Titre: Title:	Software package for the multigrid method
Auteurs: Authors:	Ricardo Camarero
Date:	1979
Type:	Rapport / Report
Référence: Citation:	Camarero, R. (1979). Software package for the multigrid method. (Technical Report n° EP-R-79-13). https://publications.polymtl.ca/5994/

Institution:	École Polytechnique de Montréal
Numéro de rapport: Report number:	EP-R-79-13
URL officiel: Official URL:	
Mention légale: Legal notice:	

MATHÉMATIQUES

Rapport Technique EP79-R-13

Classification: Library of Congress no

A SOFTWARE PACKAGE FOR THE MULTIGRID METHOD

RICARDO CAMARERO

February 1979

Ecole Polytechnique de Montréal

Campus de l'Université
de Montréal
Case postale 6079
Succursale 'A'
Montréal, Québec
H3C 3A7

A SOFTWARE PACKAGE FOR THE MULTIGRID METHOD

by

Ricardo Camarero

Department of Mathematics
Ecole Polytechnique

February 1979

TABLE OF CONTENTS

CHAPTER 1
AIMS AND OBJECTIVES

In the present report, an investigation concerning the application of a numerical method for the solution of boundary value problems is described. The particular method is based on a relaxation procedure where the convergence is accelerated by means of the multigrid method. It consists in approximating the same continuous problem by a succession of discretizations on different grids.

This method has been proposed by Brandt (1) and an excellent and through discussion is found in references (1) and (2). In effect this report is a repetition these references where the explanation and demonstrations have been recast in terms closer or more readily accessible to a user with typically a mathematical background acquired in an engineering school. The objective was to explain the multigrid method and supply a software package to allow the solution of engineering applications in fluid mechanics. It was felt that this could be done in a less abstract if somewhat less rigorous and more restricted manner than the works of references (1) and (2).

The final objective is the development of a software package computationally efficient and very simple to apply by a user. An application is usually reduced to the writting of two related subroutines which are the discretization of the differential equation of the problem.

CHAPTER 2
THE BASIC APPROACH OF MULTIGRID

2.1 INTRODUCTION

The basic approach of multigrid consists in solving an equation on a given discretized grid by an interaction between a hierarchy of coarser grids. It is a combination of two classical processes namely a relaxation procedure to reduce the residuals of an approximate solution and a technique of approximation the same problem on coarser grids. The multigrid method is best motivated from the limitations of the classical relaxation methods.

The method of relaxation is the usual procedure used for the numerical solution of boundary value problems. This consists in updating an approximate solution iteratively in such a way as to reduce the error at each node of the grid. A sweep is complete when every node of the grid has been updated. When at any stage of the relaxation procedure, the current approximate solution is substituted into the discrete operator, one obtains a residual different from zero since the solution has not been reached. The rate at which this residual tends to zero is called the convergence of the relaxation procedure. It depends on a variety of factors such as the type of equation, the particular relaxation scheme (line relaxation, overrelaxation). This is of course the most important characteristic of such procedure and has been extensively analysed by Brandt (1). These findings are now summarized in a qualitative manner. Let us suppose that after a number of relaxation sweeps, the

residual (for a one-dimensional problem) is as shown in Figure 1.

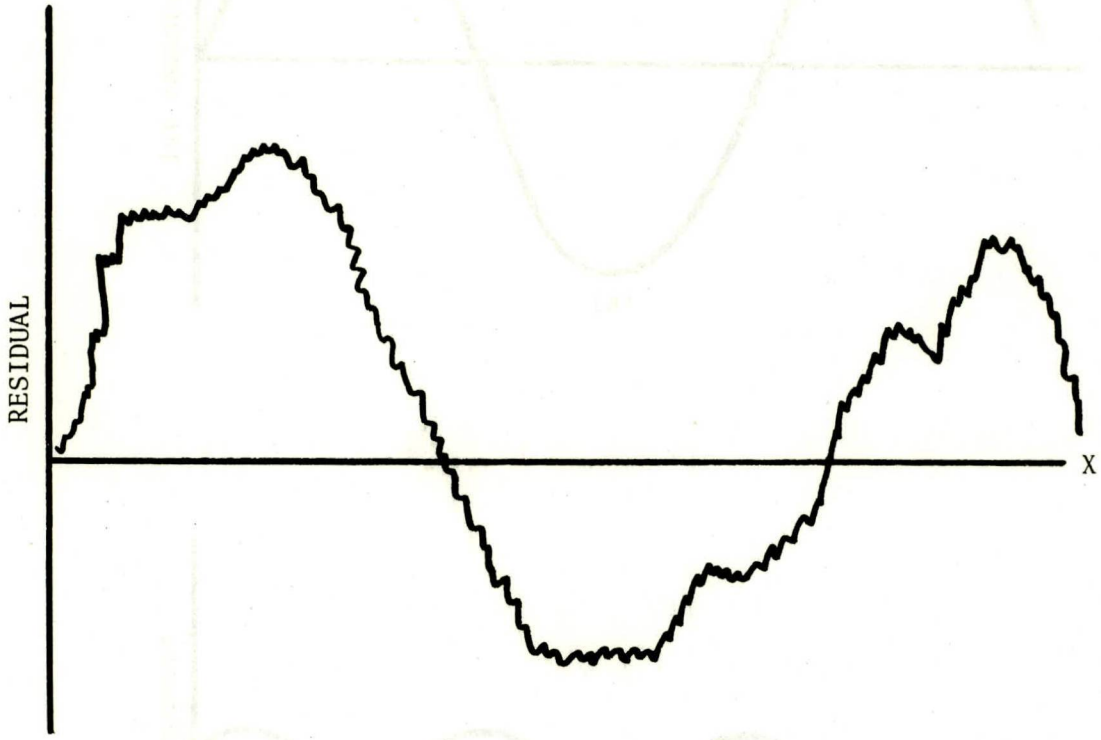


FIG. 1

This residual can be decomposed into its various Fourier components, which for illustration purposes will be taken as shown in Figure 2.

