INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 12, 81–92 (1991)

USING A SEGREGATED FINITE ELEMENT SCHEME TO SOLVE THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

C. T. SHAW

Department of Engineering, University of Warwick, Coventry CV4 7AL, U.K.

SUMMARY

In this paper, a segregated finite element scheme for the solution of the incompressible Navier–Stokes equations is proposed which is simpler in form than previously reported formulations. A pressure correction equation is derived from the momentum and continuity equations, and equal-order interpolation is used for both the velocity components and pressure. Algorithms such as this have been known to lead to checkerboard pressure oscillations; however, the pressure correction equation of this scheme should not produce these oscillations. The method is applied to several laminar flow situations, and details of the methods used to achieve converged solutions are given.

KEY WORDS Incompressible flow Finite element methods

INTRODUCTION

As the use of computational fluid dynamics grows, so the complexity of the simulations increases. One aspect of this increasing complexity is the modelling of flows in very complicated geometries. Here complex geometries are taken to be those where not only do the bounding surfaces of the flow have complex curvature, but also where the computational mesh required to describe the volume of fluid cannot have a regular structure or connectivity.

Today, two main numerical techniques are used to discretize the governing equations; finite volume (or difference) methods and finite element methods. Traditionally, finite volume techniques have demanded the use of a body-fitted mesh with a regular topology. These meshes are formed from a topological cuboid of elements, which can be stretched to fit any surface provided that the topological structure of the mesh is maintained. Consequently, with the original finite volume schemes, some complex geometrical shapes can be meshed but others can not.

When the finite element method is used, the formulation of the equations imposes no restriction on the mesh topology, and so flows in complex geometries, where an irregular mesh topology is required, can be simulated. When solving fluid flow problems, finite element methods have not been as popular as finite volume methods, since the original finite element solution schemes were computationally much more expensive to run.[1] Two main reasons can be cited for this: the difficulty in proceeding with the calculation because the momentum and continuity equations do not yield the pressure in a straightforward way, and the use of direct solvers to obtain the solution of a set of simultaneous linear equations.

Finite volume algorithms[2] solve a pressure correction equation which is based on the continuity equation, whereas the earlier finite element algorithms solved for the velocities and pressure in a

0271–2091/91/010081–12$06.00

Received June 1989
Revised January 1990

coupled way. This coupled scheme leads to a larger set of linear equations having to be solved, and so the solution time is correspondingly longer, and if a direct solver is used instead of an iterative one, the solution time is increased further.

Recently, finite volume schemes have become available that can handle irregular mesh topologies,[3] and finite element schemes have been reported[4,5] that use a segregated approach based on a pressure correction equation. This paper describes the development of such a segregated finite element scheme, which is simpler in its formulation than previous ones, and details both the computational aspects of the implementation as well as the practical numerical aspects of obtaining converged solutions. Results are given for several laminar flow cases.

SEGREGATED FINITE ELEMENT SOLUTION SCHEMES

The basic equations

For steady, incompressible, viscous flow in two dimensions the momentum equations are

$$\begin{aligned} \rho u\frac{\partial u}{\partial x}+\rho v\frac{\partial u}{\partial y}&=-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right),\\ \rho u\frac{\partial v}{\partial x}+\rho v\frac{\partial v}{\partial y}&=-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right), \end{aligned} \tag{1}$$

where u and v are the velocity components in the x- and y-directions respectively, p is the pressure, ρ is the fluid density and μ is the fluid viscosity. The continuity equation is given by

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0. \tag{2}$$

An equation for the pressure can be derived by differentiating equations (1) by x and y respectively and then applying (2) to give

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=-\rho\left(\frac{\partial u}{\partial x}\right)^2-2\rho\left(\frac{\partial u\partial v}{\partial x\partial y}\right)-\rho\left(\frac{\partial v}{\partial y}\right)^2. \tag{3}$$

Review of previous algorithms

The momentum equations (1) contain three unknowns—the two velocity components and the pressure—whereas the continuity equation (2) contains only the velocity components. It is clear that there are three equations for three unknowns, but the form of equation (2) precludes a solution for the velocity components and so they must be found from equations (1). To calculate the pressure, early finite element algorithms solved for the velocity components and pressure together, i.e. solving all three equations simultaneously.[1] This scheme is successful only if the interpolations used for the variables satisfy the Babuska–Brezzi conditions described by Zienkiewicz and Taylor.[6] As was said in the introduction, the simultaneous solution of the momentum and continuity equations leads to an inefficient algorithm, but the penalty method[7] can reduce the inefficiency by solving for the velocity components together and then finding the pressure from a Lagrange multiplier and the velocity components.

