INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 22, 353–373 (1996)

A COMPARISON OF COUPLED AND SEGREGATED ITERATIVE SOLUTION TECHNIQUES FOR INCOMPRESSIBLE SWIRLING FLOW

R. F. HANBY*

Thermofluids Section, Mechanical Engineering Department, Imperial College of Science, Technology and Medicine, Exhibition Road, London SW7 2AZ, U.K.

D. J. SILVESTER

Numerical Analysis Group, Department of Mathematics, UMIST, Manchester, U.K.

AND

J. W. CHEW

Aerothermal Methods, Rolls-Royce plc, U.K.

SUMMARY

In many popular solution algorithms for the incompressible Navier–Stokes equations the coupling between the momentum equations is neglected when the linearized momentum equations are solved to update the velocities. This is known to lead to poor convergence in highly swirling flows where coupling between the radial and tangential momentum equations is strong. Here we propose a coupled solution algorithm in which the linearized momentum and continuity equations are solved simultaneously. Comparisons between the new method and the well-known SIMPLEC method are presented.

KEY WORDS: Navier–Stokes equations; finite differences; unsymmetric linear systems; Krylov subspace methods

1. INTRODUCTION

The work described in this paper is partly motivated by the design engineer's need to calculate flow in the cavities formed between the rotating discs in gas turbine engines. These flows can be dominated by rotational effects and present difficulties for segregated solvers such as the SIMPLE algorithm[1] and its derivatives. In these methods the linearized momentum equations are solved successively without accounting for the coupling between equations. Such difficulties have been documented previously, e.g. in Reference 2. Various *ad hoc* measures have been proposed in order to improve convergence of the method (see e.g. References 3–5). Although it has been clearly demonstrated that the variations of the SIMPLE algorithm can give satisfactory solutions, the need to tune parameters for the iterative solution and the computational cost of the calculations still limit the application of the method. In the present work we propose a coupled solution of the linearized governing equations and compare the performance of the new method with the segregated SIMPLEC algorithm.[6] Whilst a coupled approach requires more core memory than a segregated approach, this is not a serious limitation on most current computers and it may offer advantages in terms of robustness, CPU time requirements and level of convergence achieved.

*Present address: Department of Mathematics, Statistics and Operational Research, Nottingham Trent University, Burton St, Nottingham NG1 4BU, UK

CCC 0271–2091/96/050353–21

Received 22 April 1994
Revised 19 April 1995

For incompressible flow there is an intrinsic difficulty in solving the governing equations, namely that the divergence-free constraint is effectively the 'equation for pressure'. Thus a discretized Navier–Stokes operator will always have a zero diagonal block, making iterative solution methods prone to divergence. (Typically, diagonal dominance is the key to rapid convergence of iterative methods.) The standard approach used to circumvent this difficulty is to uncouple the equations via a so-called segregated formulation. The divergence-free condition is replaced by a 'derived equation', usually a pressure Poisson equation. This projects a non-divergence-free velocity onto a divergence-free field as follows. Given an approximate velocity solution u, we construct a divergence-free velocity u^* by introducing a 'pressure' λ and solving the system

$$\text{grad}\,\lambda + u = u^*, \tag{1}$$

$$\text{div}\,u^* = 0. \tag{2}$$

Taking the divergence of (1) gives

$$\text{div grad}\,\lambda + \text{div}\,u = \text{div}\,u^* = 0. \tag{3}$$

Hence the pressure satisfies a Poisson equation

$$\nabla^2\lambda = -\text{div}\,u. \tag{4}$$

Substituting this value of λ into (1), the projected velocity u^* is divergence-free. The pressure Poisson equation (4) is the foundation of the popular SIMPLE class of algorithms.[1]

The projection (1)–(2) is also the basis of the transient flow algorithms proposed by Gresho[7] and Gresho and Chan,[8] which are in widespread use within the finite element community. A further point, discussed at length in Reference 7, is that the use of the derived equation for pressure (4) instead of the incompressibility constraint (2) brings a requirement that a pressure boundary condition be satisfied (or imposed) everywhere on the boundary of the flow domain. The Navier–Stokes equations only require that a hydrostatic pressure be specified. In practice a Neumann condition is commonly assumed.

A potentially more robust alternative to the SIMPLE philosophy is to retain the coupling in the equations in a fully implicit manner. Finite element codes for incompressible steady state flows (e.g. FIDAP[9]) typically take this approach, with the non-linearity handled via a Picard or Newton iteration. In general the fully implicit Jacobian systems are solved directly, often using a non-symmetric frontal solver. In this work the need for a good approach for three-dimensional problems motivated our use of *inexact* Newton-methods, where an exact Jacobian solve is replaced by a 'cheap' iteration strategy. Unfortunately, as we shall see later, the lack of diagonal dominance referred to above makes these fully coupled systems of equations extremely difficult to solve iteratively. An efficient robust non-symmetric iterative solver is needed and the issue of preconditioning is all-important. To this end our results complement those in References 10 and 11. It should be emphasized that using a direct solver for the Jacobian systems[12] may well have led to a more efficient implementation of our coupled algorithm for the two-dimensional discretizations considered herein.

A further alternative to the SIMPLE philosophy is the use of pseudo time-stepping. Typically a time-marching scheme will add a 'mass matrix' to the diagonal of the coefficient matrix. For sufficiently small time steps, diagonal dominance is achieved and the equations may be solved relatively easily. Although time-marching methods are not considered further here, we note that various degrees of coupling are possible within the time-stepping framework. Thus the present work has some relevance to this type of solution method.

The goal then is to compute steady state axisymmetric swirling flows with a finely tuned SIMPLEC method and compare its efficiency with a coupled solution algorithm. In Section 2 we present the

mathematical model and describe two test problems which we will use to compare algorithms. In Section 3 we describe a segregated method for our problems based on the SIMPLE philosophy. In Section 4 we present a coupled formulation of the model and in Section 5 we present results of numerical experiments in which we derive a reasonable set of parameters for our method. We compare the performance of the methods for our two representative flow examples in Section 6 and finally, in Section 7, present our conclusions.

1.1. Notation

Here u, v and w denote the continuous velocity components in the axial, radial and tangential directions respectively and p denotes the pressure. We discretize the continuous Navier–Stokes equations, transforming the continuous variable (u, v, w, p) to the discrete variables $(\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{p})$ or $(\mathbf{U}, \mathbf{p})$.

We use the equivalent notation

$$\begin{bmatrix} A & C \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$$

$$\begin{bmatrix} A_{uu} & 0 & 0 & C_u \\ 0 & A_{vv} & A_{vw} & C_v \\ 0 & A_{wv} & A_{ww} & 0 \\ B_u & B_v & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_v \\ \mathbf{f}_w \\ \mathbf{g} \end{bmatrix}$$

to denote the linear systems that our coupled solver generates.

2. MATHEMATICAL MODEL

We consider the system of axisymmetric steady incompressible Navier–Stokes equations in a cylindrical polar co-ordinate system (r, θ, z) as

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r u v) + \frac{\partial}{\partial z}(\rho u^2) = -\frac{\partial p}{\partial z} + \mu \nabla^2 u, \tag{5}$$

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r v^2) + \frac{\partial}{\partial z}(\rho u v) = -\frac{\partial p}{\partial r} + \mu\left(\nabla^2 v - \frac{v}{r^2}\right) + \rho\frac{w^2}{r}, \tag{6}$$

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r v w) + \frac{\partial}{\partial z}(\rho u w) = \mu\left(\nabla^2 w - \frac{w}{r^2}\right) - \rho\frac{v w}{r}, \tag{7}$$

$$\frac{\partial u}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(r v) = 0, \tag{8}$$

where we define the discrete differential operator

$$\nabla^2 \phi = \text{div grad } \phi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi}{\partial r}\right) + \frac{\partial^2 \phi}{\partial z^2},$$

ρ is the density (assumed constant) and μ is the dynamic viscosity (assumed constant). Velocities are specified via a Dirichlet condition at all points on the boundary.