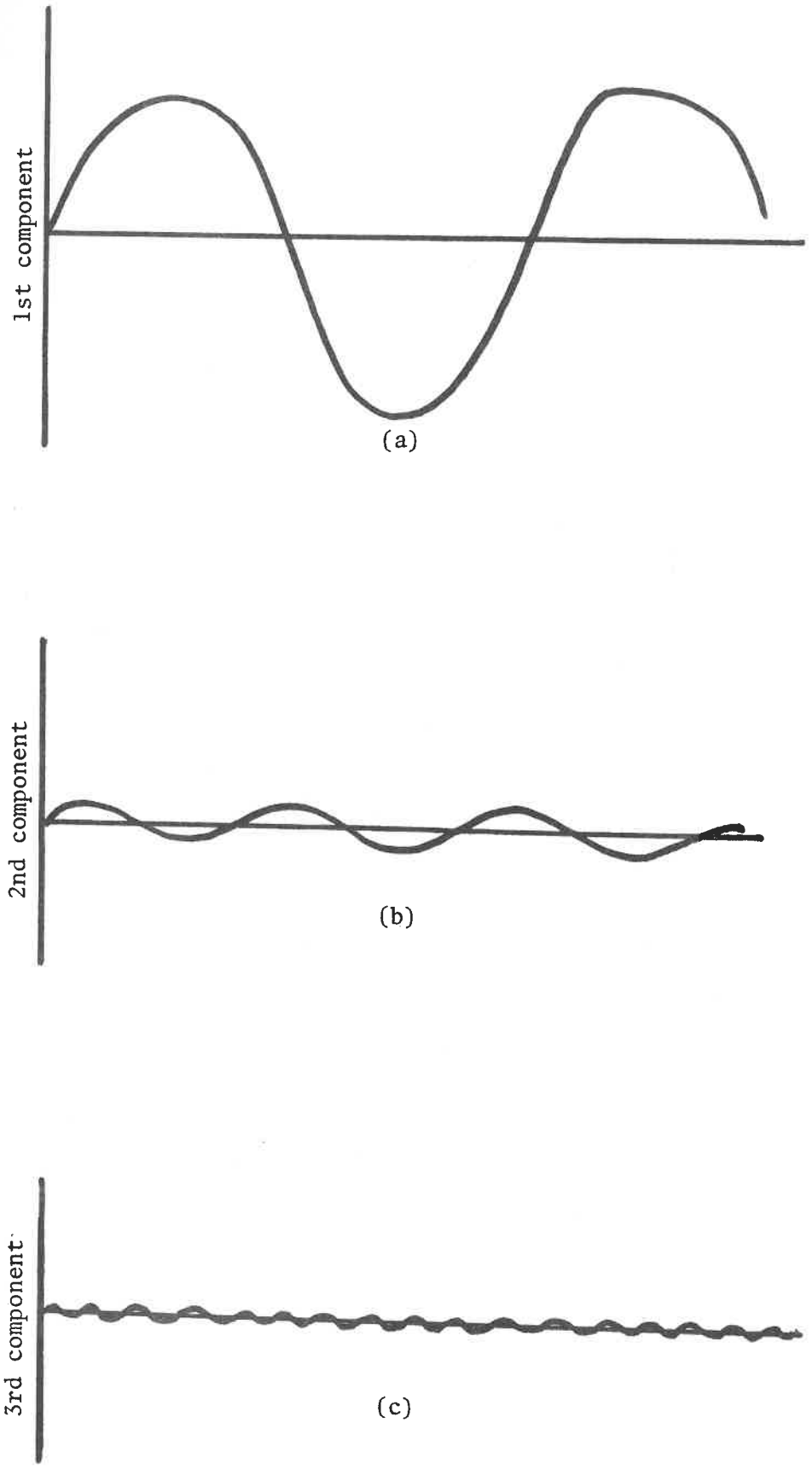


FIG. 2

The purpose of a relaxation procedure is to reduce the residual illustrated in Figure 1. Brandt (1) has shown that a given scheme does so selectively, that is will reduce certain components more quickly than others, and these are those which are of the same order as the mesh width of the discretized grid. For instance, a relaxation procedure on a grid whose spacing is of the order of half the wavelength, will be very efficient in liquidating the component of the residual corresponding to Figure 2.c, that is initially the convergence is very high. However when this component has dissapeared, the same scheme becomes totally inefficient in liquidating the other two components. This explains the general behaviour of the convergence rate of relaxation scheme, sketched in Figure 3.

