JOURNAL OF COMPUTATIONAL PHYSICS 78, 1–14 (1988)

A Nonlinear Multigrid Algorithm and Boundary-Fitted Coordinates for the Solution of a Two-Dimensional Flow in a Branching Channel

G. LONSDALE,* J. S. BRAMLEY, AND D. M. SLOAN

Department of Mathematics, University of Strathclyde, Glasgow, G1 1XH, United Kingdom.

Received April 10, 1987; revised September 29, 1987

A recent paper by Bramley and Sloan (*Computers and Fluids* **15**, 297 (1987)) describes a numerical solution for two-dimensional flow of a viscous, incompressible fluid in a branching channel. A grid generation algorithm was used to map the solution region onto a rectangle and an upwind finite difference scheme was then used to solve the Navier–Stokes equations in terms of stream function and vorticity. Here we extend the earlier work by describing an efficient solution of the problem using a nonlinear multigrid algorithm. Of particular interest is the treatment of the boundary conditions in a manner which does not destroy the interior smoothness in the neighbourhood of the boundary.

1. INTRODUCTION

Bramley and Sloan [2] have described a numerical method, using boundary-fitted coordinates, for treating the steady, two-dimensional flow of a viscous, incompressible fluid in a branching channel. A typical configuration is given in Fig. 1. The fluid flows horizontally from the left through a uniform channel which has a normalised width of 2 units. The channel divides into two symmetrically placed channels, each of width d units, and the branching angle between these channels is 2θ. Symmetry is used to restrict the solution domain to the channel regions on and above the horizontal axis through the branching point, and this is the domain shown in Fig. 1. The slope of the channel wall is continuous on the upper boundary where the direction alters, and also at the branching point, O. The fluid vorticity is infinite at points where the boundary slope is discontinuous, and boundary-fitted coordinates have been used to remove such points. The method used to remove the sharp corners is described in detail in Bramley and Sloan [2].

One of the main motivations for numerical studies of fluid flow in bifurcating channels arises from the interest in blood flow through branching arteries. Investigations have shown that regions commonly associated with arterial disease

* Now at School of Mathematical Sciences, University of Bradford, Bradford, BD7 1DP, United Kingdom.

0021-9991/88 $3.00

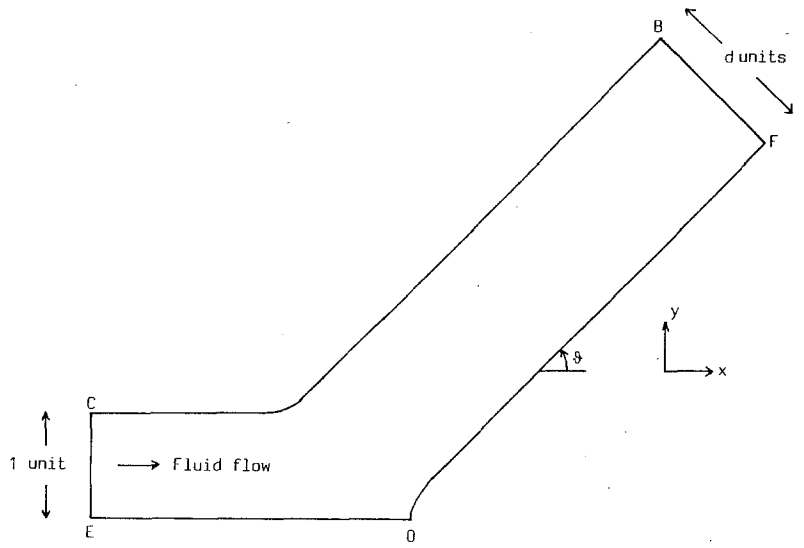


FIG. 1. Solution region in the (x, y) plane.

coincide with regions of reversed or stagnant flow. Low local velocity is one of the causes of blood coagulation within arteries. Many studies have therefore concentrated on fluid separation or recirculation and, in particular, on the influence of Reynolds number on these phenomena. Some of the early studies of flow in branching arteries are referred to in the paper by Bramley and Dennis [1]. Examples of some recent studies are cited in the paper by Perktold and Hilbert [8], which describes a finite element method for pulsatile flow through a model of the human carotid bifurcation.

