COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 55 (1986) 321–338
NORTH-HOLLAND

A CALCULATION PROCEDURE FOR THREE-DIMENSIONAL STEADY RECIRCULATING FLOWS USING MULTIGRID METHODS*

S.P. VANKA

Components Technology Division, Argonne National Laboratory, Argonne, IL 60439, U.S.A.

Received 25 March 1985
Revised manuscript received 15 July 1985

An efficient finite difference calculation procedure for three-dimensional recirculating flows is presented. The algorithm is based on a coupled solution of the three-dimensional momentum and continuity equations in primitive variables by the multigrid technique. A symmetrical coupled Gauss–Seidel technique is used for iterations and is observed to provide good rates of smoothing. Calculations have been made of the fluid motion in a three-dimensional cubic cavity with a moving top wall. The efficiency of the method is demonstrated by performing calculations at different Reynolds numbers with finite difference grids as large as 66 × 66 × 66 nodes. The CPU times and storage requirements for these calculations are observed to be very modest. The algorithm has the potential to be the basis for an efficient general-purpose calculation procedure for practical fluid flows.

Nomenclature

a	element in matrix at a node,
A	coefficient in the finite difference equation,
b	residual vector in matrix at a node,
C	convection mass flux,
D	diffusion part in the finite difference coefficient,
F	right-hand side term in the finite difference equations,
I	restriction/prolongation operator,
L	finite difference operator,
i^c, j^c, k^c	indices in x-, y-, z-directions for the coarse grid,
i^f, j^f, k^f	indices in x-, y-, z-directions for the fine grid,
p	pressure,
q	approximation to solution vector Q,
Q	solution vector,
R	residual in the finite difference equation,
Re	Reynolds number,
u, v, w	velocity components in x-, y-, and z-directions,
x, y, z	coordinate directions,
X	column vector of corrections for the node matrix,
ρ	density,
ν	kinematic viscosity,
ε	tolerance level on residuals,
δ	tolerance-adjustment factor (=0.2).

Subscripts

C	central value,
E	east $(i+1)$ value,

*This work was supported by the Ramjet Technology Division, Wright Patterson Air Force Base under interagency agreement with Argonne National Laboratory.

0045-7825/86/$3.50 © 1986, Elsevier Science Publishers B.V. (North-Holland)

H	high $(k+1)$ value,
i, j, k	node numbers in x-, y-, and z-directions,
L	lower $(k-1)$ value,
N	north $(j+1)$ value,
S	south $(j-1)$ value,
W	west $(i-1)$ value,
x, y, z	values appropriate for x-, y-, and z-directions,
new	new value,
old	old value.

Superscripts

C	continuity equation,
c	coarse-grid value,
f	fine-grid value,
u, v, w	values for u-, v-, and w-momentum equations.

1. Introduction

In many flow geometries of practical interest (gas turbine and ramjet combustors, coal and oil fired furnaces, heat exchangers, turbo-machinery passages, etc.), the flow and transport processes are three-dimensional. There exist three velocity components and the flow variables vary spatially in all three directions. In order to predict such flows, it is necessary to solve the complete three-dimensional form of the Navier–Stokes equations. Further, because many flows contain substantial flow recirculation, it is necessary to solve the fully elliptic equations that include convection and diffusion terms in three directions.

The task of solving such flows is large and complex. In order to get adequate resolution of the flow gradients, it is necessary to employ large numbers of finite difference (or finite element) nodes in each spatial direction. Typically, 100 nodes may be required in each direction in order to be confident of good numerical accuracy (the number obviously will vary with the accuracy of the discretization procedure). In addition, the governing equations for fluid flow are nonlinear and strongly coupled with each other. Of particular importance is the coupling between the momentum and continuity equations through the common pressure field. Despite much research in computational fluid dynamics in the past decade, current three-dimensional calculations of practical flows are stretching the available computer capacities from both CPU time and storage viewpoints [1]. There exists much need to develop faster algorithms that not only converge faster but can take into consideration the emerging computer architectures such as pipelining and paralleling.

This paper describes an efficient solution algorithm for three-dimensional fluid flows formulated in primitive variables (velocities and pressure). The algorithm differs from many others (such as of Williams [2], Chorin [3], Patankar and Spalding [4], Harlow and Welch [5], etc.) by employing a coupled solution of the momentum and continuity equations. The coupling of the momentum and continuity equations leads to a simultaneous update of the pressures and velocities and has been earlier observed to provide rapid convergence [6–9]. In the coupled approach, the continuity equation is used in its primitive form without converting it into a pressure or pressure-correction equation. Thus the relation between the pressure and velocities is implicitly retained. In this work, the coupled equations are solved iteratively by the multigrid technique, using a symmetrical coupled Gauss–Seidel (SCGS) technique as the relaxation procedure. Currently the technique has been applied to the calculation of the three-dimensional viscous flow in a cubic cavity caused by the movement of the top wall. Calculations have been made at Reynolds numbers of 100, 400, 1000, and 3200 and with finite difference grids consisting of as many as 66 × 66 × 66 nodes. Rapid convergence has been observed in all calculations. The CPU times for these large calculations are modest and easily affordable. Further, the algorithm is well suited for vectorization and paralleling which can significantly reduce present CPU times. The details of the method are given below.

2. The equation set and finite differencing

For a three-dimensional fully elliptic flow in Cartesian coordinates, the governing equations are

$$\frac{\partial}{\partial x}(uu)+\frac{\partial}{\partial y}(vu)+\frac{\partial}{\partial z}(wu)=-\frac{\partial p}{\rho\partial x}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right), \tag{1}$$

$$\frac{\partial}{\partial x}(vu)+\frac{\partial}{\partial y}(vv)+\frac{\partial}{\partial z}(wv)=-\frac{\partial p}{\rho\partial y}+\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right), \tag{2}$$

$$\frac{\partial}{\partial x}(wu)+\frac{\partial}{\partial y}(wv)+\frac{\partial}{\partial z}(ww)=-\frac{\partial p}{\rho\partial z}+\nu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right), \tag{3}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0\,. \tag{4}$$

The symbols u, v, and w represent the three components of the velocity vector along the x-, y-, and z-directions, respectively; ρ and ν represent the fluid density and kinematic viscosity, respectively; p is the pressure.

The above equations are discretized by a hybrid finite differencing scheme [10], which initially employs second-order central differencing on both convection and diffusion terms, but automatically modifies the convective differencing procedure to an upwind formulation when the local cell Reynolds number exceeds a value of two. The hybrid scheme has the merits of stability at high (≥2) cell Reynolds numbers (albeit of first-order accuracy) and second-order accuracy at low cell Reynolds numbers. A staggered mesh system (Fig. 1) is used, consistent with other works of the primitive variable formulation. Thus velocities are stored on the cell faces and the pressures are stored at the cell centers. The resulting finite difference equations at node (i, j, k) (for uniform grid dimensions of δx, δy, and δz) can be written as follows;

$$\begin{aligned}A_C^u u_{i+1/2,j,k} = {}& A_N^u u_{i+1/2,j+1,k}+A_S^u u_{i+1/2,j-1,k}+A_E^u u_{i+3/2,j,k}+A_W^u u_{i-1/2,j,k}\\ &+A_H^u u_{i+1/2,j,k+1}+A_L^u u_{i+1/2,j,k-1}+(p_{i,j,k}-p_{i+1,j,k})/\rho\delta x\,,\end{aligned} \tag{5}$$

$$\begin{aligned}A_C^v v_{i,j+1/2,k} = {}& A_N^v v_{i,j+3/2,k}+A_S^v v_{i,j-1/2,k}+A_E^v v_{i+1,j+1/2,k}+A_W^v v_{i-1,j+1/2,k}\\ &+A_H^v v_{i,j+1/2,k+1}+A_L^v v_{i,j+1/2,k-1}+(p_{i,j,k}-p_{i,j+1,k})/\rho\delta y\,,\end{aligned} \tag{6}$$

$$\begin{aligned}A_C^w w_{i,j,k+1/2} = {}& A_N^w w_{i,j+1,k+1/2}+A_S^w w_{i,j-1,k+1/2}+A_E^w w_{i+1,j,k+1/2}+A_W^w w_{i-1,j,k+1/2}\\ &+A_H^w w_{i,j,k+3/2}+A_L^w w_{i,j,k-1/2}+(p_{i,j,k}-p_{i,j,k+1})/\rho\delta z\,,\end{aligned} \tag{7}$$

