INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 8, 671–686 (1988)

ACCELERATION OF THE PRESSURE CORRECTION METHOD FOR A ROTATING NAVIER–STOKES PROBLEM

G. LONSDALE* AND J. E. WALSH

Department of Mathematics, University of Manchester, Manchester M13 9PL, U.K.

SUMMARY

The behaviour of the pressure correction method is studied for the solution of the incompressible steady-state Navier–Stokes and continuity equations in a rotating cylindrical-polar co-ordinate system, the specific problem being that of laminar source–sink flow between two corotating discs. Modifications to improve the linearization and the handling of the rotation terms are introduced, and we compare three extended pressure correction schemes and also the use of a multigrid algorithm in part of the calculation procedure as a linear solver.

KEY WORDS Multigrid Pressure correction Navier–Stokes

1. INTRODUCTION

In this paper we consider the behaviour of the pressure correction method[1] applied to the problem of laminar source–sink flow between two corotating discs, as in Chew.[2] We also consider modifications and extensions of the basic method and the use of a multigrid algorithm in part of the calculation as a linear solver.

A cylindrical-polar co-ordinate system (r, θ, z) rotating at angular velocity Ω (in the θ-direction) is used with variables and parameters as follows:

u	radial velocity
v	tangential velocity
w	axial velocity
p	pressure
ρ	density (assumed constant)
$p' = p - 1/2\rho\Omega^2 r^2$	reduced pressure
μ	dynamic viscosity (assumed constant)
Q	source flow rate
a	radial position of source
b	radial position of sink
s	distance between the discs

* Now at Department of Mathematics, University of Bradford, Bradford, West Yorkshire BD7 1DP, U.K.

0271–2091/88/060671–16$08.00 *Received 12 August 1986*

The incompressible steady-state Navier–Stokes and continuity equations are then

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r u^2)+\frac{\partial}{\partial z}(\rho u w)=-\frac{\partial p'}{\partial r}+\mu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{\partial^2 u}{\partial z^2}-\frac{u}{r^2}\right]+2\rho\Omega v+\rho\frac{v^2}{r}, \tag{1}$$

$$\frac{1}{\rho}\frac{\partial}{\partial r}(\rho r u v)+\frac{\partial}{\partial z}(\rho v w)=\mu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right)+\frac{\partial^2 v}{\partial z^2}-\frac{v}{r^2}\right]-\rho\frac{uv}{r}-2\rho\Omega u, \tag{2}$$

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r u w)+\frac{\partial}{\partial z}(\rho w^2)=-\frac{\partial p'}{\partial z}+\mu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w}{\partial r}\right)+\frac{\partial^2 w}{\partial z^2}\right], \tag{3}$$

$$\frac{1}{r}\frac{\partial}{\partial r}(r u)+\frac{\partial w}{\partial z}=0. \tag{4}$$

Boundary conditions for velocities are taken as follows:

$$u=Q/2\pi r s, \qquad v=0, \qquad w=0 \quad \text{at} \quad r=a, b,$$
$$u=v=w=0 \quad \text{at} \quad z=0, s.$$

In the solution procedure we require boundary conditions only for a correction to the pressure, pp, not for the pressure itself; these conditions are taken to be

$$\partial pp/\partial r=0 \quad \text{at} \quad r=a, b,$$
$$\partial pp/\partial z=0 \quad \text{at} \quad z=0, s.$$

A non-uniform staggered grid was employed as shown in Figure 1. Figure 1 also shows the points associated with the same index; the values $u(i, j)$, $w(i, j)$ refer to different spatial points from $p(i, j)$, $v(i, j)$. The radial and axial velocities are calculated at points halfway between neighbouring points for pressure and tangential velocity.

The finite difference equations are found by integration over control areas.[3] Since the control areas for the non-uniform grid are rectangular and the equation is not necessarily based at the centre of the area, the finite difference equations are more general than the ones used by Chew[2] on a

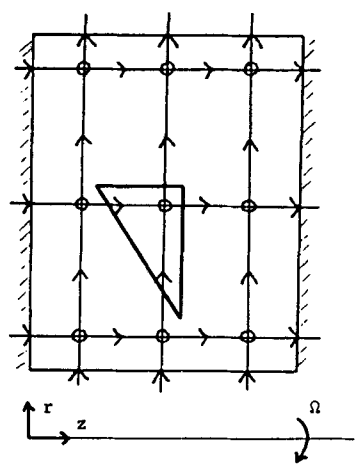

Figure 1. Non-uniform staggered grid: ○, pressure and tangential velocity; →, axial velocity; ↑, radial velocity; ◹, associated points in indexing system

regular grid (and are given in detail in Reference 4). The finite difference replacements of the momentum equations (1)–(3) include the form of upwinding used by Patankar and Spalding;[1] this is referred to by them as a 'high lateral flux modification'.

The linearization of the momentum equations is done by replacing the product of two variables by the product of one 'old' value and one 'new' value, for example in equation (1)

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r u^2) \text{ is replaced by } \frac{1}{r}\frac{\partial}{\partial r}(\rho r u^{k+1}u^k),$$

$$\frac{\partial}{\partial z}(\rho u w) \text{ is replaced by } \frac{\partial}{\partial z}(\rho u^{k+1}w^k),$$

so that the finite difference approximation of (1) gives a linear equation for the new u iterate, u^{k+1}, with coefficients depending on the old iterates u^k, w^k.

