INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 18, 43–69 (1994)

p-VERSION LEAST SQUARES FINITE ELEMENT FORMULATION FOR TWO-DIMENSIONAL, INCOMPRESSIBLE FLUID FLOW

DANIEL WINTERSCHEIDT* AND KARAN S. SURANA†

The University of Kansas, Department of Mechanical Engineering, 3013 Learned Hall, Lawrence, KS 66045, U.S.A.

SUMMARY

A p-version least squares finite element formulation for non-linear problems is applied to the problem of steady, two-dimensional, incompressible fluid flow. The Navier–Stokes equations are cast as a set of first-order equations involving viscous stresses as auxiliary variables. Both the primary and auxiliary variables are interpolated using equal-order C^0 continuity, p-version hierarchical approximation functions. The least squares functional (or error functional) is constructed using the system of coupled first-order non-linear partial differential equations without linearization, approximations or assumptions. The minimization of this least squares error functional results in finding a solution vector $\{\delta\}$ for which the partial derivative of the error functional (integrated sum of squares of the errors resulting from individual equations for the entire discretization) with respect to the nodal degrees of freedom $\{\delta\}$ becomes zero. This is accomplished by using Newton's method with a line search. Numerical examples are presented to demonstrate the convergence characteristics and accuracy of the method.

KEY WORDS Least squares Finite element p-version Error functional Degrees of freedom p-convergence Newton's method Line search Navier–Stokes Hierarchical Driven cavity Asymmetric expansion

INTRODUCTION

The finite element method has enjoyed great success in solid mechanics and heat conduction applications but has not yet achieved the same level of success in the field of fluid dynamics. Successful application of the method to fluid dynamics problems requires the selection of an appropriate formulation method. Most finite element formulations are based on either variational or weighted residual methods. Variational methods are generally viewed as producing the 'best' approximation to the exact solution of a variational problem. Unfortunately, variational principles cannot be constructed for the Navier–Stokes equations. A possible alternative is to use a weighted residual approach. Galerkin-based methods (Galerkin and Petrov–Galerkin), collocation and the least squares method are all special cases of general weighted residual methods.

Galerkin formulations of the steady state, incompressible Navier–Stokes equations in primitive variable (p, u, v) form lead to several well-known difficulties.

(i) The coefficient matrices are not only non-symmetric but are also ill-conditioned owing to the absence of the pressure variable in the continuity equation.

* Present address: ICOMP, NASA Lewis, Cleveland, OH, U.S.A.

† Author to whom correspondence should be addressed.

0271–2091/94/010043–27$18.50

Received 1 June 1993
Revised 29 July 1993

(ii) The order of approximation for velocities and pressure must satisfy the Ladyzhenskaya–Babuska–Brezzi (LBB) 'inf sup' condition. This condition precludes the use of (convenient) equal-order interpolation for all field variables.
(iii) A converged solution of the non-linear system of equations is difficult to achieve and in some cases solution procedures fail to converge.
(iv) The converged solutions many times exhibit non-physical oscillations similar to those obtained using (central) finite difference schemes.

Several methods of circumventing these problems have been investigated and reported in the literature.[1–5] In general these 'cures' are problem-dependent and require an arbitrary choice of parameters. We believe that these approaches treat only the symptoms and fail to properly diagnose the basic problem. The p-version least squares method offers a more general formulation procedure compared with Galerkin-based methods.

The least squares finite element method (LSFEM) has been advocated as a unified method for fluid dynamics and convective heat transfer problems.[6] Becker[7] observed some time ago that the least squares method satisfies the criteria desirable in a variational method. Becker's criteria include the following.

(i) The procedure should minimize errors in some sense.
(ii) The integrand of the functional should be definite (admit values of only one sign).
(iii) The procedure should be capable of treating initial value problems.

The least squares finite element method has been applied to elliptic,[8] hyperbolic[9] and mixed[10] partial differential equations. Applications include boundary layer flow,[11] gas dynamics,[12] Stokes flow,[13] inviscid compressible flow,[14] convection–diffusion[15] and phase change problems.[16]

The p-version of the finite element method is known to possess superior convergence characteristics compared with the h-version.[17] Nevertheless, most of the current finite element research in computational fluid dynamics (CFD) has involved the use of low-order element approximations. The thrust of the current work is to combine the benefits of p-version approximations with the least squares formulation based on the minimization of the exact least squares error functional derived using actual non-linear partial differential equations without linearization. Jiang and Sonnad[18] have applied a p-version least squares finite element formulation (LSFEF) to incompressible fluid flow. The p-approximation functions were based on Legendre polynomials and the least squares formulation was applied to the Navier–Stokes equations in velocity–pressure–vorticity form. A comprehensive discussion of the non-linear least squares processes can be found in Reference 19. The present formulation uses p-approximation functions based on Lagrange interpolating polynomials and includes viscous stresses as variables rather than vorticity.

The formulation presented here is a direct extension of our least squares formulation for Burgers' equation,[20] which included several unique features. The non-linear minimization problem was solved by incorporating a line search with the classical Newton method. The procedure does not require the use of preconditioners[21] and was shown to perform reliably in several numerical examples. Proper treatment of the non-linear terms (convective terms in this case) in the least squares minimization (discussed later in more detail) is essential in order to derive an error functional that corresponds to the statement of the problem and is an inherent feature of the present formation that separates it from all others published previously. In the present study the LSFEF for non-linear differential equations is applied to the solution of steady state, laminar, two-dimensional, incompressible, Newtonian flow problems.

In the next section the general least squares formulation for a set of coupled non-linear differential equations is presented. In the subsequent section the steady, two-dimensional Navier–Stokes equations are cast as a set of first-order equations involving viscous stresses as auxiliary variables. The p-version LSFEF is then applied to the set of first-order equations. Numerical examples are presented to illustrate the convergence characteristics and accuracy of the present formulation. Conclusions are contained in the last section of the paper.

LEAST SQUARES FINITE ELEMENT FORMULATION

The following is a summary of the p-version least squares finite element formulation for problems described by non-linear differential equations. For a system of N differential equations the error functional I^e for an element e is defined as

$$I^e = \sum_{i=1}^{N} \int_{\Omega^e} (E_i^e)^2 \, d\Omega, \tag{1}$$

where E_i^e, $i = 1, 2, \ldots, N$, are the errors which result when the finite element approximation to the true solution is substituted into the differential equations. For a finite element mesh consisting of NE elements the total error functional I for the entire mesh is obtained by summing the element error functionals I^e:

$$I = \sum_{e=1}^{NE} I^e. \tag{2}$$

From (2) we note that I is a function of the nodal degrees of freedom $\{\delta\}$ and thus minimization of I would require that we differentiate I with respect to $\{\delta\}$ and set it to zero:

$$\frac{\partial I}{\partial\{\delta\}} = \{g\} = \sum_{e=1}^{NE} \frac{\partial I^e}{\partial\{\delta^e\}} = \sum_{e=1}^{NE} \{g^e\} = \{0\}, \tag{3}$$

where

$$\{g^e\} = 2 \sum_{i=1}^{N} \int_{\Omega^e} \left\{\frac{\partial E_i^e}{\partial\{\delta^e\}}\right\} E_i^e \, d\Omega. \tag{4}$$

We note that $\{g\}$ is indeed the variation of I and thus minimization of I would require

$$\delta I = \sum_{e=1}^{NE} \delta I^e = \{g\} = \{0\}, \tag{5}$$

where

$$\delta I^e = 2 \sum_{i=1}^{N} \int_{\Omega^e} E_i^e \delta E_i^e \, d\Omega, \tag{6}$$

with

$$\delta E_i^e = \frac{\partial E_i^e}{\partial\{\delta^e\}}. \tag{7}$$

Equations (3) or (5) represent the true least squares minimization statement.

For a $\{\delta\}$ to minimize I, $\{g\}$ must become a null vector. The least squares minimization statement ends here. In other words, in the least squares process we minimize I given by (2), which results in finding a $\{\delta\}$ that makes $\{g\}$ a null vector. In the least squares statement presented above there are no assumptions or approximations and the non-linear terms will be treated properly (see later development); for example, terms such as $u(\partial u/\partial x)$ are differentiated with respect to $\{\delta\}$ using the product rule while formulating $\{g\}$ in (3) as opposed to linearizing it by lagging u. It is important to note that finding a vector $\{\delta\}$ which satisfies $\{g\} = \{0\}$ is a separate issue and has nothing to do with the least squares minimization principle. It may also be noted that the least squares minimization principle stated above is exact and is applicable to any non-linear system of partial differential equations regardless of the nature of the non-linearities and it does not require linearization of the non-linear terms. In the remainder of the paper when we refer to linearization of the non-linear differential equations we mean linearization by lagging coefficients.

SOLUTION PROCEDURE

We consider problems described by non-linear differential partial equations. For such problems $\{g\}$ is a non-linear function of $\{\delta\}$ and our objective is to find a $\{\delta\}$ which would make $\{g\}$ a null vector. At this point several different strategies can be adopted. In the first approach $\{g\} = \{0\}$ may be recast in the following forms:

$$[K(\{\delta\})]\{\delta\} = \{f\}, \tag{8a}$$

$$[K]\{\delta\} = \{f(\{\delta\})\}, \tag{8b}$$

$$[K(\{\delta\})]\{\delta\} = f(\{\delta\}). \tag{8c}$$

In (8a) the coefficient matrix $[K]$ is non-symmetric and in general its coefficients are non-linear functions of $\{\delta\}$. In (8b) the coefficient matrix $[K]$ is symmetric and is independent of $\{\delta\}$. In this form all $\{\delta\}$-dependent terms are contained in the vector $\{f\}$. In the third form given by (8c) the coefficient matrix $[K]$ may or may not be symmetric (depending upon which terms have been transferred to the right-hand-side vector $\{f\}$, but both $[K]$ and $\{f\}$ are functions of $\{\delta\}$. The solution vector $\{\delta\}$ may be calculated by operating on (8) iteratively. In the second scheme we operate on $\{g\}$ directly. The element-by-element conjugate gradient (including preconditioned conjugate gradient) method and methods based on the Taylor series expansion of $\{g\}$ (Newton's method) fall into this second category. Here we utilize and present the details of a solution method based on the second approach.