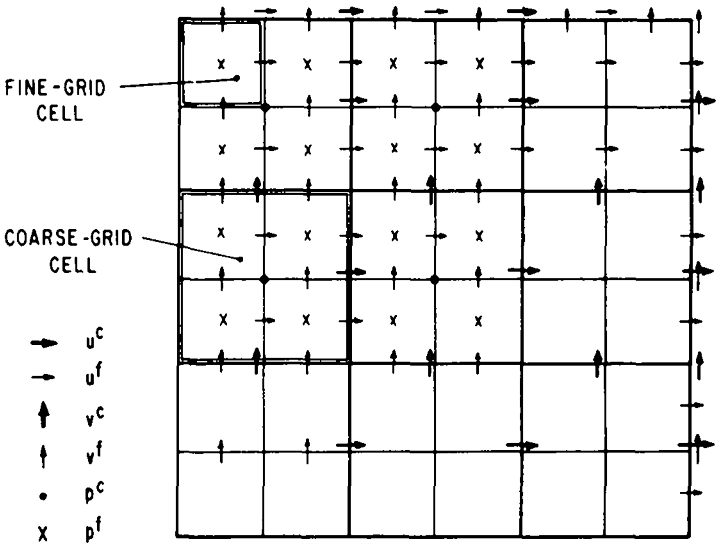


Fig. 1. Staggered mesh arrangement in x-y plane.

and

$$(u_{i+1/2,j,k} - u_{i-1/2,j,k})/\delta x + (v_{i,j+1/2,k} - v_{i,j-1/2,k})/\delta y + (w_{i,j,k+1/2} - w_{i,j,k-1/2})/\delta z = 0\,. \tag{8}$$

The superscripts relate to the variables and the subscript indices denote the location of the variables on the finite difference grid. The fractional indices refer to the staggered locations of the velocities. The coefficients A_C, A_N, A_S, A_E, A_W, etc. are given by the following expressions:

$$A_C^u = A_N^u + A_S^u + A_E^u + A_W^u + A_L^u + A_H^u\,, \tag{9}$$

$$A_C^v = A_N^v + A_S^v + A_E^v + A_W^v + A_L^v + A_H^v\,, \tag{10}$$

$$A_C^w = A_N^w + A_S^w + A_E^w + A_W^w + A_L^w + A_H^w\,, \tag{11}$$

$$A_W = \mathrm{AMAX}(|C_{x^-}|, D_{x^-}) + C_{x^-}\,, \tag{12}$$

$$A_E = \mathrm{AMAX}(|C_{x^+}|, D_{x^+}) - C_{x^+}\,, \tag{13}$$

$$A_S = \mathrm{AMAX}(|C_{y^-}|, D_{y^-}) + C_{y^-}\,, \tag{14}$$

$$A_N = \mathrm{AMAX}(|C_{y^+}|, D_{y^+}) - C_{y^+}\,, \tag{15}$$

$$A_L = \mathrm{AMAX}(|C_{z^-}|, D_{z^-}) + C_{z^-}\,, \tag{16}$$

and

$$A_H = \mathrm{AMAX}(|C_{z^+}|, D_{z^+}) - C_{z^+}\,, \tag{17}$$

where

$$C_{x^-} = 0.5\, u_{x^-}/\delta x\,, \qquad D_{x^-} = \nu/\delta x^2\,, \tag{18}$$

$$C_{x^+} = 0.5\, u_{x^+}/\delta x\,, \qquad D_{x^+} = \nu/\delta x^2\,, \tag{19}$$

$$C_{y^-} = 0.5\, v_{y^-}/\delta y\,, \qquad D_{y^-} = \nu/\delta y^2\,, \tag{20}$$

$$C_{y^+} = 0.5 v_{y^+}/\delta y\,, \qquad D_{y^+} = \nu/\delta y^2\,, \tag{21}$$

$$C_{z^-} = 0.5\, w_{z^-}/\delta z\,, \qquad D_{z^-} = \nu/\delta z^2\,, \tag{22}$$

$$C_{z^+} = 0.5\, w_{z^+}/\delta z\,, \qquad D_{z^+} = \nu/\delta z^2\,. \tag{23}$$

Here, u_{x^-}, u_{x^+}, v_{y^-}, v_{y^+}, w_{z^-}, and w_{z^+} denote velocities at the x^-, x^+, y^-, y^+, z^-, and z^+ sides of the cells surrounding the variable in question. δx, δy, and δz are dimensions of the finite difference cells.[1] The AMAX function ensures the central difference operator at cell $\mathrm{Re} \leqslant 2$ and changes the differencing to upwinding at cell $\mathrm{Re} \geqslant 2$. The hybrid differencing provides stability, but is diffusive because of the first-order accuracy at high local Reynolds numbers. Thus, in order to ensure accurate results fine enough grids must be employed. The finite difference grids used here ($66 \times 66 \times 66$ nodes) are quite fine, although to establish grid-independency a thorough comparison of the results of various grids must be made.

3. The solution algorithm

The present algorithm has two novel features. First, the momentum and continuity equations are solved simultaneously. No pressure or pressure equation is derived unlike most earlier methods of primitive variable type. Second, the coupled equations are solved iteratively by the powerful multigrid technique. This combination results in a highly efficient algorithm for three-dimensional fluid flows as will be demonstrated in Section 4. The algorithm is called BLIMM (for block-implicit multigrid method).

3.1. *The multigrid concept*

The concept of the multigrid technique may be explained as follows. Given a set of linear finite difference equations,

$$L^M Q^M = F^M\,, \tag{24}$$

for a general elliptic equation, any iterative procedure such as Gauss–Seidel, Jacobi, incomplete LU factorization, etc., is known to converge rapidly for the first few iterations and very slowly thereafter. A Fourier analysis of the error-reduction process shows that these conventional iterative procedures are most efficient in smoothing out the errors of wavelengths comparable to the mesh size, but are inefficient in annihilating low-frequency components. However, the low frequencies on any grid are relatively larger on grids that are coarser than

[1] Some minor modifications to the expressions for the diffusion terms are necessary at the near-boundary nodes.

the grid in question. The multigrid technique is based on the premise that each frequency range of error must be smoothed on the grid where it is most suitable to do so. Consequently, the multigrid technique cycles between coarser and finer grids until all the frequency components are appropriately smoothed.

In the simplest version of the multigrid technique, the solution is initiated on the finest grid M. A few iterations are performed on grid M until the relaxation procedure (iterations) fails to smooth the residuals at the desired theoretical rate. The iterations are stopped on grid M and the residuals ($L^M q^M - F^M = R^M$, q^M is an approximation to Q^M) are transferred to the next coarser grid, obtained typically by doubling the mesh size. The flow variables are similarly restricted to the coarse grid. A correction function q^{M-1} is calculated by solving the system

$$L^{M-1} q^{M-1} = I_M^{M-1} R^M , \tag{25}$$

where L^{M-1} is the operator on grid $(M-1)$ and I_M^{M-1} is the operator to restrict (interpolate) the residuals of grid M. The solution to (25) is obtained on grid $(M-1)$ accurately if $(M-1)$ is not a large grid. Otherwise, a few iterations on grid $(M-1)$ are performed after which R^{M-1} is restricted to grid $(M-2)$. When an accurate solution to (25) is obtained, q^{M-1} is prolongated to grid M, i.e.,

$$q_{\text{new}}^M = q_{\text{old}}^M + I_{M-1}^M q^{M-1} . \tag{26}$$

The restrictions and prolongations from and to grid M are continued until the residuals on grid M decrease below the desired tolerance.

