

## THE APPLICATION OF QUASI-NEWTON METHODS IN FLUID MECHANICS

M. S. ENGELMAN<sup>†</sup>, G. STRANG<sup>†</sup> AND K.-J. BATHE<sup>‡</sup>

*Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A.*

### SUMMARY

The use of quasi-Newton methods is studied for the solution of the nonlinear finite element equations that arise in the analysis of incompressible fluid flow. An effective procedure for the use of Broyden's method in finite element analysis is presented. The quasi-Newton method is compared with the commonly employed successive substitution and Newton-Raphson procedures, and it is concluded that the use of Broyden's method can constitute an effective solution strategy.

### INTRODUCTION

The finite element method has found increased use and wider acceptance for the solution of the Navier-Stokes equations governing viscous incompressible fluids. Inevitably, the analysis requires the solution of a discrete system of nonlinear equations, and this often represents the most time-consuming stage of the analysis. The choice of algorithm at this stage, its cost and its reliability, can ultimately govern or even limit the size of the finite element model. Therefore, there is substantial interest in the development of improved procedures for solution of the nonlinear equations.

To date, the methods of 'successive substitution' and Newton-Raphson seem to have been the most widely used in finite element programs for analysis of fluid response.<sup>1,2</sup> Gartling *et al.*<sup>3</sup> compared these methods and also demonstrated that techniques based on a symmetric coefficient matrix are not efficient for fluid flow problems, since, except for very low Reynolds numbers, the convective (nonlinear) term is dominant and must be included in the coefficient matrix.

A major drawback of both successive substitution and full Newton-Raphson is that the non-symmetric coefficient matrix must be computed and factored in each iteration. These computations can be prohibitively expensive for very large systems if convergence is not reached within a reasonable number of iterations. Successive substitution is a linearly convergent algorithm, whereas Newton-Raphson normally converges quadratically, but often with a smaller radius of convergence. In addition, neither method is very effective in the analysis of flows of high Reynolds' number, so that there is a need for the study of alternative solution techniques.

Over the past few years a new group of algorithms has emerged for the solution of nonlinear equations. These are known as quasi-Newton, variable metric, matrix update or modification methods. They have been used extensively in optimization, and were first applied in finite

<sup>†</sup> Department of Mathematics.

<sup>‡</sup> Department of Mechanical Engineering.

element analysis by Matthies and Strang<sup>4</sup> in collaboration with Bathe and Cimento<sup>5</sup> using the ADINA structural analysis program.<sup>6</sup> Those applications to solid or structural mechanics led to symmetric positive-definite coefficient matrices.

Our objective in this paper is to present the results of a comparative study of different techniques for the solution of the nonlinear, non-symmetric equations in fluid mechanics. We begin with a brief summary of the finite element formulation and a description of the solution strategies that we studied, with some emphasis on the implementation of a quasi-Newton method. Although the update rule that we use is well established, we have constructed a new recursion formula for the computation of the successive search directions. We then describe our numerical experiments and discuss their implications for the finite element analysis of fluids.

### SOLUTION TECHNIQUES

The solution strategies and their implementation are quite general. To illustrate their use we concentrate on the steady-state flow of a viscous incompressible fluid, governed by the dimensionless Navier-Stokes equations:

$$u_j u_{i,j} = -p_{,i} + \frac{1}{\text{Re}} u_{i,jj} + f_i \quad (1)$$

As usual,  $x_i$ ,  $u_i$ ,  $f_i$  and  $p$  are the components of the spatial position vector, the Eulerian velocity vector, the body force vector and the pressure, and  $\text{Re}$  is the Reynolds' number. Equation (1), together with the continuity equation,

$$u_{i,i} = 0 \quad (2)$$

and the appropriate boundary conditions, fully describe the steady laminar flow of a Newtonian incompressible fluid. As boundary conditions we may prescribe velocities and/or surface tractions.

The finite element method can be applied to these equations in several different ways. The approach employed here is a Galerkin procedure, together with a penalty function formulation proposed in Reference 7.

In the penalty function approach the continuity equation is replaced by

$$u_{i,i} = \epsilon p \quad (3)$$

and equation (3) is then used to eliminate the pressure variable from the equation of motion. The finite element discretization is performed using the 9-node isoparametric element with selective reduced integration for the penalty term, as discussed in detail by Bercovier and Engelman.<sup>7</sup> This formulation leads to a set of global discretized equations of motions for the fluid of the form

$$\mathbf{K}(\mathbf{u})\mathbf{u} = \mathbf{F} \quad (4)$$

The vector  $\mathbf{u}$  represents the unknown nodal degrees-of-freedom and  $\mathbf{F}$  includes prescribed body forces and surface tractions. The coefficient matrix  $\mathbf{K}(\mathbf{u})$  is a sparse banded matrix and it decomposes naturally into

