INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 10, 429–442 (1990)

AN ITERATIVE IMPLEMENTATION OF THE UZAWA ALGORITHM FOR 3-D FLUID FLOW PROBLEMS

MICHEL P. ROBICHAUD

Control Data Canada, CP 999, Place du Parc, Montreal, Quebec, H2W 2N1, Canada

PHILIPPE A. TANGUY

Department of Chemical Engineering and CERSIM, Université Laval, Québec, G1K 7P4, Canada

AND

M. FORTIN

Department of Mathematics, Université Laval, Québec, G1K 7P4, Canada

SUMMARY

A new iterative algorithm for the solution of the three-dimensional Navier–Stokes equations by the finite element method is presented. This algorithm is based on a combination of the Uzawa and the Arrow–Hurwicz algorithms and uses a preconditioning technique to enhance convergence. Numerical tests are presented for the cubic cavity problem with two elements, namely the linear brick Q_1–P_0 and the enriched linear brick $Q_1^+ - P_1$. It is shown that the proposed methodology is optimal with the enriched element and that the CPU time varies as $NEQ^{1 \cdot 44}$, where NEQ is the number of equations.

KEY WORDS Uzawa algorithm Preconditioning Incomplete factorization Three-dimensional Fluid flow

INTRODUCTION

The development of efficient finite element methods for the computation of three-dimensional incompressible viscous fluid flow problems is a current field of active research. In a three-dimensional context, iterative methods are commonly accepted to be a much better alternative than direct methods (*LDU* factorization), since they yield better numerical performance (smaller memory requirements and CPU time) and can adapt more easily to parallelism and vectorization. In flow problems the major difficulty lies in finding an iterative method which can deal effectively with the incompressibility constraint. Indeed, iterative solvers are very sensitive to conditioning and it is well known that the divergence-free constraint degrades matrix conditioning.

Among the various iterative techniques which can be used to solve linear systems, methods based on conjugate gradients are the best choice. For unconstrained elliptic or quasi-elliptic problems, the preconditioned conjugate gradient method using incomplete factorization as preconditioning (called the incomplete Choleski conjugate gradient method or ICCG) has been used very successfully.[1–3] For constrained problems such as the Navier–Stokes equations, conjugate gradient methods work well only for low-Reynolds-number steady state problems discretized with a mixed (velocity–pressure) formulation.[4]

To overcome the difficulty associated with the imposition of the divergence-free constraint, a successful decoupling technique has been proposed for the Stokes equations.[5,6] This method is

0271–2091/90/040429–14$07.00

Received 27 June 1988
Revised 18 January 1989

based on the Uzawa algorithm, which consists of computing the momentum transport equation (primal problem) separately from the incompressibility constraint (dual problem). The primal problem is solved with the ICCG method while the dual problem is preconditioned by a Poisson equation and solved iteratively. More recently,[7] an iterative method combining the Uzawa algorithm and an Arrow–Hurwicz method has been introduced. It uses block-domain decomposition as preconditioning.

In this work a variant of the two previous methods is presented. It consists of an iterative implementation of the Uzawa algorithm and makes use of an incomplete LDL^T factorization of the viscous matrix as preconditioning. The convergence properties of this algorithm are assessed for the three-dimensional lid-driven cubic cavity problem as a function of the mesh size (i.e. the number of elements) for two types of elements.

FINITE ELEMENT APPROXIMATION

Let us consider the equations governing a viscous incompressible fluid flow:

$$\rho(v\cdot\operatorname{grad} v) = -\operatorname{grad} p + \mu \operatorname{div}(\operatorname{grad} v), \tag{1}$$

$$\operatorname{div} v = 0, \tag{2}$$

where v denotes the velocity vector, p is the pressure, and ρ and μ denote the fluid density and viscosity respectively. The solution of (1), (2) is equivalent to considering the following weak variational formulation:

$$R(v, p) = \mu/\rho\, a(v, \psi) + (v\cdot\operatorname{grad} v, \psi) + 1/\rho(p, \operatorname{div}\psi) = 0, \quad \forall\psi\in V, \tag{3}$$

$$(\Phi, \operatorname{div} v) = 0, \quad \forall\Phi\in Q, \tag{4}$$

where $R(v, p)$ is the residual, and ψ and Φ are test functions belonging to the appropriate spaces V and Q. The $L^2(\Omega)$ scalar product $(,)$ is defined as

$$(u, v) = \int_\Omega uv\, d\Omega, \tag{5}$$

and $a(v, \psi)$, the diffusion term, is of the form

$$a(v, \psi) = \int_\Omega \operatorname{grad} v \operatorname{grad}\psi\, d\Omega. \tag{6}$$

The finite element method is used to discretize (3), (4). Two types of elements are considered. The first element is the standard Q_1–P_0 brick, a hexahedral element with trilinear velocity and piecewise-discontinuous constant pressure. This element does not satisfy the Brezzi compatibility condition and is thus subject to spurious pressure modes and possibly to locking if the number of constraint equations is superior to the number of degrees of freedom. The reader is referred to Reference 8 for more explanations and to Reference 4 for an example. The second element is an enriched hexahedral element called Q_1^+–P_1;[9,10] it is obtained by adding to the previous element one velocity degree of freedom on each face (normal velocity), approximated with an incomplete triquadratic polynomial, and three velocity degrees of freedom at the element centroid, approximated with a triquadratic polynomial. Such an enrichment of the discrete velocity basis allows the use of a linear approximation for the pressure; the Brezzi compatibility condition is satisfied at the element level, enabling local mass conservation. The numerical integration is performed with a

Gaussian quadrature using $2\times2\times2$ integration points for the Q_1-P_0 and $3\times3\times3$ for the $Q_1^+-P_1$.