A number of variants to the above outline of the multigrid method have been proposed, including extensions to nonlinear problems. For complete review of the MG technique reviews by Brandt [11] and by Stüben and Trottenberg [12] may be consulted.

3.2. The present algorithm

In the present work, we have employed the FAS-FMG (full approximation storage-full multigrid) algorithm originally developed by Brandt [11] and subsequently used by many investigators (e.g., Ghia et al. [13] and Vanka [9]). The FAS-FMG is a generalization of the correction scheme described above. However, unlike the correction scheme it proceeds from the coarsest to the finest grid. The flow chart of the FAS-FMG cycle, shown in Fig. 2, proceeds as follows. After a series of grids is chosen, iterations are initiated on the coarsest grid (grid number 1). On this grid, the solution of the complete nonlinear problem is sought. The coefficients in the finite difference equations are evaluated with the existing values of velocities in store and the linear equations are then solved iteratively. In the present algorithm a symmetrical coupled Gauss–Seidel technique (SCGS) is used for the iterations (relaxations). The SCGS scheme was used in an earlier study [9] and was observed to provide efficient smoothing rates. The SCGS scheme is basically a node-by-node solver; however, at each node it updates all *six* velocities (i.e., velocities located on all six faces of the three-dimensional cell) and the pressure. Thus, each velocity is updated twice during one SCGS sweep of the flow domain. The SCGS scheme solves the following equations at any given cell (i, j, k):

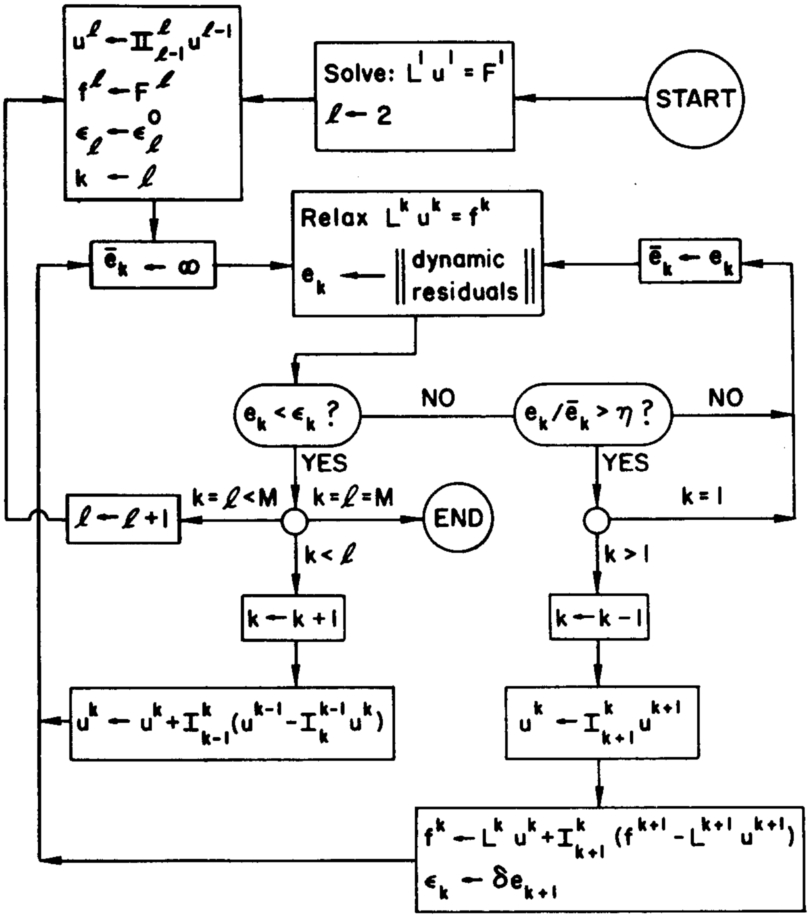

START
Solve: $L^1 u^1 = F^1$
$\ell \leftarrow 2$
$u^\ell \leftarrow \Pi^\ell_{\ell-1} u^{\ell-1}$
$f^\ell \leftarrow F^\ell$
$\epsilon_\ell \leftarrow \epsilon^0_\ell$
$k \leftarrow \ell$
$\bar{e}_k \leftarrow \infty$
Relax $L^k u^k = f^k$
$e_k \leftarrow \|$dynamic residuals$\|$
$\bar{e}_k \leftarrow e_k$
$e_k < \epsilon_k$?
NO
$e_k/\bar{e}_k > \eta$?
NO
YES
YES
$k = \ell < M$
$k = \ell = M$
END
$\ell \leftarrow \ell + 1$
$k = 1$
$k < \ell$
$k > 1$
$k \leftarrow k+1$
$k \leftarrow k-1$
$u^k \leftarrow u^k + I^k_{k-1}(u^{k-1} - I^{k-1}_k u^k)$
$u^k \leftarrow I^k_{k+1} u^{k+1}$
$f^k \leftarrow L^k u^k + I^k_{k+1}(f^{k+1} - L^{k+1} u^{k+1})$
$\epsilon_k \leftarrow \delta e_{k+1}$

Fig. 2. Flow chart of FAS-FMG solution cycle.

$$(A_C^u)_{i-1/2,j,k} u_{i-1/2,j,k} = F^u_{i-1/2,j,k} , \tag{27}$$

$$(A_C^u)_{i+1/2,j,k} u_{i+1/2,j,k} = F^u_{i+1/2,j,k} , \tag{28}$$

$$(A_C^v)_{i,j-1/2,k} v_{i,j-1/2,k} = F^v_{i,j-1/2,k} , \tag{29}$$

$$(A_C^v)_{i,j+1/2,k} v_{i,j+1/2,k} = F^v_{i,j+1/2,k} , \tag{30}$$

$$(A_C^w)_{i,j,k-1/2} w_{i,j,k-1/2} = F^w_{i,j,k-1/2} , \tag{31}$$

$$(A_C^w)_{i,j,k+1/2} w_{i,j,k+1/2} = F^w_{i,j,k+1/2} , \tag{32}$$

and

$$(u_{i+1/2,j,k} - u_{i-1/2,j,k})/\delta x + (v_{i,j+1/2,k} - v_{i,j-1/2,k})/\delta y + (w_{i,j,k+1/2} - w_{i,j,k-1/2})\delta z = 0 , \tag{33}$$

where

$$
\begin{aligned}
F^u_{i+1/2,i,k} = {} & (A^u_{\mathrm{N}})_{i+1/2,j,k}u_{i+1/2,j+1,k} + (A^u_{\mathrm{S}})_{i+1/2,j,k}u_{i+1/2,j-1,k} + (A^u_{\mathrm{E}})_{i+1/2,j,k}u_{i+3/2,j,k} \\
& + (A^u_{\mathrm{W}})_{i+1/2,j,k}u_{i-1/2,j,k} + (A^u_{\mathrm{H}})_{i+1/2,j,k}u_{i+1/2,j,k+1} + (A^u_{\mathrm{L}})_{i+1/2,j,k}u_{i+1/2,j,k-1} \\
& + (p_{i,j,k} - p_{i+1,j,k})/\rho\delta x \,,
\end{aligned}
\tag{34}
$$

etc.