$$\mathbf{K}(\mathbf{u}) = \mathbf{K}_{\text{con}}(\mathbf{u}) + \mathbf{K}_{\text{dif}}$$

where  $\mathbf{K}_{\text{con}}(\mathbf{u})$  is the convective matrix which arises from the term  $u_j u_{i,j}$  in equation (1) and  $\mathbf{K}_{\text{dif}}$  is the diffusion matrix arising from  $p_{,i}$  and  $u_{i,jj}$ .  $\mathbf{K}_{\text{con}}(\mathbf{u})$  is non-symmetric and  $\mathbf{K}_{\text{dif}}$  is symmetric. For Reynolds' numbers greater than unity the convective matrix dominates; this causes the

nonlinear character of the equations to become dominant, making the choice of algorithm to solve equation (3), and its convergence, a key issue. Two of the most common choices are:

1. *Successive substitution (Picard iteration)*, which is a fixed point iteration of the form

$$\mathbf{K}(\mathbf{u}_i)\mathbf{u}_{i+1} = \mathbf{F} \tag{5}$$

The nonlinearity is evaluated at the known iterate  $\mathbf{u}_i$ , and a linear system is solved at each step.

2. *Newton–Raphson*, which may be described as follows:

$$\mathbf{u}_{i+1} = \mathbf{u}_i - \mathbf{J}^{-1}(\mathbf{u}_i)\mathbf{R}(\mathbf{u}_i) \tag{6}$$

where  $\mathbf{R}(\mathbf{u}) = \mathbf{K}(\mathbf{u})\mathbf{u} - \mathbf{F}$  is the residual force vector and  $\mathbf{J}(\mathbf{u})$  is the Jacobian of the system of equations. In practice, this iteration is organized as

$$\mathbf{J}(\mathbf{u}_i)\Delta\mathbf{u}_i = -\mathbf{R}(\mathbf{u}_i) \quad \mathbf{u}_{i+1} = \mathbf{u}_i + \Delta\mathbf{u}_i \tag{7}$$

Both methods converge for a fair range of Reynolds’ numbers with, as expected, somewhat more rapid convergence for Newton–Raphson. The major disadvantage of both methods is that a complete factorization of an unsymmetric system is required at each iteration.

The Newton–Raphson method belongs to a more general class of iterative procedures for solving the  $n$  equations  $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ :

$$\mathbf{u}_{i+1} = \mathbf{u}_i - s_i \mathbf{H}_i \mathbf{f}_i \tag{8}$$

where  $\mathbf{f}_i = \mathbf{f}(\mathbf{u}_i)$ ,  $\mathbf{H}_i$  is an  $n \times n$  matrix determined by the particular method employed and  $s_i$  is a scaling (or acceleration) factor which may be introduced to reduce  $\mathbf{f}_{i+1}$ . In this case an additional iterative procedure (line search) is necessary to determine  $s_i$ .

In our context,  $\mathbf{f}(\mathbf{u}) = \mathbf{R}(\mathbf{u})$ , i.e.

$$\mathbf{f}(\mathbf{u}) = \mathbf{K}(\mathbf{u})\mathbf{u} - \mathbf{F} \tag{9}$$

so that (8) becomes

$$\mathbf{u}_{i+1} = \mathbf{u}_i - s_i \mathbf{H}_i \mathbf{R}(\mathbf{u}_i) \tag{10}$$

Setting  $\mathbf{H}_i = \mathbf{J}^{-1}(\mathbf{u}_i)$  and  $s_i = 1$  yields the Newton–Raphson method. If we set  $\mathbf{H}_i = \mathbf{J}(\mathbf{u}_0)^{-1}$  and  $s_i = 1$  we obtain the modified Newton method. In this algorithm the need to refactor at each iteration is obviated; the initial Jacobian  $\mathbf{J}(\mathbf{u}_0)$  is assembled, factored and stored. Then at each cycle we calculate  $\Delta\mathbf{u}_i = -\mathbf{J}^{-1}(\mathbf{u}_0)\mathbf{R}(\mathbf{u}_i)$ ; given the triangular factors of  $J$ , say with bandwidth  $m$ , this ‘back-substitution step’ requires only  $2mn$  operations. This saving may come, however, at the expense of slower convergence or even divergence.

Finally we come to a sub-class of the algorithms described by (8), known as quasi-Newton methods. They derive from the suggestion of Davidon<sup>8</sup> to update  $\mathbf{H}_i$  in a simple manner after each iteration rather than recompute it entirely (Newton–Raphson) or leave it unchanged (modified Newton). The idea has been under active development in numerical optimization by Powell, Fletcher and Broyden, among many others; see the survey by Dennis and Moré.<sup>9</sup> It was first applied to finite element analysis by Matthies and Strang<sup>4</sup> and a quasi-Newton algorithm (the BFGS update for symmetric problems) is now a standard option in the ADINA structural analysis program.