Finite volume techniques use a segregated solution scheme for these equations, where the velocity components are found from equation (1) and then pressure and velocity corrections are found to enforce equation (2). Each updated value is found from

$$u=u^*+u', \qquad v=v^*+v', \qquad p=p^*+p', \tag{4}$$

where the starred values satisfy the momentum equations (1) and the primed values are the correction values used to enforce continuity. This scheme is commonly known as the SIMPLE algorithm.[2]

If the finite element and finite volume techniques are seen purely as complementary methods of solving partial differential equations, then there is no reason for a SIMPLE-like finite element scheme to be unsuccessful. An early velocity correction procedure has been described by Schneider and Raithby,[8] which solves equations (1) for the two velocity components and then finds correction values by ensuring that equation (2) is satisfied. This is done by assuming that the corrections are the gradient of a scalar potential. Equation (3) can then be used to find the pressure directly; however, this equation demands the values of the pressure gradients at the boundaries of the domain, and these can only be found from equations (1) if second-order elements are used for the velocity components.

Recently, two truly SIMPLE-like finite element schemes have been published, which solve equations (1) first and then solve a pressure correction form of the continuity equation to find velocity corrections and a pressure or pressure correction to ensure that continuity is satisfied. Rice and Schnipke[4] use first-order elements for the velocity components and pressure, solving for pressure using boundary conditions which contain terms containing mass flux. Benim and Zinser[5] calculate the pressure correction variable and use a bilinear velocity–constant pressure element or a composite nine-noded element. This composite element removes the checkerboard modes found with the simpler element.

Current solution scheme

Ideas from the two segregated finite element schemes mentioned above have been used to produce a simpler formulation which uses bilinear elements for both velocity and pressure.

Each variable, say ϕ, is described by a local polynomial on an element which links ϕ to the values at the nodes; that is,

$$\phi = \sum_{i=1}^{n} N_i \phi_i, \tag{5}$$

where N_i is a polynomial trial function and n is the number of nodes on the element.

Equations (1) are discretized using the standard Galerkin technique, premultiplying the equation by the test function and integrating over the domain. For example, the x-momentum equation for each element becomes

$$\int_\Omega N_i \left[\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} - \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) \right] \mathrm{d}\Omega = \int_\Omega -N_i \frac{\partial p}{\partial x} \mathrm{d}\Omega, \tag{6}$$

where Ω signifies the domain under investigation, whose foundary will be denoted by Γ. Integrating by parts and substituting for the discrete form of the velocities, the discretized forms of equations (1) on each element are

$$\int_\Omega \left(\rho u N_i \frac{\partial N_j}{\partial x} + \rho v N_i \frac{\partial N_j}{\partial y} + \mu \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \mu \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) u_j \mathrm{d}\Omega = \int_\Omega -N_i \frac{\partial p}{\partial x} \mathrm{d}\Omega + \int_\Gamma N_i \left(\mu \frac{\partial u}{\partial x} n_x + \mu \frac{\partial u}{\partial y} n_y \right) \mathrm{d}\Gamma, \tag{7}$$

$$\int_\Omega \left(\rho u N_i \frac{\partial N_j}{\partial x} + \rho v N_i \frac{\partial N_j}{\partial y} + \mu \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \mu \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) v_j \mathrm{d}\Omega = \int_\Omega -N_i \frac{\partial p}{\partial y} \mathrm{d}\Omega + \int_\Gamma N_i \left(\mu \frac{\partial v}{\partial x} n_x + \mu \frac{\partial v}{\partial x} n_y \right) \mathrm{d}\Gamma, \tag{8}$$

where n_x and n_y are the components of the unit vector normal to the boundaries in the outward direction.