Since the actual solution $\{\delta\}$ which makes $\{g\}$ a null vector is not known, we assume a starting solution $\{\delta_0\}$ for which in general we have

$$\{g(\{\delta_0\})\} \neq \{0\}. \tag{9}$$

Let $\{\Delta\delta\}$ be the correction to $\{\delta_0\}$ such that

$$\{g(\{\delta_0\} + \{\Delta\delta\})\} = \{0\}. \tag{10}$$

We expand (10) in a Taylor series about $\{\delta_0\}$ and retain only the linear terms in $\{\Delta\delta\}$:

$$\{g(\{\delta_0\} + \{\Delta\delta\})\} = \{g(\{\delta_0\})\} + \left.\frac{\partial\{g\}}{\partial\{\delta\}}\right|_{\{\delta_0\}} \{\Delta\delta\} = \{0\}. \tag{11}$$

From (11) we can write

$$\left[\frac{\partial\{g\}}{\partial\{\delta\}}\right]_{\{\delta_0\}}\{\Delta\delta\} = -\{g(\{\delta_0\})\}. \tag{12}$$

Substituting from (3) into (12), we obtain

$$\left[\sum_{e=1}^{NE}\left[\frac{\partial\{g^e\}}{\partial\{\delta^e\}}\right]_{\{\delta_0^e\}}\right]\{\Delta\delta\} = -\sum_{e=1}^{NE}\{g^e(\{\delta_0^e\})\}. \tag{13}$$

Let

$$[H^e] = \left[\frac{\partial\{g^e\}}{\partial\{\delta^e\}}\right] = \frac{\partial}{\partial\{\delta^e\}}\left\{\frac{\partial\{I^e\}}{\partial\{\delta^e\}}\right\}. \tag{14}$$

The coefficient matrix $[H^e]$ is called the element Hessian matrix. Using the notation in (14), we can rewrite (12) or (13) as

$$[H]_{\{\delta_0\}}\{\Delta\delta\} = -\{g(\{\delta_0\})\}, \tag{15}$$

where

$$[H] = \sum_{e=1}^{NE}[H^e]. \tag{16}$$

Using (4), we can write the following expression for the element Hessian matrix $[H^e]$:

$$[H^e] = 2\sum_{i=1}^{N}\int_{\Omega^e}\left[\left\{\frac{\partial E_i^e}{\partial\{\delta^e\}}\right\}\left\{\frac{\partial E_i^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} + E_i^e\frac{\partial}{\partial\{\delta^e\}}\left\{\frac{\partial E_i^e}{\partial\{\delta^e\}}\right\}\right]\mathrm{d}\Omega. \tag{17}$$

From (17) we note that the Hessian matrix $[H^e]$ is symmetric.

The improved value of the solution is given by

$$\{\delta\} = \{\delta_0\} + \{\Delta\delta\}. \tag{18}$$

At this point we note that if the incremental change $\{\Delta\delta\}$ is too large, updating the solution vector $\{\delta_0\}$ according to (18) may result in an increase rather than a decrease in the error functional I. Thus we update the solution vector $\{\delta_0\}$ using

$$\{\delta\} = \{\delta_0\} + \alpha\{\Delta\delta\}. \tag{19}$$

The scalar α is selected to minimize the error functional I. In our numerical computations we consider the range of α as $0 < \alpha \leqslant 2$. In this range α is incremented in steps 0·1 and for each α the error functional I is calculated. The lowest value of I and a value on either side of the lowest value are used to construct a parabolic fit from which a value of α is calculated which minimizes I. This value of α is used in (19) to obtain the updated vector $\{\delta\}$ for the next iteration. This procedure described above is termed 'Newton's method with a line search'. When $\alpha = 1$, it reduces to the classical Newton method. In the present work we use a starting vector $\{\delta_0\} = \{0\}$ and $\alpha = 1$ for the first iteration; the line search is used for subsequent iterations. In all numerical studies conducted (including the ones presented in this paper) the actual range of α was $0 < \alpha \leqslant 1{\cdot}5$. When the updated solution is in the very close neighbourhood of the true solution, α becomes very close to unity and eventually reaches unity at convergence. For a given mesh and p-levels, iterations are performed until the error functional I and each component of $\{g\}$ are below a prespecified tolerance. More details on the number of iterations and the actual tolerances used are presented with the numerical examples.

Remarks

1. First of all some discussion regarding the element Hessian matrix is in order. Numerical studies have shown that the second term in the integrand of (17), which involves the second derivative of E_i^e with respect to $\{\delta^e\}$, can be neglected without affecting the convergence of the iterative solution procedure. In fact, in many numerical studies the convergence of the iterative procedure is improved by doing so. In problems where convective terms are the only source of non-linearities (Burgers' equation, incompressible Newtonian fluid flow, etc.), obtaining explicit expressions for the second derivative of E_i^e with respect to $\{\delta^e\}$ is not too complicated. However, in applications such as non-Newtonian fluid flow where the generalized viscosity may be represented by a power behaviour, obtaining expressions for this term is algebraically very tedious and in some cases may even become a formidable task. The additional computational time needed in computing $[H^e]$ due to inclusion of this term depends upon the degree of non-linearity. For two-dimensional incompressible Newtonian flow the increase in computational time is only a small fraction, but for non-Newtonian fluid flow applications the inclusion of this term may significantly increase the computational time for $[H^e]$. Thus in summary, including the second term in the computation of $[H^e]$ increases the complexity of programming, increases the computational effort and does not improve the convergence of the iterative process. Even though at this point we are unable to offer a mathematical justification for not including the second term in the computation of $[H^e]$, however, the benefits described here amply justify computation of $[H^e]$ without the second term.

2. It should be noted that this approximation (described in item 1) in the computation of the Hessian matrix only alters the search direction in the iterative procedure and in no way effects the least squares error functional and the minimization procedure described in the earlier section.

3. We note that $\{g\} = \{0\}$ is only a necessary condition for the minimization of I. The sufficient condition to ensure that we have the minimum of I is that $[H]$ (given by (16)) must be positive definite for a $\{\delta\}$ which satisfies $\{g\} = \{0\}$, i.e. we must show that $\{\delta^*\}^T[H(\{\delta\})]\{\delta^*\} > 0$ for an arbitrary vector $\{\delta^*\}$. Numerically this is rather easy to compute and demonstrate (even though we have not done so in this paper). Rather than proving or numerically showing that $[H(\{\delta\})]$ is positive definite, we present a different reasoning and state another condition which ensures that a $\{\delta\}$ which satisfies $\{g\} = \{0\}$ indeed minimizes I.

First we examine the mathematical aspects of the problem and then later state how these should be viewed in the numerical–computational sense. We note that I represents the summation over all elements of the mesh of the sum of squares of the errors resulting from the individual differential equations integrated over each element. Thus if $\{\delta\}$ represents the solution vector for which $\{g\} = 0$, then $I = 0$ guarantees that we have indeed found a $\{\delta\}$ for which I is the global minimum, because $I = 0$ is the global minimum we are seeking. Thus $\{g\} = 0$ and $I = 0$ are the necessary and sufficient conditions for the global minimum of I.

Numerically these conditions $\{g\} = 0$ and $I = 0$ can only be satisfied in the sense of some threshold tolerance Δ. A numerically computed 'zero' depends on the word size of the computer used. On most computers, if the computed numbers are in the range 10^{-6}–10^{-12}, they can be treated as zero. Thus we could set $\Delta = 10^{-6}$–10^{-12} and then our necessary and sufficient conditions for the global minimum of I (in the numerical–computational sense) would become $\{g\} \leqslant \Delta$ and $I \leqslant \Delta$. However, in our computations we have found that the computed pressure, velocities, stresses, temperature and heat fluxes are quite accurate (up to five or six significant places) for Δ as high as 10^{-3}–10^{-4}. Thus there is no need to set Δ as low as 10^{-6}–10^{-12}, a value of 10^{-3}–10^{-4} suffices for all practical purposes. In the numerical examples presented in this paper the solution is considered converged to the correct solution when the magnitude of each component of $\{g\}$ is less than or equal to Δ and when $I \leqslant \Delta$, where Δ is set to 10^{-3}–10^{-4}. We have also found that when the condition $I \leqslant \Delta$ is satisfied, the computed numerical values of the components of $\{g\}$ are even lower than Δ. $\{g\} \nleqslant \Delta$ and $I \nleqslant \Delta$ indicate either lack of mesh refinement or inadequate p-levels or in some cases both. It is quite obvious (but nevertheless worth mentioning) that lower values of Δ will require more refined meshes and higher p-levels but would result in more significant place accuracy in the computed solution. In the numerical examples presented in this paper the mesh and p-level combinations are for $\Delta \leqslant 10^{-3}$–10^{-4}.

4. The p-version LSFEF, when applied to the system of non-linear differential equations, results in finding a $\{\delta\}$ for which $\{g\}$ must become a null vector. However, since $\{g\}$ may be a complex non-linear function of $\{\delta\}$, we apply Newton's method with a line search to find this $\{\delta\}$. In many published papers the non-linear differential equations are linearized by lagging coefficients at the onset before constructing the least squares functional.[6,9,13] Such approaches result in an approximation to the true least squares functional. Clearly, as just demonstrated, such linearization is totally unwarranted as far as the least squares minimization functional is concerned. Another important comment regarding the linearization (based on lagging coefficients) of differential equations is that this can only be done in simple cases (e.g. convective terms in momentum equations). In more complex situations such as non-Newtonian fluid flow, linearization approaches similar to those presented and advocated in References 6, 9 and 13 may result in undesirable approximation. We argue firstly that the linearization of the differential equations is an unnecessary approximation which can be easily avoided and secondly that the least squares functional corresponding to the linearized differential equations based on lagging coefficients is not the true functional corresponding to the actual non-linear equations. Any numerical scheme based on this approximate functional will result in an approximation in the calculated solution as well. In other words, a $\{\delta\}$ which makes the error functional I resulting from the linearized system less than or equal to Δ may result in the true error functional (based on actual non-linear differential equations) $I > \Delta$, indicating that this $\{\delta\}$ is not the solution of the actual non-linear problem.

5. At this point we wish to state that the application of Newton's method to find a $\{\delta\}$ for which $\{g\}$ becomes a null vector corresponding to the true least squares functional presented here is not the same as linearizing the differential equations by lagging coefficients and then constructing the least squares functional and its minimization. As we stated earlier, the construction of the least squares functional and associated minimization and the method used to find a solution vector $\{\delta\}$ that satisfies the conditions dictated by the least squares minimization principle are two separate issues. The point to note is that linearized differential equations result in a functional that does not correspond to the statement of the original problem. In our presentation we have demonstrated that the true least squares functional and minimization principle can be constructed without introducing any approximations in the differential equations and that Newton's method with a line search can indeed be used to find the solution vector $\{\delta\}$ for which $\{g\} \leqslant \Delta$ and $I \leqslant \Delta$ which in fact gives us the minimum of I we are seeking. The approximation in the computation of the Hessian matrix only effects the search direction during iterations and has absolutely no effect on the exactness of the least squares functional and minimization principle.

6. It is worth noting that the approaches based on the minimization of the quadratic functional[19] which is constructed by using the actual error functional obtained from the

non-linear differential equations result in the same element Hessian matrix as our formulation does after deleting the terms containing the second derivative of E_i^e with respect to $\{\delta^e\}$. This procedure is also referred to as linearization based on Newton's method. In other words, the Newton linearization before minimization gives the same Hessian matrix as our formulation when the term containing the second derivative of E_i^e with respect to $\{\delta^e\}$ is deleted in equation (17).

