natural Relaxation Scheme
[d] Approximate Factorization

TABLE 2.- THEORETICAL SMOOTHING FACTORS FOR SOURCE METHOD APPLIED TO CIRCULAR CYLINDER [equation (18); n_a=0]

n=128	n=256	n=512
0.00792	0.00397	0.00199

TABLE 3.- GLOBAL SMOOTHING FACTORS OF APPROXIMATE FACTORIZATION (AF) FOR SOURCE AND DOUBLET METHODS APPLIED TO KT0012 PROFILE

	SOURCE METHOD				DOUBLET METHOD (5-p)			
n=	32	64	128	256	32	64	128	256
n_a =0	1.35	1.36	1.35	1.34	1.75	1.83	1.87	1.89
n_a =1	0.49	0.54	0.55	0.56	0.29	0.24	0.22	0.23
n_a =2	0.32	0.36	0.34	0.34	0.24	0.23	0.22	0.22
n_a =4	0.19	0.23	0.24	0.23	0.20	0.13	0.11	0.10

TABLE 4.- WORK UNITS PER 10^{-1} REDUCTION [(a)] IN THE RESIDUE OVER A RANGE OF STRATEGIES. [AF smoothing, $n_a=1$; KT0012, $\alpha=20^{\circ}$, n=256, $\ell=5$, i=4]

p	q	m	Work units per digit	p	q	m	Work Units per digit
0	1	1	2.7	2	1	1	2.5
0	2	1	2.2	2	2	1	2.3
1	0	1	2.2				
1	1	1	2.2	0	1	2	4.3
1	2	1	2.2	1	0	2	5.3
2	0	1	2.0	1	1	2	3.6

[a]Average values over last 2 MG cycles of a total of 4 MG cycles.

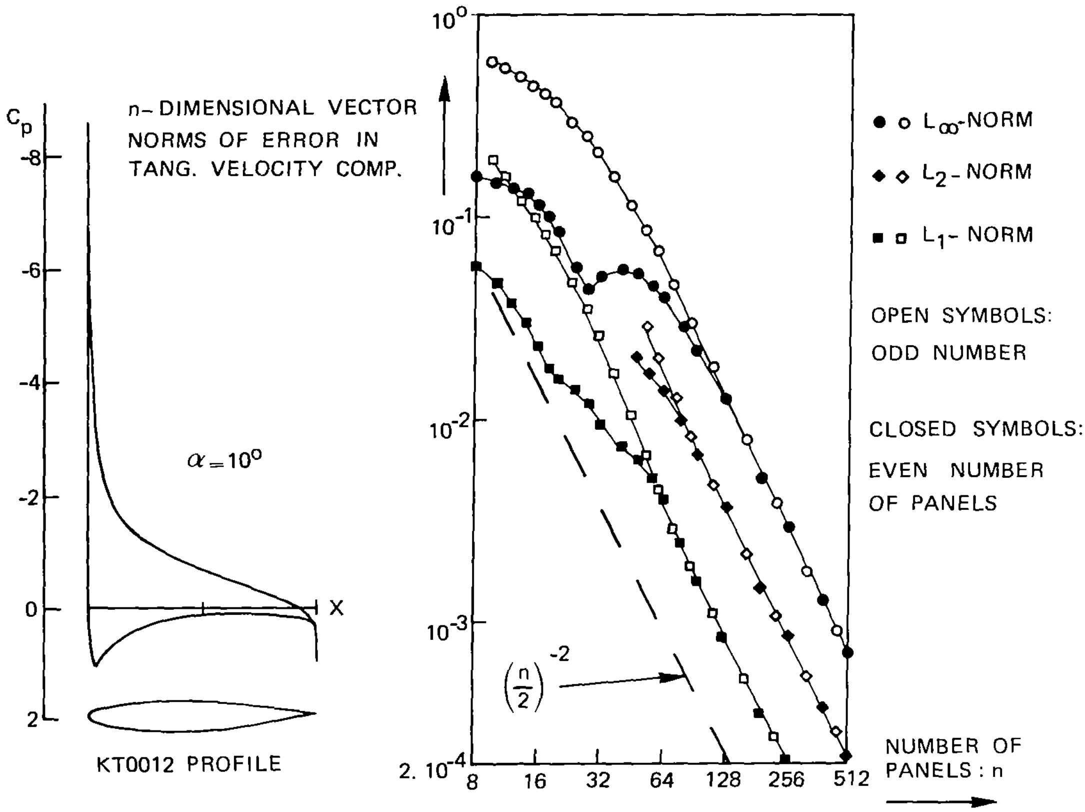


Figure 1.- Pressure distribution of 12 percent thick Kármán-Trefftz profile with 15° trailing edge angle; Discretization error of this solution as function of the number of panels.

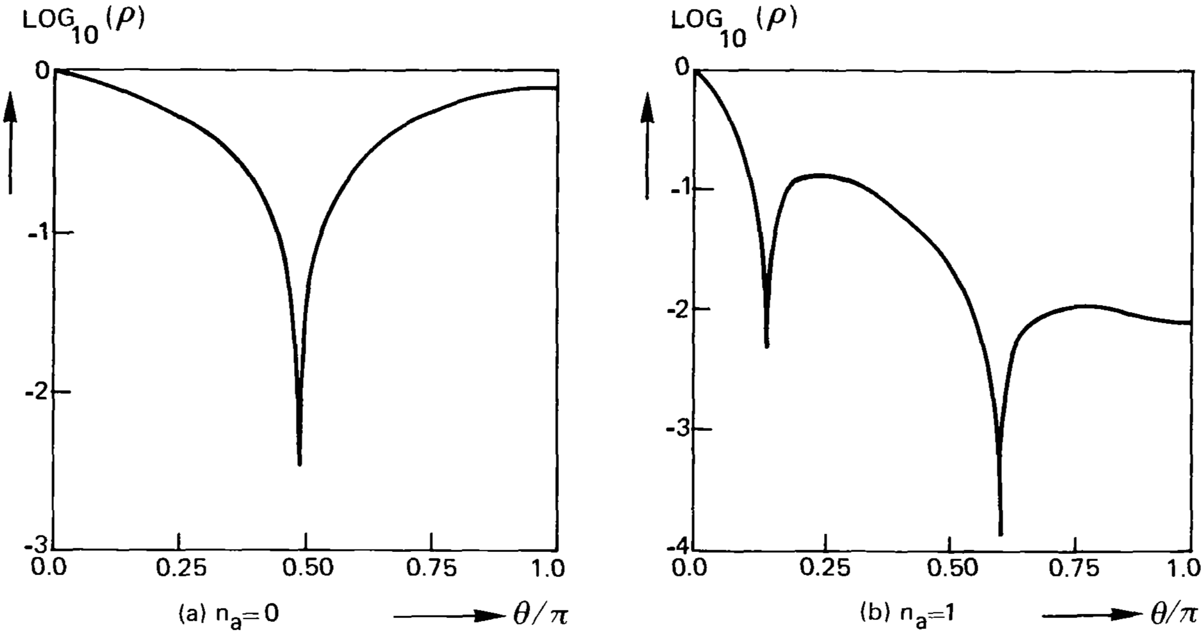


Figure 2.- Convergence factor as function of frequency for doublet method (equation (13) with 5-point differences); n_a=0 and 1, AF scheme.

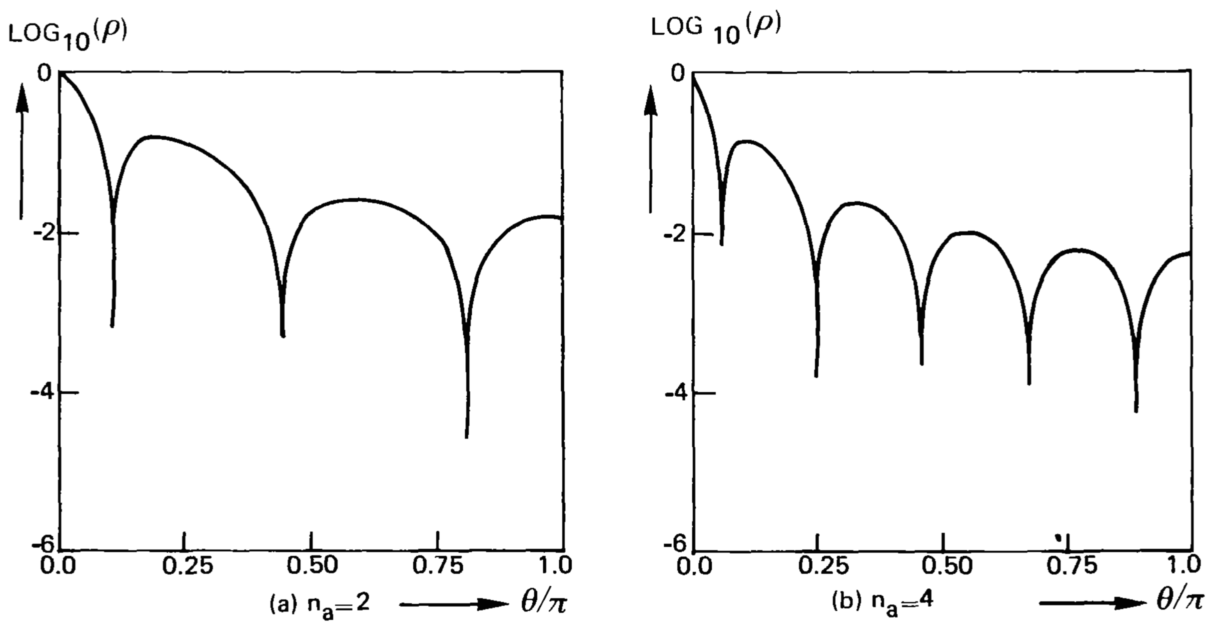


Figure 3.- Convergence factor as function of frequency for doublet method (equation (13) with 5-point differences); n_a=2 and 4, AF scheme.

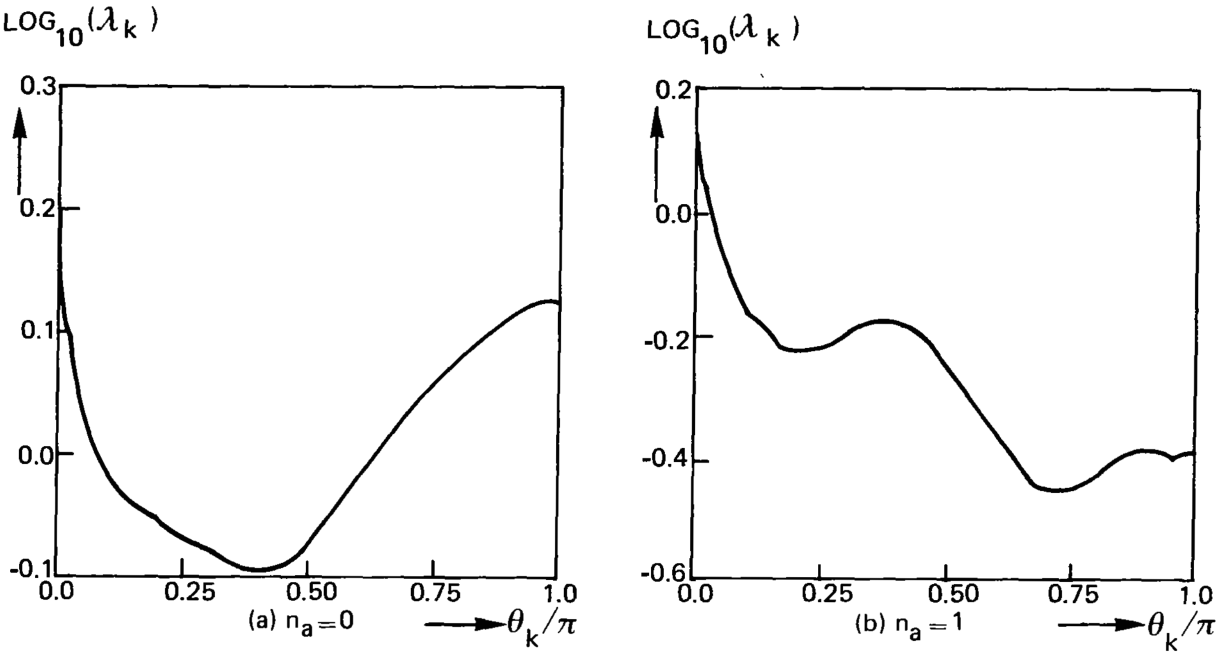


Figure 4.- Row sum of amplification matrix G as function of frequency for source method; KT0012, n=256, n_a=0 and 1, AF scheme.

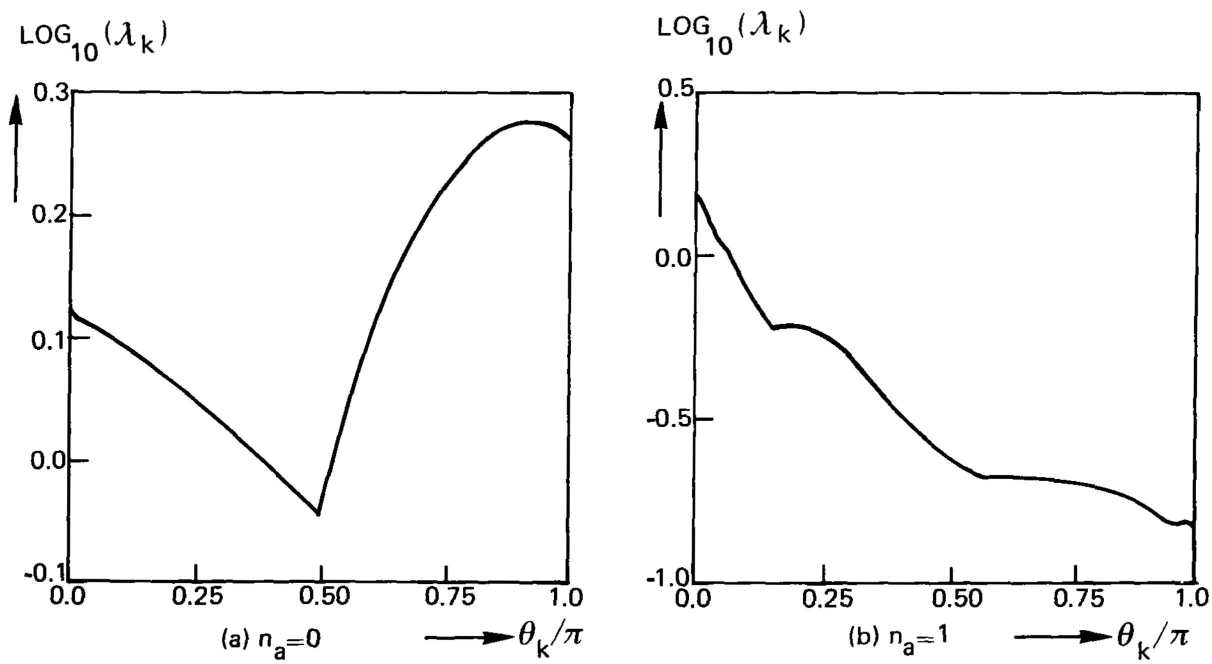
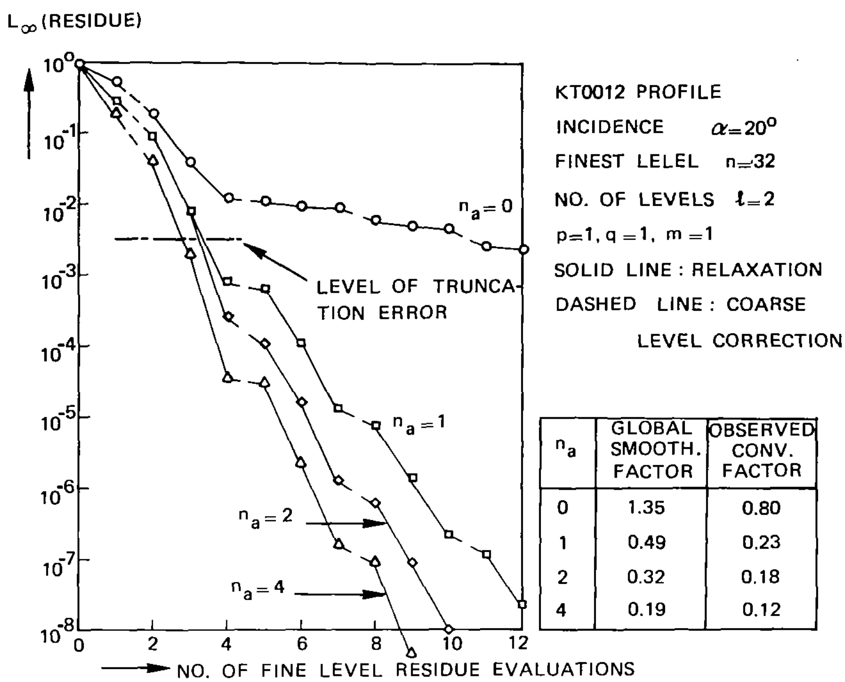


Figure 5.- Row sum of amplification matrix G as function of frequency for doublet method; KT0012, n=256, n_a=0 and 1, AF scheme.



n_a	GLOBAL SMOOTH. FACTOR	OBSERVED CONV. FACTOR
0	1.35	0.80
1	0.49	0.23
2	0.32	0.18
4	0.19	0.12

Figure 6.- Convergence history of MG algorithm; 2 levels, source method, AF scheme.

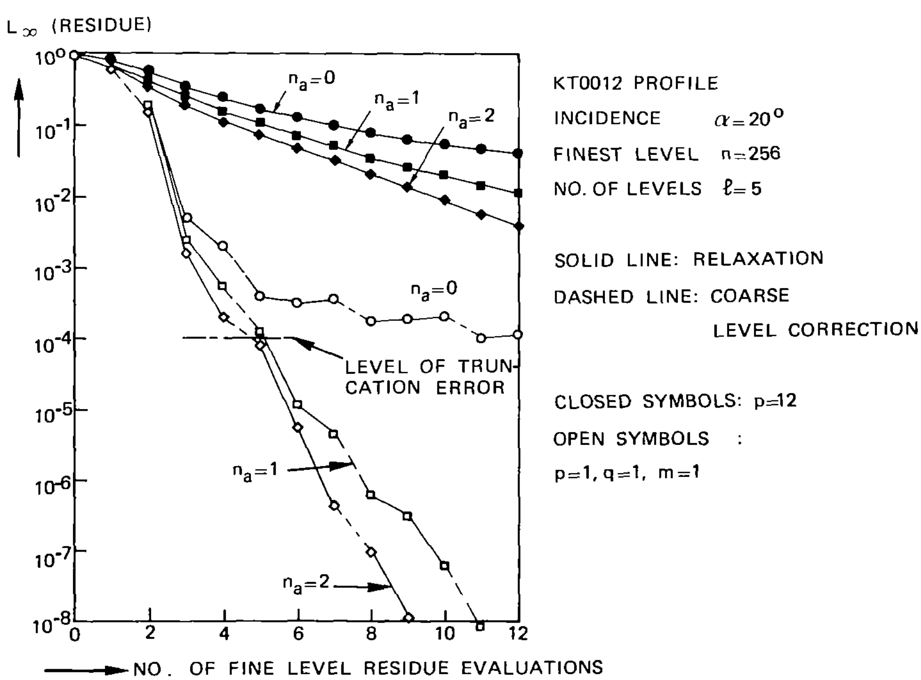


Figure 7.- Convergence history of MG algorithm; 5 levels, source method, AF scheme.