We write these algorithms in the form

$$\mathbf{u}_{i+1} = \mathbf{u}_i - s_i \mathbf{K}_i^{-1} \mathbf{f}_i \tag{11}$$

$$\mathbf{K}_{i+1} = \mathbf{K}_i + \Delta\mathbf{K}_i \tag{12}$$

and require that the  $\mathbf{K}_i$  satisfy the 'quasi-Newton equation' or 'secant condition':

$$\mathbf{K}_i(\mathbf{u}_i - \mathbf{u}_{i-1}) = \mathbf{f}_i - \mathbf{f}_{i-1}. \quad (13)$$

Introducing the notation  $\delta_i = \mathbf{u}_i - \mathbf{u}_{i-1}$ ,  $\gamma_i = \mathbf{f}_i - \mathbf{f}_{i-1}$  and  $\mathbf{d}_i = \mathbf{K}_i^{-1}\mathbf{f}_i$ , we can write equations (11)–(13) as

$$\mathbf{u}_{i+1} = \mathbf{u}_i - s_i \mathbf{d}_i \quad (14)$$

$$\mathbf{K}_{i+1} = \mathbf{K}_i + \Delta \mathbf{K}_i \quad (15)$$

$$\mathbf{K}_i \delta_i = \gamma_i \quad (16)$$

The vector  $\mathbf{d}_i$  is the search direction; for the Newton–Raphson method the search direction is  $\mathbf{d}_i = \mathbf{J}(\mathbf{u}_i)^{-1} \mathbf{R}(\mathbf{u}_i)$ , and for modified Newton it is  $\mathbf{d}_i = \mathbf{J}(\mathbf{u}_0)^{-1} \mathbf{R}(\mathbf{u}_i)$ .

The name quasi-Newton arises from the fact that the  $\mathbf{K}_i$  can be thought of as approximations to the Jacobian at  $\mathbf{u} = \mathbf{u}_i$ . If the rank of  $\Delta \mathbf{K}_i$  is  $r$ , then the scheme in equation (14)–(16) is called a direct update of rank  $r$ ; in practice, the update is either of rank one ( $\mathbf{K}_i = \mathbf{w}_i \mathbf{z}_i^T$  for some vectors  $\mathbf{w}_i$  and  $\mathbf{z}_i$ ) or rank two. Although equations (14)–(16) still require the solution of  $n$  equations at each iteration, there are a number of ways to avoid refactorization and a new elimination. In fact, the update can be expressed directly as a correction to the inverse:

$$\mathbf{K}_{i+1}^{-1} = \mathbf{K}_i^{-1} + \Delta \mathbf{K}_i^{-1} \quad \text{rank } \Delta \mathbf{K}_i^{-1} = \text{rank } \Delta \mathbf{K}_i \quad (17)$$

For a complete discussion of the choice and the calculation of direct and inverse updates we refer the reader to Dennis and Moré.<sup>9</sup> For convenience, we will use only inverse update forms.

Since the field of optimization has provided the impetus for quasi-Newton methods, research has concentrated on systems which have a symmetric positive-definite matrix  $\mathbf{K}_i$ . The most successful quasi-Newton update in this case is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update which has the property that if  $\mathbf{K}_{i-1}^{-1}$  is positive-definite symmetric then so is  $\mathbf{K}_i^{-1}$ . The update can be written in a number of equivalent forms, one being

$$\mathbf{K}_i^{-1} = (\mathbf{I} - \rho_i \delta_i \gamma_i^T) \mathbf{K}_{i-1}^{-1} (\mathbf{I} - \rho_i \gamma_i \delta_i^T) + \rho_i \delta_i \delta_i^T, \quad \rho_i = \frac{1}{\delta_i^T \gamma_i} \quad (18)$$

A more natural update for unsymmetric coefficient matrices is that of Broyden,<sup>10</sup> which was one of the first quasi-Newton formulae to be proposed. No constraints of symmetry or positive-definiteness are imposed on the updated matrix. Broyden's update in inverse form is

$$\mathbf{K}_i^{-1} = \mathbf{K}_{i-1}^{-1} + \frac{(\delta_i - \mathbf{K}_{i-1}^{-1} \gamma_i) \delta_i^T}{\delta_i^T \mathbf{K}_{i-1}^{-1} \gamma_i} \mathbf{K}_{i-1}^{-1} \quad (19)$$

We have experimented with successive substitution, Newton–Raphson, modified Newton and Broyden's update to solve equations (1) and (2). The modified Newton and Broyden's update were implemented both with  $s_i = 1$  and with a line search to reduce the quantity  $\mathbf{d}_i^T \mathbf{f}_{i+1}$ . As noted earlier, in many cases an appropriate choice of  $s_i$  can accelerate convergence. However, care must be taken to avoid many evaluations of  $\mathbf{R}(\mathbf{u})$ , since an inefficient line search can reduce the cost-effectiveness of the entire solution algorithm.