APPLICATION TO THE NAVIER–STOKES EQUATIONS

The general formulation presented in the previous section may be easily applied to the Navier–Stokes equations. Here we consider steady, two-dimensional, incompressible, Newtonian flow. The Navier–Stokes equations can be expressed as the following set of first-order differential equations:

$$
\begin{aligned}
&\frac{\partial \hat{u}}{\partial \hat{x}} + \frac{\partial \hat{v}}{\partial \hat{y}} = 0,\\
&\hat{u}\frac{\partial \hat{u}}{\partial \hat{x}} + \hat{v}\frac{\partial \hat{v}}{\partial \hat{y}} + \frac{1}{\rho}\frac{\partial \hat{p}}{\partial \hat{x}} - g_x - \frac{1}{\rho}\frac{\partial \hat{\tau}_{xx}}{\partial \hat{x}} - \frac{1}{\rho}\frac{\partial \hat{\tau}_{xy}}{\partial \hat{y}} = 0,\\
&\hat{u}\frac{\partial \hat{v}}{\partial \hat{x}} + \hat{v}\frac{\partial \hat{v}}{\partial \hat{y}} + \frac{1}{\rho}\frac{\partial \hat{p}}{\partial \hat{y}} - g_y - \frac{1}{\rho}\frac{\partial \hat{\tau}_{xy}}{\partial \hat{x}} - \frac{1}{\rho}\frac{\partial \hat{\tau}_{yy}}{\partial \hat{y}} = 0,\\
&\hat{\tau}_{xx} - 2\mu\frac{\partial \hat{u}}{\partial \hat{x}} = 0,\\
&\hat{\tau}_{xy} - \mu\left(\frac{\partial \hat{u}}{\partial \hat{y}} + \frac{\partial \hat{v}}{\partial \hat{x}}\right) = 0,\\
&\hat{\tau}_{yy} - 2\mu\frac{\partial \hat{v}}{\partial \hat{y}} = 0.
\end{aligned}
\tag{20}
$$

The purpose of reducing the equations to a set of first-order equations is to permit the use of C^0 approximation functions. This approach is completely analogous to the approach used for Burgers' equation.[20]

Assuming constant values for ρ, μ, g_x and g_y, the following dimensionless variables can be used:

$$
\begin{aligned}
&x = \frac{\hat{x}}{L}, \qquad y = \frac{\hat{y}}{L}, \qquad u = \frac{\hat{u}}{V_0}, \qquad v = \frac{\hat{v}}{V_0},\\
&P = \frac{\hat{P} - P_0}{\rho V_0^2} - (g_x)\cdot x - (g_y)\cdot y,\\
&\tau_{xx} = \frac{\hat{\tau}_{xx}}{\tau_0}, \qquad \tau_{xy} = \frac{\hat{\tau}_{xy}}{\tau_0}, \qquad \tau_{yy} = \frac{\hat{\tau}_{yy}}{\tau_0},
\end{aligned}
\tag{21}
$$

where $\tau_0 = \mu V_0/L$. Using these dimensionless variables, equations (20) can be expressed in the following dimensionless form:

$$
\begin{aligned}
&\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0,\\
&u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+\frac{\partial P}{\partial x}-\frac{1}{Re}\frac{\partial \tau_{xx}}{\partial x}-\frac{1}{Re}\frac{\partial \tau_{xy}}{\partial_y}=0,\\
&u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+\frac{\partial P}{\partial y}-\frac{1}{Re}\frac{\partial \tau_{xy}}{\partial x}-\frac{1}{Re}\frac{\partial \tau_{yy}}{\partial_y}=0,\\
&\tau_{xx}-2\frac{\partial u}{\partial x}=0,\\
&\tau_{xy}-\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)=0,\\
&\tau_{yy}-2\frac{\partial v}{\partial y}=0.
\end{aligned}
\tag{22}
$$

If P^h, u^h, v^h, τ^h_{xx}, τ^h_{xy} and τ^h_{yy} represent the finite element approximation to the true solution $(P, u, v, \tau_{xx}, \tau_{xy}, \tau_{yy})$, then the errors are given by

$$
\begin{aligned}
E^e_1&=\frac{\partial u^h}{\partial x}+\frac{\partial v^h}{\partial y},\\
E^e_2&=u^h\frac{\partial u^h}{\partial x}+v^h\frac{\partial u^h}{\partial y}+\frac{\partial P^h}{\partial x}-\frac{1}{Re}\frac{\partial \tau^h_{xx}}{\partial x}-\frac{1}{Re}\frac{\partial \tau^h_{xy}}{\partial y},\\
E^e_3&=u^h\frac{\partial v^h}{\partial x}+v^h\frac{\partial v^h}{\partial y}+\frac{\partial P^h}{\partial y}-\frac{1}{Re}\frac{\partial \tau^h_{xy}}{\partial x}-\frac{1}{Re}\frac{\partial \tau^h_{yy}}{\partial y},\\
E^e_4&=\tau^h_{xx}-2\frac{\partial u^h}{\partial x},\\
E^e_5&=\tau^h_{xy}-\left(\frac{\partial u^h}{\partial y}+\frac{\partial v^h}{\partial x}\right),\\
E^e_6&=\tau^h_{yy}-2\frac{\partial v^h}{\partial y}.
\end{aligned}
\tag{23}
$$

If we consider the same order of approximation for all field variables, then we can write

$$\Psi^h=[N]\{\Psi\}, \tag{24}$$

where $[N]$ is the hierarchical approximation function matrix and $\{\Psi\}$ are the hierarchical nodal degrees of freedom for a field variable Ψ, e.g. $u^h=[N]\{u\}$, $v^h=[N]\{v\}$, etc. The derivation of the hierarchical approximation functions and nodal variables for a nine-node two-dimensional element (Figure 1) is summarized in Reference 22. The contents of the approximation function matrix $[N]$ and the nodal degree of freedom vector $\{\Psi\}$ are presented in Table I.

In addition to the errors given by equations (23), the formulation presented in the previous section requires expressions for the derivatives of the errors with respect to the $\{\delta^e\}$-vector. In this case

$$\{\delta^e\}^{\mathrm{T}}=[\{P\}^{\mathrm{T}},\{u\}^{\mathrm{T}},\{v\}^{\mathrm{T}},\{\tau_{xx}\}^{\mathrm{T}},\{\tau_{xy}\}^{\mathrm{T}},\{\tau_{yy}\}^{\mathrm{T}}] \tag{25}$$

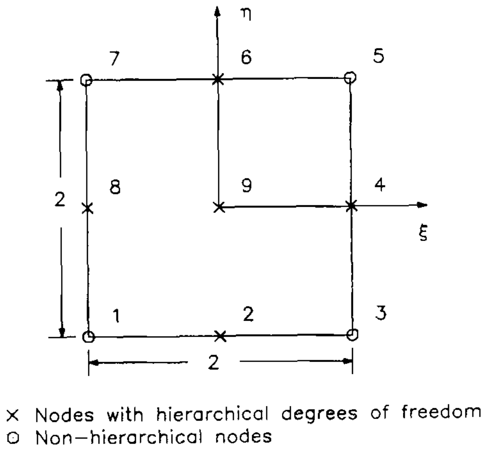

Figure 1. A nine-node p-version element in natural co-ordinate space (ξ, η)

and

$$\left\{\frac{\partial E_i^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} = \left[\left\{\frac{\partial E_i^e}{\partial\{P\}}\right\}^{\mathrm{T}}, \left\{\frac{\partial E_i^e}{\partial\{u\}}\right\}^{\mathrm{T}}, \left\{\frac{\partial E_i^e}{\partial\{v\}}\right\}^{\mathrm{T}}, \left\{\frac{\partial E_i^e}{\partial\{\tau_{xx}\}}\right\}^{\mathrm{T}}, \left\{\frac{\partial E_i^e}{\partial\{\tau_{xy}\}}\right\}^{\mathrm{T}}, \left\{\frac{\partial E_i^e}{\partial\{\tau_{yy}\}}\right\}^{\mathrm{T}}\right]. \qquad (26)$$

Table I. Hierarchical approximation functions and nodal variables for a nine-node p-version element

Node	Approximation functions	Hierarchical nodal variables $\{\Psi\}$	Order of derivatives
1	$N_1^{1\eta}, N_1^{1\xi}$	$(\Psi)_1$	0
3	$N_1^{1\eta}, N_3^{1\xi}$	$(\Psi)_3$	0
5	$N_3^{1\eta}, N_3^{1\xi}$	$(\Psi)_5$	0
7	$N_3^{1\eta}, N_1^{1\xi}$	$(\Psi)_7$	0
2	$N_1^{1\eta}, N_2^{i\xi}$	$\frac{1}{i!}\left(\frac{\partial^i\Psi}{\partial\xi^i}\right)_{\xi=0} = (\Psi_{\xi^i})_2$	$i = 2, 3, \ldots, p_\xi$
6	$N_3^{1\eta}, N_2^{i\xi}$	$\frac{1}{i!}\left(\frac{\partial^i\Psi}{\partial\xi^i}\right)_{\xi=0} = (\Psi_{\xi^i})_6$	$i = 2, 3, \ldots, p_\xi$
4	$N_2^{j\eta}, N_3^{1\xi}$	$\frac{1}{j!}\left(\frac{\partial^j\Psi}{\partial\eta^j}\right)_{\eta=0} = (\Psi_{\eta^j})_4$	$j = 2, 3, \ldots, p_\eta$
8	$N_2^{j\eta}, N_1^{1\xi}$	$\frac{1}{j!}\left(\frac{\partial^j\Psi}{\partial\eta^j}\right)_{\eta=0} = (\Psi_{\eta^j})_8$	$j = 2, 3, \ldots, p_\eta$
9	$N_2^{j\eta}, N_2^{i\xi}$	$\frac{1}{i!j!}\left[\frac{\partial^j}{\partial\eta^j}\left(\frac{\partial^i\Psi}{\partial\xi^i}\right)\right]_{\xi=0,\eta=0} = (\Psi_{\xi^i\eta^j})_9$	$i = 2, 3, \ldots, p_\xi$ $j = 2, 3, \ldots, p_\eta$

Using equation (24) with $\Psi = P$, u, v, τ_{xx}, τ_{xy} and τ_{yy}, the required expressions in (26) can be written directly:

$$\left\{\frac{\partial E_1^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} = \left[[0], \left[\frac{\partial N}{\partial x}\right], \left[\frac{\partial N}{\partial y}\right], [0], [0], [0]\right], \tag{27}$$

$$\left\{\frac{\partial E_2^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} = \left[\left[\frac{\partial N}{\partial x}\right], u^h\left[\frac{\partial N}{\partial x}\right] + \frac{\partial u^h}{\partial x}[N] + v^h\left[\frac{\partial N}{\partial y}\right], \frac{\partial u^h}{\partial y}[N], -\frac{1}{Re}\left[\frac{\partial N}{\partial x}\right], -\frac{1}{\mathrm{Re}}\left[\frac{\partial N}{\partial y}\right], [0]\right], \tag{28}$$

$$\left\{\frac{\partial E_3^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} = \left[\left[\frac{\partial N}{\partial y}\right], \frac{\partial v^h}{\partial x}[N], u^h\left[\frac{\partial N}{\partial x}\right] + v^h\left[\frac{\partial N}{\partial y}\right] + \frac{\partial v^h}{\partial y}[N], [0], -\frac{1}{Re}\left[\frac{\partial N}{\partial x}\right], -\frac{1}{Re}\left[\frac{\partial N}{\partial y}\right]\right], \tag{29}$$

$$\left\{\frac{\partial E_4^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} = \left[[0], -2\left[\frac{\partial N}{\partial x}\right], [0], [N], [0], [0]\right], \tag{30}$$

$$\left\{\frac{\partial E_5^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} = \left[[0], -\left[\frac{\partial N}{\partial y}\right], -\left[\frac{\partial N}{\partial x}\right], [0], [N], [0]\right], \tag{31}$$

$$\left\{\frac{\partial E_6^e}{\partial\{\delta^e\}}\right\}^{\mathrm{T}} = \left[[0], [0], -2\left[\frac{\partial N}{\partial y}\right], [0], [0], [N]\right]. \tag{32}$$

The errors given by equations (23) and the derivatives of the errors with respect to the degree of freedom vector $\{\delta^e\}$ given by equations (27)–(32) are used to compute the element vector $\{g^e\}$ (equation (4)) and the Hessian matrix $[H^e]$ (equation (17)). The integrals are calculated using Gaussian quadrature. Exact integration requires the use of $2p + 1$ integration points in the ξ- and η-directions, where p is the order of the polynomial approximation. We have found that the use of $p + 2$ integration points introduces very little error in the integration ($< 1\%$) but greatly reduces the required computations. In the numerical examples which follow, the integrals were computed using the '$p + 2$ rule'.