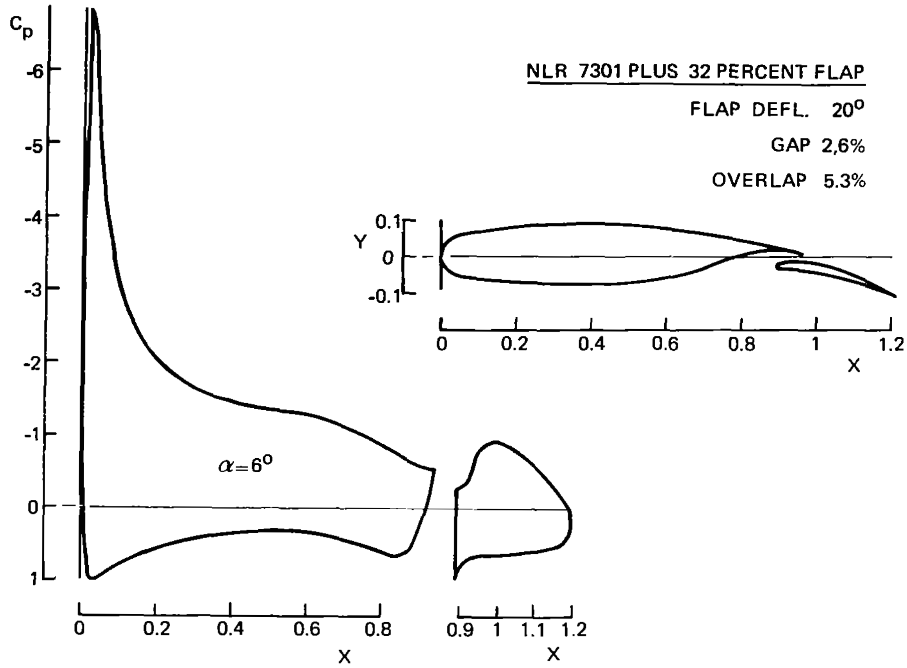


Figure 8.- Pressure distribution of NLR 7301 plus 32 percent flap at 6 degrees incidence.

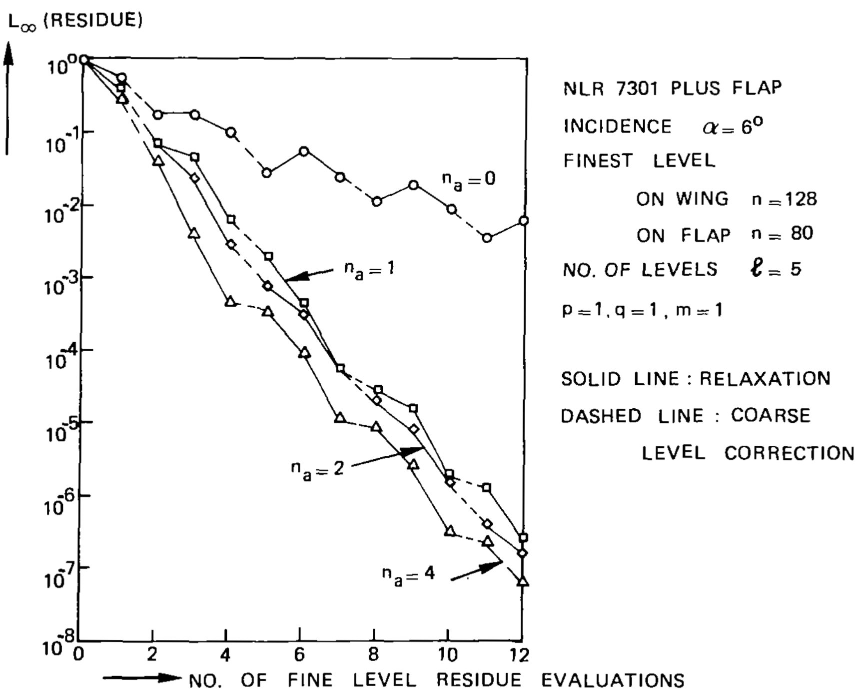


Figure 9.- Convergence history of MG algorithm; 5 levels, source method, AF scheme.

UNIGRID METHODS FOR BOUNDARY VALUE PROBLEMS WITH NONRECTANGULAR DOMAINS

W. Holland
National Center for Atmospheric Research

S. McCormick
J. Ruge
Colorado State University

1. Introduction

Multigrid methods are generally very effective for solving differential boundary value problems. This is true because the smooth error, which is slow to converge during relaxation, is reduced by iterating on the problem projected onto coarser grids, where relaxation is both cheaper and more efficient. Fine grid relaxation can then be viewed as an attempt to eliminate the high frequency error.

In lieu of coarse grid iterations, one can, in fact, modify the fine grid relaxation process in order to reduce the smooth error directly on the fine grid (i.e., without the use of coarser grids at all). Under certain assumptions (see section 3), the resulting method, so-called *unigrid* [1], is theoretically equivalent to conventional multigrid but has significantly different computational characteristics. For example, unigrid requires less storage and shorter code, but significantly more arithmetic work. More importantly, it is much easier to apply to a given problem because most of the design work for the grid transfers and coarse grid operators is automatic. Thus, existing software packages that solve possibly very complex problems by SOR, for example, can be easily modified for application of unigrid. This can usually be done by making a few changes in the relaxation routine without impacting any of the other software routines or data structures. These features make unigrid effective as a multigrid software simulator for quick and easy determination of the applicability of multigrid to a given problem.

Unigrid is developed in section 2, its relationship to multigrid is described in section 3, some simple theory is presented in section 4, and its use is illustrated with a North Atlantic basin oceanographic model problem in section 5. This application demonstrates how unigrid (and, hence, multigrid) can be used efficiently with vector computers on problems with irregular domains.

2. Unigrid

Assume given the d-dimensional operator equation:

$$AU = F, \quad U \in H_1, \tag{2.1}$$

where A: $\mathcal{H}_1 \to \mathcal{H}_2$ is a linear operator and $\mathcal{H}_1$ and $\mathcal{H}_2$ are appropriate Hilbert spaces of functions defined on a region Ω in R^d, $d \geq 2$. Assume that (2.1) admits discretizations by a family of matrix equations, parameterized by *admissible grid sizes* $h > 0$ and given by:

$$A^h U^h = f^h, \quad U^h \in \mathcal{H}^h, \tag{2.2}$$

where $\mathcal{H}^h = R^{n^h}$ and n^h is an integer (approximately proportional to h^{-d}). Upper case U^h will denote the exact solution of (2.2) and lower case u^h its approximation. The grid transfers are full rank linear operators, represented by $I_{\bar{h}}^h$: $\mathcal{H}^{\bar{h}} \to \mathcal{H}^h$, that satisfy the consistency condition $I_{\bar{h}}^h = I_{h'}^h I_{\bar{h}}^{h'}$ for admissible $\bar{h}$, h', h when $\bar{h} < h' < h$ or $\bar{h} > h' > h$.

The objective is to reduce the error from a current approximation u^h in the subspace defined by a set of directions $\mathcal{D}^h = (d_1, d_2, \ldots d_n) \subset \mathcal{H}^h$. Letting $D^h = [d_1, d_2, \ldots d_n]$, then a Ritz projection can be performed that corrects u^h by a function in the space of $\mathcal{D}^h$, so that the projection of the resulting residual over the subspace is zero. This leads to the problem of finding some $s = (s_1, \ldots, s_n)^T$ so that:

$$D^{h^T} [A^h (u^h + D^h s) - f^h] = 0$$

This can be rewritten as:

$$D^{h^T} A^h D^h s = D^{h^T} [f^h - A^h u^h]$$

Gauss-Seidel relaxation on this system with some initial approximation s and a new approximation $\bar{s}$ can be written as:

$$\bar{s}_i = (f^h - A^h u^h - \sum_{j>i} s_j A^h d_j - \sum_{j<i} \bar{s}_j A^h d_j, d_i)/(A^h d_i, d_i)$$

$$i = 1, 2, \ldots n^h$$

u^h can then be corrected by:

$$u^h \leftarrow u^h + D^h \bar{s}$$

If A^h is linear, then corrections can be made to u^h directly, rather than to s, resulting in the *directional iteration*

$$(2.3) \qquad u^h \leftarrow u^h + \frac{(f^h - A^h u^h,\ d_i)}{(A^h d_i,\ d_i)}\, d_i$$

where left arrow denotes replacement. Rewriting (2.3) as:

$$(2.4) \qquad u^h \leftarrow G^h\ (u^h,\ d)$$

then one *sweep* with initial guess u^h consists of iterating with (2.4) in sequence over d_k, $k = 1, 2, \ldots n^h$. For example, Gauss-Seidel is specified by the choice $d_k = e_k^h$, the k^{th} coordinate vector in $\mathcal{H}^h$.

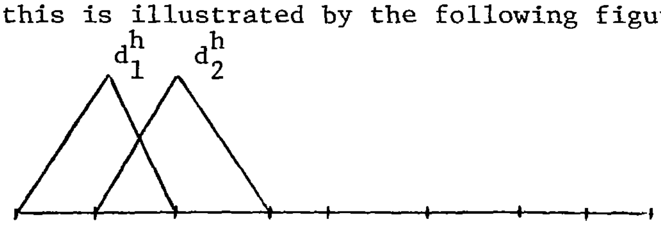
To define unigrid for a given admissible grid size $h = H_0$, suppose $m \geq 1$ is an integer so that $H_q = 2^q H_0$ are admissible, $q \leq m$. Now define the direction sets for unigrid according to:

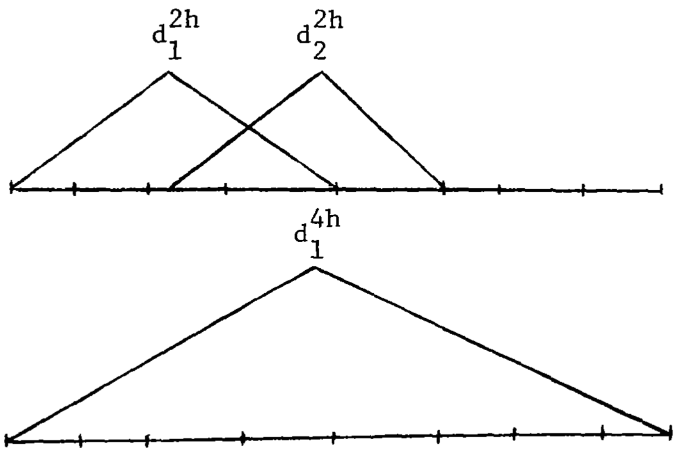
$$(2.5) \qquad \mathcal{D}_q^h = (d_1^{H_q},\ d_2^{H_q},\ \ldots,\ d_{n^{H_q}}^{H_q}),\ 0 \leq q \leq m,$$

where $d_k^{H_q} = I_{H_q}^h e_k^{H_q}$. Thus, the directions on *level* q are just the relaxation directions on grid H_q transferred to grid $h = H_0$.

One of the many possible unigrid schemes is described in terms of the relaxation parameters ν and ν_c and the cycling parameter μ. The unigrid cycles are then defined recursively by: one unigrid cycle on level q consists first of ν unigrid relaxation sweeps via (2.4) with directions $d_k^{H_q}$, $k = 1, 2, \ldots n^{H_q}$, followed for $q < m$ by μ cycles on level q+1 and for $i = m$ by ν_c more sweeps via (2.4).

Remark The directions defining unigrid depend not directly on the operator A but rather on the domain Ω. Using linear interpolation, then these directions $d_i^{\bar{h}}$ are in fact the i^{th} grid $\bar{h}$ coordinate vectors interpolated to grid H_0. In one-dimension, this is illustrated by the following figures.





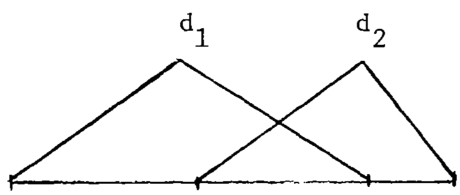
For rectangular Ω in two-dimensions, each direction is a product of two such functions, one in x and the other in y, resulting in the "tent" function described as follows.

With the usual double subscript notation and $n^h = N^h \times N^h$, then $d^h_{k,\ell} = e^h_{k,\ell}$ for $h = H_0$, $1 \le k, \ell \le N^h$. The coarse grid directions are defined so that the i,j component of $d^{H_q}_{k,\ell}$ is:

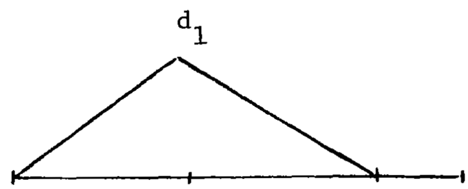
$$(2.7) \quad d^{H_q}_{k,\ell}(i,j) = \begin{cases} (2^q - |k-i|)(2^q - |\ell-j|), & |k-i|,|\ell-j| \le 2^q \\ 0, & \text{otherwise.} \end{cases}$$

This assumes that the point denoted by (k,ℓ) is a point of the H_q grid.

In irregular regions where boundaries do not lie on coarse grid lines, there are several options possible for treating these boundaries. The most obvious, which is analogous to the usual multigrid approach, is to define the directions as the interpolated coarse grid coordinate vectors and use the (zero) boundary conditions properly in interpolation. This is illustrated in one-dimension by d_2 as in:



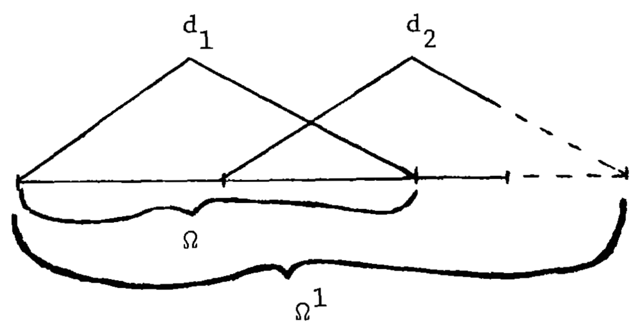
Note that this requires special handling of the coarse grid points that are adjacent to the boundary. Another approach is simply to ignore those directions which would overlap the boundary so that d_2 is suppressed as in:



In section 5, this will be referred to as the *contracted boundary* method.

This means that some points near the boundary are not corrected by smooth error iterations, so the danger is that convergence is slowed (see section 5).

Another possibility is to enlarge the region, Ω, so that it is aligned with the coarse grid directions, but ignore correcting that part of the expanded region, Ω, that does not lie in the interior of Ω. This is illustrated by:



Note that this method, which is here called the *expanded boundary* approach, does not require extra information at the boundaries so the directions can be computed once for each grid over the entire domain Ω and stored in the form of a matrix stencil.

3. Multigrid

One multigrid cycle on problem (2.2) with present approximation u^h, right-hand side $\bar{f}^h$, and $h = H_q$, is denoted by $MG_h(u^h,\bar{f}^h)$ and defined recursively by:

For $q = m$, $MG_h(u^h, \bar{f}^h)$ consists of $\nu + \nu_c$ relaxation sweeps via (2.4) with directions e_k^h, $k = 1, 2, \ldots, n^h$.

For $q < m$, $MG_h(u^h, \bar{f}^h)$ consists of:

<u>Step 1</u>. Perform ν relaxation sweeps via (2.4) with directions e_k^h, $k = 1, 2, \ldots, n^h$.

Step 2. Let $r^h = \bar{f}^h - A^h u^h$, $r^{2h} = I^{2h} r^h$, $u^{2h} \leftarrow 0$, and perform μ grid 2h cycles via $MG_{2h}(u^{2h}, \bar{f}^{2h})$ with $\bar{f}^{2h} = r^{2h}$.

Step 3. Set $u^h \leftarrow u^h + I^h_{2h} u^{2h}$.

Multigrid is theoretically equivalent to unigrid if, as is henceforth assumed, the formulation of the coarser grid equations satisfies the *variational conditions:*

$$A^{2h} = I_h^{2h} A^h I_{2h}^h \tag{3.1}$$

$$I_h^{2h} = \alpha^h (I_{2h}^h)^T, \tag{3.2}$$

where α^h is a scaler. To see this, consider the following *immediate replacement* multigrid algorithm. The method depends directly on the fine grid right-hand side f^{H_0} and its cycles on grid $h = H_q$ are denoted by $MGIR_q(f^0)$, where q is used in place of H_q as a subscript or superscript. It is characterized as a modification of conventional multigrid applied to (2.2) in which all coarse grid changes are immediately reflected in the fine grid approximation and the fine grid residual is recomputed and used to redefine the coarse grid equations. The algorithm is defined in terms of $MGIR_q(f^0)$ by:

For $q = m$, $MGIR_q(f^0)$ consists of performing $\nu + \nu_0$ relaxation sweeps via:

$$u^0 \leftarrow u^0 + \frac{\langle I_0^q(f^0 - A^0u^0),\ e_k^{H_q}\rangle}{\langle A^q e_k^{H_q},\ e_k^{H_q}\rangle} I_q^o e_k^{H_q} \qquad k = 1, 2, \ldots, n^q.$$

For $q < m$, $MGIR_q(f^0)$ consists of ν relaxation sweeps via (3.3) followed by μ levels $q - 1$ cycles via $MGIR_{q-1}(f^0)$.

Note that the immediate fine grid correction is incorporated in the relaxation scheme. This scheme on a level $q > 0$ is just (2.3) with $u^p = 0$, $0 < p \leq q$, and $r^q = I_0^q(f^0 - A^0u^0)$, followed by interpolation of the correction directly to the finest grid.

It is not difficult to see [1] that MGIR is fully equivalent to MG under condition (3.1) - (3.2). This is done by noting what the status of intermediate MG calculations would be if coarse grid changes were immediately reflected in the fine grid approximation. By examining the iterative formulae, it is easy to see that MGIR and unigrid are identical, from which it follows that multigrid designed according to the variational conditions (3.1) - (3.2) is theoretically equivalent to unigrid.

Unigrid code is typically very compact, partly because it lacks the modular structure of multigrid software. This is one reason that unigrid code can be developed very quickly. Also, there are fewer design choices with unigrid, since the coarse grid and grid transfer operators are automatically determined. This also adds to ease of programming, but restricts the flexibility of the method. The design of unigrid also guarantees convergence independently of the choice of the coarse level iteration directions and cycling scheme, so mistakes may slow convergence but do not result in divergence as often as for multigrid.

This ease of programming and small program size makes unigrid an effective method to test the convergence behavior of multigrid for many application problems. It can easily be used to replace the usual relaxation of direct solvers in existing programs in order to perform such a feasibility test. Of course, the amount of work involved makes any comparison of solution time meaningless, but actual multigrid efficiency can be determined by applying the usual multigrid operation counts to the unigrid cycling scheme. Since the methods are equivalent in terms of results when multigrid is implemented according to (3.1) - (3.2), then unigrid will accurately represent the numerical performance of such a variationally formulated multigrid scheme.

4. Theory

Assuming that A^h is symmetric and positive definite, define the *energy inner product* and *norm* on H^h by:

$$\langle x^h, y^h \rangle_{A^h} = \langle A^h x^h, y^h \rangle$$

and

$$\|x^h\|_{A^h} = \langle A^h x^h, x^h \rangle^{1/2},$$

respectively. Let $\mathcal{W}^h_\lambda$ denote the set of all A^h-unit eigenvectors of A^h whose eigenvalues are no larger than λ and let γ^h_λ denote its A^h-orthogonal complement. Let G^h denote one pass of (2.4) over the fine grid directions $\mathcal{D}^h$ and assume it spans $\mathcal{H}^h$. For each integer $\nu \geq 1$, define $G^h_{\lambda,\nu}$ as the restriction of $(A^h)^{-1/2}((G^h)^\nu)^T A^h (G^h)^\nu (A^h)^{-1/2}$ to γ^{h}_λ. (For Jacobi-type versions of (2.4), this latter operator simplifies to $(G^h)^{2\nu}$.) Then, with 2m the degree of the differential operator in (2.1), assume (cf., [2]):

A1. There exists constants $a > 0$ and $c_0 < \infty$ independent of h and so that:

$$\|w^h - R(I_{2h}^h)\|^2_{A^h} \leq c_0 (\lambda h^{2m})^a$$

for all admissible h and all $w^h \in \mathcal{W}_\lambda^h$, where $R(I_{2h}^h)$ is the range of I_{2h}^h. (Note that $R(I_{2h}^h) = \text{Span}\ (\mathcal{D}_1^h)$, the coarse grid directions.)