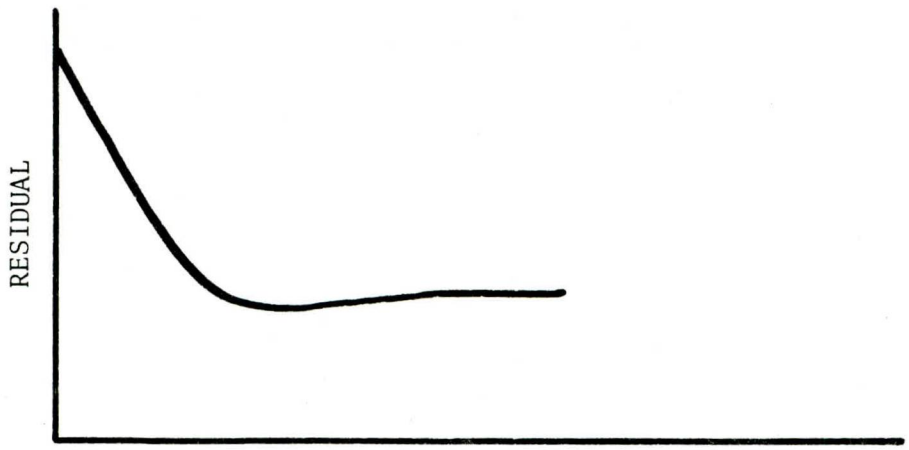


FIG. 3

Quantitatively, the convergence is characterized by an initially steep part during which the components of the order of the mesh width are smoothed out, followed by a very slow decrease thereafter.

Hence the objective of multigrid is to maintain the initial rate of convergence and it does so by relaxing simultaneously on several grid. Thus several frequency components of the residual are liquidated simultaneously yielding a very efficient relaxation procedure.

This is in essence the basic approach of multigrid and this is carried out by an interplay of relaxation sweeps and corrections from coarser to finer grids.

2.2 DEFINITIONS AND NOTATION

Before proceeding with a detailled description of these basic steps a number of definitions and notation convention are given. These follow those of Brandt (1). This section can be omitted on a first reading.

The computational domain Ω is discretized by a set of grids denoted by $G^0, G^1, \ldots, G^M$ with corresponding mesh sizes $h_0 > h_1 > \ldots h_M$. In general the mesh is not square and the spacings in the coordinate directions are dx_K and dy_K for the Kth grid, G^K. The mesh ratio $\zeta = h_K/h_{K-1}$ is for simplicity constant for any two consecutive grids but this need not be so. The values for ζ of $\frac{1}{2}$ and $\frac{1}{3}$ have been used in the present investigation. Non-interger ratio can be used in principle but this will complicate the interpolation procedures. The differential equation to be solved is denoted by

$$LU(x) = F(x) \quad \text{in the region} \quad \Omega \tag{1}$$

where L is a differential operator, linear or non-linear depending on the particular problem. For examples one can refer to Section 4. $U(x)$ is a solution to (1) subject to the following boundary condition

$$\Lambda U(x) = \Phi(x) \quad \text{on} \quad \partial\Omega \tag{2}$$

This form is general and includes Dirichlet and Newmann conditions.

The discrete representations of the continuous operators L and Λ, and the solution U are labelled L^K, Λ^K and U^K respectively, for the discretizations on G^K. This gives the following difference equations

$$L^K U^K(x) = F^K(x) \tag{3}$$

for values of x on the grid G^K, subject to

$$\Lambda^K U^K(x) = \Phi^K(x) \tag{4}$$

for values of x on the boundary ∂G^K.

In the multigrid process a numerical approximation to the solution U^M on the fine grid is sought and denoted by u^M. Substituting u^M in equations (3) and (4) yields the corresponding residual on the fine grid G^M

$$L^M u^M(x) - F^M(x) = -f^M(x) \tag{5}$$

and

$$\Lambda^M u^M(x) - \Phi^M(x) = -\phi^M(x) \tag{6}$$

We define a correction V^M as

$$V^M = U^M - u^M \tag{7}$$

Communication between grids is carried out by interpolating some quantity, i.e. a correction or a solution. This is denoted by

$$I^K_{K-1} V^{K-1}$$

meaning the interpolation of V^{K-1} on G^{K-1} to G^K. In general

$$I^{K-1}_K I^K_{K-1} V^{K-1} \neq V^{K-1}$$

For non-linear problems, variational operators relative to an approximate solution u are defined as

$$\tilde{L}(u)V \equiv L(u+v) - Lu \tag{8}$$

2.3 MULTIGRID CYCLES

Brandt (2) proposes a number of variations to carry out the relaxation in the multigrid sense. In all cases an approximate solution on a fine grid is corrected successively by relaxations on a number of coarse grids. This series of relaxation sweeps is called a multigrid cycle. Of the three cycles proposed by Brandt the simplest has been chosen, i.e. cycle A. This choice was made in view of the particular context in which these application are intented (see section 1). Primarily, the overall programming required by the user should be restricted mainly to that of his problem. Secondly, in practical engineering problems it may not be possible to resort to very many levels of coarser grids as required by cycle B. Finally some ideas of cycles B and C are incorporated into the present version of cycle A. The theory is now described.

