

Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation

THOMAS J. R. HUGHES, WING KAM LIU, AND ALEC BROOKS

*Division of Engineering and Applied Science, California Institute of Technology,
Pasadena, California 91125*

Received July 31, 1978; revised August 11, 1978

A review of recent work and new developments are presented for the penalty-function/finite element formulation of incompressible viscous flows. Basic features of the penalty method are described in the context of the steady and unsteady Navier-Stokes equations. Galerkin and "upwind" treatments of convection terms are discussed. Numerical results indicate the versatility and effectiveness of the new methods.

1. INTRODUCTION

The finite element method (FEM) is an established numerical technique which now enjoys widespread use in solid and structural mechanics. The main attributes of the FEM are its ease in handling very complex geometries and the ability to "naturally" incorporate differential-type boundary conditions. In addition, the method possesses a rich mathematical structure and, in many cases, it can be shown that "optimal" error estimates hold (see, for example, Strang and Fix [82]).

More recently, the FEM has been used increasingly for problems of fluid mechanics (see, for example, [10, 23, 24, 26, 71]). Nevertheless, in convection dominated situations, and in particular for the Navier-Stokes equations, the FEM has not achieved the level of success of existing finite difference methods, such as those described in [1, 42, 70]. We believe there are three main reasons for this:

First, all the experience in application of the FEM to problems of solid and structural mechanics involves symmetric operators. Until just recently, virtually no theoretical attention was paid to the fact that convection operators are *nonsymmetric*, and that, perhaps, basically new techniques would need to be developed for effectively treating them.

Secondly, most finite element researchers engaged in solving the Navier-Stokes equations seem to be infatuated with the use of exotic and/or "higher-order" elements, and "nonlinearly implicit," time-stepping algorithms. On small problems, these approaches often exhibit greater accuracy than do simpler computational

schemes. However, it is our opinion that for large problems, and in particular for 3-dimensional problems, the storage and computational effort engendered make such approaches cost-ineffective and noncompetitive with existing difference methods.

Thirdly, finite element techniques for handling kinematic constraints, such as incompressibility, have often lead to significant computational complexity, have been poorly understood and have not always performed well in practice.

In this paper, we review recent work, and present new techniques, aimed at developing efficient and accurate methods for solving the incompressible Navier–Stokes equations.

The main theme of the paper is the penalty-function formulation of the incompressibility constraint. The success of the penalty-function is easy to understand: It leads to the simplest, effective, finite element implementation of incompressibility. In describing the penalty method, we find it necessary to quote many references outside the finite element fluid mechanics literature, as only a small number of papers have been written on penalty formulations of the Navier–Stokes equations.

We also address ourselves to the other main considerations in developing methods for solving the Navier–Stokes equations; namely, effective treatment of convection terms, and efficient elements and solution algorithms.

Results obtained for the methods described herein indicate, we believe, that it is now possible to overcome the three deficiencies enumerated above, while retaining all the attributes of the FEM.

The remainder of this paper is outlined as follows: In Section 2, we describe pertinent features of the penalty/finite element formulation in the context of a simple class of problems: Stokes flow. In Section 3 we consider the steady Navier–Stokes equations. New ideas for treating convection operators are discussed in Section 4. Transient algorithms are described in Section 5 and sample results, illustrating the accuracy and versatility of the new techniques, are presented and compared with available data in Section 6. In Section 7, we summarize the present developments and make suggestions for further research.

2. STOKES FLOW

2.1. Preliminaries

Let Ω be an open set contained in \mathbb{R}^n , $n \geq 2$, with piecewise smooth boundary Γ . The closure of a set is denoted by a superposed bar (e.g., $\bar{\Omega}$ is the closure of Ω). Vector and tensor fields defined on Ω are written in boldface notation. The Cartesian components of vectors and tensors are written in the standard indicial notation. For example, x_i and u_i are the i th components of the position vector \mathbf{x} and velocity vector \mathbf{u} , respectively. We employ the summation convention on repeated indices i, j , and k *only* (e.g., $t_{ii} = t_{11} + t_{22} + \cdots + t_{nn}$). A comma is used to denote partial differentiation (e.g., $u_{i,j} = \partial u_i / \partial x_j$, the velocity gradients).

Let L_2 denote the space of Lebesgue square-integrable functions defined on Ω ,

and let H^1 denote the space of L_2 -functions whose partial derivatives are also in L_2 . The L_2 and H^1 norms are defined by (resp.)

$$\|u\|_0 = \left[\int_{\Omega} uu \, d\Omega \right]^{1/2}, \quad (2.1)$$

$$\|u\|_1 = \left[\int_{\Omega} (uu + u_{,i}u_{,i}) \, d\Omega \right]^{1/2}. \quad (2.2)$$

Let \mathbf{L}^2 (\mathbf{H}^1 , resp.) denote the space of vector fields whose components are in L_2 (H^1 , resp.). The \mathbf{L}_2 and \mathbf{H}^1 norms are defined by (resp.)

$$\|\mathbf{u}\|_0 = \left[\int_{\Omega} u_i u_i \, d\Omega \right]^{1/2}, \quad (2.3)$$

$$\|\mathbf{u}\|_1 = \left[\int_{\Omega} (u_i u_i + u_{i,j} u_{i,j}) \, d\Omega \right]^{1/2}. \quad (2.4)$$

Throughout we shall consider flows of a Newtonian fluid whose constitutive equation is given by

$$t_{ij} = -p\delta_{ij} + 2\mu u_{(i,j)}, \quad (2.5)$$

where t_{ij} denotes the Cauchy stress tensor; p is the pressure; δ_{ij} is the Kronecker delta; $\mu > 0$ is the dynamic viscosity; and $u_{(i,j)} = (u_{i,j} + u_{j,i})/2$, the symmetric part of the velocity gradients. Furthermore, we shall assume the flow to be incompressible, i.e.,

$$u_{i,i} = 0 \quad (2.6)$$

2.2. Prescribed Data

Let $\Gamma_{\tilde{y}}$ and $\Gamma_{\tilde{t}}$ be subsets of Γ which satisfy the following conditions:

$$\overline{\Gamma_{\tilde{y}} \cup \Gamma_{\tilde{t}}} = \Gamma, \quad (2.7)$$

$$\Gamma_{\tilde{y}} \cap \Gamma_{\tilde{t}} = \emptyset. \quad (2.8)$$

We assume the following functions are given:

$$\tilde{f}: \Omega \rightarrow \mathbb{R}^n \quad (\text{body force vector}), \quad (2.9)$$

$$\tilde{y}: \Gamma_{\tilde{y}} \rightarrow \mathbb{R}^n \quad (\text{velocity vector}), \quad (2.10)$$

$$\tilde{t}: \Gamma_{\tilde{t}} \rightarrow \mathbb{R}^n \quad (\text{traction vector}). \quad (2.11)$$

More general boundary conditions than those considered in (2.10) and (2.11) may be formulated. These engender no essential difficulties, however, they encumber the

presentation somewhat and thus to simplify matters we have chosen not to consider them in this exposition.

2.3. Formal Statement of the Boundary-Value Problem

Find $\mathbf{u}: \bar{\Omega} \rightarrow \mathbb{R}^n$ and $p: \bar{\Omega} \rightarrow \mathbb{R}$ such that

$$t_{ij,j} + f_i = 0 \quad \text{on } \Omega, \quad (2.12)$$

$$u_{i,i} = 0 \quad \text{on } \Omega, \quad (2.13)$$

$$u_i = g_i \quad \text{on } \Gamma_{\tilde{g}}, \quad (2.14)$$

$$t_{ij}n_j = h_i \quad \text{on } \Gamma_{\tilde{h}}, \quad (2.15)$$

where t_{ij} is given by (2.5).

Remarks. 1. If $\Gamma_{\tilde{h}} = \emptyset$, then we require a consistency condition emanating from (2.13) and (2.14), viz.,

$$\begin{aligned} 0 &= \int_{\Omega} u_{i,i} d\Omega \\ &= \int_{\Gamma} u_i n_i d\Gamma \\ &= \int_{\Gamma} g_i n_i d\Gamma. \end{aligned} \quad (2.16)$$

In this case, p is determined up to an arbitrary constant.

2. It is well known that under suitable hypotheses the boundary-value problem of Stokes flow is well posed (see, e.g., Temam [85]).

2.4. Penalty-Function Formulation

In the penalty-function formulation of Stokes flow, the constitutive equation is replaced by

$$t_{ij}^{(\lambda)} = -p^{(\lambda)}\delta_{ij} + 2\mu u_{(i,j)}^{(\lambda)} \quad (2.17)$$

in which

$$p^{(\lambda)} = -\lambda u_{k,k}^{(\lambda)}, \quad (2.18)$$

where $\lambda > 0$ is a parameter. Furthermore, the incompressibility condition is dropped. The boundary-value problem of the penalty function formulation is stated as follows:

Find $\mathbf{u}^{(\lambda)}: \bar{\Omega} \rightarrow \mathbb{R}^n$ such that

$$t_{ij}^{(\lambda)} + f_i = 0 \quad \text{on } \Omega, \quad (2.19)$$

$$u_i^{(\lambda)} = g_i \quad \text{on } \Gamma_{\tilde{g}}, \quad (2.20)$$

$$t_{ij}^{(\lambda)} = h_i \quad \text{on } \Gamma_{\tilde{h}}, \quad (2.21)$$

where $t_{ij}^{(\lambda)}$ is given by (2.17).

Remarks. 1. The convergence of the penalty-function solution to the Stokes flow solution has been proved by Temam [85]. A sketch of the main steps of the proof is given as follows (consult [85] for further details): Equation (2.12) is subtracted from (2.19) to obtain

$$\mu(u_i^{(\lambda)} - u_i)_{,jj} + (p^{(\lambda)} - p)_{,i} = 0. \quad (2.22a)$$

This result is multiplied by $u_i^{(\lambda)} - u_i$ and integrated over Ω . Integration by parts, use of the divergence theorem and boundary conditions, along with a standard inequality (i.e., $2ab \leq a^2 + b^2$), leads to

$$\mu \| \mathbf{u}^{(\lambda)} - \mathbf{u} \|_1^2 + \frac{1}{2\lambda} \| p^{(\lambda)} \|^2 \leq \frac{1}{2\lambda} \| p \|^2. \quad (2.22b)$$

It follows from (2.22b) that, as $\lambda \rightarrow \infty$, $\mathbf{u}^{(\lambda)} \rightarrow \mathbf{u}$ in \mathbf{H}^1 , and consequently (2.22a) may be used to argue that $p^{(\lambda)} \rightarrow p$ in L_2 . If λ is selected sufficiently large then $\mathbf{u}^{(\lambda)}$ and $p^{(\lambda)}$ differ negligibly from \mathbf{u} and p , respectively. The advantages of the penalty-function formulation are that the additional unknown p is eliminated and so is the necessity of satisfying the incompressibility condition. In numerical practice this leads to considerable simplification.

2. The equations of Stokes flow are identical to the equations of classical, isotropic, *incompressible* elasticity in which \mathbf{u} is interpreted as the displacement vector. Likewise, the penalty equations are identical to classical, isotropic, *compressible* elasticity, in which λ and μ are interpreted as the Lamé parameters. Thus the penalty approach in elasticity amounts to approximating an incompressible medium by a slightly compressible one.¹

The physical interpretation in fluids is different. Here it is the mass conservation equation which is being approximated and the associated errors amount to net fluid loss or gain (see, e.g., Hughes *et al.* [51]).

3. In the sequel we deal only with the penalty-function formulation. Thus it is notationally convenient to omit the λ superscripts in all subsequent developments.

2.5. Weak Formulation

Let

$$\mathbf{V} = \{ \mathbf{w} \in \mathbf{H}^1 \mid \mathbf{w} = \mathbf{0} \text{ on } \Gamma_{\mathcal{Z}} \}. \quad (2.23)$$

(Eq. 2.23 reads: “ \mathbf{V} is the space of all \mathbf{H}^1 -functions which vanish on $\Gamma_{\mathcal{Z}}$.”) \mathbf{V} is often called the space of *weighting functions*, or *variations*.

¹ The use of penalty methods in solid mechanics is now widespread; see, for example, [29, 37–41, 43, 46–48, 50, 53, 54, 58–60, 67, 72, 73, 84, 93, 94].

Let \tilde{g} denote a given \mathbf{H}^1 -extension of g ; that is, $\tilde{g} \in \mathbf{H}^1$ and

$$\tilde{g} = g \quad \text{on } \Gamma_{\tilde{g}}. \quad (2.24)$$

The weak form of the boundary-value problem is stated as follows:

Find $\mathbf{u} = \mathbf{w} + \tilde{g}$, $\mathbf{w} \in \mathbf{V}$, such that for all $\bar{\mathbf{w}} \in \mathbf{V}$

$$\begin{aligned} \int_{\Omega} (\lambda w_{j,j} \bar{w}_{i,i} + 2\mu w_{(i,j)} \bar{w}_{(i,j)}) d\Omega &= \int_{\Omega} f_i \bar{w}_i d\Omega + \int_{\Gamma_{\tilde{g}}} h_i \bar{w}_i d\Gamma \\ &- \int_{\Omega} (\lambda \tilde{g}_{j,j} \bar{w}_{i,i} + 2\mu \tilde{g}_{(i,j)} \bar{w}_{(i,j)}) d\Omega. \end{aligned} \quad (2.25)$$

Remark. Under appropriate smoothness hypotheses, it is a simple exercise to verify that the solution of the weak formulation is identical to the solution of (2.19)–(2.21).

2.6. Galerkin Formulation

Let \mathbf{V}^h be a finite-dimensional subspace of \mathbf{V} ; \mathbf{V}^h is to be thought of as a member of a collection of subspaces $\mathcal{V} = \{\mathbf{V}^h\}$, dense in \mathbf{V} and parameterized by h , the “mesh parameter.”

Let \tilde{g}^h denote an approximation of \tilde{g} which converges to \tilde{g} as $h \rightarrow 0$.

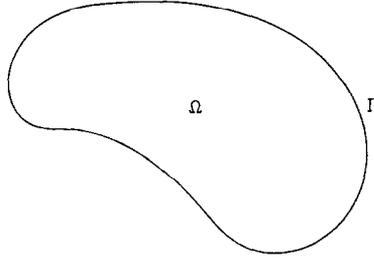
The Galerkin counterpart of the weak formulation is given as follows:

Find $\mathbf{u}^h = \mathbf{w}^h + \tilde{g}^h$, $\mathbf{w}^h \in \mathbf{V}^h$, such that for all $\bar{\mathbf{w}}^h \in \mathbf{V}^h$

$$\begin{aligned} \int_{\Omega} (\lambda w_{j,j}^h \bar{w}_{i,i}^h + 2\mu w_{(i,j)}^h \bar{w}_{(i,j)}^h) d\Omega &= \int_{\Omega} f_i \bar{w}_i^h d\Omega + \int_{\Gamma_{\tilde{g}^h}} h_i \bar{w}_i^h d\Omega \\ &- \int_{\Omega} (\lambda \tilde{g}_{j,j}^h \bar{w}_{i,i}^h + 2\mu \tilde{g}_{(i,j)}^h \bar{w}_{(i,j)}^h) d\Omega. \end{aligned} \quad (2.26)$$

2.7. Matrix Problem

The domain Ω is discretized into nonoverlapping subregions called “elements.” The e th element domain is denoted Ω^e and its boundary is denoted Γ^e (see Fig. 1). (In general, an “ e ” superscript indicates a quantity referred to the e th element.) Associated with the discretization is a set of n_{np} “nodal points.” The position vector of the A th node, $A = 1, 2, \dots, n_{\text{np}}$, is denoted \mathbf{x}_A . Let $\mathcal{N} = \{1, 2, \dots, n_{\text{np}}\}$, the set of nodal indices, and let $\mathcal{N}_{\tilde{g}} = \{A \in \mathcal{N} \mid \mathbf{x}_A \in \Gamma_{\tilde{g}}\}$, the subset of nodal indices corresponding to boundary nodes at which velocity is prescribed. The “shape function” associated with node A is denoted N_A ; the shape functions satisfy the relation



DISCRETIZATION:

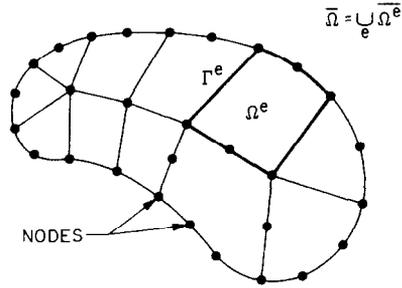


FIG. 1. Finite element spatial discretization.

$N_A(\mathbf{x}_B) = \delta_{AB}$. The solution of the Galerkin problem may be expressed in terms of the shape functions as follows:

$$w_i^h = \sum_{A \in \mathcal{N}^r} N_A v_{iA}, \quad (2.27)$$

$$\tilde{g}_i^h = \sum_{A \in \mathcal{N}^r} N_A g_{iA}, \quad (2.28)$$

where v_{iA} is the i th velocity component at node A and

$$g_{iA} = g_i(\mathbf{x}_A). \quad (2.29)$$

As can be seen by (2.29), the approximation of \tilde{g}_i assumed in (2.28) is one of nodal interpolation via the shape functions.

Substitution of (2.27) and (2.28) into (2.26) gives rise to the matrix problem

$$\mathbf{C}\mathbf{v} = \mathbf{F}, \quad (2.30)$$

where

$$\mathbf{C} = [C_{PQ}], \quad (2.31)$$

$$\mathbf{v} = \{v_Q\}, \quad (2.32)$$

$$\mathbf{F} = \{F_P\}. \quad (2.33)$$

The indices P, Q above take on the values $1, 2, \dots, n_{eq}$, where n_{eq} refers to the number of equations in the “global” system. The equation numbers are stored in a “destination array,” denoted ID , and defined as follows:

$$P = ID(i, A) \tag{2.34}$$

↗ equation number ↖ “global” node number
 ↑
 degree-of-freedom number ($1 \leq i \leq n$)

Nodal velocity components which are prescribed (i.e., “ \bar{g} -type” boundary conditions) are assigned equation number zero and are not included in the global ordering.

The matrix C is symmetric, positive definite, and possesses a band-profile structure (see Fig. 2). Very efficient solution of (2.30) is facilitated by so-called “active-column” equation solvers [6, 17, 64, 65, 83, 89, 90] in which zeros outside the profile are neither stored nor processed.

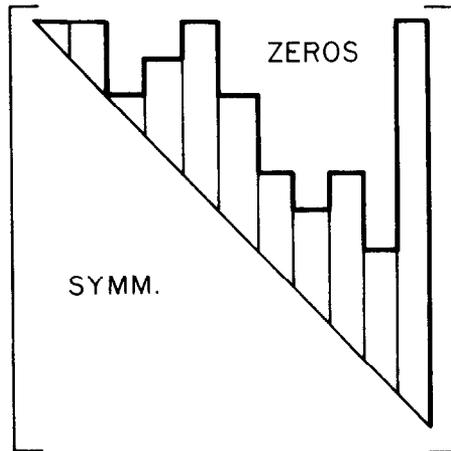


FIG. 2. Symmetric matrix possessing a band-profile structure.