The system of algebraic equations in (15) is solved using the wavefront method. The CPU time required in this method is a function of the maximum wavefront and the RMS wavefront. We have used this procedure successfully to solve problems with maximum wavefronts in excess of 3000 degrees of freedom. However, this solution method may not be the most efficient (in terms of storage and CPU time) method of solving (15) for meshes containing large numbers of degrees of freedom. Here we elected to use this method because it was easily available from other finite element subsystems in our finite element software system. The efficiency aspects of the formulation presented are neither claimed nor discussed in this paper. These aspects are currently being investigated and will be presented in a subsequent paper.

NUMERICAL EXAMPLES

Two numerical examples are presented in this section to demonstrate the accuracy and convergence characteristics of the present p-version least squares formulation. For both examples the results are compared with published results.

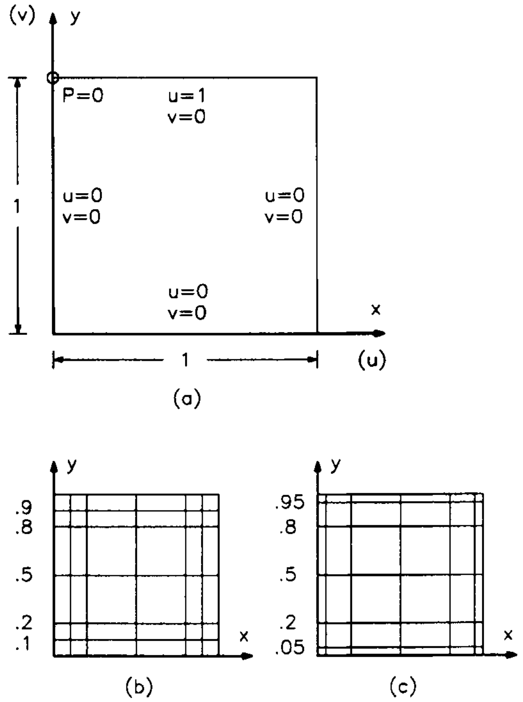

Figure 2. Driven cavity problem and finite element models: (a) schematic of driven cavity problem; (b) finite element model A of driven cavity; (c) finite element model B of driven cavity

Example 1. Driven cavity

Here we consider the 'driven cavity' problem, a schematic of which is shown in Figure 2(a). The boundary conditions for the dimensionless velocities u and v and the pressure P are also shown in Figure 2(a). Figures 2(b) and 2(c) show two different graded finite element meshes using 36 elements. The main difference between the two meshes is in the sizes of the elements adjacent to the four boundaries of the cavity. In mesh A these elements are 0·1 units whereas in mesh B they are 0·05 units.

For both meshes the p-levels in the ξ- and η-directions ($p_\xi = p_\eta$) were uniformly increased from three to nine and the results were computed for $Re = 1000$. Figure 3 shows plots for error function I versus total degrees of freedom. Both meshes yield monotonic curves and quite low values of I for p-levels beyond six, but mesh B yields slightly lower values of I than mesh A for the same number of degrees of freedom. Both meshes yield just about the same convergence rate (slopes of I versus degrees of freedom (DOF) curves), as is obvious from the graphs shown in Figure 3. On the sole basis of the graphs in Figure 3, we can conclude that mesh B is slightly better. Newton's method with a line search required less than 10 iterations for a convergence tolerance $O(10^{-4})$ or less for both $\{\delta\}$ and I.

The solution for u along the vertical centreline ($x = 0{\cdot}5$) for mesh A is presented in Figure 4 and the solution for v along the horizontal centreline ($y = 0{\cdot}5$) for mesh A is shown in Figure 5. The results reported by Ghia *et al.*[23] are used as a reference and are also presented in Figures 4 and 5. The dimensionless pressure values along the vertical and horizontal centrelines for mesh A are shown in Figures 6 and 7. We note that the solutions for u and v are converged and show

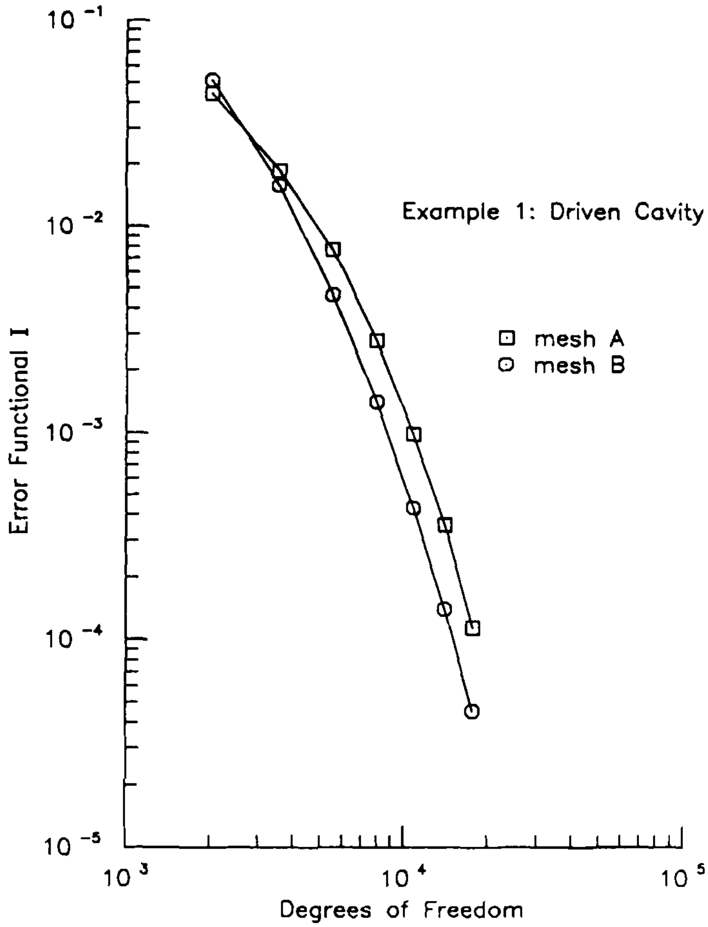

Figure 3. p-Convergence of error functional I for driven cavity problem

excellent agreement with the reference solution,[23] but the solution for pressure is not yet converged at a p-level of nine.

As part of the solution postprocessing, we compute a quantity called the element mean square error (MSE). For an element e we define

$$\mathrm{MSE} = \frac{I^e}{A^e},$$

where A^e is the area of element e. We have found that MSE values are very useful for adaptive mesh refinement. For this particular problem the largest MSE values were observed in elements occupying the upper corners of the cavity. On the basis of these MSE values, the element sizes were adjusted to generate mesh B. A symmetric mesh was used even though mesh refinement was not actually needed in the lower corners. The graphs of $u(x = 0{\cdot}5, y)$ and $v(x, y = 0{\cdot}5)$ for mesh B are shown in Figures 8 and 9 along with the reference solution. Note that the present solution and the reference solution agree at every point. Dimensionless pressure values along the vertical and horizontal centrelines for mesh B are shown in Figures 10 and 11.