A2. There exist constants $c_1, c_2 > 0$ with $c_1 < 1$ and $c_2 \leq (c_1 + 1)h^{-2m}/\rho(A^h)$, where $\rho(A^h)$ is the spectral radius of A^h, so that:

$$\rho(G_{\lambda,\nu}^h) \leq \max\{c_1^\nu, |1 - c_2 \lambda h^{2m}|^\nu\}.$$

THEOREM. Suppose $\mu > 1$ and m and ν_0 are so that the error from coarsest level does not significantly contribute to the finest level error. Then there exists a ν independent of h so that unigrid converges to the solution of (2.2) by a fixed linear rate independent of h.

PROOF. This theorem follows from the results of section 3 that relate unigrid to multigrid and from the theory of [2] slightly modified to account for the class of relaxation methods depicted in (2.4).

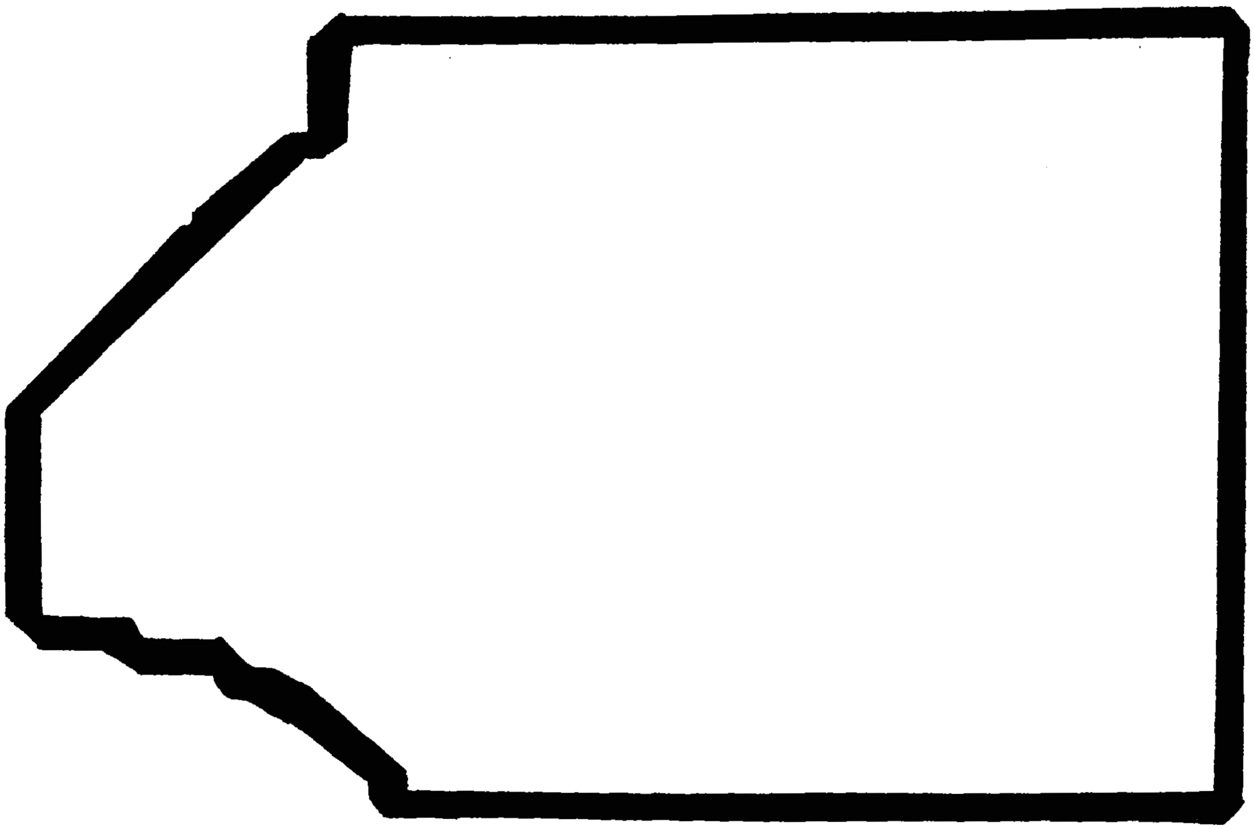
Relaxation does not generally minimize the residual error, although it should approximately. In fact, when direct application of unigrid to (2.2) exhibits convergence but does not monotonically reduce the residual error on the coarse levels, this is a signal that the directions for relaxation are improperly defined. They should be chosen to approximate the *smooth* eigenvectors of A^h, that is, those that belong to the lower end of its spectrum. This would ensure that relaxation quickly eliminates the *oscillatory* eigenvector components of the error with little effect on the smooth ones. Since the spectrum of A^h that corresponds to these oscillatory components is *relatively* narrow, then there is a close relationship between error in the energy norm, for which relaxation is a minimizer, and the residual error norm. The residual norm is not generally *minimized* by relaxation, but a proper choice of directions coupled with a good smoothing rate ensures that it will be monotonically *reduced*.

5. Numerical Results

This section contains a report on numerical experiments with unigrid applied to the solution of the model problem:

$$(5.1) \qquad \begin{aligned} -\nabla^2 u + \lambda u &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

where Ω is an irregular domain used to describe the North Atlantic basin. In this case, Ω is rectangular on three sides but irregular on the fourth, as depicted by:



λ is a given function which is set to the constant 64 in the following experiments. (Such a value for λ results in strong positive definitiveness of the operator in (5.1), leading to very fast convergence rates for multigrid. However, such a value is fairly realistic for this application and sharply depicts the disadvantage of using the contracted boundary method described in section 3.) In these experiments, the fine grid spacing is $h = 0.0625$ and the rectangle encompassing Ω is $[0,3] \times [0,2]$. In each case, a very simple grid cycling scheme with four grids is used, where each cycle involves three relaxations, each performed in turn on grids 8h, 4h, 2h, and h. Four cycles are made for each of the three problems, with $u = 0$ as the initial guess. The usual central fine point stencil was used to discretize (5.1).

The main feature of the discretization of (5.1) is that the boundary is enforced to pass through grid h vertices. Although this represents only an approximation to the actual boundary (of reduced order), it has conservative properties that are not easily obtained any other way. More specifically, conservation of kinetic energy, vorticity, and enstrophy in a dissipationless finite difference discretization of atmospheric diffusion problems can be easily guaranteed when the grid points and irregular boundary points coincide (cf., [3]). However, although this is an advantage when used on a vector processor, coarse grids in the usual multigrid process will not generally share this simplified property. The question then is whether or not one of the means for preserving this feature on coarser grids (namely, boundary contraction or expansion) will maintain the efficiency of the usual multigrid process. Such is the objective of the experiments reported in this section.

To compare the contracted and expanded boundary methods with the usual multigrid, unigrid was used on the Cray 1 at NCAR as a simple tool to simulate multigrid performance. Instead of comparisons with the usual multigrid on the irregular region, it was much simpler to compare the two methods with the analogous (i.e., naturally extended) problem defined on the entire rectangle $[0,3] \times [0,2]$. Thus, a function U on this rectangle was chosen to determine f and the usual unigrid algorithm was run on the full rectangle. The results are depicted in the first column of the table. Both the contracted and expanded boundary methods were also tried with the same f, but with f restricted to the irregular region Ω. The results are depicted in the second and third columns of the table, respectively. Note the severe degradation in convergence for the contracted boundary method. As might be expected, however, there is almost no loss of efficiency with the expanded boundary approach.

Although these are admittedly very limited experiments, they represent the numerical experience with several such tests that were conducted. Generally, although full multigrid (FMG) vastly and expectedly improves the performance of the contracted boundary method, it remains somewhat less efficient than conventional multigrid. On the other hand, the expanded boundary method seems generally as (or nearly as) efficient, and therefore, preferrable to the usual multigrid approach, especially for use on vector processors such as the Cray 1.

Cycle Number	Relaxation Level	DYNAMIC RESIDUAL ERROR: Multigrid on [0,3] x [0,2]	Contract Multigrid On Ω	Expanded Multigrid On Ω
1	8h	7.402E+03	6.027E+03	7.418E+03
		1.197E+03	1.147E+03	2.465E+03
		2.745E+02	1.753E+02	1.037E+03
	4h	2.617E+03	2.384E+03	2.364E+03
		7.758E+01	8.864E+01	2.598E+02
		3.416E+00	3.927E+00	2.187E+02
	2h	4.270E+02	5.663E+02	4.311E+02
		8.350E+01	1.359E+02	1.002E+02
		2.200E+00	4.139E+01	2.602E+01
	h	1.090E+02	2.716E+02	1.437E+02
		1.784E+01	8.348E+01	2.630E+01
		7.771E+00	3.836E+01	9.057E+00
2	8h	3.620E+01	3.602E+01	3.749E+01
		9.004E+00	6.381E+00	9.451E+00
		2.383E+00	1.145E+00	4.826E+00
	4h	3.236E+01	3.607E+01	3.166E+01
		8.235E-01	1.383E+00	4.491E+00
		3.198E-02	6.425E-02	2.348E+00
	2h	6.099E+00	4.050E+01	6.358E+00
		1.139E+00	1.006E+01	1.373E+00
		2.956E-01	3.073E+00	3.161E-01
	h	1.363E+00	2.030E+01	2.116E+00
		2.232E-01	9.095E+00	4.554E-01
		1.048E-01	4.764E+00	1.694E-01

Cycle Number	Relaxation Level	DYNAMIC RESIDUAL ERROR: Multigrid On [0,3] x [0,2]	Contract Multigrid On Ω	Expanded Multigrid On Ω
3	8h	3.602E-01 8.874E-02 2.134E-02	9.720E-01 3.538E-02 7.684E-03	5.507E-01 1.325E-01 9.404E-02
	4h	3.599E-01 8.122E-03 3.198E-04	3.644E+00 1.276E-01 6.083E-03	4.807E-01 1.743E-01 1.064E-01
	2h	1.127E-01 2.044E-02 5.259E-03	4.954E+00 1.126E+00 3.595E-01	1.354E-01 3.299E-02 9.587E-03
	h	2.329E-02 4.590E-03 2.076E-03	2.687E+00 1.272E+00 7.181E-01	4.209E-02 1.035E-02 4.465E-03
4	8h	4.434E-03 9.947E-04 2.325E-04	1.617E-01 4.012E-03 5.582E-04	8.480E-03 4.178E-03 3.872E-03
	4h	4.774E-03 1.055E-04 4.965E-06	6.814E-01 2.275E-02 1.167E-03	9.658E-03 4.446E-03 3.061E-03
	2h	2.096E-03 3.632E-04 9.150E-05	7.891E-01 1.697E-01 5.688E-02	3.475E-03 7.559E-04 2.512E-04
	h	6.079E-04 1.561E-04 6.461E-05	4.213E-01 2.089E-01 1.243E-01	1.248E-03 3.633E-04 1.590E-04

References

1. Brandt, A.; "Multilevel Adaptive Solutions to Boundary Value Problems", Math. Comp., 31 (1977), pp. 333-390

2. McCormick, S. F. and Ruge, J. W.; "Multigrid for Variational Problems", to appear.

3. Arakawa, H.; "Computational Design for Long-Term Numerical Integration of the Equations of Fluid Motion: Two-Dimensional Incompressible Flow. Part I.", J. Comp. Phys. 1 (1966), pp. 119-143.

BLACK BOX MULTIGRID

J. E. Dendy, Jr.

Los Alamos National Laboratory

Abstract

One major problem with the multigrid method has been that each new grid configuration has required a major programming effort to develop a code that specifically handles that grid configuration. Such a penalty is not required for methods like SOR, ICCG, etc.; in these methods, one need only specify the matrix problem, no matter what the grid configuration. In this paper we investigate such a situation for the multigrid method. The end result is a code, BOXMG, in which one need only specify the (logically rectangular, positive definite) matrix problem; BOXMG does everything else necessary to set up the auxilliary coarser problems to achieve a multigrid solution.

I. INTRODUCTION

In the multigrid method, one attempts to solve a discrete approximation

$$L^M U^M = F^M \tag{1}$$

to a continuous equation

$$LU = F \quad . \tag{2}$$

To do this one constructs a sequence of grids $G^1, \ldots, G^M$ with corresponding mesh sizes $h_1 > \ldots > h_M$. In its simplest mode of operations, one does a fixed number, IM, of relaxation sweeps (Gauss-Seidel, for example) on equation (1) and then drops down to grid G^{M-1} and the equation

$$L^{M-1}v^{M-1} = f^{M-1} \equiv I_M^{M-1}(F^M - L^M v^M), \tag{3}$$

where v^{M-1} is to be the coarse grid approximation to $V^M \equiv U^M - u^M$, where $v^M = u^M$ is the last iterate on grid G^M, and where I_M^{M-1} is an interpolation operator from G^M to G^{M-1}. To solve equation (3) approximately one resorts to recursion, taking ID relaxation sweeps on grid G^k before dropping down to grid G^{k-1}, $M-1 \geq k \geq 2$ and the equation

$$L^{k-1}v^{k-1} = f^{k-1} \equiv I_k^{k-1}(f^k - L^k v^k) \quad . \tag{4}$$

When grid G^1 is reached, the equation $L^1v^1 = f^1$ can be solved directly and $v^2 \leftarrow v^2 + I_1^2 v^1$ performed. Then one does IU relaxations sweeps on grid G^{k-1} before forming $v^k \leftarrow v^k + I_{k-1}^k v^{k-1}$, $3 \leq k \leq M$. (This description assumes $M \geq 3$, the cases $M = 1$ or 2 being trivial.)

One advantage of the multigrid method is that one obtains a fixed reduction of the error, significantly less than one, in the residual $F^M - L^M u^M$ per work performed per unknown on grid G^M. This is in sharp contrast to most iterative methods, for example, SOR, where the reduction increases as a function of the number of unknowns on grid G^M. Another advantage is that in many cases, multigrid achieves truncation error in work that is a small multiple of the number of unknowns. For further details, see references 1 and 2.

In most implementations of the multigrid method, the operators I_{k-1}^k have been grid dependent. In the simplest case, G^k and G^{k-1} are rectangular grids, the grid points of G^{k-1} are a subset of the grid points of G^k, the grid spacing h_{k-1} of G^{k-1} is twice the grid spacing h_k of G^k, and the interpolation I_{k-1}^k is bilinear. (See Reference 1.) If there are always to be G^k grid points on the boundary, then there is a constraint on the number of x[y] grid points NXM[NYM] on G^M that NXM = (NXO -1)2^{M-1} + 1 [NYM = (NYO - 1)2^{M-1} + 1], where NXO[NYO] is the number of x[y] grid points on G^0. Otherwise, interpolation near the boundary is a special case. The coding of interpolation is further complicated by whether the points on the boundary represent knowns (as in Dirichlet boundary conditions) or unknowns (as in Neumann boundary conditions).

Figure 1 shows two grids for a cell centered approximation to an elliptic equation. (The x's represent G^k and the ⊗'s G^{k-1}.) Now the above constraint on unknowns doesn't help since the nearest G^k point to the boundary is $h_k/2$ from the boundary and the nearest G^{k-1} point to the boundary is $h_{k-1}/4$ from the boundary, where h_{k-1} and h_k are G^1 and G^2 mesh spacings, respectively. The incorporation of a Neumann boundary condition, for example, on grid G^k leads to frequencies which are not damped out by bilinear interpolation, and convergence is degraded. Again something special (either in the interpolation routine or the relaxation routine) must be done at the boundary. This is easy in principle - especially if Brandt is nearby to advise - but is a pain in practice. There are two possible solutions in this case. One is to let $h_{k-1} = 3h_k$, which again leads to the coding of a special interpolation. The other is to use G^{k-1} unknowns that are not a subset of G^k unknowns, as in figure 2. This latter solution was the one employed in reference 3. Bilinear interpolation in this case involves special coding (for example, $a = \frac{1}{16}(9A + 3C + 3B + D)$ in figure 2), and there is again a constraint on the number of G^M unknowns to avoid special cases.

In addition to the grid structure, the actual difference equations cause programming difficulties. Consider, for example,

$$\begin{cases} -\nabla \cdot (D(x,y)\ \nabla U(x,y)) + \sigma(x,y)\ U(x,y) = f(x,y),\ (x,y) \in \Omega \\ \nu(x,y) \cdot D(x,y)\ \nabla\ U(x,y) + \gamma(x,y)\ U(x,y) = 0,\ (x,y) \in \partial\Omega, \end{cases} \tag{5a}$$

where $\Omega = (0,A) \times (0,B)$ with boundary $\partial\Omega$, ν is the outward normal to $\partial\Omega$, D is positive, σ and γ are non-negative, and D, σ, and f are allowed to be discontinuous across internal boundaries Γ of Ω; hence it is also assumed that

U and $u \cdot (D\nabla U)$ are continuous at (x,y) for almost every $(x,y) \in \Gamma$ (where $\mu(x,y)$ is a fixed normal vector at (x,y).) (5b)

If the finite difference approximation of equation (5a) is a vertex centered one as in ref. 2, then the "classic" multigrid method of Reference 1 (I_{k-1}^k = bilinear interpolation, I_k^{k-1} = a fixed nine point weighting operator, and the coefficients of L^{k-1} a fixed weighting of the coefficients of L^k) performs well as long as the discontinuities in D are not too severe and as long as Γ doesn't consist of too many line segments;

otherwise, it performs badly; indeed, it can even fail to converge in the fixed mode described above.

Reference 2 dealt with the situation in which D, σ, and f jump by orders of magnitude across Γ. It considered many possible choices of I^k_{k-1}, I^{k-1}_k, and L^{k-1}. Only one of these choices was found to be robust. This choice was $I^k_{k-1} = J^k_{k-1}$ (where J^k_{k-1} is defined below),

$$I^{k-1}_k = (I^k_{k-1})^* \tag{6}$$

and

$$L^{k-1} = (I^k_{k-1})^* L^k I^k_{k-1} \quad . \tag{7}$$

The choices, equations (6) and (7), are automatic in the finite element formulation of multigrid (ref. 4). References 4 and 5 both observed that $L^{k-1} = I^{k-1}_k L^k I^k_{k-1}$, with I^{k-1}_k not necessarily equal to $(I^k_{k-1})^*$, is a good choice in that the residual of the corrected solution vanishes when transferred to the coarse grid. This can be shown to be a good feature if L^k is symmetric. In the finite element formulation of multigrid, I^k_{k-1} is also automatic. Indeed multigrid finite element with piecewise bilinear elements was one of the methods considered in reference 2 and found not to be robust.

The crucial choice, then, given equations (6) and (7) is the choice of I^k_{k-1}. As discussed in reference 2, the first clue to the choice of I^k_{k-1} was that, because of equation (5b), I^k_{k-1} should approximately preserve the flux $\mu \cdot (D\nabla U)$ across Γ. In certain problems, however, when there were large jumps in both D and σ, it was discovered that on coarser grids where h^2 is large, the interfaces in σh^2 were as important as the interfaces in D. The obvious solution is to use the difference operator for the interpolation operator I^k_{k-1}. In one space dimension with three point difference operators, it is obvious how to do this. In two space dimensions, for the five point discrete Laplacian, it can also be done easily by the use of skewed five point discrete Laplacians; see Reference 6. This approach is doomed for equation (5) for two reasons. First, accurate skewed approximations are difficult if not impossible when interfaces are present. Second, even if L^M is a five point operator, the use of equation (7) generates nine point L^k's, $k < M$, making the above approach impossible. The solution arrived at in reference 2 is as follows: Suppose that at (IF, JF), L^k has the pointwise template

$$\begin{bmatrix} -T^k_{IF,JF+1} & -W^k_{IF,JF+1} & -R^k_{IF+1,JF+1} \\ -Q^k_{IF,JF} & S^k_{IF,JF} & -Q^k_{IF+1,JF} \\ -R^k_{IF,JF} & -W^k_{IF,JF} & -T^k_{IF+1,JF} \end{bmatrix} \tag{8}$$

Form $\tilde{Q}^k_{IF+1,JF} = T^k_{IF+1,JF+1} + Q^k_{IF+1,JF} + R^k_{IF+1,JF}$, $\tilde{S}^k_{IF+1,JF} = W_{IF+1,JF} + S^k_{IF+1,JF} + W^k_{IF+1,JF+1}$, and $\bar{Q}^k_{IF+2,JF} = T^k_{IF+2,JF} + Q^k_{IF+2,JF} + R^k_{IF+2,JF+1}$.