The geometry and fluid properties used for numerical tests are as follows:

$a=0{\cdot}019$ m, $b=0{\cdot}19$ m, $s=0{\cdot}0507$ m,

$\mu=1{\cdot}78\times10^{-5}\,\mathrm{kg\,m^{-1}\,s^{-1}}$, $\rho=1{\cdot}225\,\mathrm{kg\,m^{-3}}$, $Q=2{\cdot}761\times10^{-4}\,\mathrm{m^3\,s^{-1}}$.

This corresponds to the test case used by Chew[2] with a mass flow parameter given by

$$C_w=Q\rho/\mu b=100{\cdot}0.$$

Most of the testing to compare the efficiency and behaviour of the basic pressure correction method with our various modifications was done at a rotation rate of $\Omega=1{\cdot}0\,\mathrm{rad\,s^{-1}}$, corresponding to a rotational Reynolds number given by

$$Re_\theta=\Omega b^2\rho/\mu\simeq2{\cdot}5\times10^3.$$

This rotation rate provides a sufficiently complicated flow to enable us to assess the behaviour of the different methods without requiring excessive computing.

Because we need to cluster the grid points near the boundaries, the non-uniform grids used throughout are based on the zeros of the relevant shifted Chebyshev polynomial. For example, if we use n_r radial pressure lines, their radial positions are given by

$$r_i=0{\cdot}5\{(b+a)-(b-a)\cos[(i-0{\cdot}5)\pi/n_r]\},\quad i=1,2\ldots,n_r.$$

2. THE BASIC PRESSURE CORRECTION METHOD

We now discuss briefly the pressure correction method as applied to the problem of Section 1. For details of the pressure correction method in general see References 1 and 5.

The finite difference replacement and linearization of the momentum equations described in Section 1 leads to five-point difference equations for u, v, w at a point P of the form

$$a_p u_p=\sum_i a_i u_i+A_p(p_s-p_n)+S_u, \tag{5}$$

$$b_p v_p=\sum_i b_i v_i+S_v, \tag{6}$$

$$c_p w_p=\sum_i c_i w_i+C_p(p_w-p_e)+S_w, \tag{7}$$

where the summation Σ_i, refers to the four neighbouring points N, E, S, W on the grid. Note that the point P is different for each equation (Figure 1).

The basic idea of the pressure correction method is that after calculating new values for the velocities from the linearized momentum equations, based on the old iterates for velocity and pressure, we use the continuity equation to make changes to the velocities and pressure in order to move closer to satisfying both the momentum equations *and* the continuity equation.

To see this, let p^k be the current pressure distribution and let u^*, w^* be the velocities resulting from solving the radial and axial momentum equations with this pressure distribution, so that pointwise we have

$$a_p u_p^* = \sum_i a_i u_i^* + A_p(p_s^k - p_n^k) + S_u, \tag{8}$$

$$c_p w_p^* = \sum_i c_i w_i^* + C_p(p_w^k - p_e^k) + S_w. \tag{9}$$

Note that we do not include the tangential velocity since it does not appear in the continuity equation and has no direct dependence on pressure.

If p^k were the correct pressure distribution, the velocities u^*, w^* would satisfy the continuity equation. However, this will not be the case in general. We therefore introduce changes to get new iterates u^{k+1}, w^{k+1}, p^{k+1} given by

$$u^{k+1} = u^* + \delta u, \tag{10}$$

$$w^{k+1} = w^*, \delta w, \tag{11}$$

$$p^{k+1} = p^k + pp. \tag{12}$$

The relations between these changes can be seen by substituting equations (10)–(12) into equations (8) and (9), giving

$$a_p \delta u_p = \sum_i a_i \delta u_i + A_p(pp_s - pp_n), \tag{13}$$

$$c_p \delta w_p = \sum_i c_i \delta w_i + C_p(pp_w - pp_e). \tag{14}$$

An equation for the pressure correction pp is obtained by forcing the new u, w iterates to satisfy the continuity equation and using the relations between the velocity and pressure changes. However, the relations (13), (14) would lead to a complicated system of equations for pp. Patankar and Spalding[1] avoid this by truncating (13) and (14) to get approximate relations between the pressure and velocity changes as follows:

$$a_p \delta u_p = A_p(pp_s - pp_n), \tag{15}$$

$$c_p \delta w_p = C_p(pp_w - pp_e). \tag{16}$$

Substituting (10), (11) into the continuity equation with δu, δw given by (15), (16) gives a linear equation for the pressure correction pp. Using this pressure correction, δu, δw are then calculated from equations (15), (16).

We thus have the following basic pressure correction algorithm:

(i) guess initial values for all variables at all the internal grid points;
(ii) use the radial momentum equation to get updated u-values;
(iii) use the axial momentum equation to get updated w-values;

(iv) calculate estimates of the variation of u, w with pressure gradient in the r,z-directions as in (15), (16);
(v) calculate the pressure correction pp;
(vi) update u, w, p according to (10)–(12);
(vii) use the tangential momentum equation to get updated v-values;
(viii) if the root-mean-square (RMS) values of changes made to all the variables are greater than some prescribed tolerance, repeat from (ii).

Before giving results which illustrate the behaviour of the above method, we note the following points.