The present study does not seek to gain further insight into flow phenomena. The objective is to describe an efficient, nonlinear, multigrid algorithm for solving the vorticity-stream function formulation of the Navier–Stokes equations applied to this type of channel flow. A recent paper by Lacroix *et al.* [5] describes work which had a similar objective. However, their choice of difference scheme gave rise to a relaxation process which failed to converge for Reynolds numbers greater than 200. Furthermore, the scheme which they adoped for boundary relaxation probably had a deleterious effect on the multigrid performance. Of particular interest in the present study is the relaxation process used for the vorticity boundary conditions. Extensive numerical experiments culminated in the adoption of suggestions made by Stüben and Linden [9] concerning a form of boundary relaxation which does not destroy the interior smoothness. The final choice of boundary relaxation has produced a method which is, to a great extent, both grid and Reynolds number independent. Moreover, discretisation by the upwind scheme of Dennis and Hudson [4] has produced a multigrid method which converges for Reynolds numbers as high as 1000.

2. Grid Generation and Governing Equations

2.1. *Grid Generation*

As described in Bramley and Sloan [2] the grid generation algorithm proposed by Thompson *et al.* [11] is used to transform the solution domain from the (x, y) plane to a rectangle in the (ξ, η) plane. The transformation and the boundary-fitted coordinates are obtained using the elliptic system

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= 0, \\ \eta_{xx} + \eta_{yy} &= 0, \end{aligned} \tag{2.1}$$

where subscripts indicate partial differentiation. Calculations are performed on the rectangular domain so dependent and independent variables are interchanged in (2.1) to give

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} &= 0, \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} &= 0, \end{aligned} \tag{2.2}$$

where

$$\alpha = x_\eta^2 + y_\eta^2, \qquad \beta = x_\xi x_\eta + y_\xi y_\eta, \qquad \gamma = x_\xi^2 + y_\xi^2. \tag{2.3}$$

Equations (2.2) are solved numerically on the rectangle $0 \leqslant \xi \leqslant Th$, $0 \leqslant \eta \leqslant Jh$ using second-order central differences on a square grid of size h. Here T and J are positive integers and, without loss of generality, we may set $h = 1$. To complete the system a discretisation of the boundary of Fig. 1 is used to supply Dirichlet boundary conditions for x and y on the rectangle $0 \leqslant \xi \leqslant Th$, $0 \leqslant \eta \leqslant Jh$. The boundary discretisation used to generate the typical grid shown in Fig. 2 is described in detail in Bramley and Sloan [2] and it need not be repeated here.

The central difference discretisation of (2.2) yields a system of nonlinear, algebraic equations which is solved by a multigrid algorithm. It is appropriate to postpone the description of the algorithm until we have presented the governing equations and boundary conditions for the fluid flow. These are presented briefly in the next section, and the reader is referred to Bramley and Dennis [1] and Bramley and Sloan [2] for additional details.

2.2. *Governing Equations and Boundary Conditions*

The governing equations in terms of the stream function, ψ, and the vorticity, ζ, are

$$\alpha\psi_{\xi\xi} - 2\beta\psi_{\xi\eta} + \gamma\psi_{\eta\eta} = -\Lambda^2\zeta, \tag{2.4}$$

$$\alpha\zeta_{\xi\xi} - 2\beta\zeta_{\xi\eta} + \gamma\zeta_{\eta\eta} = \Lambda\,\mathrm{Re}(\psi_\eta\zeta_\xi - \psi_\xi\zeta_\eta), \tag{2.5}$$