Then for horizontal lines embedded in the coarse grid,

$$v^k_{IF+1,JF} = \frac{\tilde{Q}^k_{IF+1,JF}\ u^{k-1}_{IC,JC} + \bar{Q}_{IF+2,JF}\ u^{k-1}_{IC+1,JC}}{\tilde{S}^k_{IF+1,JF}} \ . \tag{9}$$

(We have just summed equation (8) vertically to average out its y-dependence.) A similar formula can be used for vertical lines embedded in the coarse grid squares. Then, at fine grid points centered in coarse grid squares, $v^k_{IF+1,JF+1}$ may be obtained from the difference formula; i.e.,

$$\begin{aligned} v^k_{IF+1,JF+1} = (Q^k_{IF+1,JF+1}\ v^k_{IF+1,JF+1} &+ Q^k_{IF+1,JF+1}\ v^k_{IF+2,JF+1} \\ + W^k_{IF+1,JF+1}\ v^k_{IF+1,JF} &+ W^k_{IF+1,JF+2}\ v^k_{IF+1,JF+2} \\ + R^k_{IF+1,JF+1}\ v^k_{IF,JF} &+ R^k_{IF+2,JF+2}\ v^k_{IF+2,JF+2} \end{aligned} \tag{10}$$

$$+ T^k_{IF+1,JF+2}\, v^k_{IF,JF+2} + T^k_{IF+2,JF+1}\, v^k_{IF+2,JF})/S^k_{IF+1,JF+1} .$$

The vertical analogue of (9), (9) and (10) constitute the definition of J^k_{k-1} alluded to immediately preceding equation (6) above.

The near ultimate insult is a cell-centered difference approximation to equation (5) using $h_{k-1} = 3h_k$ or the grid structure of fig. 2. The definition of I^k_{k-1} which approximately preserves flux across Γ in this case is not obvious, and the computation of $(I^k_{k-1})^* L^k I^k_{k-1}$ is a disaster. Desperation being the mother of invention, one soon decides there has to be a better way.

II. BLACK BOX MULTIGRID

The better way has already been described; one only needs to interpret it differently. The crucial observation is that once one has a pointwise template like equation (8) for L^M, then the definition of J^k_{k-1} and $(J^k_{k-1})^* L^k J^k_{k-1}$ is independent of where this template came from. (We refer to this method as black box multigrid not because - as some would have it - multigrid is black magic but because the code which implements the method acts as a black box for the user; he need only specify the difference equations on the finest grid since the code, BOXMG, generates the auxilliary coarse problems.

The same artifice allows one to get rid of the restriction on the number of unknowns on the finest grid. For the situation depicted in figure 3, for example, one can imagine ficticious coarse grid points. The boundary conditions on the fine grid are incorporated into the operator, as in reference 2, so that for points (IF,JF) on the right boundary, for example, $R^k_{IF+1,JF+1} = Q^k_{IF+1,JF} = T^k_{IF+1,JF} = 0$ in equation (8). The boundary of the coarse grid doesn't coincide with the boundary of the fine grid, but the boundary conditions will be picked up by the formation of $(J^k_{k-1})^* L^k J^k_{k-1}$.

An example of an extreme case of this artifice is the situation in which one wants to solve a Dirichlet problem on a given irregular region. One proceeds by embedding the region in a rectangle, writing down difference equations at points interior to the region. These difference equations incorporate the Dirichlet data on the boundary of the region in such a way that there is no coupling between the interior points and the other points. At the other points one writes down an equation $\sigma_{i,j} U_{i,j} = F_{i,j}$, where $\sigma_{i,j} \neq 0$ and $F_{i,j}$ are arbitrary. This artifice makes the problem logically rectangular. The solution to the difference equation is obtained at the interior points, and the solution $U_{i,j} = F_{i,j}/\sigma_{i,j}$ is obtained at the other points. On a serial machine, this process for solving irregular region problems may be inefficient for some regions, since the number of other points can be quite large. On a vector machine, however, the situation isn't clear, since the embedding technique is immediately vectorizable and since other techniques may vectorize with difficulty.

One disadvantage to the black box method is storage. In the situation that the coefficients of the difference equations are easy to compute (for example, Laplace's equation on a rectangle), there is a storage penalty of at least five [seven] locations per fine grid point for the black box method for a five [nine] point operator; this assumes that the right hand side is stored and that $1/S^M_{IF,JF}$ is computed and stored. If one is not going to restrict the number of unknowns on the finest grid, however, then not storing the coefficients means additional programming and checking for special cases. (If the checking involves an IF test in the inner loop of a double DO loop, the degradation in run time can be dramatic on a machine like a CDC 7600.) Moreover, we are more interested in problems like equation (5), where the coefficients of the difference equations are not easy to compute and have to be stored anyway.

If we assume that the finest grid coefficients are stored, then there is still a storage penalty for the black box method. First, even in the case that the operator on the finest grid is a fine point operator, nine point operators are generated on the coarser grids. If it is assumed that the given problem can be worked with five point operators on the coarser grids (an assumption which is not at all clear for equation (5)), then an extra two storage locations per coarse grid point are required, for a total of $2(1/4 + 1/16 + \dots) = 2/3$ locations per fine grid point. Second, the interpolation coefficients have to be stored, requiring four locations for equation (9) and its vertical analogue. (Unfortunately, since the coefficients in equation (9) don't necessarily sum to 1, both $\tilde{Q}^k_{IF+1,JF}/\tilde{S}_{IF+1,JF}$ and $\bar{Q}_{IF+2,JF}/\tilde{S}_{IF+1,JF}$ must be stored.) Equation (10) requires no additional storage but does require nine multiplies. By using equation (9) and its vertical analogue, equation (10) can be rewritten in terms of coefficients of $V^k_{IF,JF}$, $V^k_{IF+2,JF+2}$, $V^k_{IF,JF+2}$, and $V^k_{IF+2,JF}$; this reduces the computation of equation (10) to four multiplies but requires four storage locations per coarse grid point. Hence interpolation as currently implemented in BOXMG requires a total of 8(1/4 + 1/16 + ...) = 8/3 locations per fine grid point.

One can also ask the question of whether there is any disadvantage in execution time with the black box method. The worse case is the case in which the operator on the finest grid is a five point operator; to be fair, let us assume that it is not the five point Laplacian, in order that advantage cannot be taken of the very simple form of the coefficients in the five point Laplacian. To be unfair to the black box method let us assume that I^{k-1}_k is injection in the "classic" multigrid method. Experimentally, for easy equations, BOXMG achieves the reduction of the error by a factor of 0.1 [0.05] per multigrid cycle for IU = ID = 1 and IM = 2[ID = 2, IU = 1, IM = 3]. This is in contrast to figures of 0.25 and 0.125 for "classic" multigrid. If the total work for "classic" multigrid and black box multigrid is computed, including the work for I^k_{k-1} and I^{k-1}_k and if the comparison is expressed in terms of the convergence factor (convergence factor = reduction of error/work unit, where 1 work unit = 8 floating point operations, the amount of work for one Gauss-Seidel sweep on the finest grid), then the comparison is as follows:

	convergence factor, "classic"	convergence factor, black box	convergence factor, "classic" with residual weighting
IU = ID = 1, IM = 2	0.66	0.66	0.75
IU = 2, ID =1, IM = 3	0.64	0.66	0.74

Thus there is no penalty in convergence factor for the black box method. There is a penalty, however, for the black box method in that the computation of I^k_{k-1} and L^{k-1} is not without cost; this is startup calculation time that doesn't have to be performed in the "classic" multigrid method. As soon as one considers the convergence factor for "classic" multigrid if nine point residual weighting is necessary (as it will be for all but the simplest problems), then the degradation of convergence factor makes it obvious that the black box method can pay for its overhead. Moreover, the black box method will work problems that "classic" multigrid can't handle. If, however, one is solving just to trucation error, then the "classic" multigrid method is probably more efficient for problems with smooth coefficients. The extra expense of cubic interpolation the first time a grid is visited in the classic multigrid method is probably more than offset by the expense of computing L^k, $k < M$, in the black box method.

A relatively unimportant issue of implementation is whether $I_k^{k-1} = (J_{k-1}^k)^*$ is necessary. In reference 2 a heuristic argument was made for this choice, but experiments seemed to indicate that the use of a fixed nine point weighting for I_k^{k-1} did not lead to any significant degradation of convergence factor, as long as $I_{k-1}^k = J_{k-1}^k$ and equation (7) is used to define L^{k-1}. (In some problems, the fixed weighting even gave slightly better convergence.) A nine point fixed weighting for $I_k^{k-1} = (J_{k-1}^k)^*$ is automatically correct at the boundary. Hence, since J_{k-1}^k is stored, it is easier to use $(J_{k-1}^k)^*$ for I_k^{k-1}.

The multigrid algorithm described in section 1 begins on the finest grid G^M. In the full multigrid algorithm described by Brandt (ref. 1), one begins on the coarsest grid G^1 instead and uses the coarser grids to generate a good initial guess. For three grids, for example, the pattern of grid transfer is $G^1 \rightarrow G^2 \rightarrow G^1 \rightarrow G^2 \rightarrow G^3 \rightarrow G^2 \rightarrow G^1 \rightarrow G^2 \rightarrow G^3$. In Brandt's scheme, when a grid is visited for the first time, cubic interpolation is used instead of bilinear interpolation, and when the finest grid (G^3 in the example above) is visited for the second time, one has the solution to truncation error. Indeed for equations with smooth coefficients, not only are the pointwise values h^2 accurate but the centered difference quotients approximate the first and second derivatives to h^2 accuracy.

In BOXMG we have not implemented cubic interpolation nor the generalization of it ((5.12) of reference 2) for equations with discontinuous coefficients; the reason is that numerical experiments indicated no advantage for either versus J^k_{k-1} in equations with discontinuous coefficients. This issue is discussed further in Section IV.

One final issue, discovered by Brandt, is the issue of the use of the right hand side in interpolation. Generally, the use of the right hand side provides an $O(h^2)$ correction to interpolation and is not worthwhile. In black box multigrid, however, the right hand side at the boundary can contain boundary data, and in such cases, not using the right-hand side can lead to O(1) interpolation errors at the boundary, and consequently destroy all hope of solving to truncation error in one or two cycles. Thus to the right hand side of equation (9) we add $r^k_{IF+1,JF}/S^k_{IF+1,JF}$, and to the hand side of equation (10) we add $r^k_{IF+1,JF+1}/S^k_{IF+1,JF+1}$, where r^k is the residual; when a grid is visited for the first time $r^k = F^k$ (if a zero initial guess is used).

III. THE PARAMETERS OF BOXMG

In this section we discuss the parameters the user must specify to use BOXMG. These are actually discussed in the comments of BOXMG following the reading of the parameters, but we provide a little more detail here. We hope this description and the examples of Section IV will make the usage of BOXMG clear. We had originally intended to rewrite BOXMG in perfect, portable Fortran. Ignoring for the moment whether such a beast exists, we discovered that we were phychologically incapable of the quest. Nevertheless, we still hope that BOXMG will prove useful and that its coding is clear enough to be changed by others for their devious ends.

The grid in BOXMG is always logically rectangular. The parameters NXM and NYM specify the number of unknowns in the x and y coordinates respectively. HXM and HYM specify the x and y spacing respectively on the finest grid; these parameters are only used in computing the discrete L^2 norm of the residual, since the user specifies the equations on the finest grid. Indeed, since the equations on the finest grid can be written on a Lagrangian grid, HXM and HYM may have little meaning in some cases.

TOL is the tolerance. In one mode of BOXMG, iteration is continued until the discrete L^2 error of the residual is less than TOL or until the accumulated number of multigrid cycles NCYC is equal to |ISTRT|.

IFD is an indicator for the scheme on the finest grid. If IFD = 1, a five point scheme is assumed; otherwise, a nine point scheme is assumed. IU, ID, and IM have already been discussed in Section I. The recommended choices are IU = ID = 1 and IM = 2 or IU = 1, ID = 2, and IM = 3. For problems with smooth coefficients the latter choice is slightly better; for problems with rough coefficients the first choice is better. If ISTRT < 0, BOXMG will begin iterating on the finest grid. If ISTRT > 0, BOXMG will begin on the coarsest grid and will bootstrap itself up to the finest grid, as discussed in Section II, and then continue cycling.

IRELAX is an indicator for the type of relaxation. IRELAX = 1 means point relaxation. IRELAX = 2 means line relaxation by lines in x. IRELAX = 3 means line relaxation by lines in y. IRELAX = 4 means line relaxation by lines in x followed by line relaxation by lines in y. These options are included for flexibility. For equations like $\varepsilon\, u_{xx} + u_{yy} = f$, $\varepsilon \ll 1$, (or for $\Delta u = f$, where $\Delta x \gg \Delta y$ on the finest grid) line relaxation by lines in y is needed for a good smoothing rate (ref. 2). For $u_{xx} + \varepsilon\, u_{yy} = f$, line relaxation by lines in y is needed for a good smoothing rate. In some cases, both are needed.

ITAU is an indicator for computing and printing an estimate of the truncation error. If ITAU = 0, then

$$J_M^{M-1} F^M - L_M^{M-1} \tilde{I}_M^{M-1} U^M \quad , \tag{11}$$

where $\tilde{I}_M^{M-1}$ is injection, is computed and printed. If ITAU $\neq$ 0, then eq. (11) is not computed and printed. A discussion of this feature is given in Section IV.

ICOEF determines when $(J_{k-1}^k)^* L^k J_{k-1}^k$ will be computed. If ICOEF = 0, then when M, the number of grids is computed, ICOEF will be set equal to M, and $(J_{k-1}^k)^* L^k J_{k-1}^k$ will be computed for $k \leq M$. If ICOEF = 1, then L^k must be specified for every grid, G^k, $1 \leq k \leq M$ since $(J_{k-1}^k)^* L^k J_{k-1}^k$ will not be computed for any grid. This feature allows the user to run something like "classic" multigrid except that J_{k-1}^{k} will still be computed by the code and may or may not be bilinear interpolation; hence, the use of this option (ICOEF = 1) may lead to divergence if discontinuities of orders of magnitude exist in the coefficients.

IVW, MCYCL, ALPHL, and ALPHM are cycling parameters. IVW determines the type of cycle to be performed. If IVW = 1, the usual V-cycle will be performed. If IVW = 2, W-cycles will be performed. Larger values of IVW give more exotic patterns. MCYCL is for the coarser grids what ISTRT is for the finest grid; in each cycle grid G^k, $k < M$ will be visited MCYCL times before grid G^{k+1} is visited unless the discrete L^2 norm of $J_k^{k-1} f^k - L^{k-1} \tilde{I}_k^{k-1} U^k$ is less than ALPHL (ALPHM if k = M) times the discrete L^2 norm of the residual on G^k. The usual value of MCYCL is 1. The theoretical value of both ALPHL and ALPHM to achieve truncation error is 0.125. If, however, one is solving in the mode where the discrete L^2 norm of the residual is to be reduced to less than TOL, then ALPHM = 0 should be used. The flexibility provided by these four parameters is awesome.

Aside from specifying these parameters, the user must provide the subroutine PUTF, which specifies the difference equations on the finest grid. (As remarked above, certain values of ICOEF would require PUTF to make sense for coarser grids as well.) An example of a PUTF is given in the listing of BOXMG in reference 7. PUTF has one argument K and a call to KEY in it, CALL KEY(K,JST,II,JJ,HX,HY), which fetches the storage for the arrays. For IFD = 1, the user must specify the arrays FR, FA, SO, SOR, and QF. For IFD ≠ 1, he must specify FSW and FNW as well. The logical grid is assumed to be (I,J); I =1, ..., II; J = 1, ..., JJ. The sets {(1,J): J = 1, ..., JJ}, {(II,J): J = 1, ..., JJ}, {(I,1): I = 1, ..., II}, and {(I,JJ): I = 1, ..., II} are fictitious points, assumed for ease of programming. For IFD = 1, the template

$$\begin{bmatrix} & -\,\mathrm{FA(JP+I)} & \\ -\,\mathrm{FR(JO+I)} & \mathrm{SO(JO+I)} & -\,\mathrm{FR(JO+I+1)} \\ & -\,\mathrm{FA(JO+I)} & \end{bmatrix} ,$$

is assumed, where JO = JST(J) and JP = JST(J+1). For IFD $\neq$ 1, the template

$$\begin{bmatrix} -\ \mathrm{FNW(JP+I)} & -\ \mathrm{FA(JP+I)} & -\ \mathrm{FNW(JP+I+1)} \\ -\ \mathrm{FR(JO+I)} & \mathrm{SO(JO+I)} & -\ \mathrm{FR(JO+I+1)} \\ -\ \mathrm{FSW(JO+I)} & -\ \mathrm{FA(JO+I)} & -\ \mathrm{FNW(JO+I+1)} \end{bmatrix}$$

is assumed. In both cases, QF(JO+I) should be the right hand side, I = 2, ..., I1 $\equiv$ II-1; J = 2, ..., J1 $\equiv$ JJ-1 in PUTF. The boundary conditions should be incorporated into the operator, so that all coefficients referring to fictitious points should be zero. For example, FR(JO+2), FSW(JO+2), and FNW(JP+2) should all be zero, J = 1, ..., JJ.

BOXMG automatically determines the number of grids M from the input parameters NXM and NYM. It does this by bisecting the given logical grid until it arrives at a grid which cannot be practically bisected any further, i.e., when the number of x or y unknowns is three or four. (For NXM >> NYM or NXM << NYM, this may lead to the situation of its being profitable to bisect the coarsest grid only in the x or y direction, but this feature is not provided in BOXMG). Once the number of levels is determined, BOXMG computes how much storage must be allowed for the various arrays. If insufficient storage has been declared, a message is printed and the code terminates.

The storage parameters in BOXMG are:

```
  NOG = maximum number of grids
      = maximum storage for NX and NY in common block DC1
      = maximum storage for NST, IMX, JMX, HX, HY and IND in common block
        GRD
NFMAX = maximum storage for arrays Q, QF, FR, FA, SO, SOR, TOT
NCMAX = maximum storage for arrays CIA, CIR, CISW, CISE, CINW, CINE, CIL,
        CIB
NABD1 = maximum first subscript of ABD, where ABD is the array used for
        direct solution on the coarsest grid
NABD2 = maximum second subjscript of ABD
```

If IFD = 1, the storage required for FSW and FNW is NCMAX; otherwise NFMAX is required. If IRELAX = 1 or 2, SOS can be dimensioned to 1; otherwise, SOS should be dimensioned to NFMAX.