Solution of the pressure correction equations

The solution of the linear equations for the velocities in our tests was by an alternating-line Gauss–Seidel (ALGS) algorithm using only one double sweep. In the first set of tests the iterative solution of the equation for the pressure correction used a fixed number of double sweeps of the ALGS algorithm; in Section 4 we discuss the use of a linear multigrid algorithm. The emphasis on the solution of the pressure correction equations is in line with the behaviour of the algorithm as observed by Chew[2] and was also supported by numerical experiments.

Under-relaxation

In order to obtain a convergent process, it is essential to use under-relaxation for the four variables u, v, w, p as follows. For the variable x (where x represents one of the velocities) let the updated variable from the solution of the momentum equations be x, let the old x iterate be x^k and the under-relaxation parameter be α_x. Then the updated x variable actually used, x^*, is given by

$$x^* = x^k + \alpha_x(x - x^k), \qquad 0 < \alpha_x < 1.$$

The new pressure iterate is given by modifying equation (12) to

$$p^{k+1} = p^k + \alpha_p pp, \qquad 0 < \alpha_p < 1.$$

Initial values

For all tests the initial values were taken to be the values corresponding to solid body rotation at the rotation rate used for the test, together with zero reduced pressure (note that the co-ordinate system is rotating), i.e. $u = v = w = 0$, $p' = 0$.

Convergence criterion

As the variables vary greatly in magnitude, the convergence criterion was based on relative rather than absolute changes. The RMS was used to measure the variables and the changes made to the variables across the grid, the iteration being terminated when the relative change was less than $1{\cdot}0 \times 10^{-4}$ for all four variables.

Cost estimates

In order to compare the efficiency of different methods, it is necessary to make some estimate of the computational cost and also of the amount of computer storage needed. For N mesh points there are $4N$ variables and $20N$ coefficients; with N typically around 1000 to give reasonable accuracy, storage requirements are large. We therefore employed two estimates of cost as follows.

When implementing the pressure correction algorithm there are two choices for the handling of the linearized equations: the coefficients may be calculated once only and stored whenever possible, or they may be recalculated as needed. Our actual program followed the second approach; hence a comparison of the computing times of various methods is a measure of the relative efficiencies of the methods when storage is restricted. In order to reflect the situation when storage is unrestricted, we also evaluated the computation required in terms of a standard work unit, equal to the number of arithmetic operations per point for one double sweep of the ALGS algorithm, assuming storage of the coefficients. The work units for the various processes within the iteration, for the basic method and all the other methods discussed in this paper, can be found in Reference 6. When calculating work units, boundary effects on the grid operations are neglected.

Tests for the case $\Omega = 1{\cdot}0$ were carried out on 17×17 and 33×33 Chebyshev grids. In these tests we did not seek to optimize the under-relaxation parameters to any fine degree, our aim being to get a general impression of the behaviour. The values of 0·5 for the parameters α_u, α_v, α_w were those used by Chew,[2] and higher values failed to give convergence. In testing for the size of α_p, increments of 0·1 only were considered.

Table I gives the best performances (in terms of work units and CPU times) of the basic method on the two grids above. Note that the work units for each grid are based on the number of operations per point for that grid.

For both grids convergence was not obtained for higher values of α_p. For the 17×17 Chebyshev grid convergence was not obtained for $\alpha_p = 0{\cdot}2$ when fewer than 40 double sweeps for pressure correction were carried out.

3. MODIFICATIONS TO HANDLE THE COUPLING OF THE EQUATIONS

While the use of small values of α_p is necessary to prevent divergence in the early stages of the iteration, later stages show a very slow steady convergence. We therefore turn our attention to the coupling, which is the main factor which slows up convergence, and try to improve on the linearization used. For high rotation rates the dominant terms in the radial momentum equation in the main core of the region are $-\partial p'/\partial r$ and $2\rho\Omega v$. So although we are trying to find the radial velocity from the radial momentum equation, the dominant terms for that equation are the pressure gradient and the rotation term involving tangential velocity. We thus seek modifications to reflect the strong coupling of the radial and tangential momentum equations—ideally we would carry out a simultaneous Newton solution for these two equations, but this would be expensive.

Gosman *et al.*[7] incorporate a modification to the radial momentum equation of the form

$$\alpha(\rho/r)(\Omega r + v)(u_{\text{old}} - u_{\text{new}}), \tag{17}$$

motivated by the remark that if u increases, a decrease in v is expected, requiring a reduction in the

Table I. Performance, and parameters used, for the basic pressure correction method for the case $\Omega = 1{\cdot}0$

Grid	α_u, α_v, α_w	α_p	Number of ALGS sweeeps for pressure correction	Iterations to convergence	Approximate work units	CPU seconds (CDC 176)
17×17	0·5	0·2	40	184	10000	162·6
33×33	0·5	0·1	10	195	7190	370·2

centrifugal force term in the radial momentum equation. We now analyse the effect of this term and discuss the value that should be chosen for the constant α. We then present numerical results to support the relation between α and the rotation rate.