The boundary terms in equations (7) and (8) can usually be ignored, since either the velocity is given on the boundary or the flux is zero. Equations (7) and (8) can be rewritten in matrix form for an element as

$$\mathbf{A}u_j = \mathbf{B}p_j, \qquad \mathbf{A}v_j = \mathbf{C}p_j, \tag{9}$$

where matrix **A** is a function of local velocities, being the sum of a discrete convection operator and a discrete Laplace operator. If the velocity and pressure are assumed to have the form given in equation (4), then

$$\mathbf{A}(u_j^* + u_j') = \mathbf{B}(p_j^* + p_j'). \tag{10}$$

Since the solution of equation (7) is found to be

$$\mathbf{A}u_j^* = \mathbf{B}p_j^*, \tag{11}$$

then from equation (10)

$$\mathbf{A}u_j' = \mathbf{B}p_j'; \tag{12}$$

hence the nodal values of u' and v' may be described as

$$u_j' = \mathbf{A}^{-1}\mathbf{B}p_j', \qquad v_j' = \mathbf{A}^{-1}\mathbf{C}p_j'. \tag{13}$$

The matrix **A** can be described as the sum of a matrix **D** which contains the diagonal terms of **A** and a matrix **F** which contains the off-diagonal terms. Hence equations (13) can be further simplified by ignoring the off-diagonal terms to give

$$u_j' = \mathbf{D}^{-1}\mathbf{B}p_j', \qquad v_j' = \mathbf{D}^{-1}\mathbf{C}p_j'. \tag{14}$$

This simplification for the matrix **A** can be made because the pressure correction term approaches zero as the solution converges. Other simplifications could also be made, but taking the diagonal is the simplest choice.

Once p' has been found, u' and v' have to be found at each node. Equations (14) are valid for each element and so they have to be integrated over the domain:

$$\int_\Omega N_i N_j u_j' \, d\Omega = \int_\Omega N_i N_j \mathbf{D}^{-1}\mathbf{B}p_j' \, d\Omega, \tag{15}$$

$$\int_\Omega N_i N_j v_j' \, d\Omega = \int_\Omega N_i N_j \mathbf{D}^{-1}\mathbf{C}p_j' \, d\Omega. \tag{16}$$

It should be noted that this procedure described by equations (15) and (16) can be simplified to avoid the inversion of the full mass matrix on the left-hand side by mass lumping.

Now the continuity equation must be discretized to form a pressure correction equation. This is done using the standard Galerkin technique on equation (2) to give

$$\int_\Omega N_i \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) d\Omega = 0, \tag{17}$$

which becomes

$$\int_\Omega N_i \left(\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} \right) d\Omega = -\int_\Omega N_i \left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right) d\Omega \tag{18}$$

when the velocity components are split using equation (4).

Integrating the left-hand side by parts gives

$$-\int_\Omega \frac{\partial N_i}{\partial x} N_j u'_j \,d\Omega - \int_\Omega \frac{\partial N_i}{\partial y} N_j v'_j \,d\Omega + \int_\Gamma N_i(u' n_x + v' n_y)\,d\Gamma = -\int_\Omega N_i\left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y}\right) d\Omega. \tag{19}$$

Here the third term on the left-hand side is a boundary term containing the correction velocities. In many problems the velocity will be specified at the boundaries and so the correction velocities at those boundaries will be zero. Equally, the pressure may be specified and so its correction values will also be zero. Consequently the boundary integral can be ignored in many problems. Substituting equation (14) into equation (19) gives

$$-\int_\Omega \frac{\partial N_i}{\partial x} N_j \mathbf{D}^{-1}\mathbf{B}\,d\Omega\, p'_j - \int_\Omega \frac{\partial N_i}{\partial y} N_j \mathbf{D}^{-1}\mathbf{C}\,d\Omega\, p'_j = -\int_\Omega N_i\left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y}\right) d\Omega, \tag{20}$$

which is the desired pressure correction equation. It is this equation which differs from those of Rice and Schnipke[4] and Benim and Zinser,[5] being of a much simpler form. This last equation can also be written

$$(\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{B} + \mathbf{C}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{C})p'_j = \delta^*, \tag{21}$$

where δ^* accounts for the right-hand side of equation (20), the mass source due to the velocities calculated using the momentum equations. If the entries in matrix $\mathbf{D}$ are positive, then the coefficient matrix on the left-hand side of equation (21) will be symmetric and positive definite.

COMPUTATIONAL ASPECTS OF THE SCHEME

Programming the algorithm

One aim of this work has been to investigate the way in which a segregated finite element solution scheme works in practice. To do this, the above two-dimensional scheme has been implemented using only four-noded quadrilateral elements. The program has been written as a testbed for the algorithm, ignoring considerations of computational efficiency at this stage. As a result, all the solutions to equations (7), (8), (20), (15) and (16) are carried out with direct solvers.

The algorithm consists of the following stages.