We should note that there are many variations of the basic successive substitution and Newton–Raphson methods. Successive substitution is often accelerated by the modification

$$\mathbf{K}(\mathbf{u}_i) \mathbf{u}^* = \mathbf{F}; \quad \mathbf{u}_{i+1} = \alpha \mathbf{u}_i + \beta \mathbf{u}^*$$

where  $\alpha$ ,  $\beta$  can be calculated by a variety of means<sup>10</sup> and Newton–Raphson can be more effective with an incremental loading strategy.<sup>12</sup> In the present study we limited our interest to the basic properties of the different algorithms.

ALGORITHMIC IMPLEMENTATION

A number of difficulties would be encountered with the standard quasi-Newton implementation in a finite element framework. If problems of realistic size are to be analysed, the Jacobian cannot be retained entirely in core and must be stored in blocks on low-speed storage in a reduced storage mode. The update in equation (17) becomes difficult, and even if performed out of core it could destroy the sparseness pattern and the storage mode of the original matrix.

The basic idea which enables us to overcome these difficulties is to calculate at each iterative cycle the updated  $\mathbf{K}_i^{-1}$  from the original  $\mathbf{K}_0^{-1}$ . This is less cumbersome than it appears; in fact, it leads to a comparatively simple and concise implementation. We give here the details for the Broyden update; a description of the finite element implementation of the BFGS update can be found in References 4 and 5.

The algorithm begins with the choice of  $\mathbf{K}_0$ . This may be the initial Jacobian  $\mathbf{J}(\mathbf{u}_0)$ , where  $\mathbf{u}_0$  is an initial approximation—generally chosen to be the solution of the associated Stokes problem. The LU factorization of  $\mathbf{K}_0$  is computed and stored in the process of solving  $\mathbf{K}_0\mathbf{d}_0 = \mathbf{R}(\mathbf{u}_0)$ , which also gives the initial search direction  $\mathbf{d}_0$ . We then estimate, with a line search if necessary, the point  $s_0$  at which the scalar function  $G(s) = \mathbf{d}_0^T \mathbf{R}(\mathbf{u}_0 - s_0\mathbf{d}_0)$  is zero. The new guess is  $\mathbf{u}_1 = \mathbf{u}_0 - s_0\mathbf{d}_0$ . In general, let us assume that we have just computed  $\mathbf{u}_i$  and  $\mathbf{R}(\mathbf{u}_i)$  using  $\mathbf{u}_{i-1}, s_{i-1}, \mathbf{d}_{i-1}$ . The algorithm for the new search direction  $\mathbf{d}_i$  is as follows:

Given  $\mathbf{u}_i, \mathbf{u}_{i-1}, s_{i-1}, \mathbf{d}_{i-1}$  and  $\mathbf{R}(\mathbf{u}_i)$ ,

1. Calculate  $\mathbf{q}_1 = \mathbf{K}_0^{-1}\mathbf{R}(\mathbf{u}_i)$  by forward elimination and back-substitution.
2. For  $j = 1, \dots, i - 1$

$$\begin{aligned} &\text{Retrieve } \rho_j, \delta_j \text{ and } \mathbf{r}_j \\ &\text{Compute } \mathbf{q}_{j+1} = \mathbf{q}_j + \rho_j(\delta_j - \mathbf{r}_j)\delta_j^T \mathbf{q}_j \end{aligned}$$

3. Form and store

$$\begin{aligned} \mathbf{r}_i &= \mathbf{q}_i - \mathbf{d}_{i-1} \\ \delta_i &= \mathbf{u}_i - \mathbf{u}_{i-1} = -s_{i-1}\mathbf{d}_{i-1} \\ \rho_i &= \frac{1}{\delta_i^T \mathbf{r}_i} \end{aligned}$$

4. Form

$$\mathbf{d}_i = \mathbf{q}_i + \rho_i(\delta_i - \mathbf{r}_i)\delta_i^T \mathbf{q}_i$$