Let us first consider the linearization of the radial and tangential momentum equations used in the method of Section 2. Equations (1) and (2) may be rewritten in the form

$$\mu Lu - \frac{\rho}{r}\frac{\partial}{\partial r}(ru^2) - \rho\frac{\partial}{\partial z}(uw) + 2\rho\Omega v + \rho\frac{v^2}{r} = \frac{\partial p'}{\partial r}, \tag{18}$$

$$\mu Lv - \frac{\rho}{r}\frac{\partial}{\partial r}(ruv) - \rho\frac{\partial}{\partial z}(vw) - 2\rho\Omega u - \rho\frac{uv}{r} = 0, \tag{19}$$

where L is the linear operator given by

$$L = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{\partial^2}{\partial z^2} - \frac{1}{r^2}.$$

Let u^k, v^k be old iterates and u^{k+1}, v^{k+1} the new iterates given by $u^{k+1} = u^k + \delta u$, $v^{k+1} = v^k + \delta v$, and assume that the axial velocity w remains fixed. Equation (18) is then linearized for solution as

$$\mu Lu^{k+1} - \frac{\rho}{r}\frac{\partial}{\partial r}(ru^{k+1}u^k) - \rho\frac{\partial}{\partial z}(u^{k+1}w) + 2\rho\Omega v^k + \rho\frac{(v^k)^2}{r} = \frac{\partial p'}{\partial r}. \tag{20}$$

Assuming that we solve the radial momentum equation before the tangential equation, we have two choices for the linearized tangential momentum equation: either

$$\mu Lv^{k+1} - \frac{\rho}{r}\frac{\partial}{\partial r}(ru^k v^{k+1}) - \rho\frac{\partial}{\partial z}(v^{k+1}w) - 2\rho\Omega u^k - \rho\frac{u^k v^{k+1}}{r} = 0 \tag{21}$$

or

$$\mu Lv^{k+1} - \frac{\rho}{r}\frac{\partial}{\partial r}(ru^{k+1}v^{k+1}) - \rho\frac{\partial}{\partial z}(v^{k+1}w) - 2\rho\Omega u^{k+1} - \rho\frac{u^{k+1}v^{k+1}}{r} = 0. \tag{22}$$

Let us consider a simultaneous full Newton iteration for the radial and tangential momentum equations (18) and (19) respectively as follows:

$$\mu Lu^{k+1} - \frac{\rho}{r}\frac{\partial}{\partial r}\{r[(u^k)^2 + 2u^k\delta u]\} - \rho\frac{\partial}{\partial z}(u^{k+1}w) + 2\rho\Omega v^{k+1} + \frac{\rho}{r}[(v^k)^2 + 2v^k\delta v] = \frac{\partial p'}{\partial r}, \tag{23}$$

$$\mu Lv^{k+1} - \frac{\rho}{r}\frac{\partial}{\partial r}(ru^{k+1}v^{k+1}) - \rho\frac{\partial}{\partial z}(v^{k+1}w) - 2\rho\Omega u^{k+1} - \rho\frac{u^{k+1}v^{k+1}}{r} = 0. \tag{24}$$

The terms present in the full linearization of the radial momentum equation (23) but neglected in (20) are

$$\frac{2\rho}{r}(\Omega r + v^k)\delta v - \frac{\rho}{r}\frac{\partial}{\partial r}(ru^k\delta u). \tag{25}$$

Similarly the terms present in the full linearization of the tangential momentum equation (24) but neglected in (21) are

$$-\frac{\rho}{r}(2\Omega r + v^k)\delta u - \frac{\rho}{r}\frac{\partial}{\partial r}(rv^k\delta u). \tag{26}$$

We note that the form (22) of the tangential momentum equation corresponds exactly to the Newton linearization (24).

By using a relationship between δu, δv, the neglected terms (25), (26) may be included as a modification to the previously used linearization while retaining uncoupled equations. This relationship can be obtained by considering the maintenance of continuity in the direction of the rotation. While the axisymmetry of the problem implicitly forces rotational continuity, we take the viewpoint that calculating the next u-iterate from the linearized radial momentum equation, using old iterates, for v, leads to a violation of rotational continuity.

Consider the full continuity equation, retaining the θ derivative in line with the above comment, rewritten in the form

$$u+r\frac{\partial u}{\partial r}+\frac{\partial v}{\partial \theta}+r\frac{\partial w}{\partial z}=0. \tag{27}$$

Replacing the derivatives in (27) by small increments gives

$$u+r\frac{\delta u}{\delta r}+\frac{\delta v}{\delta \theta}+r\frac{\delta w}{\delta z}=0.$$

If the velocity w is taken to be fixed, this gives the equation

$$u+r\frac{\delta u}{\delta r}+\frac{\delta v}{\delta \theta}=0. \tag{28}$$

We have previously stated that it is in the core region that the rotational terms are dominant, so it is in this region that we aim to improve the handling of the coupling between the radial and tangential momentum equations. Restricting our attention to the core region, we make the further assumption that $u\simeq 0$ which, included in (28), gives

$$\delta v\simeq -r\frac{\delta \theta}{\delta r}\delta u.$$

With $\delta\theta = K\Omega$ for some constant K, we then have

$$\delta v\simeq -\frac{r\Omega}{\delta r}K\delta u. \tag{29}$$

Using the relationship (29) in the expression (23) gives the following modification to the radial momentum equation:

$$-\alpha\frac{\rho}{r}(\Omega r+v^k)\delta u-\frac{\rho}{r}\frac{\partial}{\partial r}(ru^k\delta u), \tag{30}$$

with

$$\alpha = 2r\Omega K/\delta r. \tag{31}$$

For large Ω we can neglect the second term in the expression (30), giving us the Gosman modification (17) with parameter α given by equation (31). We note that because of the sign of the term in (26), a similar modification to equation (21) would lead to a loss of diagonal dominance in the discretized equation. However, as noted above, the use of the form (22) for the tangential momentum equation requires no additional terms.