(a) Find u^* from equation (7).
(b) Find v^* from equation (8).
(c) Find p' from equation (20).
(d) Find u' from equation (15).
(e) Find v' from equation (16).
(f) Form u, v and p from equation (4).
(g) Loop back to (a) if there are more iterations to perform, else stop.

Initial conditions

All of the problems have been calculated with the initial values of the variables set to zero at all nodes, except at those nodes where boundary conditions are specified. At these nodes the fixed boundary value is enforced for all the iterations.

Achieving convergence

As with most non-linear problems, some form of relaxation is required to enable a converged solution to be obtained. Standard linear relaxation can be applied to the variables u^* and v^* as follows:

$$\phi = \alpha\phi_{\text{new}} + (1-\alpha)\phi_{\text{old}}, \tag{22}$$

where ϕ is the relaxed value of either u^* or v^* that will be carried forward in the calculation, ϕ_{new} is the value of u^* or v^* calculated from the momentum equation (7) or (8). ϕ_{old} is the previous value of u^* or v^*, and α is the relaxation factor (where $0 \leqslant \alpha \leqslant 1$). Similarly, the updated values of u, v and p are formed by modifying equation (4) to be

$$\phi = \phi^* + \beta\phi', \tag{23}$$

where ϕ is any of the three variables and β is a relaxation factor ($0 < \beta \leqslant 1$).

These two equations, (22) and (23), are used to control the values of the variables and so prevent the solution process diverging. Two main sources of divergence have been noted during the iterative process. First, it has been found that the pressure correction is usually too great and so equation (23) has to be employed when forming the updated pressure. Secondly, enforcing continuity can also lead to divergence. The velocities u' and v' should not normally be relaxed using equation (23), since such relaxation would mean that continuity is not satisfied. In some problems, however, the application of correction velocities without relaxation has led to wiggles in the velocity field near a wall boundary in the first few steps of the solution. If this occurs, a heavy relaxation has been applied for several steps to all three correction variables, u', v' and p', allowing the momentum equations to modify the velocity profiles near any walls in a smooth manner.

Upwinding

If the local cell Reynolds number is too large, the solution of the momentum equations (7) and (8) is convection-dominated. This can lead to wiggles in the velocity field and prevent convergence of the iterative scheme.[9] Where necessary, the convection terms in equations (7) and (8) have been modified to provide upwinding using the quadrature upwinding scheme of Hughes.[10] This scheme has been chosen because it is perhaps the simplest to adopt even though it is numerically diffuse. The scheme presented by Brooks and Hughes[9] is only slightly more difficult to adopt but contains much less cross-wind diffusion.

Stability

As has been stated, this algorithm has been implemented using equal-order interpolation for the velocity components and pressure. Such schemes are regarded as violating the Babuska–Brezzi conditions and as leading to oscillations in the pressure solution.

A readable account of the Babuska–Brezzi conditions is given by Zienkiewicz and Taylor in Chapter 12 of their book.[6] There are two conditions: the first refers to the coupled solution of the variables and states that the matrix operating on the pressure variables must be non-singular, whilst the second states that the divergence operator on the pressure must not have any null states when the pressure is not zero throughout the domain. The first condition is met when the number of unknown nodal values of velocity is greater than the number of unknown nodal values of pressure. The second condition ensures that only realistic modes of pressure are present in the solution. One example of this is a checkerboard mode which can exist when bilinear interpolation is used for the pressure, since the divergence operator on pressure cannot detect such a mode and consequently these modes do not affect the momentum equations.

Finite volume algorithms can overcome these problems in several ways. Many algorithms use a staggered grid for the velocity components and the pressure. Others modify the pressure terms in the momentum equations or the continuity equation. For example, the Rhie and Chow algorithm[11] allows the velocity components and the pressure to be stored at the cell centres, and interpolates the velocities from this position to the faces of the cells, as shown by Burns and Wilkes.[12]

Finite element algorithms can also produce pressure oscillations if the two Babuska–Brezzi conditions are not met. For the segregated finite element solution the first condition can be dealt with quite simply. Equation (21) is the matrix form of the discrete continuity equation, and the matrix that has to be inverted to find p' is on the left-hand side. Since the solution scheme is decoupled, this matrix will not be singular. If the scheme is coupled, this might not be the case, and the first Babuska–Brezzi condition states the requirements for the matrix not to be singular.