We want to improve an approximate a given solution u^M on a grid G^M of a problem described by equations (1) and (2). In most cases this is an initial guess. The exact solution U^M is obtained by adding a correction V^M to u^M. As discussed previously this correction (or error at this stage) is made up of several Fourier components which are most efficiently obtained if computed individually on different grids. Thus,

$$\begin{aligned} L^M(u^M + V^M) &= F^M \\ \Lambda^M(u^M + V^M) &= \Phi^M \end{aligned} \tag{9}$$

If the operators L and Λ are linear, then

$$\begin{aligned} L^M(u^M) + L^M(V^M) &= F^M \\ \Lambda^M(u^M) + \Lambda^M(V^M) &= \Phi^M \end{aligned}$$

and one can solve the residual problem

$$L^M(V^M) = f^M$$
$$\Lambda^M(V^M) = \Phi^M \tag{10}$$

where f^M and Φ^M are given in equations (5) and (6) respectively. For non-linear problems, the correction V^M will satisfy a more general variational equation,

$$L^M(u^M) + \tilde{L}^M(V^M) = F^M$$
$$\Lambda^M(u^M) + \tilde{\Lambda}^M(V^M) = \Phi^M$$

which gives the following residual problem

$$\tilde{L}^M(V^M) = f^M$$
$$\tilde{\Lambda}^M(V^M) = \Phi^M \tag{11}$$

This is similar to the residual problem of the linear case with L and Λ replaced by the corresponding variational operators relative to the approximate solution u^M as defined in equation (8). These operators $\tilde{L}$ and $\tilde{\Lambda}$ may be quite tedious to obtain in general, and there is a variation which allows the use, in the residual problem of the same operator as for linear problems, thus simplifying considerably the task of the user. This idea is called the full approximation mode and consists in working with the full current approximation instead of the current correction. The current correction is the approximation on G^K to V^M and is denoted by v^M. Similarly the full current approximation is the approximation to u^M on G^K and is defined as the interpolation of u^M on the grid G^K corrected by the current approximation, that is

$$u^K = I_M^K u^M + v^K \tag{12}$$

Writting equation (9) on G^K,

$$\begin{aligned} L^K(U^K) &= F^K \\ \Lambda^K(U^K) &= \Phi^K \end{aligned} \tag{13}$$

If the approximation u^M and the current correction v^K are sufficiently smooth (and this can be easily be achieved) then

$$U^K \approx I_M^K u^M + v^K \tag{14}$$

and upon substitution into the operator one obtains

$$\begin{aligned} L^K(I_M^K u^M + v^K) &\cong L^K(U^K) \\ \Lambda^K(\qquad\quad) &\cong \Lambda^K(U^K) \end{aligned}$$

which is rewritten as

$$\begin{aligned} L^K(I_M^K u^M) + \tilde{L}^K(v^K) &= L^K(U^K) \\ \Lambda^K(\quad) + \tilde{\Lambda}^K(\ ) &= \Lambda^K(U^K) \end{aligned} \tag{15}$$

The residual problem on the fine grid G^M can be approximated on coarser grids by

$$\begin{aligned} \tilde{L}^K(v^K) &= I_M^K f^M \\ \tilde{\Lambda}^K(v^K) &= I_M^K \Phi^M \end{aligned} \tag{16}$$

where v^K is an approximation to V^K. Substracting (16) from (15) and rearranging, yields

$$\begin{aligned} L^K(U^K) &= I_M^K f^M + L^K(I_M^K u^M) \\ &= \tilde{F}^K \\ \Lambda^K(U^K) &= I_M^K \Phi^M + \Lambda^K(I_M^K u^M) \\ &= \tilde{\Phi}^K \end{aligned} \tag{17}$$

This is called the full approximation because one no longer solves the residual on coarser grid but rather one obtain the full approximation u^K on the current grid. The main advantage is that the same operator is used on all grids, thus avoiding the need to obtain the variational operator, and this, even for non-linear problems. Consequently this particular variation was chosen other advantage appear more evidently at the programming stages and will be discussed then.

When expanding the right hand side of equation (17) one obtains the following expression.

$$\begin{aligned} L^K(U^K) &= I_M^K F^M - I_M^K L^M u^M + L^K(I_M^K u^M) \\ \Lambda^K(U^K) &= I_M^K \Phi^M - I_M^K \Lambda^M u^M + \Lambda^K(I_M^K u^M) \end{aligned} \tag{18}$$

The original problem when computed on the coarser grids is now modified by the addition of the two terms

$$-I_M^K L^M u^M + L^K(I_M^K u^M) \tag{19}$$

which clearly represent the difference of the discretization error of the operator on the fine and coarse grids. This can be interpreted as a forcing term which accelerates the convergence or overrelaxes in a manner some what different from the classical overrelaxation factor. The latter is related to the physical space whereas the effect of the forcing terms of equation (19) is felt in the frequency spectrum of the residual of the approximating solution.

2.4 THE ALGORITHM

The following steps describe the final algorithm for the full approximation mode of the multigrid relaxation. This differs slightly with Brandt's version (2).