The equations represent the complete local transport characteristics for node (i, j, k). In order to solve them, these equations are first written in terms of residuals and corrections at the node. A matrix of six diagonal elements and two bordered line elements is then constructed from the seven equations. Because of the bordered structure, the matrix can be inverted to give analytical expressions for the corrections (see Appendix A).

The SCGS scheme is applied repeatedly, updating the coefficients A_{N}, A_{S}, etc. within the iterative sweeps. Such sweeps are performed until the residuals decrease below the prescribed tolerance level. The tolerance used on the summed averaged residual is defined as

$$
R_{\mathrm{conv}} = \sum_{i,j,k} [\{(R^u_{i,j,k})^2 + (R^v_{i,j,k})^2 + (R^w_{i,j,k})^2 + (R^C_{i,j,k})^2\}/\mathrm{NEQ}]^{1/2} \,,
\tag{35}
$$

where the residuals are the differences between the left- and right-hand sides of the equations with appropriate normalizations. Also,

$$
\mathrm{NEQ} = (\mathrm{IMAX} * \mathrm{JMAX} * \mathrm{KMAX} * 4) \,.
\tag{36}
$$

When a converged solution on grid 1 is obtained, it is prolongated (extrapolated) to the next finest grid (grid 2). The solution on grid 2 is then sought. The coefficients are dynamically calculated inside the loop over the nodes and at each node equations such as (27)–(33) are assembled and solved. However, on this grid, the iterations are *not* carried on until complete convergence. Instead, the rate of decrease of R_{conv} is monitored and when this rate falls below a specified rate, i.e., when

$$
R_{p+1}/R_p > \eta
\tag{37}
$$

(where a subscript p refers to iteration count), the iterations on this grid are temporarily halted. The residuals and the solution are then restricted to the next coarser grid. The coefficients are formulated on the coarser grid with the restricted values of the variables and a solution to the coarse-grid equations is sought. The equations now solved on the coarser grid (say grid 1) are

$$
L^1q^1 = F^1 + I^1_2(F^2 - L^2q^2) \,,
\tag{38}
$$

where I^1_2 restricts the residual $(F^2 - L^2q^2)$ to grid 1 (the superscripts 1 and 2 here refer to the values on grid 1 and 2 and are not exponents). For the coarsest grid in the series of grids, (38) is solved exactly.

The solution on grid 1 is now used to correct the existing solution of grid 2. At this stage, the *solution* on grid 1 is not prolongated to grid 2, rather the difference between the initial and final solution on grid 1 is prolongated. Thus the correction to q^2 is

$$
q^2_{\mathrm{new}} = q^2_{\mathrm{old}} + I^2_1(q^1 - I^1_2q^2_{\mathrm{old}}) \,.
\tag{39}
$$

The restrictions and prolongations between grids 1 and 2 are continued until an accurate solution on grid 2 is obtained. At this stage the solution on grid 2 is prolongated to grid 3 and the residuals are smoothed until the rate of smoothing falls below the threshold value. Restrictions are then made to grid 2 from grid 3 and the residuals are iterated on grid 2 until the smoothing is satisfactory. Otherwise the remaining residuals of grid 2 are transferred to grid 1 and annihilated completely (i.e., to $\leqslant 10^{-4}$). As the solution sequence proceeds, finer and finer grids are considered and converged solutions are obtained on each grid. When a converged solution is obtained on the finest grid, M, the solution is terminated.

The tolerance level on R_{conv} on all grids except the coarsest is set to 10^{-3}. For the calculations of the flow in a cubic cavity this represents a reduction of the initial residuals by three orders of magnitude. The tolerance on grid 1 is set to a smaller value in order to obtain a nearly exact coarse-grid solution (or correction). However, the accuracy level on intermediate grids during the restriction from a finer grid is reset to a lower value. When an intermediate grid is reached by restricting the residuals from a finer grid, the solution on the intermediate grid is not obtained to 10^{-3} level of accuracy. Instead, the tolerance is reset to

$$\varepsilon_h = \delta(R_{\text{conv}})_{h+1}\,, \tag{40}$$

where $\delta = 0.2$.

3.3. Restriction and prolongation operators

The restriction and prolongation procedures are somewhat dictated by the staggered mesh arrangement. Restriction is used for transferring fine-grid values to a coarse grid, whereas prolongation is used for extrapolating a coarse-grid correction to a fine grid. The two operators were denoted earlier by I_h^{h-1} and I_{h-1}^h, respectively, where h denotes one of the grids. A frequent restriction operator is injection, i.e., the coarse-grid value is taken to be the local fine-grid value. Thus

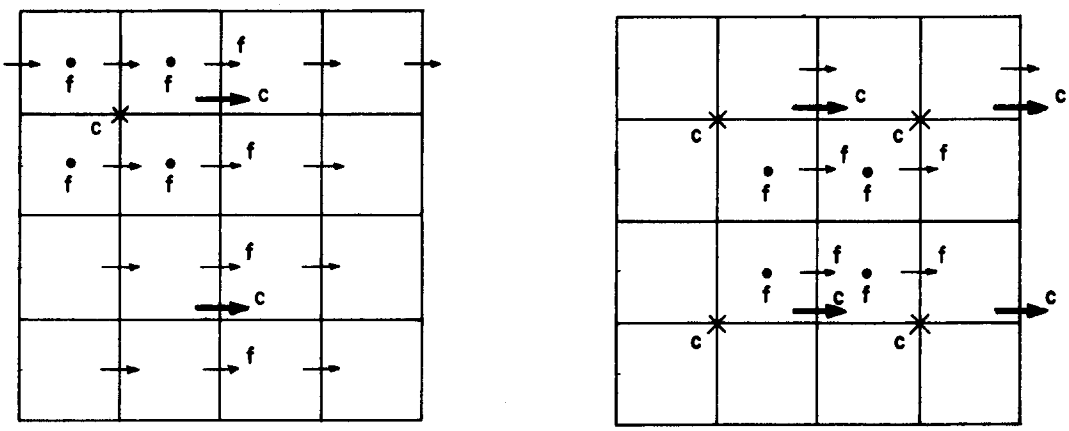
$$q^{\text{c}}_{i,j,k} = q^{\text{f}}_{2i-1,2j-1,2k-1}\,, \tag{41}$$

where the superscripts c, f denote coarse- and fine-grid values. The injection operator is not applicable for the staggered mesh because of the different locations of variables on coarse and fine meshes (see Fig. 1). In the current study, the restrictions are made by averaging neighbor values (see Fig. 3(a)). Let $(i^{\text{c}}, j^{\text{c}}, k^{\text{c}})$ and $(i^{\text{f}}, j^{\text{f}}, k^{\text{f}})$ denote coarse- and fine-mesh indices, respectively. Also, let $u_{i+1/2,j,k}$ be referred to as $u(i, j, k)$ and $u_{i-1/2,j,k}$ be referred to as $u(i-1, j, k)$, etc. Then

$$i^{\text{f}} = 2(i^{\text{c}}) - 1\,, \qquad j^{\text{f}} = 2(j^{\text{c}}) - 1\,, \qquad k^{\text{f}} = 2(k^{\text{c}}) - 1\,, \tag{42}$$

$$u^{\text{c}}(i^{\text{c}}, j^{\text{c}}, k^{\text{c}}) = \tfrac{1}{4}[u^{\text{f}}(i^{\text{f}}, j^{\text{f}}, k^{\text{f}}) + u^{\text{f}}(i^{\text{f}}, j^{\text{f}}-1, k^{\text{f}}) + u^{\text{f}}(i^{\text{f}}, j^{\text{f}}, k^{\text{f}}-1) + u^{\text{f}}(i^{\text{f}}, j^{\text{f}}-1, k^{\text{f}}-1)]\,, \tag{43}$$

$$v^{\text{c}}(i^{\text{c}}, j^{\text{c}}, k^{\text{c}}) = \tfrac{1}{4}[v^{\text{f}}(i^{\text{f}}, j^{\text{f}}, k^{\text{f}}) + v^{\text{f}}(i^{\text{f}}-1, j^{\text{f}}, k^{\text{f}}) + v^{\text{f}}(i^{\text{f}}, j^{\text{f}}, k^{\text{f}}-1) + v^{\text{f}}(i^{\text{f}}-1, j^{\text{f}}, k^{\text{f}}-1)]\,, \tag{44}$$