The investigation of the second condition, which refers to the admission of spurious pressure modes, is more difficult, since the divergence operator on the pressure variable in the discretized momentum equations (7) and (8) will not detect several unrealistic modes of pressure. These unrealistic modes of pressure must be generated from the pressure correction equation (20), and so this has been investigated further. For the simple laminar flow between parallel plates, on a mesh of square elements, the left-hand-side element equations have been calculated and assembled. Hence the resulting computational molecule used to calculate the values of the pressure correction has been found and is shown in Figure 1. From this, the value of the pressure correction at the centre node can be seen to depend on the values at each of its eight nearest neighbours, there being no zeros in the molecule, and so there is no decoupling of any of the pressure correction nodal values.

Looking at equation (20), each of the two components of the left-hand side is formed from the multiplication of two square matrices. This can be rearranged such that the diagonal terms of matrix **D** are placed on the right-hand side, and then it is easily demonstrated that the resulting left-hand-side matrix for the pressure correction is proportional to the matrix that would have been formed had the left-hand side been the Laplace operator.

Since the pressure correction values are formed from a Laplacian-like operator, the solutions for pressure will be smooth and the oscillatory modes never formed. Consequently the fact that the momentum equations are insensitive to spurious pressure modes does not cause concern.

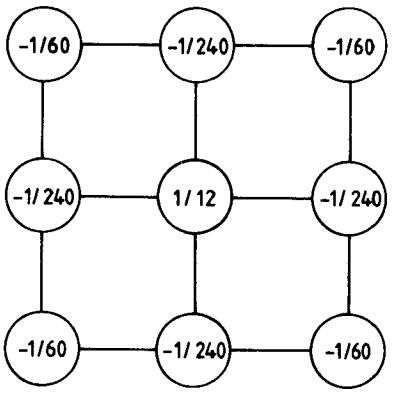

Figure 1. Computational molecule for the pressure correction equation

SOME SOLUTIONS FOR LAMINAR FLOWS

Flow under constant pressure gradient

The simplest flow that has been calculated is a unidirectional flow driven by a constant pressure gradient. Because the velocity profile is parabolic in the y-direction, the pressure gradient can be calculated from the x-momentum equation (1). This flow is defined by

$$u = 1{\cdot}0(y - y^2), \qquad \partial p/\partial x = -2{\cdot}0\mu \tag{24}$$

for the domain $0 \leqslant x \leqslant 1{\cdot}0$, $0 \leqslant y \leqslant 0{\cdot}4$ and has been calculated using eight rectangular elements, for which $\delta x = 0{\cdot}5$, $\delta y = 0{\cdot}1$. The velocities have been specified on all boundaries except the outlet, where the pressure has been set to zero. Two hundred iterations have been run with no upwinding for a flow with unit viscosity and density. The relaxation parameters have been set to $\alpha = 0{\cdot}9$ for u^* and v^*, $\beta = 0{\cdot}1$ for p' and $\beta = 1{\cdot}0$ for u' and v'.

Figure 2 shows the variation of the pressure at the inlet wall node and the average nodal value of p' with iteration number. In this figure the inlet wall node has been chosen because it is the least accurate of the five nodes at the inlet. Even so, the pressure can be seen to converge to a value close to the expected value of 2·0, and the average value of the pressure correction can be seen to fall rapidly as the solution progresses.

To assess the accuracy of the method, Table I shows the values of maximum error at any node for both u and p against iteration number. From this it can be seen that the error in the u-velocity is much smaller than the error in the pressure and that the error in the pressure is above 2%, even after 200 iterations. Further calculations have also been made with all the relaxation factors set to unity except for the factor β on pressure. By calculating the accuracy of the nodal pressures after each iteration, the calculation has been stopped when the maximum error is below a certain value. Table II contains the results of this exercise. It can be seen that higher relaxation factors can be used on the pressure, to achieve more economical results, and that even small values of overrelaxation can be used. One main feature here is the large number of iterations required to reduce the error from 3% to 2%, showing that the convergence rate becomes extremely slow as