where α, β, and γ are defined by (2.3), Re is the Reynolds number based on the

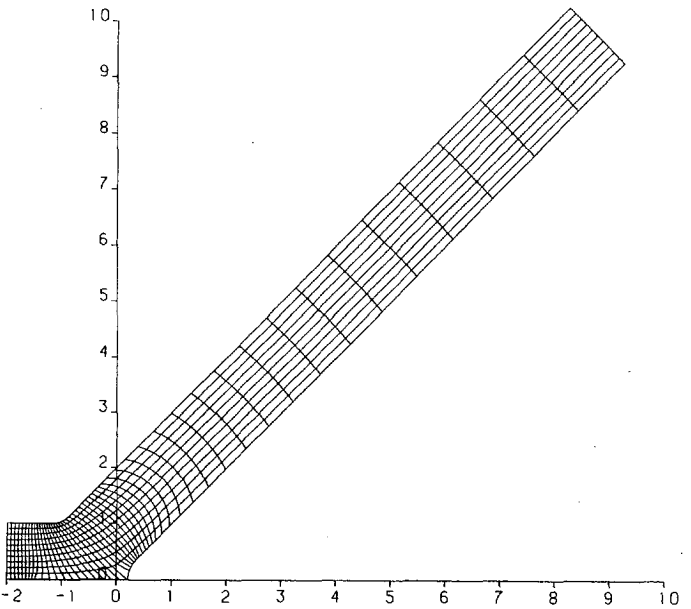


FIG. 2. A typical grid generated as described in Section 2.1.

upstream channel width and Λ is the Jacobian of the coordinate transformation, defined by

$$\Lambda = x_\xi y_\eta - x_\eta y_\xi. \tag{2.6}$$

The boundary conditions imposed on $\psi(\xi, \eta)$ and $\zeta(\xi, \eta)$ on the boundary of the rectangle $0 \leqslant \xi \leqslant Th$, $0 \leqslant \eta \leqslant Jh$ are readily described by referring to Fig. 1. In the following text a condition associated with a boundary segment such as CB will be the boundary condition on the image of CB under the coordinate transformation.

Downstream on FB the Poiseuille flow conditions are written as

$$\psi(Th, \eta) = \left(\frac{\eta}{Jh}\right)^2 \left[3 - 2\left(\frac{\eta}{Jh}\right)\right] \tag{2.7}$$

and

$$\zeta(Th, \eta) = \frac{6}{d^2}\left[2\left(\frac{\eta}{Jh}\right) - 1\right], \tag{2.8}$$

for $0 \leqslant \eta \leqslant Jh$. Poiseuille flow is also imposed upstream on EC, and this takes the form

$$\psi(0, \eta) = \left(\frac{\eta}{2Jh}\right)\left[3 - \left(\frac{\eta}{Jh}\right)^2\right] \tag{2.9}$$

and

$$\zeta(0, \eta) = 3\left(\frac{\eta}{Jh}\right), \tag{2.10}$$

for $0 \leqslant \eta \leqslant Jh$. This is a simpler form of upstream boundary conditions than that used by Bramley and Sloan [2].

On the boundaries $\eta = 0$, $\eta = Jh$ the stream function is given by

$$\psi(\xi, 0) = 0, \qquad \psi(\xi, Jh) = 1, \qquad 0 < \xi < Th, \tag{2.11}$$

and on the mid-stream line EO the vorticity condition is

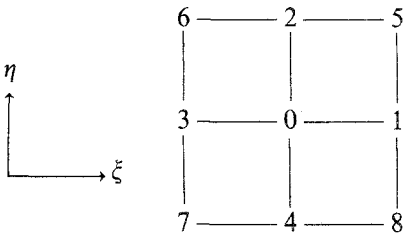
$$\zeta(\xi, 0) = 0 \qquad 0 < \xi \leqslant Nh, \tag{2.12}$$

where $\xi = Nh$ is the location of the image of O. Finally, at the lower and upper walls, OF and CB, the vorticity is given by a condition proposed by Woods [12]. On a square grid of size h the finite difference approximation which we use for vorticity at the lower boundary is

$$\zeta_o \Lambda_0^2[2 + 2h(\Lambda_\eta/\Lambda)_o - h(\gamma_\eta/\gamma)_o] = 6\gamma_o(\psi_o - \psi_1)/h^2 - \Lambda_o^2\zeta_1. \tag{2.13}$$

Here the subscript o indicates a typical boundary node at $(\xi, \eta) = (ih, 0)$, subscripts 2 and 3 are nodes at $(\xi, \eta) = ((i \pm 1)h, 0)$, respectively, and subscript 1 is at $(\xi, \eta) = (ih, h)$. Derivatives appearing in the coefficients are evaluated using second-order central differences in the ξ-direction and second-order, one-sided differences in the η-direction. The analogous condition on the upper boundary is obtained by negating the terms $(\Lambda_\eta/\Lambda)_o$ and $(\gamma_\eta/\gamma)_o$: subscript 1 now refers to a node at $(\xi, \eta) = (ih, (J-1)h)$.