The following matrix formulation can then be readily obtained:

$$[A+C(v_h)]\,v_h+B^{T}p_h=F, \tag{7}$$

$$Bv_h=0, \tag{8}$$

where v_h and p_h are the discrete velocity and pressure (nodal values) respectively, A is the viscous term matrix, B is the divergence matrix, $C(v_h)$ is the advection matrix and F is a vector which takes into account the boundary conditions.

UZAWA ALGORITHM

We recall now the basic ideas of the Uzawa algorithm.[11] Based on the decoupling of the momentum equations and the divergence-free constraint, it applies only to symmetric positive-definite problems. Let us first consider the Stokes problem (the advection matrix is omitted in (7)), which belongs to this class of problems. Its solution can be obtained by solving the following saddle-point problem:

$$L(v,p)=\operatorname*{Inf}_{u\in V}\operatorname*{Sup}_{q\in Q}L(u,q), \tag{9}$$

where the Lagrangian functional is expressed as

$$L(u,q)=J(u)+(q,Bu)=\tfrac{1}{2}(Au,u)-(F,u)+(q,Bu). \tag{10}$$

It is well known that, to improve convergence, a penalty technique can be used to impose the incompressibility constraint, leading to an augmented Lagrangian functional

$$L_r(u,q)=L(u,q)+r|Bu|^2. \tag{11}$$

The Uzawa algorithm consists of two steps. The first step, called the primal problem, is nothing but the minimization of $L(u,q)$ with respect to u. When a direct solver is used, the solution is simply (the subscript h will be omitted to simplify the notation):

$$v=A^{-1}(F-B^{T}q). \tag{12}$$

Moreover, it can then be shown[11] from (11) that, for a symmetric positive-definite matrix, the maximization in q, called the dual problem, is equivalent to solving

$$BA^{-1}B^{T}p=BA^{-1}F. \tag{13}$$

This constitutes the second step. A natural choice for the resolution of (13) is the conjugate gradient method. Indeed, this method avoids the construction of a global $BA^{-1}B^{T}$ operator, which cannot be computed for elements with discontinuous pressures. This step represents the projection of the velocity field onto a divergence-free subspace.

Uzawa algorithm

p^1 being given

Primal problem

(i) $v^1=A^{-1}(F-B^{T}p^1)$

Dual problem

For $j = 1$ to m

(ii) $d^j = Bv^j$ for $j = 1$
$d^j = Bv^j + b^j d^{j-1}$ for $j > 1$
with $b^j = (Bv^j, Bv^j)/(Bv^{j-1}, Bv^{j-1})$

(iii) $p^{j+1} = p^j + a^j d^j$
$v^{j+1} = v^j + a^j z^j$
with $z^j = A^{-1}B^{\mathrm{T}} d^j$
$a^j = (Bv^j, Bv^j)/(Bv^j, Bz^j)$

Next j.

where m is the maximum number of projections. From a practical point of view, a stopping criterion is used inside the loop to assess the convergence of the dual problem.

This algorithm was studied by Fortin and Glowinski.[11] They showed that it is very efficient when the divergence-free condition is penalized. In such a case the matrix A is replaced by

$$A_r = A + rB^{\mathrm{T}}B, \tag{14}$$

where r is the penalty parameter. For large values of r (in the range 10^6–10^8), a^j in step (iii) tends towards r. In this case it should be noted that $m = 1$ is sufficient to converge the dual problem. By using a descent method instead of a conjugate gradient method, the Uzawa algorithm can then be simplified into a classic penalty algorithm. The use of a moderate value of r (in the range 10–100) has a preconditioning effect on the dual problem and increases the convergence of the algorithm significantly.

INCOMPLETE UZAWA ALGORITHM

For three-dimensional problems the Uzawa algorithm requires large computational resources owing to the use of a direct solver for the primal problem. We propose in this work to introduce an iterative method for the solution of (7). It is based on a modified two-step Arrow–Hurwicz algorithm.[12] Let us first present this algorithm.

Instead of inverting the matrix A, an approximation of A_r^{-1}, denoted S_r^{-1}, is computed and used in an iterative fashion. The resulting algorithm is presented below.

Modified Arrow–Hurwicz algorithm

v^1 and p^1 being given

For $i = 1$ to n

Primal problem

(i) $v^{i+1/2} = v^i + a^i S_r^{-1} R_i$

with $R_i = (F - B^{\mathrm{T}}p^i - A_r v^i)$
$a^i = (R^i, S_r^{-1}R_i)/(AS_r^{-1}R_i, S_r^{-1}R_i)$

Dual problem

(ii) $p^{i+1} = p^i + b^i Bv^{i+1/2}$
$v^{i+1} = v^{i+1/2} + b^i S_r^{-1} B^{\mathrm{T}} v^{i+1/2}$
with $b^i = (Bv^{i+1/2}, Bv^{i+1/2})/(Bv^{i+1/2}, BS_r^{-1}B^{\mathrm{T}}Bv^{i+1/2})$

Next i.