In a finite element computer program, it is most convenient to form the arrays C and F in an element-by-element fashion. In this regard we may write

$$C = \mathbf{A} \left(\mathbf{c}^e \right), \quad F = \mathbf{A} \left(\mathbf{f}^e \right), \tag{2.35}$$

$\begin{matrix} n_{e1} \\ e=1 \end{matrix}$
 $\begin{matrix} n_{e1} \\ e=1 \end{matrix}$

where \mathbf{A} denotes an “assembly operator” whose function is to add elemental contributions (namely \mathbf{c}^e and \mathbf{f}^e) to the appropriate locations of C and F , and n_{e1} is the number of elements. It can be shown that the element arrays may be defined as follows:

$$\mathbf{c}^e = [c_{pq}^e], \quad \mathbf{f}^e = \{f_v^e\}, \quad 1 \leq p, q \leq n_{ee}, \quad (2.36)$$

$$c_{pq}^e = \mathbf{e}_i^T \mathbf{c}_{ab}^e \mathbf{e}_j, \quad p = n(a-1) + i, \quad q = n(b-1) + j, \quad 1 \leq a, b \leq n_{en}, \quad (2.37)$$

$$\mathbf{c}_{ab}^e = \int_{\Omega^e} (\mathbf{B}_a^e)^T \mathbf{D}_\lambda \mathbf{B}_b^e d\Omega + \int_{\Omega^e} (\mathbf{B}_a^e)^T \mathbf{D}_\mu \mathbf{B}_b^e d\Omega, \quad (2.38)$$

$$f_v^e = \int_{\Omega^e} N_a^e f_i d\Omega + \int_{\Gamma_{\tilde{h}}^e} N_a^e h_i d\Gamma - \sum_{q=1}^{n_{ee}} c_{pq}^e g_q^e, \quad \Gamma_{\tilde{h}}^e = \Gamma_{\tilde{h}} \cap \Gamma^e, \quad (2.39)$$

$$g_q^e = g_j(\mathbf{x}_b^e) \quad \text{if } \mathbf{x}_b^e \in \Gamma_{\tilde{g}}, \\ = 0 \quad \text{if } \mathbf{x}_b^e \notin \Gamma_{\tilde{g}}. \quad (2.40)$$

In the above, a superscript T denotes transpose; n_{en} is the number of element nodes; $n_{ee} = n_{en} \cdot n$ is the number of element equations; a and b are element ("local") node numbers; p and q are element ("local") equation numbers; N_a^e is the shape function associated with node a of the e th element; and \mathbf{e}_i is the i th canonical basis vector of \mathbb{R}^n , e.g.,

$$(n = 2)$$

$$\mathbf{e}_1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}, \quad \mathbf{e}_2 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}. \quad (2.41)$$

$$(n = 3)$$

$$\mathbf{e}_1 = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}, \quad \mathbf{e}_2 = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}, \quad \mathbf{e}_3 = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}. \quad (2.42)$$

The arrays \mathbf{B}_a^e , \mathbf{D}_λ , and \mathbf{D}_μ for the cases of most practical interest are given as follows:

$$(n = 2: \text{rectilinear}; d\Omega = dx_1 dx_2.)$$

$$\mathbf{B}_a^e = \begin{bmatrix} N_{a,1}^e & 0 \\ 0 & N_{a,2}^e \\ N_{a,2}^e & N_{a,1}^e \end{bmatrix}, \quad (2.43)^2$$

$$\mathbf{D}_\lambda = \lambda \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (2.44)$$

$$\mathbf{D}_\mu = \mu \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.45)$$

² The reader is reminded that commas denote partial differentiation (e.g., $N_{a,1}^e = \partial N_a^e / \partial x_1$).

($n = 2$: axisymmetric; $d\Omega = 2\pi x_1 dx_1 dx_2$, where $x_1 = r$ and $x_2 = z$ are cylindrical coordinates)

$$\mathbf{B}_a^e = \begin{bmatrix} N_{a,1}^e & 0 \\ 0 & N_{a,2}^e \\ N_{a,2}^e & N_{a,1}^e \\ N_{a,1}^e/x_1 & 0 \end{bmatrix}, \quad (2.46)$$

$$\mathbf{D}_\lambda = \lambda \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}, \quad (2.47)$$

$$\mathbf{D}_\mu = \mu \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad (2.48)$$

For the axisymmetric case, one need replace all Cartesian coordinate representations, such as (2.17) and (2.18), by their counterparts in cylindrical coordinates (see, e.g., Batchelor [5]).

($n = 3$: rectilinear; $d\Omega = dx_1 dx_2 dx_3$.)

$$\mathbf{B}_a^e = \begin{bmatrix} N_{a,1}^e & 0 & 0 \\ 0 & N_{a,2}^e & 0 \\ 0 & 0 & N_{a,3}^e \\ 0 & N_{a,3}^e & N_{a,2}^e \\ N_{a,3}^e & 0 & N_{a,1}^e \\ N_{a,2}^e & N_{a,1}^e & 0 \end{bmatrix}, \quad (2.49)$$

$$\mathbf{D}_\lambda = \lambda \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.50)$$

$$\mathbf{D}_\mu = \mu \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.51)$$

2.8. Numerical Integration

The integrations appearing in (2.38) and (2.39) are carried out with the aid of numerical integration formulas. In most finite element work it is only necessary to use a “sufficiently accurate” integration rule (see [91] for details). However, in the penalty method, it is crucial to “underintegrate” the λ -term in (2.38). The reasons for this have been discussed at length in [60] so we only provide a brief summary here.

By virtue of (2.38), we may write

$$\mathbf{C} = \mathbf{C}_\lambda + \mathbf{C}_\mu, \quad (2.52)$$

where \mathbf{C}_λ and \mathbf{C}_μ are proportional to λ and μ , respectively. It is apparent that if \mathbf{C}_λ is nonsingular, then $\mathbf{v} \rightarrow \mathbf{0}$ as $\lambda \rightarrow \infty$. Consequently, \mathbf{C}_λ *must* be singular so that the penalty method works [22]. It turns out that virtually all commonly used conforming³ finite elements result in nonsingular \mathbf{C}_λ when exact integration is performed. To singularize \mathbf{C}_λ one need only lower the order of the λ -term integration rule. The rank of the matrix \mathbf{C} is retained by employing a sufficiently high-order integration on the μ -term in (2.38). (This follows from a Korn inequality; see [20]). An example is illustrative of these ideas:

EXAMPLE. Consider the 4-node, bilinear, isoparametric quadrilateral element. For this element it is standard to employ 2×2 Gauss–Legendre integration. However, this rule renders \mathbf{C}_λ nonsingular and thus fails in our applications. On the other hand, a 1-point Gauss–Legendre rule on the λ -term results in \mathbf{C}_λ being singular. The 2×2 rule is retained on the μ -term to insure nonsingularity of \mathbf{C} . This element, suggested and employed in [51, 52], has proven to be the simplest effective one for use in the penalty function formulation.

The following facts may be proved for this element in the rectilinear case [43, 60]:

- (i) As $\lambda \rightarrow \infty$, $u_{i,i}^h \rightarrow 0$ at the origin of the element natural coordinate system.
- (ii) $(\int_{\Omega^e} u_{i,i}^h d\Omega) / (\int_{\Omega^e} d\Omega) \rightarrow 0$ as $\lambda \rightarrow \infty$ (“incompressibility in the mean”) and furthermore the pressure at the origin, computed via (2.18), is the mean pressure for the element.

Remarks. 1. When a quadrature rule of lower order than the “standard” one is employed, this is called *reduced integration*. If all terms⁴ employ the same reduced integration, this is called *uniform reduced integration*; if reduced integration is used on some terms while standard integration is used on others (as in the above example), this is called *selective reduced integration*. Selective integration procedures are subject to an “invariance criterion” [48]. In the present applications, this condition requires that both \mathbf{C}_λ and \mathbf{C}_μ generate quadratic forms which are invariant with respect to change of reference frame. It is a simple exercise to show that this is the case.

³ A conforming, or compatible, finite element, in the present context is one which leads to the members of V^h being continuous functions.

⁴ In the present context, we mean the λ and μ terms in (2.38).

2. Studies have been undertaken to determine the most effective elements and quadrature schemes for use with the penalty method. So far, these efforts have been largely empirical as no rigorous general theory yet exists. However, an heuristic theory [60] has been remarkably accurate in predicting the behavior of elements and integration schemes. Briefly, the theory suggests that the most effective elements in applications of the type considered here are the so-called “Lagrange” isoparametric elements with appropriate selective integration schemes. These elements, for the 2-dimensional case, are schematically illustrated in Fig. 3. Triangular elements and “serendipity” quadrilateral elements are predicted to exhibit inferior behavior, which has been confirmed numerically [59].

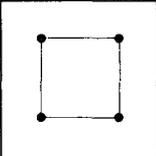
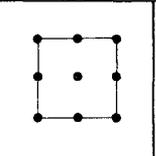
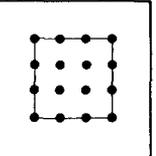
			
shape functions	bilinear	biquadratic	bicubic
λ -term	1 point	2×2	3×3
μ -term	2×2	3×3	4×4

FIG. 3. Selective Gauss–Legendre integration rules for 2-dimensional isoparametric Lagrange elements.

3. To each element/integration scheme used in the penalty-function formulation corresponds a “mixed finite element” method based upon the velocity–pressure formulation of Stokes flow, namely (2.12)–(2.15). General “equivalence theorems” of this type, which are also applicable to nonlinear situations, have been proven in [60]. The identification of penalty and mixed methods facilitates an “interpretation” of the pressure field in the penalty case. It turns out that in all cases the pressure field is to be viewed as discontinuous across element boundaries, the velocity field being continuous. The appropriate points at which to sample the pressure are determined to be the integration points of the rule used to integrate the λ -term in (2.38).

4. When the 4-node, selectively integrated, bilinear element, discussed previously, is used in axisymmetric analysis, the property of “incompressibility in the mean” is lost. Nevertheless, the element is still convergent.

The following alternative formulation produces a 4-node quadrilateral element

which possesses the mean incompressibility property [66]. In the definition of \mathbf{c}_{ab}^e (Eq. 2.38), replace the λ -term by

$$(\tilde{\mathbf{B}}_a^e)^T \mathbf{D}_\lambda \tilde{\mathbf{B}}_b^e / \left(\int_{\Omega^e} d\Omega \right), \quad (2.53)$$

where

$$\tilde{\mathbf{B}}_a^e = \int_{\Omega^e} \mathbf{B}_a^e d\Omega. \quad (2.54)$$

The integral above should be calculated exactly; analysis of the integrand reveals that the 2×2 Gauss–Legendre rule is sufficient in this regard. The 2×2 rule should be maintained for the μ -term in (2.38). The element mean pressure may be determined by computing the mean value of

$$p = -\lambda(u_{1,1}^h + u_{2,2}^h + u_1^h/x_1). \quad (2.55)$$

Again, the 2×2 rule may be used effectively.

5. In programming the element arrays every effort is made to minimize computations. In this regard we note that there is considerable sparseness and redundancy in the “ $\mathbf{B}^T \mathbf{D} \mathbf{B}$ ” products of (2.38), all of which may be taken advantage of in coding. For useful ideas along these lines, and good-quality software, see [83].

2.9. Selection of Penalty Parameter λ

In the penalty method the question naturally arises: “How does one select the value of λ ?” Clearly, λ must be large enough so that the compressibility and pressure errors are negligible, yet not so large that numerical ill conditioning ensues. Dimensional analysis reveals that, for Stokes flow, λ should be picked according to the relation⁵

$$\lambda = c\mu, \quad (2.56)$$

where c is a constant which depends only on the computer word length and, in particular, is independent of the mesh parameter h . Numerical studies reveal that for floating-point word lengths of 60 to 64 bits, an appropriate choice of c is 10^7 . (This choice seems to be problem independent.)

2.10. Pressure Smoothing

As was mentioned previously, the pressure field is to be viewed as discontinuous from element to element. In fact, all velocity derivatives for isoparametric elements are in general discontinuous on element boundaries. Thus, for plotting purposes, it is desirable to employ a smoothing procedure, which redefines the field under consideration in terms of the shape functions N_A .

⁵ This criterion needs to be generalized to be applicable to the full Navier–Stokes equations; see Section 3.2.

With specific reference to the pressure, there is at least one other reason for employing a smoothing procedure. It has been shown that, in certain situations, discontinuous-pressure, mixed-method finite elements exhibit a rank deficiency in the assembled pressure equations [68]. By the equivalence results of [60], "problems" are also to be expected with the pressure field of the penalty function formulation. These problems typically manifest themselves as pressure oscillations. For example, if 4-node, quadrilateral elements are employed in a square mesh, with an even number of square elements in each direction, subjected to all velocity boundary conditions, then a "checkerboard" pressure oscillation is produced. (This phenomenon apparently also afflicts the MAC finite difference method [70].) Despite the pressure oscillations, the velocity field remains good. From the standpoint of error analysis these developments lead us to make the following conjectures: (i) Optimal L_2 -convergence rates in velocity hold for the selectively-integrated Lagrange elements;⁶ (ii) H^1 -convergence is provable only if explicit account is taken of a filtering procedure which removes the "checkerboard mode," or analogous pathologies.

Fortunately, smoothing procedures of a least-squares type [36] seem to perform the necessary filtering as a by-product. A comprehensive study of such techniques has been performed by Lee *et al.* [56]. The methods we have been using for constant-pressure elements, which involve slight modifications of schemes proposed in [56], are described below.

Let the discontinuous pressure field be written as

$$p = \sum_{e=1}^{n_{e1}} p^e \psi^e, \quad (2.57)$$

where p^e is the element mean pressure and ψ^e is the e th element "characteristic function," i.e.,

$$\begin{aligned} \psi^e(\mathbf{x}) &= 1 & \text{if } \mathbf{x} \in \Omega^e \\ &= 0 & \text{if } \mathbf{x} \notin \Omega^e. \end{aligned} \quad (2.58)$$

The smoothed pressure is written

$$\tilde{p} = \sum_{A=1}^{n_{np}} \tilde{p}_A N_A. \quad (2.59)$$

The standard least-squares procedure gives rise to the following matrix problem:

$$\mathbf{Y}\tilde{\mathbf{p}} = \mathbf{P}, \quad (2.60)$$

where

$$\mathbf{Y} = [Y_{AB}], \quad (2.61)$$

$$\tilde{\mathbf{p}} = \{\tilde{p}_B\}, \quad (2.62)$$

⁶ Mercier [61] has achieved an L_2 -convergence proof of the 4-node element in a related situation by way of finite difference methods.

and

$$\mathbf{P} = \{P_A\}. \quad (2.63)$$

The indices A, B take on the values $1, 2, \dots, n_{np}$. The construction of \mathbf{Y} and \mathbf{P} is performed in the usual element-by-element fashion, viz.,⁷

$$\mathbf{Y} = \mathbf{A} \begin{matrix} n_{e1} \\ (y^e), \\ e=1 \end{matrix} \quad \mathbf{P} = \mathbf{A} \begin{matrix} n_{e1} \\ (p^e) \\ e=1 \end{matrix} \quad (2.64)$$

in which

$$\mathbf{y}^e = [y_{ab}^e], \quad \mathbf{p}^e = \{p_a^e\}, \quad 1 \leq a, b \leq n_{en}, \quad (2.65)$$

$$y_{ab}^e = \int_{\Omega^e} N_a^e N_b^e d\Omega, \quad p_a^e = p^e \int_{\Omega^e} N_a^e d\Omega. \quad (2.66)$$

As it stands, the matrix \mathbf{Y} is symmetric, positive-definite, and possesses a band-profile structure. Additional simplification may be engendered by replacing \mathbf{Y} by an associated diagonal matrix.⁸ This is done by approximating the first equation of (2.66); the procedures we use are summarized as follows:

($n = 2$; rectilinear case). The 2×2 product, trapezoidal, integration rule may be used to diagonalize \mathbf{y}^e , i.e.,

$$y_{ab}^e = \delta_{ab} j^e(\xi_a, \eta_a), \quad (2.67)$$

where

$$j^e = \det \begin{bmatrix} x_{1,\xi}^e & x_{1,\eta}^e \\ x_{2,\xi}^e & x_{2,\eta}^e \end{bmatrix} \quad (\text{Jacobian determinant}), \quad (2.68)$$

$$\mathbf{x}^e = \sum_{a=1}^{n_{en}} N_a^e \mathbf{x}_a^e, \quad (2.69)$$

and ξ_a and η_a are the coordinates of node "a" in the element "natural" coordinate system. Applying the same integration scheme to the second equation of (2.66) yields

$$p_a^e = p^e j^e(\xi_a, \eta_a). \quad (2.70)$$

Further simplification may be achieved by approximating $j^e(\xi_a, \eta_a)$ in (2.67) and (2.70) by $j^e(0, 0)$. (In the case when Ω^e is parallelogrammic, j^e is constant and no loss of accuracy is incurred by this procedure.)

The 3-dimensional case is the straightforward generalization of the above, so we omit the details.

⁷ The 'assembly operators' in (2.64) are not the same as those used previously in (2.35). Here, no boundary conditions are taken account of and there is only one degree of freedom per node.

⁸ Lee *et al.* [56] have also found that higher accuracy is attained when \mathbf{Y} is diagonal!

($n = 2$; axisymmetric case.) If we attempt to apply the above procedure in the axisymmetric case we encounter a difficulty due to the factor x_1 (i.e., “ r ”) in the integrands. Along the x_2 -axis, $x_1 = 0$, hence the trapezoidal integration technique produces a zero diagonal entry in Y . In this case we employ a “row-sum” diagonalization technique in which

$$y_{ab}^e = \delta_{ab} \int_{\Omega^e} N_a^e d\Omega \quad (\text{no sum on “}a\text{”}). \quad (2.71)$$

The above integration, which also suffices for the second part of (2.66), may be performed by either 1-point or 2×2 Gauss–Legendre integration—the latter scheme being exact.

The procedures described above render the formation, storage, and solution of the matrix equation (2.60) very efficient. The results produced tend to be very good at interior nodes, but leave something to be desired at boundary nodes. To improve upon the results a “correction” at each boundary node is performed. We describe the procedure used for 4-node elements with the aid of an example.

Consider the mesh illustrated in Fig. 4a. The nodes are segregated into four groups. The boundary node corrections are carried out in the following steps in order:

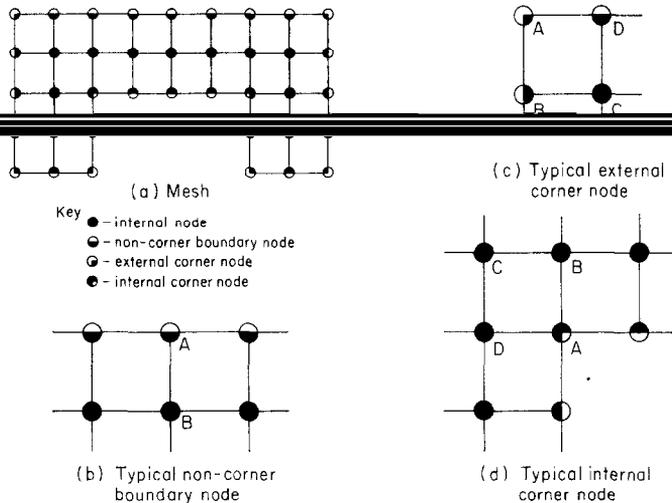


FIG. 4. Example mesh for 4-node element, pressure-smoothing algorithm.

Step 1: noncorner, boundary nodes. A typical case of a noncorner, boundary node is depicted in Fig. 4b. It may be observed that the unaltered value of \tilde{p}_A is actually a higher-order approximation to the pressure at the midpoint of the line joining nodes A and B (see Barlow [4]). Thus we redefine the \tilde{p}_A by way of linear extrapolation, i.e.,

$$\tilde{p}_A \leftarrow 2\tilde{p}_A - \tilde{p}_B \quad (2.72)$$

Step 2: external corner nodes. A typical situation is depicted in Fig. 4c. The unaltered value of \hat{p}_A is precisely the constant pressure p^e , because the above procedures reduce to “do-nothing” calculations at external corners. (If “checkerboarding” was occurring in the p^e 's, the value of \hat{p}_A would be grossly in error.) In this case we employ linear extrapolation through nodes B , C , and D , i.e.,

$$\hat{p}_A \leftarrow (\tilde{L}_B \hat{p}_B + \tilde{L}_C \hat{p}_C + \tilde{L}_D \hat{p}_D) / L, \quad (2.73)$$

where

$$\tilde{L}_B = L_B + (x_{2C} - x_{2D}) x_{1A} + (x_{1D} - x_{1C}) x_{2A}, \quad (2.74)$$

$$\tilde{L}_C = L_C + (x_{2D} - x_{2B}) x_{1A} + (x_{1B} - x_{1D}) x_{2A}, \quad (2.75)$$

$$\tilde{L}_D = L_D + (x_{2B} - x_{2C}) x_{1A} + (x_{1C} - x_{1B}) x_{2A}, \quad (2.76)$$

$$L_B = x_{1C} x_{2D} - x_{1D} x_{2C}, \quad (2.77)$$

$$L_C = x_{1D} x_{2B} - x_{1B} x_{2D}, \quad (2.78)$$

$$L_D = x_{1B} x_{2C} - x_{1C} x_{2B}, \quad (2.79)$$

$$L = L_B + L_C + L_D. \quad (2.80)$$

Step 3: internal corner nodes. A typical configuration is shown in Fig. 4d. In this case the unaltered \hat{p}_A is essentially a weighted average of the p^e 's associated with the three elements which have node A in common. As in Step 2, if “checkerboarding” had occurred, the unaltered \hat{p}_A would be significantly in error. Again we use linear extrapolation; namely, (2.73)–(2.80).

Slight generalizations of the above procedures may be used for smoothing pressures in higher-order elements. Vorticity data, for example, may be handled in the same fashion.