In the segregated approach (described in Section 3) the pressure terms and cross-terms in the momentum equations (5)–(7) are treated semi-explicitly. Although the individual linear systems thus

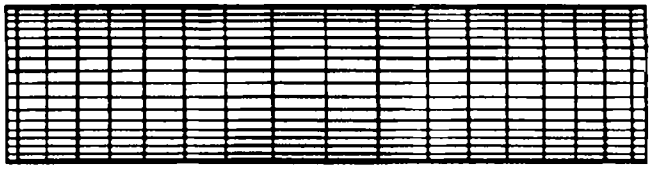


Figure 1. Expanding grid (17 × 17)

generated are easy to solve, there is no coupling between the equations, resulting in poor convergence of the non-linear iteration as the rotational rate increases. In our approach (described in Section 4) the terms on the right-hand side of (5)–(7) are treated implicitly, coupling the three momentum equations and the incompressibility constraint.

Our starting point for solving these equations is the computer program described in Reference 4. When the number of multigrid levels in this code is specified equal to one, the solution procedure reverts to the SIMPLEC algorithm. We use a finite volume approximation on orthogonal Cartesian grids with either a uniform or a non-uniform spacing. The problem domain is discretized on a rectangular two-dimensional grid making use of the axisymmetry of the problem. In the non-uniform case the grid spacing h expands away from the boundaries by a fixed factor of 1·2 (see Figure 1). This is because for high Reynolds number flows it would be computationally expensive to use a uniform grid fine enough to accurately capture the flow close to the walls of the cavity. A standard staggered grid discretization has been used, consistent with most earlier work, where axial and radial velocities are evaluated on cell faces and pressure and swirl velocities are evaluated at cell centres (see Figure 2).

The two test problems are now described.

2.1. *Problem 1: the rotating inner cylinder problem*

The general problem is the study of the flow of fluid contained in an annulus between two infinitely long concentric cylinders of radii r_1 and r_2 ($r_1 < r_2$) which are rotating about the same axis with angular velocities Ω_1 and Ω_2.[13] We consider the special case where the outer cylinder is fixed and only the inner cylinder rotates (with angular velocity $\Omega_1 = 1{\cdot}0$), i.e. $\Omega_2 = 0$. This problem has a rotational Reynolds number $Re_\theta = \Omega_1 r_2^2 \rho/\mu = 2{\cdot}2$ (see Figure 3).

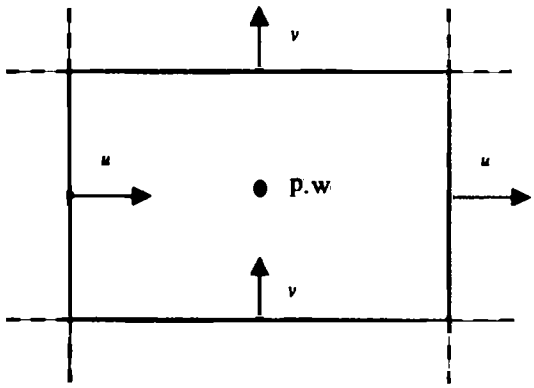


Figure 2. Pressure control volume showing staggered grid arrangement

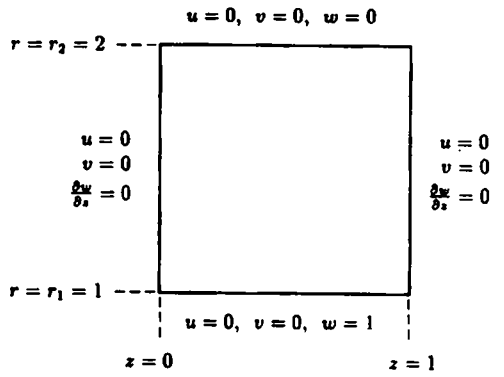


Figure 3. Problem 1—rotating inner cylinder problem (SI units)

The problem thus defined has the general analytic solution

$$u(r, z) = 0, \tag{9}$$

$$v(r, z) = 0, \tag{10}$$

$$w(r, z) = \frac{C_1}{r} + C_2 r, \tag{11}$$

$$p(r, z) = \tfrac{1}{2}C_2^2 r^2 + 2C_1 C_2 \log_e r - \frac{1}{2}\frac{C_1^2}{r^2} + P_c, \tag{12}$$

where P_c is a hydrostatic pressure constant and the constants C_1 and C_2 are defined as

$$C_1 = \frac{(\Omega_1 - \Omega_2) r_1^2 r_2^2}{r_2^2 - r_1^2}, \qquad C_2 = \frac{\Omega_2 r_2^2 - \Omega_1 r_1^2}{r_2^2 - r_1^2}.$$

The grid used in this test problem is uniform, since the Reynolds number of the flow is low.

2.2. *Problem 2: the rotating cavity problem*

The boundary conditions for this problem are shown in Figure 4. Here Ω is the angular velocity of the cavity. This has been varied to give rotational Reynolds numbers of 250, 2500 and $2{\cdot}5 \times 10^4$. There is a net radial outflow through the cavity corresponding to a non-dimensional mass flow parameter $C_w (= \dot{m}/\mu r_2)$ of 192·0. This and similar flows have been studied in some detail by Chew *et al.*[14] and adopted as benchmark test cases in several numerical studies, e.g. References 4 and 15.

Contours of the axisymmetric streamfunction from a solution of this problem at a Reynolds number of 2500 are shown in Figure 5. Various flow regions may be identified. On the discs and outer cylindrical shroud, boundary layers are formed. These are often referred to as Ekman-type and Stewartson-type layers respectively after analytical work by these authors. Away from the boundary layers a source region forms near the inner cylinder and a rotating core develops in the outer part of the cavity. In the source region the flow approximates to a free vortex. In the rotating core the radial and axial components of velocity vanish and the tangential component becomes independent of the axial co-ordinate (but varies radially). At higher Reynolds numbers the boundary layers become narrower, the radial extent of the source region is reduced and the rotating core occupies more of the cavity.

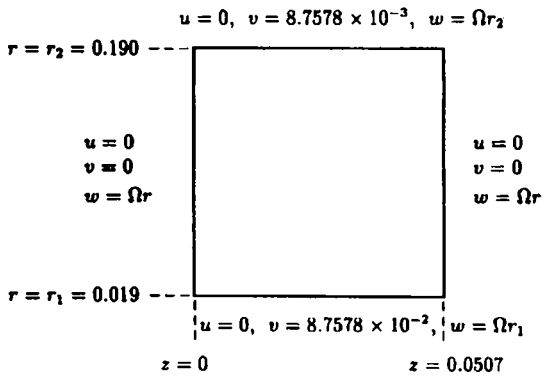


Figure 4. Problem 2—rotating cavity problem; $Re_\theta = 2{\cdot}5 \times 10^3$ (SI units)

Most of our numerical experiments are carried out at a Reynolds number of 2500 in order to avoid excessive computing times, although this flow is still sufficiently complicated to allow a useful assessment of our method. The grid used for the discretization in this example, as mentioned above, expands away from the walls of the domain by a factor of 1·2. Therefore under refinement we generate a non-nested set of grids.

3. A SEGREGATED APPROACH

We first define the block diagonal matrix

$$A_B = \begin{bmatrix} A_{uu} & 0 & 0 \\ 0 & A_{vv} & 0 \\ 0 & 0 & A_{ww} \end{bmatrix}.$$

The SIMPLE algorithm for axisymmetric steady incompressible flow in cylindrical polar co-ordinates is stated in Algorithm 1. Notice that in steps 3 and 4 we use the updated (starred) quantities as soon as they are available. Note also that underrelaxation may be applied at steps 2–4 and 7.

