INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 8, 123–138 (1988)

A FINITE ELEMENT CONVERGENCE STUDY FOR SHEAR-THINNING FLOW PROBLEMS

A. KARAGIANNIS, H. MAVRIDIS, A. N. HRYMAK AND J. VLACHOPOULOS

Department of Chemical Engineering, McMaster University, Hamilton, Ontario, Canada L8S 4L7

SUMMARY

The solution of the non-linear set of equations arising from the application of the finite element method to non-Newtonian fluid flow problems often requires large amounts of computer time. Four iteration schemes (Picard, Newton–Raphson, Broyden and Dominant Eigenvalue method) are compared in three different flow geometries using a shear-thinning fluid model. Points of comparison involve the computer time necessary to converge the equations, ease of implementation, radius of convergence and rate of convergence.

KEY WORDS Finite elements Shear-thinning flow Convergence study Newton–Raphson Picard Dominant eigenvalue Broyden

INTRODUCTION

Finite element simulation of the steady state Navier–Stokes equations for shear-thinning fluids involves the solution of a large set of non-linear equations. A non-linear equation solving algorithm must be robust with respect to the degree of non-linearity and starting point and be capable of handling very large equation sets if it is to be used in a large variety of physical situations. The rank order of algorithm efficiency may change depending on the specific problem and the 'best' algorithm may be different depending upon the stage in the solution procedure in solving a particular problem. A strategy for solving the non-linear equation sets may be necessary where different algorithms are used in sequence for the desirable properties that each may possess. An efficient steady state non-linear equation technique is an important basis for the solution of the time-dependent problem.

Non-linearities arise from a number of different sources. For example, it is well known that high Reynolds number flows cause convergence problems due to the non-linearity from the convective terms. Non-Newtonian problems have the stress tensor as a non-linear function of the rate-of-strain tensor. Free boundary problems are very non-linear when the free surface position co-ordinates are determined with the nodal velocities and pressures.

The difficulty of solving non-linear fluid flow problems can often be related to a parameter intrinsic to the fluid/geometry system. For convective problems the degree of difficulty increases with the Reynolds number of the fluid. Shear-thinning fluids described by a power law model.[1]

$$\boldsymbol{\tau} = -k(\tfrac{1}{2}\dot{\boldsymbol{\gamma}}:\dot{\boldsymbol{\gamma}})^{(n-1)/2}\dot{\boldsymbol{\gamma}} \tag{1}$$

lead to increasingly non-linear equations as n is decreased. The convergence of free boundary problems is sensitive to the curvature of the free surface and to the application of the boundary

0271–2091/88/020123–16$08.00

Received 29 September 1986
Revised 23 March 1987

conditions at the contact lines between fluid and solid. The solution of very non-linear equations requires an incremental approach. For convection-dominated flow, the creeping flow case is used as a starting solution and the Reynolds number is incremented in a series of steps to the desired value. Similarly, in shear-thinning fluids a series of problems with decreasing power law index are solved and for free boundary problems a series of interface shapes are examined until all the boundary conditions are satisfied. In polymer-processing simulations it is common to find combinations of these non-linearities in a single problem; for example, the flow of a shear-thinning polymer melt into a mould cavity.

For most polymer-processing studies the non-Newtonian effects (viscoelasticity, shear thinning) are more significant than the convective effects because of the low Reynolds numbers involved. This work focuses on finite element equation convergence problems for shear-thinning effects and is complementary to previous work.[2-5]

When the Galerkin finite element method is applied to the fluid momentum equations, the resulting non-linear equations are in the following form:

$$\mathbf{K}(\mathbf{u})\mathbf{u} = \mathbf{F}, \tag{2}$$

where $\mathbf{u}$ is a vector containing all the velocities and pressures at the finite element nodes. There are many solution procedures available, but in this work four are compared: fixed point iteration (Picard iteration, PI), Newton–Raphson (NR), a quasi-Newton method by Broyden (BR) and the Dominant Eigenvalue method, Kaniel–Stein version (DE). There has been considerable work done with the first three methods and we propose the DE as an alternative. A comparative study of NR,PI and BR on convective and shear-thinning flow problems was also undertaken by Gartling[4] and Engelman.[5]

We shall use the terminology that 'problem' refers to the solution of the finite element equations at the parameter of interest (e.g., $n = 0{\cdot}25$ for a shear-thinning fluid), 'subproblem' refers to the solution of the finite element equations at some intermediate value of the parameter (e.g., solve the equations for $n = 0{\cdot}5$ to obtain a starting guess for $n = 0{\cdot}25$) and 'iterations' refer to the number of applications of a particular convergence scheme to obtain the solution for a subproblem.

Selection of a robust non-linear equation solving technique involves two important issues. The radius of convergence is the difference between the initial trial vector for the unknowns and the true solution vector such that the equation set can be solved starting from the initial trial vector. For a given problem parameter value, it is desirable to have the radius of convergence to be as large as possible. The rate of convergence of the iterates should be maximized to minimize the number of iterations. Each of the iteration methods mentioned previously has its own strengths and weaknesses with regard to these two issues, which can conflict with each other as in a method that has a high rate of convergence but small radius of convergence.

PROBLEMS

The test problems are creeping flows ($Re = 0$) in the following standard geometries: square duct,[6-8] driven cavity[9] and a 4:1 planar contraction.

