

**STREAMLINE UPWIND/PETROV-GALERKIN FORMULATIONS
FOR CONVECTION DOMINATED FLOWS WITH
PARTICULAR EMPHASIS ON THE INCOMPRESSIBLE
NAVIER-STOKES EQUATIONS**

Alexander N. BROOKS*

AeroVironment Inc., Pasadena, CA 91107, U.S.A.

Thomas J.R. HUGHES**

Division of Applied Mechanics Stanford University, Stanford, CA 94305, U.S.A.

A new finite element formulation for convection dominated flows is developed. The basis of the formulation is the streamline upwind concept, which provides an accurate multidimensional generalization of optimal one-dimensional upwind schemes. When implemented as a consistent Petrov-Galerkin weighted residual method, it is shown that the new formulation is not subject to the artificial diffusion criticisms associated with many classical upwind methods.

The accuracy of the streamline upwind/Petrov-Galerkin formulation for the linear advection diffusion equation is demonstrated on several numerical examples. The formulation is extended to the incompressible Navier-Stokes equations. An efficient implicit pressure/explicit velocity transient algorithm is developed which accommodates several treatments of the incompressibility constraint and allows for multiple iterations within a time step. The effectiveness of the algorithm is demonstrated on the problem of vortex shedding from a circular cylinder at a Reynolds number of 100.

1. Introduction

The finite element method is a valuable tool in the solution of many engineering problems. In situations where the governing equations are known, but complicated geometry or boundary conditions render analytical solutions difficult or impossible to obtain, the finite element method is often employed. The finite element method makes use of a spatial discretization and a weighted residual formulation to arrive at a system of matrix equations. Solution of the matrix equations yields an approximate solution to the original boundary value problem.

The most common weighted residual formulation employed has been the Galerkin method, in which weighting and interpolation functions are from the same class of functions. The Galerkin method, when applied to most structures or heat conduction problems, leads to symmetric stiffness matrices. In this case, it can be shown that the solution possesses the 'best

*Fluid Dynamics Engineer, formerly Graduate Research Assistant, California Institute of Technology and Stanford University.

**Associate Professor of Mechanical Engineering.

approximation' property. That is, the difference between the finite element solution and the exact solution is minimized with respect to a certain norm. The success of the Galerkin finite element method in structures applications is largely due to the 'best approximation' result.

Initial finite element formulations for convective transport problems also used the Galerkin method, but with mixed results. In fluid flows or convective heat transfer, the matrix associated with the convective term is nonsymmetric, and as a result, the 'best approximation' property is lost. In practice, solutions are often corrupted by spurious node-to-node oscillations or 'wiggles'. Wiggles are most likely to appear in convection dominated cases (high Peclet or Reynolds number) when a downstream boundary condition forces a rapid change in the solution. An example of this phenomenon is shown in Fig. 1.1 for fluid flow past a block in a narrow channel. The only way to eliminate the oscillations is to severely refine the mesh, such that convection no longer dominates on an element level. Obviously, mesh refinement is needed in regions where accurate representation of the solution is required. However, often only the global solution features are desired, and in this case mesh refinement is required only to prevent oscillations. This has provided motivation for development of an alternative to the Galerkin formulation which precludes spurious oscillations, regardless of mesh refinement.

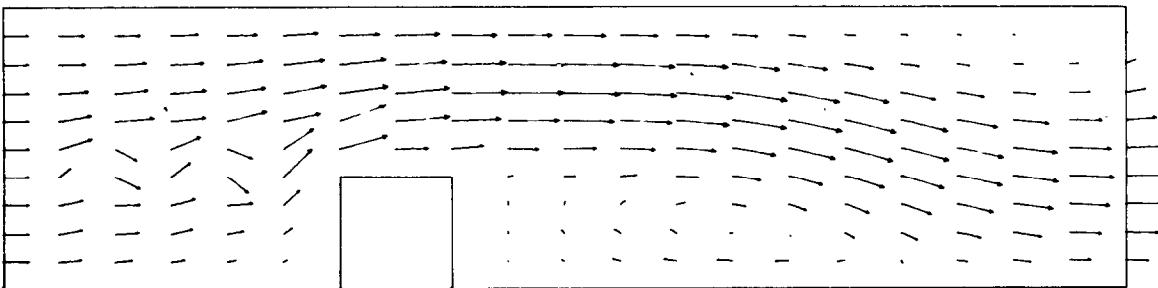


Fig. 1.1. Oscillations generated upstream of a block ($Re = 200$).

It is well known that the Galerkin finite element method gives rise to central-difference type approximations of differential operators. It is thus not surprising that 'wiggles' have also afflicted central-difference finite difference solutions [41]. In fact, obtaining solutions at all has been problematic due to the reliance on iterative solvers in finite differences coupled with the lack of diagonal dominance of central-difference equations.

It was discovered, however, that wiggle-free solutions could be obtained by the use of 'upwind' differencing on the convective term. Upwind differencing amounts to approximating the convective derivatives with solution values at the upstream and central nodes of a three-node stencil. The drawback is that upwind differences are only first-order accurate (central differences are second-order accurate). The loss of accuracy is manifested as overly diffuse solutions. It is well known that the upwinded convective term can be constructed simply by adding artificial diffusion to a central difference treatment. This artificial diffusion interpretation has been the basis for extensive criticism of upwind methods (see e.g. [7, 12, 33]).

Subsequently, it became apparent that a combination of central and upwind differences, based on Peclet or Reynolds number, was better than either upwind or central differences alone. For a simple one-dimensional model problem, it was possible to select the combination

which resulted in exact nodal solutions. Equivalently, the proper amount of artificial diffusion could be added to the central formulation. This procedure is usually referred to as an ‘optimal’ or ‘smart’ upwind method.

In a finite element framework, upwinded convective terms can be developed in several different ways. The initial upwind finite element formulation, presented by Christie et al. [5] for the one-dimensional advection-diffusion equation, employed modified weighting functions to achieve the upwind effect. In essence, the element upstream of a node is weighted more heavily than the element downstream of a node. The method of [5] was later generalized to the two-dimensional case by Heinrich et al. [17]. Since the modified weighting function is applied to all terms in the equations, these formulations were consistent Petrov–Galerkin weighted residual methods.

In [19], Hughes presented a simple method for generating upwind elements, which made use of a modified quadrature rule for the convective term.

In a different approach, Hughes and Atkinson [22] presented an optimal upwind method that was derived from a variational principle, demonstrating that upwind methods may be developed from a firm theoretical basis.

Many of the optimal upwind finite element formulations lead to the same system of matrix equations and give exact solutions for the one-dimensional model problem. Unfortunately, when generalized to more complicated situations, some of these formulations are far from optimal. In multidimensional problems, solutions often exhibit excessive diffusion perpendicular to the flow direction. Overly diffuse results also appear in transient problems, or when source terms are present. In addition, in many instances the Galerkin formulation gives oscillation-free solutions which are more accurate than upwind solutions.

Obviously, such results cause upwind techniques to be viewed with some suspicion (and even contempt). Some investigators believe that use of the Galerkin formulation, along with mesh refinement, is the only possible way to achieve solutions that are both accurate and wiggle free.

The outlook for upwind finite elements, despite the problems and criticisms, is not bleak at all. Recent developments have shown that when upwind methods are constructed through the proper Petrov–Galerkin weighted residual formulation, none of the above problems are encountered (see, e.g., [23–25]). The only ostensible shortcoming of this new formulation is that its name may contain the word ‘upwind’, which carries with it so many negative connotations.

In this paper, the streamline upwind/Petrov–Galerkin formulation is presented. This method has the robust qualities of a classical upwind method, but is not subject to any of the artificial diffusion criticisms mentioned above. The basic idea of the streamline upwind method is to add diffusion (or viscosity) which acts only in the flow direction. Extended to a Petrov–Galerkin formulation, the standard Galerkin weighting functions are modified by adding a streamline upwind perturbation, which again acts only in the flow direction. The modified weighting function is applied to all terms in the equation, resulting in a consistent weighted residual formulation.

Further background on the development of upwind techniques is contained in Section 2. It is shown that Galerkin (or central difference) solutions are often underdiffuse.

In Section 3, the streamline upwind/Petrov–Galerkin method is developed for the linear advection-diffusion equation. Several alternatives are discussed for picking a free parameter

which determines the amount of upwind weighting. It is noted that the structure of the modified weighting functions is far more important than the actual value of the parameter, and that the additional terms should not be interpreted as artificial diffusion. Numerical examples are used to demonstrate the superiority of the streamline upwind/Petrov–Galerkin method over Galerkin or classical upwind methods.