IV. EXAMPLES

The first example is for eq. (5) for $\Omega = (0,24) \times (0,24)$. The boundary conditions are

$$\frac{\partial u}{\partial \nu} = \begin{cases} -u/2D \text{ on } y = 24 \text{ or } x = 24 \\ 0, \text{ otherwise,} \end{cases}$$

and D is given by

$$D(x,y) = \begin{cases} 1, \text{ if } (x,y) \in [0,12) \times [0,12) \cup (12,20] \times (12,20] \\ 1000, \text{ otherwise} \end{cases} .$$

We take $\sigma = 1/3D$, and $f = 0$ when $D = 1$ and $f = 1$ when $D = 1000$. The results are summarized in Table 4.1. In this table 1.64, -1, for example, is used for 1.64×10^{-1}. Also $\tau_{i,j}^{M-1}$ is the quantity in (11); the number r is the exponent in the asymptotic expansion of the error in the τ's; it is computed by the formula $\log\left(\frac{\text{error}(2h)}{\text{error}(h)}\right)/\log 2$. The first three rows show the results of running one cycle starting on the coarsest grid; the next four rows continue from there until the discrete L^2 norm of the error is less than 10^{-6}. In the last row, HXM and HYM are still .5, so that the region is really $(0.,23) \times (0.,23.)$; this example illustrates the picture of Fig. 3 for the transition from G^k to G^{k-1}, $k = 5, 4, 3, 2$. Since $(0.,23.) \times (0.,23.)$ is a small perturbation of $(0.,24.) \times (0.,24.)$, one would expect comparable results for the two cases. An example of parameters is for the last row, where NXM = NYM = 48, HXM = HYM = .5, TOL = whatever, IFD = 1, IU = 1, ID = 1, ISTRT = 20, IRELAX =1, ITAU = 0, ICOEF = 0, IVW = 1, MCYCL = 1, ALPHL = 0.125, ALPHM = 0.125.

The second example is the same as the first example except that cell-centered differencing is employed. For the runs made, the interface comes midway between cell centers. In one dimension, if an interface is located at ih and $D(x) = D_+$ if $x > ih$ and $D(x) = D_-$ if $x < ih$, then the difference equation at $(i-\frac{1}{2})h$ is

$$-D_- u_{-i-3/2} + \left(D_- + \frac{1}{1/2(D_+ + D_-)}\right) u_{i-1/2} - \frac{1}{1/2(D_+ + D_-)} u_{i+1/2} \quad ;$$

Table 4.1

Number of Unknowns	CPU Time in Seconds on CDC 7600	$\max_{i,j}\lvert\tau_{i,j}^{M-1}\rvert$ and Estimate of r	Reduction in Discrete L^2 Norm of Residual in last cycle and number of cycles
13 × 13 ALPHM = .125	.017	2.06	.09; NCYC = 3
25 × 25 ALPHM = .125	.052	6.00, -1; r = 1.61	.09; NCYC = 3
49 × 49 ALPHM = .125	.170	1.84, -1; r = 1.71	.07; NCYC = 3
13 × 13 TOL = 10^{-6}	.036	2.06	.13; NCYC = 8
25 × 25 TOL = 10^{-6}	.126	6.00, -1; r = 1.61	.34; NCYC = 10
49 × 49 TOL = 10^{-6}	.430	1.84, -1; r = 1.71	.46; NCYC = 14
49 × 49 TOL = 10^{-6}, IVW = 2	.554	1.84, -1	.23; NCYC = 9
48 × 48 ALPHM = .125	.156	2.74, -1	.07; NCYC = 3

a similar formula holds in two dimensions. The results are summarized in Table 4.2. An example of parameters for this problem is for the last row, where NXM = NYM = 48, HXM = HYM = .5, TOL = 10^{-6}, IFD = 1, IU = 1, ID = 1, IM=2, ISTRT = 50, IRELAX = 1, ITAU = 0, ICOEF = 0, IVW = 1, MCYCL = 1, ALPHL = 0.125, ALPHM = 0. $\operatorname{Max}_{i,j}\lvert\tau_{i,j}^{M-1}\rvert$ is assumed next to (24., 12.), near the interface and right boundary. Away from the interfaces the $\tau_{i,j}^{M-1}$'s are well behaved. Let us examine the answers at (24., 12.) and compare them with those obtained from the vertex centered scheme. By using the approximation to the boundary condition for a horizontal averaging and conservation of flux for vertical averaging, we can get approximations to the solution at (24., 12.) for the cell-centered scheme; call them

Table 4.2

Number of Unknowns	CPU Time in Seconds on CDC 7600	$\max_{i,j} \lvert \tau_{i,j}^{M-1} \rvert$ and Estimate of r	Reduction in Discrete L^2 Norm of Residual in last cycle and number of cycles
12 × 12 ALPHM = .125	.014	4.82	.13; NCYC = 3
24 × 24 ALPHM = .125	.045	3.73; r = .37	.10, NCYC = 3
48 × 48 ALPHM = .125	.116	2.30; r = .70	.08; NCYC = 2
12 × 12 TOL = 10^{-6}	.038	4.87	.14; NCYC = 10
24 × 24 TOL = 10^{-6}	.103	3.75; r = .38	.12; NCYC = 9
48 × 48 TOL = 10^{-6}	.354	2.26; r = .73	.10; NCYC = 9

$\bar{u}^3_{cc}$, $\bar{u}^4_{cc}$, $\bar{u}^5_{cc}$ (for finest grid 12 × 12, 24 × 24, and 48 × 48 respectively and tolerance 10^{-6}). Let u^3_{vc}, u^4_{vc}, u^5_{vc} be the answers from the vertex centered scheme at (24., 12.) (for finest grid 13 × 13, 25 × 25, and 49 × 49 respectively and tolerance 10^{-6}). Compute $\bar{u}^{RE}_{cc} = 4/3\bar{u}^5_{cc} - 1/3\bar{u}^4_{cc}$ and $u^{RE}_{vc} = 4/3u^5_{vc} - 1/3u^4_{vc}$. (RE stands for Richardson extrapolation.) Then $\bar{u}^3_{cc} - \bar{u}^{RE}_{cc} = .801$, $\bar{u}^4_{cc} - u^{RE}_{cc} = .226$, and $\bar{u}^5_{cc} - u^{RE}_{cc} = .056$; and $u^3_{vc} - u^{RE}_{vc} = .766$, $u^4_{vc} - u^{RE}_{vc} = .224$, and $u^5_{vc} - u^{RE}_{vc} = .056$. Thus the assumption of asymptotic error of Ch^2 at (24., 12.) for both schemes is justified, and -- at least for this example -- there is no reason from considerations of accuracy to prefer the vertex centered scheme to the cell centered scheme. (We have also checked points away from the interfaces and boundaries, and the same conclusion -- less interesting in these cases -- is valid.)

The third example is

$$\begin{cases} -\Delta U = F \text{ on } \Omega = (0,1) \times (0,1) \\ U = 0 \text{ on } \partial\Omega, \end{cases}$$

where F is chosen so that the solution is $U(x,y) = 3e^{x}e^{-y}xy(1-x)(1-y)$. The only way one can handle such a Dirichlet problem with BOXMG is to incorporate the boundary data into the right hand side of the finest grid. Thus the difference operator along the x = 0 boundary away from the corners is

$$\left[\frac{1}{h^2} \begin{matrix} & -1 & \\ 0 & 4 & -1 \\ & -1 & \end{matrix}\right] .$$

I.e., the boundary is not treated as part of the grid at all. To have the boundary treated as part of the grid in this case would have required a lot of special cases in BOXMG; hence, we decided not to implement this option.

The results are summarized in Table 4.3. In this table,

$$D^0U_{i,j} = \frac{u_{i+i,j} - u_{i-i,j}}{2h} .$$

In this example, two cycles appears to be sufficient to solve nearly to truncation error in both the function values and their derivatives even though cubic interpolation is not employed. An example of parameters for this problem is for the fourth row, when NXM = NYM = 9, HXM = HYM =.1, TOL = 10^{-6}, IFD = 1, IU = 1, ID = 1, ISTRT = 50, ITAU = 0, ICOEF = 0, IVW = 1, MCYCL = 1, ALPHL = 0.125, ALPHM = 0.

The fourth example is

$$\begin{cases} -\Delta U = F \text{ on } \Omega = \text{circle of diameter 1. centered at } (0., 0.) \\ U(x,y) = g(x,y) = 3e^{x}e^{-y}xy(1-x)(1-y), \text{ if } (x,y) \in \partial\Omega, \end{cases} \qquad (12)$$

where F is chosen so that the solution is $U(x,y) = 3e^{x}e^{-y}xy(1-x)(1-y)$. This example illustrates the technique of embedding. We embed Ω in $\Omega' = (-.5, .5) \times (-.5, .5)$. At points in $\Omega'\backslash\bar{\Omega}$ we write down the equation

Table 4.3

Number of Unknowns	CPU Time in Seconds on CDC 7600	$\max_{i,j}\lvert u_{i,j} - U_{i,j}\rvert$ and Experimental p	$\max_{i,j}\lvert D^0 u_{i,j} - \frac{\partial U}{\partial x}\rvert_{i,j}\rvert$ and Experimental q	$\max_{i,j}\lvert \tau^{M-1}_{i,j}\rvert$ and Experimental r	Reduction in Discrete L^2 Norm of Residual in Last Cycle and Number of Cycles
9 × 9 ALPHM = .125	.011	1.01, -3	1.13, -2	6.26, -3	.05; NCYC = 3
19 × 19 ALPHM = .125	.032	2.53, -4; p = 2.00	3.15, -3; q = 1.84	8.17, -4; r = 2.93	.03; NCYC = 3
39 × 39 ALPHM = .125	.109	6.37, -5; p = 1.99	8.37, -4; q = 1.91	7.28, -5; r = 3.48	.02; NCYC = 3
9 × 9 TOL = 10^{-6}	.018	1.02, -3	1.12, -2	6.27, -3	.09; NCYC = 5
19 × 19 TOL = 10^{-6}	.052	2.56, -4; p = 1.99	3.15, -3; q = 1.83	8.16, -4; r = 2.99	.11; NCYC = 6
39 × 39 TOL = 10^{-6}	.174	6.42, -5; p = 2.00	8.37, -4; q = 1.91	7.24, -5; r = 3.49	.11; NCYC = 6
39 × 39 ISTRT = 1	.082	7.48, -5; p = 1.76	9.82, -4; q = 1.76	6.99, -5; r = 3.54	.01; NCYC = 2

$u = 0.$; at points in Ω whose north, south, east, and west neighbors are in $\bar{\Omega}$, we use the usual five point Laplacian. For simplicity we use the simplest treatment of points that don't fall into either of the above sets. Consider, for example, a point $U^M_{i,j}$ in Ω whose neighbor $U^M_{i+i,j}$ is not in $\bar{\Omega}$, and let the distance from $U^M_{i,j}$ to $\partial\Omega$ be θh. Approximating $-h^2U_{xx}$ by $(-U^M_{i-i,j} + 2U^M_{i,j} - U^M_{i+i,j})$ and using the relation $g((i+\theta)h,jh) \cong (1-\theta)U^M_{i,j} + \theta U^M_{i+i,j}$ to solve for $U^M_{i+i,j}$ gives the following difference equation at (ih,jh):

$$
\begin{aligned}
-U^M_{i-i,j} + (3 + \tfrac{1}{\theta})U^M_{i,j} - U^M_{i,j-1} - U^M_{i,j+1} \\
= h^2F(ih,jh) + \tfrac{1}{\theta}g((i+\sigma)h,jh);
\end{aligned}
\tag{13}
$$

note that there is no coupling between (ih,jh) and $((i+1)h,jh)$.

The results are summarized in Table 4.4.

The fifth example is the same as the fourth example except that in this case we use mapping to solve it. That is, we map the boundary of Ω onto the boundary of $\Omega'' = (0,1) \times (0,1)$ giving x and y as a function of ξ and η on $\partial\Omega''$ and solving approximately the problem

$$
\begin{cases} \Delta x = 0, \ (\xi,\eta) \ \varepsilon \ \Omega'' \\ \ x = x(\xi,\eta), \ (\xi,\eta) \ \varepsilon \ \partial\Omega'' \end{cases}
\qquad
\begin{cases} \Delta y = 0, \ (\xi,\eta) \ \varepsilon \ \Omega^H \\ \ y = y(\xi,\eta), \ (\xi,\eta) \ \varepsilon \ \partial\Omega'' \end{cases} .
\tag{14}
$$

We do this by discretizing Ω'', approximating eq. (14) by five point Laplacians and specifying

$$x(ih,1) = \frac{1}{2}\cos(\frac{3\pi}{4} - ih\frac{\pi}{2}) \quad , \quad y(ih,1) = \frac{1}{2}\sin(\frac{3\pi}{4} - ih\frac{\pi}{2})$$

$$x(ih,o) = \frac{1}{2}\cos(\frac{5\pi}{4} + ih\frac{\pi}{2}) \quad , \quad y(ih,o) = \frac{1}{2}\sin(\frac{5\pi}{4} + ih\frac{\pi}{2})$$

$$x(o,ih) = \frac{1}{2}\cos(\frac{5\pi}{4} - ih\frac{\pi}{2}) \quad , \quad y(o,ih) = \frac{1}{2}\sin(\frac{5\pi}{4} - ih\frac{\pi}{2})$$

Table 4.4

Number of Unknowns	CPU Time in Seconds on CDC 7600	$\max_{i,j}\lvert u^{0}_{i,j} - U_{i,j}\rvert$ and Experimental p	$\max_{i,j}\lvert D_0 U^{M-1}_{i,j} - \frac{\partial U}{\partial x}\rvert_{i,j}\rvert$ and Experimental q	$\max_{i,j}\lvert \tau_{i,j}\rvert$ and Experimental r	Discrete L^2 Norm of Error and Estimate of P	Reduction in Discrete L^2 Norm of Residual in Last Cycle and number of cycles
9 × 9 ALPHM = .125	.011	3.21, -3	2.53, -2	3.15, -2	5.37, -4	.04; NCYC = 3
19 × 19 ALPHM = .125	.032	9.31, -4; p = 1.78	8.11, -3; q = 1.64	1.71, -1; r = 2.44	1.72, -4; p = 1.64	.07, NCYC = 3
39 × 39 ALPHM = 1.25	.109	7.24, -4; p = .36	3.77, -3; q = 1.11	1.62, -1; r = .08	1.91, -4; p = -.15	.10, NCYC = 3
9 × 9 TOL = 10^{-6}	.018	3.21, -3	2.53, -2	3.15, -2	5.37, -4	.04, NCYC = 6
19 × 19 TOL = 10^{-6}	.073	9.14, -4; p = 1.81	7.94, -3; q = 1.67	1.71, -1; r = -2.44	1.66, -4; p = 1.69	.10, NCYC = 9
39 × 39 TOL = 10^{-6}	.269	2.50, -4; p = 1.87	2.77, -3; q = 1.51	1.61, -1; r = .09	4.82, -5; p = 1.78	.13, NCYC = 10
39 × 39 ALPHM = .05	.134	2.77, -4; p = 1.74	2.80, -3; q = 1.53	1.61, -1; r = .09	6.04, -5; p = 1.51	.11, NCYC = 4

$$x(1,ih) = \frac{1}{2}\cos(\frac{-\pi}{4} + ih\frac{\pi}{2}) \quad , \quad y(1,ih) = \frac{1}{2}\sin(\frac{-\pi}{4} + ih\frac{\pi}{2}) \quad .$$

Ideally, eq. (14) should be solved by multigrid, but for simplicity in this example we used SOR. The equation (13) transforms to the following in the ξ-η coordinate system:

$$\begin{cases} (G_1 y_\eta - G_2 x_\eta)_\xi + (G_1 x_\xi - G_2 y_\xi)_\eta = FJ, \ (\xi,\eta) \in \Omega'' \\ u(\xi,\eta) = g(\xi,\eta), \ (\xi,\eta) \in \partial\Omega'' \quad , \end{cases} \tag{15}$$

where $G_1 = \frac{D}{J}(u_\xi y_\eta - u_\eta y_\xi)$, $G_2 = \frac{D}{J}(u_\eta x_\xi - u_\xi x_\eta)$, and $J = x_\xi y_\eta - x_\eta y_\xi$.

Equation (15) is differenced in cell-centered form. The results are summarized in Table 4.5.

Since J is singular at the corners of Ω'', it is not surprising that the error in the approximation to the x-derivative (the finite difference version of $\frac{1}{J}(y_\eta u_\xi - y_\xi u_\eta)$) grows larger as the mesh is refined. For the fixed point (.1,.1) -- the interior point nearest (0,0) on the 10×10 ξ-η grid -- this error decreases; nevertheless the maximum error in the approximation to the x-derivative grows and is always assumed at a point nearest one of the corners of Ω''.

The sixth example uses the mesh in Fig. 4, which was the mesh used in a Rayleigh-Taylor calculation in Ref. 3. We include it since it is rather distorted (in fact, as commented in Ref. 3, a "bowtie" forms on the next time step) and represents a challenge to the black box approach. We use the same differencing as employed for eq. (15). For this example, it is not clear what continuous system is being approximated. If however, we use Dirichlet data identically equal to 1. and $F \equiv 0.$, then the solution to the difference equations is identically 1. The results are summarized in Table 4.6.

Table 4.5

No. of Unknowns	CPU Time in Seconds on CDC 7600	$\max_{i,j}\lvert u_{i,j} - U_{i,j}\rvert$ and Experimental p	Maximum Derivative Error and Estimate of q	Discrete L^2 Norm of Error and Estimate of p	Discrete L^2 Norm of x-derivative Error and Estimate of q	x-derivative Error at (.1, .1) and Estimate of q	x-derivative Error at (.5,.5) and Estimate of q	$\max_{i,j}\lvert\tau^{M-1}_{i,j}\rvert$ and Estimate of r	Reduction in Discrete L^2 Norm of Residual in Last Cycle and Number of Cycles
10 × 10 ALPHM = .125	.012	1.66, -2	3.35, -1	4.45, -3	6.63, -2	3.34, -1	1.97, -2	2.52, -1	.04, NCYC = 3
20 × 20 ALPHM = .125	.037	7.35, -3; p = 1.17	3.95, -1; q = -.24	1.76, -3; p = 1.33	3.85, -2; q = .26	1.86, -1; q = .04	5.48, -3; q = 1.85	4.67, -1; r = -.03	.04, NCYC = 3
40 × 40 ALPHM = .125	.140	4.89, -3; p = .58	4.19, -1; q = -.09	8.89, -4; p = .99	2.75, -2; q = .48	9.22, -2; q = 1.01	1.53, -3; q = 1.84	8.03, -1; r = -.85	.05, NCYC = 3
10 × 10 TOL = 10^{-6}	.025	1.66, -2	3.35, -1	4.45, -3	6.63, -2	3.35, -1	1.93, -2	2.52, -1	.07, NCYC = 6
20 × 20 TOL = 10^{-6}	.078	7.40, -3; p = 1.17	3.96, -1; q = -.24	1.77, -3; p = 1.33	3.85, -2; q = .26	1.86, -1; q = .04	5.58, -3; q = 1.80	4.46, -1; r = -.82	.30, NCYC = 8
40 × 40 TOL = 10^{-6}	.327	4.86, -3; p = .61	4.22, -1; q = -.09	8.95, -4; p = .98	2.75, -2; q = .49	9.33, -2; q = 1.00	1.56, -3; q = 1.84	8.03, -1; r = -.85	.13, NCYC = 10

Table 4.6

Number of Unknowns	CPU Time in in Seconds on CDC 7600	$\max_{i,j}\lvert U_{i,j} - U_{i,j}\rvert$	Maximum Error in x-derivative	L^2 Norm. of Error	L^2 Norm. of Error in x-derivative	$\max_{i,j}\lvert\tau^{M-1}_{i,j}\rvert$	Reduction in Discrete L^2 Norm. of Residual in Last Cycle and Number of Cycles
12 × 12 ALPHM = .125	.016	1.46, -1	1.29, -2	1.95, 0	1.50, -1	2.98, -1	.42, NCYC = 3
12 × 12 TOL = 10^{-6}	.082	8.55, -7	1.45, -5	1.05, -5	1.48, -6	3.56, 1	.52, NCYC = 21
12 × 12 ALPHM = .125 IRELAX = 4	.017	8.34, -2	6.34, -3	1.37, 0	1.02, -1	3.32, 1	.02, NCYC = 3
12 × 12 IRELAX = 4 TOL = 10^{-6}	.080	1.89, -7	1.49, -8	3.17, -6	2.22, -7	3.57, 1	.24, NCYC = 11

V. CAVEATS AND EXTENSIONS

The examples of Section IV all exhibit good behavior. We begin this section with three examples which do not. The first is

$$\begin{cases} -\Delta u - \varepsilon u = F \text{ on } \Omega = (0.1) \times (0.1) \\ \dfrac{\partial u}{\partial \nu} = 0 \text{ on } \partial\Omega \quad . \end{cases}$$

One can solve this problem with BOXMG until ε becomes too large; ε is too large when relaxation sweeps on grid G^2 magnify, instead of reduce the error. The remedy would be to change BOXMG to allow the coarsest grid G^1 to be finer; with this remedy BOXMG could be extended to handle some non-definite symmetric problems. See the discussion in Section IV of reference 1.