1) Start with an initial guess u^M on the fine grid, G^M,

2) This approximation solution is smoothed by s relaxation sweeps

$$u^M = \text{Relax} \begin{matrix} L^M \cdot = F^M \\ \Lambda^M \cdot = \Phi^M \end{matrix} u^M \tag{20}$$

3) The residuals on G^M are stored

$$\begin{aligned} f^M &= F^M - L^M u^M \\ \phi^M &= \Phi^M - \Lambda^M u^M \end{aligned} \tag{21}$$

4) The residual equation (17) is solved on the coarsest grid G^0

$$\begin{aligned} L^0 U^0 &= \tilde{F}^0 \\ \Lambda^0 U^0 &= \tilde{\Phi}^0 \end{aligned} \tag{22}$$

where the right hand sides of equation (22) are computed by interpolating the residuals, equation (21), and the solution equation (20) from G^M to G^0 and substituting into equation (17). If equation (22) are solved by relaxation sweeps over G^0, one can start with

$$u^0 = I_M^0 u^M$$

5) A correction v^0 is computed on G^0

$$v^0 = I_M u^M - u^0$$

and interpolated to the next grid, G^1,

6) The full approximation on G^k is solved by s relaxations sweeps

$$u^k = \text{Relax} \begin{matrix} L^K. = \tilde{F}^K \\ \Lambda^K. = \tilde{\Phi} \end{matrix} \quad \tilde{I}^K_{K-1} u^K \tag{23}$$

where the starting solution $\tilde{I}^K_{K-1} u^K$ is obtained by

$$\tilde{I}^K_{K-1} u^K = I^K_M u^M - v^{K-1}$$

7) A correction v^K is computed on G^K

$$v^K = I^K_M u^M - u^K$$

and is interpolated to the next grid

$$K = K+1$$

8) If $K < M$, steps 6) and 7) are repeated,

9) If $K = M$, the original equation is solved on G^M by relaxation in step 2) and the result becomes the new approximation and one multigrid has been completed. Additional cycles are obtained by carrying out steps 2) to 9).

2.5 THE STARTING SOLUTION

For linear problem it is not necessary to spend much effort to obtain a good first approximation, and setting $u^M = 0$ will produce a convergent overall algorithm. For the nonlinear problems which are presently investigated it was found that simplest procedure is to set $u^M = 0$ and execute a few relaxation sweeps on G^M (this is step 2) of the algorithm). This will result is a reasonnably smooth initial guess from which multigrid can proceed. For highly nonlinear problem, a continuation technique may be used. Such problems have been attempted so far in the present investigation.

Relaxation procedure

The most important element in the multigrid method are the relaxation procedures which are used to solve the residual equation on the coarse, fine and intermediate grids. To keep the programming to a minimum it is suggested that the same procedure be used for all three steps. Further this should be carried out by the same subroutine and varying the arguments of the call statement. This is possible as the same operator is used throughout. This is done quite efficiently in terms of computer time and memory as shown in the next section where the software aspects are discussed.

Two types of relaxations schemes are been used, line, column relaxation as well as point relaxation, including an overrelaxation parameter. An alternating line and column relaxation scheme has been programmed for a system of two equations. A discussion and references to these various schemes is given in the section on applications.

2.6 RESIDUALS

It has been found practical and time saving to compute the residual on a given grid G^K resulting from a given approximate solution u^K by a distinct step from that of the relaxation sweeps. Here again the same subroutine is used in all cases with varying the arguments of the call statement. And this is possible since the same operator is used on all grids.

2.7 INTERPOLATION

In the present algorithm interpolation of the approximate solution and the correction is required. In the first case, the interpolation must

be carried out from the fine grid G^M to coaser grids G^K; in the second case the interpolation is always from a grid G^K to the next, G^{K+1}. The most important characteristic of an interpolation is the order. For the multigrid method Brandt (2) has shown that the order should at least be equal to that of the differential equation being solved. As the problems envisaged were described by second order equations, second and third order interpolation schemes were written. The applications were two dimensional problems but to simplify this step the interpolation routines are respectively bi-quadratic and bi-cubic respectively. The next most important parameter, is the mesh ratio ξ which in principle can have any value. Again to keep the interpolation manageable, simple ratio of 2 and 3 (depending on the application) were used. This allowed considerable simplification of the interpolation problem, but just as important it allowed to write generalized relaxation as will be discussed in the next section. The main advantage for the interpolation is that simple injection for the step I_M^K is possible. Furthermore, because of the particular correspondence between the physical grids and the storing matrices for the variables on these grids, this step required no calculation and no programming statements and is more efficient than simple injection. The step I_K^{K+1} is carried out by two distinct routines specialized respectively for mesh ratios of 2 and 3. The disadvantage of this approach is the resulting constraint on the number of nodes. For example, for $\xi = 2$ and second order interpolation, the number of nodes must be odd and furthermore satisfy the relation 2^M+1 where M is the number of grids. For a mesh ratio of 3, the constraint is that the number of nodes satisfy 3^M+1.

CHAPTER 3
PRACTICAL ASPECTS AND SOFTWARE

In the present investigation the multigrid method was applied to a number of engineering problems and in this context a particular approach for the solution of the software problems has evolved. It consists of a main program, an interpolation routine, and two user-supplied subroutines. The main program sets the initial solution, the various grids and makes the appropriate calls to the subroutines, in the sequence described in the previous section. The function of the user-supplied subroutine is, given an approximate solution to either compute the corresponding residual or to perform a number of relaxation sweeps. The choice of the relaxation method rests with the user, however it must be written in a particular form which is now described.