Implicit/explicit transient algorithms are reviewed in Section 4. It is shown that explicit Galerkin solutions can be quite *underdiffuse*, effectively increasing the Peclet or Reynolds number. An explicit multiple-iteration algorithm is proposed for the streamline upwind/Petrov–Galerkin method. The algorithm is shown to be exceptionally accurate, exhibiting almost no negative (or positive) artificial diffusion. Time step restrictions for this algorithm are seen to be much more favorable than those for an explicit Galerkin algorithm.

In Section 5, the streamline upwind/Petrov–Galerkin formulation is developed for the incompressible Navier–Stokes equations. Several different options for treatment of the incompressibility constraint are included in the formulation. The transient algorithms from Section 4 are generalized for use with the Navier–Stokes equations. An effective procedure is developed in which pressure is treated implicitly and velocity is treated explicitly.

In Section 6, the new Navier–Stokes algorithm is applied to a challenging flow simulation. The problem is the development of vortex shedding from a circular cylinder at a Reynolds number of 100. Results obtained are quite good, and the method is shown to be very efficient computationally.

Finally, in Section 7, the present work is summarized and suggestions for further research are made.

2. Review of the development of upwind techniques

2.1. One-dimensional model problem

In this section we consider a simple one-dimensional model problem which demonstrates the motivation behind upwind methods. Consider the one-dimensional advection-diffusion equation,

$$u\varphi_{,x} = k\varphi_{,xx} \quad (2.1.1)$$

where u is the given flow velocity, and k is the diffusivity. For this case, both u and k are assumed constant and positive. The problem consists of finding φ satisfying (2.1.1) and

$$\varphi = 0 \quad \text{at } x = 0, \quad (2.1.2)$$

$$\varphi = 1 \quad \text{at } x = L. \quad (2.1.3)$$

The exact solution of (2.1.1)–(2.1.3) is,

$$\varphi(x) = \frac{1 - e^{Pe x/L}}{1 - e^{Pe}}, \quad (2.1.4)$$

where Pe is the global Peclet number ($Pe = uL/k$).

At very low Peclet number, the solution, which is diffusion dominated, is virtually a straight line between the inflow boundary condition at $x = 0$ and the outflow boundary condition at $x = L$. In advection dominated situations (high Pe), the solution is equal to the inflow boundary condition, except in a thin boundary layer at the outflow boundary. Upwind methods are developed to overcome problems exhibited by conventional methods in the advection dominated case.

At this point it is useful to introduce the concept of the finite difference stencil. In finite difference methods (and indirectly in finite element methods), differential operators are approximated by difference operators. In the above example, consider a set of equally spaced points between $x = 0$ and $x = L$, and a corresponding set of approximate solution values. Let x_A denote an interior point, and let x_{A-1} and x_{A+1} denote points neighboring x_A . In the finite difference method, the central approximations to $\varphi_{,x}(x_A)$ and $\varphi_{,xx}(x_A)$ are defined by,

$$\varphi_{,x}(x_A) \approx (-\varphi_{A-1} + \varphi_{A+1})/2h, \quad (2.1.5)$$

$$\varphi_{,xx}(x_A) \approx (\varphi_{A-1} - 2\varphi_A + \varphi_{A+1})/h^2, \quad (2.1.6)$$

where h is the point spacing. The *finite difference stencils* corresponding to (2.1.5) and (2.1.6) are defined by,

$$\frac{d}{dx} \Big|_{x_A} \sim \frac{1}{2h} (-1 \ 0 \ 1), \quad (2.1.7)$$

$$\frac{d^2}{dx^2} \Big|_{x_A} \sim \frac{1}{h^2} (1 \ -2 \ 1). \quad (2.1.8)$$

The values are seen to be simply the coefficients of φ_{A-1} , φ_A , and φ_{A+1} . Eqs. (2.1.7) and (2.1.8) are referred to as the first and second central difference operators, respectively. It is also useful to define the grid Peclet number α , viz.

$$\alpha = uh/(2k). \quad (2.1.9)$$

REMARK. It should be noted that a Galerkin finite element discretization of (2.1.1), involving piecewise linear interpolations, results in the above central-difference approximations.

Central-difference solutions to (2.1.1)–(2.1.3) for a grid of 11 points are compared with nodally exact solutions in Fig. 2.1. The problem with the central-difference method is easily seen—spurious oscillations, or ‘wiggles’, occur at grid Peclet numbers greater than 1, rendering the solution useless.

2.2. Upwind differences

A cure for the unwanted wiggles is provided by use of upwind differences to approximate the convective term in (2.1.1). Specifically, $\varphi_{,x}(x_A)$ is approximated only by φ_A and the value of φ at the point upstream of x_A . The upwind difference stencils are

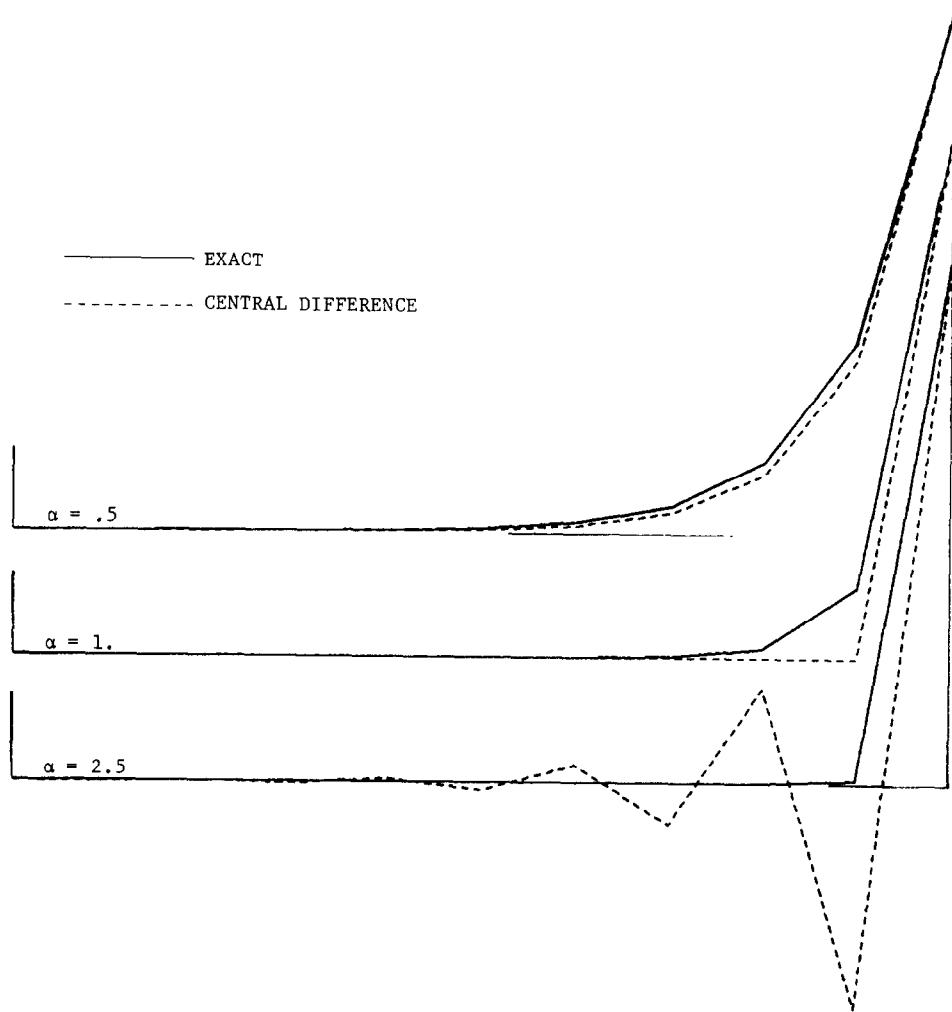


Fig. 2.1. Steady advection-diffusion in one dimension: Central differences.

$$\frac{d}{dx} \Big|_{x_A} \sim \frac{1}{h} (-1 \ 1 \ 0), \quad u > 0, \quad (2.2.1)$$

$$\frac{d}{dx} \Big|_{x_A} \sim \frac{1}{h} (0 \ -1 \ 1), \quad u < 0. \quad (2.2.2)$$

Thus, the use of upwind differences allows convective information only from upstream, and removes the difficult outflow boundary condition's effect on the convective term.