The first problem considers the fully developed flow of an inelastic power law fluid in a square duct (Figure 1(a)). The flow is assumed to be unidirectional and is governed by the Poisson equation with the no-slip boundary conditions

$$\frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right)=\frac{\mathrm{d}P}{\mathrm{d}z}, \tag{3}$$

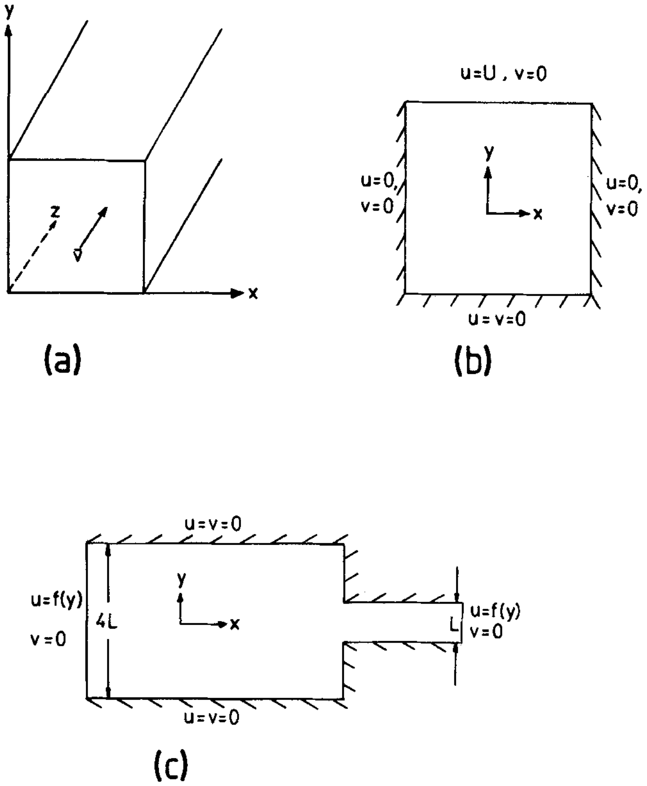

Figure 1. Problem statement: (*a*) square duct; (*b*) driven cavity; (*c*) 4:1 contraction

where

$$\mu = k\left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right]^{(n-1)/2} \tag{4}$$

The second and third problems consider the flow of an inelastic power law fluid inside a square cavity where one wall is moving with constant velocity (Figure 1(b)) and in a 4:1 contraction (Figure 1(c)), respectively. The governing equations are

$$\frac{\partial}{\partial x}\left(2\mu\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] = \frac{\partial P}{\partial x}, \tag{5a}$$

$$\frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left(2\mu\frac{\partial v}{\partial y}\right) = \frac{\partial P}{\partial y}, \tag{5b}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{5c}$$

where

$$u = k\left[2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)^2\right]^{(n-1)/2} \tag{6}$$

The equations were discretized using a standard Galerkin finite element *u*–*v*–*p* formulation with triangular-shaped elements which were $C^0 - P^2$ for the nodal velocities and $C^0 - P^1$ linear for the pressures.[10]

The square duct problem mesh is shown in Figure 2 (200 elements, 441 nodes or 361 variables). The driven cavity mesh for 200 elements is the same as that used for the square duct problem (861 variables) and the 72 element case is shown in Figure 3 (169 nodes or 300 variables). The pressure contours are presented in Figure 4 for the driven cavity problem with a power law index of 0·5 on the 200 element mesh, with the appropriate velocity vector solution in Figure 5. The 4:1 contraction problem mesh is shown in Figure 6 and contains 220 elements (501 nodes or 966 variables). The pressure contours and velocity vectors are shown in Figures 7 and 8 for a power law index of 0·5.

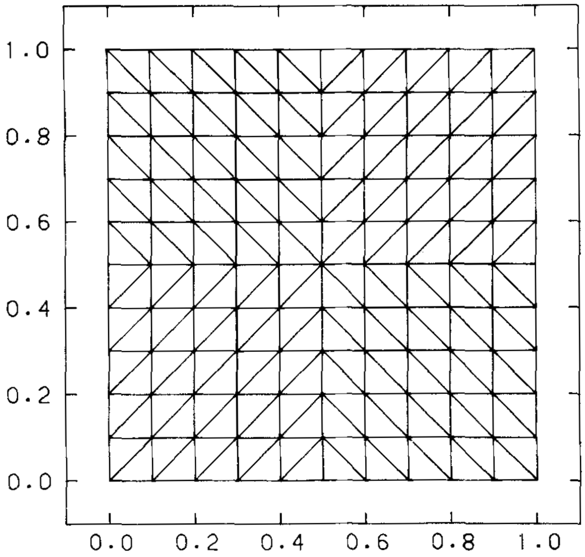

Figure 2. Finite element grid, square duct and driven cavity problem, 200 elements