This algorithm requires the calculation and storage of two vectors of dimension  $n$  for each iteration. Most quasi-Newton programs would actually carry out the updating at each step—not on the  $\mathbf{K}_i$  which are never explicitly calculated but on the  $\mathbf{L}_i$  and  $\mathbf{U}_i$  of the Gauss factorization  $\mathbf{K}_i = \mathbf{L}_i\mathbf{U}_i$ . In this case there is no need to save the vectors  $\delta_i$  and  $\mathbf{r}_i$  after the update, but a very serious drawback is that the new factors  $\mathbf{L}_i$  and  $\mathbf{U}_i$  have to be computed and their original sparsity can be lost. Therefore, we prefer to keep and reintroduce the updating vectors, up to a limit imposed by the user. We can estimate the number of operations when  $m$  is the mean bandwidth of  $\mathbf{K}_0$  (in practice, we use a skyline storage scheme with variable bandwidth). The operation count to compute Broyden's new search direction  $\mathbf{d}_N$ , at iteration  $N$ , is  $(2m + 3N + 2)n$ . In practice,  $m$  greatly dominates  $N$ , since we set an upper limit on the number of factors stored, say 5 or 10. When this upper limit is reached we have one of two options: (a) to update the Jacobian matrix and restart the algorithm with  $\mathbf{u}_0 = \mathbf{u}_n$ , or (b) to shift the updating vectors one position downwards (thus losing the first pair) and continue.

## NUMERICAL EXPERIMENTS

A convergence study is best conducted by comparing numerical solutions to known exact solutions, rather than through inference from a sequence of numerical results. However, except for trivial cases in which inertial terms vanish there are very few analytical solutions of the full Navier–Stokes equations that are suitable for such a study. The most suitable available example, in which inertial terms interact non-trivially with viscous terms, appears to be the Hamel problem in which fluid flows between intersecting infinite plates (Figure 1). Hamel flow has been studied numerically by a number of authors using finite element techniques [3], [13] and an excellent theoretical discussion is given by Batchelor [14].

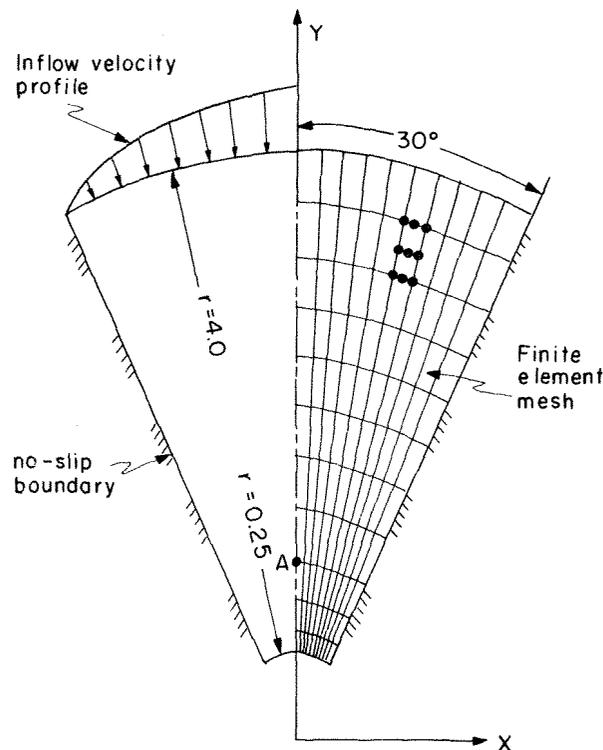


Figure 1. Hamel problem

For the ideal Hamel problem, with infinite plates, the velocity profile is self-similar at all radii. Each fluid particle moves radially towards the sink (or away from the source) at the plate juncture, and either the inflow or outflow versions of the problem could be solved. However, since only a finite wedge can be modelled with finite elements, a set of computational boundary conditions were applied on the upstream end of the wedge and zero traction boundary conditions on the downstream end. Since an unknown shear stress distribution is present at the downstream end, the computational boundary condition introduces a disturbance that propagates a small distance upstream. This distance is small compared to the overall grid dimensions, so that ample undisturbed regions remain for comparison purposes with the exact solution.

In our experiments the flow was taken to be inward (sink flow). Inflow velocities were applied at the radius  $r = 4$  and exit boundary conditions at  $r = 1/4$ . The flow centreline was forced to be a symmetry plane and no-slip velocity conditions were prescribed at the plate wall, as shown in Figure 1. The Reynolds' number is  $Re = (\alpha u_0 r) / \nu$ , where  $\nu$  is the kinematic viscosity,  $\alpha$  the wedge half-angle,  $r$  the radial co-ordinate and  $u_0$  the velocity along the centreline at  $r$ . Note that  $Re$  is constant in the true solution, since  $u_0 r$  is constant.

The finite element mesh used consisted of 110 elements, spaced uniformly in the circumferential direction and non-uniformly in the radial direction in order to create a nearly constant product of velocity and radius (Figure 1). For half-angle  $30^\circ$ , the exact solution was obtained using the computer program of Gartling.<sup>3</sup>

The Hamel problem, although fully nonlinear, has a somewhat 'regular' solution. Therefore, the different solution methods, although differing in rates of convergence, are reasonably well behaved even for large Reynolds' numbers. For this reason we chose a second problem in which numerical difficulties have been encountered—the classical wall-driven cavity problem. This problem is difficult to analyse at higher Reynolds' numbers because of the singularities at the top corner nodes. For the cavity problem we worked on a regular  $10 \times 10$  grid (Figure 2).