Approximations to ψ and ζ are obtained using finite difference representations of (2.4) and (2.5), together with the Dirichlet boundary conditions (2.7)–(2.12) and the Neumann boundary condition (2.13). The interior discretisation is based on the upwind scheme proposed by Dennis and Hudson [4] and it is conveniently represented by a single subscript notation defined by the finite difference molecule



The discretisations of Eqs. (2.4) and (2.5) at node 0 are, respectively,

$$\alpha_0(\psi_1 + \psi_3)/h^2 + \gamma_o(\psi_2 + \psi_4)/h^2 - \tfrac{1}{2}\beta_o(\psi_5 - \psi_6 + \psi_7 - \psi_8)/h^2 + (\Lambda^2\zeta)_o$$
$$= 2(\alpha_o + \gamma_o)\,\psi_o/h^2 \tag{2.14}$$

$$\left(\frac{\alpha_o}{h^2}-\frac{K}{2h}+\frac{K^2}{4\alpha_o}\right)\zeta_1+\left(\frac{\alpha_o}{h^2}+\frac{K}{2h}+\frac{K^2}{4\alpha_o}\right)\zeta_3+\left(\frac{\gamma_0}{2h}-\frac{L}{2h}+\frac{L^2}{4\gamma_o}\right)\zeta_2$$
$$+\left(\frac{\gamma_o}{h^2}+\frac{L}{2h}+\frac{L^2}{4\gamma_o}\right)\zeta_4-\frac{1}{2}\beta_o(\zeta_5-\zeta_6+\zeta_7-\zeta_8)/h^2$$
$$=2\left(\frac{\alpha_o+\gamma_o}{h^2}+\frac{K^2}{4\alpha_o}+\frac{L^2}{4\gamma_o}\right)\zeta_0, \tag{2.15}$$

where $K=\operatorname{Re}\Lambda_0(\psi_2-\psi_4)/2h$, $L=\operatorname{Re}\Lambda_0(\psi_3-\psi_1)/2h$ and the subscripts indicate the nodes. Derivatives appearing in the coefficients are evaluated using second-order central differences. Bramley and Sloan [2] have commented on the artificial viscosity associated with the difference scheme (2.15). Our current interest is primarily directed at the formation of a suitable nonlinear multigrid algorithm so we shall not consider possible inaccuracies due to artificial viscosity.

3. Nonlinear Multigrid Algorithm

3.1. *General Multigrid Scheme*

Here we give a brief description of the nonlinear multigrid solution of a discrete problem such as that outlined in the preceding section. There are two possible approaches to the solution of such a nonlinear system: a global linearisation method combined with the use of a linear multigrid algorithm at each stage, or a full approximation scheme (FAS) multigrid algorithm applied directly to the nonlinear system. We adopt the FAS approach and henceforth refer to it as a nonlinear multigrid algorithm. Multigrid ideas and comparisons of the two approaches may be found in the papers by Brandt [3] and Stüben and Trottenberg [10].

Efficient multigrid algorithms are produced by properly constructed interactions between the smoothing properties of a relaxation method, such as the Gauss–Seidel method and coarse grid correction. Let the nonlinear system arising from the discretisation be denoted by

$$N_h(x_h)=f_h \qquad (N_h: G(R_h)\to G(R_h)), \tag{3.1}$$

where R_h is the fine grid and $G(R_h))$ is the space of grid functions on R_h. We assume that N_h has an inverse.

If x_h^j is the current approximation to x_h the algebraic defect, or residual, at this stage is defined by

$$d_h^j=f_h-N_h(x_h^j). \tag{3.2}$$

Local Fourier analysis may be used to show that relaxation methods are efficient at reducing the amplitude of high frequency components of the error, and thus of the algebraic defect. The ultimate slow convergence is due to the low frequency

components corresponding to the largest eigenvalues of the iteration operator. Relaxation methods may therefore be regarded as efficient smoothers. The principle of coarse grid correction may be seen if we rewrite (3.1) in terms of Eq. (3.2) as

$$N_h(x_h^j + y_h^j) - N_h(x_h^j) = d_h^j, \tag{3.3}$$

where the solution is given by $x_h = x_h^j + y_h^j$. If the high frequency components of the defect are negligible relative to the low frequency components we can represent Eq. (3.3) on a coarser grid, R_H. In this context a high frequency component is one which cannot be represented on the coarse grid. The current approximation, x_h^j, and the defect, d_h^j, are transferred to R_H by means of restriction operators to give grid functions x_H^j and d_H^j, respectively, in the space $G(R_H)$. On R_H (3.3) takes the form