The variants of the SIMPLE algorithm obtain new values for the velocity and pressure fields (in steps 5 and 6) in different ways. For example, at step 6, whereas SIMPLE uses the diagonal matrix $\hat{A}_B = (A_B)$ (the diagonal of A_B), the SIMPLEC method uses the approximation

$$\hat{A}_B = \text{diag}(A_B) - \sum_{n,\,s,\,e,\,w} A_B,$$

where $\sum_{n,\,s,\,e,\,w} A_B$ denotes the diagonal matrix of the sums of the nearest neighbours to each element

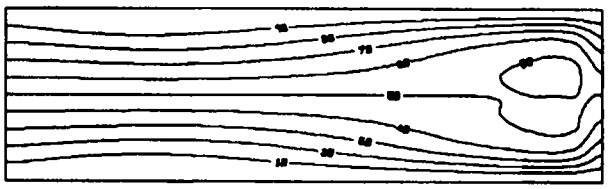


Figure 5. Problem 2—solution for u–v streamfunction (percentages of maximum streamfunction)

Algorithm 1. The SIMPLE algorithm

1. Guess initial velocity and pressure fields $\mathbf{u}_0$, $\mathbf{v}_0$, $\mathbf{w}_0$ and $\mathbf{p}_0$.

   For $i = 0, 1, \ldots,$ do:

   2. Solve for $\mathbf{u}^*$:

      $$\frac{1}{r}\frac{\partial}{\partial r}(\rho r \mathbf{v}_i \mathbf{u}^*) + \frac{\partial}{\partial z}(\rho \mathbf{u}_i \mathbf{u}^*) - \mu \nabla^2 \mathbf{u}^* = -\frac{\partial \mathbf{p}_i}{\partial z}.$$

   3. Solve for $\mathbf{v}^*$:

      $$\frac{1}{r}\frac{\partial}{\partial r}(\rho r \mathbf{v}_i \mathbf{v}^*) + \frac{\partial}{\partial z}(\rho \mathbf{u}^* \mathbf{v}^*) - \mu\left(\nabla^2 \mathbf{v}^* - \frac{\mathbf{v}^*}{r^2}\right) = -\frac{\partial \mathbf{p}_i}{\partial r} + \rho\frac{\mathbf{w}_i^2}{r}.$$

   4. Solve for $\mathbf{w}^*$:

      $$\frac{1}{r}\frac{\partial}{\partial r}(\rho r \mathbf{v}^* \mathbf{w}^*) + \frac{\partial}{\partial z}(\rho \mathbf{u}^* \mathbf{w}^*) - \mu\left(\nabla^2 \mathbf{w}^* - \frac{\mathbf{w}^*}{r^2}\right) + \rho\frac{\mathbf{v}^* \mathbf{w}^*}{r} = 0.$$

   5. Obtain corrections for the pressure field $\mathbf{p}'$ using the pressure Poisson (pressure correction) equation

      $$\text{div } \hat{A}_B^{-1} \text{ grad } \mathbf{p}' = -\text{div } \mathbf{U}^*,$$

      where $\hat{A}_B$ is an approximation to A_B which will be more fully explained shortly.

   6. Obtain corrections for the velocity field $\mathbf{U}'$ using $\mathbf{p}'$ and the equation

      $$\mathbf{U}' = \hat{A}_B^{-1} \text{ grad } \mathbf{p}'.$$

   7. Update the velocity and pressure fields using the relations

      $$\mathbf{U}_{i+1} = \mathbf{U}^* + \mathbf{U}', \qquad \mathbf{p}_{i+1} = \mathbf{p}_i + \mathbf{p}'.$$

      If converged, then stop.

on the computational grid. The SIMPLER algorithm uses the pressure correction matrix to correct the velocities, but a separate pressure equation is used to calculate the pressure before the solution of the momentum equations. Note that since $\hat{A}_B$ is always a diagonal matrix, it is trivial to invert. The amount of underrelaxation required varies with the problem to be solved, but generally it increases as the Reynolds number of the problem increases. The key to the success of the process is that the small linear systems in steps 2–5 need not be solved exactly. We address each of these types of equation below.

3.1. The momentum equations

These systems are 'easily' solved if two conditions are satisfied.

1. A hybrid finite differencing scheme[16] is used, which employs second-order central differencing on both convection and diffusion terms but automatically modifies the convective differencing procedure to an upwind scheme when the local cell Reynolds number exceeds two. This ensures that the discrete operator is diagonally dominant at high ($\geqslant 2$) cell Reynolds numbers, although in this case only first-order accuracy is obtained.
2. A relaxation method is applied which follows the flow direction. In practice this can be difficult, since typical problems have no overall flow direction (e.g. a circulating flow). In such cases an alternating line relaxation method is usually a good choice of solver. (For example, here we typically use two to four alternating line Gauss–Seidel sweeps per linear solve.)

3.2. *The pressure correction equation*

This is a standard elliptic equation and would be symmetric positive definite except for the assumed Neumann boundary conditions for pressure. In this case the pressure Poisson matrix has one zero eigenvalue. Even though the system is singular, it can be effectively solved using relaxation methods as above or suitably preconditioned conjugate gradient (CG) or conjugate residual (CR) methods.[17] Here the alternating line Gauss–Seidel method is again used.

3.3. *Gosman et al.'s modification*

In the case of highly swirling flow there may be regions in which the centripetal force is balanced solely by the radial pressure gradient. In this case the dominant terms in the radial momentum equation (6) in the majority of the computational domain are

$$-\frac{\partial \mathbf{p}}{\partial r} \quad \text{and} \quad \rho\frac{\mathbf{w}^2}{r}, \tag{13}$$

i.e. the pressure gradient and the rotation term involving tangential velocity. However, we are solving this equation for the radial velocity v. We would therefore like to modify our equations to overcome the problem of the strong coupling between the radial and tangential momentum equations.

If we make the assumption that as the radial velocity component increases we observe a proportional decrease in the corresponding tangential velocity, the second term in (13) can be written as

$$\frac{\rho \mathbf{w}_i^2}{r} - \frac{\alpha\rho}{r}\mathbf{w}_i(\mathbf{v}^* - \mathbf{v}_i), \tag{14}$$

where α is a constant referred to below as the Gosman factor.[3]

Lonsdale and Walsh[15] demonstrate the effectiveness of the Gosman modification. They show that with a suitable choice of α, solution times can be at least halved compared with the unmodified SIMPLEC iteration.

4. A COUPLED APPROACH

In our approach the non-linear cross-terms involving w on the right-hand side of equations (6) and (7) are linearized and treated implicitly using a Newton-type scheme in which products of variables are represented by

$$\mathbf{x}_{i+1}\,\mathbf{y}_{i+1} = \mathbf{x}_{i+1}\,\mathbf{y}_i + \mathbf{x}_i\,\mathbf{y}_{i+1} - \mathbf{x}_i\,\mathbf{y}_i,$$

where the index i refers to the non-linear iteration number. In addition, pressure derivative terms are treated implicitly. We state our coupled algorithm in Algorithm 2.

Thus the iteration gives rise to a coefficient matrix that can be expressed in the block form

$$\begin{bmatrix} A_{uu} & 0 & 0 & C_u \\ 0 & A_{vv} & A_{vw} & C_v \\ 0 & A_{wv} & A_{ww} & 0 \\ B_u & B_v & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_v \\ \mathbf{f}_w \\ \mathbf{g} \end{bmatrix} = \begin{bmatrix} 0 \\ -\rho\mathbf{w}_i^2/r \\ (\rho/r)\mathbf{v}_i\mathbf{w}_i \\ 0 \end{bmatrix}, \tag{15}$$

which must be solved at each outer non-linear iteration. In the case of a non-staggered grid formulation in a Cartesian co-ordinate system the blocks C_u and C_v will be equal to the transposes of B_u and B_v

Algorithm 2. A coupled algorithm

1. Guess initial velocity and pressure fields $\mathbf{u}_0$, $\mathbf{v}_0$, $\mathbf{w}_0$ and $\mathbf{p}_0$.

   For $i = 0, 1, \ldots,$

   2. Solve the coupled system below for $(\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{p})^*$:

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r \mathbf{v}_i \mathbf{u}^*) + \frac{\partial}{\partial z}(\rho \mathbf{u}_i \mathbf{u}^*) - \mu\nabla^2 \mathbf{u}^* + \frac{\partial \mathbf{p}^*}{\partial z} = \mathbf{f}_u,$$

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r \mathbf{v}_i \mathbf{v}^*) + \frac{\partial}{\partial z}(\rho \mathbf{u}_i \mathbf{v}^*) - \mu\left(\nabla^2 \mathbf{v}^* - \frac{\mathbf{v}^*}{r^2}\right) - \frac{2\rho}{r}\mathbf{w}_i \mathbf{w}^* + \frac{\partial \mathbf{p}^*}{\partial r} = \mathbf{f}_v,$$