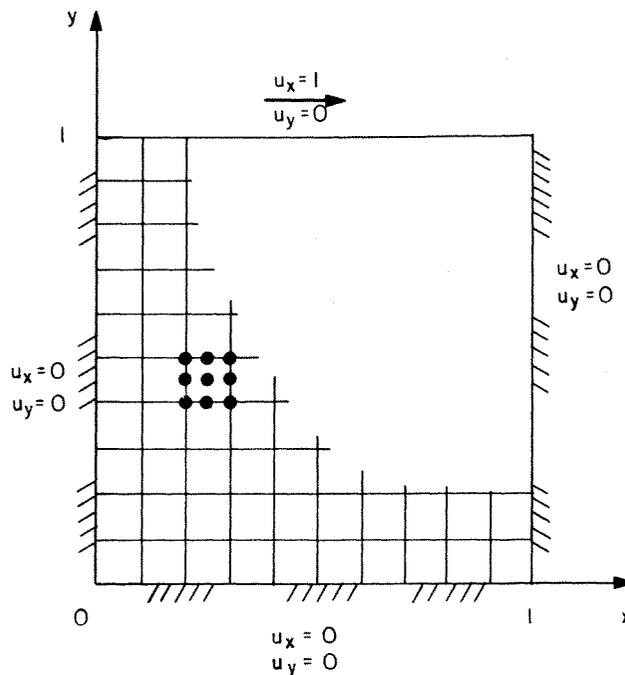


Figure 2. Cavity problem

An important aspect of the iterative solution is also the choice of an appropriate convergence criterion. This choice is more difficult when line searches are part of the algorithm. The standard test

$$\frac{\|\mathbf{u}_i - \mathbf{u}_{i-1}\|}{\|\mathbf{u}_i\|} \leq \epsilon_1, \quad \|\mathbf{u}_i\| = \text{Euclidean length of } \mathbf{u}_i$$

is no longer sufficient, since  $\|\mathbf{u}_i - \mathbf{u}_{i-1}\|$  may be small only because  $s_i$  is small. This criterion is supplemented by a condition on the residual forces:

$$\frac{\|\mathbf{R}(\mathbf{u}_i)\|}{\|\mathbf{R}_0\|} \leq \varepsilon_2$$

where  $\mathbf{R}_0$  is some reference force vector, for example  $\mathbf{R}(\mathbf{u}_0)$ . In our studies we used  $\varepsilon_1 = \varepsilon_2 = 10^{-3}$ . This small tolerance was chosen in order to differentiate more clearly between the different solution methods.

The Hamel flow solution was computed for two Reynolds' numbers,  $Re = 61$  and  $Re = 1088$ . The number of iterations required by each method is tabulated in Table I. Because the Hamel solution is regular, all methods converged quickly and neither modified Newton nor Broyden's update required line searches even at  $Re = 1088$ . For successive substitution and Newton-Raphson we reproduce the results of Gartling *et al.*<sup>3</sup>

Table I. Iterations used in Hamel problem

Method	Re = 61	Re = 1088
Successive substitution	9	11
Newton-Raphson	3	4
Modified Newton	4	7
Broyden	4	7

To obtain a clearer understanding of the behaviour of the different algorithms, the vertical (radial) velocity at the point  $r = 1$ ,  $\theta = 0$  (point A in Figure 1), was plotted as a function of the iteration number; see Figure 3(a) for  $Re = 61$  and Figure 3(b) for  $Re = 1088$ . An important observation is that both modified Newton and Broyden's update share the general characteristics of the Newton-Raphson method, as against the sharply oscillatory behaviour of successive substitution.

A quite different picture emerges for the cavity flow. Four Reynolds' numbers were studied,  $Re = 1, 100, 400, 1000$ . The iteration counts and the relative execution times are given in Table II. For Reynolds' numbers 1 and 100 all methods perform more or less equivalently, with slightly faster convergence for the Newton-Raphson method. The execution times for  $Re = 100$  indicate the possible savings, even in a few iterations, when the stiffness matrix is not reformed at each step.

The increasing dominance of the convective term is reflected in the results for  $Re = 400$ . The modified Newton algorithm now converges very slowly and for  $Re = 1000$  failed to converge. Broyden's update, as shown in Table II, was implemented using four different strategies:

1. The updating vectors are shifted one downwards after 10 iterations.
2. The updating vectors are shifted one downwards after 5 iterations.
3. The Jacobian matrix is reformed after 10 iterations.
4. The Jacobian matrix is reformed after 5 iterations.