Upwind difference solutions to (2.1.1)–(2.1.3) are shown in Fig. 2.2. The results are better than those for central differences, in that there are no wiggles, but there is still room for improvement.

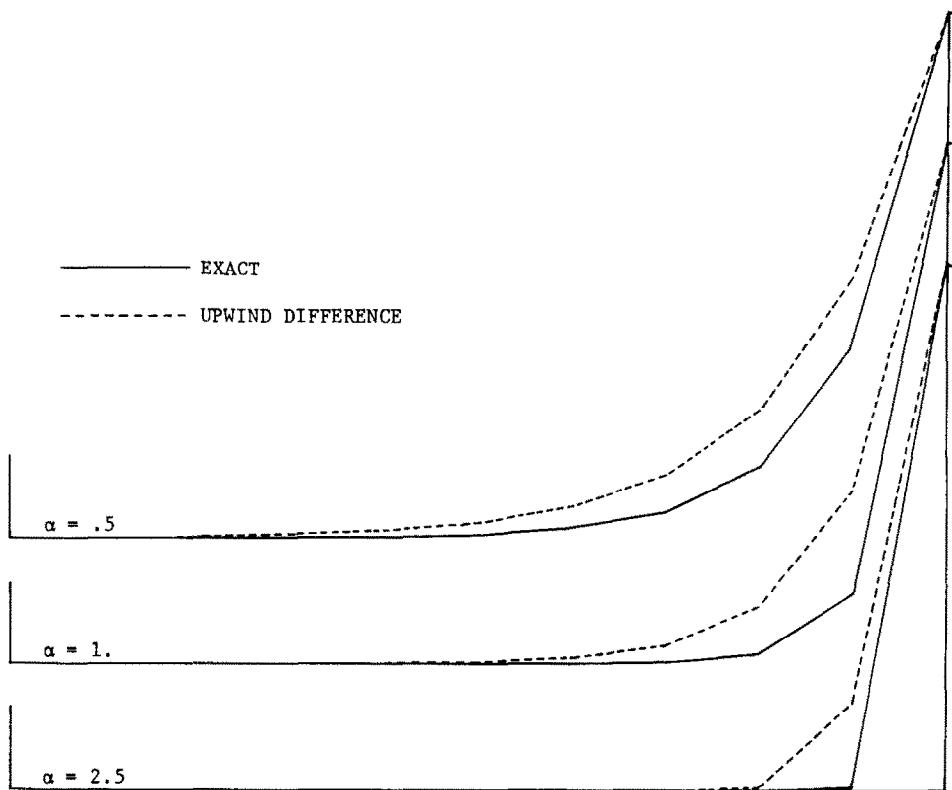


Fig. 2.2. Steady advection-diffusion in one dimension: Upwind differences.

2.3. Artificial diffusion interpretation

It is possible to interpret upwind differences as central differences plus artificial diffusion, \tilde{k} . When $\tilde{k} = uh/2$, the analogy is seen by examination of the appropriate stencils, viz.

$$\begin{aligned}
 & u \frac{1}{2h} (-1 \ 0 \ 1) \quad \text{central,} \\
 & \frac{uh}{2} \frac{1}{h^2} (-1 \ 2 \ -1) \quad \text{second central,} \\
 & + \underline{\qquad\qquad\qquad} \\
 & u \frac{1}{h} (-1 \ 1 \ 0) \quad \text{upwind.}
 \end{aligned}$$

Considerable criticism has been leveled against upwind methods because of the artificial diffusion interpretation. It should be remembered, however, that the artificial diffusion is *relative* to central differences, *not* to the actual physics of the problem. As will be shown subsequently, it is actually more correct to consider central difference methods as having *negative* artificial diffusion.

2.4. Optimal upwind methods

Careful examination of Figs. 2.1 and 2.2 reveals that central difference solutions appear to be underdiffuse, and that upwind solutions appear overdifuse. It is therefore possible to construct a new, optimized solution method as a linear combination of central and upwind differences. Equivalently, the new method may be constructed by adding the proper amount of artificial diffusion to the central difference method. It has been shown [23] that when the artificial diffusion is given by

$$\tilde{k} = (uh/2)\tilde{\xi}, \quad (2.4.1)$$

$$\tilde{\xi} = \coth(\alpha) - 1/\alpha, \quad (2.4.2)$$

$$\alpha = uh/2k, \quad (2.4.3)$$

the solution is nodally exact. It is thus seen that the central-difference method effectively has *negative* artificial diffusion, given by (2.4.1)–(2.4.3), with respect to the exact solution.

Gresho and Lee [13] have suggested that the wiggles often seen in central-difference solutions are a valuable asset, in that they signal that the grid spacing is too coarse to resolve the boundary layer. We believe, however, that it is relatively easy to determine whether boundary layers are adequately resolved, without the aid of wiggles, and thus the use of the central difference method only as a ‘wiggle signal’ represents an unnecessary loss of solution accuracy. Also, in many cases, it is not necessary or desirable to resolve all boundary layers present. The method defined by (2.4.1)–(2.4.3) suggests that when boundary layers are well resolved (i.e., wiggle free central difference), *optimal* upwind methods may be able to give more accurate results than central differences.

2.5. Upwind finite elements

As previously mentioned, conventional Galerkin discretization of (2.1.1) leads to central difference approximations. It is thus not surprising that spurious wiggles often appear in convection dominated Galerkin finite element solutions. Inspired by upwind finite differences, upwind finite elements have been developed to preclude wiggles.

There have been three basic techniques utilized to achieve the upwind effect in finite elements:

(1) *Artificial diffusion*. Artificial diffusion, given by (2.4.1)–(2.4.3) is added to the physical diffusion, and a conventional Galerkin finite element discretization is employed. This is actually a ‘balancing diffusion’, in that it balances the *negative* diffusion of the Galerkin treatment (see also [31]).

(2) *Quadrature*. As proposed by Hughes [19], the numerical quadrature rule for the convection term is modified to achieve the upwind effect. In the one-dimensional case, in which piecewise linear elements are employed, a single quadrature point, ξ , is positioned within the element according to (2.4.2).

(3) *Petrov–Galerkin*. The weighting function for a typical node is modified to weight the element upwind of the node more heavily than the downwind element. An example of such a weighting function is shown in Fig. 2.3. This approach is followed in [5, 17, 22, 35].

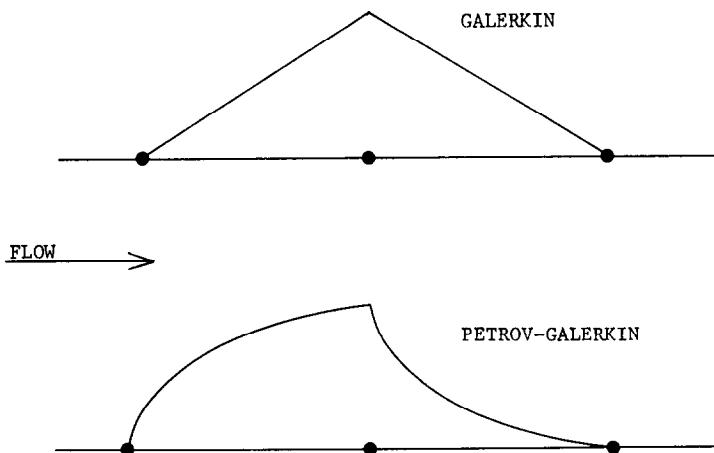


Fig. 2.3. Galerkin weighting function and upwind Petrov–Galerkin weighting function.

All of the above methods, when applied to the example of Section 2.1, give equivalent matrix equations and exact nodal solutions. Important differences in these methods will become apparent in multidimensional and transient cases. Additional information on upwind finite element techniques is contained in [1, 16, 18, 26].

2.6. Shortcomings of early upwind finite elements

The methods described in the preceding section represented a significant improvement over the Galerkin method for the simple model problem described in Section 2.1. Unfortunately, when applied to more complicated situations, generalizations of some of these methods often gave results that were much worse than those obtained by Galerkin's method.

Problems have been noted with the treatment of source terms, time dependent behavior, and with the generalization to multidimensions. In these cases, pronounced diffusion corrupts the true solution. In the last case, this has manifested itself as a so-called spurious crosswind diffusion effect [33, 37, 38, 49]. Solutions showed excessive diffusion perpendicular to the flow when the flow direction was skew to the mesh. These shortcomings also apply to many upwind finite difference methods.