The above algorithm calls for a few comments. The second step was introduced by Aboulaich *et al.*[7] in order to minimize the divergence of v^{i+1}. It can be viewed as one step of a gradient method, applied to the problem of projecting v^i onto a divergence-free subspace with respect to the norm associated with the matrix A_r (A_r-norm). To make things clearer and to grasp better the significance of the method described later, we shall first analyse the Uzawa algorithm in more detail. The first step of this algorithm minimizes, p^1 being given, the Lagrangian $L_r(u, q)$ with respect to u. The second step projects v^i onto the divergence-free subspace with respect to the A_r-norm. Indeed, let us consider the problem

$$\operatorname*{Inf}_{Bv=0,\ v\in V} \tfrac{1}{2}(A_r(v-v^1),\ v-v^1), \tag{15}$$

where V is the space of admissible velocities. It is easily seen that the solution of this equation satisfies

$$A_r v + B^{\mathrm{T}} p = A_r v^1, \tag{16}$$

$$Bv = 0. \tag{17}$$

But every projection iteration of the Uzawa algorithm leads to

$$A_r v^{i+1} = A_r v^i + B^{\mathrm{T}} p^n. \tag{18}$$

The solution of the Stokes problem is then nothing else but the projection of the unconstrained solution onto the divergence-free subspace with respect to the A_r-norm.

Suppose now that we have computed an approximation S_r^{-1} of A_r^{-1}. An important point is that r should not be chosen too large since the condition number of A_r would increase, thus decreasing the convergence rate of the iterative method. Let us consider an implementation of the Uzawa algorithm into the modified Arrow–Hurwicz algorithm. As for the previous algorithms, it would comprise two steps. The first step is similar to the first step of the Arrow–Hurwicz algorithm and consists of solving the primal problem by an iterative method using $S_r^{-1}R_i$ as the descent direction. The second step is nothing but the resolution of the dual problem in an Uzawa-like fashion.

The primal and dual problems can be solved either by a descent or by a conjugate gradient method. The choice of a method may vary from one step to another. It depends on the convergence properties and the respective amount of computational time the methods require. It should be noted that this choice may depend on the problem size and may also be influenced by the computer architecture (vector or parallel computer). In the following we consider the use of a descent method for the resolution of the primal problem and a conjugate gradient method for the resolution of the dual problem. The resulting algorithm, called the incomplete Uzawa algorithm, is as follows.

Incomplete Uzawa algorithm

v^1 and p^1 being given

For $i = 1$ to n

Primal problem

(i) $v^{i+1} = v^i + c^i S_r^{-1} R^i$
with $R^i = (F - B^{\mathrm{T}} p^i - Av^i)$
$c^i = (R^i,\ S_r^{-1}R^i)/(AS_r^{-1}R^i,\ S_r^{-1}R^i)$

Dual problem

For $j = 1$ to m

Let $v^j = v^{i+1}$ and $p^j = p^{i+1}$ for $j = 1$

(ii) $d^j = Bv^j$ for $j = 1$

$d^j = Bv^j + b^j d^{j-1}$ for $j > 1$

with $b^j = (Bv^j, Bv^j)/(Bv^{j-1}, Bv^{j-1})$

(iii) $p^{j+1} = p^j + a^j d^j$

$v^{j+1} = v^j + a^j z^j$

with $z^j = S_r^{-1} B^T d^j$

$a^j = (Bv^j, Bv^j)/(Bv^j, Bz^j)$

Next j

(iv) $v^{i+1} = v^{j+1}$

$p^{j+1} = p^{i+1}$

Next i.

Several ways of preconditioning may be considered to enhance the convergence of the above algorithm: block-domain decomposition,[7] element-by-element preconditioners,[13] multigrid methods[14] and incomplete factorization, among others. The incomplete factorization is a very interesting choice since it yields a computationally efficient algorithm when combined with the preconditioned conjugate gradient method. In the present work an incomplete LDL^T factorization coupled to a packed skyline static storage method[4] is used.

The remainder of the paper is devoted to the numerical study of the convergence properties of the incomplete Uzawa algorithm. An important issue is to obtain an iterative method that does not slow down too much as the number of elements increases. Our goal is to be as close as possible to the performance of the ICCG method used for unconstrained systems. We recall that, for an $N \times N \times N$ mesh (lid-driven cavity problem), the ICCG method applied to a second-order elliptic problem[15] is of order $\mathrm{NEQ}^{1.17}$, where NEQ is the number of equations. Since there is a price to pay for the solution of the dual problem in our algorithm, we hope to obtain a figure comprised between the order of the ICCG method and the order of the LDU factorization,[15] $\mathrm{NEQ}^{2.33}$. Since there is no theoretical background to select optimal values for the number of primal and dual iterations, numerical experiments will be used to devise an algorithm of optimal order.

NUMERICAL RESULTS

Similarly to the original Uzawa algorithm, the incomplete Uzawa algorithm requires a symmetric positive-definite matrix system. This algorithm can then be used directly to get the solution of the Stokes equation, for both creeping or quasi-creeping flow problems (low Reynolds number), provided the convective terms have been placed on the right-hand side of the equation. There is obviously a threshold value for the Reynolds number above which this Picard-like strategy will not work any longer. Prior to evaluating this threshold value, let us first study the numerical behaviour of the solution of the Stokes problem.

Stokes problem

The classical lid-driven cubic cavity problem is considered for the numerical tests. Three regular structured meshes are used: $7 \times 7 \times 7$ elements, $9 \times 9 \times 9$ elements and $11 \times 11 \times 11$

elements (see Table I). The convergence of the algorithm is analysed by plotting the norm of the residual of (7), $\|R\|$, defined as $(R, R)^{1/2}$, and the norm of the divergence of the velocity field, $\|Bv\|$, defined as $(Bv, Bv)^{1/2}$, versus the number of iterations. It must be first pointed out that the convergence of $\|R\|$ will be influenced by the convergence of $\|Bv\|$ since the pressure, which cannot be more accurate than the divergence of the flow field, appears explicitly in the expression of $\|R\|$.