$$N_H(z_H^j) - N_H(x_H^j) = d_H^j, \tag{3.4}$$

where $z_H^j = x_H^j + \hat{y}_H^j$, the operator N_H^{-1} is assumed to exist and the dimension of $G(R_H)$ is much smaller than the dimension of $G(R_h)$. If (3.4) is solved for z_H^j we find $\hat{y}_H^j$ and interpolate to the finer grid to obtain the approximation $\hat{y}_h^j$ to y_h^j. The new approximation to x_h is then given by

$$x_h^{j+1} = x_h^j + \hat{y}_h^j. \tag{3.5}$$

In order to represent the defect equation (3.3) on the coarser grid the defect must first be smoothed, and this is effected using a well-chosen relaxation process. This relaxation process involves a local linearisation, and this is the only linearisation which occurs in the multigrid algorithm.

The multigrid method extends the idea of coarse grid correction, together with the use of a smoother, to a series of sequentially coarser grids. The complete multigrid method combines the use of a relaxation method on each grid with correction on coarser grids. Good convergence properties can only be obtained if a suitable combination of these processes is used.

3.2. *Algorithms Used*

The scheme used for the solution of the grid generation equations described in Section 2.1 and the vorticity-stream function equations described in Section 2.2 uses square grids on the (ξ, η) plane. Given any grid, the next coarser grid is obtained by doubling the mesh size, and corrections are transferred from the coarser to the finer grid using bilinear interpolation. The restriction operator from fine to coarse grid is the full-weighting operator which has been described by Stüben and Trottenberg [10]. The coarse grid operator is similar to the fine grid operator, apart from the change in grid size. A fixed type, cycling, FAS algorithm was used, characterised by the following four parameters:

ν_b, the number of relaxation sweeps before coarse grid correction;

ν_a, the number of relaxation sweeps after coarse grid correction;

ν_c, the number of relaxation sweeps on the coarsest grid;

γ, the number of iterations of the multigrid algorithm for the coarse grid ($\gamma = 1$ gives V-cycles and $\gamma = 2$ gives W-cycles).

Details of the various algorithms may be found in the paper by Stüben and Trottenberg [10].

While all the above components are quite standard, care must be taken over the choice of a relaxation scheme. To solve the grid generation equations in Section 2.1 a line Gauss–Seidel scheme is used, solving simultaneously at points on the lines $\xi = \text{constant}$. A local Fourier analysis suggested that this would be a suitable smoother, and its value in producing a fast multigrid solution was supported by numerical experiments. Neither point Gauss–Seidel nor line Gauss–Seidel on lines $\eta = \text{constant}$ will give satisfactory smoothing. It is to be noted that the Dirichlet boundary conditions do not affect the interior smoothing rate of the relaxation method.

For the solution of the fluid flow equations in Section 2.2 the relaxation scheme must include interior and boundary schemes, due to the Neumann boundary conditions for the vorticity on the walls. The aim in using a boundary relaxation scheme is to smooth the defect of the boundary equations so that the boundary condition can be represented on the coarse grid in the same way as the interior equations. However, care must be taken to ensure that the relaxation of the boundary equations does not destroy interior smoothness at the nodes adjacent to the boundary.

The most obvious way of organising the relaxation is to relax all the interior equations with the boundary values held fixed and then to relax the boundary equations by updating the boundary values. A good interior smoothing rate can be achieved using an alternating line Gauss–Seidel (ALGS) iteration, but, as we shall see in the next section, the effect of the boundary relaxation is to force underrelaxation of the interior equations, with a resulting convergence rate which deteriorates seriously as Reynolds number increases. The boundary relaxation destroys the smoothness at the adjacent interior nodes, even when the solution of the boundary equations is included in the ALGS iteration.

The paper by Stüben and Linden [9] suggests a way around this difficulty by treating those interior points connected with the boundary condition separately from the remaining interior points. Here, this separate treatment is required for relaxation of the vorticity equation at nodes adjacent to the walls OF and CB in Fig. 1. Thus, the ALGS algorithm is used to relax the vorticity equation at all interior nodes, apart from those adjacent to the walls, and the boundary relaxation then deals with the equations at the boundary and adjacent points simultaneously. In fact, the equations at the boundary and adjacent points can be solved directly at

the cost of just one tridiagonal matrix inversion, since the system of equations may be written in the form

$$\begin{bmatrix} T_1 & T_2 \\ D_1 & D_2 \end{bmatrix} \zeta = \mathbf{f},$$

where T_1, T_2 are tridiagonal and D_1, D_2 are diagonal matrices.