The Gosman modification (17) represents an improved linearization of the radial momentum equation with a correction that gives improved continuity in the θ-direction. The above discussion suggests that the parameter α should be a function of r. However, this would make the procedure more complicated and our tests have been based on a constant α.

Table II. Performance, and parameters used, for the pressure correction method modified by term (17), for $\Omega = 1.0$

Grid	α_p	α	Number of ALGS sweeps for pressure correction	Iterations to convergence	Approximate work units	CPU seconds (CDC 176)
17×17	0·3	25·0	10	66	1640	20·1
33×33	0·1	75·0	10	168	4170	210·7

Numerical results confirmed that the best value of the parameter α in (17) should be increased as the mesh size decreases. 'Optimum' values for the 17×17 and 33×33 Chebyshev grids at rotation rate $\Omega = 1{\cdot}0$ were found to be 25·0 and 75·0 respectively.

Numerical testing was also undertaken to support the linear variation of the 'optimum' value of α in (17) with rotation rate as predicted by (31); these tests used the non-linear multigrid algorithm of Reference 8. For rotation rates 1·0, 10·0, 20·0, 30·0 the optimum values of α were found to be 25·0, 100·0, 180·0, 260·0 respectively. Another important feature shown up by the tests was that the use of values of α smaller than the optimum values required greater under-relaxation of the pressure correction step to maintain convergence, while values higher than the optimum merely gave slower convergence.

Table II shows the effect of including the modification (17) in the radial momentum equation. The values of α used for the two grids were found to be 'optimal' from numerical experiments and the values of α_p were those which gave most efficient convergence. The values of α_u, α_v, α_w were as in Table I.

Comparing Table II with Table I clearly shows a large gain in efficiency by including the Gosman term (17) with a suitable value of α.

We now take as our standard method the basic algorithm modified by the Gosman term (17), and for the case $\Omega = 1{\cdot}0$ the optimum parameter values given in Table II.

4. MULTIGRID METHOD FOR SOLVING THE PRESSURE CORRECTION EQUATION

In this section we consider the use of a multigrid algorithm for the solution of the pressure correction equation. Attention is restricted to the pressure correction equation, since it is only for this equation that a significant accuracy is required at each major step in order that the non-linear iteration should converge.

For a complete description of multigrid ideas and algorithms see References 9 and 10. We give a brief outline of the basic multigrid principles and algorithms to be used. Efficient multigrid algorithms arise from the interaction between the smoothing properties of a relaxation method, such as the alternating-line Gauss–Seidel iteration, and coarse grid correction. By local Fourier analysis it can be shown that relaxation methods, without convergence acceleration parameters, are efficient at reducing the amplitude of high-frequency components of the error, and thus of the algebraic residual, while their ultimate slow convergence is due to the low-frequency components corresponding to the largest eigenvalues of the iteration operator. Relaxation methods can thus be viewed as efficient smoothers.

The principle of coarse grid correction can be seen by considering the specific problem of solving the pressure correction equation, which is a linear equation. Let the grid for pressure be R_h and let

$G(R_h)$ be the space of all grid functions on R_h. Then the pressure correction equation can be written in the form

$$L_h pp_h = f_h(R_h), \tag{32}$$

where $pp_h, f_h \in G(R_h)$, L_h: $G(R_h) \to G(R_h)$ and we assume L_h^{-1} to exist. Let pp_h^j be our current approximation to pp_h and define the algebraic defect (or residual) as

$$d_h = f_h - L_h pp_h^j. \tag{33}$$

Then the exact solution pp_h is given by

$$pp_h = pp_h^j + q_h^j,$$

where q_h^j is the solution of the defect equation

$$L_h q_h^j = d_h^j. \tag{34}$$

If the high-frequency components of the defect are negligible relative to the low-frequency components, we can represent equation (34) on a coarser grid, R_H. The exact definition of 'low' and 'high' frequencies depends on the relationship between the two grids—a high-frequency component is one which cannot be represented on the coarse grid. We thus get the coarse grid equation

$$L_H \hat{q}_H^j = d_H^j, \tag{35}$$

where $\dim(G(R_H)) \ll \dim(G(R_h))$, L_H: $G(R_H) \to G(R_H)$ and we assume L_H^{-1} to exist. If we solve equation (35), we can interpolate q_H^j to the finer grid, giving an approximation $\hat{q}_h^j$ to q_h^j and taking $pp_h^{j+1} = pp_h^j + \hat{q}_h^j$ as our new approximation to pp_h.

The approximation to the solution on the fine grid should be smooth enough to allow the defect equation to be represented on a coarse grid. This criterion is satisfied by using a suitable relaxation method before transferring the defect equation to the coarser grid.

The idea may then be extended to the solution of the coarse grid equation (35), leading to a series of coarser and coarser grids. Thus the complete multigrid method combines the use of a relaxation method on each grid with correction on coarser grids. Neither process alone gives a satisfactory method; it is only when a suitable combination of the two methods is used that very good convergence and efficiency properties can be obtained.

The algorithm used for the problem of Section 1 was based on a cycling correction scheme.[9,10] The use of non-uniform grids meant that some of the multigrid components needed to be modified. In particular, the grid coarsening was done by taking every other fine grid line as a coarse grid line; this coarsening means that the first and last grid lines remain adjacent to the boundaries and we use numbers of grid lines of the form 2^{n+1}.