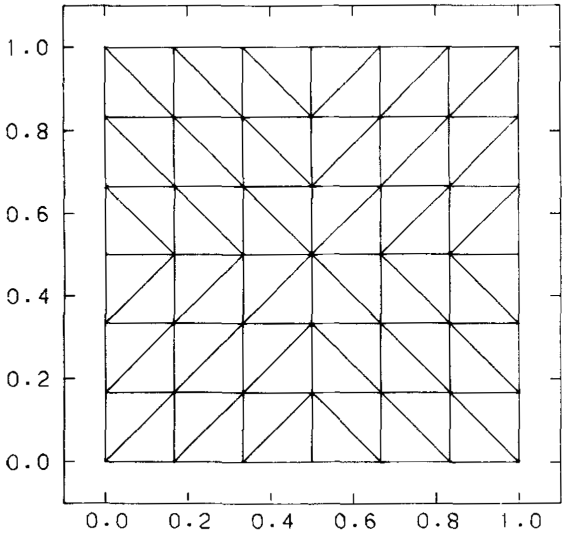

Figure 3. Finite element grid, driven cavity problem, 72 elements

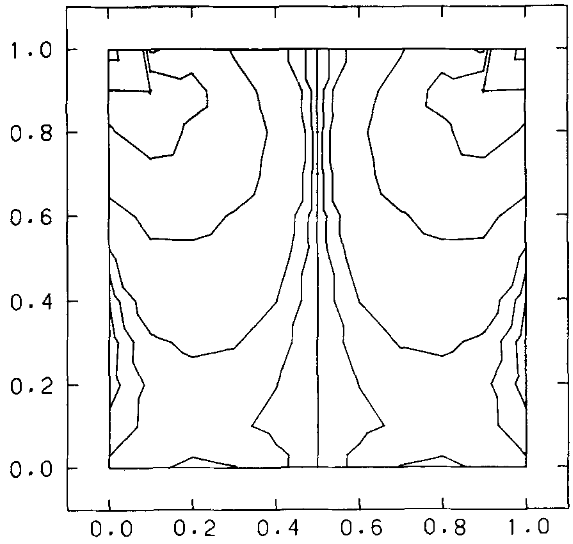

Figure 4. Pressure contours, driven cavity problem, 200 element grid; power law index 0·5

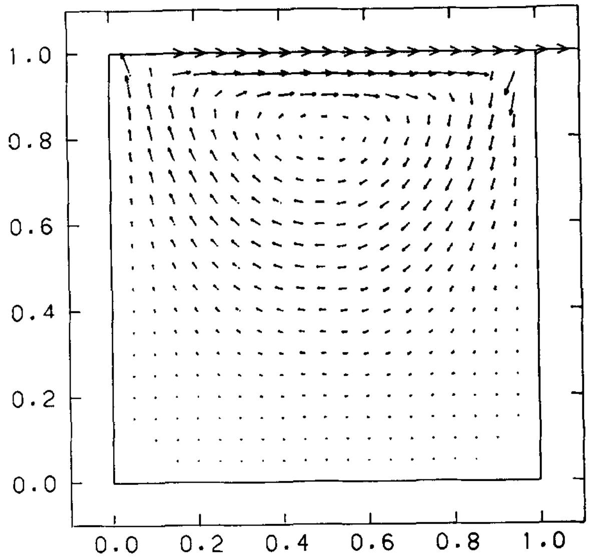

Figure 5. Velocity vectors, driven cavity problem, 200 element grid; power law index 0·5

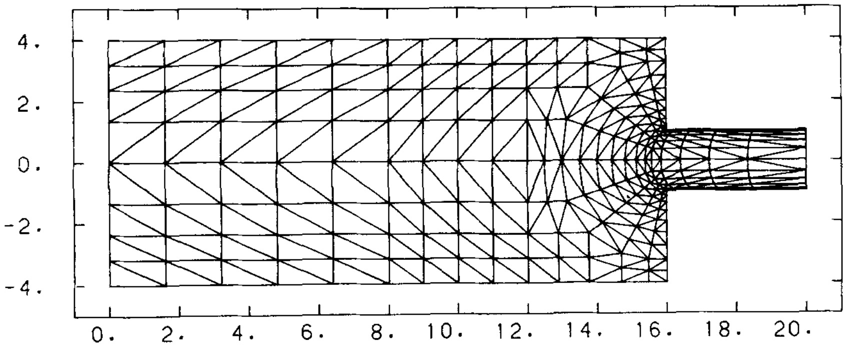

Figure 6. Finite element grid, 4:1 contraction problem

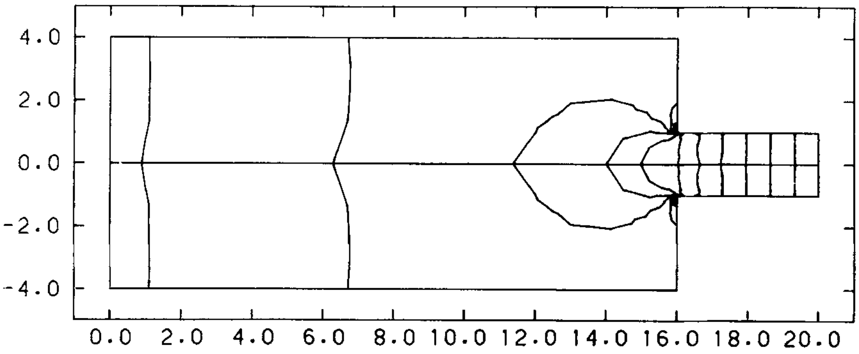

Figure 7. Pressure contours, 4:1 contraction problem; power law index 0·5

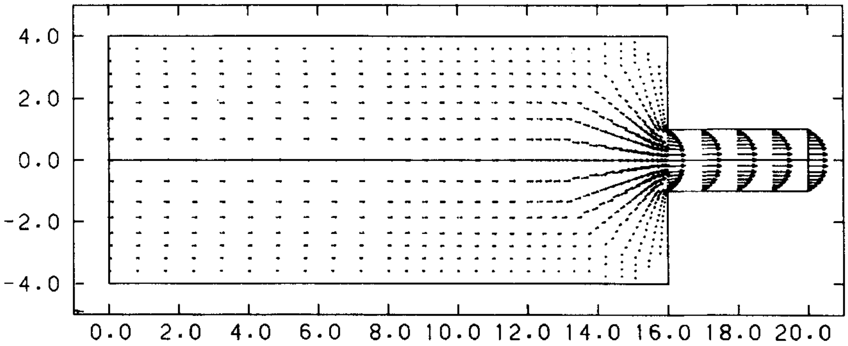

Figure 8. Velocity vectors 4:1 contraction problem; power law index 0·5

ALGORITHMS

Picard iteration

Picard or fixed point iteration is the simplest scheme to implement and is written as

$$\mathbf{K}(\mathbf{u}^i)\mathbf{u}^{i+1} = \mathbf{F}. \tag{7}$$

The non-linear coefficient matrix terms are evaluated using the values of **u** from the previous iteration and the entire equation set is solved again to determine the new vector quantities. The method's convergence rate is asymptotically linear (slow), but it converges for a relatively large parameter range and is not very sensitive to the initial vector $\mathbf{u}^0$.