There are two arrays named U(I,J) for the solution and V(I,J) for the correction. The solutions and corrections on all grids use these arrays and no other are required. These are dimensionned for the fine grid. In the discretization of the differential equation, the user uses the following method. A first order derivative

$$\frac{\partial U}{\partial X} \approx \frac{U(I + INCK, J) - U(I - INCK, J)}{2\ DXK}$$

where DXK - mesh spacing of G^K

INCK - number of nodes for the increment on G^K

Similarly,

$$\frac{\partial U}{\partial Y} \approx \frac{U(I, J + INCK) - U(I, J - INCK)}{2\ DYK}$$

The values of INCK, DXK, and DYK are related and depend on the mesh ratio between successive grids. For instance, if $\rho = 2$

```
INCK = 2**(KG)
 DXK = DX*INCK
 DYK = DY*INCK
```

where DX and DY are the mesh spacing on the fine grid, i.e. on G^M, and KG is the grid number. It is noted that the index KG does not correspond to the variable K of the previous section. In fact it runs opposite, i.e. KG = 1 corresponds to G^M and KG = M+1 corresponds to G^0.

Using these ideas, the finite difference for $\partial^2 U/\partial X^2$ is written as

$$\frac{U(I + INCK, J) - 2U(I,J) + U(I - INCK, J)}{DXK*DXK}$$

It is stressed that the operator L of the problem must be discretized accordingly. In this manner, the computation of the residual, for instance, can be carried out with the same subroutine with different values of INCK and the same array U is used for all grids. This eliminates the need for the interpolation, or injection, of U^M to coarser grids, and the storage of these interpolated or injected functions does not required any additional variables.

In the present algorithm the array V(I,J) represents two quantities. It is used to store the current full approximation and also to store the current correction. It is possible to use the same array for both of these quantities because after the full approximation is computed by the relaxation routine, the correction is obtained by substracting it from the array U. It is noted that this correction is calculated only at the nodes of G^K, i.e. at every INCK node, and stored in the array V. This correction is then

interpolated to the next grid. A bi-quadratic interpolation is used and essentially generates an additional value in between every two nodes along a column, and a additional column between every two columns. This is also stored in the array V, from which the full approximation is computed on the new grid by adding V to U at the appropriate nodes, now separated by the new value of INCK. Thus during the course of a multigrid cycle the array U contains the approximate solution u^M and the array V contains, alternately the full approximation u^K and the current correction v^K.

In the supplied routine called RELAX, the user writes a relaxation procedure based on the operator of the particular problem to solve the full approximation equation (17) or (23) where L is the same operator as in problem and where $\tilde{F}$ and $\tilde{\Lambda}$ are modified forcing terms defined by the right hand side of Equation (17). They contain the original forcing term of the problem to which Equation (17) has been added. The quantities $\tilde{F}$ and $\tilde{\Lambda}$ are stored in the array FRCING and are obtained by two calls to the user supplied routine called OPERA for the computation of the operator. A first call will yield the operator on the fine grid with the array U as an argument.

A first call with the array U as an argument and INCK set to one (i.e. the fine grid) will yield

$$L^M u^M$$

This quantity is stored in an array OPERI and is thus available for all the following operations

$$I_M^K L^M u^M$$

A second call (repeated for every grid for a given multigrid cycle) with the array U and the appropriate value of INCK as arguments will yield the term

$$L^K(I_M^K u^M)$$

It is repeated that with the present approach none of the interpolations or injections denoted by I_M^K are actually performed. The quantities $\tilde{F}$ and $\tilde{A}$ are passed to RELAX in the array FRCING.