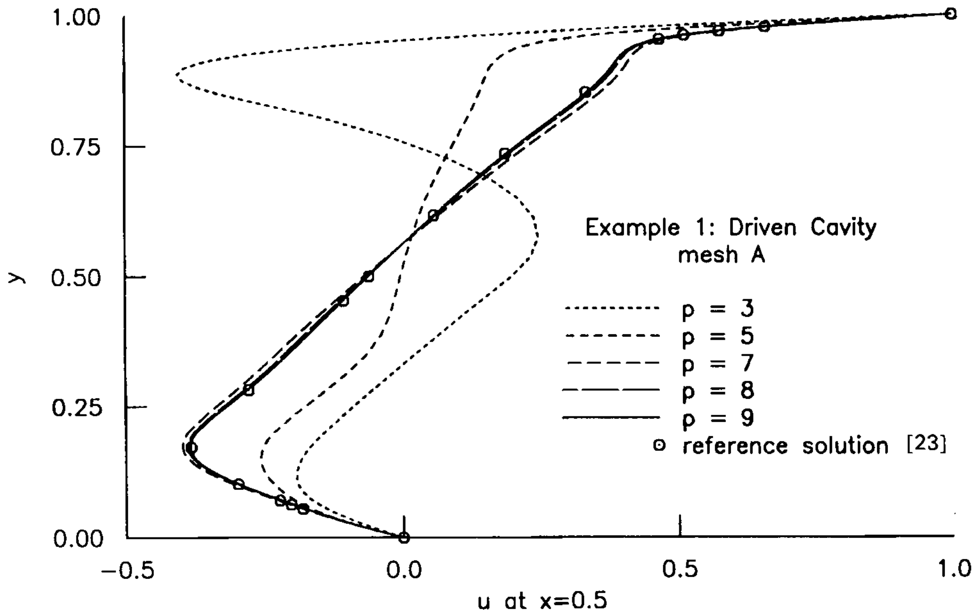

Figure 4. Plots of $u(x = 0{\cdot}5, y)$ at $Re = 1000$ using mesh A

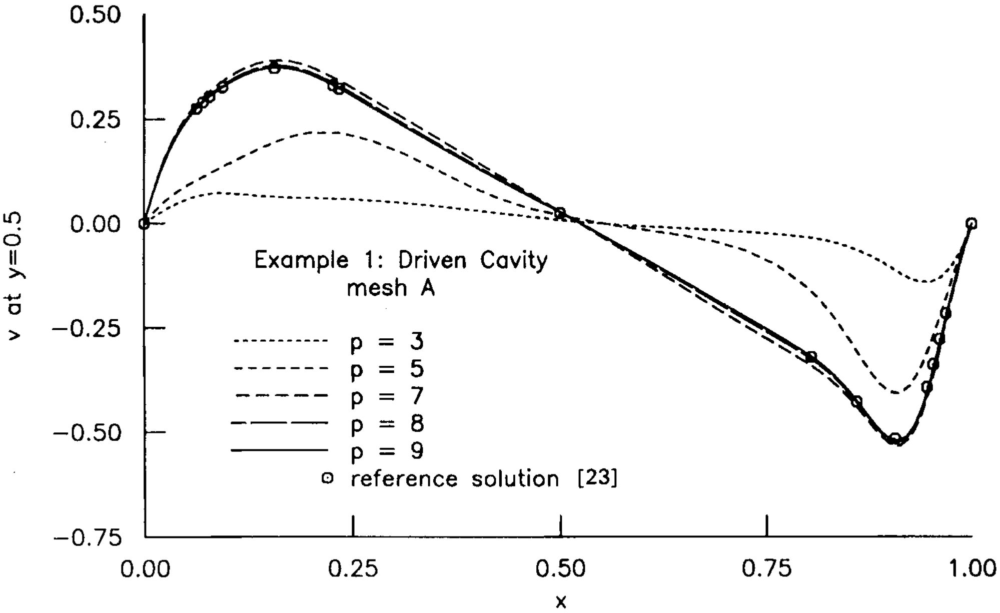

Figure 5. Plots of $v(x, y = 0{\cdot}5)$ at $Re = 1000$ using mesh A

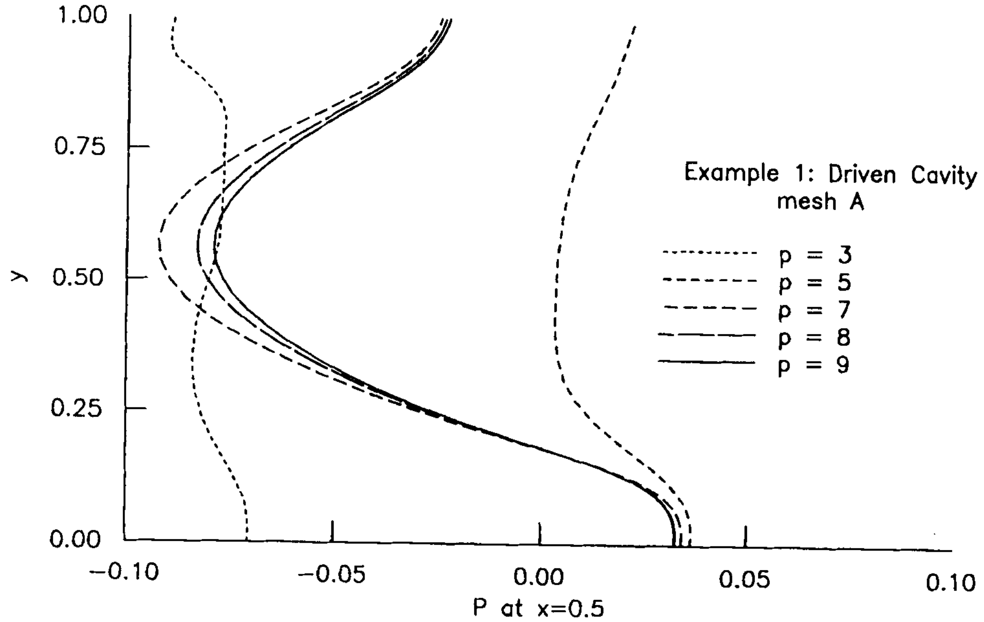

Figure 6. Plots of $P(x = 0{\cdot}5, y)$ at $Re = 1000$ using mesh A

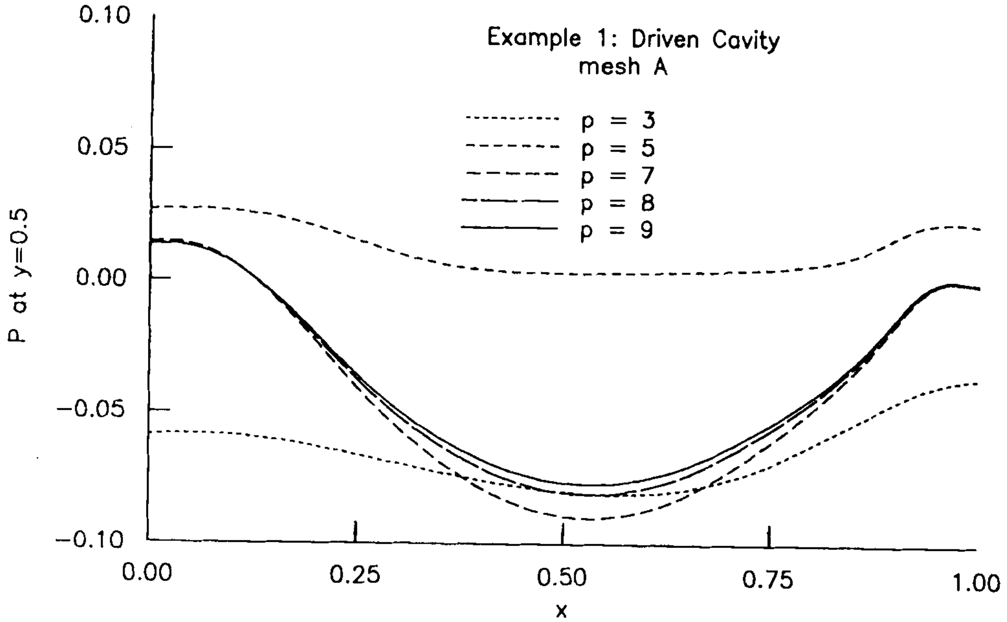

Figure 7. Plots of $P(x, y = 0{\cdot}5)$ at $Re = 1000$ using mesh A

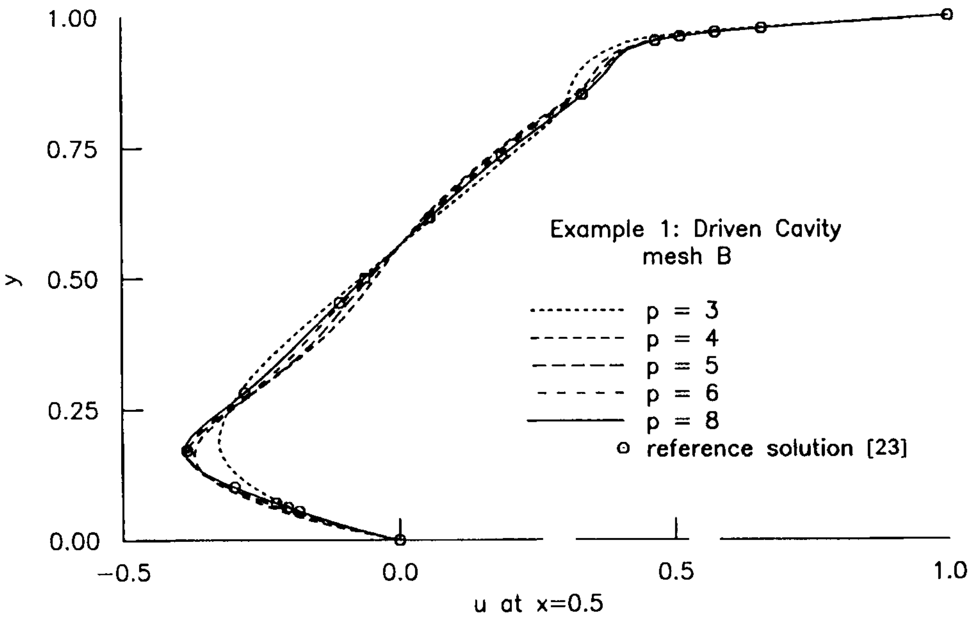

Figure 8. Plots of $u(x = 0{\cdot}5, y)$ at $Re = 1000$ using mesh B

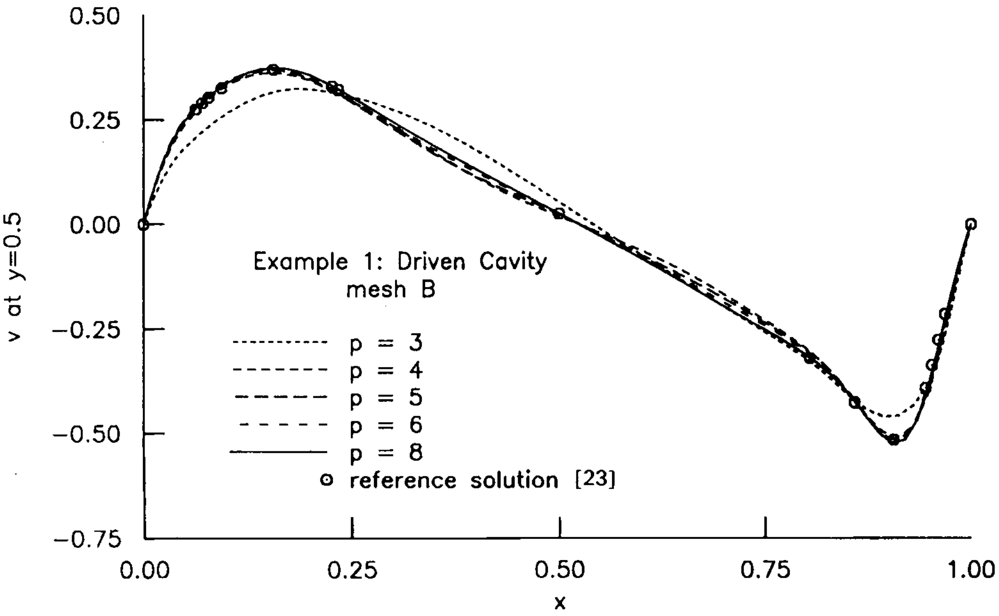

Figure 9. Plots of $v(x, y = 0{\cdot}5)$ at $Re = 1000$ using mesh B

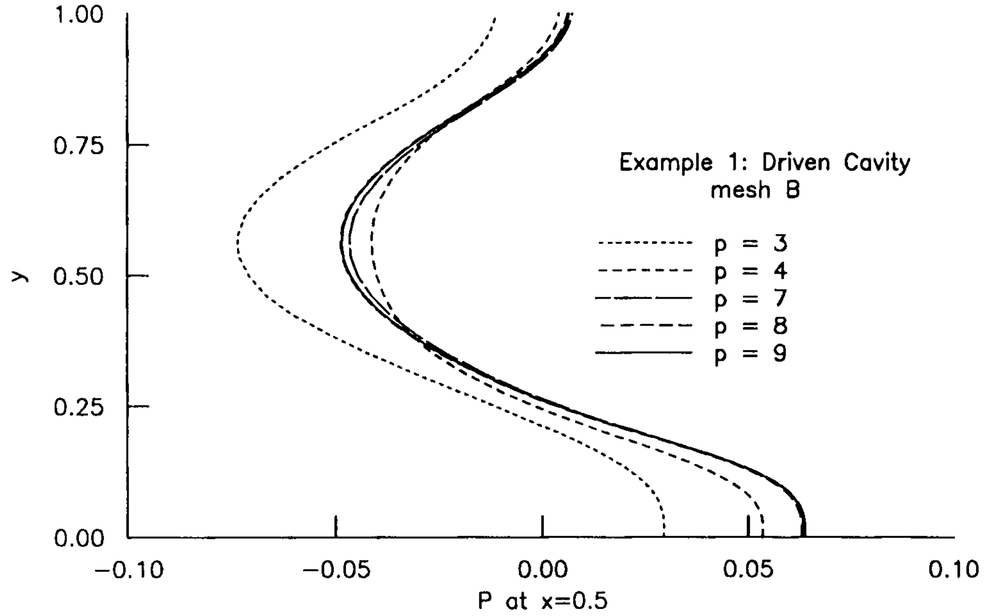

Figure 10. Plots of $P(x = 0{\cdot}5, y)$ at $Re = 1000$ using mesh B

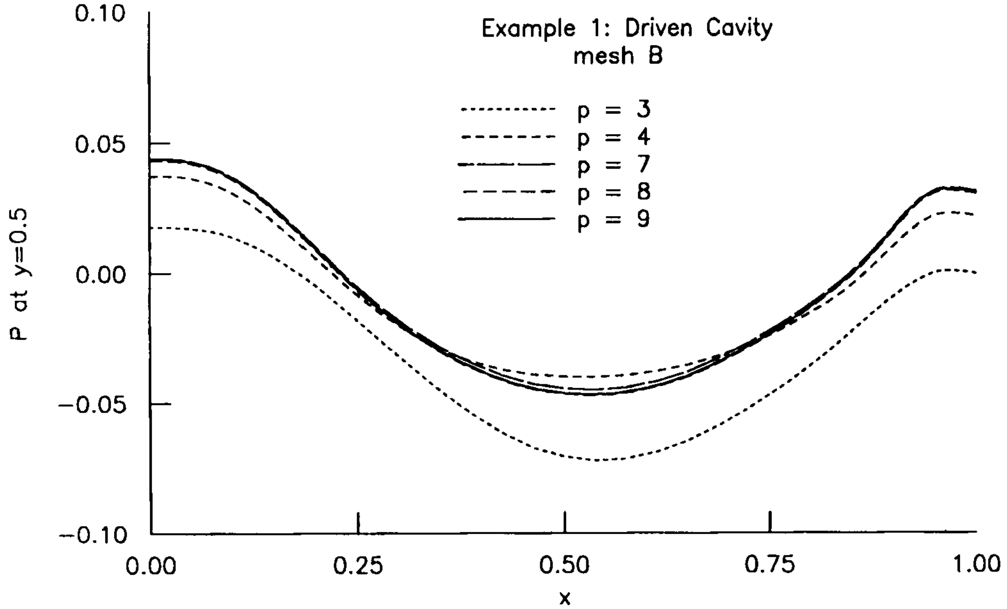

Figure 11. Plots of $P(x, y = 0{\cdot}5)$ at $Re = 1000$ using mesh B

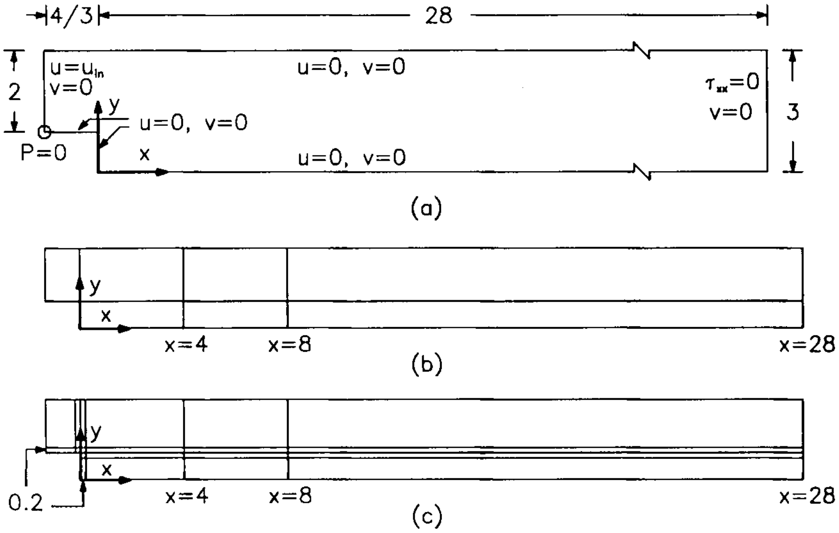

Figure 12. Asymmetric sudden expansion problem and finite element models: (a) schematic of asymmetric expansion problem; (b) finite element model A of asymmetric expansion; (c) finite element model B of asymmetric expansion

From the velocity graphs for meshes A and B (Figures 4, 5 and 8, 9) we note that mesh A produces quite inaccurate values of u and v for lower p-levels whereas the mesh B results for even a p-level of four are quite good. It is worth noting that from the I versus DOF graphs (Figure 3) it might seem that there is only little improvement in the I-value for mesh B but the resulting improvements in the values of u and v are dramatic for lower p-levels. The error functional versus DOF graphs serve as important indicators of the quality of the total solution. From these graphs the best solution can be selected without knowledge of the exact solution.

Example 2. Asymmetric sudden expansion

Here we examine a 3:2 asymmetric sudden expansion problem and compare our numerical results with the experimental results of Denham and Patrick.[24] We consider the cases of lowest and highest Reynolds number from the experiment, $Re = 73$ and 229 respectively, which are based on the mean velocity and channel half-width at the inlet. The problem geometry and boundary conditions are shown in Figure 12(a). The co-ordinates are expressed in step height units. At $Re = 73$ the inlet velocity profile was taken to be parabolic. The actual experiment values were very close to a parabolic profile. At $Re = 229$ the inlet boundary conditions were determined using a least squares fit to the actual inlet velocity measurements. The pressure was constrained at a single node and fully developed conditions ($v = 0$, $\tau_{xx} = 0$) were imposed 28 units downstream.