Sometimes the basic scheme is modified by the addition of a weighting factor for the previous iteration solution:

$$\mathbf{K}(\mathbf{u}^i)\mathbf{u}^* = \mathbf{F}, \tag{8}$$

$$\mathbf{u}^{i+1} = \alpha\mathbf{u}^i + (1-\alpha)\mathbf{u}^*. \tag{9}$$

The factor α varies between 0 and 1, with $\alpha = 0$ being the standard direct substitution case.

Newton–Raphson

We solve the following equation set:

$$\mathbf{R}(\mathbf{u}) = \mathbf{K}(\mathbf{u})\mathbf{u} - \mathbf{F} = \mathbf{0}. \tag{10}$$

Using a first-order Taylor series expansion around a nominal point and assuming a relatively small change in **u** from iteration i to iteration $i+1$,

$$\mathbf{J}(\mathbf{u}^i)\Delta\mathbf{u} = -\mathbf{R}(\mathbf{u}^i), \tag{11}$$

where

$$\mathbf{J}(\mathbf{u}^i) = \frac{\partial \mathbf{R}}{\partial \mathbf{u}}\bigg|\mathbf{u}^i, \tag{12}$$

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \beta\,\Delta\mathbf{u} \tag{13}$$

and β can be used to modify the step length with a line search, to minimize the sum of the residuals along a given search direction, if necessary.

The Newton–Raphson method has a quadratic asymptotic rate of convergence. However, the radius of convergence is quite small; i.e., one must start near the solution for very non-linear problems. Generally, the initial vector is the solution of a linear version of the problem and a parameter is incremented in a series of subproblems to approach the solution of interest. This initialization strategy is sometimes called incremental loading or zero-order continuation. The superior convergence properties of the method insure that for 'good' choices of the parameter increments the subproblems will be solved cost effectively. It is not possible at the present time to predict the values of the parameters for the least number of subproblem solutions. It is possible that many subproblems must be solved, leading to longer total computation times than expected.

An automatic extension of a parameter incrementation strategy is to use a first-order continuation method (or homotopy method) which uses the Jacobian from the solution to a given value of the parameter to predict the next parameter value to be used.[11] We have not pursued first-order continuation method/Newton–Raphson combination approaches in this work.

Another alternative is the modification of the step length with a line search. The procedure increases the radius of convergence considerably[5] but may still lead to long computation times if many line searches are performed.

The Jacobian for Newton–Raphson iterations was calculated using analytical derivatives of the finite element equations for all the examples in this work, and the variation in effective viscosity with velocity was included. Finite difference approximations to the derivatives could be used but would add another complication in comparing different non-linear equation solving schemes. The accuracy of the finite derivative approximation directly affects the rate of convergence of the Newton–Raphson method and a whole study would be needed for the effect of the perturbation selected in the numerical differencing.

Quasi-Newton methods

The major objection to the Newton–Raphson method is the necessity of recalculating Jacobians and decomposing the linearized set of equations at every iteration. A quasi-Newton procedure (variable metric method) generalizes the Newton–Raphson method[3] to

$$\mathbf{u}^{i+1} = \mathbf{u}^i - s^i\mathbf{H}^i\mathbf{R}^i. \tag{14}$$

$\mathbf{R}^i$ is the residual evaluated at $\mathbf{u}^i$, $\mathbf{H}^i$ is an approximation to the inverse of the Jacobian matrix and s^i is analogous to β of the Newton–Raphson method as a line search variable to force descent in the residual sum. The Jacobian inverse approximation is updated at each iteration:

$$\mathbf{H}^{i+1} = \mathbf{H}^i + \Delta\mathbf{H}^i. \tag{15}$$

The inverse form of the rank-out update given by Boryden's[12–14] method is

$$\mathbf{H}^{i+1} = \mathbf{H}^i + (\Delta\mathbf{u}^i - \mathbf{H}^i\Delta\mathbf{R}^i)(\Delta\mathbf{u}^{iT}\mathbf{H}^i)/(\Delta\mathbf{u}^{iT}\mathbf{H}^i\Delta\mathbf{R}^i). \tag{16}$$

Note that the Jacobian inverse approximation is a full matrix. The method should be implemented to preserve sparsity and not use the full Jacobian matrix approximation as presented in the above equations. Update vectors are stored and used to modify an initial approximation to the Jacobian inverse matrix.