CHAPTER 4
APPLICATIONS AND CONCLUSIONS

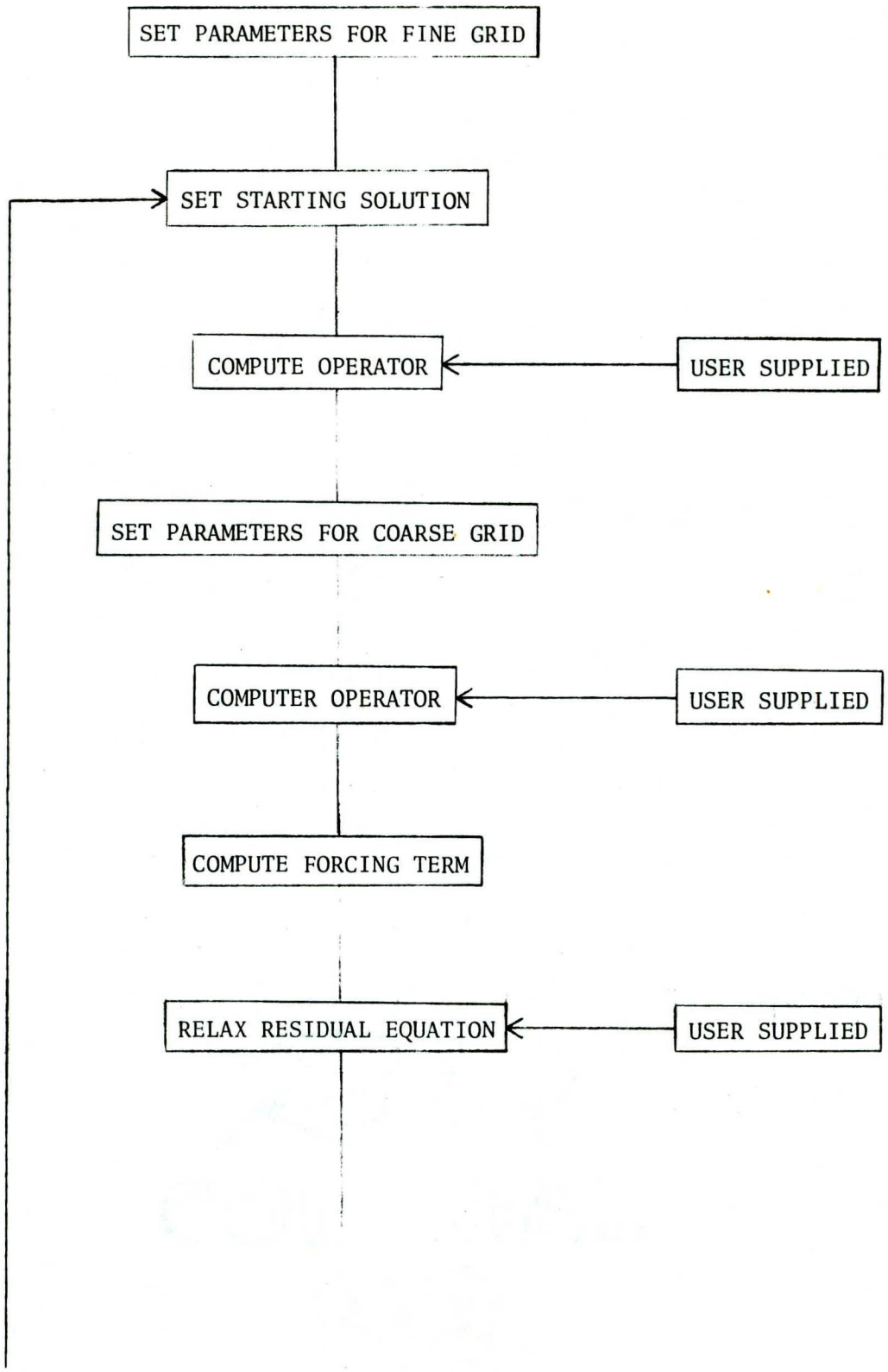
Multigrid has been applied to the solution of transsonic flow problems by J.C. South and A. Brandt (3), semiconductor transport equations by S.P. Gaur and A. Brandt (4) and aeronautics by A. Roberts (5). In the present study, the multigrid method was applied to a variety of fluid mechanics problems. The small perturbation equation for the transonic flow in a channel (Ref. 6) was solved using the Nurman approach with multigrid. The problem is similar to that of Ref. 3 except that the flow is within a channel rather than an airfoil. This was then extended to the case of an the flow past a non-lifting cascade (Ref. 7). The objective was to test multigrid with different types of boundary conditions; namely periodic, Neumann and the Kutta conditions.