Another measure of the convergence properties is also used. It is defined differently for the primal and dual problems. Two parameters are introduced:

$$K_p = -\log_{10}(\|R\|^i/\|R\|^0)/i, \tag{19}$$

$$K_d = -\log_{10}(\|Bv\|^j/\|Bv\|^0)/j, \tag{20}$$

where i and j represent an iteration counter for the primal and dual problems. K_p and K_d represent the slope of $\|R\|$ and $\|Bv\|$ with respect to the number of iterations on a semi-logarithmic graph. A similar measure of the convergence rate has been used for multigrid methods.[14]

Primal and dual problem convergence

It is first interesting to study the convergence of the primal and dual problems separately. The primal problem is a standard unconstrained problem solved with an ICCG method. Its convergence rate is of order $\mathrm{NEQ}^{1\cdot 17}$. For the dual problem the most important aspect is the rate at which the divergence vanishes. The convergence of $\|Bv\|$ versus the number of projection iterations is presented in Figure 1 for the three meshes (Q_1–P_0 and Q_1^+–P_1 elements). It can be observed that for the latter element the convergence is linear and does not depend on the problem size. This is indeed a property of the Uzawa algorithm.

For the Q_1–P_0 element a quite different trend can be noted. The rate of convergence is large during the first iterations but then decreases rapidly. When the value of $\|Bv\|$ reaches 10^{-5}, the rate of convergence starts increasing, to become finally linear after $\|Bv\|$ reaches 10^{-6}. Furthermore, the number of iterations needed before reaching the linear convergence zone increases significantly with the problem size. This phenomenon is due to the fact that the Q_1–P_0 element does not satisfy the Brezzi compatibility condition. It can be shown that some eigenvalues of the dual problem tend towards zero with the element size, and that their number increases with the number of elements.[16] The dual problem is thus ill conditioned and the convergence of the algorithm degrades.

The convergence of the Uzawa algorithm may be improved by using a penalty technique to impose the incompressibility constraint. It should be pointed out, however, that penalizing decreases the convergence rate of the ICCG method for the primal problem, because it affects the matrix conditioning. The penalization effect of the convergence rate of both the primal and dual problems is displayed in Table II for the three meshes considered (Q_1^+–P_1 elements). In the case of the dual problem the convergence rate (K_d) is presented for the conjugate gradient–Uzawa (CG) algorithm and for the optimal descent–Uzawa (OD) algorithm. For the primal problem the

Table I. Description of the cavity meshes

Number of elements		Matrix rank
7×7×7	(343)	720
9×9×9	(729)	1664
11×11×11	(1331)	3260

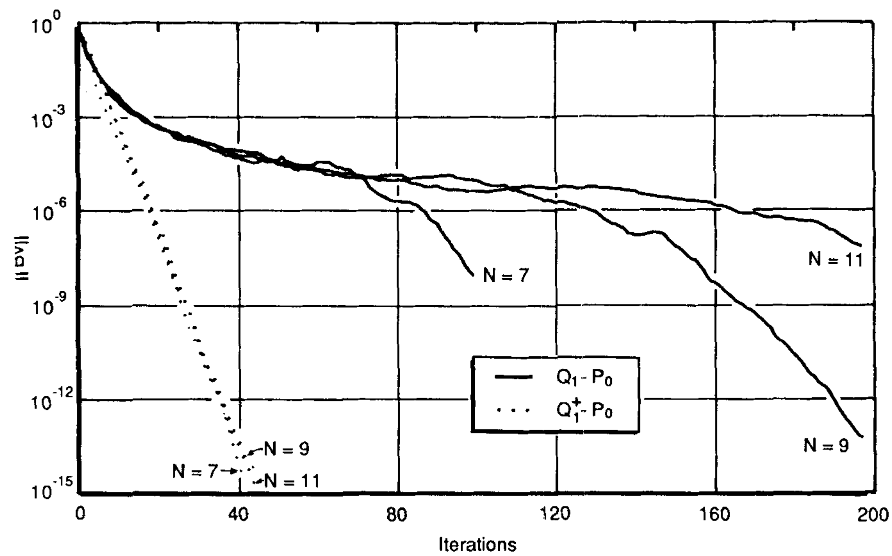

Figure 1. Convergence of the Uzawa algorithm (three meshes)

Table II. Influence of penalization on the convergence rate for the primal and dual problems (Q_1^+–P_1 element)

	$N=7$		$N=9$		$N=11$	
	$r=0$	$r=10$	$r=0$	$r=10$	$r=0$	$r=10$
CG (K_d)	0·34	0·96	0·33	0·92	0·33	0·92
OD (K_d)	0·11	0·62	0·12	0·60	0·12	0·60
PCG (K_p)	0·48	0·40	0·36	0·30	0·26	0·22

convergence rate (K_p) is also presented using the ICCG method with penalty parameter values of 0 and 10.

The convergence of the dual problem is faster with the conjugate gradient method than with the optimal descent method. It is interesting to note that, when using a small value of the penalty parameter ($r = 10$), the convergence rate increase for the dual problem is significantly larger than the convergence rate decrease for the primal problem. It can also be seen that increasing the problem size decreases the convergence rate of the primal problem but does not influence the convergence rate of the dual problem.