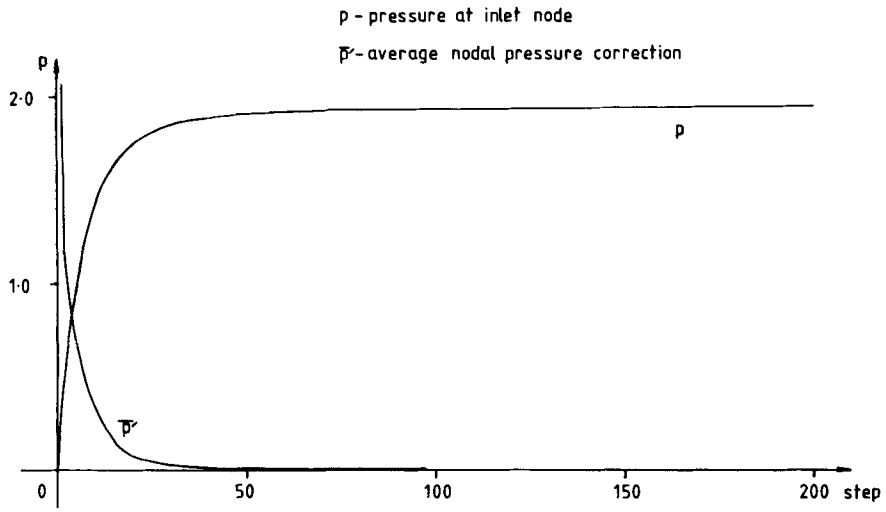

Figure 2. Pressure history for a flow with constant pressure gradient

Table I. Errors at a series of iterations

Iteration number	Maximum error in u (%)	Maximum error in p (%)
40	1·44	5·80
80	0·34	3·58
120	0·11	3·04
160	0·07	2·83
200	0·05	2·67

Table II. Iterations to achieve a given minimum error

Relaxation factor on pressure	Iterations for error 5%	Iterations for error 4%	Iterations for error 3%	Iterations for error 2%
0·1	46	63	115	505
0·2	23	31	57	253
0·3	15	20	38	169
0·4	11	15	28	127
0·5	9	12	22	101
0·6	7	10	18	84
0·7	6	8	16	72
0·8	5	7	13	63
0·9	6	7	12	56
1·0	7	9	9	51
1·1	12	13	15	46
1·2	21	34	39	47

the error is reduced. To achieve 1% error, 129 iterations are required for a relaxation factor $\beta = 1{\cdot}1$. In fact, with a factor $\beta = 1{\cdot}2$ the solution eventually diverges, as it always does with a factor $\beta = 1{\cdot}3$.

This slow convergence may be due to the use of the diagonal to approximate matrix **A**. Other approximations could be made, and this will be the subject of further research.

Driven cavity flow

A truly two-dimensional flow has been calculated; the flow in a square cavity which is driven by a moving lid. Here the upper side of the square is given a fixed uniform velocity in the horizontal direction, and the velocity on the other boundaries is set to zero. In this case the pressure is set to zero at one point on the boundary to anchor its value. This forms a suitable numerical experiment since Burggraf[13] has provided very accurate numerical solutions using a stream-function–vorticity formulation solved on a very fine grid.

At a Reynolds number of 100, based on the lid velocity and cavity width, upwinding is not required to solve this problem on a mesh of 16×16 elements. It should be noted that the mesh is biased towards all the solid boundaries. Obtaining convergence for this problem is more difficult than for the previous problem. With relaxation factors $\alpha = 1{\cdot}0$ for u^* and v^* and $\beta = 1{\cdot}0$ for u' and v', the relaxation factor for pressure must be set to a small value such as $\beta = 0{\cdot}1$. Even then the

convergence is not smooth. In fact, setting $\alpha = 0{\cdot}5$ for u^* and v^* improves the convergence rate and so a trade-off in relaxation factors is necessary. Smooth convergence can also be achieved for many other combinations of relaxation factor provided that the higher the value of β for pressure, the lower is the value of α for u^* and v^*. One such combination is $\beta = 0{\cdot}4$ and $\alpha = 0{\cdot}1$. Two hundred iterations have been run for this combination, and Figure 3 shows the variation of u-velocity with height on the vertical centreline of the cavity for a Reynolds number of 100.

Looking at Figure 3, the finite element solution is plotted at 10, 20, 50 and 100 iterations, together with sample points from the Burggraf solution.[13] At 200 iterations the results are indistinguishable from those at 100 iterations to the scale of the figure. Clearly the comparison is good at 100 iterations, but even at 50 iterations the finite element predictions are not too far from the Burggraf results.

Flow over a back step

When calculating two-dimensional flows, the classic problem involving separation is the flow over a backward-facing step. Denham and Patrick[14] have performed a detailed series of experiments on this flow at various Reynolds numbers for a step height of 15 mm, which is one-half of the inlet height.