Another example of poor behavior is for a difference operator with a template like

$$\begin{bmatrix} -\varepsilon & -\varepsilon & -1 \\ -\varepsilon & 2+6\varepsilon & -\varepsilon \\ -1 & -\varepsilon & -\varepsilon \end{bmatrix} , \tag{16}$$

where $\varepsilon \ll 1$. None of the relaxation options in BOXMG provide good smoothing on the finest grid for such an operator. The remedy is to write a block relaxation routine which relaxes the strongly coupled one dimensional sets as blocks; in this case they are the southwest to northeast diagonals. Such a template as eq. (16) can arise in physically meaningful problems; see reference 8, for example. (In reference 8, however, situations like eq. (16) would arise so infrequently as to be not worth the effort of special treatment.)

A final example of poor behavior is when the difference operator on the finest grid is close to the skewed Laplacian (or any operator with strong connections like the skewed Laplacian):

$$\begin{bmatrix} -1 & -\varepsilon & -1 \\ -\varepsilon & 4(1+\varepsilon) & -\varepsilon \\ -1 & -\varepsilon & -1 \end{bmatrix} \quad ,$$

where $\varepsilon \ll 1$. This situation was discussed in reference 3. Here we know of no remedy that would fit into the general framework of BOXMG.

Several extensions of BOXMG are possible and are under investigation. Two, fairly straightforward, are to symmetric systems and three dimensional problems. The third, more difficult, is to nonsymmetric equations. The fourth is to handle equations on arbitrary regions without resorting to embedding. The fifth is local mesh refinement - both fixed and adaptive. For all except the first two extensions, it is not clear at this time how much of the black box philosophy can be retained, and in the third and fourth extensions, it is not clear if there is a uniform strategy for both serial and vector machines.

Finally, we thank Achi Brandt for advice that improved this paper.

VII. CODE LISTING

[In, LA-UR, list.]

A listing of BOXMG is contained in Ref. 7, which may be otained by requesting it from:

J. E. Dendy, Jr.
T-7, MS-610
Los Alamos National Laboratory
Los Alamos, NM 87545

REFERENCES

1. Brandt, Achi: Multi-Level Adaptive Solutions to Boundary-Value Problems. Math. Comp., Vol. 31, 1977, pp. 333-390.

2. Alcouffe, R. E.; Brandt, Achi; Dendy, J. E., Jr.; and Painter, J. W.: The Multigrid Method for the Diffusion Equation with Strongly Discontinuous Coefficients. SIAM J. SISSC (to be published).

3. Brandt, Achi; Dendy, J. E., Jr.; and Ruppel, Hans: The Multigrid Method for Semi-Implicit Hydrodynamics Codes. J. Comp. Phys., Vol. 34, No. 3, March 1980, pp. 348-370.

4. Nicolaides, R. A.: On Multiple Grid and Related Techniques for Solving Discrete Elliptic Systems. J. Comp. Phys., Vol. 19, 1975, pp. 418-431.

5. Hackbush, W.: On the Multi-Grid Method Applied to Difference Equations. Computing, Vol. 20, 1978, pp. 291-306.

6. Hyman, J. M.: Mesh Refinement and Local Inversion of Elliptic Partial Differential Equations, J. Comp. Phys., Vol. 23, No. 2, February 1977, pp. 124-134.

7. Dendy, J. E.: Black Box Multigrid, Los Alamos report LA-UR-81-2337.

8. Hewett, D.: A Global Method of Solving the Electron-Field Equations in a Zero-Inertia-Electron-Hybrid Plasma Simulation Code, J. Comp. Phys., Vol. 38, No. 3, December 1980, pp. 378-395.

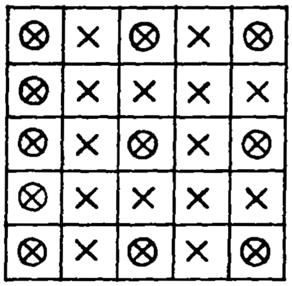


Figure 1.

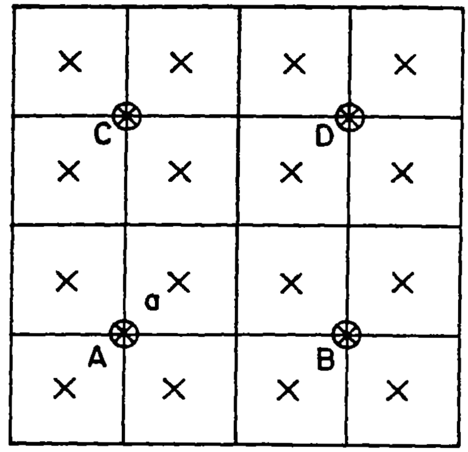


Figure 2.

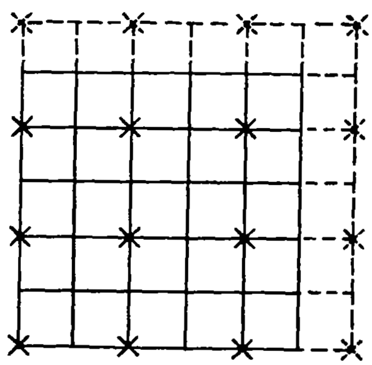


Figure 3.

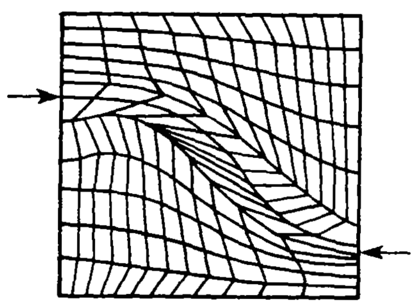


Figure 4.

HIGH ORDER MULTI-GRID METHODS TO SOLVE THE POISSON EQUATION

Steve Schaffer
Colorado State University

I. Introduction

This paper treats several high order multi-grid methods based on finite difference discretization of the model problem:

$$(0) \qquad \begin{aligned} LU &= F \quad \text{on the interior of } \Omega \\ U &= G \quad \text{on } \partial\Omega \end{aligned}$$

Here, L is the Laplace operator, Ω the unit square, and the functions F and G are at least piecewise continuous.

In section II, a fixed high order FMG-FAS multi-grid algorithm (which underlies each of the high order methods) is briefly discussed. In section III, the high order methods are described. In section IV, results are presented on four problems using each method with the same underlying fixed FMG-FAS algorithm. It is noted that optimal efficiency of any one of these methods is not attained with this fixed algorithm, the purpose of the experiments being, rather, to give a comparative point of view for the different methods.

II. The Fixed High Order FMG-FAS Algorithm

A sequence of uniform grids, Ω^h, on the unit square is given, with increasing mesh sizes h = 1/2, 1/4,...1/64. For each h, the discretization of (0) is denoted by:

$$(1) \qquad \begin{aligned} L_s^h U^h &= F^h \quad \text{on the interior of } \Omega^h \\ U^h &= G^h \quad \text{on the } \partial\Omega^h \end{aligned}$$

where L_s^h is a finite difference operator indexed by s that approximates L, G^h is the injection of G onto Ω^h, and F^h is the injection of F onto Ω^h (or some weighted average of F to be described later).

The full approximation scheme (FAS) multi-grid cycle is used to solve equations of the form (1) (see ref. 1). We describe it in the following together with several multi-grid features that are used in all of our experiments. Subsequent reference to an FAS cycle will always imply L_s^h is second order. The grid function, u^h, will represent the current approximation held on Ω^h. We first define the two-level FAS cycle by the following four steps.

1. Given some initial approximation, make 2 Gauss-Seidel relaxation sweeps on (1) using checkerboard ordering of the grid points (ref. 2).

2. Solve the FAS coarse grid equation on Ω^{2h} given by:

$$(2) \qquad L_s^{2h}U^{2h} = L_s^{2h}I_h^{2h}u^h + II_h^{2h}(F^h - L_s^h u^h) \qquad \text{- on the interior of } \Omega^{2h}$$

$$U^{2h} = G^{2h} \qquad \text{- on } \partial\Omega^{2h}$$

Here, the grid transfer operator I_h^{2h} represents injection and II_h^{2h} represents "full weighting" defined by the stencil:

$$\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 4 & 1 \end{pmatrix}_h .$$

3. Correct the current approximation on grid h via:

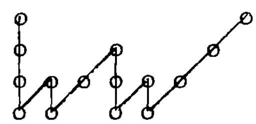
$$(3) \qquad u^h \leftarrow u^h + I_{2h}^h (U^{2h} - I_h^{2h}u^h)$$

where the symbol "$\leftarrow$" represents replacement and I_{2h}^h represents interpolation (linear for second order methods and cubic for fourth order methods).

4. Repeat step 1 with 1 relàxation sweep using the current u^h for the initial approximation.

A multi-level FAS cycle (or, simply FAS cycle) is then defined recursively by using this two-level FAS cycle to solve (2) on Ω^{2h} and similarily on Ω^{4h}. Continuing in this way until equation (2) is formed on grid h = 1/2 where the equation is then solved exactly. We demand two such cycles on each coarser grid, except the coarsest, before correcting the approximation on the next finer grid (step 3). This is called a "W" cycle and can be represented by the following diagram (for h = 1/2, 1/4, 1/8, 1/16).

h = 1/16
h = 1/8
h = 1/4
h = 1/2



where o, | and / represent, respectively, relaxation (step 1 or 4), fine to coarse grid transfer (step 2) and coarse to fine correction (step 3).

Our high order full multi-grid (HFMG) algorithm begins on grid h = 1/4. Using a zero initial guess, one FAS cycle (second order) is made on (1). The resulting approximate solution is then cubically interpolated to grid $h = 1/8$ and is used as an initial approximation for one FAS cycle on equation (1) formulated on this grid. At this point, the final approximation is used as the initial approximation for one of the fourth order methods described in the next section. This fourth order method is then used successively on the next three finer grids using the cubic interpolant of the final approximation on the previous grid as the initial guess. The second order FMG algorithm (ref. 3) proceeds as above without the switch to higher order.

III. High Order Methods

The following finite difference operators are considered.

L_1^h - The usual second order five point star operator.

L_2^h - The operators $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$ are each approximated in their respective directions by fourth order finite differences. For example, at a point $(x,y) \in \Omega^h$, $\partial^2/\partial x^2$ is approximated by the stencil.

$$1/12h^2 \; (-1 \;\; 16 \;\; -\underline{30} \;\; 16 \;\; -1)_{h,x} \tag{4}$$

when $h < x < 1 - h$, and by

$$1/12h^2 \; (10 \;\; -\underline{15} \;\; -4 \;\; 14 \;\; -6 \;\; 1)_{h,x} \tag{5}$$

when $x = h$. The symbol "__" marks the central coefficient.

L_3^h - The operator used in Mehrstellen Verfahren. A second order finite difference operator is used with a weighted average of F to produce the fourth order equation:

$$1/6h^2 \begin{pmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{pmatrix}_h U^h(x,y) = 1/12 \begin{pmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{pmatrix}_h F(x,y) \tag{6}$$

L_4^h - This operator agrees with L_2^h at all points except those whose distance to the boundary is h. At these points, where noncentral differencing occured in L_2^h, the Mehrstellen Verfahren (6) is used.

The fourth order methods are described on a given grid h where the initial approximation is obtained by the HFMG algorithm. Relaxation will always refer to Gauss-Seidel with checkerboard ordering of the points.

MG2 - The second order FMG-FAS algorithm on equation (1), which is used here for comparison.

MGW2 - The second order FMG-FAS algorithm using the full weighting of F for F^h in equation (1).

H_s - High order relaxations. Make three relaxation sweeps on:

(7) $$L_s^h U^h = F^h$$

where s = 2, 3 or 4. Next, solve the coarse grid equation:

(8) $$L_1^{2h} U^{2h} = L_1^{2h} I_h^{2h} u^h + II_h^{2h} (F^h - L_s^h u^h)$$

using an FAS cycle (second order). Then correct the current approximation, u^h, using equation (3) and make one relaxation sweep on (7).

D_s - Outerloop defect corrections (ref. 4). Make one relaxation sweep on the second order equation (1). Form the equation:

(9) $$L_1^h \bar{u}^h = F^h + L_1^h u^h - L_s^h u^h \quad , \quad s = 2, 3 \text{ or } 4$$

on grid h and make an FAS cycle on (9) using only one relaxation sweep on grid at step 1.

τ - The method of τ-extrapolation is based on the assumption that the second order truncation error function, τ^h, has the local expansion:

(10) $$\tau^h \equiv L_1^h U - F^h = Ah^2 + 0(h^4).$$

where the function A is independent of h. Extrapolating, we obtain:

(11) $$\tau^{2h} = \frac{4}{3} (\tau^{2h} - \tau^h) + 0(h^4).$$

If the grid function, u^h, is reasonably close to the second order solution of (1), then it is not difficult to show that if we define:

$$(12) \quad \tau_h^{2h} = \frac{4}{3} [(L_1^{2h} I_h^{2h} u^h - F^{2h}) - II_h^{2h} (L_1^h u^h - F^h)],$$

then

$$(13) \quad \tau_h^{2h} - \frac{4}{3} (\tau^{2h} - \tau^h) = 0(h^4).$$

Thus, it follows from adding τ_h^{2h} to the grid 2h version of (1) that a fourth order equation is produced there. The method, then, proceeds as follows. Make one relaxation sweep on the second order equation (1). Form the extrapolated grid 2h equation:

$$(14) \quad L^{2h} U^{2h} = F^{2h} + \tau_h^{2h}$$

and solve by an FAS cycle. Then correct the current approximation, u^h, using equation (3). At this point, a relaxation sweep on the second order equation (1) is performed only to smooth out the error for further extrapolations on the next finer grid. Otherwise, the corrected u^h will be the final approximation for the τ-extrapolation method.

$W\tau$ - The method of weighted τ-extrapolation is exactly the same as τ-extrapolation except that F^h is defined by the full weighting of F given by:

$$(15) \quad F^h(x,y) = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}_{h/2} F(x,y) = F(x,y) + B(x,y)h^2 + 0(h^4).$$

at points $(x,y) \in \Omega^h$. The function $B(x,y)$ is independent of h. The same arguments used in τ-extrapolation carry over here owing to the similarity of the expansions in (10) and (15).

IV. Numerical Results

In our experiments, the solution to the continuous problem was preselected and the functions F and G were defined accordingly.

PROBLEM 1 (smooth solution)

$$U(x,y) = \sin(\pi x)\sin(2\pi y)$$

PROBLEM 2 (oscillatory solution) On grid $h = 1/16$ the solution contains an average of 4.2 grid points per wavelength.

$$U(x,y) = \sin(\pi(7x + 96))$$

PROBLEM 3,4 (jump discontinuities in the n^{th} derivative of the solution) Using the functions

$$T(x,y) = y - x^2 + x - .75$$

and

$$C(x,y) = \begin{cases} 1 & T(x,y) \geq 0 \\ -1 & T(x,y) < 0 \end{cases}$$

we define

$$U(x,y) = \mathrm{EXP}\,(C(x,y)\,T^n(x,y))$$

where $n = 2$ in problem 3 and $n = 4$ in problem 4. The discontinuity lies along a parabola which passes through the central grid point for all grids and two boundary grid points on all but the coarsest grid.

Operation counts were made for every step of the multi-grid algorithm, including residual formation and transfer and interpolations, where one multiplication was counted as two additions. For each method, the fixed algorithm accumulated $C \cdot N_h$ operations to obtain an approximation on grid h, where N_h is the total number of points on grid h. We report the constant C occurring for each method.

Method	MG2	MGW2	τ	Wτ	H44	H49	H99	D44	D49	D99
C	71	85	85	100	141	141	148	107	107	117

In the experiments, the weighted discrete L_2-norm of the true error, $E^h = U - u^h$, taken at the end of the iteration on each grid h, is given by:

$$A(h) \equiv \Big(\sum_{\substack{i\\j}} (E^h_{ij})^2\Big)^{\frac{1}{2}} \cdot h$$

where the summation is over the interior points of grid h. This discrete L_2-norm makes the norms on different grids comparable. Table 1 lists A(h) (h = 1/16, 1/32 and 1/64) for the 10 methods on problems 1, 2 and 4. The ratios

A(2h)/A(h) are also given to show the relative gain in accuracy by going to a finer grid.

In problem 1, methods MGW2 and Wτ give consistently smaller errors than methods MG2 and τ, respectively. This relationship is also found, though less significantly, in problems 2 and 4. The operator, L_3^h , gives the smallest errors in both the high order relaxation and defect correction methods for problems 1, 2 and 4. The operator L_2^h, gives the largest errors. It should be noted that the Mehrstellen Verfahren discretization, L_3^h, is special to our model problem, but has its generalization in the HODIE methods (ref. 5).

The second order errors obtained by method MG2 on problem 3 were: $A(1/16) = .90 \times 10^{-3}$, $A(1/32) = .66 \times 10^{-3}$ and $A(1/64) = .14 \times 10^{-3}$. The fourth order methods all gave very nearly the same errors as MG2 which would be expected for such a nonsmooth problem.

TABLE 1 - COMPARISON OF ERRORS

PROBLEM	METHOD	$A(\frac{1}{16})$	$A(\frac{1}{32})$	$A(\frac{1}{64})$	$A(\frac{1}{16})/A(\frac{1}{32})$	$A(\frac{1}{32})/A(\frac{1}{64})$
1	MG2	.55(-2)	.14(-2)	.34(-3)	3.9	4.1
	MGW2	.58(-3)	.14(-3)	.35(-4)	4.1	4.0
	τ	.30(-3)	.19(-4)	.12(-5)	16.	16.
	$W\tau$	.19(-3)	.47(-5)	.17(-6)	40.	28.
	H_2	.36(-3)	.22(-4)	.89(-6)	16.	25.
	H_4	.11(-3)	.71(-5)	.44(-6)	16.	16.
	H_3	.26(-4)	.14(-5)	.83(-7)	19.	17.
	D_2	.75(-3)	.38(-4)	.16(-5)	20.	24.
	D_4	.43(-3)	.19(-4)	.90(-6)	23.	21.
	D_3	.11(-3)	.49(-5)	.21(-6)	22.	23.
2	MG2	.17(+0)	.39(-1)	.95(-2)	4.3	4.1
	MGW2	.63(-1)	.15(-1)	.39(-2)	4.2	3.8
	τ	.27(+0)	.84(-2)	.42(-3)	32.	20.
	$W\tau$	.68(-1)	.13(-1)	.77(-3)	5.	17.
	H_2	.20(+0)	.18(-1)	.12(-2)	11.	15.
	H_4	.43(-1)	.30(-2)	.22(-3)	14.	14.
	H_3	.36(-2)	.81(-3)	.88(-4)	4.	9.
	D_2	.20(+0)	.18(-1)	.15(-2)	11.	12.
	D_4	.42(-1)	.66(-2)	.42(-3)	6.	16.
	D_3	.37(-1)	.33(-2)	.25(-3)	11.	13.

TABLE 1 - COMPARISON OF ERRORS (CONT'D.)