3. STEADY NAVIER-STOKES EQUATIONS

3.1. Penalty-Function Formulation

The boundary-value problem of the penalty-function formulation is stated as follows:

Find \mathbf{u} : $\bar{\Omega} \rightarrow \mathbb{R}^n$ such that

$$\rho u_j u_{i,j} = t_{ij,j} + f_i \quad \text{on } \Omega, \quad (3.1)$$

$$u_i = g_i \quad \text{on } \Gamma_g, \quad (3.2)$$

$$t_{ij} n_j = h_i \quad \text{on } \Gamma_h, \quad (3.3)$$

where

$$t_{ij} = -p \delta_{ij} + 2\mu u_{(i,j)}, \quad (3.4)$$

$$p = -\lambda u_{k,k}, \quad (3.5)$$

and $\rho > 0$ is the density—a given function of \mathbf{x} .

Remark. We have experimented with several modifications of the convective momentum (i.e., left-hand side of (3.1)). In particular, we have added the “correction term” proposed by Temam [85], and also put the convective momentum in “conservation form,” as is often favored by workers in finite differences [75]. On sample problems, all other things being equal, no discernible difference could be noted when compared with the simple expression in (3.1).

3.2. Selection of the Penalty Parameter

The criterion for selecting the value of the penalty parameter, λ , in the present case is different from that for Stokes flow (cf. (2.56)). This is due to the presence of the convective momentum term which generally dominates the viscous term. The criterion for the Navier–Stokes equations is

$$\lambda = c \max\{\mu, \mu Re\}, \quad (3.6)$$

where c is the same constant as in (2.56) and Re is a Reynolds number. To make this definition precise, we must say how we intend to compute the Reynolds number. If

$$Re = UL/\nu, \quad (3.7)$$

where $\nu = \mu/\rho$ (“kinematic viscosity”), and U and L are “characteristic” velocity and length, respectively, this entails specification of U and L . We usually take U to be the maximum expected velocity in the flow and L to be a major dimension (e.g., the “diameter” of Ω). Generally U may be estimated to sufficient accuracy from boundary data and simple physical considerations.

In singular situations, such as Hamel flow [5], we employ classical definitions which result in finite values of Re , despite the maximum flow velocity being infinite.

We wish to emphasize that it is not necessary to be very fussy about the selection of λ as it may vary over several orders of magnitude with essentially insignificant effect on results.

3.3. Weak and Galerkin Formulations

A weak form of the boundary-value problem is stated as follows:

Find $\mathbf{u} = \mathbf{w} + \tilde{\mathcal{G}}$, $\mathbf{w} \in \mathbf{V}$, such that for all $\bar{\mathbf{w}} \in \mathbf{V}$

$$\begin{aligned} & \int_{\Omega} (\rho \mathbf{u}_j \mathbf{u}_{i,j} \bar{\mathbf{w}}_i + \lambda w_{j,i} \bar{w}_{i,i} + 2\mu w_{(i,j)} \bar{w}_{(i,j)}) d\Omega \\ & = \int_{\Omega} f_i \bar{w}_i d\Omega + \int_{\Gamma_{\tilde{\mathcal{G}}}} h_i \bar{w}_i d\Gamma - \int_{\Omega} (\lambda \tilde{\mathcal{G}}_{j,i} \bar{w}_{i,i} + 2\mu \tilde{\mathcal{G}}_{(i,j)} \bar{w}_{(i,j)}) d\Omega. \end{aligned} \quad (3.8)$$

Under appropriate hypotheses, solutions of (3.1) through (3.5) and (3.8) can be shown to be equivalent.

The Galerkin formulation may be obtained from the above by replacing \mathbf{u} , \mathbf{w} , $\tilde{\mathcal{G}}$, and \mathbf{V} by \mathbf{u}^h , \mathbf{w}^h , $\tilde{\mathcal{G}}^h$, and \mathbf{V}^h , respectively.

3.4. Matrix Problem

Employing (2.27) and (2.28) in (3.8) gives rise to the matrix problem

$$\mathbf{C}\mathbf{v} + \mathbf{N}(\mathbf{v}) = \mathbf{F}, \quad (3.9)$$

where \mathbf{C} and \mathbf{F} are the same as in Section 2.7, and $\mathbf{N}: \mathbb{R}^{n_{ea}} \rightarrow \mathbb{R}^{n_{ea}}$ is a nonlinear mapping defined as follows:

$$\mathbf{N}(\mathbf{v}) = \mathbf{A} \begin{pmatrix} n_{e1} \\ \vdots \\ n_p^e(\mathbf{v}^e) \\ \vdots \end{pmatrix}, \quad (3.10)$$

$$\mathbf{n}^e(\mathbf{v}^e) = \{n_p^e(\mathbf{v}^e)\}, \quad (3.11)$$

$$\mathbf{v}^e = \{v_p^e\}, \quad v_p^e = v_{ia}^e, \quad p = n(a-1) + i, \quad (3.12)$$

$$\begin{aligned} v_{ia}^e &= u_i^h(\mathbf{x}_a^e) = g_i(\mathbf{x}_a^e) & \text{if } \mathbf{x}_a^e \in \Gamma_{\mathcal{G}} \\ &= w_i^h(\mathbf{x}_a^e) & \text{if } \mathbf{x}_a^e \notin \Gamma_{\mathcal{G}}, \end{aligned} \quad (3.13)$$

$$u_i^h(\mathbf{x}) = \sum_{a=1}^{n_{en}} N_a^e(\mathbf{x}) v_{ia}^e \quad \mathbf{x} \in \Omega^e, \quad (3.14)$$

$$u_{i,j}^h(\mathbf{x}) = \sum_{a=1}^{n_{en}} N_{a,j}^e(\mathbf{x}) v_{ia}^e \quad \mathbf{x} \in \Omega^e, \quad (3.15)$$

$$n_p^e(\mathbf{v}^e) = \int_{\Omega^e} \rho N_a^e u_j^h u_{i,j}^h d\Omega. \quad (3.16)$$

3.5. Solution Algorithm

Since (3.9) is nonlinear, some form of iteration need be employed. We have found it effective to use an incremental Newton–Raphson scheme in which density is used as a “load parameter.” Within each load level we iterate until convergence is achieved; the converged solution is used as the initial guess for the next load level. We initialize the first load level with the solution of the Stokes problem. It is well known that, under appropriate hypotheses, the Newton–Raphson scheme exhibits second-order convergence of the iterates.

In a typical load level, the iterates are computed from

$$\mathbf{C}^*(\bar{\mathbf{v}}) \Delta \mathbf{v} = \Delta \mathbf{F}, \quad (3.17)$$

$$\bar{\mathbf{v}} \leftarrow \bar{\mathbf{v}} + \Delta \mathbf{v}, \quad (3.18)$$

where $\bar{\mathbf{v}}$ is the latest approximation to the solution,

$$\Delta \mathbf{F} = \mathbf{F} - \mathbf{C}\bar{\mathbf{v}} - \mathbf{N}(\bar{\mathbf{v}}) \quad (\text{"out-of-balance force"}), \quad (3.19)$$

$$\mathbf{C}^*(\bar{\mathbf{v}}) = \mathbf{C} + D\mathbf{N}(\bar{\mathbf{v}}), \quad (3.20)$$

$$D\mathbf{N}(\bar{\mathbf{v}}) = \mathbf{A} \sum_{e=1}^{n_{e1}} (D\mathbf{n}^e(\bar{\mathbf{v}}^e)) \quad (\text{"tangent convection matrix"}), \quad (3.21)$$

$$(D\mathbf{n}^e(\bar{\mathbf{v}}^e))_{pq} = \int_{\Omega^e} \rho N_a^e (\bar{u}_{i,j}^h N_b^e + \delta_{ij} \bar{u}_k^h N_{b,k}^e) d\Omega. \quad (3.22)$$

The definition of the convection matrix, $D\mathbf{N}$, is obtained with the aid of the directional ("variational") derivative, i.e.,

$$D\mathbf{N}(\bar{\mathbf{v}}) \Delta \mathbf{v} = \left. \left\{ \frac{d}{d\epsilon} \mathbf{N}(\bar{\mathbf{v}} + \epsilon \Delta \mathbf{v}) \right\} \right|_{\epsilon=0}. \quad (3.23)$$

The convergence condition we require is that the ratio of $\Delta \mathbf{F}$ to its original value for the load level becomes less than a preassigned tolerance, say 10^{-5} .

We may observe from (3.20) through (3.22) that \mathbf{C}^* is a nonsymmetric matrix. However, it possesses a symmetric profile (see Fig. 5). We employ a Crout elimination algorithm [83] which fully exploits this structure in that zeros outside the profile are neither stored nor operated upon.

The preceding algorithm, which is felt to be the most effective for this class of problems, has been employed with success by Hughes *et al.* [52].

It may be noted that the definitions of \mathbf{n}^e and $D\mathbf{n}^e$ are identical for both the axisymmetric and rectilinear cases.

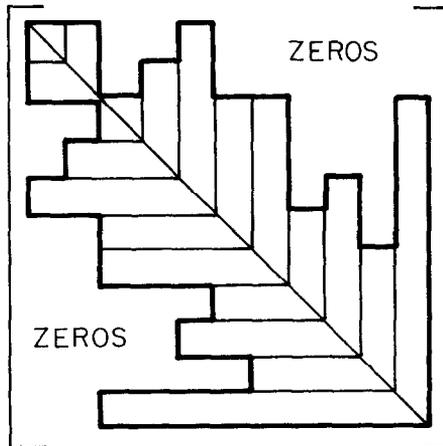


FIG. 5. Nonsymmetric matrix possessing a symmetric band-profile structure.

3.6. Numerical Integration

To complete the description of the steady Navier–Stokes algorithm, it is necessary to define the quadrature treatment of the element convection arrays (namely \mathbf{n}^e and $D\mathbf{n}^e$). The most straightforward procedure for quadrilaterals and “bricks” is to use iterated Gauss–Legendre rules. For the 4-node element, the 2×2 rule has been employed by Hughes *et al.* [52]; for the 9-node Lagrange element, the 3×3 rule has been used by Bercovier and Engelman [8]. It has been demonstrated in these studies that many problems may be adequately solved with simple Gauss–Legendre integration of the convection arrays and relatively crude meshes. Some sample results for a much studied problem are contained in the following example.

EXAMPLE—Driven Cavity Flow. A problem description is shown in Fig. 6. The penalty parameter is set in accord with (3.6). Note that the boundary conditions are discontinuous at the upper corners. Two meshes of 4-node elements, employing different approximations of the boundary conditions, are shown in Fig. 7. As can be seen from the midplane velocity profiles, Fig. 8, the different treatments of the boundary conditions can result in significant quantitative differences, especially as the Reynolds number is increased. Several investigators have studied this problem and quite a variety of approximations of the corner discontinuities have been employed (see, e.g., [8, 11, 25, 52, 86]). By virtue of the sensitivity of the results to the treatment of the boundary conditions, care must be taken in interpretation.

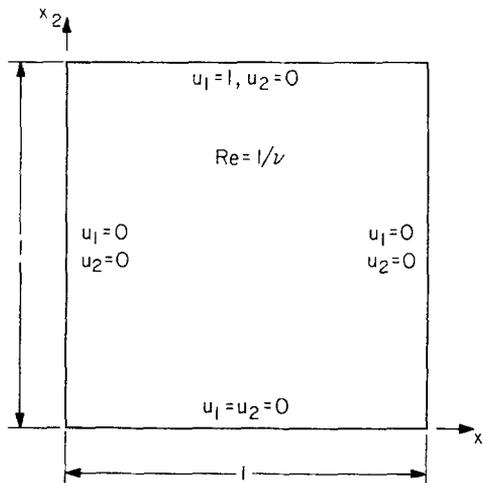


FIG. 6. Driven cavity flow: problem description.

For example, consider a 10×10 mesh of square, 9-node elements. This mesh would have the same number of degrees of freedom as the 20×20 mesh of 4-node elements and it may seem appropriate to compare results. However, one should keep in mind that setting the nodal boundary conditions in identical fashion actually

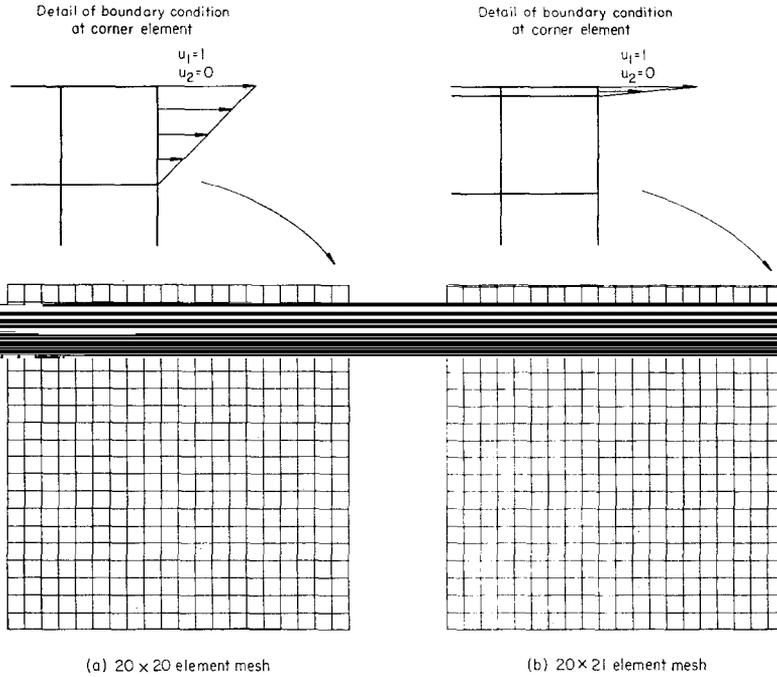


FIG. 7. Driven cavity flow: Finite element meshes employing different approximations of discontinuous corner boundary conditions.

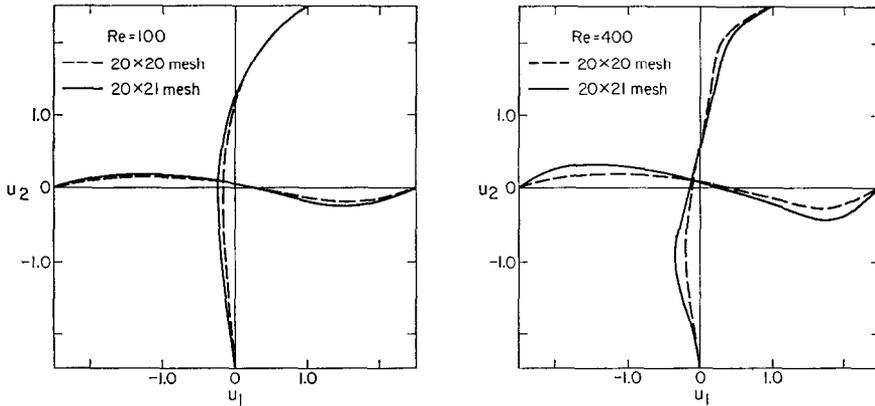


FIG. 8. Driven cavity flow: midplane velocity profiles.

implies different representations along the vertical edges of the corner elements (cf. Figs. 7 and 9). Discrepancies noted between midplane velocity profiles (see [35]) may be attributed to the different boundary conditions, and are not indicative of the respective merits of the elements.

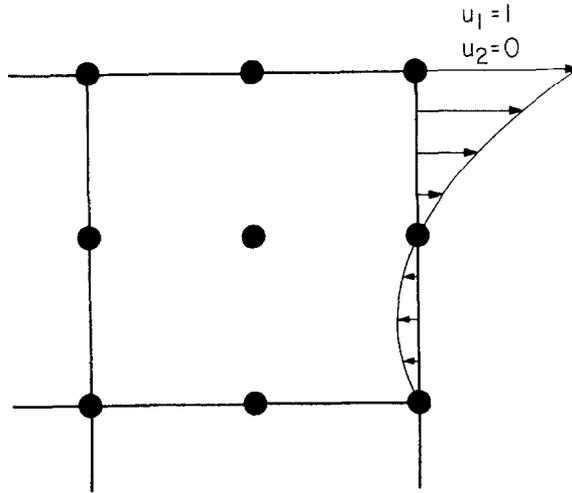


FIG. 9. Driven cavity flow: Approximation of discontinuous corner boundary condition for a 9-node element.

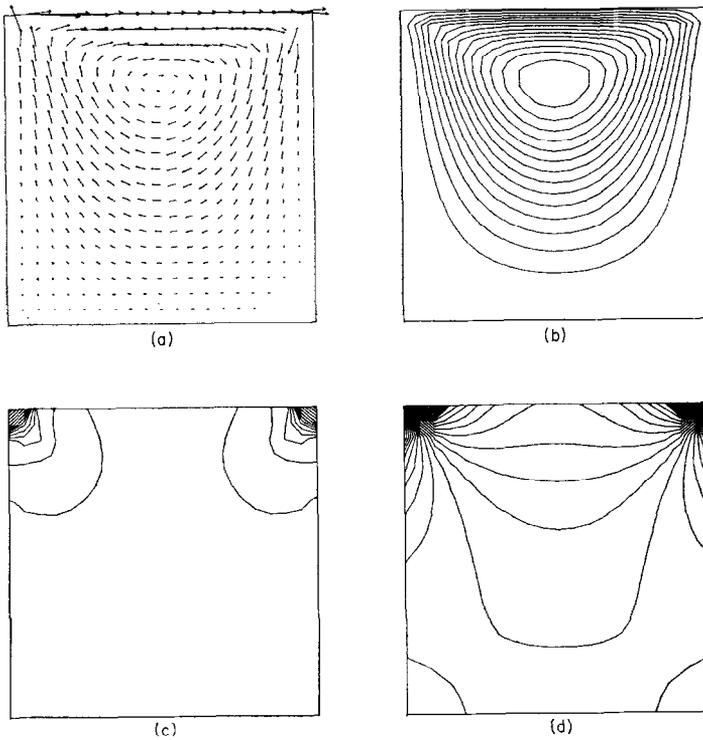


FIG. 10. Driven cavity flow ($Re = 0$; 20×21 mesh): (a) Velocity vectors; (b) streamlines; (c) pressure contours; and (d) vorticity contours.

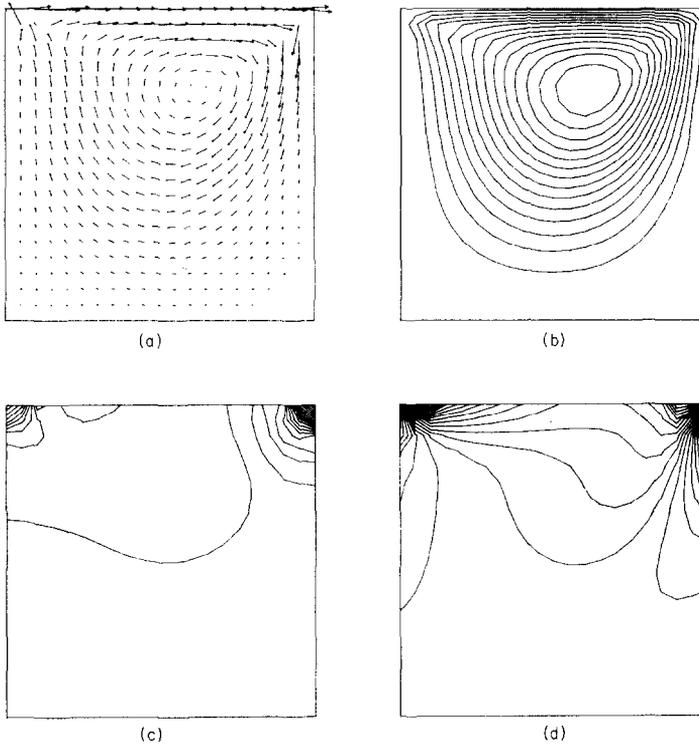


FIG. 11. Driven cavity flow ($Re = 100$; 20×21 mesh): (a) Velocity vectors; (b) streamlines; (c) pressure contours; and (d) vorticity contours.

In passing, we note that most studies of cavity flows, even those at high Reynolds numbers, have employed uniform meshes. As the steep gradients in the problem are all adjacent to the boundaries, and are especially severe in the corners, it would seem propitious in future studies to grade the mesh accordingly. As is well known, this may be easily done with finite element methods.

Further results for the 20×21 mesh are shown in Figs. 10 through 12. Continuous pressures and vorticities are calculated by way of the algorithms described in Section 2.10.