The specific Broyden algorithm implemented is that of Engleman *et al.*[3] and Engleman[5] which calculates the updated $\mathbf{H}^i$ from the original Jacobian inverse $\mathbf{H}^0$ at each iterative cycle. A Broyden algorithm to solve $\mathbf{R}(\mathbf{u}) = \mathbf{K}(\mathbf{u})\mathbf{u} - \mathbf{F} = \mathbf{0}$ is as follows:

1. Given $\mathbf{u}^0$, $\mathbf{J}^0$ (Jacobian matrix after one NR or an approximation to the Jacobian after one PI iteration)
   set $i = 0$
   compute
   $$\mathbf{R}^0 = \mathbf{R}(\mathbf{u}^0), \quad \mathbf{J}^0\mathbf{d}^0 = \mathbf{R}^0, \quad \mathbf{d}^0 \text{ is the initial search direction}$$
   Estimate s^0 with a line search if necessary
   update
   $$\mathbf{u}^1 = \mathbf{u}^0 - s^0\mathbf{d}^0.$$
2. Set i to $i + 1$
   compute
   $$\mathbf{R}^i = \mathbf{R}(\mathbf{u}^i), \quad \mathbf{J}^0\mathbf{q}^i = \mathbf{R}^i$$
   if $i = 1$ go to 4.
3. For $j = 1, \ldots, i - 1$
   compute
   $$\mathbf{q}^{j+1} = \mathbf{q}^j + p^j(\boldsymbol{\delta}^j - \mathbf{r}^j)\boldsymbol{\delta}^j\mathbf{q}^j.$$
4. Compute and store
   $$\mathbf{r}^i = \mathbf{q}^i - \mathbf{d}^{i-1}, \quad \boldsymbol{\delta}^i = \mathbf{u}^i - \mathbf{u}^{i-1} = -s^{i-1}\mathbf{d}^{i-1}, \quad p^i = 1/\boldsymbol{\delta}^{iT}\mathbf{r}^i$$
   compute
   $$\mathbf{d}^i = \mathbf{q}^i + p^i(\boldsymbol{\delta}^i - \mathbf{r}^i)\boldsymbol{\delta}^{iT}\mathbf{q}^i, \text{ the new search direction,}$$
   estimate s^i with a line search if necessary
   update
   $$\mathbf{u}^{i+1} = \mathbf{u}^i - s^i\mathbf{d}^i.$$
5. If norm of residuals is less than tolerance and norm of the error is less than tolerance, then stop
   else if $i < i_{\max}$ go to 2
   else stop

In the implementation of the method, the LU factors of the linear equation decomposition that were found in step 1 are used as the initial Jacobian inverse. Then, if Broyden's method is interchanged with Newton–Raphson, the LU factors provide the inverse of an exact Jacobian, whereas if Broyden's method is interchanged with Picard, the LU factors provide the inverse of an approximate 'Jacobian'. Step 2 uses the LU factors and so involves only a back-substitution which is relatively inexpensive. An alternative would be to use an initial Jacobian found by using $\mathbf{u}^0$ and the analytical derivatives of $\mathbf{R}^0$.

While the algorithm uses all the updating vectors previously calculated, in general only a fixed number of updating vectors may be used. At the limit of stored update vectors one would restart the algorithm with the Jacobian inverse available at that time. Another alternative is to remove the oldest update vector and continue with the same fixed number of update vectors using the original estimate for the matrix. We investigate both alternatives. The former requires more computer time for matrix factorization while the latter option has slower convergence. The specific line search algorithm implemented was that of Matthies and Strang.[15] Line searches are not performed unless[15]

$$|G| > \mathrm{stol}^* |G0|,$$

where

$$G = \mathbf{d}^{\mathrm{T}}\mathbf{R}(\mathbf{u}-\mathbf{d}), \quad G0 = \mathbf{d}^{\mathrm{T}}\mathbf{R}(\mathbf{u})$$

and

$$\mathrm{stol} = 0{\cdot}5, \text{ the tolerance}$$

Dominant eigenvalue method

The dominant eigenvalue method[16–18] takes into account interaction between variables in determining a weighted combination of previous iteration solutions to predict the solution of the iteration sequence. The basis of the method is to assume that the iterations approximately follow a linear matrix difference equation and compute the apparent solution by using estimates of the dominant eigenvalues.

The basic idea is to do a number of Picard iterations on the equation set, building up an iteration history, and then to accelerate the iteration process by predicting the apparent solution from the previous iterations. The predicted solution, after $\mathbf{u}^{i+1}$ is calculated, is given by

$$\mathbf{u}^{i\mathrm{P}} = \sum_j \mu_j \mathbf{u}^{i-j+1} \Big/ \sum_j \mu_j, \tag{17}$$

where $j = 0, \ldots, v$ and v is the number of dominant eigenvalues. The coefficients μ_j are determined by minimizing the following expression with respect to the coefficients:

$$\min \left\| \sum_j \mu_j \Delta\mathbf{u}^{i-j} \right\|, \tag{18}$$

where $1 \leqslant v \leqslant i \leqslant m$ and m is the number of equations. Solving the overdetermined set of equations,

$$\sum_j \mu_j b_{jk} = 0, \quad k = 1, 2, \ldots, v \tag{19}$$

with (Kaniel–Stein[18])

$$\sum_j \mu_j = 1 \tag{20a}$$

or (Crowe and Nishio[17])

$$\mu_0 = 1, \tag{20b}$$

$$b_{jk} = \langle \Delta\mathbf{u}^{i-j}, \Delta\mathbf{u}^{i-k} \rangle, \tag{21}$$

provides the coefficients.

The number of dominant eigenvalues is problem-dependent, but we found that for the problems presented in this work only one was needed; i.e., we weighted the $i+1$ and i iterations. More coefficients may be used, but we found only marginal improvement in computation time. The criterion for taking a promotion step is to promote when estimates of the apparent solution on iteration i, $\mathbf{u}^{i\mathrm{P}}$, do not change more than a prescribed amount in the norm compared with the next prediction, $\mathbf{u}^{i+1\mathrm{P}}$. Promoting too frequently leads to numerical instability. The method is stabilized by taking a promotion step no more than every v iterations. For the simple one dominant eigenvalue case, according to Kaniel and Stein[18]

$$\mathbf{u}^{\mathrm{P}} = \mathbf{u}^{i} + \mu_0(\mathbf{u}^{i+1} - \mathbf{u}^{i}), \tag{22}$$

$$\mu_0 = -b_{11}/(b_{01} - b_{11}), \tag{23a}$$

$$\mu_1 = b_{01}/(b_{01} - b_{11}) \tag{23b}$$

or according to Crowe and Nishio[17]

$$\mathbf{u}^{\mathrm{P}} = \mathbf{u}^{i} + (\mathbf{u}^{i+1} - \mathbf{u}^{i})/(1 + \mu_1), \tag{24}$$

$$\mu_1 = -b_{01}/b_{11}. \tag{25}$$

Discussion

The deficiencies of each of the methods can probably be overcome by combining methods for different purposes. For example, Picard iteration has a large radius of convergence while Newton–Raphson has a fast rate of convergence, so a strategy could be to start with Picard and switch to Newton–Raphson.