One sweep of the relaxation method for equations (2.14), (2.15), together with the Woods boundary condition (2.13) is:

One ALGS sweep with the vorticity equation (2.15) over all interior nodes, apart from those adjacent to the walls;

One ALGS sweep with the stream function equation (2.14) over all interior nodes;

A direct solution of the vorticity equations (2.13) and (2.15) at all nodes on the walls and adjacent to the walls.

4. Numerical Results

In this section we present numerical results for the solution of the Navier–Stokes equations of Section 2.2. Results showing the efficient multigrid solution of the grid generation equations are presented in Lonsdale [7] and need not be repeated here; we simply note that a convergence rate similar to that for the solution of Poisson's equation can be achieved provided that the correct relaxation process is used, as mentioned in Section 2.2.

In Section 3.2 two possibilities for the multigrid handling of the boundary conditions were discussed:

(i) handling only the boundary nodes separately;

(ii) solving for the boundary and adjacent nodes simultaneously, and separately from the remaining interior nodes.

Numerical results for both methods (i) and (ii) are presented, and it will be shown that method (ii) should be the approach adopted for the problem presented in Section 2.2. The results for method (ii) are then compared with those obtained using a single grid to emphasise the gain in efficiency achieved by using a multigrid algorithm.

In presenting results, several parameters remain fixed throughout. The specific case of the branching channel geometry was that given by an angle $\theta = 45°$ with channel width $d = \sqrt{2}$ and upstream and downstream channel lengths of 3 units and 20 units, respectively (see Section 1). Three grid sizes were used as the finest grid in numerical experiments: 257×33, 129×33, and 129×17, where $N_1 \times N_2$ represents N_1 lines $\xi = \text{constant}$ and N_2 lines $\eta = \text{constant}$. Three Reynolds numbers were taken to investigate the behaviour of the algorithms with changing Reynolds

TABLE I

Performance of Relaxation Method (i).
Number of Multigrid Iterations to Convergence

	Re (Reynolds number)		
Grid	50	250	1000
129×17	8	11	14
129×33	7	10	13
257×33	8	13	14

number, these being $\mathrm{Re} = 50$, 250, and 1000. For the case $\mathrm{Re} = 50$ the initial solution used was Poiseuille flow throughout the region and for the higher Reynolds numbers the initial solution was taken to be the converged solution at the next lower Reynolds number.

In the multigrid process the cycling parameters given in Section 3.2 were held fixed at values which were found to give "optimal" convergence rates (in the sense of minimising overall CPU times to convergence in a series of numerical experiments). These parameter values were

$$\nu_b = 1, \qquad \nu_a = 1, \qquad \nu_c = 2, \qquad \gamma = 2.$$

Convergence was considered to have been achieved when the root-mean-square (rms) values of changes made to both ζ, ψ were less than a prescribed tolerance, which for these tests was set at 1.0×10^{-4}. The choice of W-cycles ($\gamma = 2$) provides a more robust algorithm: use of the less expensive V-cycles ($\gamma = 1$) can in certain cases result in the need for further underrelaxation parameters. For the related problem of the solution of the biharmonic equation, Linden [6] found that V-cycles could lead to divergence.

Numerous variations of boundary relaxation were attempted with method (i), including distributing changes to the adjacent nodes and solving for the boundary nodes within the ALGS iteration. However, the variations in boundary relaxation

TABLE II

Performance of Relaxation Method (ii).
Number of Multigrid Iterations to Convergence

	Re (Reynolds Number)		
Grid	50	250	1000
129×17	6	6	8
129×33	6	7	7
257×33	6	7	9

made no significant difference to the performance of the method. For all versions of method (i), to obtain a convergent process it was necessary to use underrelaxation of both the boundary and the interior equations. Underrelaxation values of 0.5 were used for all the equations: significantly larger values led to divergence or much slower convergence. Table I gives the performance of method (i) for all three grids and Reynolds numbers, in terms of multigrid iterations to convergence.