(a) RESTRICTION (b) PROLONGATION

Fig. 3. Sketch of restriction and prolongation processes: c = coarse grid, f = fine grid (two-dimensional plane).

$$w^c(i^c, j^c, k^c) = \tfrac{1}{4}[w^f(i^f, j^f, k^f) + w^f(i^f, j^f - 1, k^f) + w^f(i^f - 1, j^f, k^f) + w^f(i^f - 1, j^f - 1, k^f)]\,, \tag{45}$$

$$\begin{aligned} p^c(i^c, j^c, k^c) = \tfrac{1}{8}[&p^f(i^f, j^f, k^f) + p^f(i^f, j^f - 1, k^f) + p^f(i^f - 1, j^f, k^f) + p^f(i^f - 1, j^f - 1, k^f) \\ &+ p^f(i^f, j^f, k^f - 1) + p^f(i^f, j^f - 1, k^f - 1) \\ &+ p^f(i^f - 1, j^f, k^f - 1) + p^f(i^f - 1, j^f - 1, k^f - 1)]\,. \end{aligned} \tag{46}$$

The prolongation relations are derived by a trilinear prolongation. For each coarse-grid node, eight fine-grid values are calculated. For the u-velocity, they are

$$u^f(i^f, j^f, k^f) = \tfrac{1}{16}[9u_1^c + 3u_3^c + 3u_5^c + u_7^c]\,, \tag{47}$$

$$u^f(i^f, j^f + 1, k^f) = \tfrac{1}{16}[9u_3^c + 3u_1^c + 3u_7^c + u_5^c]\,, \tag{48}$$

$$u^f(i^f, j^f, k^f + 1) = \tfrac{1}{16}[9u_5^c + 3u_7^c + 3u_1^c + u_3^c]\,, \tag{49}$$

$$u^f(i^f, j^f + 1, k^f + 1) = \tfrac{1}{16}[9u_7^c + 3u_5^c + 3u_3^c + u_1^c]\,, \tag{50}$$

$$u^f(i^f + 1, j^f, k^f) = \tfrac{1}{32}[9u_1^c + 9u_2^c + 3u_3^c + 3u_4^c + 3u_5^c + 3u_6^c + u_7^c + u_8^c]\,, \tag{51}$$

$$u^f(i^f + 1, j^f + 1, k^f) = \tfrac{1}{32}[9u_3^c + 9u_4^c + 3u_1^c + 3u_2^c + 3u_7^c + 3u_8^c + u_5^c + u_6^c]\,, \tag{52}$$

$$u^f(i^f + 1, j^f, k^f + 1) = \tfrac{1}{32}[9u_5^c + 9u_6^c + 3u_7^c + 3u_8^c + 3u_1^c + 3u_2^c + u_3^c + u_4^c]\,, \tag{53}$$

$$u^f(i^f + 1, j^f + 1, k^f + 1) = \tfrac{1}{32}[9u_7^c + 9u_8^c + 3u_5^c + 3u_6^c + 3u_3^c + 3u_4^c + u_1^c + u_2^c]\,, \tag{54}$$

where

$$u_1^c = u^c(i^c, j^c, k^c)\,, \qquad u_2^c = u^c(i^c+1, j^c, k^c)\,, \tag{55}$$

$$u_3^c = u^c(i^c, j^c+1, k^c)\,, \qquad u_4^c = u^c(i^c+1, j^c+1, k^c)\,, \tag{56}$$

$$u_5^c = u^c(i^c, j^c, k^c+1)\,, \qquad u_6^c = u^c(i^c+1, j^c, k^c+1)\,, \tag{57}$$

$$u_7^c = u^c(i^c, j^c+1, k^c+1)\,, \qquad u_8^c = u^c(i^c+1, j^c+1, k^c+1)\,. \tag{58}$$

The v- and w-velocities can be prolongated by equivalent relations obtained by rotating the coordinates appropriately. The prolongations for the pressures have different weightings because of their cell-centered locations. For each coarse-grid cell eight fine-grid pressures are obtained. The relations are

$$p^f(i^f, j^f, k^f) = \tfrac{1}{64}[27p_1^c + 9p_2^c + 9p_3^c + 3p_4^c + 9p_5^c + 3p_6^c + 3p_7^c + p_8^c]\,, \tag{59}$$

$$p^f(i^f, j^f+1, k^f) = \tfrac{1}{64}[27p_3^c + 9p_4^c + 9p_1^c + 3p_2^c + 9p_7^c + 3p_8^c + 3p_5^c + p_6^c]\,, \tag{60}$$

$$p^f(i^f+1, j^f, k^f) = \tfrac{1}{64}[27p_2^c + 9p_1^c + 9p_4^c + 3p_3^c + 9p_6^c + 3p_5^c + 3p_8^c + p_7^c]\,, \tag{61}$$

$$p^f(i^f+1, j^f+1, k^f) = \tfrac{1}{64}[27p_4^c + 9p_3^c + 9p_2^c + 3p_1^c + 9p_8^c + 3p_7^c + 3p_6^c + p_5^c]\,, \tag{62}$$

$$p^f(i^f, j^f, k^f+1) = \tfrac{1}{64}[27p_5^c + 9p_6^c + 9p_7^c + 3p_8^c + 9p_1^c + 3p_2^c + 3p_3^c + p_4^c]\,, \tag{63}$$

$$p^f(i^f, j^f+1, k^f+1) = \tfrac{1}{64}[27p_7^c + 9p_8^c + 9p_5^c + 3p_6^c + 9p_3^c + 3p_4^c + 3p_1^c + p_2^c]\,, \tag{64}$$

$$p^f(i^f+1, j^f, k^f+1) = \tfrac{1}{64}[27p_6^c + 9p_5^c + 9p_8^c + 3p_7^c + 9p_2^c + 3p_1^c + 3p_4^c + p_3^c]\,, \tag{65}$$

$$p^f(i^f+1, j^f+1, k^f+1) = \tfrac{1}{64}[27p_8^c + 9p_7^c + 9p_6^c + 3p_5^c + 9p_4^c + 3p_3^c + 3p_2^c + p_1^c]\,, \tag{66}$$

where

$$p_1^c = p^c(i^c, j^c, k^c)\,, \qquad p_2^c = p^c(i^c+1, j^c, k^c)\,, \tag{67}$$

$$p_3^c = p^c(i^c, j^c+1, k^c)\,, \qquad p_4^c = p^c(i^c+1, j^c+1, k^c)\,, \tag{68}$$

$$p_5^c = p^c(i^c, j^c, k^c+1)\,, \qquad p_6^c = p^c(i^c+1, j^c, k^c+1)\,, \tag{69}$$

$$p_7^c = p^c(i^c, j^c+1, k^c+1)\,, \qquad p_8^c = p^c(i^c+1, j^c+1, k^c+1)\,. \tag{70}$$

The above relations are slightly modified near the boundaries to avoid using the boundary pressures. The modified relations are obtained by assuming a zero-derivative condition near the boundaries.