Despite some successes at surprisingly high Reynolds numbers [52], it is our experience that Gauss-Legendre treatment of the convection arrays is not generally effective at high Reynolds numbers. A typical situation which exhibits the shortcomings of such schemes is encountered upstream of obstacles; see Section 6.4. The nonphysical “wiggles” which appear can only be removed by mesh refinement. At high Reynolds numbers, the degree of refinement necessary to preclude such pathologies is felt to be excessive, undermining the practical utility of the methods. For these reasons, alternative treatments of the convection terms have been investigated.

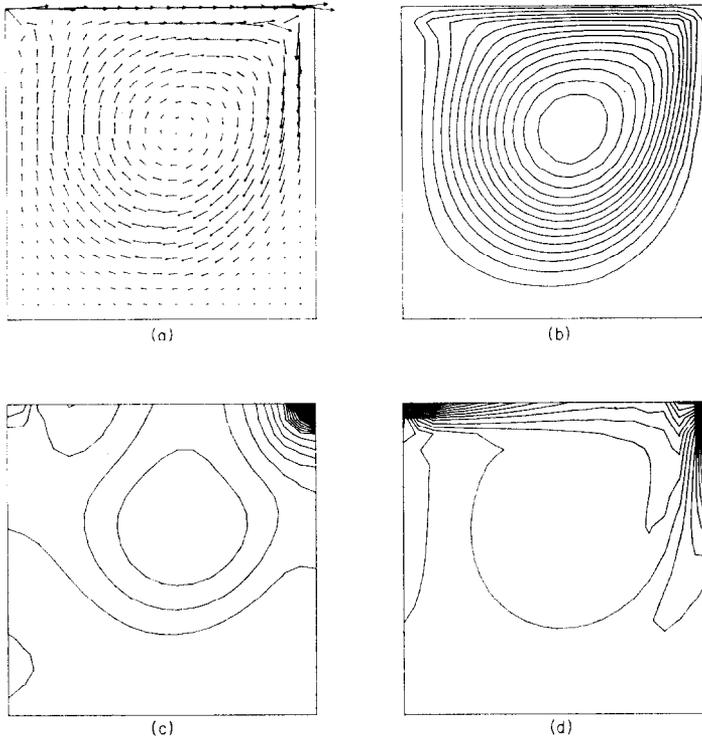


FIG. 12. Driven cavity flow ($Re = 400$; 20×21 mesh): (a) Velocity vectors; (b) streamlines; (c) pressure contours; and (d) vorticity contours.

4. "UPWIND" FINITE ELEMENTS

Galerkin finite element methods give rise to central difference approximations of first-order differential (e.g., convection) operators. The inability of difference approximations of this type to handle convection dominated phenomena has been known for some time in the finite difference literature [75]. To circumvent these problems, "upwind differences" have been proposed which preclude oscillations, but are formally less accurate than central differences.

Recently, schemes which attempt to achieve similar ends have been proposed within the context of weighted residual/finite element methods (see [2, 7, 12, 21, 32–34, 44, 45, 62, 63, 87]). Although these schemes differ considerably in concept and implementation, they have collectively become known as "upwind finite element methods." At this time it is still too early to evaluate the relative merits of the various schemes which have been proposed. We may note, however, that rigorous foundations have already been elucidated for several of the new schemes (see [2, 45]).

In the following sections we describe the upwind finite element techniques that

we have been using. These methods are based upon numerical integration rules and are at once very easy to implement, efficient, and effective.

We begin with a linear model problem—the advection–diffusion equation.

4.1. Advection–Diffusion Equation

The boundary-value problem for the steady advection–diffusion equation is stated as follows:

Find $\phi: \bar{\Omega} \rightarrow \mathbb{R}$ such that

$$u_i \phi_{,i} = (k \phi_{,i})_{,i} + f \quad \text{on } \Omega \quad (4.1)$$

$$\phi = g \quad \text{on } \Gamma_g, \quad (4.2)$$

$$k \phi_{,i} n_i = h \quad \text{on } \Gamma_h, \quad (4.3)$$

where u_i is the given flow velocity, $k > 0$ is the diffusivity, and f , g , and h are the prescribed data.

A weak form of the problem is given by the following:

Find $\phi = w + \tilde{g}$, $w \in V$, such that for all $\bar{w} \in V$

$$\int_{\Omega} (u_i w_{,i} \bar{w} + k w_{,i} \bar{w}_{,i}) d\Omega = \int_{\Omega} f \bar{w} d\Omega + \int_{\Gamma_h} h \bar{w} d\Gamma - \int_{\Omega} (u_i \tilde{g}_{,i} \bar{w} + k \tilde{g}_{,i} \bar{w}_{,i}) d\Omega. \quad (4.4)$$

Under suitable hypotheses, solutions of (4.1) through (4.3) and (4.4) may be shown to be equivalent.

The Galerkin formulation corresponding to (4.4) leads to the following matrix problem:

$$\mathbf{K} \mathbf{d} = \mathbf{F}, \quad (4.5)$$

where

$$\mathbf{K} = [K_{PQ}], \quad 1 \leq P, Q \leq n_{eq}, \quad (4.6)$$

$$\mathbf{d} = \{d_Q\}, \quad (4.7)$$

$$\mathbf{F} = \{F_P\}. \quad (4.8)$$

The global arrays may be constructed from the element arrays in the usual fashion; viz.,

$$\mathbf{K} = \mathbf{A} \begin{matrix} n_{e1} \\ \mathbf{k}^e \\ e=1 \end{matrix}, \quad \mathbf{F} = \mathbf{A} \begin{matrix} n_{e1} \\ \mathbf{f}^e \\ e=1 \end{matrix}, \quad (4.9)$$

$$\mathbf{k}^e = [k_{ab}^e], \quad \mathbf{f}^e = \{f_a^e\}, \quad 1 \leq a, b \leq n_{en}, \quad (4.10)$$

$$k_{ab}^e = \int_{\Omega^e} (N_a^e u_i N_{b,i}^e + N_{a,i}^e k N_{b,i}^e) d\Omega, \quad (4.11)$$

$$f_a^e = \int_{\Omega^e} N_a^e f d\Omega + \int_{\Gamma^e} N_a^e h d\Gamma - \sum_{b=1}^{n_{en}} k_{ab}^e g_b^e, \quad \Gamma^e = \Gamma_h \cap \Gamma^e, \quad (4.12)$$

$$\begin{aligned} g_b^e &= g(\mathbf{x}_b^e) & \text{if } \mathbf{x}_b^e \in \Gamma_\rho, \\ &= 0 & \text{if } \mathbf{x}_b^e \notin \Gamma_\rho. \end{aligned} \quad (4.13)$$

The matrix \mathbf{K} is nonsymmetric due to the advection term, but possesses a symmetric profile (cf. Fig. 5). In advection dominated situations, the definition of \mathbf{k}^e given above (i.e., Eq. (4.11)) may lead to significant oscillations unless the mesh is sufficiently refined. To circumvent this occurrence for the basic isoparametric elements (i.e., the linear, bilinear, and trilinear elements in 1, 2, and 3 dimensions, respectively) we adopt the following alternative definition of \mathbf{k}^e :

$$k_{ab}^e = N_a^e(\xi^e) u_i(\mathbf{0}^e) N_{b,i}^e(\xi^e) j(\mathbf{0}^e) W + \int_{\Omega^e} N_{a,i}^e k N_{b,i}^e d\Omega, \quad (4.14)$$

where ξ^e is some point in the e th element domain, $\mathbf{0}^e$ is the origin of isoparametric coordinates in the e th element, and W is a weight factor which equals 2, 4, or 8 when the problem is 1-, 2-, or 3-dimensional, respectively. Any sufficiently accurate integration rule may be used to evaluate the diffusion terms (e.g., the standard Gauss–Legendre rules). The location of the point ξ^e determines the degree of “upwinding” in each element.

As an example of how the point ξ^e is picked, we consider a typical 4-node quadrilateral and, for notational convenience, drop the e superscripts. A representative geometry is illustrated in Fig. 13. Let $\xi = (\xi, \eta)^T$, let \mathbf{e}_ξ and \mathbf{e}_η denote the unit basis vectors at $\mathbf{0}$ in the ξ and η directions, respectively, and let h_ξ and h_η be the lengths of the element in the ξ and η directions, respectively, as depicted in Fig. 13. Then ξ and η are given by the following relations:

$$\begin{aligned} u_\xi &= \mathbf{e}_\xi^T \mathbf{u}(\mathbf{0}), & u_\eta &= \mathbf{e}_\eta^T \mathbf{u}(\mathbf{0}), \\ \alpha_\xi &= u_\xi h_\xi / (2k(\mathbf{0})), & \alpha_\eta &= u_\eta h_\eta / (2k(\mathbf{0})), & \text{("element Peclet numbers")}. \\ \xi &= (\coth \alpha_\xi) - 1/\alpha_\xi, & \eta &= (\coth \alpha_\eta) - 1/\alpha_\eta, \end{aligned} \quad (4.15)$$

The derivation which leads to (4.15) is contained in Hughes [44], in which it is shown that for the 1-dimensional, constant coefficient case, the above procedure results in exact nodal values for all mesh lengths (i.e., “superconvergence”). For this

be shown to be second-order accurate (see Roscoe [76–78] for related ideas).

A simplified scheme which avoids the calculation of hyperbolic cotangents, precludes spurious oscillations and maintains second-order accuracy is given by

$$\begin{aligned} \xi &= -1 - 1/\alpha_\xi, & \alpha_\xi &< -1, \\ &= 0, & -1 &\leq \alpha_\xi \leq 1, \\ &= 1 - 1/\alpha_\xi, & 1 &< \alpha_\xi, \end{aligned} \quad (4.16)$$

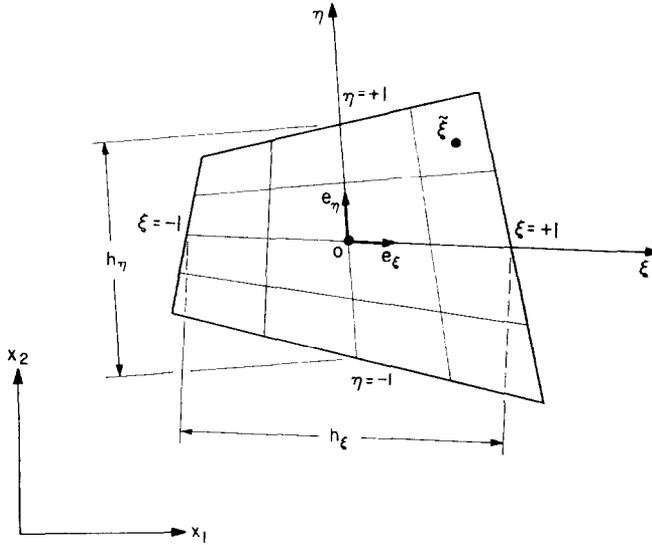


FIG. 13. Typical 4-node quadrilateral finite element geometry.

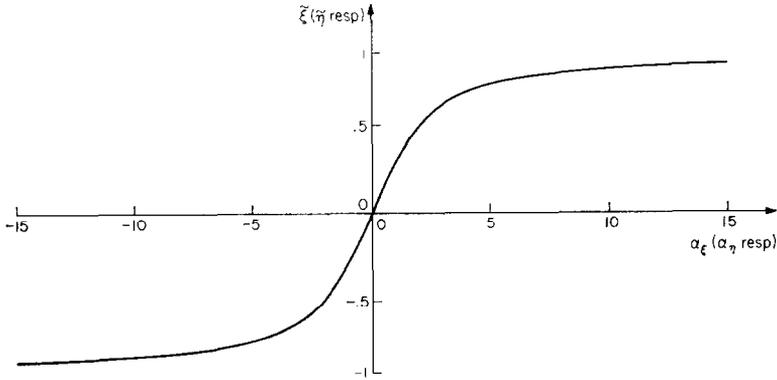


FIG. 14. Integration rule for optimal upwind scheme.

and

$$\begin{aligned} \tilde{\eta} &= -1 - 1/\alpha_\eta, & \alpha_\eta < -1, \\ &= 0, & -1 \leq \alpha_\eta \leq 1, \\ &= 1 - 1/\alpha_\eta, & 1 < \alpha_\eta. \end{aligned} \tag{4.17}$$

For reasons given in [44], we refer to (4.16) and (4.17) as the *critical upwind scheme* (see Fig. 15).

The *full upwind scheme*, defined by

$$\begin{aligned} \tilde{\xi} &= -1, & \alpha_\xi < 0, \\ &= 0, & \alpha_\xi = 0, \\ &= 1, & \alpha_\xi > 0, \end{aligned} \tag{4.18}$$

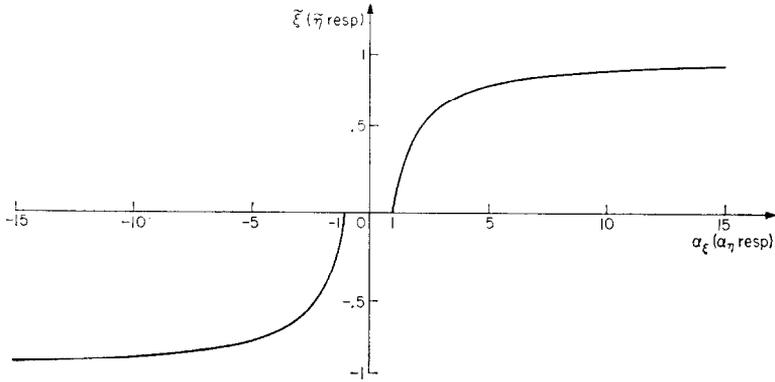


FIG. 15. Integration rule for critical upwind scheme.

and

$$\begin{aligned} \tilde{\eta} &= -1, & \alpha_{\tilde{\eta}} < 0, \\ &= 0, & \alpha_{\tilde{\eta}} = 0, \\ &= 1, & \alpha_{\tilde{\eta}} > 0, \end{aligned} \tag{4.19}$$

which results in the standard upwind difference approximation to the advection term (see [75, p. 65]) is not recommended in general as accuracy drops to first order.

Remarks. 1. Triangles may also be formed using the technique of degeneration.

2. The generalization of the above procedures to three dimensions is obvious and need not be considered further.

EXAMPLE—Entry Flow in a Channel. The effectiveness of the present scheme may be seen from the problem shown in Fig. 16. The velocity field was obtained by

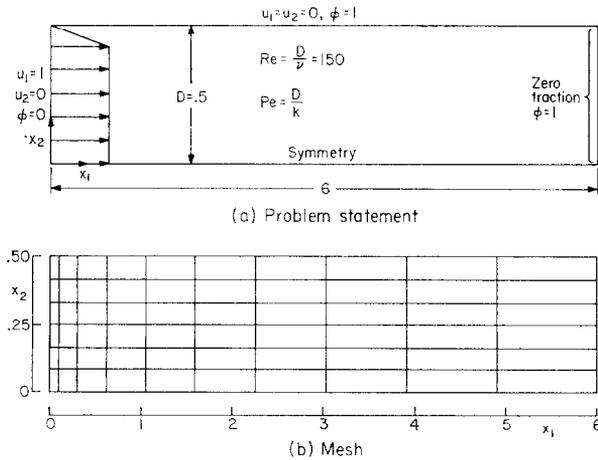


FIG. 16. Entry flow in a channel: problem description and finite element mesh.

employing the steady Navier–Stokes algorithm described in the previous sections. (2×2 Gaussian integration was used on the convection terms.) A comparison of four different integration treatments of the advection terms is contained in Fig. 17 for a Peclet number of 150. As can be clearly seen, the upwind schemes are superior to the Gauss schemes. For this high a Peclet number, there is little difference between the optimal and full upwind schemes.

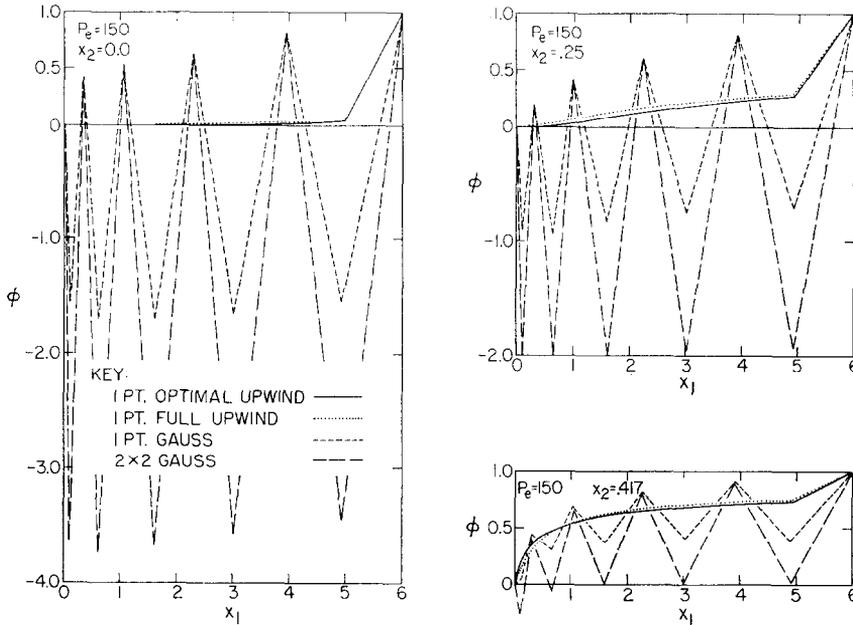


FIG. 17. Entry flow in a channel: comparison of results for four different integration treatments of the advection term.

The downstream boundary layer is entirely contained within the last column of elements and thus the present mesh is incapable of resolving it. The virtue of the upwind schemes is that the rest of the solution is not vitiated by the failure to capture the boundary layer, as is the case for the Gauss integration treatment of the advection term. This is an important feature when one considers use of multigrid schemes [9] to resurrect details such as boundary layers.

At a much higher Peclet number (i.e., 1.5×10^7) the optimal and full upwind schemes give identical results whereas the Gauss schemes plot off scale (see Fig. 18).

Some controversy has arisen as to the appropriateness of this problem as a test for evaluating the relative merits of upwind and Galerkin schemes (see [27, 92]). It is our opinion that the results do show the superiority of the upwind approach here. The usual (and valid) criticism that upwinding is only first-order accurate does not apply to our optimal and critical schemes.

Remark. The present formulation of upwind finite elements is extremely simple

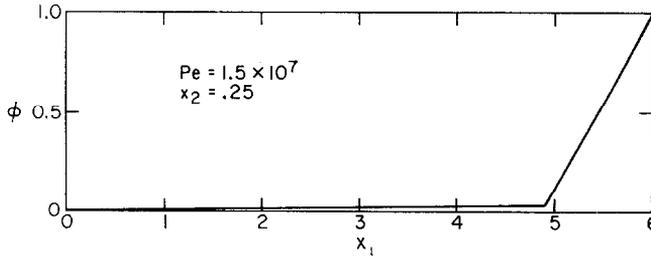


FIG. 18. Entry flow in a channel: results for optimal and full upwind schemes at high Peclet number.

to implement into existing computer codes employing Galerkin finite element methods. The computational cost is also reduced compared with the schemes of [34] as the present technique is, in essence, a one-point evaluation of the advection term, in which all other terms of the Galerkin formulation remain unaltered. The unsymmetric location of the evaluation point controls the degree of upwinding, so all the inherent advantages indicated in [34] are realized.

4.2. Upwind Integration Rules

An important point to observe from the preceding developments is that upwind elements were constructed from the usual basis and weighting functions by adjustment of the integration rule. This suggests that the theory of upwind elements may be reducible to special integration rules. It is of interest to construct higher-order rules, involving more points, analogous to the Gauss–Legendre rules. From results obtained in [2] we infer that the pertinent integration rules may be determined from the relation

$$\left(\int_{-1}^{+1} e^{\alpha\xi} g(\xi) d\xi \right) / \sigma = \sum_{l=1}^{n_{\text{int}}} W_l g(\xi_l), \quad (4.20)$$

where

$$\sigma = \frac{1}{2} \int_{-1}^{+1} e^{\alpha\xi} d\xi \quad (4.21)$$

in which n_{int} is the number of integration points, ξ_l is the l th point and W_l is the associated weight. The points and weights are determined by satisfying (4.20) for $g(\xi) = 1, \xi, \xi^2, \dots$, etc. The 1-point rule, which is exact for $g(\xi) = 1$ and ξ , is given by

$$\xi_1 = (\coth \alpha) - 1/\alpha, \quad W_1 = 2. \quad (4.22)$$

This is the rule which is used in the optimal upwind scheme.

Closed form expressions for integration points and weights of higher-order rules are extremely complicated. Graphs of numerically obtained results, as well as asymptotic approximations for small and large α , are contained in [2].