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho r \mathbf{v}_i \mathbf{w}^*) + \frac{\partial}{\partial z}(\rho \mathbf{u}_i \mathbf{w}^*) - \mu\left(\nabla^2 \mathbf{w}^* - \frac{\mathbf{w}^*}{r^2}\right) - \frac{\rho}{r}\mathbf{w}_i \mathbf{v}^* + \frac{\rho}{r}\mathbf{v}_i \mathbf{w}^* = \mathbf{f}_w,$$

$$\frac{\partial \mathbf{u}^*}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(r\mathbf{v})^* = \mathbf{g}.$$

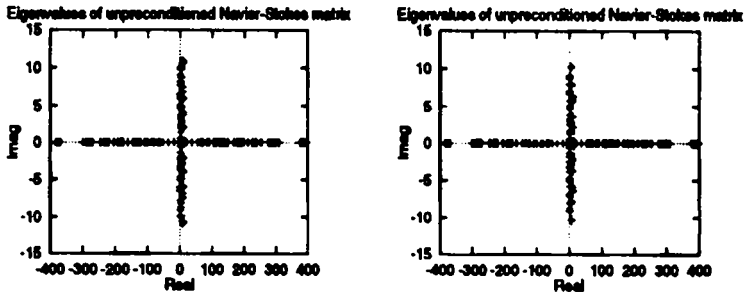
   3. Apply underrelaxation to all variables in the solution according to

$$\mathbf{x}_{i+1} = \varepsilon \mathbf{x}^* + (1 - \varepsilon)\mathbf{x}_i.$$

   If converged, then stop.

respectively. However, here they are only approximate transposes owing to the non-coincident variables and the curvature effect of the polar co-ordinate system. The system matrix is non-symmetric and highly indefinite because of the zero diagonal block. It is also singular (with a null space of dimension one) owing to the hydrostatic pressure mode. As in the segregated case above, the iterative solver is applied directly to the singular system—no explicit specification of a pressure datum is required.

The eigenvalue spectra of two typical matrices (15) at two Reynolds numbers have been computed (on a coarse grid) for our second example and are shown in Figure 6. These spectra suggest that the unpreconditioned linear systems are going to be extremely expensive/difficult to solve iteratively. Notice that the imaginary parts of the eigenvalues are relatively small, however. As we shall see, we can improve the properties of these linear systems substantially by preconditioning.



(a) $Re_\theta = 2.5 \times 10^3$ (b) $Re_\theta = 2.5 \times 10^4$

Figure 6. Problem 2–eigenvalue spectra of typical unpreconditioned linear systems; $N = 422$

To solve such a system requires a robust iteration such as the generalized minimal residual algorithm (GMRES)[18] or the quasi minimal residual algorithm (QMR).[19] In our application GMRES was ineffective as an iterative solver. Once restarted, the convergence of the method became extremely poor. For this reason we do not consider this method further; all following results rely on the use of the QMR method as linear solver.

4.1. *The QMR algorithm*

The QMR algorithm is designed to be robust and overcome the limitations of methods such as the biconjugate gradient (BCG) algorithm which are liable to break down or behave in an unstable manner. The method is usually used in conjunction with the look-ahead Lanczos algorithm[20] and therefore avoids almost all possibilities for breakdown typically associated with the BCG method. The look-ahead Lanczos algorithm 'steps over' iterations where the Lanczos algorithm (which underlies methods such as biconjugate gradients) would either break down completely or cause large oscillations in the residual norm. The convergence of the method in the 2-norm is thus nearly monotonic (see Figure 7) and comparable with unrestarted GMRES. However, unlike the GMRES algorithm, the method uses only a three-term vector recursion to generate each new iterate and is therefore only about as expensive as the BCG algorithm. We found that a large number of look-ahead steps were required during the solution of the linear systems that we generated, suggesting that other iterative methods would perform erratically or even diverge when solving these problems.[21]

4.2. *Preconditioning*

From experiments in solving linear systems generated by our algorithm, it is clear that preconditioning is essential. If we perform block Gaussian elimination on the system (15),

$$\begin{bmatrix} A & C \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix},$$

we obtain the solution

$$\mathbf{p} = -(BA^{-1}C)^{-1}(\mathbf{g} - BA^{-1}\mathbf{f}), \qquad \mathbf{U} = A^{-1}(\mathbf{f} - C\mathbf{p}).$$

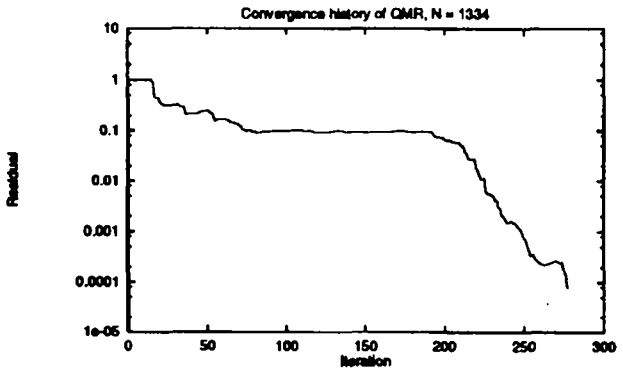


Figure 7. Problem 2—Convergence of ALJ(2)/DS-preconditioned QMR for typical linear system generated by the coupled solver; 17 × 17 grid

Therefore our strategy is to approximate the block convection–diffusion operator A and the matrix $BA^{-1}C$ within a block preconditioner of the form

$$M = \begin{bmatrix} A & 0 \\ 0 & BA^{-1}C \end{bmatrix}.$$

In general an effective preconditioner should cluster the eigenvalues of the preconditioned system away from the origin.[22] We must solve one linear system of the type

$$M\mathbf{y} = \mathbf{z} \tag{16}$$

and one of the type

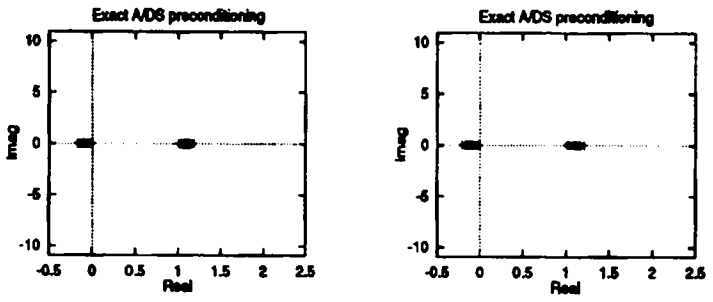
$$M^{\mathrm{T}}\mathbf{y} = \mathbf{z} \tag{17}$$

at each iteration, so the preconditioner M needs to be such that the linear systems (16) and (17) can be solved cheaply. Before presenting numerical results, we show some eigenspectra of the preconditioned system for various preconditioners M that we have used.

We approximate the matrix $BA^{-1}C$ by a diagonal matrix, the scaling of which takes into account the rotational Reynolds number and the grid and domain dimensions of the problem in all the following cases (see Reference 21 for details). Later we also apply one or two sweeps of the alternating line Jacobi method to a standard five-point approximation to the Poisson equation which approximates $BA^{-1}C$.

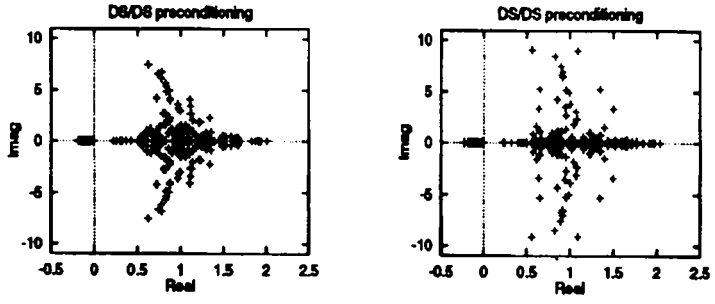
We firstly 'approximate' the convection–diffusion block A by A itself. The preconditioning matrix M thus formed then represents the best available using our approach. Figure 8 shows the effect of this preconditioning on the same linear systems as before. The eigenvalues are well-clustered with small imaginary parts.