The mesh for this problem contains some 600 elements and extends six step heights upstream of the step and 44 step heights downstream. A fully developed parabolic velocity profile has been given at the inlet, and the velocities have been set to zero on the walls. The outlet pressure has been specified as zero.

To run this test case, five iterations have been calculated with relaxation factors $\beta = 0{\cdot}01$ on p', $\alpha = 0{\cdot}9$ on u^* and v^* and $\beta = 0{\cdot}01$ on u' and v'. This allows the velocity field to be smooth, eliminating any wiggles at the walls. Then 20 further iterations have been run with relaxation factors $\beta = 0{\cdot}1$ on p', $\alpha = 0{\cdot}5$ on u^* and v^* and $\beta = 1{\cdot}0$ on u' and v' to enforce continuity. The separation lengths have been measured after the first 15 iterations and then after 25 iterations. The difference in the length between the two is about 1%. Over the 25 iterations the average pressure correction has reduced by a factor of the order of 0·0001.

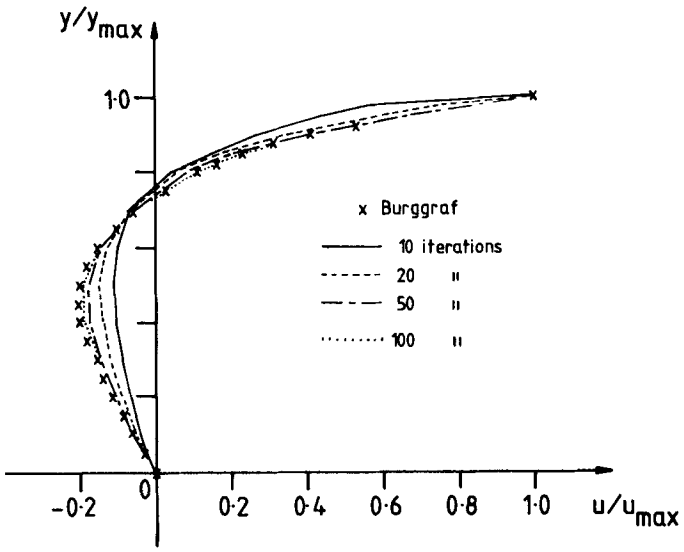

Figure 3. Centreline velocity profiles for a driven cavity flow

For flows at Reynolds numbers (based on step height) of 12·5 and 25, convergence has been achieved without upwinding. However, at Reynolds numbers of 50 and above, upwinding is required to prevent the iterative scheme diverging. In Figure 4 the comparison of the computed separation length with the experimental values for the lower-Reynolds-number cases is given, and Figure 5 shows the velocity profile at one downstream station for a Reynolds number of 73.

Predictions of separation length are reasonable, both with and without upwinding. The error for the case of Reynolds number 100 is probably due to the increasing coarseness of the mesh