In general, n_{int} points are required to satisfy (4.20) for $g(\xi) = 1, \xi, \dots, \xi^{2n_{\text{int}}-1}$. When $\alpha = 0$, the rules reduce to the Gauss–Legendre rules. For all rules, as $\alpha \rightarrow \pm \infty$,

$\xi_l \rightarrow \pm 1$, $l = 1, 2, \dots, n_{\text{int}}$; that is, in the infinite Peclet number limit, the general n_{int} -point rule degenerates to the 1-point rule. Product rules for 2- and 3-dimensional domains may be found by standard techniques. Let us assume the product rule under consideration consists of n_{int} integration points, ξ_l , and weights W_l . Then the advection-term contribution to \mathbf{k}^e may be written as

$$\sum_{l=1}^{n_{\text{int}}} N_a(\xi_l) u_i(\mathbf{0}) N_{b,i}(\xi_l) j(\mathbf{0}) W_l, \quad (4.23)$$

4.3. Upwind Treatments of the Navier–Stokes Equations

Although application of the preceding techniques to the Navier–Stokes equations is straightforward, we have found certain deficiencies in numerical tests. Our experiences indicate that the upwind technique does eliminate certain spurious oscillations associated with the pure Galerkin formulation. For example, oscillations upstream of obstacles (see, e.g., [80]) are removed. However, in some problems for which the Galerkin method worked well (e.g., the Hamel problem; see [52]) a degradation of accuracy occurred with the upwind schemes described previously. We have determined, largely by numerical experiments, that these difficulties are encountered when the fluid experiences significant “squeezing,” such as occurs in highly singular situations (e.g., the Hamel problem). Under these circumstances Gauss quadrature works quite well, whereas upwind quadrature appears to be superior when the fluid is “stretching.” To appropriately account for these different effects, we have designed a modified upwind quadrature rule, based on the linearized Burgers’ equation

$$u\phi_{,x} + u'\phi = \nu\phi_{,xx}, \quad (4.24)$$

where we assume ν , u ; and u' are given constants. A weak form is established, similar to (4.4), and linear shape functions are assumed. We allow either a 1-point unsymmetric integration, or 2-point symmetric integration (i.e., $\xi_2 = -\xi_1 = \xi$), of the terms emanating from the left-hand side of (4.24). Assuming equal-sized elements, the difference equation at an interior node takes the form

$$(-\alpha - 1 - \omega) d_{p-1} + (\beta + 2(1 + \omega)) d_p + (\alpha - 1 - \omega) d_{p+1} = 0, \quad (4.25)$$

where

$$\begin{aligned} \omega &= \alpha\xi - \beta(1 - \xi^2)/4, & \text{1-point rule,} \\ &= -\beta(1 - \xi^2)/4, & \text{2-point rule,} \end{aligned} \quad (4.26)$$

in which

$$\alpha = uh/(2\nu) \quad (\text{“element Reynolds number”})$$

and

$$\beta = u'h^2/\nu. \quad (4.28)$$

β is a nondimensional measure of the u' -term. The term ω is a nondimensional “artificial viscosity.”

The integration rule is determined as follows:

If $\beta \geq 0$, we ignore its effect and revert back to the 1-point upwind scheme being employed (i.e., either optimal or critical; see Section 4.1).

If $\beta < 0$, ω/α is set equal to the evaluation point of the 1-point upwind scheme, and the new ξ is determined from (4.26). If $|\beta| \leq 4|\alpha|$, the first of (4.26) is employed, whereas if $|\beta| > 4|\alpha|$, the second is employed.

A schematic of the modified upwind integration technique is shown in Fig. 19 for positive α . As can be seen, negative β causes the integration point to move towards the center of the element. As the integration point passes through the center, it bifurcates and subsequently the two points move symmetrically towards the nodes. In the limit $\beta \rightarrow -\infty$, the integration scheme becomes the trapezoidal rule (i.e., nodal quadrature).

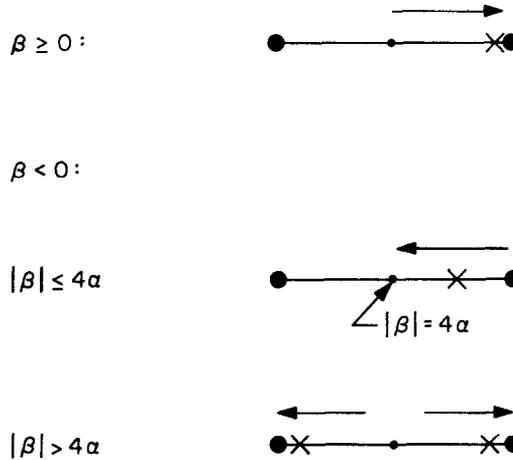


FIG. 19. Movement of integration points for modified upwind scheme in the case of positive α .

A detailed generalization of the scheme for a 2-dimensional element is presented in Table I. The 3-dimensional case proceeds analogously.

Although the logic is somewhat long, it is rarely the case that there is more than one integration point and thus the evaluation of the convective terms can be performed efficiently. The counterparts of (3.16) and (3.22) for the modified upwind scheme are, respectively,

$$n_p^e(\mathbf{v}^e) = \rho j(\mathbf{0}^e) W \sum_{l=1}^{n_{\text{int}}} N_a^e(\xi_l) u_i^h(\mathbf{0}^e) u_{i,j}^h(\xi_l^e) \quad (4.29)$$

and

$$(D\mathbf{n}^e(\bar{\mathbf{v}}^e))_{pq} = \rho j(\mathbf{0}^e) W \sum_{l=1}^{n_{\text{int}}} \{N_a^e(\xi_l) [\bar{u}_{i,j}^h(\xi_l) N_b^e(\mathbf{0}^e) + \delta_{ij} \bar{u}_k^h(\mathbf{0}^e) N_{b,k}(\xi_l)]\}. \quad (4.30)$$

TABLE I—Continued

-
- 8a. $\tilde{\eta}_1 = (1 - \delta_\eta)^{1/2}$; $\tilde{\eta}_2 = -\tilde{\eta}_1$; $n_{int}^\eta = 2$; go to 10.
- 8b. $\tilde{\eta}_1 = \tilde{\eta}_2 = 0$; $n_{int}^\eta = 1$; go to 10.
- 8c. $\tilde{\eta}_{1,2} = (\gamma_\eta + [\gamma_\eta^2 - 4(\delta_\eta - 1)]/2)/2$; $n_{int}^\eta = 1$.
 If $\tilde{\eta} < 0$, go to 8d,
 ≥ 0 , go to 8e.
- 8d. If $\tilde{\eta}_1 < \tilde{\eta}$, or $\tilde{\eta}_1 > 0$, $\tilde{\eta}_1 = \tilde{\eta}$.
 If $\tilde{\eta}_2 < \tilde{\eta}$, or $\tilde{\eta}_2 > 0$, $\tilde{\eta}_2 = \tilde{\eta}$.
 $\tilde{\eta} = \max(\tilde{\eta}_1, \tilde{\eta}_2)$; go to 9.
- 8e. If $\tilde{\eta}_1 > \tilde{\eta}$, or $\tilde{\eta}_1 < 0$, $\tilde{\eta}_1 = \tilde{\eta}$.
 If $\tilde{\eta}_2 > \tilde{\eta}$, or $\tilde{\eta}_2 < 0$, $\tilde{\eta}_2 = \tilde{\eta}$.
 $\tilde{\eta} = \min(\tilde{\eta}_1, \tilde{\eta}_2)$
9. $\tilde{\eta}_1 = \tilde{\eta}_2 = \tilde{\eta}$; $n_{int}^\eta = 1$.
10. $n_{int} = n_{int}^\xi \cdot n_{int}^\eta$
 If $n_{int} = 1$, $W = 4$,
 $= 2$, $W = 2$,
 $= 4$, $W = 1$.
- If $n_{int} = 4$, then $\tilde{\xi}_3 = \tilde{\xi}_1$, $\tilde{\xi}_4 = \tilde{\xi}_2$, $\tilde{\eta}_3 = \tilde{\eta}_2$ and $\tilde{\eta}_4 = \tilde{\eta}_1$.^b
-

^b For a 2-dimensional, incompressible flow, if the ξ and η axes are perpendicular, one of u'_ξ and u'_η must be ≥ 0 , by the continuity equation. In this case it is impossible that $n_{int} = 4$.

As almost all of our calculations with the modified upwind scheme have been performed in the context of the unsteady equations, we postpone presentation of sample problems until we describe our transient algorithms. We note, however, that the modified scheme has performed satisfactorily in all cases considered. Nevertheless, since the scheme was designed on the basis of empirical results, and in view of the fact that the mathematical theory of upwind elements is now an active research area, we feel that simplifications and improvements will doubtless be forthcoming.

5. UNSTEADY NAVIER-STOKES EQUATIONS

5.1. Preliminaries

Let $[0, T]$ denote the closed time interval in question; $]0, T[$ denotes the corresponding open interval (i.e., end points omitted). A general point in $[0, T]$ is denoted t . A comma followed by a subscript t is used to denote partial differentiation (e.g., $u_{i,t} = \partial u_i / \partial t$).

If $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is any time-dependent function of \mathbf{x} and t , then we denote by $\mathbf{u}(t)$ the function of \mathbf{x} obtained by freezing t (e.g., $\mathbf{u}(0) = \mathbf{u}(\mathbf{x}, 0)$, the initial value of \mathbf{u}).

Prescribed data are henceforth considered to be time dependent. That is, we assume the following functions are given (cf. Section 2.2):

$$\tilde{f}: \Omega \times]0, T[\rightarrow \mathbb{R}^n, \quad (5.1)$$

$$\tilde{g}: \Gamma_{\tilde{g}} \times]0, T[\rightarrow \mathbb{R}^n, \quad (5.2)$$

$$\tilde{h}: \Gamma_{\tilde{h}} \times]0, T[\rightarrow \mathbb{R}^n. \quad (5.3)$$

We also assume a given starting velocity, namely

$$\mathbf{u}_0: \Omega \rightarrow \mathbb{R}^n. \quad (5.4)$$

5.2. Penalty-Function Formulation

The initial/boundary-value problem for the penalty-function formulation is stated as follows:

Find $\mathbf{u}: \bar{\Omega} \times [0, T] \rightarrow \mathbb{R}^n$ such that

$$\rho(u_{i,t} + u_j u_{i,j}) = t_{ij,j} + f_i \quad \text{on } \Omega \times]0, T[, \quad (5.5)$$

$$u_i = g_i \quad \text{on } \Gamma_{\tilde{g}} \times]0, T[, \quad (5.6)$$

$$t_{ij} n_j = h_i \quad \text{on } \Gamma_{\tilde{h}} \times]0, T[, \quad (5.7)$$

$$u_i(0) = u_{0i} \quad \text{on } \Omega, \quad (5.8)$$

where t_{ij} is defined by (3.4) and (3.5).

Remark. The selection of the penalty parameter is done in the same way as in the steady case; see Section 3.2.

5.3. Weak and Galerkin Formulations

Let $\tilde{g}:]0, T[\rightarrow \mathbf{H}^1$ denote a given extension of \tilde{g} ; that is,

$$\tilde{g} = \tilde{g} \quad \text{on } \Gamma_{\tilde{g}} \times]0, T[. \quad (5.9)$$

A weak form of the initial/boundary-value problem is stated as follows:

Find $\mathbf{u} = \mathbf{w} + \tilde{g}$, $\mathbf{w}: [0, T] \rightarrow \mathbf{V}$, such that for all $\bar{\mathbf{w}} \in \mathbf{V}$

$$\begin{aligned} & \int_{\Omega} (\rho[w_{i,t} + u_j w_{i,j}] \bar{w}_i + \lambda w_{j,j} \bar{w}_{i,i} + 2\mu w_{(i,j)} \bar{w}_{(i,j)}) d\Omega \\ &= \int_{\Omega} f_i \bar{w}_i d\Omega + \int_{\Gamma_{\tilde{h}}} h_i \bar{w}_i d\Gamma - \int (\rho \tilde{g}_{i,t} \bar{w}_i + \lambda \tilde{g}_{j,j} \bar{w}_{i,i} + 2\mu \tilde{g}_{(i,j)} \bar{w}_{(i,j)}) d\Omega, \end{aligned} \quad (5.10)$$

$$\int_{\Omega} w_i(0) \bar{w}_i d\Omega = \int_{\Omega} (u_{i0} - \tilde{g}_i(0)) \bar{w}_i d\Omega. \quad (5.11)$$

The Galerkin formulation may be obtained from the above by replacing \mathbf{u} , \mathbf{w} , \tilde{f} , and \mathbf{V} by \mathbf{u}^h , \mathbf{w}^h , \tilde{f}^h , and \mathbf{V}^h , respectively.

5.4. Matrix Problem

The nodal values, v_{iA} and g_{iA} (cf. Eqs. (2.27) and (2.28)), now become time-dependent functions (e.g., $v_{iA} : [0, T] \rightarrow \mathbb{R}$). Substituting (2.27) and (2.28) into the Galerkin equivalent of (5.10) gives rise to the ordinary differential equation

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}\mathbf{v} + \mathbf{N}(\mathbf{v}) = \tilde{\mathbf{F}}, \quad (5.12)$$

where $\mathbf{v} : [0, T] \rightarrow \mathbb{R}^{n_{eq}}$; a superposed dot is used to denote time differentiation; \mathbf{C} and \mathbf{N} are the same as in Sections 2 through 4; \mathbf{M} is the mass matrix defined by

$$\mathbf{M} = \mathbf{A} \begin{matrix} n_{e1} \\ \mathbf{m}^e \\ e=1 \end{matrix}, \quad (5.13)$$

$$\mathbf{m}^e = [m_{pq}^e], \quad (5.14)$$

$$m_{pq}^e = \delta_{ij} \int_{\Omega^e} \rho N_a^e N_b^e d\Omega, \quad p = n(a-1) + i, \quad q = n(b-1) + j; \quad (5.15)$$

and $\tilde{\mathbf{F}} : [0, T] \rightarrow \mathbb{R}^n$ is defined by

$$\tilde{\mathbf{F}} = \mathbf{F} - \mathbf{A} \begin{matrix} n_{e1} \\ \tilde{\mathbf{f}}^e \\ e=1 \end{matrix}, \quad (5.16)$$

$$\tilde{\mathbf{f}}^e = \{\tilde{f}_p^e\}, \quad (5.17)$$

$$\tilde{f}_p^e = \sum_{q=1}^{n_{e1}} m_{pq}^e \dot{f}_q^e. \quad (5.18)$$

5.5. Lumped Mass Matrices

Throughout we employ techniques which render the mass matrix diagonal (i.e., "lumped" cf. Section 2.10) as this results in several computational simplifications.

For 2- and 3-dimensional, rectilinear, Lagrange elements we use product Lobatto integration rules which diagonalize (5.15). The first two Lobatto rules are the trapezoidal and Simpson's rules, whose products are appropriate for the 4- and 9-node elements, respectively. As this amounts to nodal integration, (5.15) is diagonalized by virtue of the property $N_a^e(\mathbf{x}_b^e) = \delta_{ab}$.

In the axisymmetric case, nodal integration results in zero masses along the z -axis due to the factor r appearing in the volume element. This can cause problems for a transient integration algorithm, and thus we use other techniques which circumvent

this difficulty. For the 4-node, axisymmetric quadrilateral, we use a “row-sum” technique in which

$$m_{pq}^e = \delta_{ij}\delta_{ab} \int_{\Omega^e} \rho N_a^c d\Omega \quad (\text{no sum on “}a\text{”}). \quad (5.19)$$

The integral in (5.19) is evaluated by 2×2 Gauss–Legendre integration.

An evaluation of these, and other, mass lumping techniques, for a variety of elements applied to problems of plate bending, is given in [41]. However, mass lumping is still a controversial issue in fluid mechanics, due to the results of Gresho *et al.* [30].

Mass lumping simplifies (5.16) in that the second term on the right-hand side vanishes; that is,

$$\tilde{\mathbf{F}} = \mathbf{F}. \quad (5.20)$$

Application of corresponding diagonalization techniques to the matrix equation resulting from substituting (2.27) and (2.28) in (5.11), yields a simplified version of the initial condition; namely,

$$\mathbf{v}(0) = \mathbf{v}_0, \quad (5.21)$$

where

$$\mathbf{v}_0 = \{v_{0P}\}, \quad (5.22)$$

$$v_{0P} = u_{0i}(\mathbf{x}_A), \quad P = ID(i, A). \quad (5.23)$$

We assume (5.21) through (5.23) hold henceforth.

5.6. Transient Algorithm

Equations (5.12), (5.20), and (5.21) constitute an initial-value problem for a system of nonlinear ordinary differential equations. To approximately solve this problem, a time-stepping algorithm need be introduced. This is an area of much current research activity and many different ideas have been proposed (see, e.g., [3, 31, 81, 85]). The algorithms we have employed are one-step, “linearly implicit,” predictor–corrector methods and are summarized as follows:

$$(\mathbf{M} + \gamma \Delta t \mathbf{C}) \mathbf{v}_{n+1}^{(i+1)} = \mathbf{M} \tilde{\mathbf{v}}_{n+1} + \gamma \Delta t [\mathbf{F}_{n+1} - \mathbf{N}(\mathbf{v}_{n+1}^{(i)})] \quad (\text{“corrector”}), \quad (5.24)$$

$$\tilde{\mathbf{v}}_{n+1} = \mathbf{v}_n + (1 - \gamma) \Delta t \mathbf{a}_n \quad (\text{“predictor”}), \quad (5.25)$$

$$\mathbf{v}_{n+1}^{(0)} = \tilde{\mathbf{v}}_{n+1}, \quad (5.26)$$

$$\mathbf{a}_{n+1} = (\mathbf{v}_{n+1} - \tilde{\mathbf{v}}_{n+1}) / (\gamma \Delta t), \quad (5.27)$$

where Δt is the time step; $\mathbf{F}_n = \mathbf{F}(t_n)$; \mathbf{v}_n and \mathbf{a}_n are the approximations of $\mathbf{v}(t_n)$ and $\dot{\mathbf{v}}(t_n)$, respectively; γ is a positive parameter which governs the stability and accuracy of the algorithms; and superscripts in parentheses are iteration numbers.

If I denotes the total number of iterations to be performed, then the velocity vector at time t_{n+1} is defined by

$$\mathbf{v}_{n+1} = \mathbf{v}_{n+1}^{(I+1)}. \quad (5.28)$$

Given \mathbf{v}_n and \mathbf{a}_n , (5.24) through (5.28) serve to uniquely define \mathbf{v}_{n+1} and \mathbf{a}_{n+1} .

Remarks. 1. In each time step, (5.24) is used $I + 1$ times. The matrix on the left-hand side of (5.24) is symmetric, positive-definite, and possesses the band-profile structure of \mathbf{C} . In our work so far we have relied on direct elimination schemes for solving (5.24), although we are now experimenting with iterative techniques to cut down storage requirements. For a fixed time step, only one factorization need be performed. The major contributors to the computational cost of the algorithm are the formation of the nonlinear term, $\mathbf{N}(\mathbf{v}_{n+1}^{(i)})$, and the forward reduction/back substitution of the factorized array in obtaining $\mathbf{v}_{n+1}^{(i+1)}$.

2. A local truncation-error analysis reveals that if $I = 0$, the algorithm is first-order accurate, whereas if $I = 1$ and $\gamma = \frac{1}{2}$, second-order accuracy is achieved. The latter algorithm is roughly twice as expensive as the former since twice as many solutions of (5.24) need be performed.

3. We observe from (5.24) that the \mathbf{C} -term is treated "implicitly," whereas the \mathbf{N} -term is treated "explicitly." (We see no hope in treating \mathbf{C} explicitly due to the penalty term.) As long as $\gamma \geq \frac{1}{2}$, no stability condition is engendered by the \mathbf{C} -term.

The stability condition induced by the \mathbf{N} -term has been investigated by way of Fourier analyses [74] of the 1-dimensional, transient, linear, advection-diffusion equation.

The results of these analyses indicate that if $I = 0$, the upwind schemes of Section 4.1 are stable if Δt satisfies a Courant condition. Generalized to two dimensions, the criterion employed takes the form (see Section 4.1 for notation)

$$\Delta t \leq 1 / \left(\frac{|u_\xi|}{h_\xi} + \frac{|u_\eta|}{h_\eta} \right). \quad (5.29)$$

The above inequality must be satisfied for each element in the mesh. We may note that (5.29) is solely a convection condition and, in particular, is independent of the Reynolds number.