The component multigrid operators were as follows: relaxation—alternating-line Gauss–Seidel; restriction—half-weighting operator modified to take account of the non-uniform grid;[10] interpolation—fine grid corrections and initial approximations were obtained using bilinear interpolation. A higher-order interpolation for the initial fine grid approximations was not used because of the increased cost implied by the non-uniform grid. Two types of multigrid algorithm were used, dependent on the initial approximation used on the finest grid: the FMG version used an initial approximation interpolated from a coarse grid solution obtained by multigrid iterations on the coarser grids; the MG version took the initial fine grid values corresponding to solid body rotation as in Section 2. For the $\Omega = 1{\cdot}0$ test case the FMG version of the multigrid algorithm proved to be more efficient than the MG version (in terms of the overall cost to convergence), the additional cost of calculating an initial fine grid approximation being compensated for by

increased accuracy in the pressure correction solution and a reduction in the number of outer iterations required. However, for higher-Reynolds-number flows, with thinner boundary layers, starting the multigrid solution process on the coarsest grid can lead to bad initial solutions on the finest grid, making it necessary either to increase the number of relaxation sweeps before and after coarse grid correction or to increase the amount of work on the coarse grids. In this case it is more efficient to start the calculation on the finest grid (the MG mode).

Table III gives the performance of the pressure correction method with and without the use of the multigrid algorithm, for the test case $\Omega = 1{\cdot}0$ and for the two previously considered grids. For both grids the multigrid algorithm used three grids, V-cycles, one ALGS sweep before and after coarse grid correction and two ALGS sweeps on the coarsest grid, this scheme having been found to be most efficient in terms of the overall work units and CPU times of the outer iteration. The under-relaxation parameters and values of α (in equation (17)) for both grids are exactly as in Table II.

Table III shows a clear advantage in employing the multigrid algorithm, the reduction in CPU time being approximately 41% for both grids.

5. EXTENDED PRESSURE CORRECTION SCHEMES

We now consider three separate schemes which aim to improve the pressure correction method via the handling of the pressure correction equation.

Method I

The first method is that proposed by Connell and Stow.[11] An iterative scheme is introduced with the aim of approximately solving the full pressure correction equation, given by using not the truncated velocity–pressure relations (15), (16) but the full relations (13), (14).

The iteration replacing steps (v), (vi) in the basic algorithm of Section 2 is as follows:

(1) using the usual pressure correction equation obtained from the truncated relations (13), (16), calculate a pressure correction pp^0;
(2) For $j = 1, n_2$
(a) calculate δu^{j-1}, δw^{j-1} from the truncated relations

$$a_p \delta u_p^{j-1} = A_p(pp_s^{j-1} - pp_n^{j-1}), \tag{36}$$

$$c_p \delta w_p^{j-1} = C_p(pp_w^{j-1} - pp_e^{j-1}), \tag{37}$$

Table III. Comparison of ten ALGS sweeps with the multigrid algorithm as the linear solver for pressure correction, for $\Omega = 1{\cdot}0$

Grid	Linear solver for pressure correction	Iterations to convergence	Approximate work units	CPU seconds (CDC 176)
17×17	10 ALGS sweeps	66	1640	20·1
	FMG	62	1220	11·7
33×33	10 ALGS sweeps	168	4170	210·7
	FMG	164	3240	123·6

(b) calculate an improved pressure correction pp^j from the approximate full relations

$$a_p \delta u_p^j = \sum_i a_i \delta u_i^{j-1} + A_p(pp_s^j - pp_n^j), \tag{38}$$

$$c_p \delta w_p^j = \sum_i c_i \delta w_i^{j-1} + C_p(pp_w^j - pp_e^j), \tag{39}$$

with δu_p^j, δw_p^j eliminated by using continuity;

(3) Update u, w, p by

$$u^{k+1} = u^* + \delta u^{n_2 - 1},$$
$$w^{k+1} = w^* + \delta w^{n_2 - 1}$$
$$p^{k+1} = p^k + \alpha_p pp^{n_2}.$$

Connell and Stow[11] use a fixed number n_1 of ALGS sweeps for each of the pressure correction solutions (steps 1 and 2b above). In later results we also use the multigrid algorithm for these solutions.

Initial numerical tests with the algorithm modified as above were undertaken on the 17×17 grid with under-relaxation parameters and value of α given by the optimum values for the case $\Omega = 1{\cdot}0$ as in Section 2. Our aim was to investigate the overall convergence of method I with respect to the two parameters n_1, n_2. Table IV shows the results of these tests.

The results in Table IV suggest the choice of parameters $n_1 = 4$, $n_2 = 1$. This combination was used in later tests with the ALGS algorithm as linear solver for the pressure correction; the choice $n_2 = 1$ was also used when employing the multigrid algorithm. Taking $n_2 = 3$ gave divergence in most cases.

Using the values $n_1 = 4$, $n_2 = 1$ it was found that under-relaxation of the pressure correction, on both 17×17 and 33×33 grids, was no longer necessary, i.e. a value of $\alpha_p = 1{\cdot}0$ could be used. However, for the 33×33 grid a value of $\alpha_p = 0{\cdot}5$ was found to give faster convergence. Just as a large number of ALGS sweeps was no longer required for the pressure correction solutions, the greater accuracy provided by the FMG mode, rather than the MG mode, of the multigrid algorithm was no longer required. Thus the MG mode proved to be more efficient in terms of the overall cost to achieve convergence.