PROBLEM	METHOD	$A(\frac{1}{16})$	$A(\frac{1}{32})$	$A(\frac{1}{64})$	$A(\frac{1}{16})/A(\frac{1}{32})$	$A(\frac{1}{32})/A(\frac{1}{64})$
4	MG2	.24(-3)	.60(-4)	.15(-4)	4.0	4.0
	MGW2	.14(-3)	.36(-4)	.91(-5)	3.9	4.0
		.13(-4)	.94(-6)	.64(-7)	14.	15.
	W	.11(-4)	.67(-6)	.36(-7)	16.	19.
	H_2	.59(-4)	.43(-5)	.30(-6)	14.	14.
	H_4	.24(-5)	.24(-6)	.18(-7)	10.	13.
	H_3	.25(-5)	.21(-6)	.16(-7)	12.	13.
	D_2	.73(-4)	.55(-5)	.41(-6)	13.	13.
	D_4	.17(-4)	.15(-5)	.12(-6)	11.	13.
	D_3	.58(-5)	.44(-6)	.26(-7)	13.	17.

References

1. Brandt, Achi; Multi-Level Adaptive Solutions to Boundary-Value Problems, Math. Comp. 31, 1977, pp. 333-390.

2. Foerster, H., Stuben, K. and Trottenberg, U.; Non-Standard Multi-Grid Techniques Using Checkered Relaxation and Intermediate Grids, "Elliptic Problem Solvers", (Martin Schultz, et.), Academic Press, New York, 1980.

3. Brandt, Achi; Multi-Level Adaptive Finite-Element Methods, Variational Problems, "Spectial Topics of Applied Mathematics", (J. Frehse, D. Pallaschke, U. Trottenberg, eds.), North-Holland Publishing Co., 1980.

4. Stetter, Hans, J.; The Defect Correction Principle and Discretization Methods, Numer. Math. 29, 1978, pp. 425-443.

5. Schaffer, Steve; High Order Multi-Grid Methods, Ph.D. Thesis, Colorado State University, Dec., 1981.

Accelerated Convergence of Structured Banded Systems Using Constrained Corrections*

Karl Kneile
Sverdrup Technology, Inc.

1. INTRODUCTION

The goal of this paper is to describe an efficient iterative method for solving a structured banded system of equations. While the method was developed for a full potential flow program, it will be presented in general terms applicable to a wide range of problems. The central issue here is the solution of a large linear system of equations. The linear system may arise directly in the problem or may result from an iteration in a nonlinear problem. For large 2-D and 3-D applications, this linear system becomes increasingly expensive to solve directly. As a result efficient iterative methods have become attractive for large problems. In the nonlinear cases, these iterations may be effectively merged to improve convergence rates.

Conventional iterative methods (Jacobi, Gauss-Seidel, ADI, etc.) rapidly reach a state where convergence rates are limited by the large eigenvalues of the system. This phenomenon is especially restrictive for large problems. Various approaches have been tried to accelerate convergence. Relaxation made some modest gains, but obtaining an optimum or near optimum parameter was sometimes difficult. Others tried more elaborate iterative methods (incomplete Crout, strongly implicit procedure (SIP), and SIP/conjugate gradient) with considerable success. However, the most dramatic improvements have been seen recently with the revival of multigrid concepts.

The method presented in this paper uses a basic iteration step (incomplete Crout reduction), a dynamic relation step, and a multigrid concept of constraining iterative corrections.

* The work reported herein was conducted for the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC) by Sverdrup Technology, Inc., an operating contractor for the AEDC. Further reproduction is authorized to satisfy needs of the U. S. Government.

2. METHOD OF CONSTRAINED CORRECTIONS

The method of constrained corrections uses a variational form of the problem. This variational form may be part of the problem definition or may be artificially created as described later.

The discretized variational form may be represented as $L(\phi)$, where L is a scalar function of the n component vector ϕ. The components of ϕ are obtained by solving the n simultaneous equations

$$F = \partial L/\partial \phi = 0 \tag{1}$$

An iterative procedure (Newton's) for solving this system is

$$A\delta = r$$

where (2)

$$\delta = \phi_{i+1} - \phi_i \ , \ r = -F_i \ , \text{ and } A = \partial F/\partial \phi|_i = \partial^2 L/\partial \phi^2|_i$$

For linear problems the iteration process is trivial ending with the first iteration.

If a variational principle is not part of the problem, one can define

$$L^*(\delta) = \tfrac{1}{2}\delta^T A\delta - \delta^T r \tag{3}$$

and use

$$\partial L^*/\partial \delta = 0 \tag{4}$$

as the variational form. This is equivalent, both here and in later considerations, to using

$$\partial L(\phi_i + \delta)/\partial \delta = 0 \tag{5}$$

coupled with Newton's method if (5) is nonlinear in δ. The method of constrained corrections defines δ as

$$\delta = Ck \tag{6}$$

The vector k has $p < n$ components. The matrix C prescribes each component of δ as a linear combination of the components of k. It will be assumed that C is of rank p (i.e., the columns of C are linearly independent). This is equivalent to imposing (n-p) linear combinations of the components of δ as zero. That is

$$C^{*}\delta = 0 \tag{7}$$

where C^* is an $(n-p) \times n$ matrix. It is more convenient to use these constraints in the form of (6).

Substitution of (6) into the variational form results in the system of equations

$$C^{T}AC\ k = C^{T}r \tag{8}$$

for the unknown k vector. The δ vector is then obtained from (6). With judicious selections of C and k, the convergence rate can be substantially improved.

3. DYNAMIC RELAXATION

Consider again the basic linear system (2). When using an iterative method, one obtains an approximate δ denoted by δ_a. By letting C in (6) be the vector δ_a, the relaxation parameter k is then given by

$$k = \delta_a^T r / \delta_a^T A \delta_a \tag{9}$$

The iteration then takes the form

$$\phi_{i+1} = \phi_i + k\delta_a \tag{10}$$

The residual r in (9) is the original residual vector using ϕ_i and not from using $\phi_i + \delta_a$.

4. INTERPOLATED FORM

It is easier to describe this form for one dimensional problems. The components of ϕ are associated with a positional value along this dimension. As mentioned earlier, conventional iterative methods rapidly reach an asymptotic convergence limited by the larger eigenvalues. It is well known that these iterations rapidly remove the smaller wavelength components, leaving the smooth longer wavelength components. Conventional multigrid methods exploit this smoothness to justify using a coarse grid operator. This paper will also take advantage of this smoothing property, but will direct emphasis towards the smoothness of the correction vector δ. A basis vector δ_b is selected. The correction vector is then constrained to the following form

$$\delta = C\delta_b = \begin{bmatrix} I \\ \hline B \end{bmatrix} \delta_b \tag{11}$$

In actual practice the components of δ_b are interspersed within δ and the rows of B are merged with the rows of the identity matrix. The above representation (separated I and B) will be used to simplify notation. For this special case the constraints take the form

$$C^* \delta = \left[\begin{array}{c|c} B & -I \end{array}\right] \delta = 0 \tag{12}$$

The B matrix represents interpolation coefficients for the nonbasis components. Solutions of

$$C^T AC \ \delta_b = C^T r \tag{13}$$

coupled with (11) will then "solve the original problem" subject to the constraints. The effectiveness of this constrained form depends upon the form of interpolation used, the smoothness of δ, and the difficulty in solving the new system (13). The smaller the dimension of δ_b, the simpler system (13) is to solve. However, more iterations are needed to precondition the smoothness required for effective interpolation.

The dynamic relaxation step described in the previous section can be used to improve overall convergence. The relaxation factor may be calculated using the basis δ_b.

$$k = \delta_b^T (C^T r) / \delta_b^T (C^T AC) \delta_b \tag{14}$$

5. MULTIGRID

The constrained corrections method described in the previous section can be easily adapted to a multigrid concept. A nested sequence of basis vectors is defined by

$$\delta_o = \delta$$

$$\delta_{i-1} = C_i \delta_i \qquad i = 1,2,\cdots,m \tag{15}$$

$$C_i = \begin{bmatrix} I \\ \hline B_i \end{bmatrix}$$

where δ_o represents the original fine grid and δ_m the coarsest level with the fewest components. The B_i are interpolation coefficients from the i^{th} level to the (i-1) level. The above interpolations may be combined to form

$$\delta_o = D_i \delta_i$$
$$D_i = C_1 C_2 \cdots C_i = D_{i-1} C_i \tag{16}$$

The constrained system of equations is

$$D_i^T A D_i \ \delta_i = D_i^T r \tag{17}$$

These equations easily lend themselves to the following iterative algorithm.

A smoothing pass is made on the fine grid system.

$$A\delta = r \tag{18}$$

The ϕ vector is updated by (10), whereby (18) now represents the next iterative pass. A compression step is taken to obtain the system

$$A_1 \delta_1 = r_1$$

where

$$A_1 = D_1^T A D_1 \tag{19}$$

and

$$r_1 = D_1^T r$$

A smoothing pass is now made on this system. The ϕ vector is again updated and the next iterative pass is taken at the second level

$$A_2 \delta_2 = r_2$$

where

$$A_2 = D_2^T A D_2 \tag{20}$$

and

$$r_2 = D_2^T r$$

The process is repeated down through the coarsest level.

The above describes a multigrid cycle. This cycle is then repeated until sufficient convergence is obtained. Many variations of the above algorithm are possible. A few of these are compared in Section 8.

A computational advantage can be obtained from the nesting or recursive definitions of the D_i. The next level system can be calculated directly from the current level

$$
\begin{aligned}
A_{i+1} &= C^T_{i+1} A_i C_{i+1} \\
r_{i+1} &= C^T_{i+1} r_i
\end{aligned}
\tag{21}
$$

It is not necessary to calculate the updated residuals at the fine grid level. They may be calculated at the current i^{th} level and then compressed down one level.

6. STRUCTURED BANDING

Consider those problems where a quantity ϕ is to be determined over a 2-D or 3-D space. The space is discretized by an $(n_1 \times n_2)$ or $(n_1 \times n_2 \times n_3)$ grid. The A matrix in (2) takes a structured banded form. That is, only a few of the diagonals of A have nonzero elements. The particular structure of A depends upon the approximations used in describing the original equations at the grid points. This paper will cover the details for a nine diagonal structure typical of a 2-D finite element approach. Adaptations to other type problems should not pose any difficulties. Occasional comments concerning other type problems will be made at appropriate places.

The structure of A can be considered as a block tridiagonal system where the blocks are also tridiagonal. This simple structure allows computationally efficient smoothing algorithms. It is therefore desirable to maintain this structure through the multigrid levels. This imposes limitations on the interpolation matrix B. For simplicity, a 1-D case will be described first. The structured banded A matrix is tridiagonal. In order to maintain this tridiagonal structure, the interpolation for any point must be limited to a nearest neighbor principle. That is, an interpolated point is a function of only the two neighboring points, one on each side. This suggests a linear interpolation.

Figure 1 depicts a case where the δ vector has been smoothed. The corresponding C matrix has the form (Note: The rows of B and I are merged)

$$
C^T = \begin{bmatrix}
1 & 1/2 & & & & & & & \\
 & 1/2 & 1 & 2/3 & 1/3 & & & & \\
 & & & 1/3 & 2/3 & 1 & 3/4 & 1/2 & 1/4 & \\
 & & & & & & 1/4 & 1/2 & 3/4 & 1
\end{bmatrix}
\tag{22}
$$

In the typical multigrid patterns where every other point is used, C has the following form for a nine point to five point compression.

$$C^T = \begin{bmatrix} 1 & 1/2 & & & & & & & \\ & 1/2 & 1 & 1/2 & & & & & \\ & & & 1/2 & 1 & 1/2 & & & \\ & & & & & 1/2 & 1 & 1/2 & \\ & & & & & & & 1/2 & 1 \end{bmatrix} \tag{23}$$

The nearest neighbor principle for 2-D problems is shown in Figure 2. The coarse grid basis is represented by solid dots. The open circles are interpolated points. The arrows point to the basis components that are used for interpolation. This interpolation can be factored into two 1-D steps. Figures 3a and 3b show the two steps. The first step reduces from a 5 x 5 grid to a 5 x 3 grid. The second then reduces down to a 3 x 3 grid. The same principle will factor a 3-D problem in three steps.

Interpolation for unequal spacing and irregular geometries is more involved. A convenient alternative is to interpolate as if the geometry were regular with equal spacings. This retains the calculations in a simple form. The result is a "nonlinear form" of interpolation. The longer wavelength information is still passed down to the coarser level. The interpolation errors, as with linear interpolation, are short wavelength in nature and are reduced with the next smoothing pass at the current level.

7. SMOOTHING PASS

One of the key elements of the multigrid algorithm is that the wavelength components comparable to grid size must be damped before going to a coarser level. Fortunately, this is the strong point of the conventional iterative methods. The method emphasized in this report is incomplete Crout reduction. Two variations are used in this report. The methods are identical to a complete Crout reduction with the following modifications during the forward pass. In the short version when zeroing an element below the main diagonal, all operations which modify an off-diagonal element are not performed. The result is a quick efficient iteration which damps out the short wavelength error components. In the long version, all operations which modify the nonzero structured banded elements are kept. All other operations which would modify zero elements are not performed. The long version has better iterative properties at the expense of the additional work required. For the nine point star 2-D case, the increase in operations is about 60%.

8. TEST CASE

The preceding methods were used to solve Laplace's equation on a rectangular grid. Dirichlet conditions were imposed at the boundaries. The primary goal of this test case was to verify the method and help in comparing alternatives. The system (2) was obtained using isoparametric quadrilateral finite elements. The basic algorithm consisted of the following. A multigrid cycle of (m + 1) levels was used (m = 0 means fine grid only). The coarser levels were obtained by removing every other point in each dimension. Each level contained one smoothing pass. (The short version of incomplete Crout followed by a dynamic relaxation.) The results (convergence rates) are given in the number of work units required to reduce the error by one order of magnitude. A work unit is defined as the time to set up a fine grid system and make one smoothing pass. It was assumed that the time spent at a lower level was one fourth that of the next higher level. For comparison with conventional multigrid methods it is also assumed that the time required to compress down to the next lower level is equivalent to that of evaluating the operator at that level. For simple linear problems such as Laplace's or Poisson's equation, the operator evaluation should be quicker. However, for nonlinear problems such as full potential flow, the compression step will probably be faster. The rates given are estimates of the asymptotic rate. They were obtained by iterating until the rates "leveled off". In cases where convergence showed cyclic or erratic behavior, an average of a selected final group of iterations was used. Most of the results are given for 9 x 9, 17 x 17, and 33 x 33 grids with equal spacings. A few results where $n_1 \neq n_2$ and where $\Delta X \neq \Delta Y$ are given at the end of the section.

Table 1 shows the convergence rates for the basic algorithm.

$n_1 x n_2$ \ m	0	1	2	3	4	5
9 x 9	3.8	1.1	1.2	1.2		
17 x 17	11.5	2.1	1.3	1.3	1.3	
31 x 31	39.7	6.6	1.8	1.3	1.3	1.3

TABLE 1. Convergence Rates for Basic Algorithm

Without multigrid levels, the convergence rate rapidly deteriorates with increasing grid size. Using multigrid levels improves the convergence rates for each grid size, and the results appear independent of grid size. The table indicates that the extra coarse grids are not needed, but nothing is lost by the conservative attitude of using more levels than needed. For comparison purposes the 1.3 convergence rate is equivalent to an error reduction factor of 0.17 per work unit or 0.093 per multigrid cycle. This table should be used as a reference for comparison of alternative methods given in this section.

Table 1 considers the case where the $n_1 \times n_2$ grid points are all interior to the boundary. The boundary conditions had been transferred to the right hand side of the equations. An alternative would be to include the boundary points as part of the $n_1 \times n_2$ grid. The boundary points have no error and the neighboring points are interpolated using this zero error boundary. The results are shown in Table 2.

$n_1 x n_2$ \ m	0	1	2	3	4	5
9 x 9	2.7	1.3	1.2	1.2		
17 x 17	9.2	2.3	1.2	1.3	1.3	
31 x 31	34.3	9.5	2.0	1.4	1.4	1.4

TABLE 2. Convergence Rates for Alternate Boundary Conditions

The faster convergence at the zero level is due to the fewer non-zero error components. However, where multigrid levels are used, this trend is reversed. The difference is small though, when sufficient levels are used. Ease of application should probably be the deciding factor.

Table 3 shows the results when a Jacobi ADI method is used for smoothing. The dynamic relaxer was applied after each of the two ADI sweeps. The multigrid convergence rates are about the same as the incomplete Crout reduction. The main reason for choosing the incomplete Crout was its efficiency. It is easily programmed. For large systems it requires 5 divides (with common divisor), and 12 multiply-add combinations per equation. If applied to a 5 point star system obtained from finite differences, the operation count is 3 divides and 6 multiply-adds per equation.

$n_1 x n_2$ \ m	0	1	2	3	4	5
9 x 9	6.6	1.4	1.2	1.3		
17 x 17	21.1	3.5	1.5	1.4	1.3	
31 x 31	76.6	12.0	2.6	1.5	1.4	1.3

TABLE 3. Convergence Rates for ADI Smoothing

Several alternative methods of cycling through the multigrid levels were tried. The order did not significantly change the asymptotic convergence rates. That is, it makes no difference whether one starts with the fine grid and works down to the coarse or vice versa. Attempts at weighting the coarse grid passes were also tried. Table 4 shows results where the number of passes at each level varied linearly from 1 for level 0 to 6 for level 5.

$n_1 x n_2$ \ m	0	1	2	3	4	5
9 x 9	3.8	1.4	1.6	1.6		
17 x 17	11.5	1.4	1.6	1.7	1.7	
31 x 31	39.7	3.6	1.6	1.7	1.7	1.7

TABLE 4. Convergence Rates for Weighted Passes

Some improvement is noticed for a few cases where the number of levels are insufficient. However, when sufficient levels of grid are used, equal weighting (one per pass) is better. The degradation of convergence rate is due to the additional work involved. If the rates were given on a "per multigrid cycle" basis, they would be about equal to those using one pass per level. It is important to emphasize that these are asymptotic rates. It was noticed that the weighted cycle was superior in the initial stages. This was attributed to the smooth errors present with the initial guesses. Such a phenomenon is likely with actual problems. Therefore, it is suggested that the early passes be weighted towards the coarser levels with later passes of one per level.

Table 5 shows results without using the dynamic relaxer.

$n_1 x n_2$ \ m	0	1	2	3	4	5
9 x 9	6.7	2.0	1.3	1.3		
17 x 17	20.8	5.5	1.6	1.4	1.4	
31 x 31	73.0	18.6	4.9	1.5	1.4	1.4

TABLE 5. Convergence Rates Without Dynamic Relaxer

While the relaxer made significant improvements at the zero level, only modest gains were achieved when multigrid levels were used. Since its cost is minimal, the dynamic relaxer was retained in the basic algorithm. Its potential gain for other applications may be significant. For example, a 25% savings in time was obtained in the multigrid/ADI cases. For comparison purposes, a fixed optimum parameter was determined by trial and error. Convergence rates for the dynamic relaxer and the fixed optimum were essentially the same.

Simultaneous relaxation parameters were also tried. The corrections at each level were saved and used as columns of C in (6). The system of equations (8) can then be solved for the relaxation parameters (components of k). The results were disappointing. Only trivial gains were noticed, not worth the extra work and storage required.

An interesting alternative is to use a constant times the residuals as the smoothers. The dynamic relaxer can be used to determine the unknown constant. Convergence was erratic with rates in the 2.0 to 4.0 range. For linear problems, this rate would be attractive since the time per iteration is minimal. For nonlinear problems, calculating the fine grid system and compression to coarser levels takes most of the time, and the overall rates would be considerably slower. A problem with this alternative is the relative scaling of each equation.