Initially the seven-element mesh (mesh A) shown in Figure 12(b) was used. At $Re = 73$ it was observed that the MSE values were largest for the elements adjacent to the corner point at $x = 0$, $y = 1$. To refine the mesh, an additional row and column of elements were added next to the corner point. It was again observed that the largest MSE values occurred in the elements located next to the corner point. The size of these elements was reduced until the solution no longer changed. The final mesh (mesh B) is shown in Figure 12(c). Newton's method with a line search took less than 10 iterations for a convergence tolerance $O(10^{-3})$ or less for both $\{\delta\}$ and I.

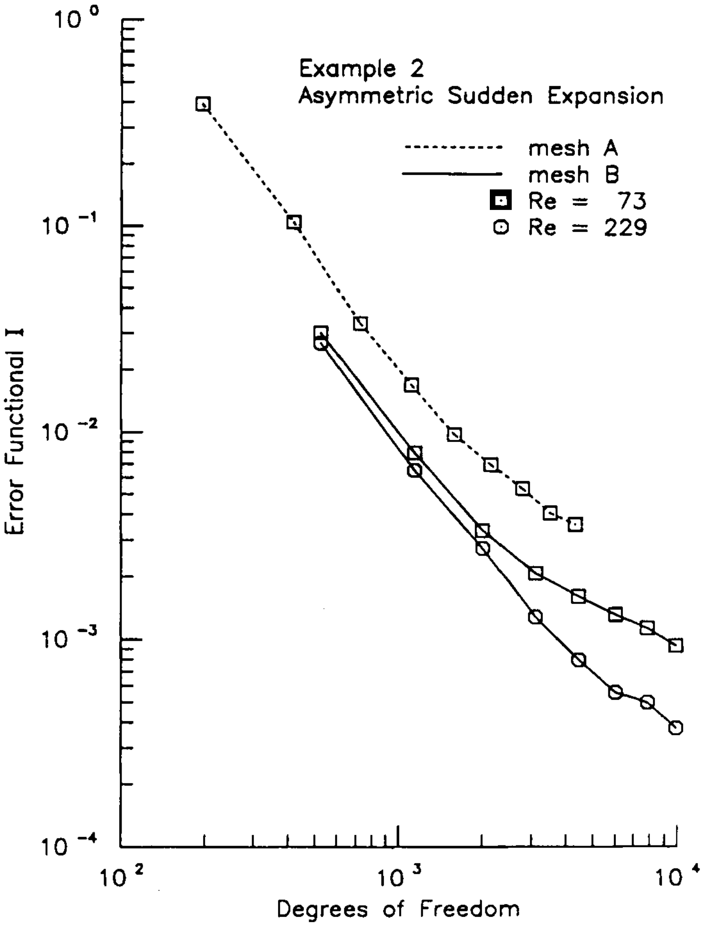

Figure 13. *p*-Convergence of the error functional *I* for asymmetric sudden expansion problem

The error functional versus DOF graphs obtained using uniform *p*-refinement (*p*-convergence of *I*) are shown in Figure 13. The use of mesh B clearly results in much lower error functional values. The final error functional *I* is less than 10^{-3} for both $Re = 73$ and 229. In Figure 14 the dimensionless shear stress τ_{xy} obtained using mesh A is plotted along the lower surface ($y = 0$) for several *p*-levels. Note that different solutions are obtained using mesh A and mesh B and thus the recirculation zone length L_R prediction is somewhat different. In Figure 15 several τ_{xy} solution curves obtained using mesh B are presented. The curves are all smooth and correctly predict a value of $\tau_{xy} = 0$ at $x = 0$. The τ_{xy} results obtained for $Re = 229$ are shown in Figure 16. For this case the recirculation zone length prediction of $L_R = 9{\cdot}7$ agrees very well with the experimental results. At $Re = 73$, however, the recirculation zone length prediction of $L_R = 5{\cdot}3$ is different from the experimental value of $L_R \approx 4{\cdot}0$. Other researchers have also predicted larger recirculation zones for $Re = 73$. For example, Hackman *et al.*[25] predict the recirculation zone length to be about 5·6 units.

The converged velocity profiles at $x = 0{\cdot}0$, $0{\cdot}8$, $2{\cdot}0$, $4{\cdot}0$, $6{\cdot}0$, $8{\cdot}0$ and $10{\cdot}0$ are presented in Figures 17–23. The agreement between the experimental and numerical results is good, although