It is also interesting to compare the convergence of the dual problem for different types of elements. The convergence of the conjugate gradient–Uzawa algorithm is presented in Table III for the Q_1^+–P_1 element and for the P_2–P_1 element with continuous pressures,[6] two elements which satisfy the Brezzi condition. It can be observed that the convergence is significantly better with the Q_1^+–P_1 element. Furthermore, preconditioning the Uzawa algorithm by an incomplete factorization has proven as efficient as the technique developed by Cahouet and Chabard.[6]

As mentioned above, the Q_1–P_0 element is not well suited for Uzawa-type algorithms. Since it is the most commonly used three-dimensional element, its influence on the convergence properties of the incomplete Uzawa algorithm will be assessed.

Incomplete Uzawa algorithm

The convergence properties of the incomplete Uzawa algorithm are first studied with the Q_1–P_0. The convergence of $\| Bv \|$ and $\| R \|$ is presented in Figure 2 versus the number of descent iterations, n, for a $7 \times 7 \times 7$ element mesh without penalization ($r = 0$). In the case of $\| Bv \|$ the convergence history is shown for each primal iteration. It can be seen that $\| R \|$ converges linearly when the number of projection iterations is sufficiently large ($m = 100$). This is a property of the descent method applied to the primal problem. On the other hand, when the divergence is only partially projected ($m = 10$) a linear convergence down to 10^{-5} is first obtained, followed by a significant decrease of the convergence rate. The convergence rate of the algorithm is then limited by the convergence rate of $\| Bv \|$ for that problem.

A different convergence behaviour is obtained with the Q_1^+–P_1 element (see Figure 3). When only one projection iteration is made, the convergence of $\| R \|$ is limited by the convergence of the dual problem. It can be noted that the convergence is linear since, when $m = 1$, the dual problem is solved by an optimal descent method. With as few as three projection iterations, however, the

Table III. Comparison of K_d for the Uzawa algorithm with Q_1^+–P_1 and P_2–P_1 elements

	This work (Q_1^+–P_1)		Reference 6 (P_2–P_1)	
	$r = 0$	$r = 10$	Not precond.	Precond.
K_d	0·34	0·96	0·051	0·35

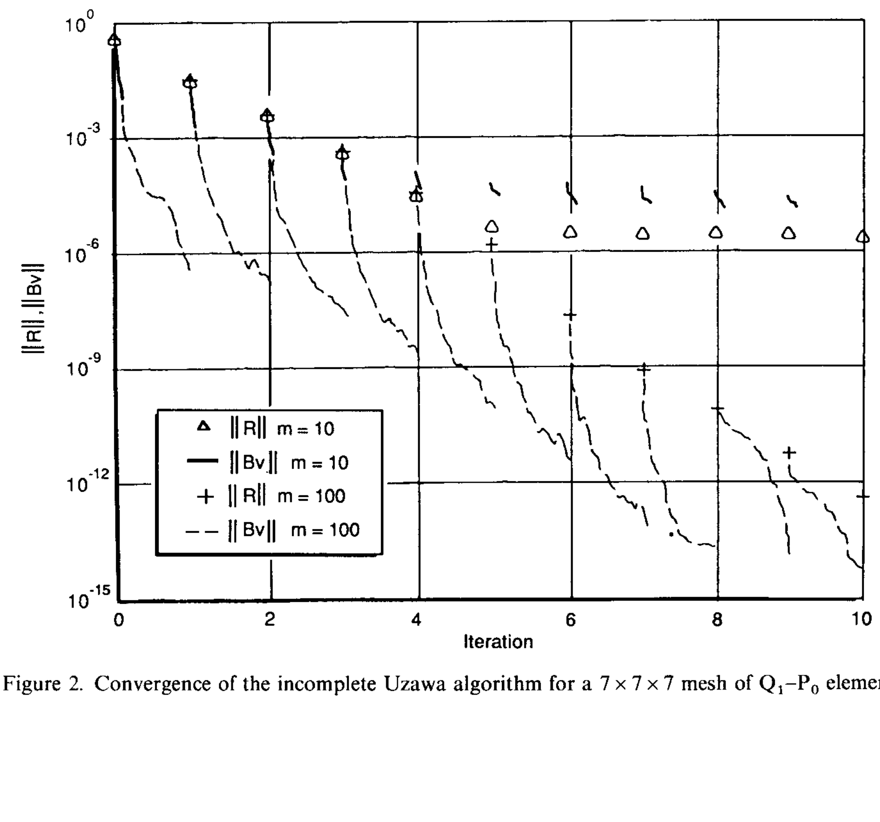

Figure 2. Convergence of the incomplete Uzawa algorithm for a $7 \times 7 \times 7$ mesh of Q_1–P_0 elements

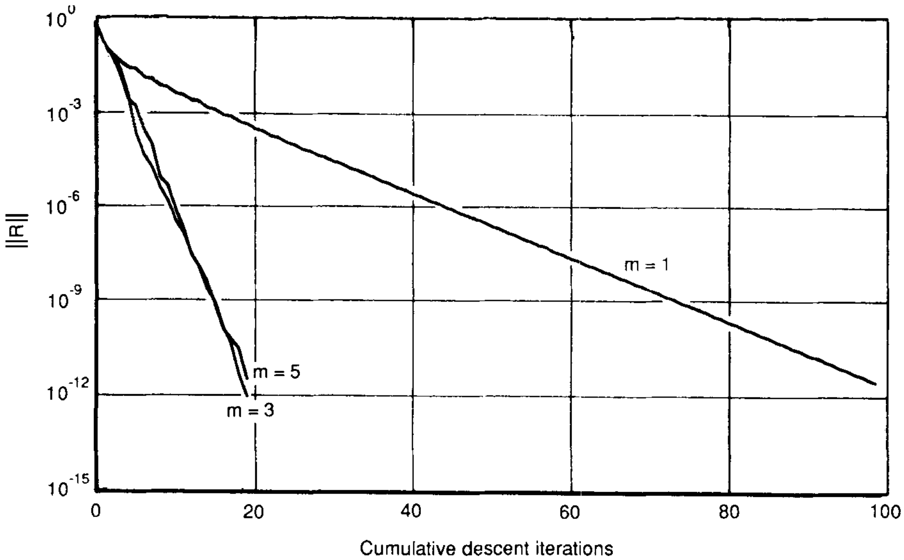

Figure 3. Convergence of $\|R\|$ for a $7 \times 7 \times 7$ mesh of $Q_1^+-P_1$ elements with the incomplete Uzawa algorithm

convergence becomes limited by the primal problem and does not depend on m. Again the convergence is linear since the primal problem is solved by an optimal descent method.