In practice, because of the need to solve equations (16) and (17), the use of an exact convection–diffusion block will be too expensive. Therefore we use a diagonal scaling (DS) or a few sweeps of a line relaxation method applied to each of the diagonal blocks of A_{B} as an approximation to the convection–diffusion block A. Figure 9 shows the result of using the diagonal of A_{B} to approximate A and Figure 10 shows the result of using two sweeps of alternating line Jacobi (ALJ(2)) (applied to each diagonal block of A_{B}) to approximate A. The eigenvalues are loosely clustered when using a diagonal approximation. However, a much more well-defined cluster begins to appear when we apply two sweeps of alternating line Jacobi, although a few eigenvalues appear to be more spread out. Applying more than two sweeps of alternating line Jacobi does not appreciably cluster the eigenspectrum of the



(a) $Re_\theta = 2.5 \times 10^3$ (b) $Re_\theta = 2.5 \times 10^4$

Figure 8. Problem 2—eigenvalue spectra of typical exact A/DS-preconditioned linear systems; $N = 422$



(a) $Re_\theta = 2.5 \times 10^3$ (b) $Re_\theta = 2.5 \times 10^4$

Figure 9. Problem 2—eigenvalue spectra of typical DS/DS-preconditioned linear systems; $N = 422$

preconditioned system further. This is presumably due to the fact that the off-diagonal blocks in A which are not included in the matrix A_B are not taken into account by the preconditioner.

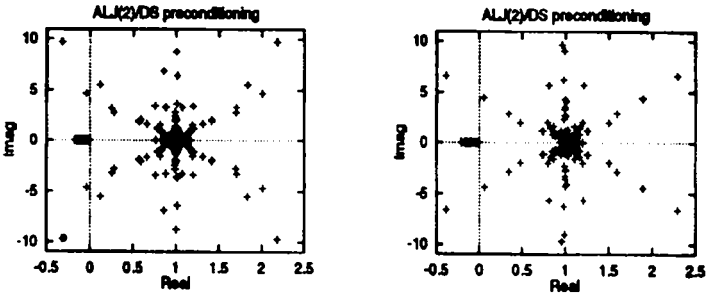
4.3. Storage

The SIMPLE method was proposed when computer storage was at a premium. Any new method may take advantage of the relatively abundant memory to be found on most supercomputers of today. If we denote the maximum number of equations in any of the four variables by N_v and the total number of equations in the coupled system by N, so that $N \approx 4N_v$, then the core of the SIMPLE algorithm requires six vectors of size N_v for coefficient and right-hand side storage plus four vectors of size N_v for the unknowns. The coupled solver requires 31 vectors of size N_v for coefficient storage; the QMR algorithm requires 64 vectors of size N_v and the lookahead Lanczos algorithm with 10 Lanczos vectors requires another 136 vectors of size N_v. Thus the coupled solver requires a total of $221N_v$ extra REAL storage locations. In particular, using REAL*8 arithmetic, a problem defined on a 65×65 grid will require about 7–8 MB extra storage compared with the SIMPLE algorithm.

5. NUMERICAL RESULTS FOR THE COUPLED SOLVER

The following notation is used in the remaining sections:

α Gosman factor



(a) $Re_\theta = 2.5 \times 10^3$ (b) $Re_\theta = 2.5 \times 10^4$

Figure 10. Problem 2—detail of eigenvalue spectra of typical ALJ(2)/DS-preconditioned linear systems; $N = 422$

β_1 non-linear convergence criterion—once all RMS changes in solution fall below this level, the solution is considered converged

β_2 linear convergence criterion—once the linear residual is reduced by this factor, the current linear solution is considered converged

β_3 underrelaxation tolerance—once all non-linear residuals are below this absolute level underrelaxation is applied to the current solution

ε underrelaxation factor for coupled system.

All experiments in this section were performed on a Sun 630MP workstation; all timings are given in seconds on this machine.

5.1. *Preconditioning*

Figure 11 shows the total number of linear iterations performed during the first five non-linear iterations of our coupled solver for a variety of preconditioners and grids. It hence gives a measure of the amount of work done for a given grid dimension. We have concentrated on Problem 2 here, since this generates linear systems which are more difficult to solve and therefore require a more effective preconditioner than Problem 1. The figures show that the most effective preconditioner is ALJ(2)/DS. Therefore we use this preconditioner to obtain all later results.

In Figures 11 and 12 we have also approximated the matrix $BA^{-1}C$ in our preconditioner using alternating line Jacobi as discussed above. Although this gave good pressure residual convergence, it was two or three times as costly as using a diagonal approximation to the matrix. We refer to one or two sweeps of alternating line Jacobi applied to the five-point difference operator as PP(1) and PP(2) respectively.

Once the preconditioner is fixed, the amount of CPU time required to perform one linear iteration is approximately constant for a given grid dimension on a given machine. Thus we use the total number of linear iterations performed whilst obtaining a solution to a problem as a measure of the amount of work required to obtain that solution.

5.2. *Problem 1*

Figure 13 shows the number of non-linear and total number of linear iterations required to obtain a converged solution for our first problem when the linear systems generated are solved to varying levels

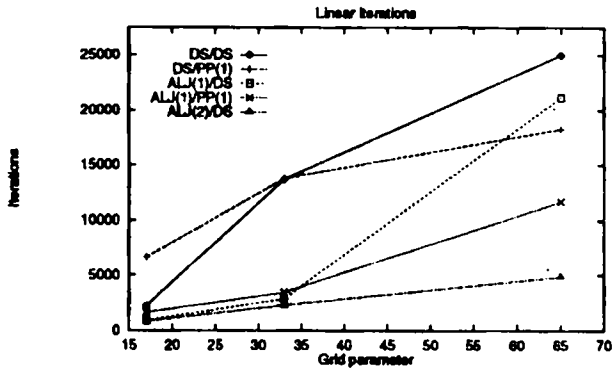


Figure 11. Problem 2—number of linear iterations required to perform five non-linear iterations for various preconditioners; $Re_\theta = 2{\cdot}5 \times 10^3$, $\beta_2 = 1{\cdot}0 \times 10^{-4}$, $\varepsilon = 1{\cdot}0$

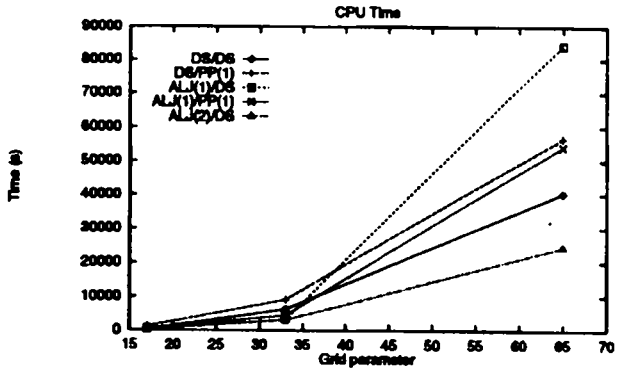


Figure 12. Problem 2—time required to perform five non-linear iterations for various preconditioners; $Re_\theta = 2{\cdot}5 \times 10^3$, $\beta_2 = 1{\cdot}0 \times 10^{-4}$, $\varepsilon = 1{\cdot}0$

of accuracy. For the 17×17 and 33×33 grids, reducing the linear residual by a factor of 10^{-4} gives the best result.

For the case of the 65×65 grid the result is not so clear, since a seemingly anomalous result was obtained at this level of linear residual reduction. The reason for this is shown in Figure 14. This shows the convergence history of the preconditioned QMR algorithm for the linear system generated by the second of the three non-linear iterations required for solution. The number of linear iterations performed is extremely large owing to the fact that convergence to the strict tolerance of $1{\cdot}0 \times 10^{-4}$ was not quite achieved despite fairly rapid initial progress. A better preconditioning would alleviate this type of difficulty and we hope to explore this possibility in future work. Notwithstanding, the amount of work required appears to be lowest 'in the vicinity of' 10^{-4}. The number of non-linear iterations required to obtain a solution is extremely low owing to the fact that this problem is not very non-linear. The number of non-linear iterations required is also independent of the grid dimension provided that we reduce the linear residual sufficiently. Note also that no underrelaxation of the non-linear iterates is required for this problem, again owing to the fact that this problem is almost linear.

5.3. *Problem 2*

Figure 15 shows the effect of varying the global relaxation factor ε for Problem 2 at a rotational Reynolds number $Re_\theta = 2{\cdot}5 \times 10^3$. As the relaxation factor is varied away from the value of 0·65, the number of linear (and nonlinear) iterations increases on both a 17×17 and a 33×33 grid.