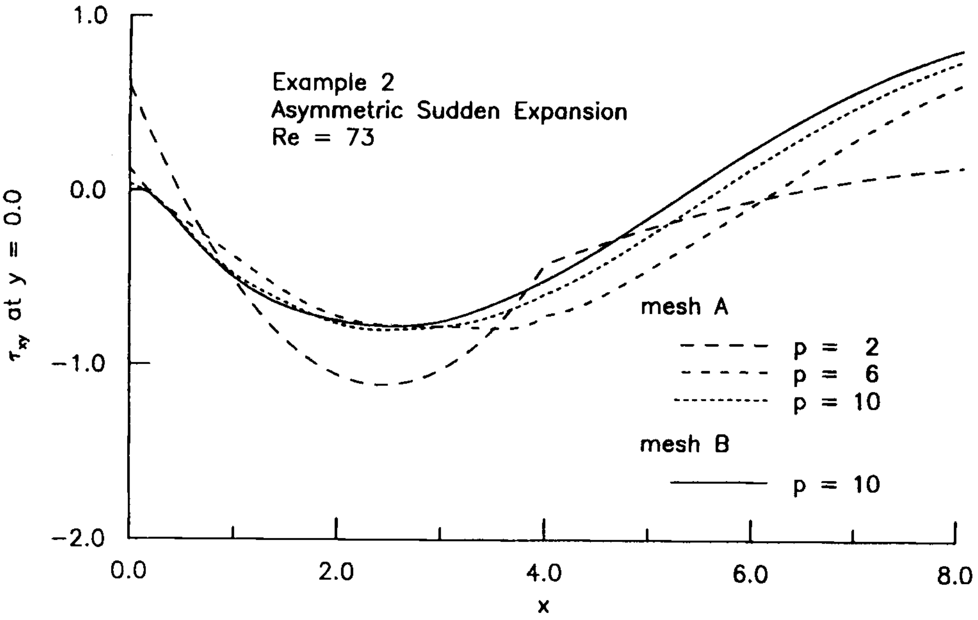

Figure 14. Plots of $\tau_{xy}(x, y = 0)$ at $Re = 73$ using mesh A

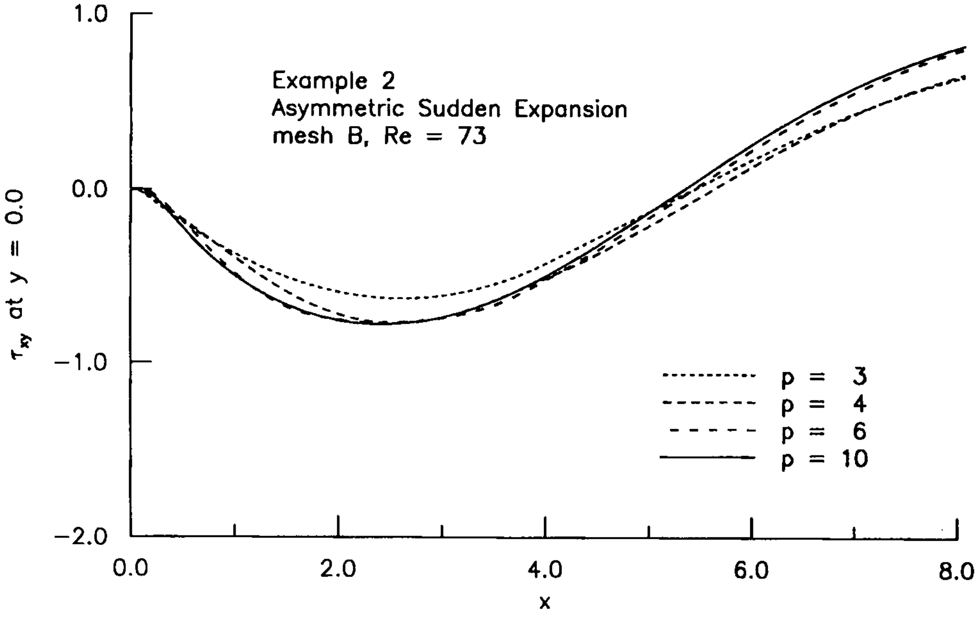

Figure 15. Plots of $\tau_{xy}(x, y = 0)$ at $Re = 73$ using mesh B

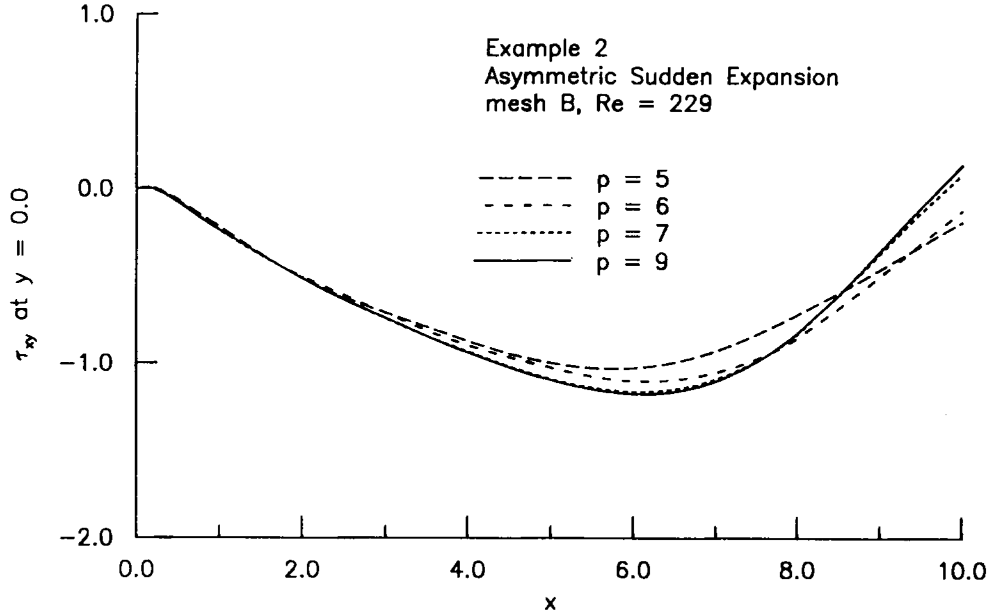

Figure 16. Plots of $\tau_{xy}(x, y = 0)$ at $Re = 229$ using mesh B

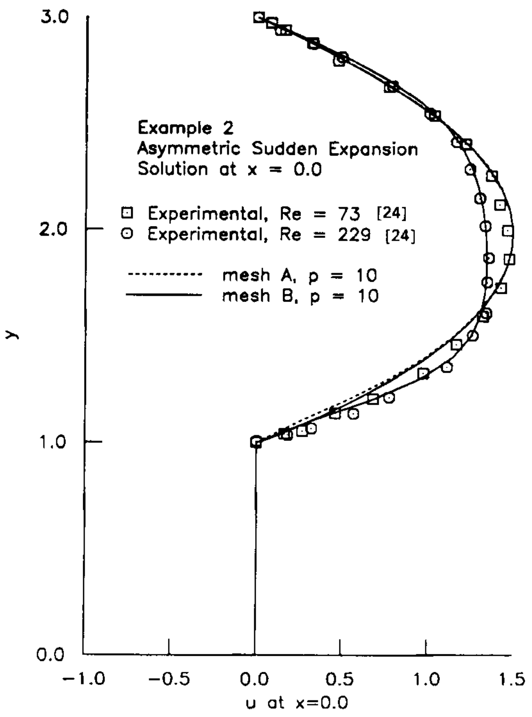

Figure 17. Velocity profiles at $x = 0{\cdot}0$

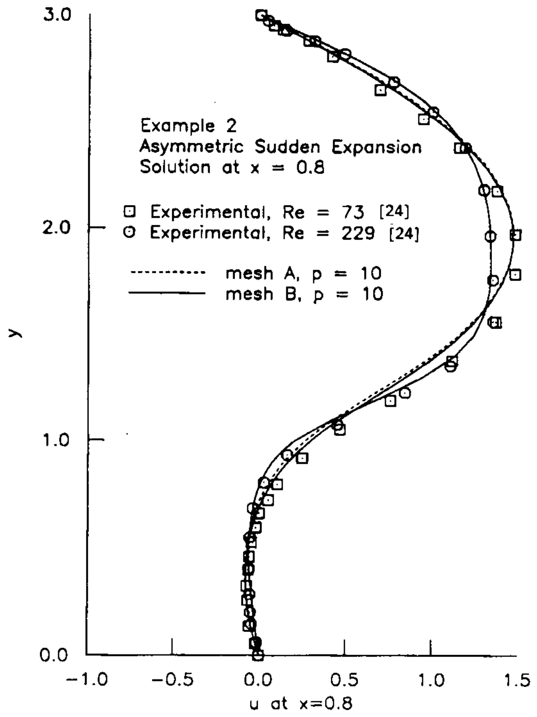

Figure 18. Velocity profiles at $x = 0{\cdot}8$

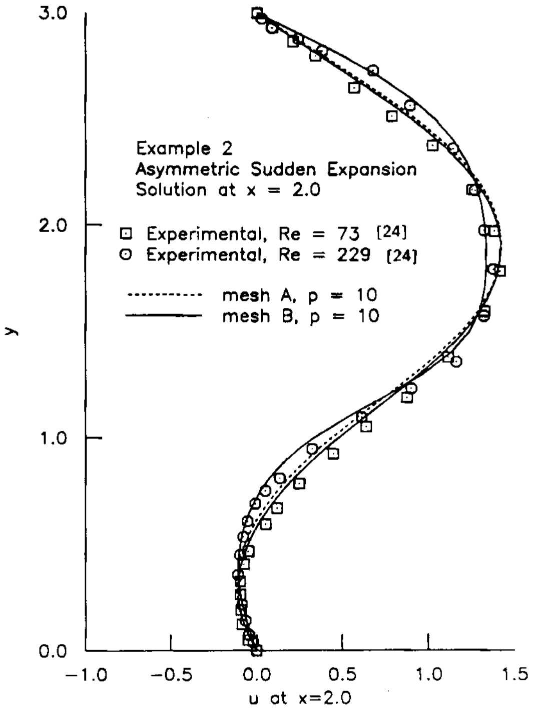

Figure 19. Velocity profiles at $x = 2{\cdot}0$

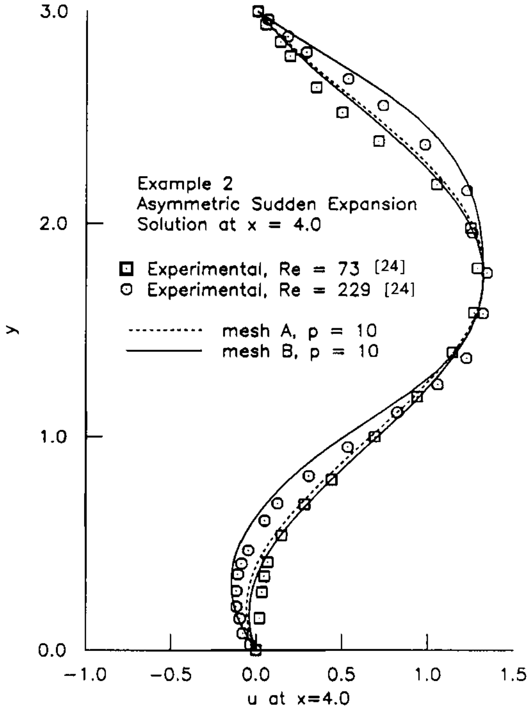

Figure 20. Velocity profiles at $x = 4{\cdot}0$

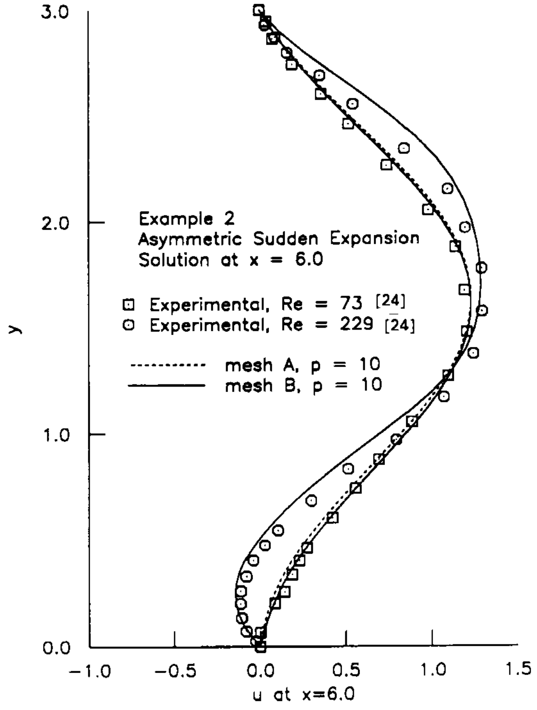

Figure 21. Velocity profiles at $x = 6{\cdot}0$

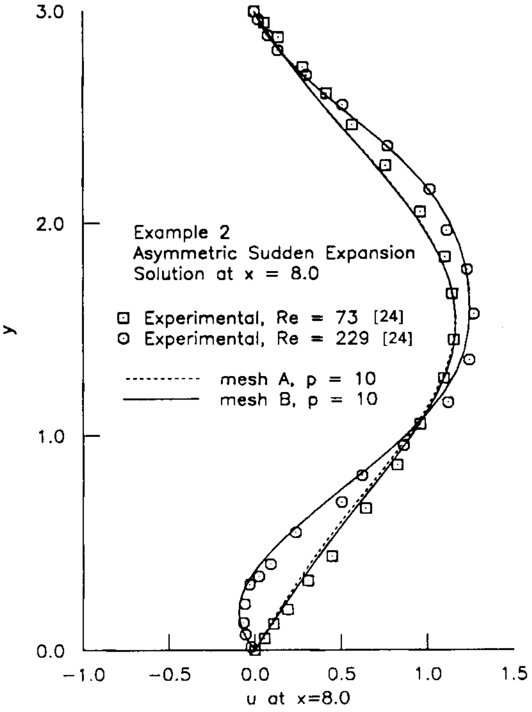

Figure 22. Velocity profiles at $x = 8{\cdot}0$

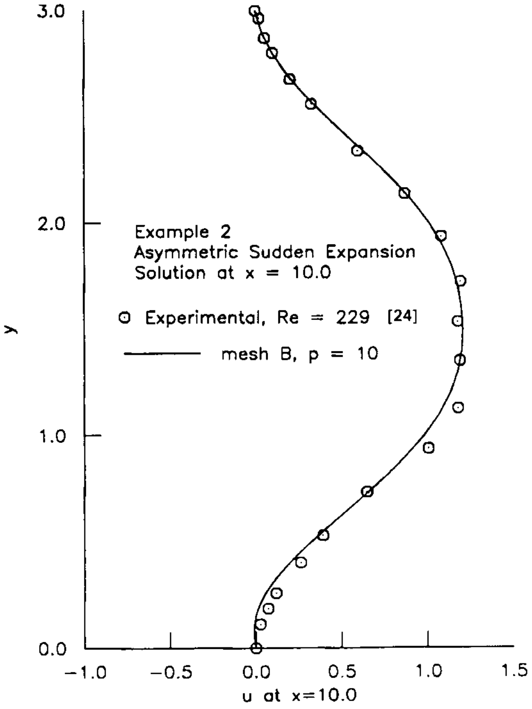

Figure 23. Velocity profiles at $x = 10{\cdot}0$

there are some obvious differences (but well within acceptable tolerances). Some error ($\approx 2\%$) may also have been introduced in the process of extracting the velocity values from the graphical results of Denham and Patrick.[24]

CONCLUSIONS

A true p-version least squares minimization procedure has been presented for coupled non-linear partial differential equations. In the development of the least squares error functional the actual non-linear partial differential equations are utilized without linearizing the non-linear terms or introducing any other approximations. This procedure has been applied to the specific case of two- dimensional, incompressible, Newtonian, steady state flow described by Navier–Stokes equations. For non-linear problems the least squares minimization principle requires finding a $\{\delta\}$ which makes $\{g\}$ (a vector of partial derivatives of the error functional with respect to the global nodal degrees of freedom) a null vector. This is accomplished by utilizing Newton's method with a line search. Two numerical examples are presented to demonstrate the accuracy and convergence characteristics of the method. The following specific conclusions can be drawn based on this research.

The least squares minimization procedure presented here is applicable to any set of linear or non-linear differential equations and will result in the true least squares functional free of assumptions and approximations. Linearization of the non-linear differential equations by lagging coefficients before constructing the least squares functional unnecessarily destroys the true least squares functional that corresponds to the actual problem at hand.

The construction of the least squares functional and finding a solution vector $\{\delta\}$ that satisfies the necessary and sufficient conditions for minimization are two theoretically unrelated issues. Assumptions or approximations employed in devising a solution procedure to find such a $\{\delta\}$ have absolutely no effect on the exactness of the least squares functional to be minimized.

The condition $\{g(\{\delta\})\} = (0)$ for a solution vector $\{\delta\}$ only represents the necessary conditions for the minimum of I. The sufficient conditions is given by $\{\delta^*\}^{\mathrm{T}}[H(\{\delta\})]\ \{\delta^*\} > 0$ for an arbitrary vector $\{\delta^*\}$. Since $I = 0$ is the minimum we are seeking, a solution vector $\{\delta\}$ which satisfies $I = 0$ will automatically satisfy the sufficiency condition of the positive definiteness of the Hessian matrix $[H]$. Thus the sufficient condition given by $\{\delta^*\}^{\mathrm{T}}[H(\{\delta\})]\ \{\delta^*\} > 0$ can be replaced with $I = 0$. In the numerical–computational sense $\{g\} \leqslant \Delta$ and $I \leqslant \Delta$ (where Δ may be chosen to be $O(10^{-3})$ or $O(10^{-4})$) are the necessary and sufficient conditions for the numerically global minimum of I.

Making an approximation in the computation of the Hessian matrix $[H^e]$ (neglecting the term containing the second derivative of E_i^e with respect to $\{\delta^e\}$) only alters the search directions during the Newton iteration procedure and has absolutely no effect on the exactness of the least squares minimization functional. This approximation has the beneficial effects of simplifying the programming, speeding up the computations and accelerating the convergence of the Newton method with a line search.

The least squares formulation automatically provides a measure of the solution error through the error functional. The element error functional values are very useful in adaptive mesh refinement or adaptive p-level changes. The total error functional is a monotonic function of the total degrees of freedom as the p-levels are increased for a fixed mesh.

The computational results agree very well with both experimental measurements and other numerical solutions considered to be quite accurate.