The results are markedly different; after reformation of the Jacobian stiffness matrix there is relatively rapid convergence. It is known that Newton-based methods display quadratic convergence within a certain radius of convergence; if  $\mathbf{u}_i$  is within this radius at the time of reformation, then such rapid convergence would be expected. The poor performance of

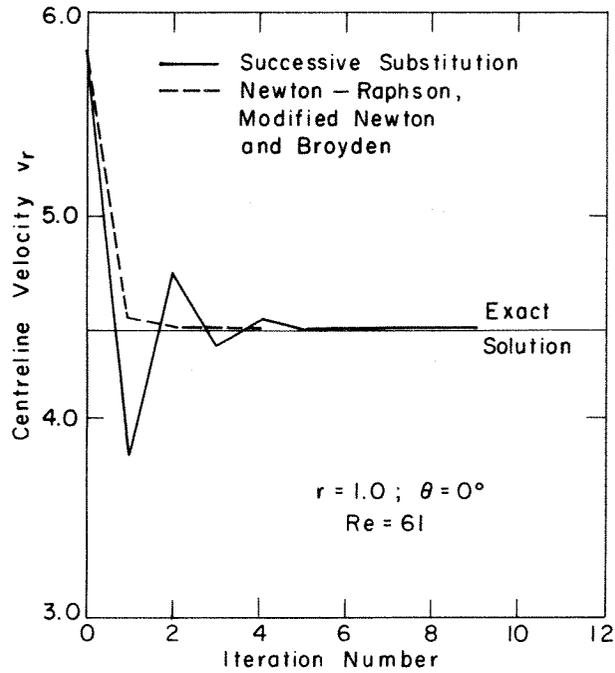


Figure 3(a). Centreline velocity vs. iteration number,  $r = 1.0$ ,  $\theta = 0^\circ$ ,  $Re = 61$

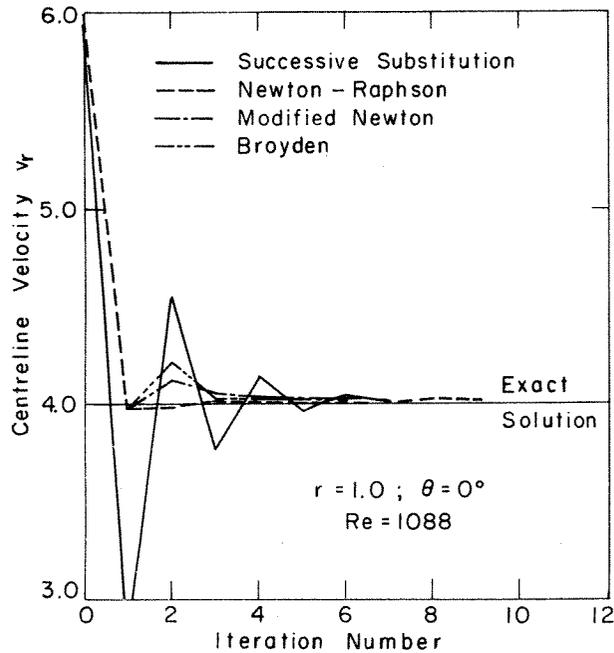


Figure 3(b). Centreline velocity vs. iteration number,  $r = 1.0$ ,  $\theta = 0^\circ$ ,  $Re = 1088$

Table II. Iterative solution history in analysis of cavity problem

Method	Re 100			Re 400			Re 1000		
	1	no. of iterations	rel. time	no. of iterations	rel. time	no. of iterations	no. of iterations	rel. time	rel. time
Successive substitution	1	6	4.6	8	2.6	21	20	9.4	
Newton-Raphson	1	3	2.2	5	1.6	divergence	4	2.0	
Modified Newton	1	5	1.1	45	2.3	divergence	divergence		
Broyden:									
shift at 10 iterations	1	4	1	24	1.5	divergence	8	1.1	
shift at 5 iterations				31	1.8		8	1	
reform at 10 iterations				12	1.2				
reform at 5 iterations				9	1		6	1.3	

 $u_0 = \text{Stokes flow}$  $u_0 = \text{1st iter. succ. subst.}$

strategies (a) and (b) indicates that the updating history is of crucial importance to the success of the algorithm. Once again we find a substantial savings in execution time using Broyden's update, even with reformation of the stiffness matrix.

At  $Re = 1000$ , three methods—Newton–Raphson, modified Newton and quasi-Newton—failed to converge. In this case the initial starting solution—the Stokes solution—is far from the actual solution. This suggested the use of the first iterate from successive substitution as a starting vector for the other solution schemes, and the resulting rapid convergence of Newton–Raphson and quasi-Newton (Table II) confirms that the difficulty was in the initial iteration vector. It should be noted that the Reynolds' number at which divergence of the solution algorithm occurs depends also on the finite element discretization used; this relationship was not investigated in the present work.