On the other hand, if $I = 0$, Gauss-Legendre integration on the \mathbf{N} -term is unstable for all $\Delta t > 0$. A stable scheme, accommodating Gauss-Legendre treatment of the \mathbf{N} -term, can be obtained by using the \mathbf{N} -term and Δt as a function of Δt , however, that this results in the most efficient Gauss-Legendre scheme being roughly twice as expensive as the most efficient upwind scheme.

All our multidimensional calculations have been in accord with the stability criteria outlined above. However, further theoretical work is warranted to make these ideas more precise.

4. The algorithm is initialized by specification of \mathbf{v}_0 and \mathbf{a}_0 . It often suffices to commence calculations with a quiescent state (i.e., $\mathbf{v}_0 = \mathbf{a}_0 = \mathbf{0}$).

5. In time-dependent, creeping-flow problems the nonlinear term may often be neglected. This is important as unconditional stability may be thereby achieved and thus only accuracy governs the size of the time step taken. We refer to omitting the N-term in (5.24) as the “transient Stokes option.”

6. Our program may be run at a constant (input) time step, or at a step redefined adaptively, for each t_n , according to the right-hand side of (5.29).

It is important to cut down on refactorization costs when Δt is being selected adaptively. A scheme we have been employing with success consists of redefining γ to compensate for the step-size change. Specifically, we proceed as follows. Let Δt_{fact} and γ_{fact} denote the values of Δt and γ , respectively, used during the last factorization, and let

$$c = \gamma_{\text{fact}} \Delta t_{\text{fact}}. \quad (5.30)$$

On the basis of the velocity field at t_n , calculate Δt_{crit} according to the right-hand side of (5.29). Define

$$\gamma_{n+1} = c / \Delta t_{\text{crit}}. \quad (5.31)$$

If $\gamma_{n+1} \in [\frac{1}{2}, 1]$, do not refactorize, but set $\gamma = \gamma_{n+1}$ and $\Delta t = \Delta t_{\text{crit}}$ in (5.24) through (5.28). If, on the other hand, $\gamma_{n+1} \notin [\frac{1}{2}, 1]$, set $\Delta t = \Delta t_{\text{crit}}$, $\gamma = \frac{3}{4}$ and refactorize. This value of γ is picked to reduce the likelihood of refactorization in subsequent steps. Other procedures along these lines are under investigation. (See also Park [69] for related ideas.)

7. The convection stability condition of the present schemes is, in some cases, a drawback. For example, if we are interested in an essentially steady flow, and the length of the time interval, T , required to attain steady conditions engenders many steps (e.g., cavity flows), a “fully implicit,” unconditionally stable scheme is no doubt superior. Schemes of this sort, which have been described in [31, 81], have about the same computational structure in each time step as the steady-flow algorithm described in Section 3.5. This involves a nonlinear, nonsymmetric, implicit system involving twice the storage of the linear system required here. We believe that the storage and computational effort engendered by fully implicit algorithms are prohibitive in most cases, as often accuracy dictates taking as small a time step as the convective stability condition. Indeed, our experiences have so far indicated that the transient algorithm of (5.24) through (5.28) is often more reliable and cost effective in obtaining steady flows than the steady-flow algorithm of Section 3.5. Further research needs to be performed to deduce practical guidelines in this matter.

It would be worthwhile to attempt to construct a hybrid, fully/linearly implicit scheme which possessed the virtues of each constituent, but not the defects.

8. Higher-order accuracy may be achieved without iterating by going to a multistep method. This would engender storing additional state vectors, but no

additional calculations, and consequently it might represent a worthwhile improvement.

9. For free-surface flows, and problems of fluid–structure interaction, it is important to be able to move the finite element mesh with the fluid. The generalizations of the preceding algorithm to achieve a so-called “arbitrary Lagrangian–Eulerian” (ALE) formulation (see, e.g., [1, 42, 70]) are discussed in Hughes *et al.* [49]. ■

The effectiveness of the transient algorithm, in solving a variety of flow problems, is illustrated in the next section.

6. NUMERICAL EXAMPLES

In the following we present a sampling of problems which demonstrate the versatility and accuracy of the methods described in this paper. Throughout, unless otherwise specified, we use 4-node quadrilaterals, and the modified (optimal) upwind treatment of the convective forces, as described in Section 4.3. If an estimate of the critical time step is available beforehand, we use a constant time step; if no estimate is available, the variable-step procedure, described in Remark 6 of Section 5.6, is employed. The penalty parameter is selected according to (3.6) unless otherwise specified.

6.1. Couette Flow

The mesh and problem description are shown in Fig. 20 and results are shown in Fig. 21. This is a simple problem in which the convection term is identically zero. A boundary layer develops along the lower edge and diffuses upwards, forming a steady, linear, velocity profile as t increases.

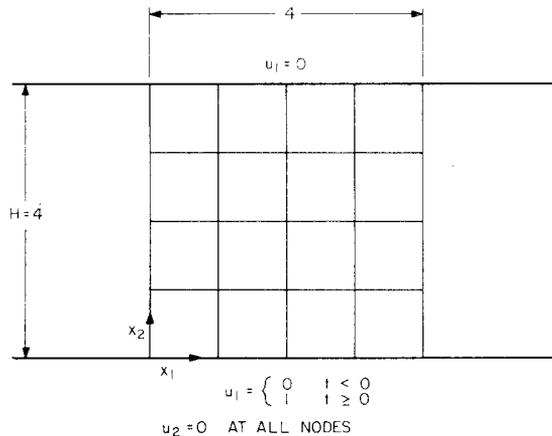


FIG. 20. Couette flow: finite element mesh and problem description.

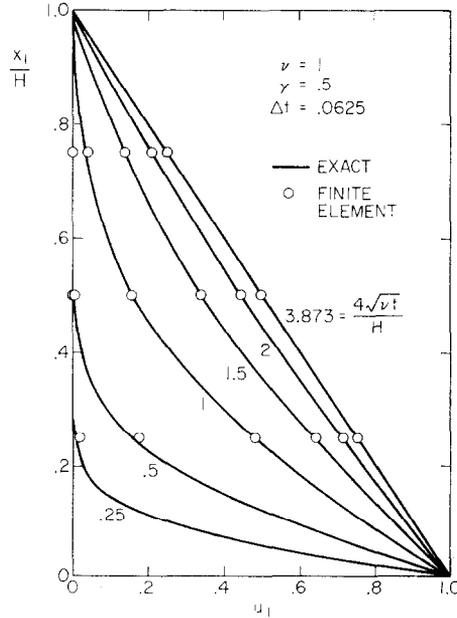


FIG. 21. Couette flow: comparison of finite element results with exact solution.

6.2. Dam-Reservoir Problem

In earthquake engineering, it is of interest to calculate the pressure distribution on the face of a dam caused by suddenly accelerating the dam into a contiguous reservoir. The problem statement is shown in Fig. 22. The dam is assumed rigid. The initial conditions are quiescent and at $t = 0^+$ the dam face is set in constantly accelerating motion towards the reservoir. The transient Stokes option is employed. Data for the problems are given as follows: $\mu = 0$; $\rho = 1$; $L = 2$, $H = 1$; $\lambda = 10^7$; $\gamma = 1$; $\Delta t = 0.025$; and $T = 0.1$ (4 time steps). Meshes and results for two cases are shown in Fig. 23.

Pressures are compared in Fig. 24 with an exact, potential-flow solution due to Chwang [13]. As can be seen the results are in good agreement. (There has been considerable interest in this problem; see, e.g., [14, 55, 79, 88].)

6.3. Hamel Problem

The Hamel problem of convergent flow in a channel ("inflow problem") has been considered recently by several investigators (see [28, 35, 52]). The finite element mesh employed here is shown in Fig. 25. A radial velocity profile, in accord with the high Reynolds number approximation to the exact solution (see Batchelor [5, pp. 294-298]), is set at the outer radius ($r = 4$) having center line magnitude of $\frac{1}{4}$. The circumferential velocity at $r = 4$ is set to zero. The outflow boundary condition at $r = \frac{1}{4}$ is assumed traction-free. (This results in negligible differences at high

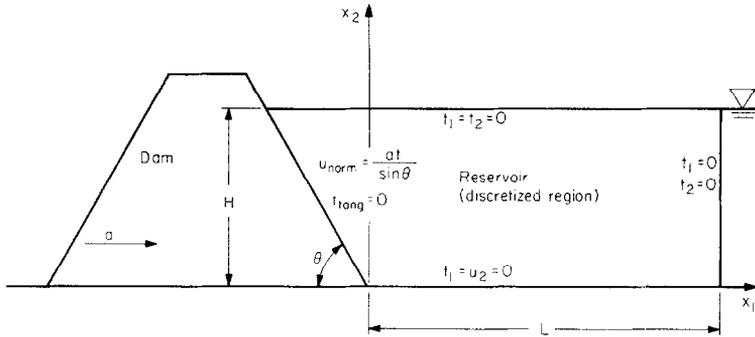


FIG. 22. Dam-reservoir problem: description.

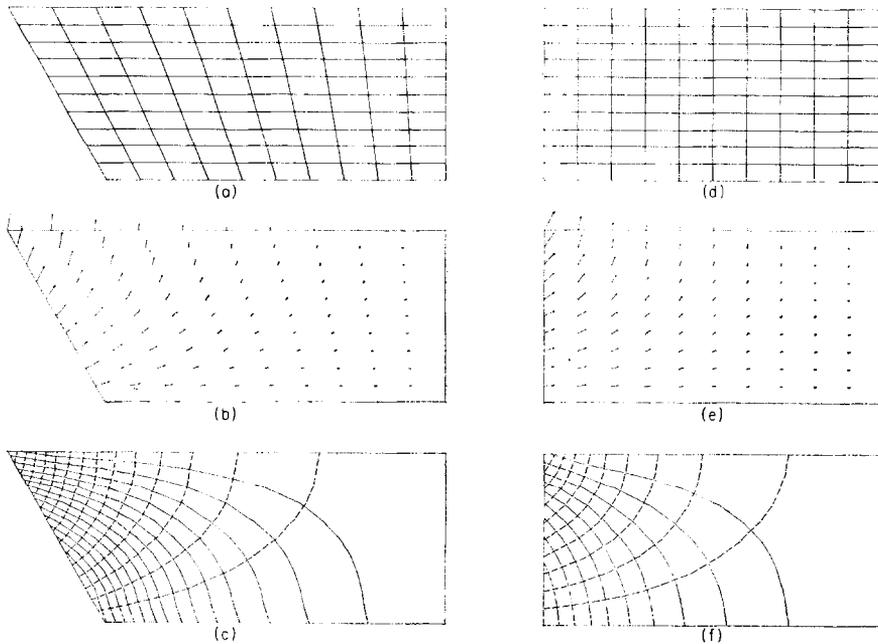


FIG. 23. Dam-reservoir problem. ($\theta = 60^\circ$): (a) Finite element mesh; (b) velocity vectors; and (c) pressure contours (—) and streamlines (---). ($\theta = 90^\circ$): (d) Finite element mesh; (e) velocity

Reynolds numbers when compared with the exact traction boundary condition.) The Reynolds number, following Batchelor [5, p. 295], is given by $Re = \pi/(6\nu)$.

Results for $Re = 500$ and 5×10^7 are presented in Figs. 26 and 27, respectively. As can be seen, the correlation with the exact solution is very good. (Pressures in Figs. 26 and 27 are reported at the element centers and are "unsmoothed.")

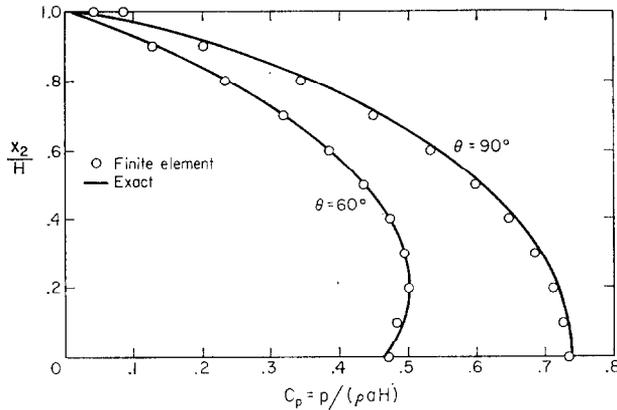


FIG. 24. Dam-reservoir problem: comparison of finite element and exact results.

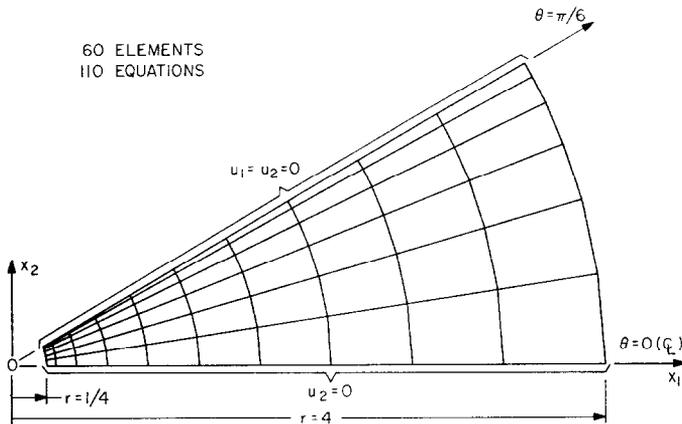


FIG. 25. Hamel flow: finite element mesh.

6.4. Flow over a Step

A problem statement is depicted in Fig. 28.

In Fig. 29 we present results of a calculation performed with 9-node Lagrange elements. A (product) Simpson's rule was used to construct the mass matrix, whereas Gauss-Legendre rules of order 3×3 , 3×3 , and 2×2 , were used on the convection, μ , and λ -terms, respectively. Data employed were $\mu = 1$; $\rho = 200$; $\lambda = 10^9$; $\gamma = 1$; and $\Delta t = 0.07$. As is clearly visible, "wiggles" appear upstream of the step. Similar results are obtained for the 4-node elements employing Gauss-Legendre integration of the convection term. We believe this problem demonstrates the inappropriateness of Gauss-Legendre integration of the convection term.

Results for 4-node elements which employ the modified upwind treatment of the convection term at $Re = 200$ and 10^7 are shown in Figs. 30 and 31, respectively.

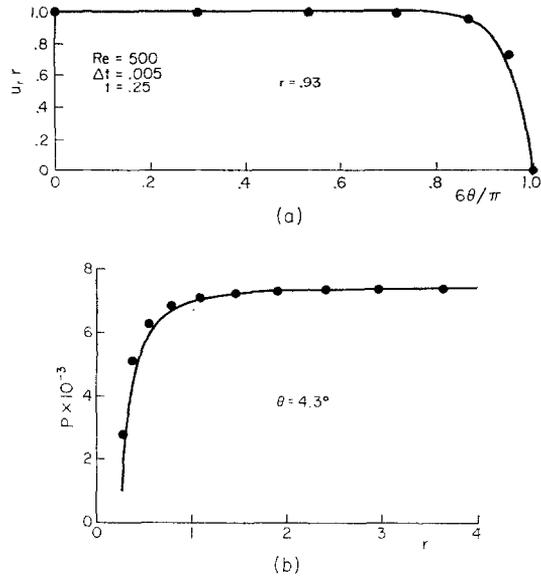


FIG. 26. Hamel flow: Comparison of finite element (●) with exact (—) results at low Reynolds number. (a) Velocity; and (b) pressure.

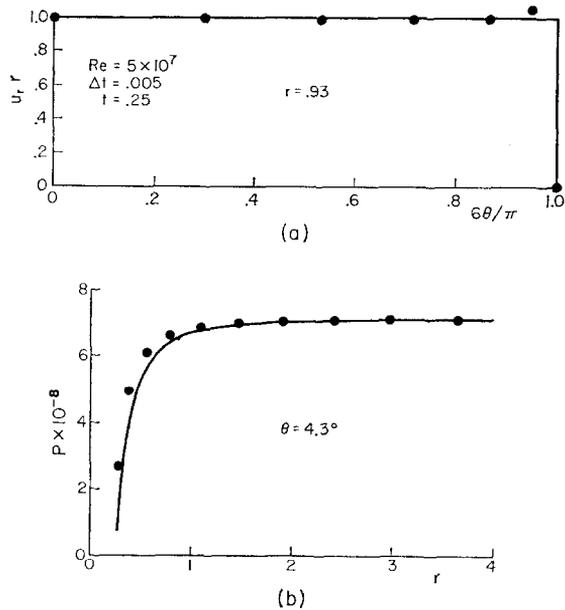


FIG. 27. Hamel flow: comparison of finite element (●) with exact (—) results at high Reynolds number. (a) Velocity; and (b) pressure.

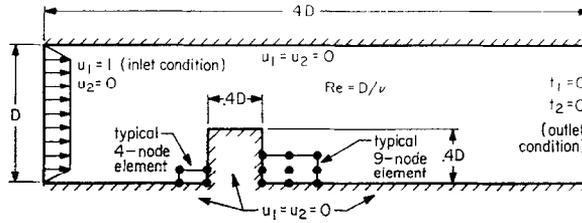


FIG. 28. Flow over a step: problem description and finite element mesh information.

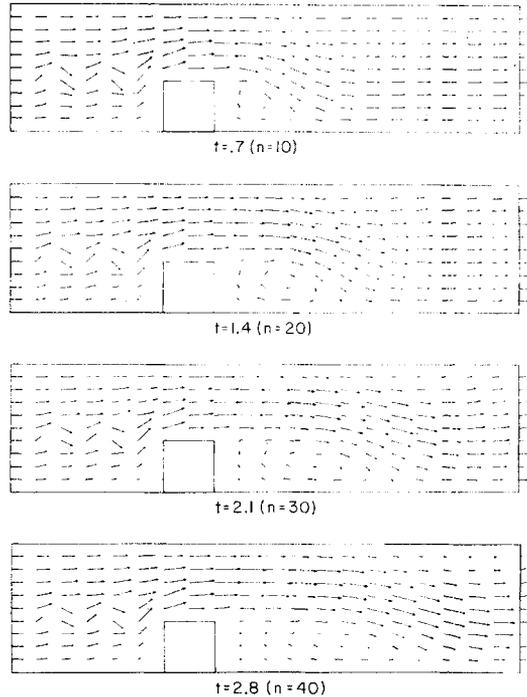


FIG. 29. Flow over a step ($Re = 200$): finite element results for 9-node elements with Gauss-Legendre integration of convection terms.

(Data employed in these cases were: ($Re = 200$) $\mu = 1$; $\rho = 200$; $\lambda = 10^8$; $\gamma = 1$; ($Re = 10^7$) $\mu = 1$; $\rho = 10^7$; $\lambda = 10^{13}$; and $\gamma = 1$. Time steps were selected adaptively.) As can be seen, the upstream “wiggles” are removed in both cases.

6.5. Axisymmetric Flow through a Sudden Enlargement

A problem description is contained in Fig. 32. The domain and mesh are split at section (iii) for pictorial purposes only. Calculations were performed with the transient algorithm at a fixed time step, $\Delta t = 0.5$, and $\gamma = 0.75$. The dynamic viscosity, μ , was set to 1 throughout. Thus $Re = \rho$ and the penalty parameter was

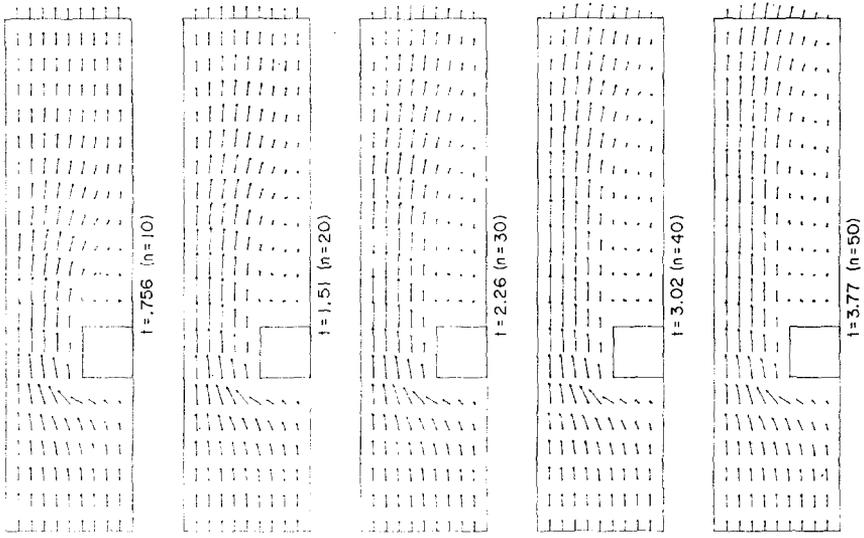


FIG. 31. Flow over a step ($Re = 10^2$): Finite element results for 4-node elements with modified upwind treatment of convection terms.