In the following sections, these problems are addressed in detail, and proposed solutions are illustrated via numerical examples.

3. The streamline upwind/Petrov–Galerkin method

3.1. The streamline upwind method

As shown in the previous section, upwind finite elements may be constructed via the Galerkin method with added artificial diffusion. In the one-dimensional case, the artificial diffusion, when optimally selected, balanced the negative diffusion inherent in the Galerkin method, resulting in exact nodal solutions for the model problem. However, the multi-

dimensional generalizations of some finite element (and finite difference) upwind schemes were unsuccessful due to the crosswind diffusion problem (see Fig. 3.1).

It was apparent that the upwind effect, arrived at by whatever means, was needed only in the direction of flow. In [23], the ‘streamline upwind method’ was introduced. This represented the logical application of the artificial diffusion concept to the multidimensional advection-diffusion equation.

In this method, the artificial diffusion operator is constructed to act only in the flow direction, *a priori* eliminating the possibility of any crosswind diffusion. The idea was subsequently described in [31] as ‘anisotropic balancing dissipation’.

The scalar artificial diffusivity, \tilde{k} , used in the one-dimensional case is replaced by an artificial diffusivity tensor \tilde{k}_{ij} to achieve the desired effect. Specifically, let

$$\tilde{k}_{ij} = \tilde{k} \hat{u}_i \hat{u}_j \quad (3.1.1)$$

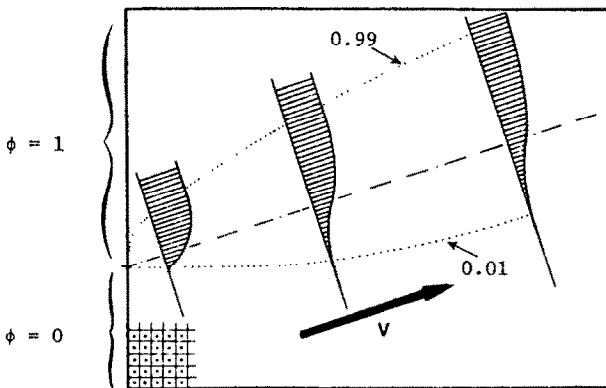
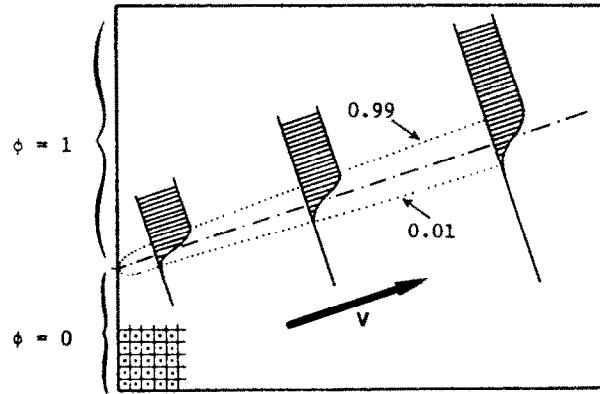


Fig. 3.1. Illustration of the crosswind diffusion problem: $\alpha = 100$ (from ref. [33]).

where

$$\hat{u}_i = u_i / \|u\|, \quad (3.1.2)$$

$$\|u\|^2 = u_i u_i \quad (\text{sum}) \quad (3.1.3)$$

and where \tilde{k} is a scalar artificial diffusivity, and u_i is the flow velocity. Selection of \tilde{k} will be treated in Section 3.3.

Note that (3.1.1) represents a diffusivity acting only in the direction of the flow. For example, if the x_1 coordinate direction is chosen to be aligned with the flow direction, the artificial diffusion matrix defined by (3.1.1) is

$$\tilde{k} = \tilde{k} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad (3.1.4)$$

clearly indicating the absence of crosswind diffusion.

Numerical examples indicating the effectiveness of the streamline upwind method are contained in Section 3.4.

3.2. The streamline upwind/Petrov–Galerkin formulation

3.2.1. Introduction

The streamline upwind method did indeed solve the crosswind problem, but several deficiencies remained. The upwinded convection term was not consistent with the centrally weighted source and transient terms, resulting in excessively diffuse solutions when these terms were present. Clearly, upwind weighting of all terms in the equation was needed. In [23], it was noted that the streamline upwind method could be implemented as a modification of the weighting function for the convection term. Application of this modified weighting function to all terms of the equation gives the desired effect, and defines a consistent Petrov–Galerkin formulation. This technique was introduced in [3] as the ‘streamline upwind/Petrov–Galerkin method’ (SU/PG). The theory of Petrov–Galerkin methods with discontinuous weighting functions (required in the SU/PG method) is set forth in [25].

It will be shown that this consistent weighting lays to rest all negative connotations of artificial diffusion, and indeed the concept of ‘artificial diffusion’ seems to be no longer even applicable.

3.2.2. Preliminaries

Let Ω be a bounded region in $R^{n_{sd}}$, where n_{sd} is the number of space dimensions, and assume Ω has a piecewise smooth boundary Γ . Let $x = \{x_i\}$, $i = 1, 2, \dots, n_{sd}$, denote a general point in $\bar{\Omega}$ and let $n = \{n_i\}$ be the outward normal vector to Γ . Let Γ_s and Γ_k be subsets of Γ which satisfy the following conditions:

$$\overline{\Gamma_s \cup \Gamma_k} = \Gamma, \quad (3.2.1)$$

$$\Gamma_s \cap \Gamma_k = \emptyset. \quad (3.2.2)$$

The superposed bar in (3.2.1) represents set closure and \emptyset , in (3.2.2), denotes the empty set.

Consider a discretization of Ω into element subdomains Ω^e , $e = 1, 2, \dots, n_{\text{el}}$, where n_{el} is the number of elements. Let Γ^e denote the boundary of Ω^e . Furthermore, assume

$$\bigcup_e \overline{\Omega^e} = \Omega, \quad (3.2.3)$$

$$\bigcap_e \Omega^e = \emptyset. \quad (3.2.4)$$

Finally, define the ‘interior boundary’, Γ_{int} , as follows,

$$\Gamma_{\text{int}} = \bigcup_e \Gamma^e - \Gamma. \quad (3.2.5)$$

The summation convention on repeated indices is assumed in force (e.g., if $n_{\text{sd}} = 3$, then $\sigma_i n_i = \sigma_1 n_1 + \sigma_2 n_2 + \sigma_3 n_3$) and a comma is used to denote partial differentiation (e.g. $\varphi_{,i} = \partial \varphi / \partial x_i$). The Kronecker delta is denoted by δ_{ij} ; if $i = j$, then $\delta_{ij} = 1$, otherwise $\delta_{ij} = 0$. The subscript n will be used to denote the component of a vector normal to a boundary, viz.

$$\sigma_n = \sigma_i n_i. \quad (3.2.6)$$

3.2.3. Transient advection-diffusion equation

Consider the unsteady linear advection-diffusion equation for incompressible flow fields:

$$\varphi_{,t} + \sigma_{i,i} = f \quad (3.2.7)$$

where

$$\sigma_i = \sigma_i^a + \sigma_i^d \quad (\text{total flux}), \quad (3.2.8)$$

$$\sigma_i^a = u_i \varphi \quad (\text{advective flux}), \quad (3.2.9)$$

$$\sigma_i^d = -k_{ij} \varphi_{,j} \quad (\text{diffusive flux}). \quad (3.2.10)$$

In the above, f is a source term, u_i is the flow velocity, and k_{ij} is the diffusivity. Each of f , u_i , and k_{ij} are assumed to be given functions of x and t . Furthermore, the velocity field, u_i , is assumed to be divergence free.

An initial-boundary value problem for (3.2.7) consists of finding a function $\varphi(x, t)$ which satisfies (3.2.7) on Ω , and

$$\varphi = g \quad \text{on } \Gamma_s, \quad (3.2.11)$$

$$-\sigma_n^d = h \quad \text{on } \Gamma_a. \quad (3.2.12)$$

$$\varphi(x, 0) = \varphi_0 \quad (3.2.13)$$

where \mathbf{g} and \mathbf{h} are given functions of \mathbf{x} and t , and the initial condition φ_0 is a given function of \mathbf{x} .

REMARK. Eq. (3.2.12) is a diffusive-flux boundary condition. Another possibility is the total-flux boundary condition, viz.

$$-\sigma_n = \mathbf{h} \quad \text{on } \Gamma_h. \quad (3.2.14)$$

It was noted in [25] that the total-flux boundary condition is inappropriate in certain numerical simulations, so only the diffusive-flux boundary condition will be considered here.