It is necessary to remember that in the FAS-FMG algorithm, the values on the coarse grid are not directly prolongated; instead, the *changes* from the previously restricted values are prolongated. That is

$$q_{\text{new}}^h = q_{\text{old}}^h + I_{h-1}^h \delta q^{h-1}\,, \tag{71}$$

and

$$\delta q^{h-1} = q^{h-1} - I_h^{h-1} q^h_{\text{old}} \,. \tag{72}$$

Relations (47)–(70) are used on δq^{h-1}.

4. Application to flow in a cubic cavity

The convergence characteristics of the algorithm are demonstrated by considering the solution of the three-dimensional viscous flow in a cubic cavity. This problem characterizes the nonlinear elliptic nature of many flows of practical interest so much that it has been frequently considered as a standard test case for innovative numerical algorithms. The counterpart of this problem in two dimensions, i.e., the flow in a driven square cavity has been studied in depth in several earlier works (e.g., [9, 13, 14, 15]). Fig. 4 shows the problem currently considered. A cube of dimension D is considered with the top wall moving at a velocity U. The flow Reynolds number is defined to be UD/ν. Fluid motion is set up by the shear of the top wall, resulting in a complex three-dimensional flow field consisting of three-dimensional vortex structures.

In the application of the present algorithm to the cubic-cavity problem, a number of flow Reynolds numbers and several finite difference grids have been considered. Calculations have been made for Reynolds numbers of 100, 400, 1000, and 3200 with grids consisting of $16 \times 16 \times 16$, $32 \times 32 \times 32$, and $64 \times 64 \times 64$ finite difference cells. For $\mathrm{Re} = 3200$ the first two grids were calculated to 10^{-3} accuracy. Because the estimated time for the third grid was about 80 minutes it was not continued to 10^{-3} level of residuals. The boundary conditions were of the Dirichlet type for velocities, and no boundary conditions were necessary for the pressure. The singularity of the system of equations (because of the incompressibility condition) was resolved by holding the pressure fixed at the corner node (2, 2, 2). The threshold smoothing rate, η, for switching from fine to coarse grids was fixed at 0.5. All calculations were initiated from zero velocity and pressure fields without using information from previous solutions at lower Reynolds numbers. The convergence criterion was based on the value of R_{conv} to be less than 10^{-3} on the finest grid. The nodes were numbered in a lexicographic order. For the first

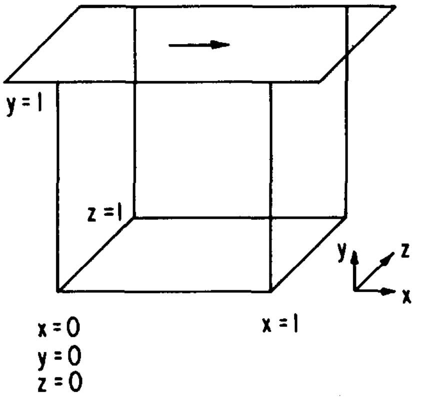


Fig. 4. Problem considered.

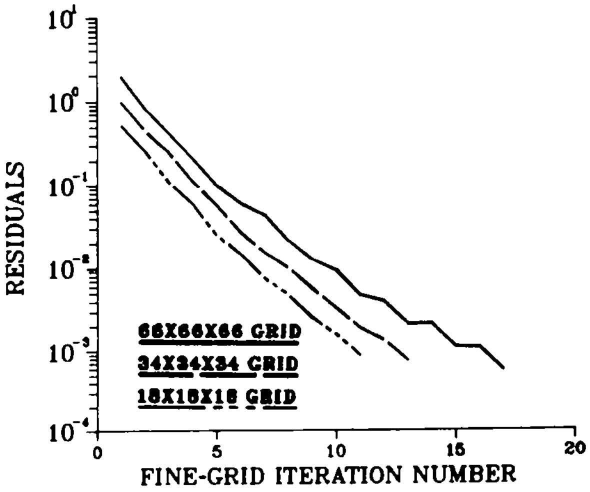


Fig. 5. Rate of convergence for Re = 100.

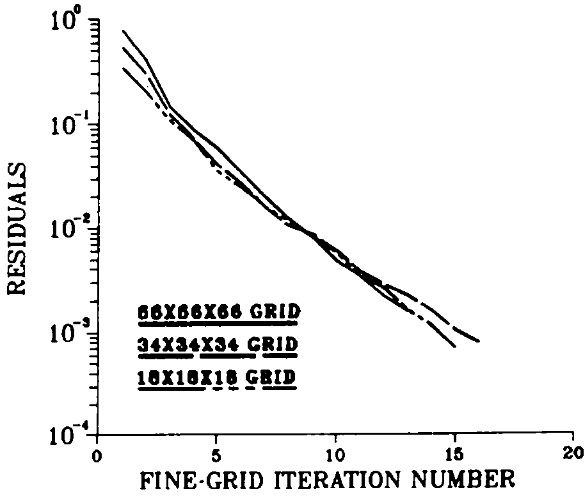


Fig. 6. Rate of convergence for Re = 400.

three Reynolds numbers a value of 0.8 for the relaxation factor was found to give the best rate of convergence. For Re = 3200, a value of 0.7 was used.

The rate of convergence of the calculations, as monitored by the summed average residuals, is shown in Figs. 5, 6, 7, and 8. It is seen that the residuals decrease rapidly by about three orders of magnitude in roughly twenty iterations. This is very encouraging because other algorithms based on primitive variable formulation typically require several hundred iterations, especially at fine grids and large Reynolds numbers. The CPU times for the matrix of calculations are given in Table 1. The times are on an IBM 3033 machine with FORTHX compiler with OPT(2) optimization. Also given in Table 1 are numbers of fine-grid iterations and the equivalent work units for the total calculation (a work unit is equal to the time for one

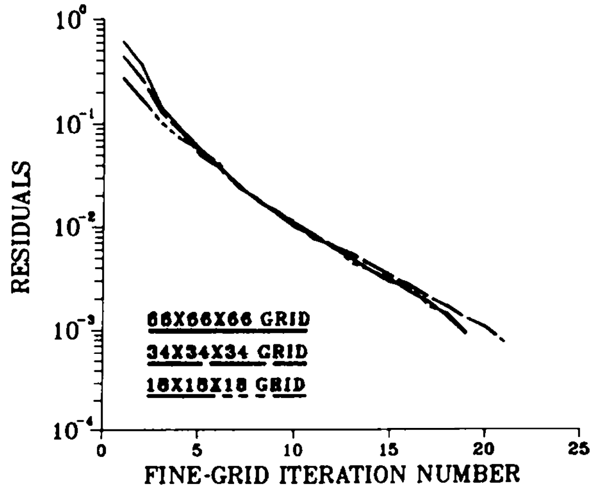


Fig. 7. Rate of convergence for Re = 1000.

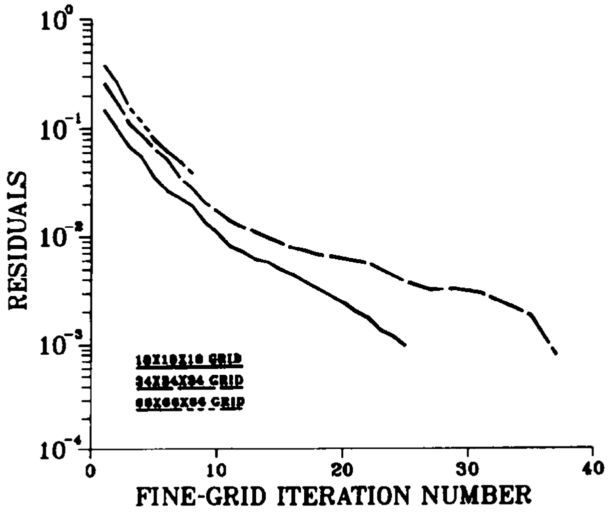


Fig. 8. Rate of convergence for Re = 3200.