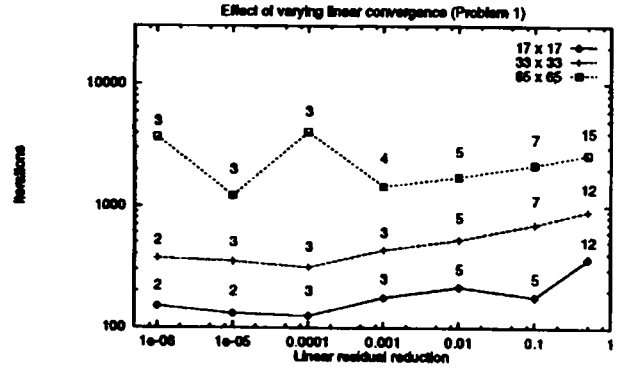


Figure 13. Problem 1—number of linear iterations required for solution for varying linear convergence criterion β_2; $\beta_1 = 1{\cdot}0 \times 10^{-4}$, $\varepsilon = 1{\cdot}0$. Numbers on the graph denote the number of non-linear iterations performed

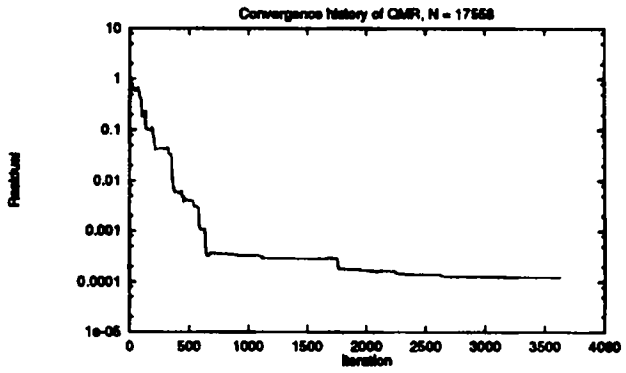


Figure 14. Problem 1—convergence of QMR solver when things go wrong; 65 × 65 grid, $\beta_2 = 1{\cdot}0 \times 10^{-4}$, non-linear iteration 2 (see Figure 13)

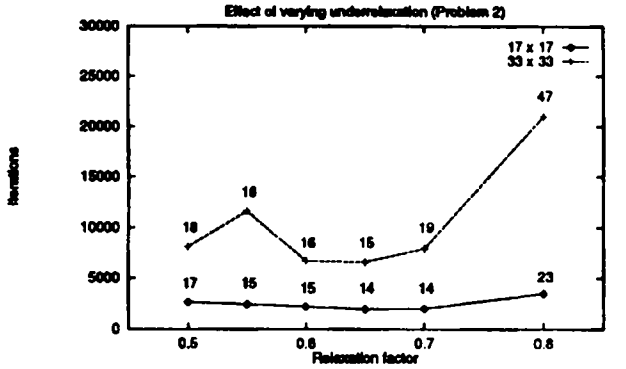


Figure 15. Problem 2—number of linear iterations required for solution for varying global relaxation factor ε; $Re_\theta = 2{\cdot}5 \times 10^3$, $\beta_1 = 1{\cdot}0 \times 10^{-4}$, $\beta_2 = 1{\cdot}0 \times 10^{-4}$, $\beta_3 = 1{\cdot}0 \times 10^{-4}$. Numbers on the graph denote the number of non-linear iterations performed

This value is then fixed (for this problem at this Reynolds number) and the amount of work done in terms of linear residual reduction at each non-linear step is varied (Figure 16). The amount of work required decreases as the linear residual reduction approaches $1{\cdot}0 \times 10^{-1}$ or $1{\cdot}0 \times 10^{-2}$, then starts to rise again as unnecessary extra work is performed in converging the linear system.

Figure 17 shows the effect of increasing the rotational Reynolds number on the amount of work required to solve the linear systems generated by the coupled solver. The figure shows the number of linear iterations required to solve the linear systems generated on a 17 × 17 grid by the first five iterations of our solver, with no relaxation applied. The solution obtained on this grid does not represent an accurate solution of the higher Reynolds number problems. This is due to the much narrower boundary layers in this problem. However, the iteration counts obtained give us insight into the difficulty of solving the linear systems. On the 17 × 17 grid the amount of work required (in terms of number of linear iterations) increases by a factor of 17 as the rotational Reynolds number increases from $2{\cdot}5 \times 10^3$ to $2{\cdot}5 \times 10^4$. Similarly, on the 33 × 33 grid the amount of work increases by a large factor as the Reynolds number is increases. The main difficulty is that the linear systems are becoming increasingly difficult to solve with increasing Reynolds number.

Although extensive experiments were not performed for this problem at the rotational Reynolds number of $2{\cdot}5 \times 10^4$, we found that an underrelaxation factor of 0·2 was necessary to obtain a

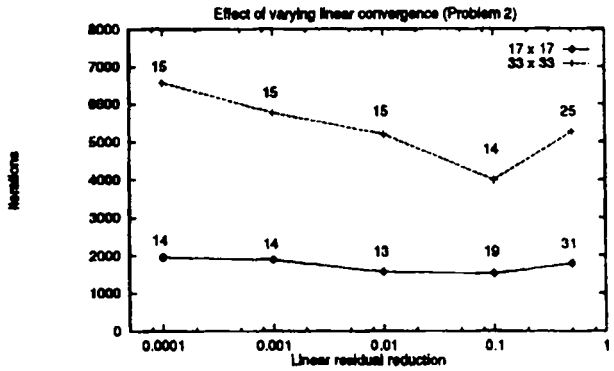


Figure 16. Problem 2—number of linear iterations required for solution for varying linear residual reduction β_2; $Re_\theta = 2{\cdot}5 \times 10^3$, $\beta_1 = 1{\cdot}0 \times 10^{-4}$, $\varepsilon = 0{\cdot}65$, $\beta_3 = 1{\cdot}0 \times 10^{-4}$. Numbers on the graph denote the number of non-linear iterations performed

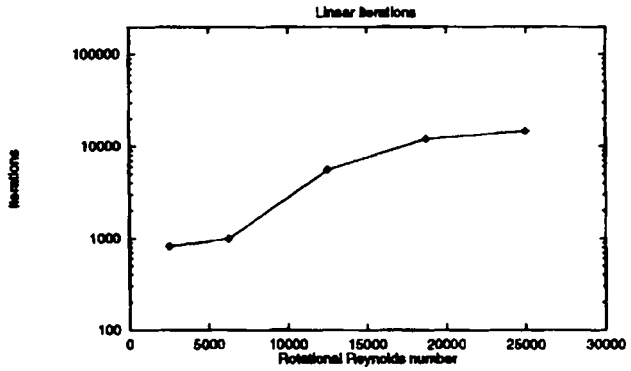


Figure 17. Problem 2—number of linear iterations required to perform five non-linear iterations for varying rotational Reynolds number Re_θ; $\beta_1 = 1{\cdot}0 \times 10^{-4}$, $\beta_2 = 1{\cdot}0 \times 10^{-4}$, $\varepsilon = 1{\cdot}0$, $\beta_3 = 1{\cdot}0 \times 10^{-4}$

converged solution. Figure 18 shows that the number of non-linear iterations required to solve the higher rotational Reynolds number problem is increased by a factor of about four or five over the number required for the lower Reynolds number problem. However, the number of linear iterations (and hence the CPU time) required has increased by a large amount, a factor of 50–100. In fact the average number of linear iterations required per non-linear iteration as we increase the rotational Reynolds number by a factor of 10 from $2{\cdot}5 \times 10^3$ to $2{\cdot}5 \times 10^4$ (with a linear convergence criterion of $\beta_2 = 1{\cdot}0 \times 10^{-4}$) has increased from 140 to at least 3200 for the 17×17 grid and from 440 to at least 4600 for the 33×33 grid. Indeed, these linear systems were already extremely difficult to solve at the original rotational Reynolds number of $2{\cdot}5 \times 10^3$. It is because the computational times were so great for the highest Reynolds number test case that most of our results were obtained at the lower Reynolds number.