3.2.4. Weighted residual formulation

In a usual Galerkin weighted residual method, the weighting functions are considered to be continuous across interelement boundaries. The streamline upwind/Petrov–Galerkin formulation, however, requires discontinuous weighting functions of the form

$$\tilde{w} = w + p \quad (3.2.15)$$

where w is a continuous weighting function, and p is the discontinuous streamline upwind contribution. Both w and p are assumed to be smooth on the element interiors.

Consider a point \mathbf{x} in Γ_{int} . Designate (arbitrarily) one side of Γ_{int} to be the ‘plus side’ and the other to be the ‘minus side’. Let \mathbf{n}^+ and \mathbf{n}^- be unit normal vectors to Γ_{int} at \mathbf{x} which point in the plus and minus directions, respectively. Clearly, $\mathbf{n}^- = -\mathbf{n}^+$. Let σ_i^+ and σ_i^- denote the values of σ_i obtained by approaching \mathbf{x} from the positive and negative sides, respectively. The ‘jump’ in σ_n at \mathbf{x} is defined to be

$$[\sigma_n] = (\sigma_i^+ - \sigma_i^-)\mathbf{n}_i^+ = \sigma_i^+ \mathbf{n}_i^+ + \sigma_i^- \mathbf{n}_i^-. \quad (3.2.16)$$

As may be readily verified from (3.2.16), the jump is invariant with respect to reversing the plus and minus designations.

Throughout, we shall assume that trial solutions, φ , satisfy $\varphi = \mathbf{g}$ on Γ_s and weighting functions, w , satisfy $w = 0$ on Γ_s .

The streamline upwind/Petrov–Galerkin weighted residual formulation for the initial-boundary value problem of Section 3.2.3 is

$$\int_{\Omega} w(\dot{\varphi} + \sigma_{i,i}^a) d\Omega - \int_{\Omega} w_i \sigma_i^d d\Omega + \sum_e \int_{\Omega^e} p(\dot{\varphi} + \sigma_{i,i} - f) d\Omega = \int_{\Omega} w f d\Omega + \int_{\Gamma_h} w h d\Gamma. \quad (3.2.17)$$

Integrating (3.2.17) by parts yields

$$\sum_e \int_{\Omega^e} \tilde{w}(\dot{\varphi} + \sigma_{i,i} - f) d\Omega - \int_{\Gamma_h} w(\sigma_n^d + h) d\Gamma - \int_{\Gamma_{\text{int}}} w [\sigma_n^d] d\Gamma = 0. \quad (3.2.18)$$

From (3.2.18) it is apparent that the Euler–Lagrange equations are (3.2.7) *restricted to the element interiors*, (3.2.12), and the diffusive flux continuity condition across interelement boundaries, viz.

$$[\sigma_n^d] = 0 \quad \text{across } \Gamma_{\text{int}}. \quad (3.2.19)$$

REMARKS. (1) In (3.2.17) it is evident that the streamline upwind modification, p , weights only on the element *interiors*, and therefore does not affect continuity conditions or in the weighting of \hbar .

(2) Assume the following conditions hold:

(i) $k_{ij} = k\delta_{ij}$ (isotropy).

(ii) φ is interpolated with multilinear isoparametric interpolation functions (e.g., the bilinear functions in two dimensions).

(iii) The element domains are rectangular.

The above assumptions imply that on the interior of each element

$$\sigma_{i,i}^d = 0 \Leftrightarrow \sigma_{i,i}^a = \sigma_{i,i}. \quad (3.2.20)$$

In this case, it is seen from (3.2.17) that the streamline upwind modification, p , does not affect the weighting of the diffusion term.

(3) When the element domains are not rectangular, $\varphi_{,ii}$ will not in general vanish identically, and thus $\sigma_{i,i}^d$ may not be zero. However, it is expected that for *reasonable* element shapes, the contribution of $p\sigma_{i,i}^d$ will be small and can be neglected. This is not generally the case for the higher-order elements (see [25] for further discussion).

3.2.5. Streamline upwind/Petrov–Galerkin weighting function

In Section 3.1, the streamline upwind artificial diffusivity tensor was introduced as

$$\tilde{k}_{ij} = \tilde{k}\hat{u}_i\hat{u}_j. \quad (3.2.21)$$

The natural way to implement the streamline upwind method is to modify the diffusion term in the Galerkin variational equation, viz.

$$\int_{\Omega} w_{,i}(\tilde{k}_{ij} + \tilde{k}_{ij})\varphi_{,j} d\Omega. \quad (3.2.22)$$

Alternatively, the same result is achieved if the weighting function for the convective term is modified, viz.

$$\int_{\Omega} \tilde{w}u_i\varphi_{,i} d\Omega \quad (3.2.23)$$

where

$$\tilde{w} = w + \tilde{k}\hat{u}_j w_j / \|u\|. \quad (3.2.24)$$

Specification of the perturbation to the weighting function completes the definition of the streamline upwind/Petrov–Galerkin weighted residual formulation described in the previous

section. From (3.2.24) the perturbation is seen to be

$$p = \tilde{k} \hat{u} w_s / \|u\|. \quad (3.2.25)$$

Clearly, if w is continuous across element boundaries (e.g., typical Galerkin finite element weighting functions), then p , and thus \tilde{w} , will be discontinuous. Examples of w and \tilde{w} for one-dimensional linear finite elements are shown in Fig. 3.2.

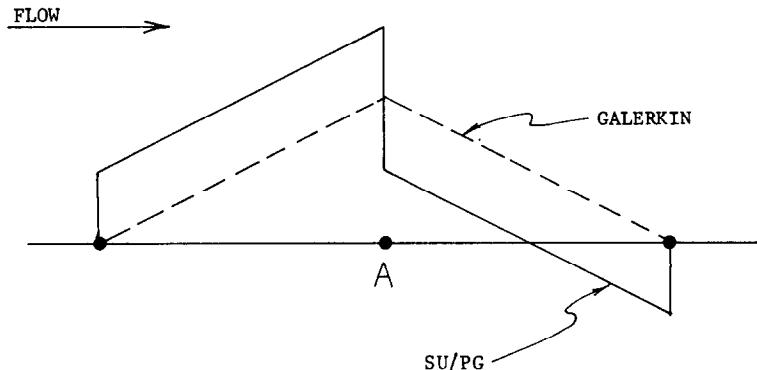


Fig. 3.2. Comparison of streamline upwind/Petrov–Galerkin (SU/PG) and Galerkin weighting functions for node A.

REMARKS. (1) Note that calculation of the streamline upwind weighting functions is quite simple, and requires only the ‘normal’ finite element shape functions, their derivatives, and the velocity field. Higher order functions, such as those employed in [51], are not required.

(2) The streamline upwind/Petrov–Galerkin method generalizes earlier work of Dendy [6], Wahlbin [50], and Raymond and Garder [39]. Baker [2] has also used these ideas in solving advection equations and Burger’s equation.

(3) Some analytical error estimates for the streamline upwind method, in the two-dimensional steady case, have been developed by Johnson and Nävert [30].

3.3. Identification of the upwind parameter

3.3.1. One-dimensional case

In the one-dimensional steady-state case in which u and k are assumed constant, $f = 0$, and linear elements of length h are employed, it can be shown that the parameters \tilde{k} , ξ , and α , given by (2.4.1)–(2.4.3), respectively, are optimal in the sense that they lead to nodally exact solutions. This result has usually been based on analysis of difference equations [19], but recently it has been derived within the finite element context through a calculation of boundary layer dissipation [24].

In an effort to improve computational efficiency, we have often employed the following simplifications of (2.4.2):

Doubly asymptotic approximation.

$$\xi = \begin{cases} \alpha/3, & -3 \leq \alpha \leq 3, \\ \operatorname{sgn} \alpha, & |\alpha| > 3. \end{cases} \quad (3.3.1)$$

Critical approximation [5].

$$\tilde{\xi} = \begin{cases} -1 - 1/\alpha, & \alpha < 1, \\ 0, & -1 \leq \alpha \leq 1, \\ 1 - 1/\alpha, & 1 < \alpha. \end{cases} \quad (3.3.2)$$

Eqs. (2.4.2), (3.3.1) and (3.3.2) are plotted in Fig. 3.3.