Table 1
CPU times (min:sec) and work units 10^{-3} accuracy

Grid \ Re	100	400	1000	3200
18 × 18 × 18				
CPU time	0:18	0:26	0:34	0:40
Iterations	11	15	19	25
Work units	19	29	37	43
34 × 34 × 34				
CPU time	2:38	3:43	5:20	7:47
Iterations	13	16	21	37
Work units	21	29	42	61
66 × 66 × 66				
CPU time	26:19	24:34	37:48	—
Iterations	17	14	19	—
Work units	26	25	38	—

fine-grid SCGS iteration). It is seen that the CPU times increase almost linearly with the number of grid nodes except at Re = 3200 where some nonlinearity is observed. This confirms the attractive feature of the multigrid technique. Another attractive feature of the present algorithm is that computer storage is necessary only for the variables u, v, w, and p and the vectors of residuals. The coefficients are evaluated dynamically during the iteration stage for each node without storing them. The variables are stored for all the grids including the coarse ones. This means that an additional 1/7th of the storage for the finest grid is required. The residuals, however, are stored only for the coarse grids. The total array storage for a $(m \times m \times m)$ grid is approximately $5m^3$. Thus for the 66 × 66 × 66 grid (counting boundary nodes), a total job-card storage of 1600K words was adequate.

5. Results

The three-dimensional flow in a cubic cavity has a very complex structure. The bulk of the flow consists of a primary vortex structure similar to that in a two-dimensional square cavity but is significantly more complicated because of the end walls. In planes perpendicular to the z-direction (spanwise), the flow consists of a central vortex with two corner vortices at the bottom wall (at higher Re a vortex at the top left corner may also form). The strengths of these vortices vary with the distance from the end walls. At locations close to the end walls, the flow is retarded by the wall friction; consequently, the vortices are weaker near the end walls. In the central plane, the strengths of the vortices are of the same order of magnitude as those of a two-dimensional flow. In planes perpendicular to the x-direction (i.e., y-z planes), the flow consists of roll-up corner vortices which again are strongest in the central plane. These roll-up vortices gradually develop from $x = 0$ to $x = 0.5$ and then dissipate near $x = 1.0$. The complete details of the observed three-dimensional flow field will be presented in a forthcom-

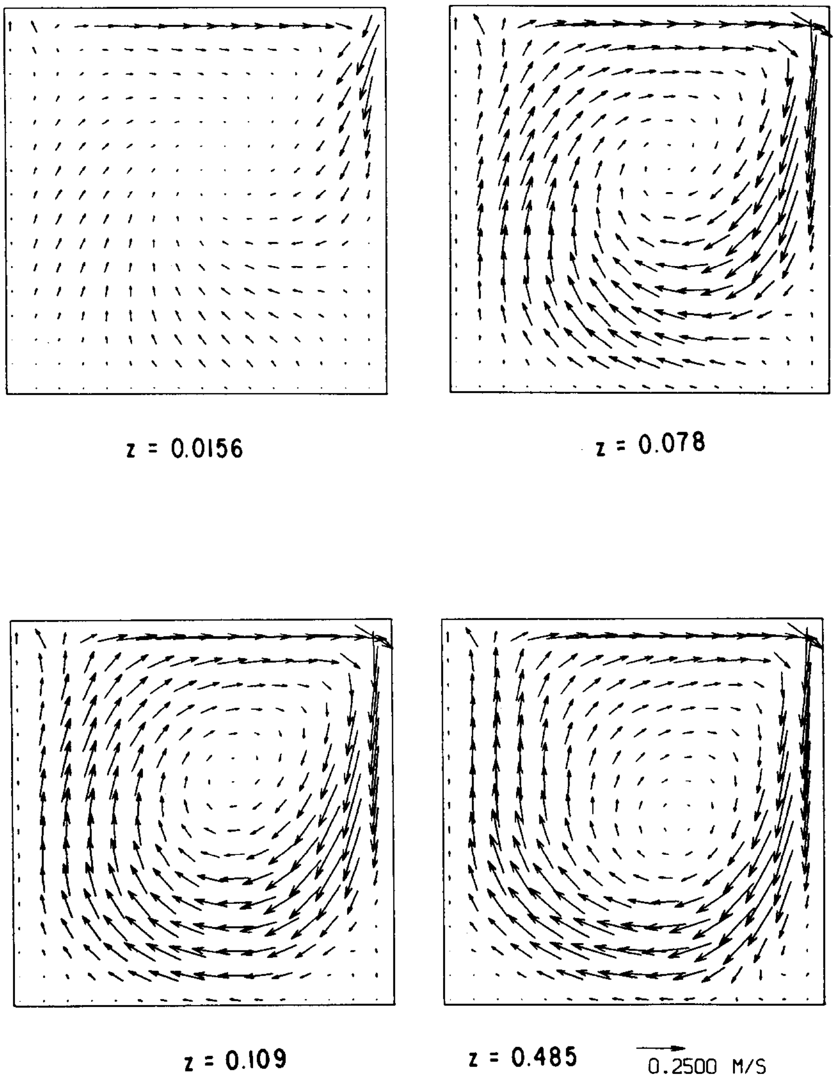


Fig. 9. Flow development in x-y planes.

ing paper. For brevity, only a few figures of the flow structure for Re = 1000 are given in Figs. 9 and 10. These figures show the typical flow patterns, and are self-explanatory. These flow fields are in good qualitative agreement with flow visualizations and calculations of Koseff et al. [16], for a 3:1 spanwise ratio rectangular cavity.

6. Summary

A calculation procedure for steady-state three-dimensional Navier–Stokes equations in the primitive variable formulation is presented in detail. The method is based on a coupled

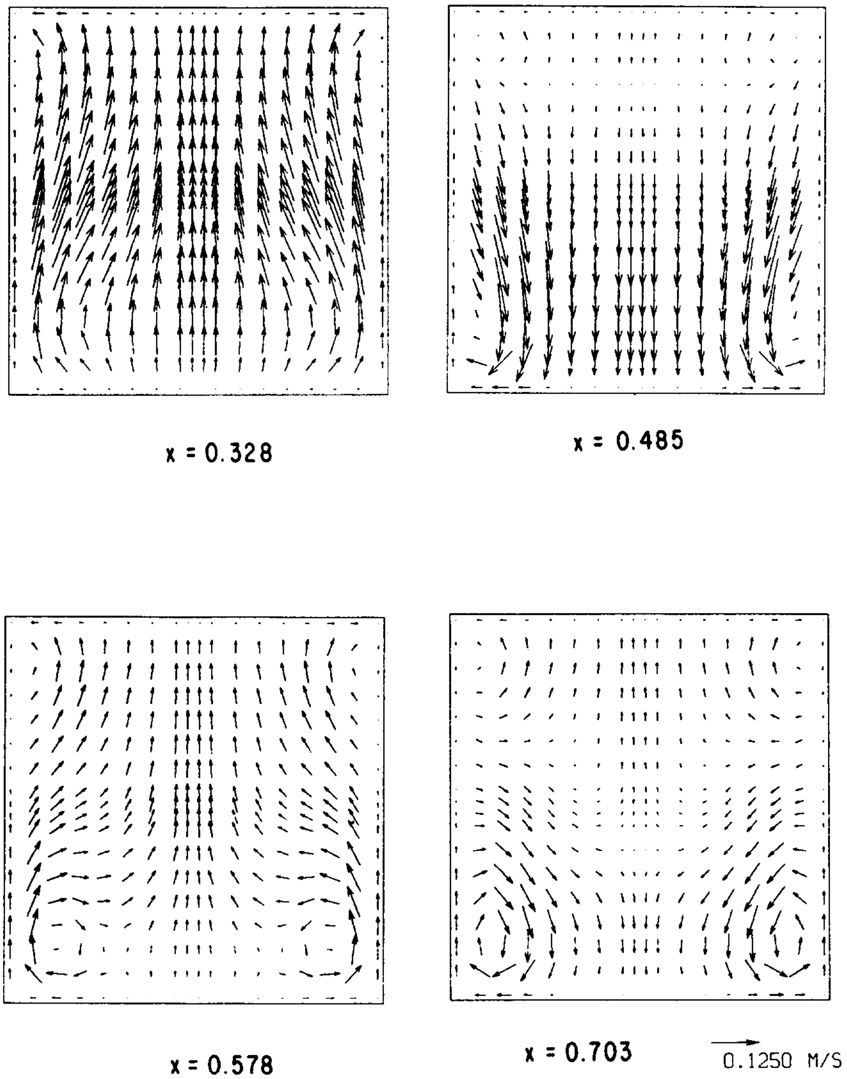


Fig. 10. Flow development in y-z planes.