In summary, the *p*-version least squares formulation presented here produces excellent results, has good convergence characteristics and provides an accurate tool for numerical simulation of two-dimensional, incompressible, Newtonian flow.

ACKNOWLEDGEMENTS

The computing facilities provided by the Computational Mechanics Laboratory (CML) of the Department of Mechanical Engineering are gratefully acknowledged. The financial support provided to the first author by the Department of Mechanical Engineering through the Carey Fellowship is also acknowledged. The authors wish to thank Dr. P. M. Gresho for many valuable suggestions which have helped in improving the quality of presentation in the paper.

REFERENCES

1. T. J. R. Hughes, W. K. Liu and A. Brooks, 'Finite element analysis of incompressible viscous flows by the penalty function formulation', *J. Comput. Phys.*, **30**, 1–60 (1979).
2. A. N. Brooks and T. J. R. Hughes, 'Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **32**, 199–259 (1982).
3. G. F. Carey and R. Krishnan, 'Continuation techniques for a penalty approximation of the Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **48**, 265–282 (1985).
4. T. J. R. Hughes, L. P. Franca and M. Balestra, 'A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal- order interpolations', *Comput. Methods Appl. Mech. Eng.*, **59**, 85–99 (1986).
5. T. J. R. Hughes and L. P. Franca, 'A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces', *Comput. Methods Appl. Mech. Eng.*, **65**, 85–96 (1987).
6. B. N. Jiang and L. A. Povinelli, 'Least squares finite element method for fluid dynamics', *Comput. Methods Appl. Mech. Eng.*, **81**, 13–37 (1990).
7. M. Becker, *The Principles and Applications of Variational Methods, Research Monograph 27*, Massachusetts Institute of Technology Press, Cambridge, MA, 1964.
8. P. P. Lynn and S. K. Arya, 'Use of the least squares criterion in the finite element formulation', *Int. J. Numer. Methods Eng.*, **6**, 75–88 (1973).
9. G. F. Carey and B. N. Jiang, 'Least-squares finite elements for first-order hyperbolic systems', *Int. J. Numer. Methods Eng.*, **26**, 81–93 (1988).
10. G. J. Fix and M. D. Gunzburger, 'On least squares approximations to indefinite problems of the mixed type', *Int. J. Numer. Methods Eng.*, **12**, 453–469 (1978).
11. P. P. Lynn, 'Least squares finite element analysis of laminar boundary layer flows', *Int. J. Numer. Methods Eng.*, **8**, 865–876 (1974).
12. J. F. Polk and P. P. Lynn, 'A least squares finite element approach to unsteady gas dynamics', *Int. J. Numer. Methods Eng.*, **12**, 3–10 (1978).
13. B. N. Jiang and C. L. Chang, 'Least-squares finite elements for the Stokes problem', *Comput. Methods Appl. Mech. Eng.*, **78**, 297–311 (1990).
14. C. A. J. Fletcher, 'A primitive variable finite element formulation for inviscid, compressible flow', *J. Comput. Phys.*, **33**, 301–312 (1979).
15. H. Nguyen and J. Reynen, 'A space–time least-square finite element scheme for advection–diffusion equations', *Comput. Methods Appl. Mech. Eng.*, **42**, 331–342 (1984).
16. I. Kececioglu and B. Rubinsky, 'A mixed-variable continuously deforming finite element method for parabolic evolution problems. Part II: The coupled problem of phase-change in porous media', *Int. J. Numer. Methods Eng.*, **28**, 2609–2634 (1989).
17. I. Babuska, O. C. Zienkiewicz, J. Gago and E. R. de A. Oliveira, *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Wiley, New York, 1986.
18. B. N. Jiang and V. Sonnad, 'Least-squares solution of incompressible Navier–Stokes equations with the *p*-version of finite elements', *NASA Tech. Mem. 105203, ICOMP-91-14*, September 1991.
19. A. Björck, 'Solution of equations in R^n (Part 1)', in *Nonlinear Least Squares*, North-Holland, Amsterdam, 1990, Chap. 6.
20. D. Winterscheidt and K. S. Surana, '*p*-Version least squares finite element formulation of Burgers' equation', *Int. J. Numer. Methods Eng.*, in press.
21. C. F. Carey and B. N. Jiang, 'Least-squares finite element method and preconditioned conjugate gradient solution', *Int. J. Numer. Methods Eng.*, **24**, 1283–1296 (1987).

22. D. Winterscheidt and K. S. Surana, '*p*-Version least squares finite element formulation for convection–diffusion problems', *Int. J. Numer. Methods Eng.*, **36,** 111–113 (1993).
23. U. Ghia, K. N. Ghia and C. T. Shin, 'High-*Re* solutions for incompressible flow using the Navier–Stokes equations and a multigrid method', *J. Comput. Phys.*, **48,** 387–411 (1982).
24. M. K. Denham and M. A. Patrick, 'Laminar flow over a downstream-facing step in a two-dimensional channel', *Trans. Inst. Chem. Eng.*, **52,** 361–367 (1974).
25. L. P. Hackman, G. D. Raithby and A. B. Strong, 'Numercial predictions of flows over backward- facing steps', *Int. J. Numer. Methods Eng.*, **4,** 711–724 (1984).