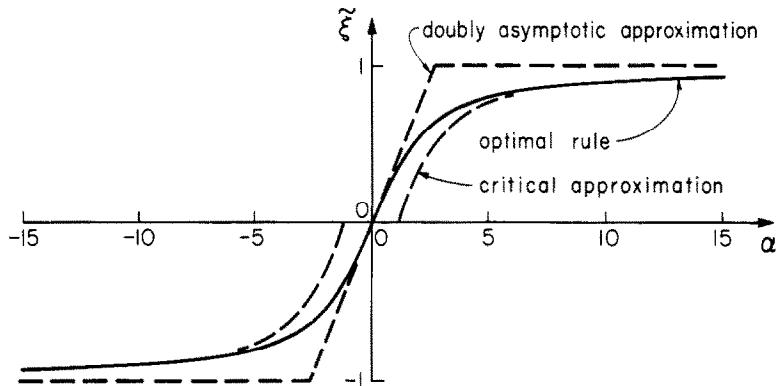


Fig. 3.3. Integration rule for optimal upwind scheme, doubly asymptotic approximation, and critical approximation.

3.3.2. Multidimensional case

In application to multilinear elements, we have employed ad hoc generalizations of (2.4.1)–(2.4.3), (3.3.1), and (3.3.2) based on directional Peclet numbers. For example, in the case of the bilinear isoparametric quadrilateral we have used the following relations:

$$\tilde{k} = (\tilde{\xi} u_\xi h_\xi + \tilde{\eta} u_\eta h_\eta)/2 \quad (3.3.3)$$

where

$$\tilde{\xi} = (\coth \alpha_\xi) - 1/\alpha_\xi, \quad \tilde{\eta} = (\coth \alpha_\eta) - 1/\alpha_\eta, \quad (3.3.4)$$

$$\alpha_\xi = u_\xi h_\xi/2k, \quad \alpha_\eta = u_\eta h_\eta/2k, \quad (3.3.5)$$

$$u_\xi = \mathbf{e}_\xi \cdot \mathbf{u}, \quad u_\eta = \mathbf{e}_\eta \cdot \mathbf{u}, \quad (3.3.6)$$

in which the unit vectors \mathbf{e}_ξ and \mathbf{e}_η , and element characteristic lengths h_ξ and h_η are defined in Fig. 3.4, and \mathbf{u} and k are evaluated at the element center.

3.3.3. Transient case

In the transient case, \tilde{k} may be selected to maximize accuracy near boundary layers as in the steady case, or to maximize phase accuracy. Raymond and Garder [39] have shown for a one-dimensional, pure advection ($k = 0$) model problem that phase errors are minimized if

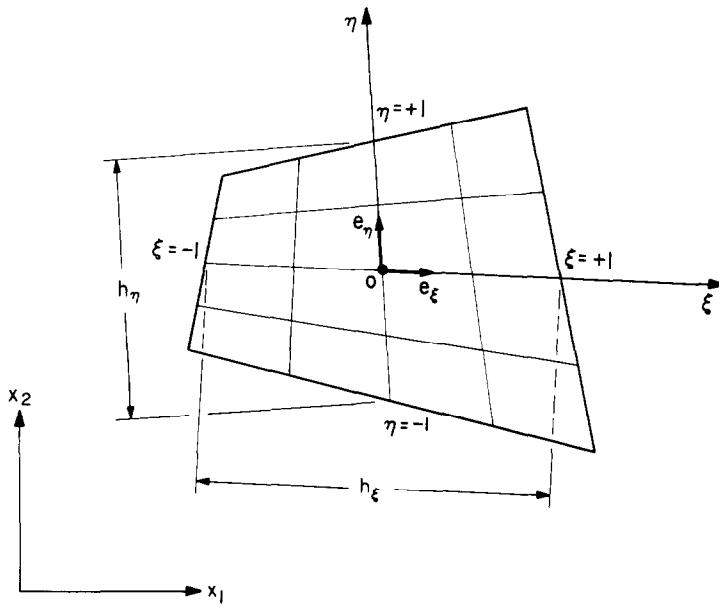


Fig. 3.4. Typical four-node quadrilateral finite element geometry.

$$\bar{k} = |u| h / \sqrt{15}. \quad (3.3.10)$$

In the multilinear case when $k \neq 0$ we have employed the following generalization of (3.3.10):

$$\bar{k} = (\tilde{\xi} u_\xi h_\xi + \tilde{\eta} u_\eta h_\eta) / \sqrt{15}. \quad (3.3.11)$$

3.3.4. Remark

The choice of \bar{k} for multidimensional and transient cases, as outlined above, may seem rather ad hoc. It is expected that further research may yield a more rigorous approach to these cases. However, experience with many numerical examples has indicated that the structure of the streamline upwind weighting function is far more important than the precise value of \bar{k} .

3.4. Numerical examples

3.4.1. Introduction

The following methods were investigated:

Galerkin (G). In this case standard procedures were used on the variational eq. (3.2.17), with no modification to the weighting function w (i.e., $p = 0$). Two-by-two Gaussian quadrature was employed to integrate all elemental contributions.

Quadrature upwind (QU). The quadrature upwind scheme is described in Section 2.5 for one dimension and in [19] for multidimensions. It exhibits many of the shortcomings (including excessive crosswind diffusion) of the classical upwind difference schemes. A special quadrature rule is used on the advection term to achieve the upwind effect. The two-by-two Gaussian rule is used on all other terms.

Streamline upwind/Petrov–Galerkin (SU1, SU2). The streamline upwind/Petrov–Galerkin

method is based on the variational eq. (3.2.17) with the weighting function modification, p , defined by (3.2.25). In steady examples, \bar{k} was selected according to (3.3.6), and according to (3.3.11) in the transient case. In the case of SU1, one point Gaussian quadrature is used on the advection term, whereas for SU2, two-by-two quadrature is employed. All other element integrals are calculated with the two-by-two rule.

Exact (E). The results are compared with exact solutions when available. In elevation plots, the 'exact' results are the nodal interpolates of the exact solution via the piecewise bilinear finite element interpolation functions.

All computations in this section were performed in double precision (64 bits per word) on the California Institute of Technology IBM 3032 computer. Linear elements were used for one-dimensional problems, and bilinear isoparametric quadrilaterals were used for two-dimensional problems. The trapezoidal rule was employed in all transient cases.

3.4.2. Streamline upwind examples

In this section, examples demonstrate the effectiveness of the streamline upwind method in precluding both 'wiggles' and the spurious crosswind diffusion that afflicts classical upwind schemes.

Advection skew to the mesh. The problem statement is depicted in Fig. 3.5. The flow is

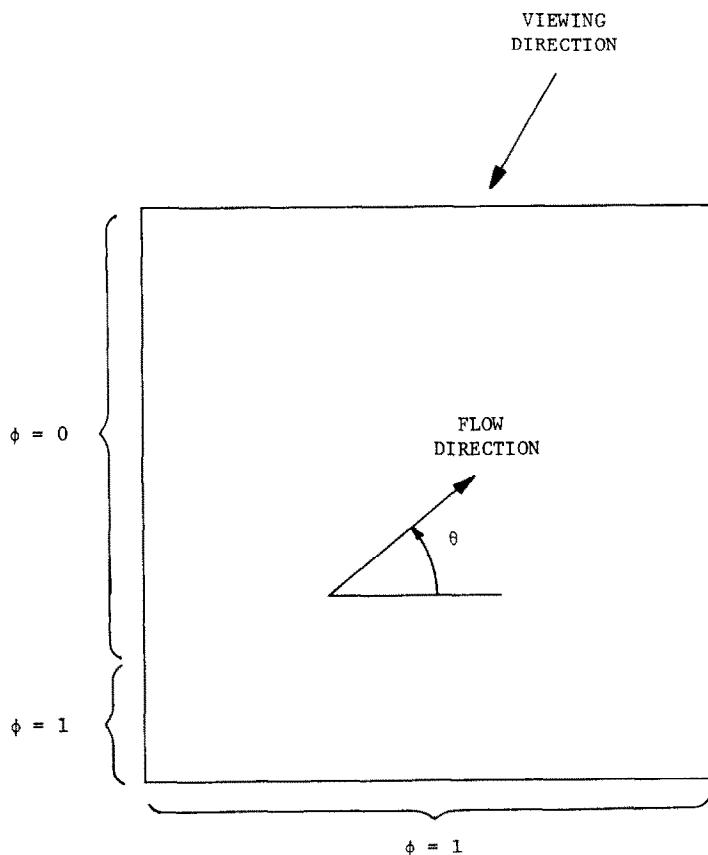


Fig. 3.5. Advection skew to the mesh: Problem statement.

unidirectional, constant ($\|u\| = 1$), and skew to the mesh. The diffusivity coefficient was taken to be 10^{-6} resulting in a Peclet number of 10^6 . In all cases, a ten-by-ten mesh of equal sized square elements was employed. The inflow boundary condition is discontinuous, as shown, and two different outflow boundary conditions were considered:

Homogeneous natural boundary condition (i.e., $\varphi_n = 0$). For this case, by virtue of the magnitude of the Peclet number, the solution is essentially one of pure advection. The ‘exact’ solution is then simply an advection of the inflow boundary condition in the flow direction. The results shown in Fig. 3.6 clearly demonstrate the presence of spurious crosswind diffusion for the quadrature upwind scheme. The Galerkin and streamline upwind schemes are significantly better in this respect. In general the Galerkin scheme and SU2 are somewhat superior to SU1. The exception is the case $\theta = 45^\circ$ in which SU1 is nodally exact.