solution of the finite difference equations by the multigrid technique. A symmetrical coupled Gauss–Seidel (SCGS) technique is used for the smoothing process. Calculations have been made of the three-dimensional flow in a cubic cavity for Reynolds numbers up to 3200 and with finite difference grids consisting of as many as 66 × 66 × 66 nodes. Good convergence has been observed in all the cases. The computer times are observed to vary as O(n), the theoretically expected rate for the multigrid technique. The current CPU times are significantly smaller than other primitive variable formulations using decoupled solutions and one-grid iteration concepts. The present algorithm has the potential of being the basis for a general-purpose algorithm that can calculate many practical three-dimensional turbulent reacting/nonreacting flows.

Appendix A

At each node (i, j, k), the set of equations in terms of corrections to the variables can be formulated as follows.

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 & a_{17} \\ 0 & a_{22} & 0 & 0 & 0 & 0 & a_{27} \\ 0 & 0 & a_{33} & 0 & 0 & 0 & a_{37} \\ 0 & 0 & 0 & a_{44} & 0 & 0 & a_{47} \\ 0 & 0 & 0 & 0 & a_{55} & 0 & a_{57} \\ 0 & 0 & 0 & 0 & 0 & a_{66} & a_{67} \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}, \tag{A1}$$

where

$$\begin{aligned} &a_{11} = (A_C^u)_{i-1/2,j,k}, && a_{22} = (A_C^u)_{i+1/2,j,k}, && a_{33} = (A_C^v)_{i,j-1/2,k}, \\ &a_{44} = (A_C^v)_{i,\,j+1/2,k}, && a_{55} = (A_C^w)_{i,j,k-1/2}, && a_{66} = (A_C^w)_{i,j,k+1/2}, \\ &a_{17} = 1/\rho\delta x\ , && a_{27} = -1/\rho\delta x\ , && a_{37} = 1/\rho\delta y\ , \\ &a_{47} = -1/\rho\delta y\ , && a_{57} = 1/\rho\delta z\ , && a_{67} = -1/\rho\delta z\ , \\ &a_{71} = -1/\delta x, && a_{72} = 1/\delta x, && a_{73} = -1/\delta y, \\ &a_{74} = 1/\delta y, && a_{75} = -1/\delta z, && a_{76} = 1/\delta z, \\ &X_1 = u'_{i-1/2,j,k}, && X_2 = u'_{i+1/2,j,k}, && X_3 = v'_{i,j-1/2,k}, \\ &X_4 = v'_{i,j+1/2,k}, && X_5 = w'_{i,j,k-1/2}, && X_6 = w'_{i,j,k+1/2}, \qquad X_7 = p'_{i,j,k}\,. \end{aligned} \tag{A.2}$$

The solution vector X can be obtained from the following operations: define

$$\begin{aligned} &r_1 = a_{71}/a_{11}, && r_2 = a_{72}/a_{22}, && r_3 = a_{73}/a_{33}, \\ &r_4 = a_{74}/a_{44}, && r_5 = a_{75}/a_{55}, && r_6 = a_{76}/a_{66}, \end{aligned}$$

$$\mathrm{DEN} = r_1 a_{17} + r_2 a_{27} + r_3 a_{37} + r_4 a_{47} + r_5 a_{57} + r_6 a_{67}\,,$$

then,

$$\begin{aligned} &X_7 = (r_1 b_1 + r_2 b_2 + r_3 b_3 + r_4 b_4 + r_5 b_5 + r_6 b_6 - b_7)/\mathrm{DEN} \\ &X_1 = (b_1 - a_{17}X_7)/a_{11}, \quad X_2 = (b_2 - a_{27}X_7)/a_{22}, \quad X_3 = (b_3 - a_{37}X_7)/a_{33}, \\ &X_4 = (b_4 - a_{47}X_7)/a_{44}, \quad X_5 = (b_5 - a_{57}X_7)/a_{55}, \quad X_6 = (b_6 - a_{67}X_7)/a_{66}\,. \end{aligned} \tag{A.3}$$

Acknowledgment

I am grateful to Drs. F.D. Stull and R.R. Craig for their support and encouragement.

References

[1] R. Srinivasan, R. Reynolds, I. Ball, R. Berry, K. Johnson and H. Mongia, Aerothermal modeling program, phase I, final report, NASA-CR-168423, 1983.

[2] G.P. Williams, Numerical integration of the three-dimensional Navier–Stokes equations for incompressible flow, J. Fluid Mech. 37 (1969) 727–750.

[3] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.

[4] F.H. Harlow and J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow, Phys. Fluids 8 (1965) 2182–2189.

[5] S.V. Patankar and D.B. Spalding, A calculation procedure for heat, mass, and momentum transfer in three-dimensional parabolic flows, Internat. J. Heat Mass Transfer 15 (1972) 1787–1806.

[6] S.P. Vanka and G.K. Leaf, An efficient finite-difference calculation procedure for multi-dimensional fluid flows, AIAA-84-1242, presented at the AIAA/SAE/ASME 20th Joint Propulsion Conference, Cincinatti, OH, June 1984.

[7] S.P. Vanka, Block-implicit calculation of steady turbulent recirculating fluid flows, Internat. J. Heat Mass Transfer 28 (1985) 2093–2103.

[8] S.P. Vanka, Calculation of axisymmetric turbulent confined diffusion flames, AIAA J., to appear.

[9] S.P. Vanka, Block-implicit multigrid solution of Navier Stokes equations in primitive variables, J. Comput. Phys., to appear.

[10] D.B. Spalding, A novel finite-difference formulation for differential expressions involving both first and second derivatives, Internat. J. Numer. Meths. Engrg. 4 (1972) 551–559.

[11] A. Brandt, Guide to multigrid development, in: W. Hackbusch and U. Trottenberg, eds., Multigrid Methods, Lecture Notes in Mathematics 960 (Springer, Berlin, 1982).

[12] K. Stüben and U. Trottenberg, Multigrid methods, fundamental algorithms, model problem analysis and applications, in: W. Hackbusch and U. Trottenberg, eds., Multigrid Methods, Lecture Notes in Mathematics 960 (Springer, Berlin, 1982).

[13] U. Ghia, K.N. Ghia, and C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.

[14] R. Schreiber and H.B. Keller, Driven cavity flows by efficient numerical techniques, J. Comput. Phys. 49 (1983) 310–333.

[15] A.S. Benjamin and V.E. Denny, On the convergence of numerical solutions for 2-D flows in a cavity at large Re, J. Comput. Phys. 33 (1979) 340–358.

[16] J.R. Koseff and R.L. Street, Visualization studies of a shear driven three-dimensional recirculating flow, J. Fluids Engrg. 106 (1984) 21–29.