One important point to note from Table I is that for any given Reynolds number the method gives approximately grid-independent convergence, which may lead one to presume that the optimum multigrid convergence had been achieved. However, this can be seen not to be the case. By implementing the algorithm using relaxation method (ii) an improvement on the performance of method (i) can be achieved.

With an underrelaxation parameter of 0.5 used for the boundary relaxation, no

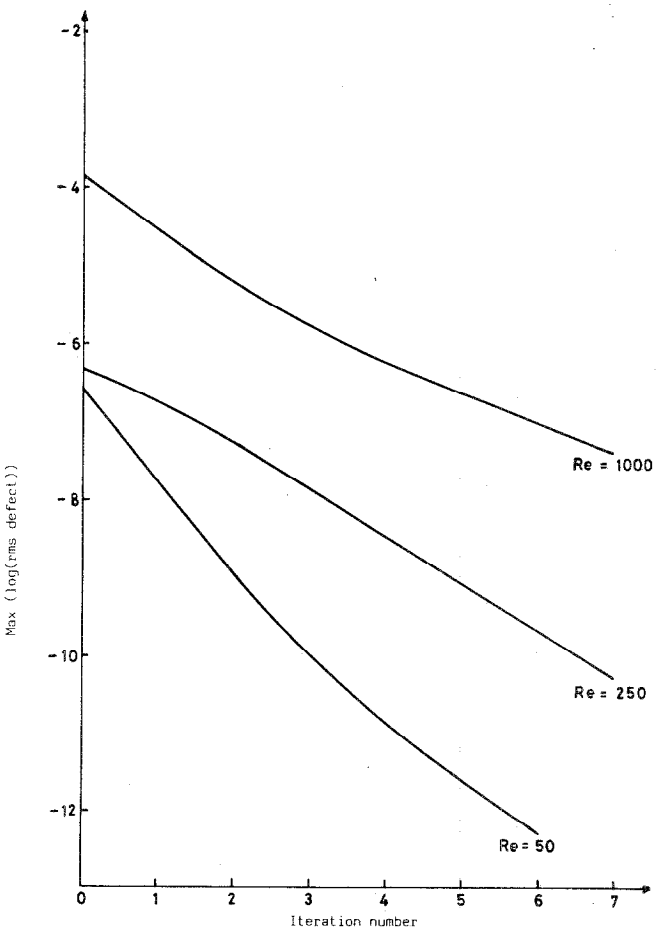


FIG. 3. Max(log(rms defect)) against iteration number for the 129×33 grid at $Re = 50, 250, 1000$.

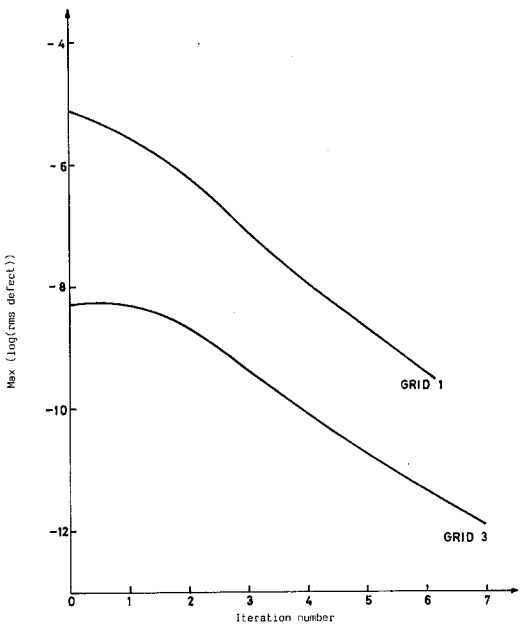


FIG. 4. Max(log(rms defect)) against iteration number for grid 1 (129×17) and grid 3 (257×33) at $Re = 250$.

underrelaxation was required for the interior equations with any combination of grid and Reynolds number. One peculiarity at $Re = 50$ was that a slight improvement in performance could be achieved by using an underrelaxation parameter of 0.75 for both the interior and boundary equations. Table II gives the best performances of the algorithm using relaxation method (ii), in terms of multigrid iterations to convergence.