It should be pointed out that for both elements the projection step of the incomplete Uzawa algorithm requires most of the computational time. It is then important to minimize the number of projection iterations. The penalization of the matrix A then becomes an interesting alternative to improve the convergence of the projection step, a feature which appears essential for the Q_1-P_0 element. As mentioned above, the penalization degrades the convergence speed of the minimization step. This comes from the fact that the penalization is only partially transmitted to S_r^{-1} owing to the incomplete factorization. In fact, for a penalty parameter as low as 20 the original Uzawa algorithm does not converge for a $7 \times 7 \times 7$ mesh of Q_1-P_0 elements.

The effect of penalization on the convergence of $\|Bv\|$ versus the cumulative projection iterations (m) is shown in Figure 4 for the Q_1-P_0. For values of $\|Bv\|$ down to 10^{-5} the convergence is limited by the convergence of the primal problem and is slightly influenced by the penalization. When $\|Bv\|$ has converged further, the convergence is limited by the convergence of the dual problem and is significantly improved by using a penalty parameter value of 10.

The variation of the convergence behaviour with the problem size is shown in Figure 5. The convergence of $\|Bv\|$ is presented for $N = 7, 9$ and 11 with $m = 10$ and $r = 10$ for the Q_1-P_0, and with $m = 2$ and $r = 1$ for the $Q_1^+-P_1$. It can be observed that the convergence is linear for the $Q_1^+-P_1$ and decreases as the problem size increases. The convergence with the Q_1-P_0 exhibits the same trend as previously. Furthermore, the rate of convergence also decreases with the problem size.

For the Q_1-P_0 the optimal numerical strategy for solving the Stokes problem on a $7 \times 7 \times 7$ mesh is to penalize with $r = 10$ and to iterate 10 times on the projection step ($m = 10$). For larger problems the optimal strategy varies, and for $N = 9$ and 11 it consists of penalizing with $r = 10$ and increasing the number of projection iterations to 15 and 20 respectively.

The effect of penalization on the $Q_1^+-P_1$ element is not significant since the convergence of the dual problem is excellent and does not require a large number of projection iterations. The

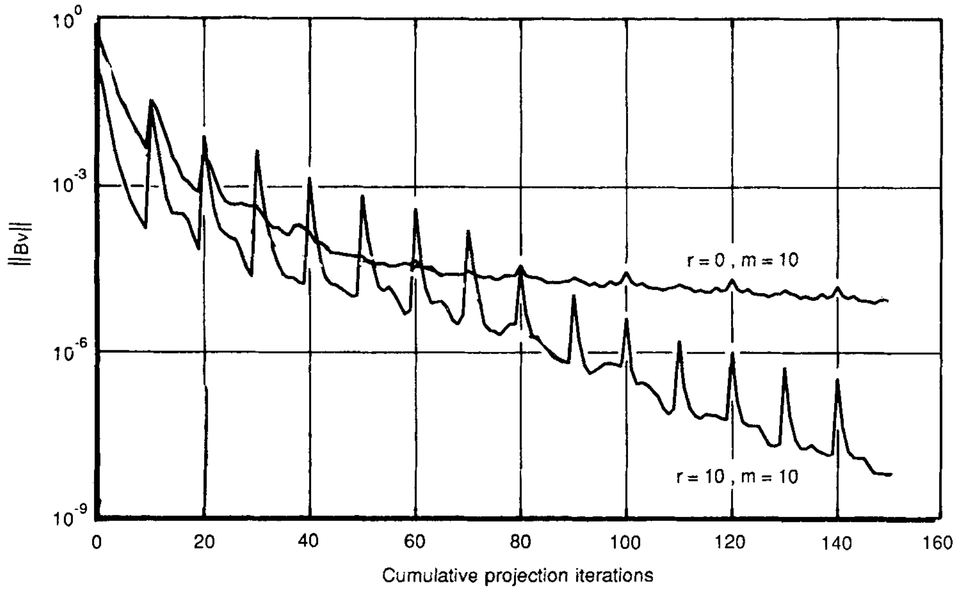

Figure 4. Influence of penalization on the convergence of $\|Bv\|$ for a $7\times7\times7$ mesh of Q_1–P_0 elements with the incomplete Uzawa algorithm