*Remark 1.* The reformation of the tangent stiffness matrix after 5–10 BFGS steps is a possible modification also in applications of the Matthies–Strang algorithm to structural analysis. In addition, the factor STOL which governs the frequency of line searches might be tested at 0.9 (as in optimization calculations) and the maximum step SMAX reduced to 4. The ultimate goal is to link recommended strategies to specific application areas in both structures and fluids.

*Remark 2.* The numerical solutions for large Reynolds' number are highly oscillatory along part of the boundary, and our numerical experiments—which concentrated on solving the nonlinear finite element equations for a fixed discretization—did not determine the behaviour of these oscillations as the mesh is refined.

## SUMMARY AND CONCLUSIONS

Despite the limited nature of the present study we believe that several conclusions are possible. We stress that all our experiments were computed with a penalty function formulation for the pressure variable; this may have influenced the performance of the algorithms, but we think not dramatically. Our results indicate that there is not yet a globally applicable cost-effective solution algorithm for all types of flows and range of Reynolds' numbers.

Our implementation of Broyden's update can be introduced with relative ease into existing computer programs, and can constitute a very effective solution procedure by itself or in combination with other basic techniques.

The study indicates that the effective solution of a given fluid mechanics problem may require a combination of solution algorithms (for example, a few iterations of successive substitution followed by a Newton-based method). Any general fluid mechanics code should allow the user a choice of solution algorithms and, more importantly, a possibility of combining them to form a solution strategy. An important area of further research is the development of techniques which would automatically determine a solution strategy.

We have demonstrated that, in the examples considered, Broyden's update, which may be thought of as a perturbed or accelerated modified Newton algorithm, has the same general convergence characteristics as the Newton–Raphson method. Each iteration requires only the solution of a single linear system (in which the triangular factors of the coefficient matrix are already known) plus the vector operations which take account of the updates. For low Reynolds' numbers the savings can be substantial, and for strong nonlinearities a periodic reformation of the stiffness matrix increased the overall effectiveness of the algorithm.

Among the aspects still to be investigated are

1. Application to transient analysis with implicit time integration; and
2. Incremental solution of non-Newtonian flow.

The experiments reported here, which are our first experiences with the quasi-Newton method in fluid mechanics, suggest that it may play a constructive part in the finite element analysis of fluid flow.

#### ACKNOWLEDGEMENTS

We thank D. Gartling for providing us with his computer program for calculating the exact solution to the Hamel problem. We are grateful for the financial support of the ADINA users' group, the Army Research Office and the National Science Foundation (MCS-7812363).

#### REFERENCES

1. D. K. Gartling, 'Texas fluid analysis program—user's manual', *Report 75-2*, Texas Institute for Computational Mechanics (1974).
2. C. Taylor and P. Hood, 'A numerical solution of the Navier–Stokes equations using the finite element technique', *Comp. Fluids*, **1**, 73 (1973).
3. D. K. Gartling, R. E. Nickell and R. I. Tanner, 'A finite element convergence study for accelerating flow problems', *Int. J. num. Meth. Engng*, **11**, 1155 (1977).
4. H. Matthies and G. Strang, 'The solution of nonlinear finite element equations', *Int. J. num. Meth. Engng*, **14**, 1613 (1979).
5. K.-J. Bathe and A. Cimento, 'Some practical procedures for the solution of nonlinear finite element equations', *Comp. Meth. Appl. Mech. Eng.*, **22**, 59 (1980).
6. K.-J. Bathe, 'ADINA', *Report 82448-1*, Acoustics and Vibration Lab., Dept. of Mechanical Engineering, MIT, Sept. 1975 (rev. Dec. 1978).
7. M. Bercovier and M. S. Engelman, 'A finite element for the numerical solution of viscous incompressible flows', *J. Comp. Phys.* **30**, 181 (1979).
8. W. C. Davidon, 'Variable metric method for minimization', *Rep. ANL-5990*, Argonne National Laboratories, Argonne, IL (rev. 1959).
9. J. E. Dennis and J. Moré, 'Quasi-Newton methods, motivation and theory', *SIAM Rev.* **19**, 46 (1977).
10. C. G. Broyden, 'A class of methods for solving nonlinear simultaneous equations', *Math. Comp.* **19**, 577 (1965).
11. C. Taylor and A. Z. Ijam, 'A finite element numerical solution of natural convection in enclosed cavities', *Comp. Meth. Appl. Mech. Eng.* **19**, 429 (1979).
12. T. J. Chung, *Finite Element Analysis in Fluid Dynamics*, McGraw-Hill, Inc., 1978.
13. T. J. R. Hughes, R. L. Taylor and J. F. Levy, 'High Reynolds number, steady, incompressible flows by a finite element method', in *Finite Elements in Fluids*, vol. 3, Wiley, 1978.
14. G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 1970.