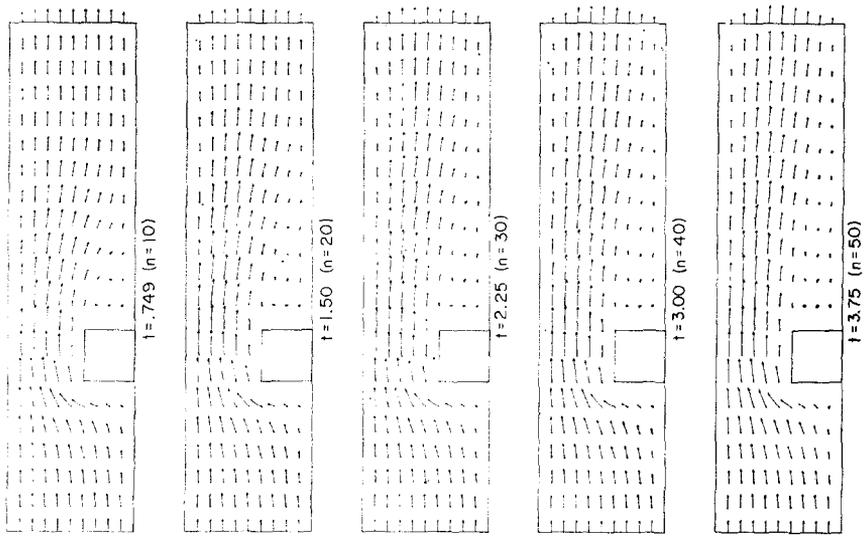


FIG. 30. Flow over a step ($Re = 200$): Finite element results for 4-node elements with modified upwind treatment of convection terms.

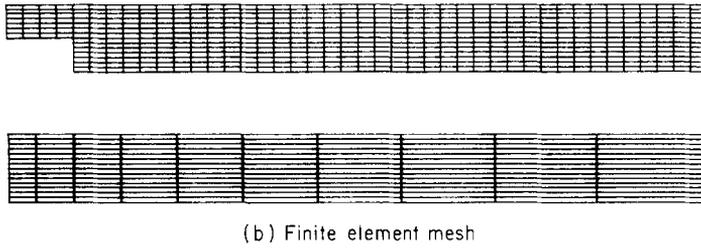
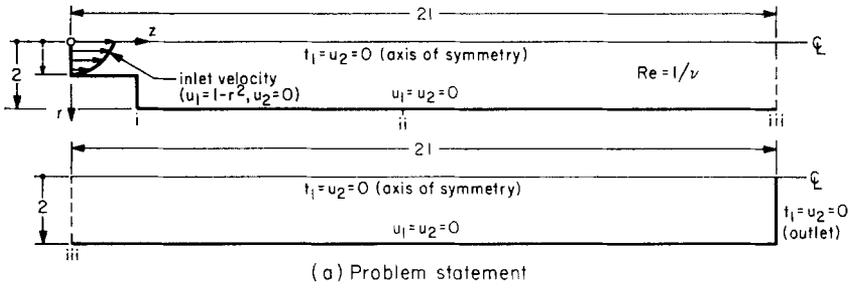


FIG. 32. Axisymmetric flow through a sudden enlargement: problem description.

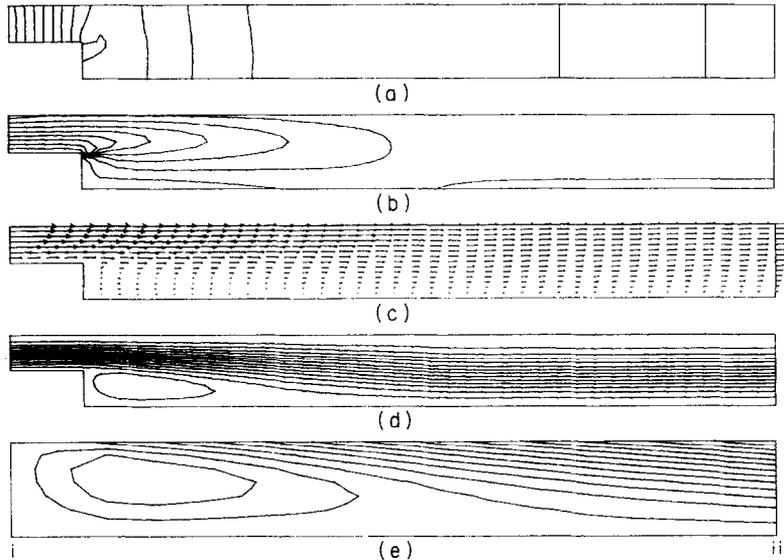


FIG. 33. Axisymmetric flow through a sudden enlargement ($Re = 60$): (a) Pressure contours; (b) vorticity contours; (c) velocity vectors; (d) streamlines; and (e) detail of streamlines in recirculation region.

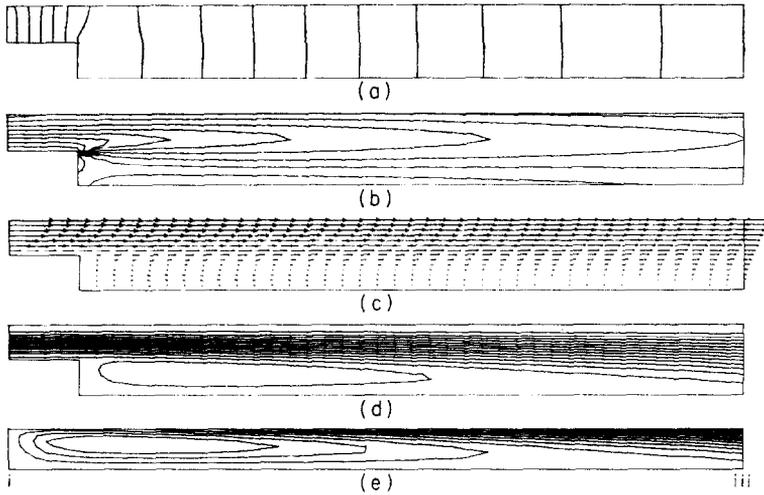


FIG. 34. Axisymmetric flow through a sudden enlargement ($Re = 200$): (a) Pressure contours; (b) vorticity contours; (c) velocity vectors; (d) streamlines; and (e) detail of streamlines in recirculation region.

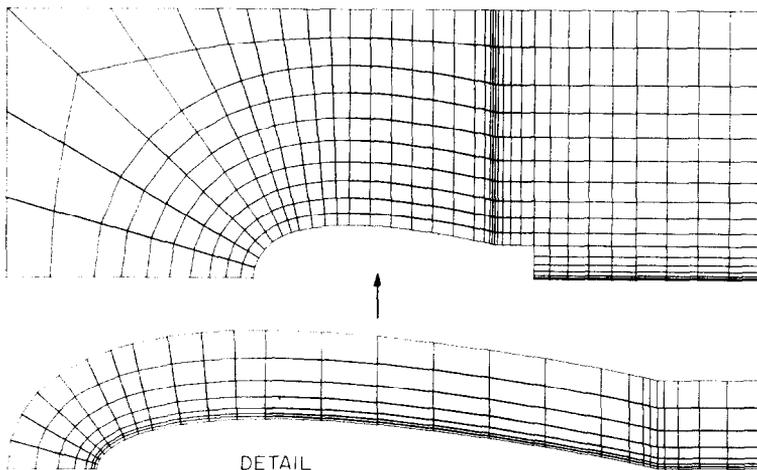
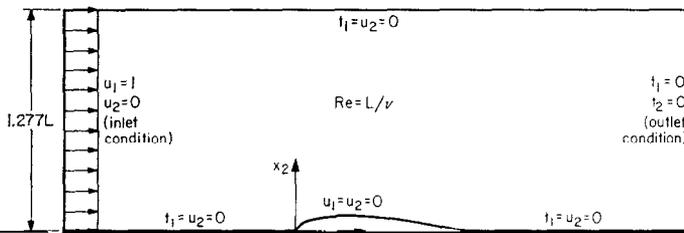


FIG. 36. Viscous flow about an airfoil: finite element mesh.

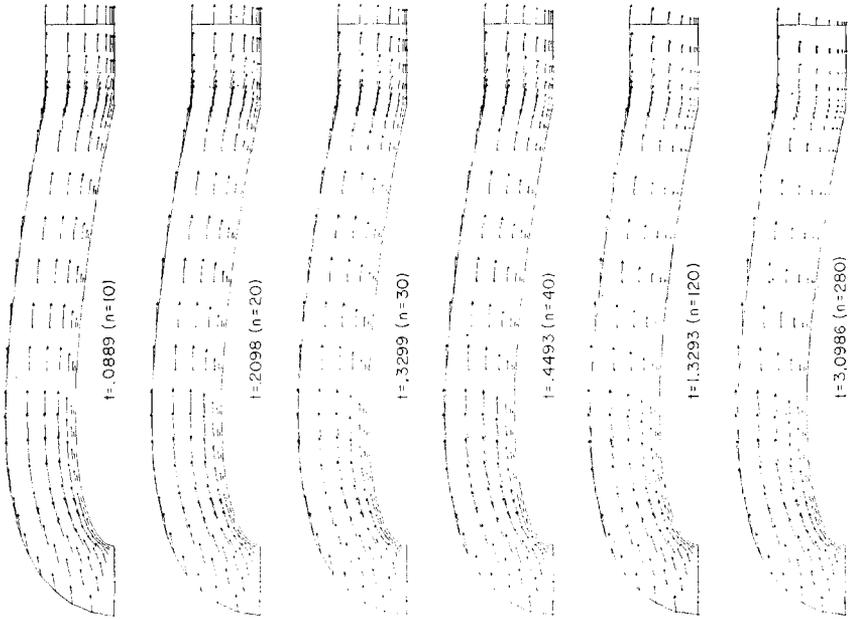


FIG. 38. Viscous flow about an airfoil ($Re = 400$): velocity vectors.

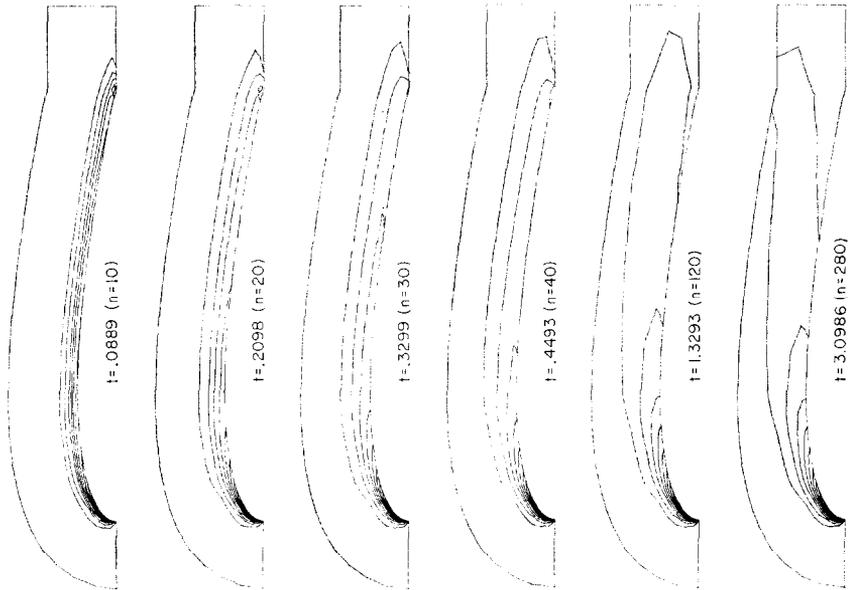


FIG. 37. Viscous flow about an airfoil ($Re = 400$): vorticity contours.

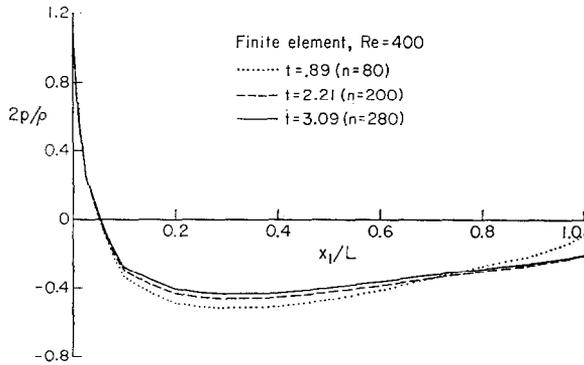


FIG. 39. Viscous flow about an airfoil ($Re = 400$): pressures.

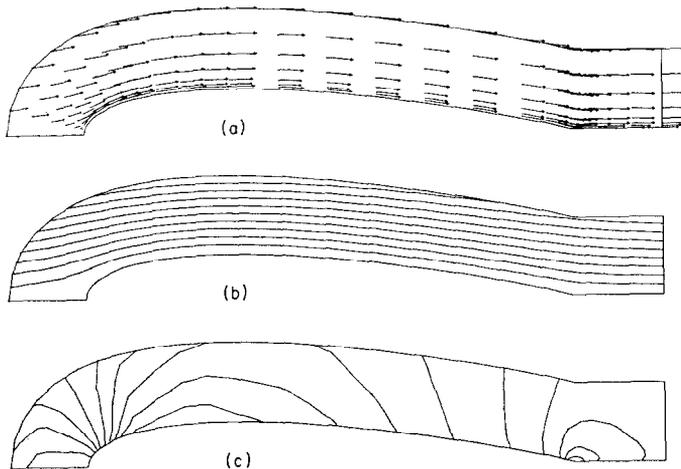


FIG. 40. Viscous flow about an airfoil ($Re = 10^8$):(a) velocity vectors; (b) streamlines; and (c) pressure contours.

taken to be $10^7\rho$. One-point integration of the λ -term was employed. We were interested in steady-flow solutions for comparison with the experimental and numerical results of Macagno and Hung [57]. The calculations were performed in three sequences. In the first sequence, the initial conditions were quiescent and $Re = 30$. The sequence consisted of 60 time steps and a steady flow was achieved after approximately 30 steps. This flow was used as the initial condition for the second sequence in which $Re = 60$. This sequence consisted of 40 steps and a steady condition was attained after 20 steps. Results for this flow are presented in Fig. 33. With this flow as initial condition, the final sequence, in which $Re = 200$, was run for 90 steps. It took almost all this time for the region just upstream of section (iii) to become steady. Results are presented in Fig. 34.

The overall flow patterns are in very good agreement with the experimental and numerical results of Macagno and Hung. The length scales of the trapped annular

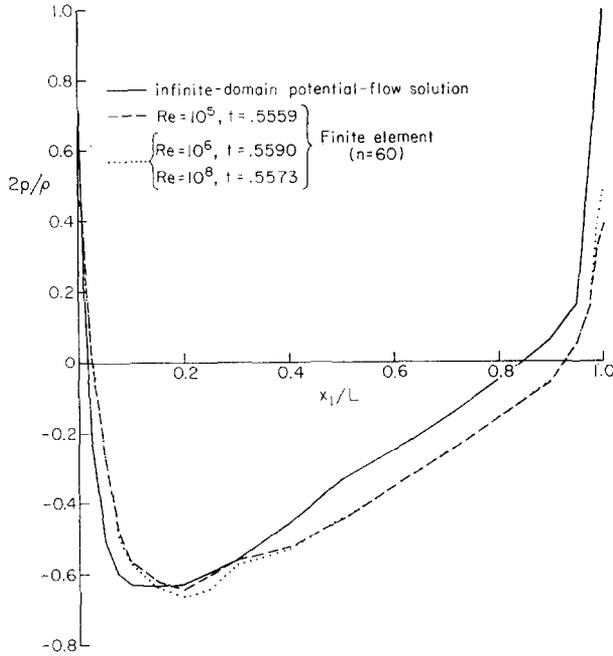


FIG. 41. Viscous flow about an airfoil: Comparison of pressures at different Reynolds numbers.

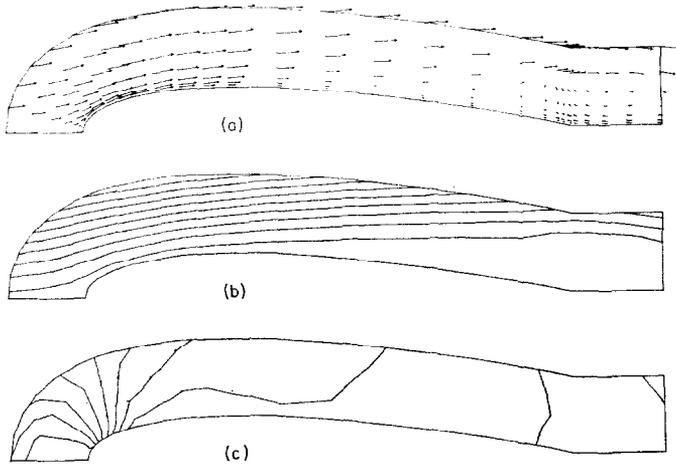


FIG. 42. Viscous flow about an airfoil: Results obtained at $t = 1.2$ after changing Re from 400 to 10^6 . (a) Velocity vectors; (b) streamlines; and (c) pressure contours.

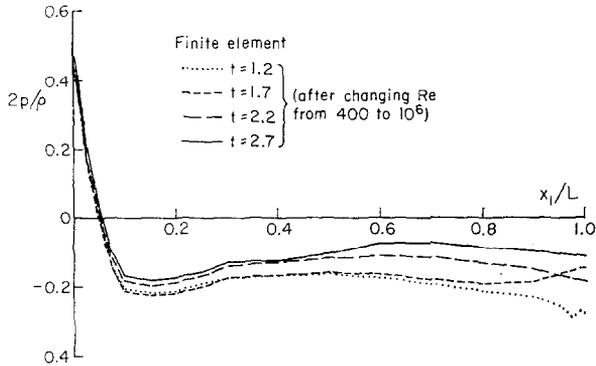
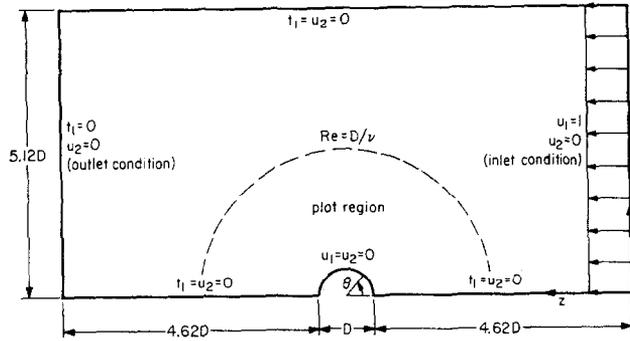
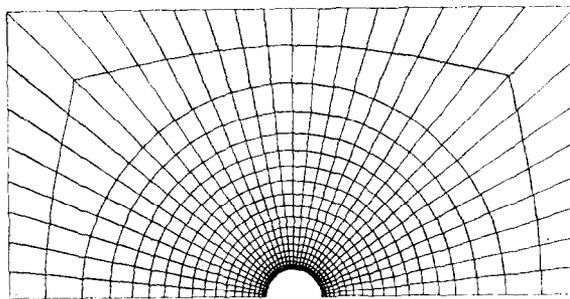


FIG. 43. Viscous flow about an airfoil: Pressures obtained after changing Re from 400 to 10^6 .



(a) Problem statement



(b) Finite element mesh

FIG. 44. Axisymmetric flow around a sphere: problem description and finite element mesh.

eddies correlate particularly well. To study flows at higher Re would require refinement and extension, of the mesh downstream of section (iii), as the recirculation regime tends to stretch out considerably with increasing Re .

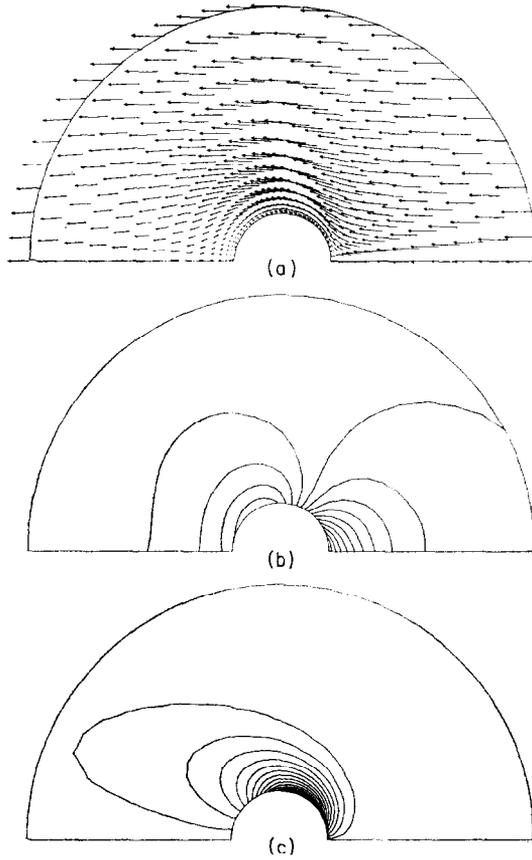


FIG. 45. Axisymmetric flow around a sphere ($Re = 10$): (a) Velocity vectors; (b) pressure contours; and vorticity contours.