Comparison between Tables I and II shows that the convergence rates for method (ii) are far less affected by increasing Reynolds number than those for method (i). Figure 3 shows the convergence of method (ii) on the 129×33 grid for each of the three Reynolds numbers. The maximum of the logs of the rms values of

TABLE III

Approximate CPU Seconds to Convergence for The Nonlinear Multigrid and One-Grid Methods at $Re = 50$

Grid	Nonlinear Multigrid	One-grid
129×17	140	440
129×33	270	1630
257×33	560	5400

the defects is plotted against the iteration number. Again we note that, unlike the typical behaviour of a one-grid algorithm, the convergence rate of the multigrid algorithm does not deteriorate as the grid is refined. The h-independent convergence is shown in Fig. 4, where max(log(rms defect)) is plotted against iteration number for the 129×17 and 257×33 grids at $\mathrm{Re} = 250$. (Note that the different aspect ratios of the 129×33 grid give a slightly different convergence rate and the results are therefore not included in Fig. 4). This can be further illustrated by comparing the multigrid convergence times with those of the relaxation scheme used as a one-grid method. Numerical tests showed that the handling of the boundary equations had little influence on the effectiveness of the relaxation scheme when used as a solution technique on a single grid, relaxation method (ii) being the fastest of the methods tested. Table III gives the CPU times to convergence for the nonlinear-multigrid algorithm, using relaxation method (ii), and the corresponding one-grid method, at Reynolds number 50 (all CPU times given are for the Vax 11/782 computer).

From Table III it can be seen that the nonlinear multigrid algorithm gives a reduction in CPU seconds of approximately 90% on the finest grid.

5. Conclusions

A nonlinear multigrid algorithm has been described for solving the Navier–Stokes equations governing steady flow in a branching channel. A stream function and vorticity formulation was used with a numerically generated curvilinear coordinate system. Numerical experiments were performed using different methods of incorporating the derivative boundary condition for vorticity into the multigrid algorithm. The experiments have shown that when only the rigid boundary nodes are treated separately from the interior nodes during vorticity relaxation it is necessary to use underrelaxation at the interior nodes. Furthermore, the convergence rate diminishes substantially as the Reynolds number is increased. If the rigid boundary nodes and their nearest interior neighbours are treated separately a more robust algorithm is produced. No interior underrelaxation is required and the resulting algorithm is almost grid and Reynolds number independent. Results show that this multigrid algorithm gives a large reduction on computing time relative to the relaxation scheme used as a one-grid method. The value of this work is in the demonstration that proper treatment of the vorticity boundary condition is readily achieved in a nonlinear multigrid algorithm for the Navier–Stokes equations in boundary-fitted coordinates.

Acknowledgment

The work of one of the authors (G. L.) has been supported by the Science and Engineering Research Council under Research Grant GR/D/23138.

REFERENCES

1. J. S. BRAMLEY AND S. C. R. DENNIS, *Comput. Fluids* **12**, 339 (1984).
2. J. S. BRAMLEY AND D. M. SLOAN, *Comput. Fluids* **15**, 297 (1987).
3. A. BRANDT, Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, GMD-Studien Nr. 85, Bonn, 1984 (unpublished).
4. S. C. R. DENNIS AND J. D. HUDSON, in *Proceedings, 1st Conf. Num. Meth. Laminar and Turbulent Flow* (Pentech, London, 1978).
5. M. LACROIX, R. CAMERERO, AND A. TAPUCU, *Num. Heat Transfer* **7**, 375 (1984).
6. J. LINDEN, Dissertation, Universität Bonn, 1985; GMD-Bericht Nr. 164 (Oldenbourg, Munich, 1986).
7. G. LONSDALE, Proceedings, Numerical Grid Generation Conference, Landschut, W. Germany, 1986.
8. K. PERKTOLD AND D. HILBERT, *J. Biomed. Eng.* **8**, 193 (1986).
9. K. STÜBEN AND J. LINDEN, Proceedings, Numerical Grid Generation Conference, Landschut, W. Germany, 1986.
10. K. STÜBEN AND U. TROTTENBERG, "Multigrid Methods," in Lecture Notes in Mathematics Vol. 960, edited by W. Hackbush and U. Trottenberg (Springer-Verlag, New York/Berlin, 1982), p. 1.
11. J. F. THOMPSON, F. C. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **24**, 274 (1977).
12. L. C. WOODS, *Aeronaut Q.* **5**, 176 (1954).