The increased difficulty in solving the linear systems as the Reynolds number increases is due to the problem becoming more dominated by rotational effects. The off-diagonal blocks A_{vw} and A_{wv} become increasingly important, reducing the diagonal dominance of the linear systems generated. Figure 6(b) shows the eigenvalues of an unpreconditioned matrix at the higher Reynolds number of $2{\cdot}5 \times 10^4$. This matrix was produced after 40 non-linear iterations (the problem was converged after about 65 iterations). Our linear solver required 1147 iterations to reduce the linear residual of the system of dimension 422 by a factor of 10^{-2}. The spectrum appears to be largely similar to the spectrum

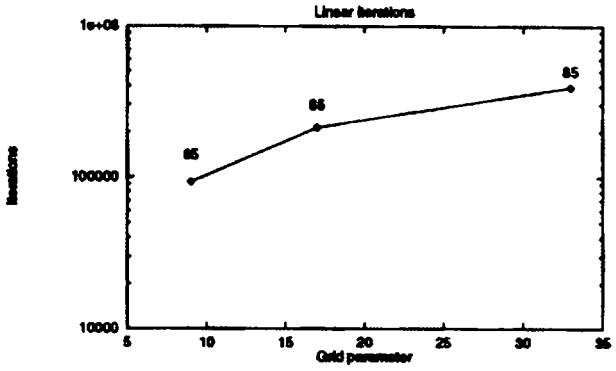


Figure 18. Problem 2, $Re_\theta = 2{\cdot}5 \times 10^4$—number of linear iterations required for solution; $\beta_1 = 1{\cdot}0 \times 10^{-4}$, $\beta_2 = 1{\cdot}0 \times 10^{-4}$, $\varepsilon = 0{\cdot}2$, $\beta_3 = 1{\cdot}0$. Numbers on the graph denote the number of non-linear iterations performed

produced at the lower Reynolds number, as does the spectrum of the exact A/DS-preconditioned matrix (Figure 8). However, when we apply our 'realistic' preconditioners, diagonal scaling and alternating line Jacobi, the differences are more noticeable. Figure 9 shows the eigenvalue spectra for the diagonal scaling preconditioner at both Reynolds numbers; Figure 10 gives the corresponding spectra for the alternating line Jacobi preconditioner which we have used in our experiments. At the higher Reynolds number the spectra for both preconditioners, although similar in form, are noticeably more dispersed.

6. PERFORMANCE COMPARISON OF COUPLED AND SEGREGATED SOLVERS

Figures 19–21 show the performance of the coupled solver compared with the SIMPLEC algorithm. The parameters used with the coupled and segregated solvers are summarized in Tables I and II respectively. The figures show the number of non-linear iterations and CPU times. The convergence criterion of RMS changes was not comparable for the two solution techniques, so instead we used a reduction of the residuals for all the variables by a factor of 10^{-4} as a stopping criterion. For the SIMPLEC calculations the choice of parameters was based on experience with this computer code and results of earlier work rather than any optimization in the present study. SIMPLEC(alpha) refers to the SIMPLEC algorithm with a Gosman factor of 50·0, otherwise the value of this parameter was zero.

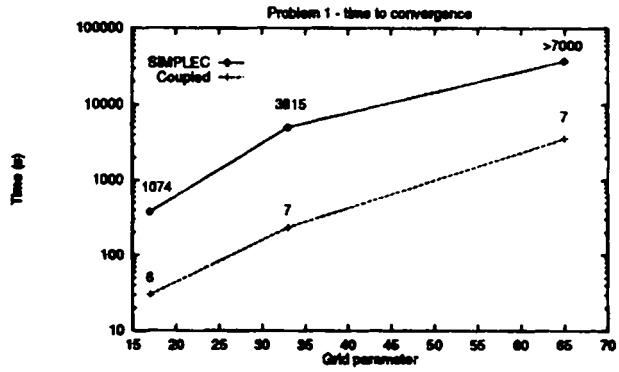


Figure 19. Problem 1—time to reduce non-linear residuals by 10^{-4}. Numbers on the graph denote the number of non-linear iterations performed

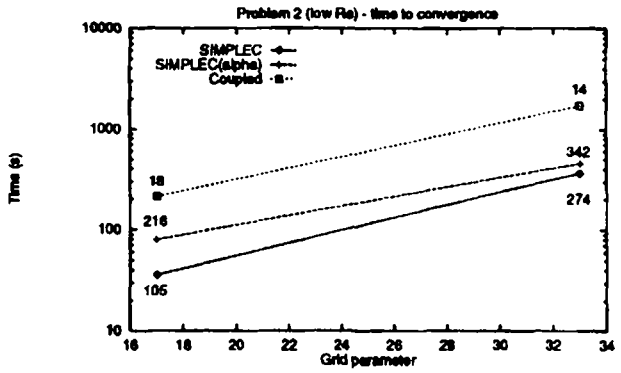


Figure 20. Problem 2—time to reduce non-linear residuals by 10^{-4}; $Re_\theta = 2{\cdot}5 \times 10^3$. Numbers on the graph denote the number of non-linear iterations performed

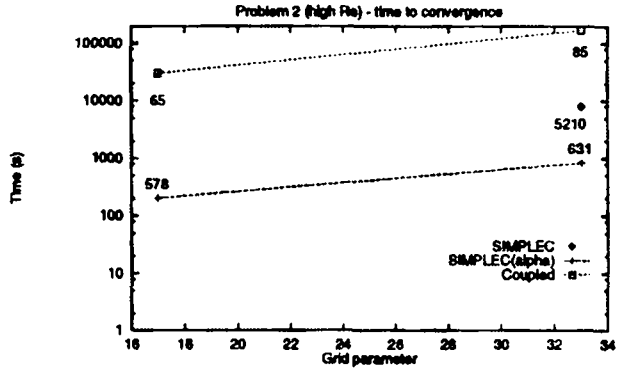


Figure 21. Problem 2—time to reduce non-linear residuals by 10^{-4}; $Re_\theta = 2{\cdot}5 \times 10^4$. Numbers on the graph denote the number of non-linear iterations performed

Table I. Summary of parameters used with coupled solver for performance comparison

Parameter		Problem 1	Problem 2	Problem 2
Preconditioner		ALJ(2)/DS	ALJ(2)/DS	ALJ(2)/DS
Reynolds number	Re_θ	2·2	$2{\cdot}5 \times 10^3$	$2{\cdot}5 \times 10^4$
Non-linear convergence	β_1	10^{-4}	10^{-4}	10^{-4}
Linear convergence	β_2	10^{-4}	10^{-1}	10^{-4}
Underrelaxation tolerance	β_3	—	10^{-4}	∞
Underrelaxation factor	ε	1·0	0·65	0·2

Figure 19 shows the results for Problem 1. Although the SIMPLEC iterates were converging slowly, the number of iterations and CPU times were appreciably larger than the coupled solver for this problem. The use of a non-zero Gosman factor with the SIMPLEC algorithm did not change the convergence behaviour of the method for this problem.

Figure 20 gives a comparison of the methods for Problem 2 at the rotational Reynolds number of $2{\cdot}5 \times 10^3$. The coupled solver is about five times slower in terms of CPU time to converge than the SIMPLEC method.

Table II. Summary of parameters used with segregated solver for performance comparison

Parameter		Problem 1	Problem 2	Problem 2	
Reynolds number	Re_θ	2·2	$2{\cdot}5 \times 10^3$	$2{\cdot}5 \times 10^4$	
Non-linear convergence	β_1	10^{-4}	10^{-4}	10^{-4}	
Gosman factor	α	0·0	0·0, 50·0	0·0	50·0
Momentum sweeps	n_u	2	2	2	2
Pressure sweeps	n_p	3	3	10	3
u-relaxation	α_u	0·5	0·5	0·2	0·5
v-relaxation	α_v	0·5	0·5	0·2	0·5
w-relaxation	α_w	0·5	0·8	0·2	0·8
p-relaxation	α_p	0·8	1·0	0·4	1·0

Figure 21 shows the convergence of the methods for Problem 2 at the higher rotational Reynolds number of $2{\cdot}5 \times 10^4$. Although the coupled solver performed well in terms of number of non-linear iterations, the linear systems generated were extremely difficult to solve, resulting in relatively slow convergence. The SIMPLEC algorithm stagnated for the problem on the 17×17 grid. On the 33×33 grid a Gosman factor of 50·0 accelerated the convergence of the SIMPLEC algorithm by a factor of about 10, compared with the method with a Gosman factor of zero. This clearly emphasizes the sensitivity of the segregated approach to parameter choices—using the same Gosman factor $\alpha = 50{\cdot}0$ at the lower Reynolds number (i.e. Figure 20) actually slowed the convergence of the SIMPLEC method.