Table IV. Performance of method I; 17×17 grid, $\Omega = 1{\cdot}0$

		n_2	
n_1		1	2
1	Iterations to convergence	189	130
	Work units	3750	3120
2	Iterations to convergence	112	77
	Work units	2440	2080
3	Iterations to convergence	85	66
	Work units	2030	1980
4	Iterations to convergence	76	63
	Work units	1960	2080
5	Iterations to convergence	73	62
	Work units	2030	2230

Table V gives the best performances of method I, using both the ALGS algorithm and the multigrid algorithm as linear solvers for the pressure correction for the case $\Omega = 1{\cdot}0$.

Table V shows up clearly the importance of storage availability when assessing the relative efficiencies of two methods. If we are in the situation where storage is unrestricted, then by considering work units we can see:

(a) there is only a marginal gain in using the multigrid algorithm to replace the four ALGS sweeps;
(b) method I is more efficient than the basic method (cf. Table III).

However, if storage is restricted, comparing CPU times from Tables III and V we see that method I, without the multigrid algorithm, is now less efficient than the basic method.

Method II

Method II is the modification to the pressure correction method proposed by Van Doormaal and Raithby.[12] Instead of the truncated pressure–velocity relations (15), (16) they introduce an approximation to the full relations as follows.

If we subtract $\Sigma_i a_i \delta u_p$ from both sides of equation (13) and subtract $\Sigma_i c_i \delta w_p$ from both sides of equation (14), we get

$$\left(a_p - \sum_i a_i\right)\delta u_p = \sum_i a_i(\delta u_i - \delta u_p) + A_p(pp_s - pp_n),$$

$$\left(c_p - \sum_i c_i\right)\delta w_p = \sum_i c_i(\delta w_i - \delta w_p) + C_p(pp_w - pp_e).$$

Van Doormaal and Raithby[12] argue that the changes δu_p, δu_i and δw_p, δw_i will be of the same order, so that ignoring the terms

$$\sum_i a_i(\delta u_i - \delta u_p), \qquad \sum_i c_i(\delta w_i - \delta w_p)$$

will lead to a consistent approximation to the full relations (13), (14).

When obtaining the pressure correction equation, they use the relations

$$\left(a_p - \sum_i a_i\right)\delta u_p = A_p(pp_s - pp_n), \tag{40}$$

$$\left(c_p - \sum_i c_i\right)\delta w_p = C_p(pp_w - pp_e). \tag{41}$$

Table V. Performance of method I; $\Omega = 1{\cdot}0$

Grid	Linear solver for pressure correction	α_p	Iterations to convergence	Approximate work units	CPU seconds (CDC 176)
17×17	4 ALGS sweeps	1·0	39	1000	28·1
	MG	1·0	37	950	10·7
33×33	4 ALGS sweeps	0·5	107	2760	319·1
	MG	0·5	98	2520	115·1

Here we note that Van Doormaal and Raithby[12] assume that under-relaxation is implemented inside the linear solver rather than in the way described in Section 2. This is of great practical importance for the axial momentum equation, where we have $c_p = \Sigma_i c_i$ which would make δw_p infinite.[4] Method II was thus implemented using the damped relations

$$\left(a_p - \alpha_u \sum_i a_i\right)\delta u_p = A_p(pp_s - pp_n), \tag{42}$$

$$\left(c_p - \alpha_w \sum_i c_i\right)\delta w_p = C_p(pp_w - pp_e). \tag{43}$$

Although Van Doormaal and Raithby state that the pressure correction should not be under-relaxed, this leads to divergence for the case $\Omega = 1{\cdot}0$ (with α_u, α_v, α_w, α as in Tables I and II). However, less accuracy was required for the solution of the pressure correction equation, so that in terms of overall cost it was more efficient to use just three ALGS sweeps, or in the case of the multigrid algorithm the MG rather than FMG mode. Table VI gives the best performances of method II for the $\Omega = 1{\cdot}0$ case using the 17×17 and 33×33 grids and values of α_u, α_v, α_w, α as in Tables I and II.

Table VI shows a reduction in both work units and CPU time to convergence over the basic method (of Table III), this reduction being at the cost of very little change to the program. The benefit in using the multigrid algorithm can also be seen here.

Method III

Method III is the revised pressure correction method of Patankar.[5] The essence of the method is that the previously derived pressure correction equation is used to correct the velocities, but a separate pressure equation is used to calculate the pressure before the solution of the momentum equations. The advantage of this scheme is that there is no need to use a guessed pressure; if the exact velocities were provided as starting values, the exact pressure would be calculated and the iterative process would not move away from the correct solution, as happens with the normal pressure correction method.

The pressure equation is derived from the momentum and continuity equations as follows. Defining variables $\hat{u}$, $\hat{w}$ by

$$\hat{u}_p = \frac{\sum_i a_i u_i + S_u}{a_p}, \tag{44}$$

$$\hat{w}_p = \frac{\sum_i c_i w_i + S_w}{c_p}, \tag{45}$$

Table VI. Performance of method II (Van Doormaal and Raithby terms); $\Omega = 1{\cdot}0$

Grid	Linear solver for pressure correction	α_p	Iterations to convergence	Approximate work units	CPU seconds (CDC 176)
17×17	3 ALGS sweeps	0·7	66	1200	11·3
	MG	0·7	54	1010	9·4
33×33	3 ALGS sweeps	0·3	162	2940	112·5
	MG	0·3	115	2160	78·0

the momentum equations (5), (7) can be rewritten in the form

$$a_p u_p = a_p \hat{u}_p + A_p(p_s - p_n), \tag{46}$$

$$c_p w_p = c_p \hat{w}_p + C_p(p_w - p_e). \tag{47}$$

Equations (46), (47) have exactly the same form as the truncated relations (15), (16), so that substituting into the continuity equation will give an equation for pressure with the same coefficients as the pressure correction equation. The difference between the two equations occurs in the right-hand side: the pressure correction equation depends on u^*, w^*; the pressure equation depends on $\hat{u}$, $\hat{w}$.