In passing we note that when $\theta = 0^\circ$ all cases considered are nodally exact and thus are not shown.

Homogeneous essential boundary condition (i.e., $\varphi = 0$). In this case the solution is identical to the previous one, except in a small neighborhood of the downwind boundary where a very thin ‘boundary layer’ forms. It is a very difficult task for such a crude mesh to capture the essential features of the exact solution under these circumstances. Elevations of φ are shown in Fig. 3.7. No results are shown for the Galerkin scheme since they are wildly oscillatory and bear no resemblance to the exact results. Again the spurious crosswind diffusion of the quadrature upwind scheme is in evidence. Of the streamline upwind schemes, SU2 may be seen to be superior to SU1 for this case.

All schemes considered, except the Galerkin formulation, are nodally exact for the case $\theta = 0^\circ$ and thus are not shown.

Advection of a cosine hill in a rotating flow field. The problem statement is shown in Fig. 3.8. The flow is rigid rotation about the center of the bi-unit square domain, with velocity components given by

$$u_1 = -x_2, \quad u_2 = x_1.$$

The problem is advection dominated, with diffusivity of 10^{-6} . Along the external boundary φ is set to zero, and on the internal ‘boundary’ OA, φ is prescribed to be a cosine hill. A 30-by-30 finite element mesh was employed. The exact solution is essentially a pure advection of the OA boundary condition along the circular streamlines. Elevations of φ are shown in Fig. 3.9.

For this problem, Galerkin and streamline upwind schemes produce very good results. The streamline schemes are somewhat better than the Galerkin scheme due to the small-amplitude oscillations of the latter. The quadrature upwind scheme is very poor due to the pronounced crosswind diffusion effect.

3.4.3. Model problems for the Petrov–Galerkin method

In this section, simple model problems are presented to demonstrate the need for consistent Petrov–Galerkin weighting of all terms.

Pure advection with a source term. Leonard [33] has shown that some upwind formulations give very poor results when a source term, f , is present. The test case used in [33] to illustrate the problem is shown here in Fig. 3.10. The following methods are compared: SU2 and classical upwind differences.

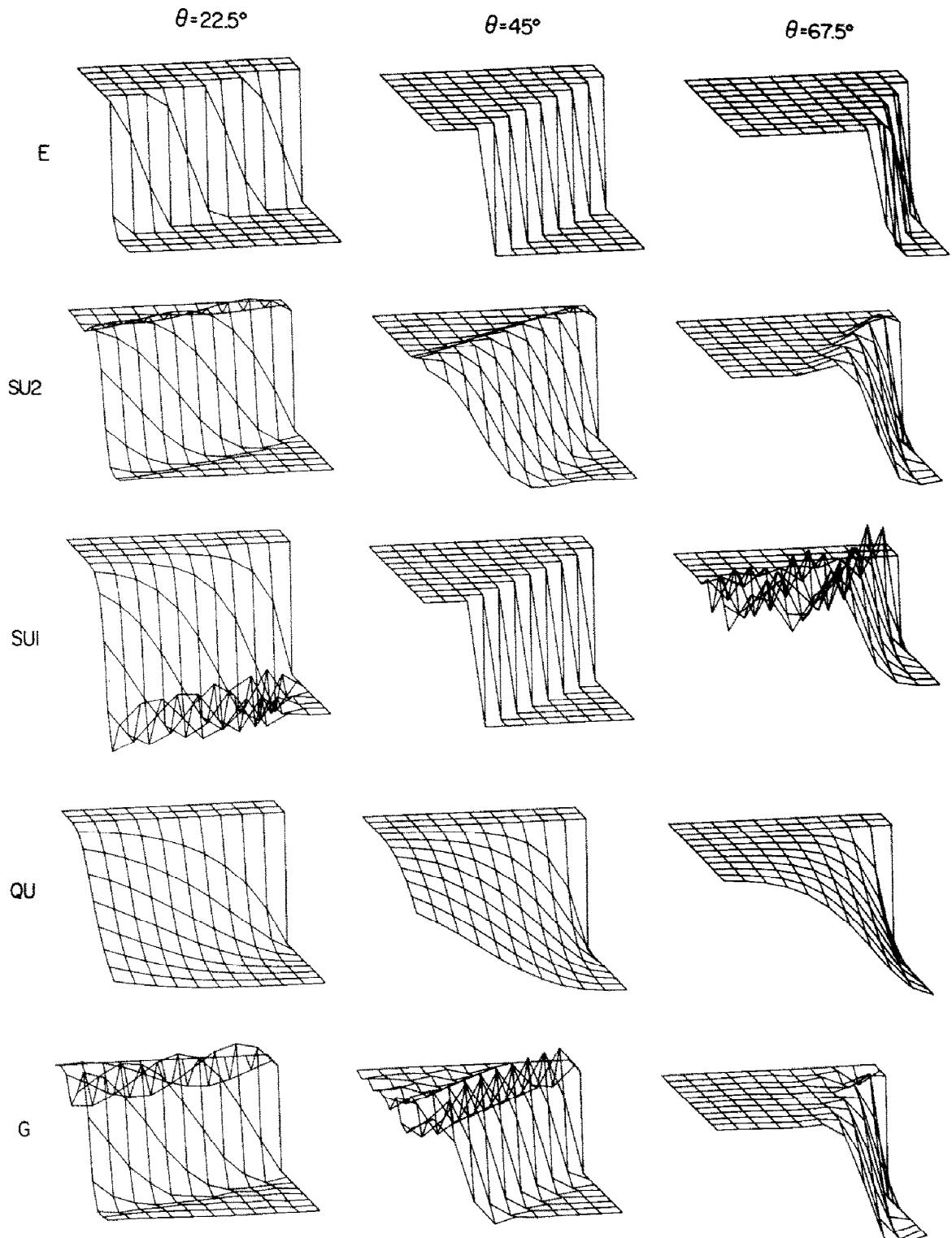


Fig. 3.6. Advection skew to the mesh with homogeneous natural outflow boundary condition: Elevations of ϕ .