Several rectangular grids were also tried. The convergence rate for a 9 x 33 grid was in the 1.2 to 1.3 range that was obtained for the square grids. The incomplete Crout smoother is order dependent. That is, different results are obtained depending upon whether the grid is numbered by rows or by columns. The convergence rates for the 9 x 33 grid were essentially unaffected by the direction of node ordering. Ordering along the short dimension gave less than a 2% improvement over the other direction.

Figure 4 shows results for varying aspect ratios. A 33 x 33 grid was used in this study. The solid line shows the results using the short incomplete Crout reductions. The nodes were numbered in the Y direction. As the figure indicates, the convergence rate rapidly becomes impractical for even moderate aspect ratios. Numbering the nodes in the X direction gave the same behavior. The dashed line shows results for the long version of incomplete Crout reduction (nodes numbered in the Y direction). The worst convergence ratio occurs at an aspect ratio of about 10. About twice as many iterations are needed at this aspect ratio. At larger ratios, the convergence rate rapidly improves. Ordering the nodes in the X direction gives the results shown by the dotted line. Except for a small increase at the smaller aspect ratios, this ordering gave better results. The unexpected improvement at large aspect ratios is probably due to the regular rectangular geometry. Extrapolation of these results to more general geometries would be speculative. Further study is needed particularly in the selection of the smoother.

9. SUMMARY

A constrained corrections algorithm was described in the previous sections. The method was used to solve Laplace's equation on a rectangle. A convergence rate of 1.3 fine grid work units per decade reduction in error was obtained.

The algorithm uses a multigrid concept with the following components.

(1) Incomplete Crout reduction is used to smooth the errors.

(2) A dynamic relaxation parameter is used.

(3) Coarse grid systems are obtained by constraining the corrections at the fine grid level. These constraints are in the form of simple interpolation.

The method has some drawbacks. The system of equations needs to be stored. Recalculation of the fine grid at each level would increase the computational effort by a factor approximately proportional to the number of levels used. For nonlinear problems, updating the nonlinear parts can only be accomplished at the fine grid level. Another drawback occurs with the simple forms typical of finite difference methods. For example, a 2-D finite difference method usually uses a 5 point star rather than 9 points. The interpolation used in this paper will not maintain this 5 diagonal system, but expands to a 9 diagonal system.

The main advantage of the method is the influence of the interpolation formulas. The coarse grid systems contain not only the "average residuals", but fine grid geometry information and the implied interpolation of the solution back to the fine grid. It is expected that this unification between the multigrid phases will prove advantageous when general distorted geometries are used. The method is easy to use and does not require guesswork for determining parameters. Simple interpolation forms are used, producing efficient iterations.

It is the opinion of the author that the advantages will outweigh the disadvantages. A 3-D full potential program is being developed using the method in this paper.

10. NOMENCLATURE

A	Coefficient Matrix for a linear system of equations
A_i	Coefficient Matrix at i^{th} multigrid level
B	Interpolation coefficient matrix
B_i	Interpolation coefficient matrix from level i to level (i-1)
C	Augmented interpolation coefficient matrix
C_i	Augmented interpolation coefficient matrix from level i to level (i-1)
C^*	Matrix defining linear constraints implied by C
D_i	Interpolation coefficient matrix from level i to fine grid
F	Vector defined by $\partial L/\partial\phi$
k	Vector of unknowns in constrained correction formulation
L	Variational form
L^*	Alternate variational form
m	Maximum number of levels used ($m = 0$ means fine grid only)
n_1, n_2, n_3	Number of nodes in a given direction of the grid
r	Residual vector
r_i	Residual vector at i^{th} level
δ	Correction vector
δ_a	Correction vector approximation
δ_b	A basis vector used for interpolation
δ_i	Basis vector at the i^{th} level
ϕ	Solution vector
ϕ_i	i^{th} iteration for the solution vector

$(\ )^T$	Transpose operator
$\partial L/\partial\phi$	A vector composed of derivatives of L with respect to elements of ϕ.
$\partial F/\partial\phi$	Matrix composed of derivatives of the elements of F with respect to elements of ϕ. The F components determine rows and the ϕ components determine columns.
$\partial^2 L/\partial\phi^2$	Alternate notation for $\partial F/\partial\phi$. The matrix composed of second derivatives of L with respect to elements of ϕ.

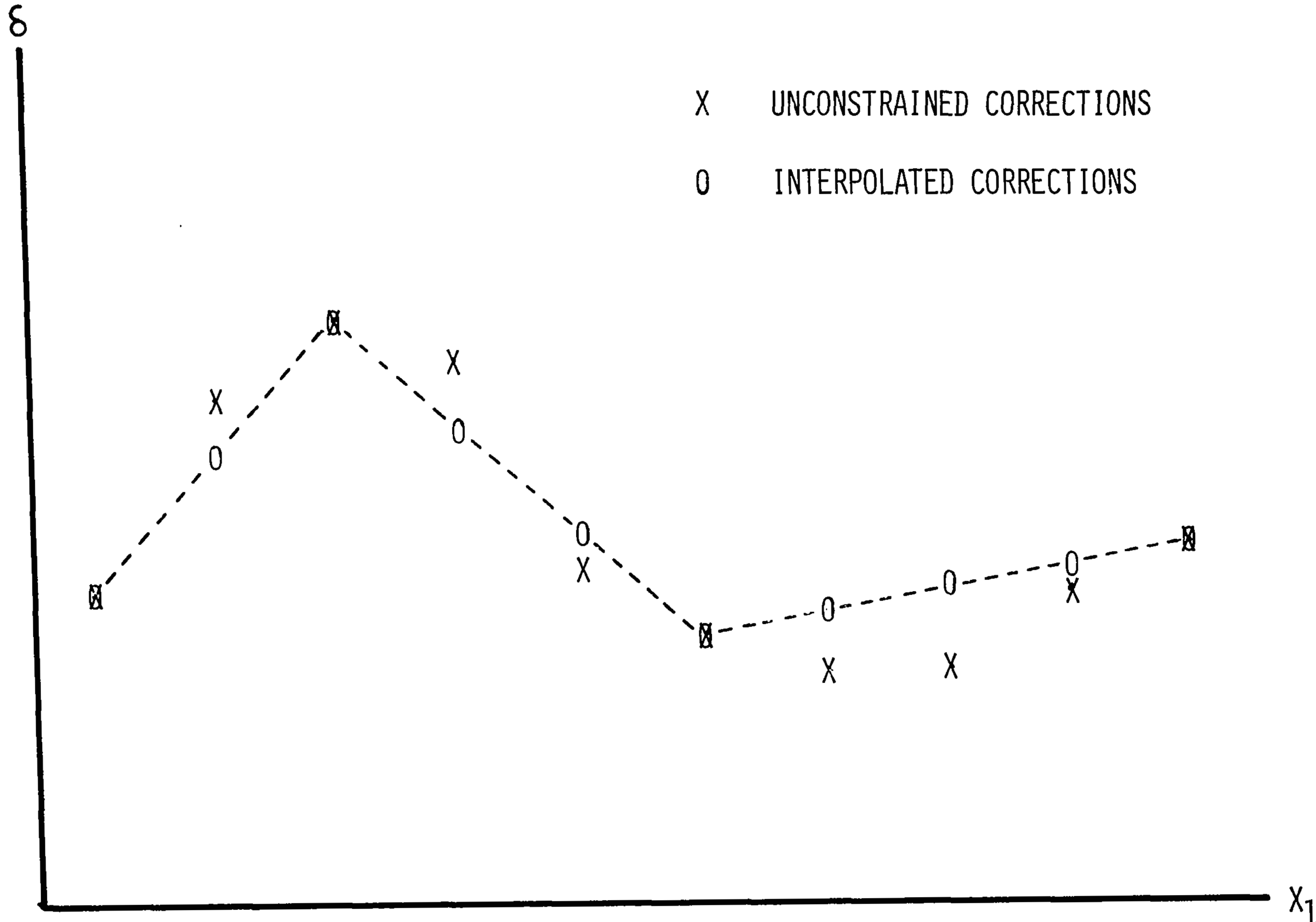


Figure 1.- 1-D Interpolation.

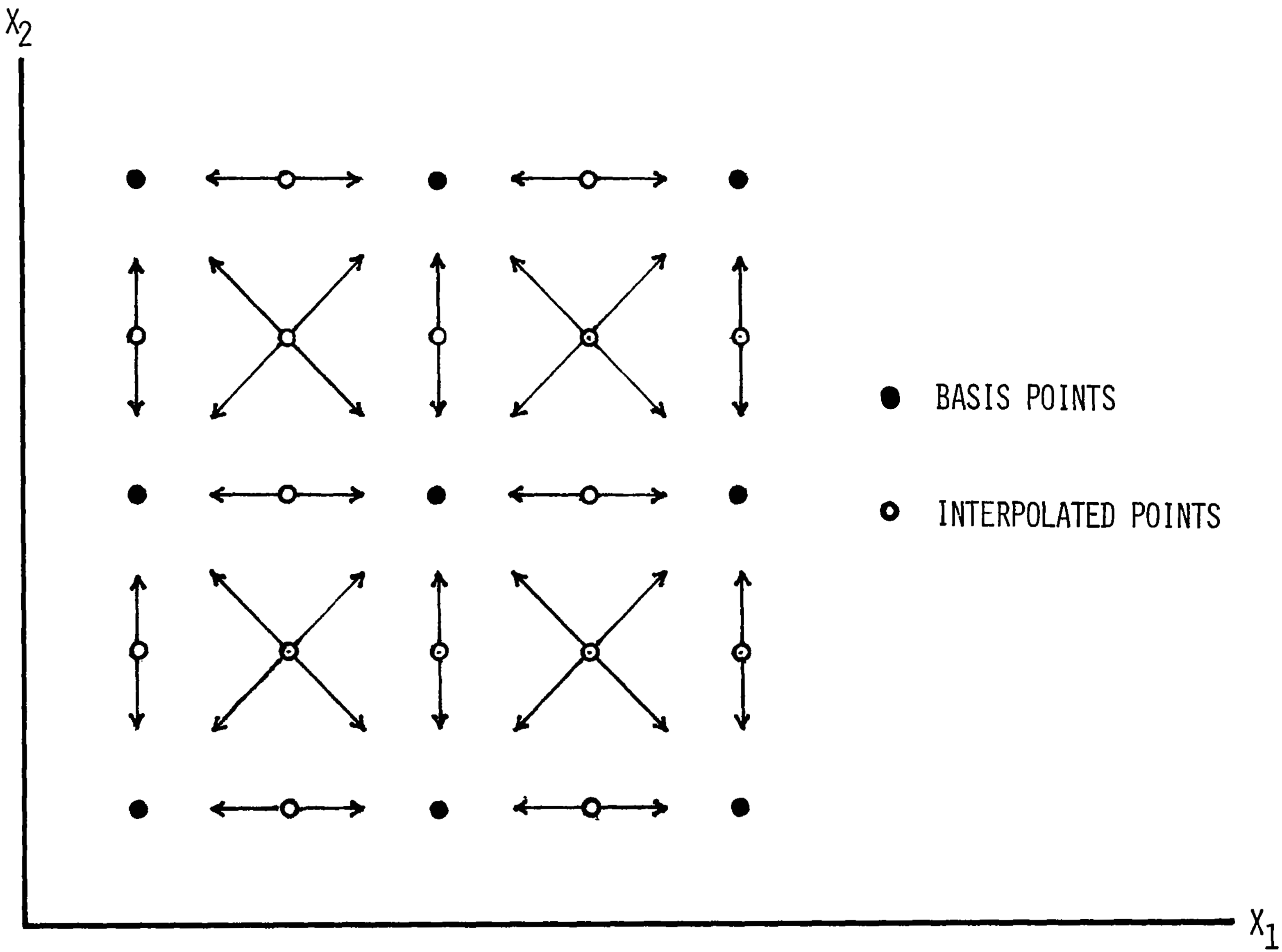
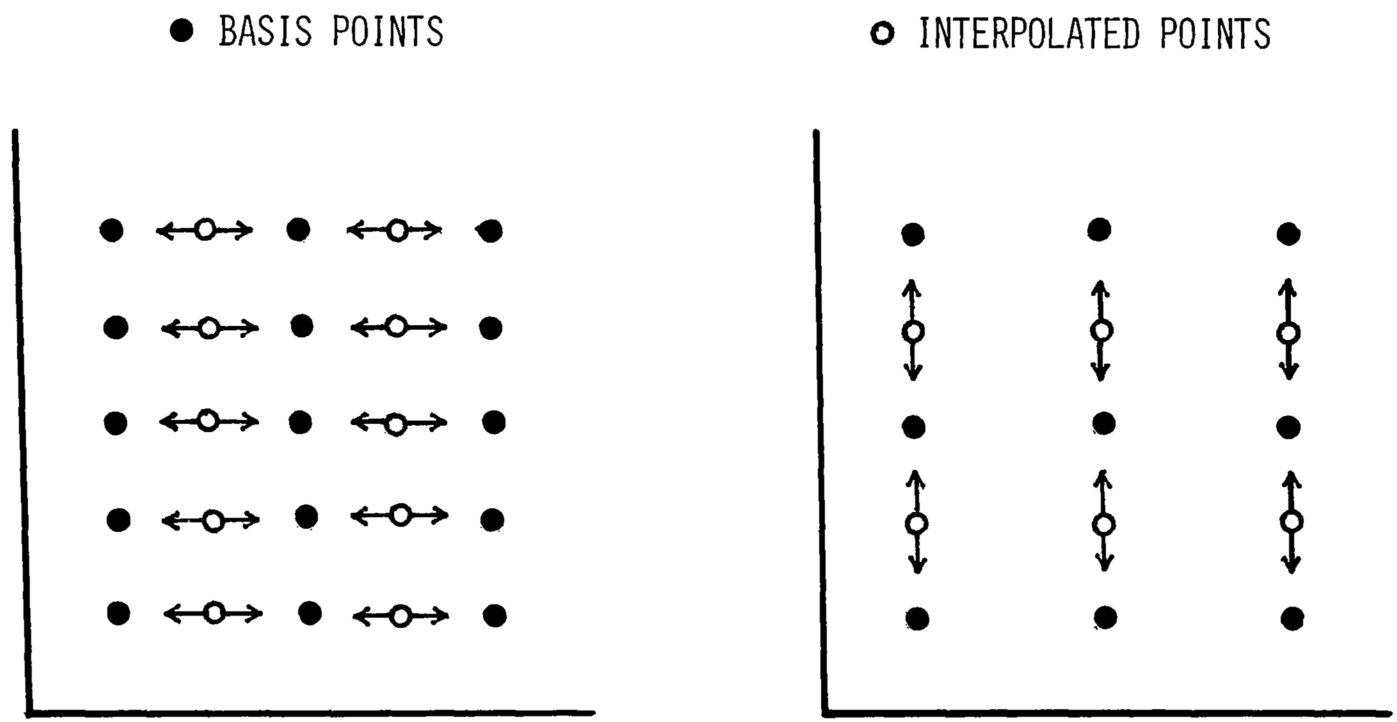


Figure 2.- 2-D Interpolation.



(a) Step #1. (b) Step #2.

Figure 3.- 2-D Interpolation in Split Form.

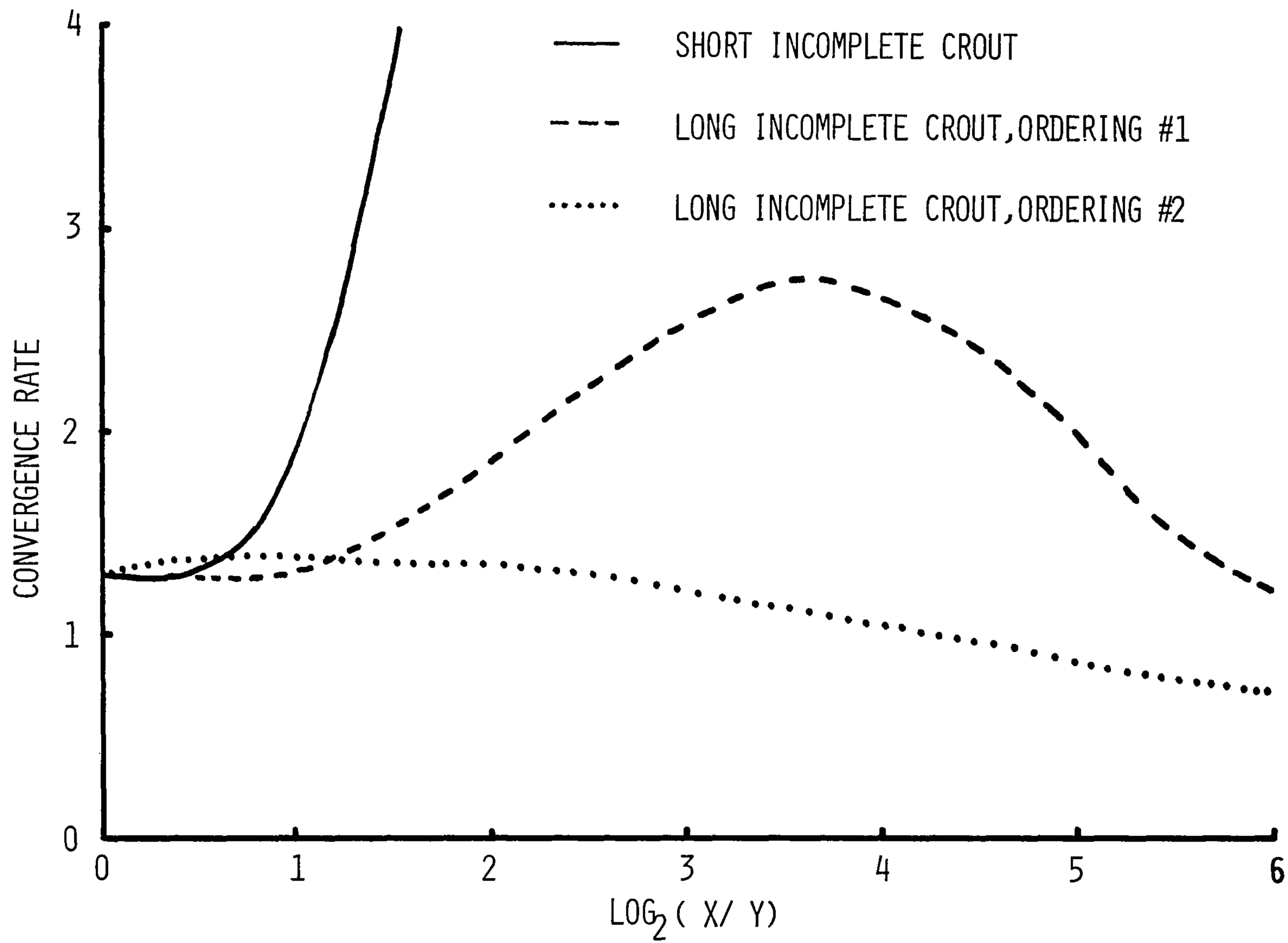


Figure 4.- Aspect Ratio vs. Convergence Rates

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CP-2202		

4. Title and Subtitle MULTIGRID METHODS	5. Report Date March 1982
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No. A-8738
9. Performing Organization Name and Address NASA Ames Research Center Moffett Field, California 94035	10. Work Unit No. 505-31-11
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	13. Type of Report and Period Covered Conference Publication
	14. Sponsoring Agency Code

15. Supplementary Notes A symposium held at NASA Ames Research Center, Moffett Field, CA 94035, October 21-22, 1981
Harvard Lomax, General Chairman (415) 965-5124; FTS 448-5124

16. Abstract

This Conference Publication is a compilation of papers presented at the Symposium on Multigrid Methods held at NASA Ames Research Center on October 21-22, 1981. The papers represent an international sampling of the most recent developments in the numerical solution of certain types of partial differential equations using multigrids, whereby rapidly converging sequences of operations are carried out on supporting grids ranging in size from very fine to very coarse in covering a bounded space.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computational fluid dynamics Numerical methods Multigrid methods	Unclassified - Unlimited STAR Category 64

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	310	A14

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1982