The basic algorithm is then modified as follows:

(a) before step (ii) calculate $\hat{u}$, $\hat{w}$, solve the pressure equation and update the pressure;
(b) in step (vi) only u, w are updated.

Numerical tests using method III, at rotation rate $\Omega = 1{\cdot}0$, showed up several features:

(a) Under-relaxation of the pressure equation was unnecessary and was less efficient.
(b) A gain in efficiency could be achieved by using three rather than ten ALGS sweeps for the solution of the equations for pressure and pressure correction.
(c) While a good initial approximation to the fine grid pressure correction may be obtained by multigrid iterations on coarser grids (the FMG mode), this cannot be done satisfactorily for the pressure, which varies too much to be represented well on the coarse grids. Thus the multigrid solution of the pressure equation started on the fine grid with the latest pressure iterate.

The VII gives the best performances of method III for the $\Omega = 1{\cdot}0$ case using 17×17 and 33×33 grids, α_u, α_v, α_w, α as in Tables I and II and with $\alpha_p = 1{\cdot}0$.

Comparisons with the other extended methods are unfavourable. For both grids the work units are increased from those achieved by method I and the CPU times are greater than those of method II.

6. CONCLUSIONS

The inclusion of the term (17) in the basic pressure correction method, to improve the handling of the rotation terms, has been shown to give a great improvement in the convergence of the pressure correction method, provided a suitable value for the parameter α in (17) can be chosen.

Using the multigrid algorithm as a linear solver for the pressure correction leads to a gain in efficiency of around 40% in CPU time.

Table VII. Performance of method III, Patankar's revised method; $\alpha_p = 1{\cdot}0$

Grid	Linear solver for pressure and pressure correction	Iterations to convergence	Approximate work units	CPU seconds (CDC 176)
17×17	3 ALGS sweeps	52	1190	13·2
	Multigrid	42	1080	11·8
33×33	3 ALGS sweeps	137	3070	137·2
	Multigrid	120	3080	135·4

Of the extended methods it is method II, combined with the use of the multigrid algorithm, which gives the best improvements in overall efficiency. In comparing methods, the importance of storage strategy in assessing the efficiency has been shown. This is because of the amount of work involved in recomputing coefficients when storage is limited. If storage is unrestricted, method I without the multigrid algorithm gives a significant gain in efficiency over the basic method; however, if storage is restricted, the same method is less efficient than the basic method. A further point is worth bearing in mind when comparing the extended methods. Methods I and III both appear to be relatively insensitive to the choice of α_p, and for the $\Omega = 1{\cdot}0$ case neither method required under-relaxation in the pressure correction for convergence, i.e. $\alpha_p = 1{\cdot}0$ could be used.

ACKNOWLEDGEMENTS

The first author would like to acknowledge the support of the Science and Engineering Research Council and of Rolls-Royce Ltd, Derby through the CASE Scheme.

REFERENCES

1. S. V. Patankar and D. B. Spalding, 'A calculation procedure for heat and mass transfer in three-dimensional parabolic flows', *Int. J. Heat Mass Transfer*, **15**, 1787–1806 (1972).
2. J. W. Chew, 'Development of a computer program for the prediction of flow in a rotating cavity', *Int. j. numer. methods fluids*, **4**, 667–683 (1984).
3. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, 1962.
4. G. Lonsdale, 'Multigrid methods for the solution of Navier–Stokes equations', *Ph.D. Thesis*, Manchester University, 1985.
5. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, 1980.
6. G. Lonsdale and J. E. Walsh, 'The pressure correction method, and the use of a multigrid technique, for laminar source–sink flow between corotating discs', *Numer. Anal. Tech. Rep. 95*, Manchester University, 1984.
7. A. D. Gosman, M. L. Koosinlin, F. C. Lockwood and D. B. Spalding, 'Transfer of heat in rotating systems', *ASME Publication 76-GT-25*, 1976.
8. G. Lonsdale, 'Solution of a rotating Navier–Stokes problem by a nonlinear multigrid algorithm', *Numer. Anal. Tech. Rep. 105*, Manchester University, 1985.
9. A. Brandt, 'Multigrid techniques: 1984 guide with applications to fluid dynamics', *GMD-Studien No. 85*, Bonn, West Germany, 1984.
10. K. Stüben and U. Trottenberg, 'Multigrid methods: Fundamental algorithms, model problem analysis, and applications', *Multigrid Methods. Proceedings*, Koln-Porz, 1981, in W. Hackbush and U. Trottenberg (eds), *Lecture Notes in Mathematics 960*, Springer-Verlag, Berlin, 1982, pp. 1–176.
11. S. D. Connell and P. Stow, 'The pressure correction method', *Comput. Fluids*, **14**, 1–10 (1986).
12. J. P. Van Doormaal and G. D. Raithby, 'Enhancements of the SIMPLE method for predicting incompressible fluid flows', *Numer. Heat Transfer*, **7**, 147–163 (1984).