Finally, Figures 22 and 23 show convergence histories of the SIMPLEC algorithm with a Gosman factor of zero and our coupled solver applied to Problem 2 at a rotational Reynolds number of $2{\cdot}5 \times 10^3$ on a 33×33 grid. The residuals of the coupled solver are almost monotonically decreasing, whereas the convergence of the SIMPLEC iterates appears to be mildly unstable.

7. CONCLUSIONS

We have presented a comparison of two quite different solution strategies for the incompressible steady Navier–Stokes equations. In the coupled solution method, linearized equations for the momentum and continuity equations are solved simultaneously. The segregated algorithm is based on the SIMPLE method in which linearized momentum equations are solved successively, followed by a pressure

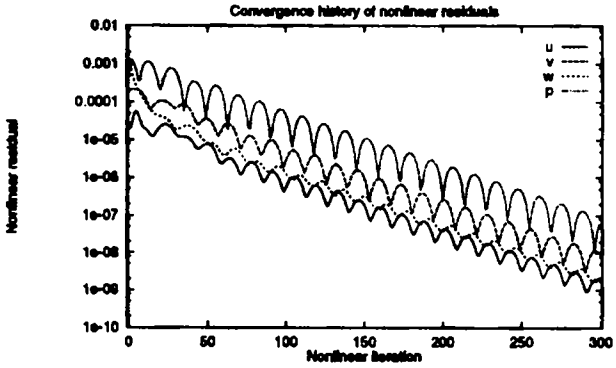


Figure 22. Problem 2—typical convergence history for SIMPLEC algorithm; 33×33 grid, $Re_\theta = 2{\cdot}5 \times 10^3$

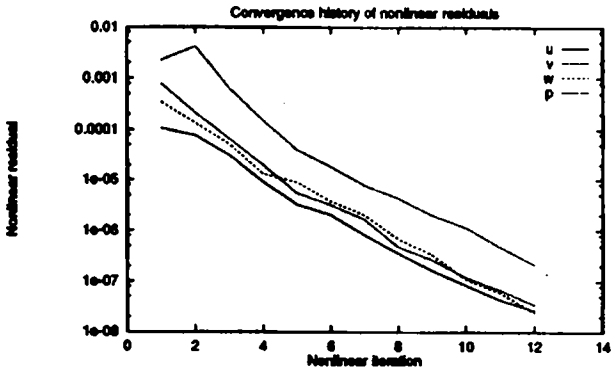


Figure 23. Problem 2—typical convergence history for coupled solver; 33 × 33 grid, $Re_\theta = 2{\cdot}5 \times 10^3$

correction step which links the continuity equation and a crude approximation to the momentum equations. In both cases a staggered grid, hybrid finite difference discretization scheme has been employed. The evaluation of the two methods has centred on examples of swirling flows which are known to give difficulties using the segregated approach.

As might be expected, the coupled solver gave convergence in considerably fewer non-linear (or outer) iterations than the segregated method. Generally the number of non-linear iterations was reduced by at least an order of magnitude using the coupled approach. The performance of the linear solver depended on the 'difficulty' of the flow problem; our line relaxation preconditioner worked well for low Reynolds number flow and in this case our coupled solver was superior (by an order of magnitude). The performance of our linear solver deteriorated with increasing Reynolds number and in this case a more effective preconditioner seems to be essential if the coupled approach is to be competitive with our benchmark SIMPLE approach.

We emphasize that some caution is required when interpreting the comparisons between solvers—even for the limited range of test cases considered here, large variations in performance can arise owing to the choice of underrelaxation factors and other parameters in the algorithms. For the most difficult problem that we have considered, the rotating cavity at a Reynolds number of $2{\cdot}5 \times 10^4$, the performance of the SIMPLEC algorithm with the treatment of the v–w coupling proposed by Gosman *et al.* is particularly impressive. It should be noted, however, that such near-optimal performance may only be achieved after considerable experimentation.

Finally, it is interesting to compare our results with those of Haroutunian *et al.*[23] These workers examined the performance of segregated and coupled finite element algorithms for a number of non-rotating flow problems. In the coupled approach a direct linear solver was used, whilst direct and iterative solvers were used with the segregated method. Based on their tests, they concluded that segregated methods with iterative solvers are to be preferred for difficult problems. Considering that we have specifically chosen test problems for which rotation introduces strong coupling between the momentum equations, our results seem to support the conclusions in their work.

ACKNOWLEDGEMENTS

We gratefully acknowledge the use of the QMR routines written by Noël Nachtigal (available from NETLIB) within our code.

The work of the first author was supported by the SERC and Rolls-Royce plc.

REFERENCES

1. 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).
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. A. D. Gosman, M. L. Koosinlin, F. C. Lockwood and D. B. Spalding, 'Transfer of heat in rotating systems', *ASME Publ. 76-GT-25*, 1976.
4. C. M. Vaughan, S. Gilham and J. W. Chew, 'Numerical solutions of rotating disc flows using a non-linear multigrid algorithm', *Proc. 6th Int. Conf. on Numerical Methods in Laminar and Turbulent Flow*, Swansea, July 1989.
5. D. S. Jang and S. Acharya, 'Source-term decomposition to improve convergence of swirling flow calculations', *AIAA J.* **26**, 372–374 (1988).
6. 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).
7. P. M. Gresho, 'On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory', *Int. J. numer. methods fluids*, **11**, 587–620 (1990).
8. P. M. Gresho and S. T. Chan, 'On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation,' *Int. j. numer. methods fluids*, **11**, 621–660 (1990).
9. M. S. Engelman, *FIDAP Theoretical Manual, Version 6.0*, Fluid Dynamics International (FDI), 1991.
10. P. R. McHugh and D. A. Knoll, 'Fully coupled finite volume solutions of the incompressible Navier–Stokes equations using an inexact Newton method', *Int. j. numer. methods fluids*, **19**, 439–455 (1994).
11. S. S. Clift and P. A. Forsyth, 'Linear and nonlinear iterative methods for the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **18**, 229–256 (1994).
12. D. A. Knoll and P. R. McHugh, 'A fully implicit direct Newton's method for the steady-state Navier–Stokes equations', *Int. j. numer. methods fluids*, **17**, 449–461 (1993).
13. D. J. Silvester, 'An analysis of finite element approximation for swirling flow', *Ph.D. Thesis*, Department of Mathematics, University of Manchester, Institute of Science and Technology, 1984.
14. J. W. Chew, J. M. Owen and J. R. Pincombe, 'Numerical predictions for source–sink flow in a rotating cavity', *J. Fluid Mech.*, **143**, 451–466 (1984).
15. 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', *Numerical Analysis Rep. 95*, Department of Mathematics, University of Manchester, 1984.
16. D. B. Spalding, 'A novel finite-difference formulation for differential expressions involving both first and second derivatives', *Int. j. numer. methods eng.*, **4**, 551–559 (1972).
17. J. R. Kightley, 'The conjugate gradient method applied to turbulent flow calculations', *Tech. Rep. CSS 184*, Harwell Laboratory, 1985.
18. Y. Saad and M. H. Schultz, 'GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).
19. R. W. Freund and N. M. Nachtigal, 'QMR: a quasi-minimal residual method for non-Hermitian linear systems', *Numer. Math.*, **60**, 315–339 (1991).
20. R. W. Freund, M. H. Gutknecht and N. M. Nachtigal, 'An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices', *SIAM J. Sci. Stat. Comput.*, **14**, 137–158 (1993).
21. R. F. Hanby, 'Segregated and coupled iterative solution algorithms for incompressible swirling flows', *Ph.D. Thesis*, 1995.
22. R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1993.
23. V. Haroutunian, M. S. Engelman and I. Hasbani, 'Segregated finite element algorithms for the numerical solution of large-scale incompressible flow problems', *Int. j. numer. methods fluids*, **17**, 323–348 (1993).