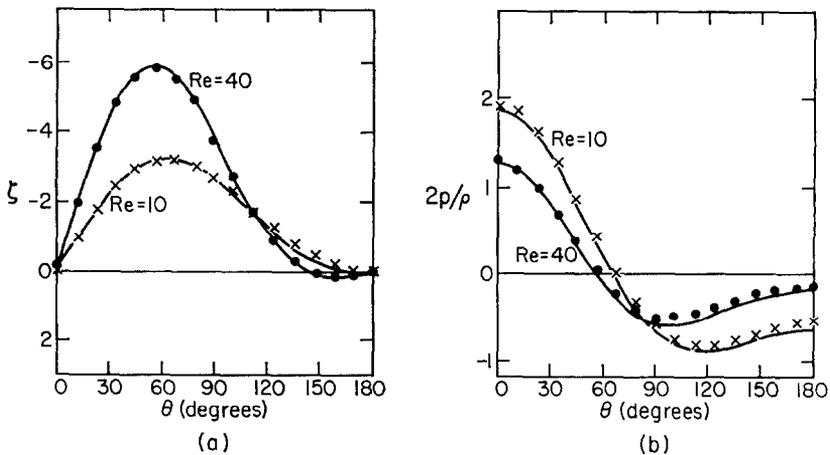


FIG. 46. Axisymmetric flow around a sphere: Comparison of finite element (\bullet , \times) and analytical (—) results. (a) Vorticity; and (b) pressure.

6.6. Viscous Flow About an Airfoil

The problem statement is shown in Fig. 35. An NACA 0018 airfoil geometry is employed in the calculations. The finite element mesh is shown in Fig. 36. We employed unit chord length (i.e., L) throughout.

Several runs were made with this mesh. The first, at low Reynolds number (i.e., 400), and quiescent initial conditions, nicely shows the diffusion of vorticity with the development of the boundary layer; see Figs. 37 and 38. Accompanying pressure profiles are shown in Fig. 39.

Steady, high Reynolds number results, are presented in Figs. 40 and 41. The infinite-domain, potential flow, pressure profiles are presented in Fig. 41 for comparison purposes. We conjecture that enlarging the domain and refining the leading-edge region of the mesh would bring the results in even closer agreement, although the pressure drop at the trailing edge is to be expected in a viscous computation.

An $Re = 10^6$ calculation was made using the $Re = 400$ solution at time 3.0986 as initial condition. In this case the flow separated; sample results are shown in Figs. 42 and 43. At later times (not shown) the separation point and recirculation regime moved downstream.

6.7. Axisymmetric Flow Around a Sphere

The problem description and mesh are shown in Fig. 44. Initial conditions were assumed quiescent and the mean-incompressible treatment of the λ -term was employed (see Remark 4, Section 2.8).

Runs at $Re = 10$ and 40 were made; sample results are shown in Figs. 45 and 46. After the flows became steady, comparisons were made with the infinite-domain, analytical results of Dennis and Walker [15]. As can be seen from Fig. 46, vorticity (ζ) and pressure are in good agreement. (It is somewhat surprising that the results at $Re = 40$ are in such good agreement as our finite-domain model seems hardly adequate for this high a Reynolds number.) In Table II pressure-drag (C_p), viscous-drag (C_v), and total-drag (C_D) coefficients are compared with the results of Dennis and Walker. (The coefficients are normalized as follows: $\text{drag}/(\pi\rho(D/2)^2)$.) The agreement is quite good overall, but better at $Re = 10$ than $Re = 40$, as may be expected.

TABLE II
Comparison of Drag Coefficients for Flow Around a Sphere

Coefficient	Re = 10		Re = 40	
	Present study	Dennis and Walker	Present study	Dennis and Walker
C_p	0.774	0.785	0.347	0.368
C_v	1.419	1.427	0.520	0.536
C_D	2.194	2.212	0.868	0.904

7. CONCLUSIONS

In this paper we have reviewed recent work, and presented new ideas and results, concerning the penalty/finite element formulation of the Navier–Stokes equations. The methods advocated herein have been shown to be effective and, at the same time, are computationally simpler and more versatile than FEMs proposed heretofore.

Nevertheless, considerable research remains to be done. In particular, we look forward to: Improvements in the understanding and implementation of “upwind” treatments of convection terms; a more rigorous stability theory for the transient algorithms, including precise time-step estimates; more effective strategies for changing time steps without refactorizing; second-order accurate time-stepping schemes which involve only one forward reduction/back substitution per step; hybrid, non-linear/linear implicit transient schemes; effective iterative equation solving techniques; a more precise theory for determining the value of the penalty parameter; and the development of a rigorous mathematical theory of convergence and accuracy for the finite element/penalty-function formulation.

In our own research, we plan to continue the development of the present techniques, and merge these procedures with structural/solid mechanics capabilities into a fluid–structure interaction computer program. Although we do not wish to engage in the controversy concerning the relative merits of finite difference and element methods, we do believe that there are compelling reasons for preferring finite elements in the area of fluid–structure interaction. These are due mainly to the intricate geometries which often characterize these problems, interface condition, and the high degree of development of finite element structural models.

ACKNOWLEDGMENTS

We wish to express our gratitude to the Computer Center, California Institute of Technology, for providing computer funds, and to the National Science Foundation for providing support for two of us (W. K. L. and A. B.) under Grant ATA74-19135. We also wish to thank Dr. P. M. Gresho for suggesting to the editors that we write the present article.

REFERENCES

1. A. A. AMSDEN AND C. W. HIRT, YAQUI: An arbitrary Lagrangian–Eulerian computer program for fluid flow at all speeds, Los Alamos Scientific Laboratory, Report LA-5100, Los Alamos, New Mexico, March 1973.
2. J. D. ATKINSON AND T. J. R. HUGHES, Upwind finite element schemes for convective–diffusive equations, Charles Kolling Laboratory Technical Note C-2, The University of Sydney, Sydney, N.S.W., December 1977.
3. A. J. BAKER, A highly stable explicit integration technique for computational continuum mechanics, in “Numerical Methods in Fluid Dynamics” (J. J. Connor and C. A. Brebbia, Eds.), pp. 99–121, Pentech Press, London, 1974.
4. J. BARLOW, *Int. J. Num. Methods Eng.* **10** (1976), 243–251.

5. G. K. BATCHELOR, "An Introduction to Fluid Mechanics," Cambridge Univ. Press, Cambridge, England, 1970.
6. K. J. BATHE AND E. L. WILSON, "Numerical Methods in Finite Element Analysis," Prentice-Hall, Englewood Cliffs, N.J., 1976.
7. T. B. BELYTSCHKO AND J. M. KENNEDY, Computer models for subassembly simulation, preprint.
8. M. BERCOVIER AND M. ENGELMAN, *J. Comput. Phys.*, in press.
9. A. BRANDT, Multi-level adaptive techniques (MLAT) for partial differential equations: Ideas and software, Report No. 77-20, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia, November 1977.
10. C. A. BREBBIA AND J. J. CONNOR (Eds.), "Numerical Methods in Fluid Dynamics" (Proceedings of the International Conference held at the University of Southampton, England, from 26th to 28th September 1973), Pentech Press, London, 1974.
11. O. R. BURGGRAF, *J. Fluid Mech.* **24** (1966), 113-151.
12. I. CHRISTIE, D. F. GRIFFITHS, A. R. MITCHELL, AND O. C. ZIENKIEWICZ, *Int. J. Num. Methods Eng.* **10** (1976), 1389-1396.
13. A. T. CHWANG, *J. Fluid Mech.* **87**, in press.
14. A. T. CHWANG AND G. W. HOUSNER, *J. Fluid Mech.* **87**, in press.
15. S. C. R. DENNIS AND J. D. A. WALKER, *J. Fluid Mech.* **48** (1971), 771-789.
16. R. S. FALK, An analysis of the penalty method and extrapolation for the stationary Stokes equations, in "Advances in Computer Methods for Partial Differential Equations" (R. Vichnevetsky, Ed.), AICA, 1975.
17. C. A. FELIPPA, *Comput. Struct.* **5** (1975), 13-29.
18. C. A. FELIPPA, *Int. J. Num. Methods Eng.* **11** (1977), 709-728.
19. C. A. FELIPPA, *Int. J. Num. Methods Eng.*, in press.
20. G. FICHERA, Existence theorems in elasticity, in "Handbuch der Physik" (C. Truesdell, Ed.), Vol. IVa 2, Springer-Verlag, Berlin, 1972.
21. M. FORTIN AND F. THOMASSET, *J. Comput. Phys.*, in press.
22. I. FRIED, *Int. J. Solids Struct.* **10** (1974), 993-1002.
23. R. H. GALLAGHER, J. T. ODEN, C. TAYLOR, AND O. C. ZIENKIEWICZ (Eds.), "Finite Elements in Fluids—Volume 1, Viscous Flow and Hydrodynamics," Wiley, London, 1975.
24. R. H. GALLAGHER, J. T. ODEN, C. TAYLOR AND O. C. ZIENKIEWICZ (Eds.), "Finite Elements in Fluids—Volume 2, Mathematical Foundations, Aerodynamics and Lubrication," Wiley, London, 1975.
25. D. K. GARTLING, Finite element analysis of viscous incompressible flow, TICOM Report 74-8, University of Texas, Austin, December 1974.
26. D. K. GARTLING, Recent developments in the use of finite element methods in fluid dynamics, in "Computing in Applied Mechanics," AMD-Vol. 18, pp. 65-92, ASME, New York, 1976.
27. D. K. GARTLING, *Int. J. Num. Methods Eng.* **12** (1978), 187-190.
28. D. K. GARTLING, R. E. NICKELL, AND R. I. TANNER, *Int. J. Num. Methods Eng.* **11** (1977), 1155-1174.
29. G. L. GOUDREAU, A computer module for one-step dynamic response of an axisymmetric or plane linear elastic thin shell, Lawrence Livermore Laboratory Report UCID-17730, February 1978.
30. P. M. GRESHO, R. L. LEE, AND R. L. SANI, Advection-dominated flows, with emphasis on the consequences of mass lumping, in "Preprints of the Second International Symposium on Finite Element Methods in Flow Problems, S. Margherita Ligure, Italy, June 14-18, 1976."
31. P. M. GRESHO, R. L. LEE, T. W. STULLICH, AND R. L. SANI, Solution of the time-dependent Navier-Stokes equations via F.E.M., in "Second International Conference on Finite Elements in Water Resources, London, England, July 10-14, 1978."
32. D. F. GRIFFITHS, On the approximation of convection problems in fluid dynamics, Research Paper No. 317, Department of Mathematics and Statistics, University of Calgary, Calgary, Alberta, Canada, September 1976.

33. D. F. GRIFFITHS AND J. LORENZ, An analysis of the Petrov–Galerkin method applied to a model problem, Research Paper 334, Department of Mathematics and Statistics, University of Calgary, Calgary, Alberta, Canada, February 1977.
34. J. C. HEINRICH, P. S. HUyakORN, O. C. ZIENKIEWICZ, AND A. R. MITCHELL, *Int. J. Num. Methods Eng.* **11** (1977), 131–143.
35. J. C. HEINRICH, R. S. MARSHALL, AND O. C. ZIENKIEWICZ, Solution of Navier–Stokes equation by a penalty function finite element method, Report No. C/R/308/78, Department of Civil Engineering, University of Wales, Swansea, 1978.
36. E. HINTON, Least squares analysis using finite elements, M. Sc. Thesis, Civil Engineering Department, University of Wales, Swansea, 1968.
37. E. HINTON, *J. Sound Vibr.* **46** (1976), 465–472.
38. E. HINTON, D. R. J. OWEN, AND D. SHANTARAM, Dynamic transient linear and nonlinear behaviour of thick and thin plates, in “Conference on the Mathematics of Finite Elements and Applications II, Brunel University, 1975.”
39. E. HINTON AND E. D. L. PUGH, Some quadrilateral isoparametric finite elements based on Mindlin plate theory, in “Proceedings of the Symposium on Applications of Computer Methods in Engineering, University of Southern California, Los Angeles,” Vol. II, pp. 851–858, 1977.
40. E. HINTON, A. RAZZAQUE, O. C. ZIENKIEWICZ, AND J. D. DAVIES, *Proc. Inst. Civil Eng.* **59** (1975), 43–65.
41. E. HINTON AND N. BICANIC, *Comput Struct.*, in press.
42. C. W. HIRT, J. D. RAMSHAW, AND L. R. STEIN, Numerical simulation of three-dimensional flow past bluff bodies, Los Alamos Scientific Laboratory, Report LA-OR-77-1420, Los Alamos, New Mexico, 1977.
43. T. J. R. HUGHES, *J. Appl. Mech.* **44** (1977), 181–183.
44. T. J. R. HUGHES, *Int. J. Num. Methods Eng.* **12** (1978), 1359–1365.
45. T. J. R. HUGHES AND J. ATKINSON, A variational basis for “upwind” finite elements, in “IUTAM Symposium on Variational Methods in the Mechanics of Solids, Northwestern University, Evanston, Illinois, September, 1978.”
46. T. J. R. HUGHES AND M. COHEN, *Comput. Struct.* **9** (1978), 445–450.
47. T. J. R. HUGHES AND D. S. MALKUS, On the equivalence of mixed finite element methods with reduced/selective integration displacement methods, in “Proceedings of the Symposium on Applications of Computer Methods in Engineering, University of Southern California, Los Angeles,” Vol. I, pp. 23–32, 1977.
48. T. J. R. HUGHES, M. COHEN, AND M. HAROUN, *Nucl. Eng. Des.* **46** (1978), 203–222.
49. T. J. R. HUGHES, W. K. LIU, AND T. ZIMMERMAN, Lagrangian–Eulerian finite element formulation for incompressible, viscous flows, in “U.S.–Japan Seminar on Interdisciplinary Finite Element Analysis, Cornell University, Ithaca, New York, August 7–11, 1978.”
50. T. J. R. HUGHES, R. L. TAYLOR AND W. KANOKNUKULCHAI, *Int. J. Num. Methods Eng.* **11** (1977), 1529–1543.
51. T. J. R. HUGHES, R. L. TAYLOR, AND J. F. LEVY, A finite element method for incompressible viscous flows, in “Preprints of the Second International Symposium on Finite Element Methods in Flow Problems, S. Margherita Ligure (Italy) June 14–18, 1976.”
52. T. J. R. HUGHES, R. L. TAYLOR, AND J. F. LEVY, High Reynolds number, steady, incompressible flows by a finite element method, in “Finite Elements in Fluids,” Vol. 3, Wiley, London, 1978.
53. T. J. R. HUGHES, R. L. TAYLOR, AND J. L. SACKMAN, Finite element formulation and solution of contact–impact problems in continuum mechanics—III, SESM Report No. 75-7, Department of Civil Engineering, University of California, Berkeley, July 1975.
54. T. J. R. HUGHES, R. L. TAYLOR, J. L. SACKMAN, AND W. KANOKNUKULCHAI, Finite element formulation and solution of contact–impact problems in continuum mechanics—IV, SESM Report No. 76-4, Department of Civil Engineering, University of California, Berkeley, July 1976.
55. T. VON KARMAN, *Trans. Amer. Soc. Civil Eng.* **98** (1933), 418–433.

56. R. L. LEE, P. M. GRESHO, AND R. L. SANI, Numerical smoothing techniques applied to some finite element solutions of the Navier–Stokes equations, in “Second International Conference on Finite Elements in Water Resources, London, England, July 10–14, 1978”; see also: *Int. J. Num. Methods Eng.*, in press.
57. E. O. MACAGNO AND T. K. HUNG, *J. Fluid Mech.* **28** (1967), 43–67.
58. D. S. MALKUS, Finite element analysis of incompressible solids, Ph.D. Thesis, Boston University, 1975.
59. D. S. MALKUS, *Int. J. Solids Struct.* **12** (1976), 731–738.
60. D. S. MALKUS AND T. J. R. HUGHES, *Comput. Methods Appl. Mech. Eng.* **15** (1978), 63–81.
61. B. MERCIER, Solution of the limit load problem via the finite element method, in “Proceedings of the Symposium on Applications of Computer Methods in Engineering,” Vol. 1, pp. 445–454, University of Southern California, Los Angeles, August 23–26, 1977.
62. A. R. MITCHELL AND D. F. GRIFFITHS, Generalized Galerkin methods for second order equations with significant first derivative terms, pp. 90–104 in “Numerical Analysis, Proceedings, Biennial Conference, Dundee, 1977,” Lecture Notes in Mathematics No. 630, Springer-Verlag, Berlin, 1978.
63. A. R. MITCHELL AND D. F. GRIFFITHS, Semi-discrete generalized Galerkin methods for time-dependent conduction–convection problems, Report NA/24, Department of Mathematics, University of Dundee, Scotland, April 1978.
64. D. P. MONDKAR AND G. H. POWELL, *Comput. Struct.* **4** (1974), 531–548.
65. D. P. MONDKAR AND G. H. POWELL, *Comput. Struct.* **4** (1974), 699–728.
66. J. C. NAGTEGAAL, D. M. PARKS, AND J. R. RICE, *Comput. Methods Appl. Mech. Eng.* **4** (1974), 153–178.
67. D. J. NAYLOR, *Int. J. Num. Methods Eng.* **8** (1974), 443–460.
68. M. D. OLSON AND S. Y. TUANN, Primitive variables versus stream function finite element solutions of the Navier–Stokes equations, in “Preprints of the Second International Symposium on Finite Elements in Flow Problems, S. Margherita Ligure, Italy, June 14–18, 1976.”
69. K. C. PARK, An implicit, variable-step technique for fluid dynamics problems, pp. 33–39 in “Preprints of the Second International Symposium on Finite Element Methods in Flow Problems, S. Margherita Ligure, Italy, June 14–18, 1976.”
70. W. E. PRACTH AND J. U. BRACKBILL, BAAL: A code for calculating three-dimensional fluid flows at all speeds with an Eulerian–Lagrangian computing mesh, Los Alamos Scientific Laborat-

83. R. L. TAYLOR, Computer procedures for finite element analysis, Chap. 24 in "The Finite Element Method," 3rd ed., McGraw-Hill, London, 1977.
84. R. L. TAYLOR AND O. C. ZIENKIEWICZ, Complementary energy with penalty functions in finite element analysis, preprint.
85. R. TEMAM, "Navier-Stokes Equations," North-Holland, Amsterdam, 1977.
86. S. Y. TUANN AND M. D. OLSON, Studies of rectangular cavity flow with Reynolds number by a finite element method, Structural Research Series. Report No. 19, Department of Engineering, University of British Columbia, Vancouver, British Columbia, June 1977.
87. E. L. WACHSPRESS, Isojacobic Crosswind Differencing," pp. 190-199, in "Numerical Analysis, Proceedings, Biennial Conference, Dundee, 1977," Lecture Notes in Mathematics No. 630, Springer-Verlag, Berlin, 1978.
88. H. M. WESTERGAARD, *Trans. Amer. Soc. Civil Eng.* **98** (1933), 418-433.
89. E. L. WILSON, K. J. BATHE, AND W. P. DOHERTY, *Comput. Struct.* **4** (1974), 363-372.
90. E. L. WILSON AND H. H. DOVEY, Solution or reduction of equilibrium equations for large complex structural systems, preprint.
91. O. C. ZIENKIEWICZ, "The Finite Element Method," 3rd ed., McGraw-Hill, London, 1977.
92. O. C. ZIENKIEWICZ, *Int. J. Num. Methods Eng.* **12** (1978), 191.
93. O. C. ZIENKIEWICZ AND E. HINTON, *J. Franklin Inst.* **302** (1976), 443-461.
94. O. C. ZIENKIEWICZ, R. L. TAYLOR AND J. M. TOO, *Int. J. Num. Methods Eng.* **3** (1971), 275-290.