Iterative solutions to non-linear equations require a criterion to end the iteration cycles. The criterion must not only signal an acceptable solution that is within specified tolerances, but also warn of divergence. There is always a trade-off between accuracy and computer time, since high accuracy demands many more iterations. Relative error criteria are preferred to absolute error criteria because of differences in scale in the nodal variables.

Two standard termination criteria are that the relative change in the solution vector be less than a preset error,

$$\|\Delta\mathbf{u}^{i}\| \leqslant \varepsilon\,\|\mathbf{u}^{i}\|, \tag{26}$$

and that the value of residual sum be less than a preset error,

$$\|\mathbf{R}(\mathbf{u}^{i})\| \leqslant \varepsilon. \tag{27}$$

For all matrix factorizations we used an unsymmetric band solver using LU decomposition from the LINPACK library (DGBFA and DGBSL). The finite element mesh was generated and automatically renumbered for minimum band width using standard mesh techniques. A frontal method or a skyline method[19] for solving the finite element equations may be used, but in this work a generally available equation solver was chosen.[20]

RESULTS

For all runs performed, the Newtonian problem was solved first to obtain an initial solution and then a Picard or Newton–Raphson iteration was performed with the power law index set to the first subproblem. Then either Picard, Newton–Raphson, Broyden or DE was performed with or without line search. The single iteration needed to obtain the Newtonian (linear) solution is not included in the iteration count for the non-linear problem. The convergence criterion used was that the norm of the residuals be less than 10^{-6}. All runs were performed on a VAX8600 computer in double precision.

In cases using the Broyden method, one Picard or Newton–Raphson iteration is performed after every four Broyden iterations except where noted. This strategy proved to stabilize the procedure, leading to convergence even for very non-linear cases (power law index $n = 0{\cdot}2$).

Table I. Square duct

	Method	Total iteration	No. of line searches	CPU time(s)
$n = 0{\cdot}5$				
	NR	6	2	69·1
	PI	16	—	123·0
	DE-1	10	—	82·5
	BR-P	15	11	49·1
	BR-N	8	11	43·9
$n = 0{\cdot}3$				
	NR	13	26	147·2
	PI	31	—	186·5
	DE-I	19	—	148·4
	BR-P	20	29	72·2
	BR-N	31	85	165·0

Table I shows that, for the square duct problem, all the methods converged for $n = 0{\cdot}5$ and Broyden's method was fastest in total computer time. The solution is very regular. DE-1 signifies that only one dominant eigenvalue was assumed. In the case of $n = 0{\cdot}3$ the NR method would not converge from the Newtonian case, while it converges with the use of line searches. The cost of NR with line searches is equal to that of the DE method. Note that in the case where the Broyden algorithm uses an approximate Jacobian (from a Picard iteration), the solution proceeds without difficulty, whereas in the case where an exact Jacobian (from a Newton–Raphson iteration) is used, convergence was achieved with some difficulty. The method needed an excessive number of line searches, resulting in a high cost. The superior properties of the Picard iteration with respect to the radius of convergence seem to stabilize Broyden's algorithm.

Table II shows the results for the driven cavity problem with 72 elements. For $n = 0{\cdot}5$ Broyden's method is very fast relative to the other methods. At $n = 0{\cdot}3$ the Picard is quite slow and Newton–Raphson cannot converge from the Newtonian solution unless line searches or continuation are performed. What is significant is that the DE works for every case at a cost approximately half that of the Picard and comparable with that of Newton–Raphson in the very non-linear case of $n = 0{\cdot}2$. Broyden's method with an approximate Jacobian is able to converge with the lowest cost, even when no line searches are performed. However, NR required the implementation of line searches or continuation and Broyden's method required some strategy to obtain a solution (which was not obvious), while the DE required no extra steps at all. When the 200 element case is compared in Table III it is seen that for $n = 0{\cdot}5$ and $n = 0{\cdot}3$ DE requires half the cost of Picard, while Broyden's method is again the fastest, but when the exact Jacobian is used in the case of $n = 0{\cdot}3$ the algorithm will not converge. In the case of $n = 0{\cdot}5$ Broyden's method with approximate Jacobian did not require any line searches, while in the 72 element case it required four line searches. The significant result is that for the case of $n = 0{\cdot}2$ the DE method is faster than the NR, which required line searches to converge.