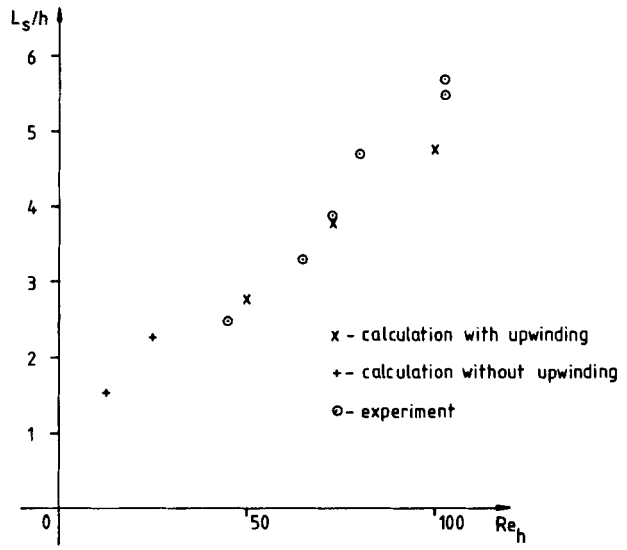

Figure 4. Variation of separation length behind the step with Reynolds number

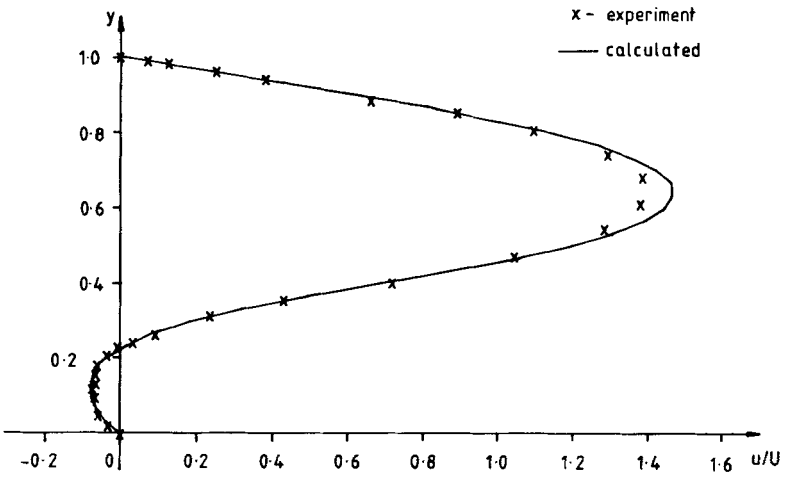

Figure 5. Velocity profile 12 mm downstream of the step

downstream of the step to save on computer storage. At 12 mm downstream of the step the prediction of the velocity component u is very good, as shown in Figure 5, with the reversed flow area being predicted accurately.

CONCLUSIONS

From the results presented here, it is clear that a segregated finite element solution scheme with equal-order interpolation for the velocity components and the pressure can be used. With the pressure correction derived here, checkerboard solutions for pressure should not occur.

If such schemes are to make an impact on the solution of real-life engineering problems, then much more work needs to be done. In particular, the efficiency of the solution scheme needs to be improved by using iterative linear equation solvers and perhaps modifying the approximations in the pressure correction equation, some form of turbulence model needs to be implemented and the scheme needs to be rewritten for three-dimensional time-dependent flows.

ACKNOWLEDGEMENTS

Stimulating discussions have been held with Phil Gresho, Ali Benim and many people at Harwell Laboratory, and the comments of the referees have been most helpful.

REFERENCES

1. C. Taylor and T. G. Hughes, *Finite Element Programming of the Navier–Stokes Equations*, Pineridge Press, Swansea, 1981.
2. S. V. Patankar and D. B. Spalding, 'A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows', *Int. J. Heat Mass Transfer*, **15**, 1787–1806 (1972).
3. R. D. Lonsdale, 'An algorithm for solving thermal hydraulic equations in complex geometries: the ASTEC code', *UKAEA Internal Report*, 1988 (unpublished).
4. J. G. Rice and R. J. Schnipke, 'An equal order velocity–pressure formulation that does not exhibit spurious pressure modes', *Comput. Methods Appl. Mech. Eng.*, **58**, 135–149 (1986).
5. A. C. Benim and W. Zinser, 'A segregated formulation of Navier–Stokes equations with finite elements', *Comput. Methods Appl. Mech. Eng.*, **57**, 223–237 (1986).
6. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method, Vol. 1: Basic Formulation and Linear Problems*, 4th edn, McGraw-Hill, New York, 1988.
7. J. N. Reddy, 'On penalty function methods in the finite-element analysis of flow problems', *Int. j. numer. methods fluids*, **2**, 151–171 (1982).
8. G. E. Schneider and G. D. Raithby, 'Finite element analysis of incompressible fluid flow incorporating equal order pressure and velocity interpolation', in K. Morgan, C. Taylor and C. A. Brebbia (eds), *Computer Methods in Fluids*, Pentech Press, London, 1980, p. 49.
9. A. N. Brooks and T. J. R. Hughes, 'Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **32**, 199–259 (1982).
10. T. J. R. Hughes, 'A simple scheme for developing "upwind" finite elements', *Int. j. numer. methods eng.*, **12**, 1359–1365 (1978).
11. C. M. Rhie and W. L. Chow, 'Numerical study of the turbulent flow past an airfoil with trailing edge separation', *AIAA J.*, **21**, 1527–1532 (1983).
12. A. D. Burns and N. S. Wilkes, 'A finite difference method for the computation of fluid flows in complex three dimensional geometries', *AERE-R 12342*, HMSO, London, 1987.
13. O. R. Burggraf, 'Analytical and numerical studies of the structure of steady separated flow', *J. Fluid Mech.*, **24**, 113–151 (1966).
14. M. K. Denham and M. A. Patrick, 'Laminar flow over a downstream-facing step in a two-dimensional flow channel', *Trans. Inst. Chem. Eng.*, **52**, 361–367 (1974).