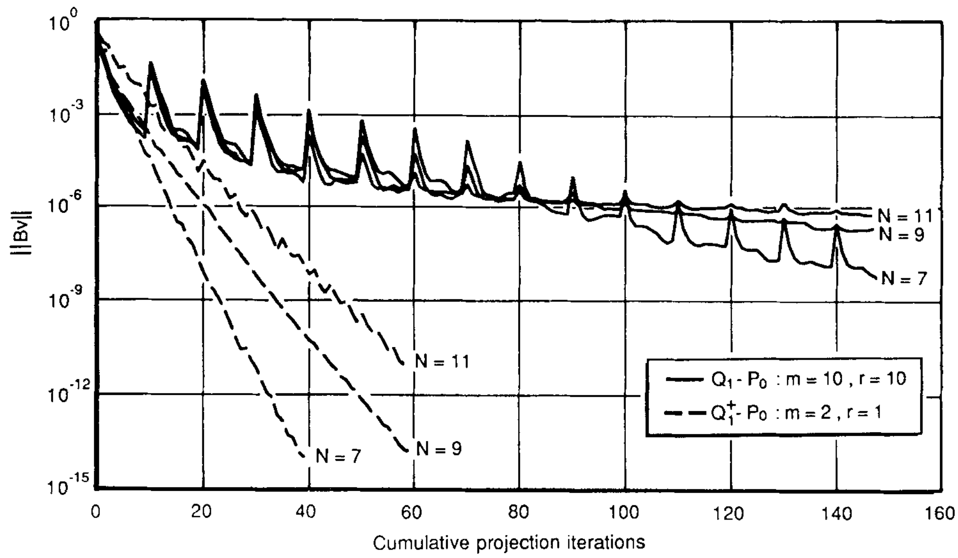

Figure 5. Convergence of the incomplete Uzawa algorithm (three meshes)

optimal strategy for a $7\times7\times7$ mesh is then to penalize slightly ($r = 1$) and to iterate twice on the dual problem ($m = 2$). For $N = 9$ and 11 the optimal strategy is to take $r = 2{\cdot}5$ and $m = 1$.

The performace of the overall algorithm can be assessed in Figure 6, where the variation of the CPU time to reach a residual of 10^{-10} is plotted versus the rank of the matrix system. All the tests were performed with the optimal penalty parameter and number of projection iterations. It can be seen that for the Q_1^+–P_1 element the computational labour of the incomplete Uzawa algorithm varies as $O(\text{NEQ}^{1.44})$ compared with $O(\text{NEQ}^{1.17})$ for the ICCG method and $O(\text{NEQ}^{2.33})$ for Gaussian elimination. For the Q_1–P_0 element the proposed algorithm is $O(\text{NEQ}^{1.65})$.

Another very important feature of iterative methods is their low storage requirements. Indeed, both the ICCG and the incomplete Uzawa algorithms are $O(\text{NEQ})$ while Gaussian elimination is $O(\text{NEQ}^{1.67})$.[15] These figures yield, for a matrix rank of 10^4, an improvement of one order of magnitude over Gaussian elimination for both the computational and storage requirements (see Table IV). It must be noted that the largest problem solved with the incomplete Uzawa algorithm was for a matrix of rank 31 000.

Navier–Stokes problems

In the case of the Navier–Stokes equations the matrix is not symmetric and the maximization in q of (9) is no longer equivalent to solving (13). To preserve the symmetry of the matrix, the advective terms introduced by the Navier–Stokes equations have been taken into account by a fixed-point method. This has been done by introducing the advective terms in the definition of the residual R_i for the primal problem of the incomplete Uzawa algorithm. The same lid-driven cavity

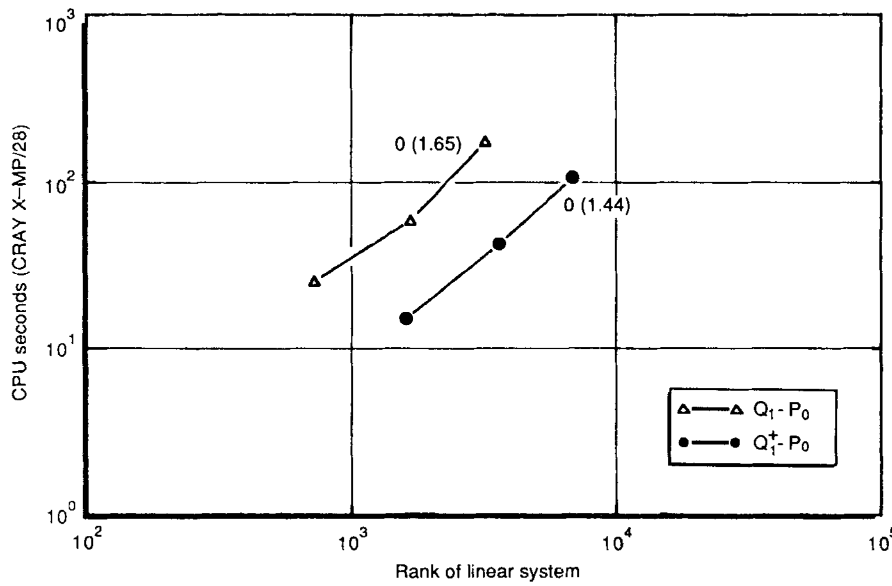

Figure 6. Variation of the CPU time with the problem size for the incomplete Uzawa algorithm

Table IV. Comparison of computer resources (4000 linear elements, 10000 equations)

	Gaussian elimination	Incomplete Uzawa algorithm
CPU time (Cyber 835)	30 h	3 h
Storage	80 Mbytes	8 Mbytes