Table IV shows the results for the 4:1 contraction problem, which was the most difficult problem of the group, mainly because of the variety of flow regimes in the solution. For $n = 0{\cdot}5$ Newton–Raphson with line searches and Broyden with exact Jacobian are surprisingly time-consuming, while DE and Broyden with approximate Jacobian have no difficulties. However, in the $n = 0{\cdot}3$ case Broyden with line searches and exact Jacobian would not converge, while the case with approximate Jacobian had convergence difficulties, leading to high computation times. Note that the Broyden with approximate Jacobian/$n = 0{\cdot}5$ case did not have any difficulty in the square duct

Table II. Driven cavity, 72 elements

	Method	Total iterations	No. of line searches	CPU time(s)
$n = 0{\cdot}5$				
	NR	7	2	105
	PI	21	—	184
	DE-1	13	—	119
	DE-3	14	—	130
	BR-P	14	4	55
	BR-N	10	9	89
$n = 0{\cdot}3$				
	NR	9	8	134
	NR*	18	—	244
	PI	49	—	417
	DE-1	24	—	210
	BR-P	23	—	73
	BR-P	25	24	99
	BR-N	16	28	152
$n = 0{\cdot}2$				
	NR	20	39	296
	DE-1	35	—	305
	BR-P	49	—	137

*Newton–Raphson method without line search required a series of subproblems of $n = 1$, $n = 0{\cdot}5$, $n = 0{\cdot}35$, $n = 0{\cdot}3$ for convergence and the time is total of all CPU times. One Picard iteration was performed before switching to Newton–Raphson in all subproblems.

Table III. Driven cavity, 200 elements

	Method	Total iterations	No. of line searches	CPU time(s)
$n = 0{\cdot}5$				
	NR	8	1	682
	PI	18	—	1221
	DE-1	11	—	775
	DE-3	13	—	958
	BR-P	14	0	313
	BR-N	11	8	452
$n = 0{\cdot}3$				
	NR	13	15	1087
	PI	34	—	2254
	DE-1	20	—	1367
	BR-P	30	—	558
	BR-P	23	28	549
	BR-N	Divergence		
$n = 0{\cdot}2$				
	NR	26	49	2165
	DE-1	29	—	1986

Table IV. 4:1 contraction

	Method	Total iterations	No. of line searches	CPU time(s)
$n = 0{\cdot}5$				
	NR	10	10	411
	PI	21	—	457
	DE-1	12	—	276
	DE-3	14	—	312
	BR-P	17	3	183
	BR-N	16	24	431
$n = 0{\cdot}3$				
	NR	13	16	516
	PI	38	—	798
	DE-1	18	—	398
	BR-P	28	—	230
	BR-P	33	52	414
	BR-N	Divergence		

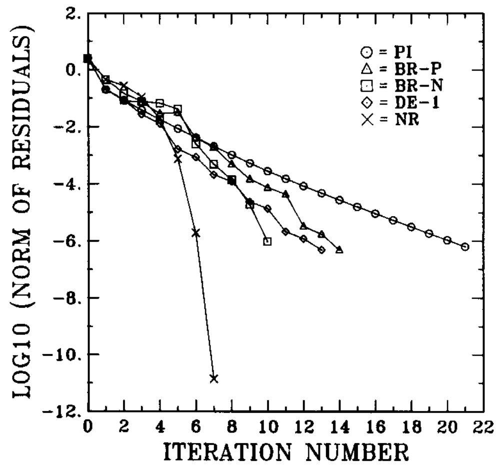

Figure 9. Logarithm of the norm of residuals versus iteration number: driven cavity problem, 72 elements; power law index 0·5

or the driven cavity, and this is an example of how methods can fail depending on the specific fluid/geometry problem. DE converged without any special steps for both $n = 0{\cdot}5$ and $n = 0{\cdot}3$ cases from the Newtonian solution, and what is more significant is that in the case of $n = 0{\cdot}3$ DE was the fastest, surpassed only by Broyden without line searches.

The results in the cases where line searches were performed are in general agreement with those of Engelman[5] for the Picard, Broyden and Newton–Raphson methods.

Figure 9 shows a plot of the norm of the residuals versus the iteration number for the 72 element driven cavity, and Figure 10 for the 4:1 contraction. In both plots, NR converges very rapidly, Picard very slowly and the other methods are somewhere in between. The tables also point out the fact that even though NR has few iterations, the iterations are expensive because they entail a Jacobian evaluation and a matrix factorization. The cost of the algorithms increases when excessive line searches are performed. In Tables I–IV the number of line searches shown is the number of function evaluations within the line search subroutine.

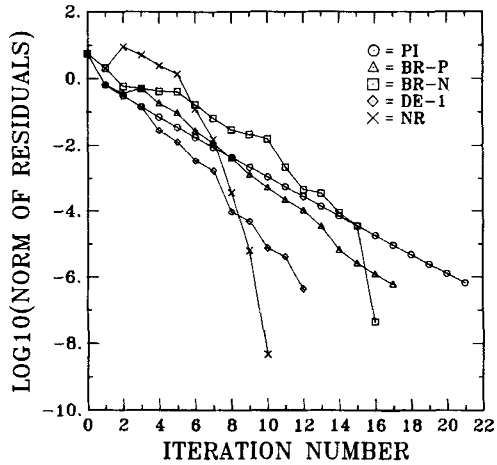

Figure 10. Logarithm of the norm of residuals versus iteration number: 4:1 contraction problem; power law index 0·5

CONCLUSIONS

Broyden's method is the least expensive solution method computationally, but does not converge for strongly non-linear problems. One solution is to reform the Jacobian matrix after a prescribed number of Broyden iterations. The use of an approximate initial Jacobian (Picard) seems to increase the radius of convergence of the method more than the use of an exact initial Jacobian (Newton–Raphson), which seems to have a better rate of convergence. Line searches did not always help the convergence of the algorithm. The Newton–Raphson method for strongly non-linear problems requires line searches or continuation to converge. The cost in some cases becomes excessive. The dominant eigenvalue method appears to be robust and converged in all cases examined without any additional steps required. In very non-linear cases, the dominant eigenvalue proved faster than the Newton–Raphson method. Dominant eigenvalue is very simple to implement and provides an alternative for users who want to keep a Picard iteration as the basic iteration scheme.