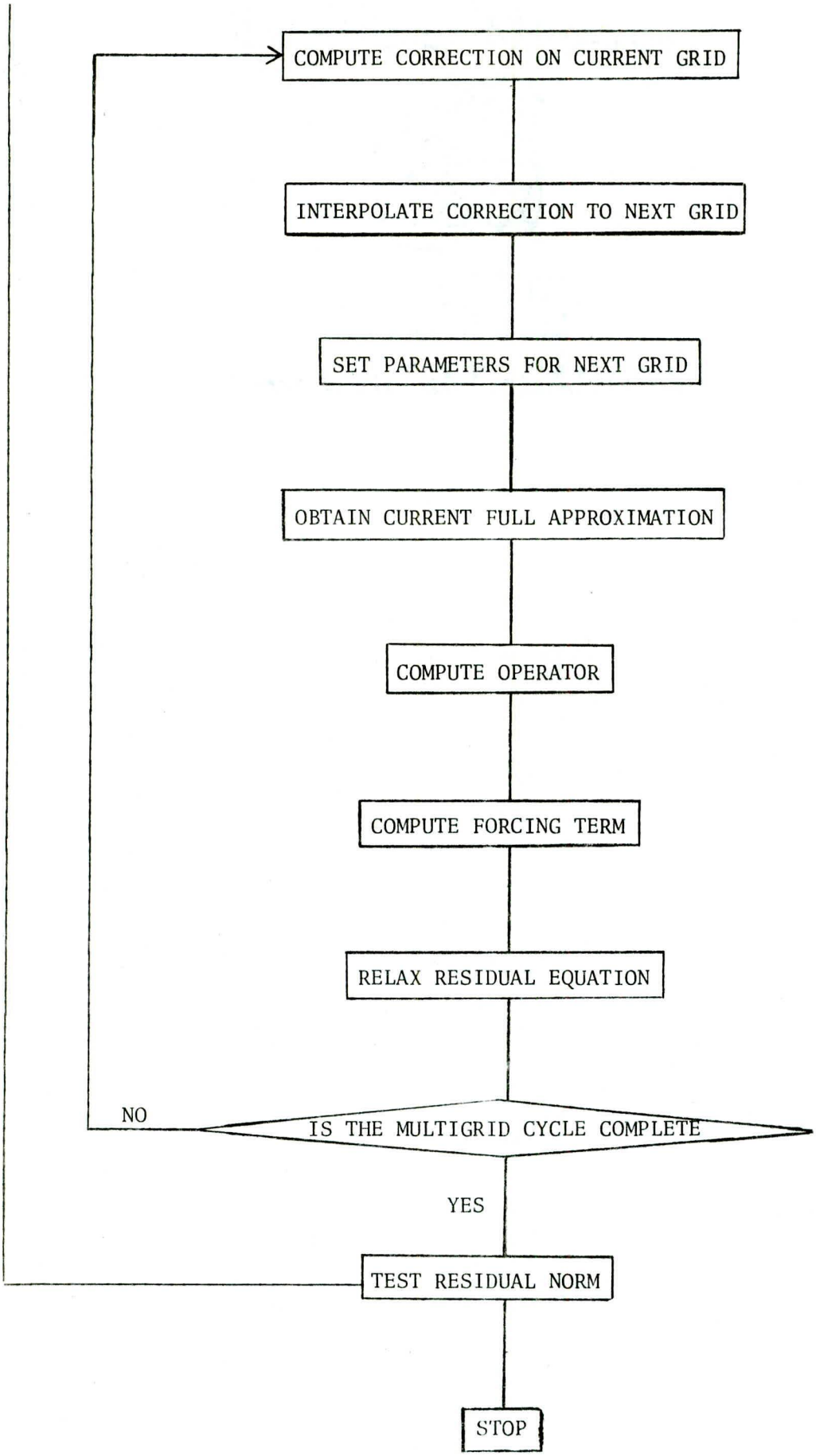
The solution of the Navier-Stokes was then attempted. The problem solved is the unsteady viscous flow is a cavity with an impulsively started wall. The vorticity stream function formulation is used, and yields to equations. The first is the vorticity diffusion equation and is solved by the A.D.I. method. The second is a Poisson equation where the forcing term is the vorticity obtained from the first equation. This Poisson equation is solved using multigrid (Ref. 8).

Finally, two problems concerning the generation of body fitted coordinates were solved. The first is the generation of a computational grid for cascade flows. This consists in solving two non-linear Poisson-type

equations, one for each of the coordinates. This was solved by using a line relaxation procedure alternating with a column relaxation procedure, together with a multigrid cycle (Ref.9). A similar problem for the generation of body fitted coordinates for multiply-connected regions resulting from applications of estuaries with islands (Ref. 10).

The applications given in Ref. 5 to 10 all use the multigrid cycle based on the full approximation mode and the software described in the present report. A complete description of the problem and particularly the associated boundary conditions is given in the appropriate reports.


SET PARAMETERS FOR FINE GRID
SET STARTING SOLUTION
COMPUTE OPERATOR
USER SUPPLIED
SET PARAMETERS FOR COARSE GRID
COMPUTER OPERATOR
USER SUPPLIED
COMPUTE FORCING TERM
RELAX RESIDUAL EQUATION
USER SUPPLIED


COMPUTE CORRECTION ON CURRENT GRID
INTERPOLATE CORRECTION TO NEXT GRID
SET PARAMETERS FOR NEXT GRID
OBTAIN CURRENT FULL APPROXIMATION
COMPUTE OPERATOR
COMPUTE FORCING TERM
RELAX RESIDUAL EQUATION
NO
IS THE MULTIGRID CYCLE COMPLETE
YES
TEST RESIDUAL NORM
STOP

REFERENCES

1) A. BRANDT, "Multi-level Adaptive Technique, I. The Multigrid Method", IBM Research Report RC6026, 1976.

2) A. BRANDT, "Multi-level Adaptive Solutions to Boundary Value Problems", IBM Research Report RC6159, 1976.

3) J.C. SOUTH and A. BRANDT, "Application of a Multi-level Grid Method to Transsonic Flow Calculations", ICASE Report #76-8, March 1976.

4) S.P. GAUR and A. BRANDT, "Numerical Solution of Semiconductor Transport Equations in Two Dimensions by Multigrid Method", pages 327-329, Advances in computer methods for partial differential equations, 1977.

5) A. ROBERTS, "Multigrid Systems for the Rapid Inversion of Partial Differential Equations", Von Karman lecture series 1978-4, 1978.

6) R. CAMARERO, "Application of the Multigrid Method to Transsonic Flow in a Channel", Ecole Polytechnique, Report # , 1979.

7) R. CAMARERO and Y. BEGIN, "Calculation of non-lifting Transsonic Cascade Flow with Multigrid", Ecole Polytechnique, Report # 1979.

8) D. PELLETIER and R. CAMARERO, "Unsteady Viscous Flow Calculations Using the Multigrid Method", Ecole Polytechnique, Report # 1979.

9) M. YOUNIS and R. CAMARERO, "Generation of Body-fitted Coordinates for Cascade Computation Using Multigrid", Ecole Polytechnique, Report #

10) R. CAMARERO, "Application of Body-fitted Coordinates to Hydrodynamical Problems by the Multigrid Method", Ecole Polytechnique, Report #

A CONSULTER
SUR PLACE