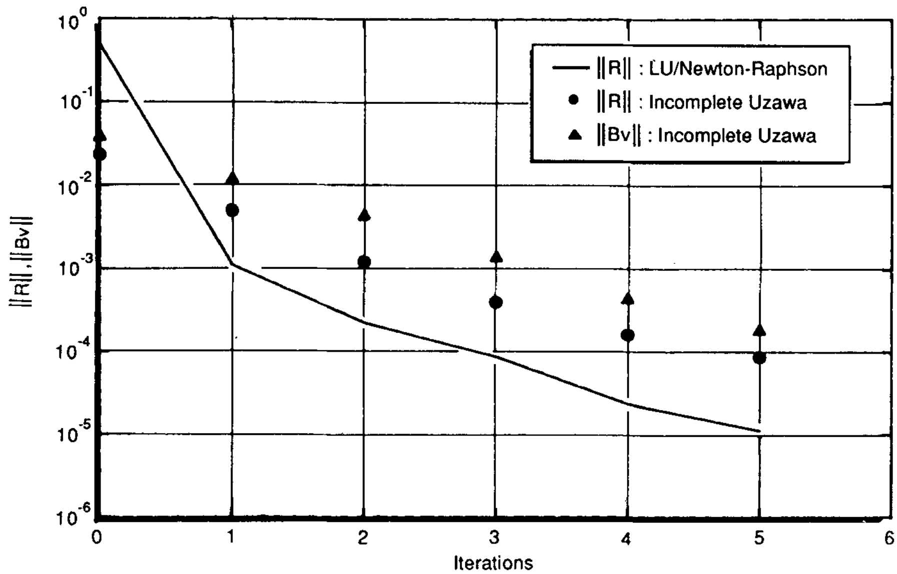

Figure 7. Comparison of convergence behaviour for different algorithms at $Re = 50$

problem is considered to study the convergence of the incomplete Uzawa algorithm for the Navier–Stokes equations, the domain being discretized with a mesh of 800 elements (1863 equations). Solutions have been computed for the following Reynolds numbers: $Re = 25, 50, 75$ and 100. The convergence of $\|R\|$ and $\|Bv\|$ for $Re = 50$, starting from a solution at $Re = 25$, is presented in Figure 7 for different algorithms. It can be seen that the convergence of the incomplete Uzawa algorithm follows the same trend as the convergence of the standard methodology (penalty technique, LU factorization, quasi-Newton linearization technique). The difference between the two convergence curves is due to the penalty term in the expression of the residual.

The incomplete Uzawa algorithm could give solutions up to $Re = 200$. For larger Reynolds numbers the fixed-point method for the advective terms diverges. A cure could be to treat the convective terms through the use of an operator-splitting method, which combines the computation of the hyperbolic part of the equation by the method of characteristics, and the computation of a Stokes problem.[5,6] Another strategy could be to solve an unsteady problem while still treating the advective terms by a fixed-point method. Preliminary tests have shown that the incomplete Uzawa algorithm converges very well in the latter case, and it was possible to obtain a solution at $Re = 3300$ for the lid-driven cubic cavity problem. A detailed convergence analysis is under way and will be published later.

CONCLUSION

We have presented in this paper an iterative implementation of the Uzawa algorithm for the solution of 3D fluid flow problems. This new algorithm, called the incomplete Uzawa algorithm, is a hybrid of the standard Uzawa algorithm and the Arrow–Hurwicz algorithm. Its advantage over the former lies in the use of an iterative solver for the resolution of the primal problem, yielding a drastic reduction of the computational labour. On the other hand, the incomplete Uzawa algorithm allows a full projection of the velocity field onto the divergence-free subspace (dual problem), which is not the case with the Arrow–Hurwicz algorithm. By carefully selecting the level of penalization in the primal problem and by using an incomplete LDL^T preconditioner, it was possible to solve the Navier–Stokes equation with a computational labour of order $NEQ^{1\cdot 44}$.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the NSERC and FCAR for their financial assistance and AES for providing access to a CRAY X-MP/28 supercomputer. Thanks are also directed to Control Data Canada for the generous grant of computing time on several Cyber 180 mainframes.

REFERENCES

1. P. Concus, G. H. Golub and D. P. O'Leary, *Sparse Matrix Computations*, Academic Press, New York, 1976.
2. D. S. Kershaw, *J. Comput. Phys.*, **26**, 43 (1978).
3. T. A., Manteuffel, *Math. Comput.*, **24**, 473 (1980).
4. M. P. Robichaud and P. A. Tanguy, *Int. j. numer. methods eng.*, **24**, 447 (1987).
5. J. P. Chabard, O. Daubert, J. P. Grégoire and P. Hemmerich, in *Numerical Methods for Laminar and Turbulent Flows*, Montréal, 1987, p. 673.
6. J. Cahouet and J. P. Chabard, *Int. j. numer. methods fluids*, **8**, 869 (1988).
7. R. Aboulaich, M. Fortin, M. P. Robichaud and P. A. Tanguy, in *Finite Element Methods in Flow Problems*, Antibes, 1986, p. 345.
8. M. Fortin, *Int. J. numer. methods fluids*, **4**, 347 (1981).
9. M. Fortin and A. Fortin, *Commun. Appl. Numer. Methods*, **1**, 205 (1985).
10. M. P. Robichaud and P. A. Tanguy, *Commun. Appl. Numer. Methods*, **3**, 223 (1987).
11. M. Fortin and R. Glowinski, *Méthodes de Lagrangien Augmenté*, Dunod, Paris, 1982.
12. K. J. Arrow, L. Hurwicz and H. Uzawa, *Studies in Non-Linear Programming*, Stanford University Press, 1958.
13. S. O. Wille, in *Numerical Grid Generation in Computational Fluid Dynamics*, Swansea, 1986, p. 57.
14. R. Verfurth, *IMA J. Numer. Anal.*, **4**, 441 (1984).
15. O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems*, Academic Press, New York, 1984.
16. J. T. Oden and O. P. Jacquotte, *Comput. Methods Appl. Mech. Eng.*, **43**, 231 (1984).