ACKNOWLEDGEMENTS

Financial assistance from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged. H. Mavridis was also the recipient of a Shell Canada Graduate Research Fellowship.

NOTATIONS

b_{jk}	inner product used in DE method, equation (21)
BR	Broyden's method
BR-N	Broyden's method interchanged with Newton–Raphson
BR-P	Broyden's method interchanged with Picard
$\mathbf{d}$	search direction
DE-1	One dominant eigenvalue
DE-3	Three dominant eigenvalues
$\mathbf{f}$	vector of functions

$\mathbf{F}$	load vector
$\mathbf{H}$	approximation to the inverse of the Jacobian matrix
$\mathbf{J}$	Jacobian matrix
$\mathbf{K}$	'stiffness' matrix
k	material constant of power law model
n	power law index
NR	Newton–Raphson
PI	Picard
p	pressure
p_i, p_j	used in Broyden's algorithm
$\mathbf{q}$	vector used in Broyden's algorithm
$\mathbf{R}$	vector of residuals
$\mathbf{r}$	vector used in Broyden's algorithm
s	line search variable, equation (14)
u	velocity in x-direction
$\mathbf{u}$	vector of unknowns
$\mathbf{u}^{iP}$	predicted solution in DE method, equation (17)
v	velocity in y direction
$\mathbf{v}$	velocity vector

Greek letters

α	weighting factor, equation (8)
β	damping factor, equation (12)
$\dot{\boldsymbol{\gamma}}$	rate-of-strain tensor
$\boldsymbol{\delta}$	vector used in Broyden's algorithm
ε	tolerance
μ	dominant eigenvalue, viscosity
ν	number of dominant eigenvalues
$\boldsymbol{\tau}$	extra stress tensor

Subscripts, superscripts

i, j, k	dummy indices

Symbols

$\langle \cdot, \cdot \rangle$	inner product
$\lVert \cdot \rVert$	norm

REFERENCES

1. Z. Tadmor and C. G. Gogos, *Principles of Polymer Processing*, Wiley, 1979.
2. D. K. Gartling, R. E. Nickell and R. I. Tanner, 'A finite element convergence study for accelerating flow problems', *Int. j. numer. methods eng.*, **11**, 1155–1174 (1977).
3. M. S. Engelman, G. Strang and K. J. Bathe, 'The application of quasi-Newton methods in fluid mechanics', *Int. j. numer. methods eng.*, **17**, 707–718 (1981).
4. D. K. Gartling, Unpublished report, private communication to authors, 1985.
5. M. S. Engleman, 'Quasi-Newton methods in fluid dynamics', in J. R. Whiteman (ed.), *Mathematics of Finite Elements*

and Applications, IV, Academic Press, London, 1982, pp. 479–487; and *FIDAP Theoretical Manual, Revision 3.0*, Fluid Dynamics International, Inc., Evanston, IL, 1986.

6. Ta-Jo Liu, 'Fully developed flow of power-law fluids in ducts', *Ind. Eng. Chem. Fund.*, **22,** 183–186 (1983).
7. Ta-Jo Liu, 'An efficient matrix solver for finite-element analysis of non-Newtonian fluid flow problems', *Int. j. numer. methods fluids*, **5,** 929–938 (1985).
8. J. A. Wheeler and E. H. Wissler, 'The friction factor–Reynolds number relation for the steady flow at pseudoplastic fluids through rectangular ducts', *AIChE J.*, **11,** 207–212 (1965).
9. M. D. Olson and S. Y. Tuann, 'New finite element results for the square cavity', *Comp. Fluids*, **7,** 123–135 (1979).
10. K. H. Huebner and E. A. Thorton, *The Finite Element Method for Engineers, 2nd edn.*, Wiley, 1982.
11. H. S. Kheshgi, O. A. Basaran, R. E. Benner, S. F. Kistler and L. E. Scriven, 'Continuation in a parameter: Experience with viscous and free surface flows, in C. Taylor, J. A. Johnson and W. R. Smith (eds), *Numerical Methods in Laminar and Turbulent Flows*, Pineridge Press, Swansea, 1983.
12. C. G. Broyden, 'A class of methods for solving nonlinear simultaneous equations', *Math. Comput.*, **19,** 577–593 (1965).
13. J. Dennis and J. More, 'Quasi-Newton method, motivation and theory', *SIAM Rev.*, **19,** 46–89 (1977).
14. R. E. White, *An Introduction to the Finite Element Method with Applications to Nonlinear Problems*, Wiley, New York, 1985.
15. H. Matthies and G. Strang, 'The solution of nonlinear finite element equations', *Int. j. numer. methods eng.*, **14,** 1613–1616 (1979).
16. O. Orbach and C. M. Crowe, 'Convergence promotion in the simulation of chemical processes with recycle—The dominant eigenvalue method', *Can. J. Chem. Eng.*, **49,** 509–513 (1971).
17. C. M. Crowe and M. Nishio, 'Convergence promotion in the simulation of chemical processes—The general dominant eigenvalue method, *AIChE J.*, **21,** 528–533 (1975).
18. S. Kaniel and J. Stein, 'Least-square acceleration of iterative methods for linear equations', *J. Opt. Theory Appl.*, **14,** 431–437 (1974).
19. I. Hasbani and M. Engelman, 'Out of core solution of linear equations with non-symmetric coefficient matrix', *Comput. Fluids*, **7,** 13–31 (1979).
20. J. J. Dongarra, J. R. Bunch, C. B. Moler and G. W. Stewart, *Linpack Users' Guide*, SIAM, Philadelphia, 1979.