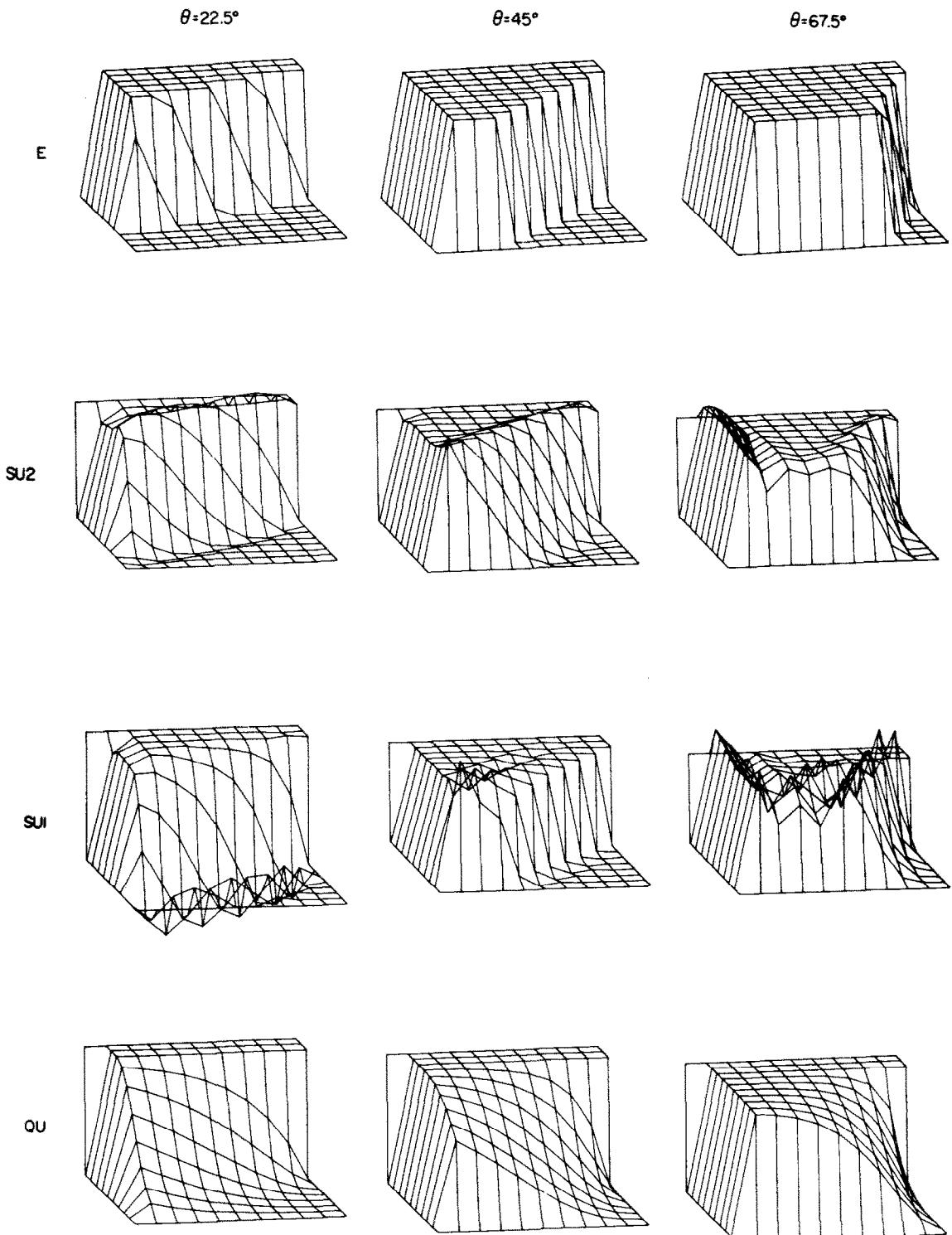


Fig. 3.7. Advection skew to the mesh with homogeneous essential outflow boundary bondition: Elevations of ϕ .

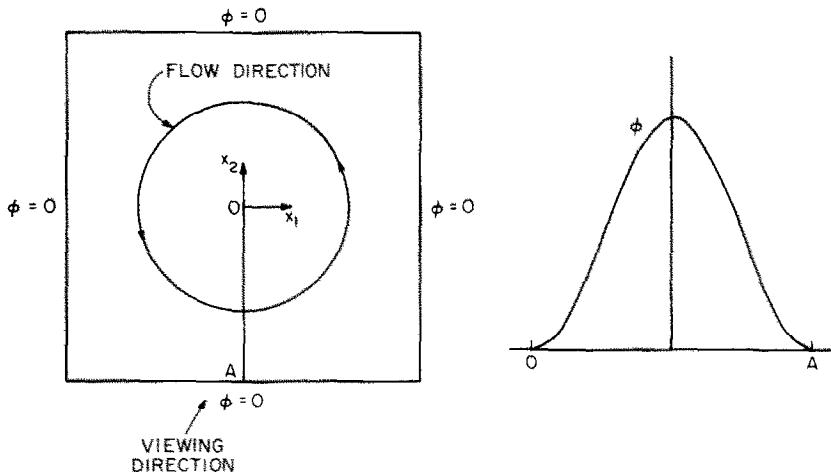
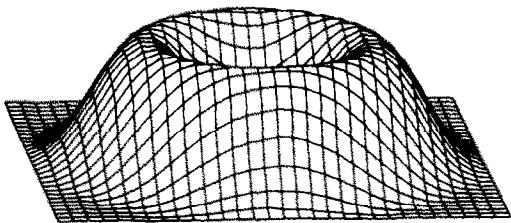
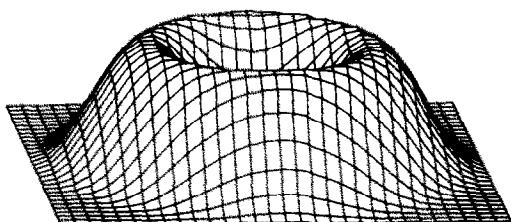


Fig. 3.8. Advection in a rotating flow field: Problem statement.

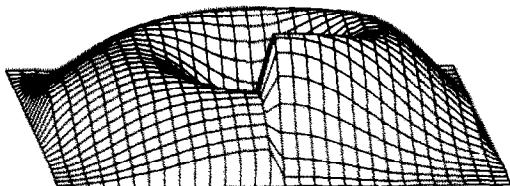
SU2



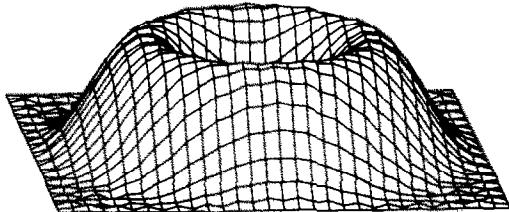
SUII



QU



G

Fig. 3.9. Advection in a rotating flow field: Elevations of ϕ .

The results in Fig. 3.11 show poor results for the classical upwind method. The upwind result is actually a very accurate solution to a diffuse problem with element Peclet number, α , equal to one. This is not surprising, as an artificial diffusivity, \bar{k} , of $uh/2$ was added, and recall that $\alpha = uh/2k$. The Petrov-Galerkin results (SU2), on the other hand, are very good, being nearly exact. It is seen that the concept of artificial diffusion is clearly not applicable when using consistent Petrov-Galerkin weighting.

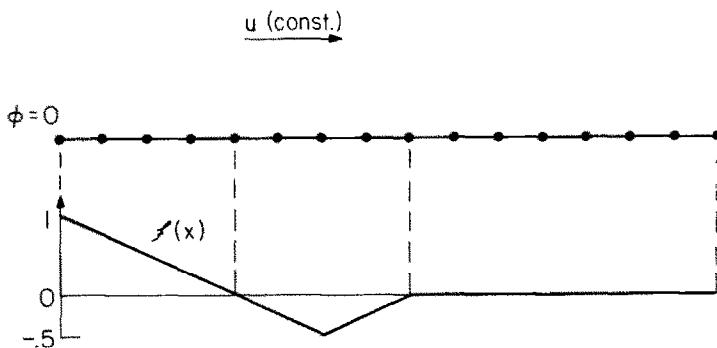


Fig. 3.10. Pure advection with a source term: Problem statement.

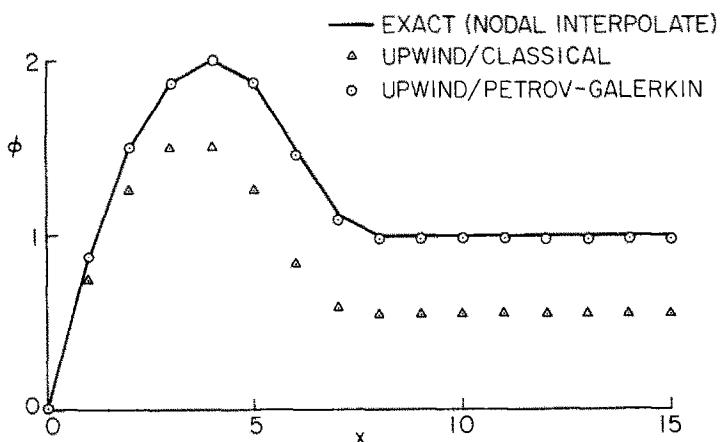


Fig. 3.11. Pure advection with a source term: Comparison of results.

Transient advection. The problem is a simple one-dimensional advection of a cosine hill. The diffusion, k , is zero, and the Courant number ($Cr = u\Delta t/h$) is $1/2$. The exact solution is just a pure advection of the initial condition to the right. The methods compared are those of the previous example. Results in Fig. 3.12 again show that artificial diffusion renders the classical upwind solution virtually useless, while the consistently weighted Petrov-Galerkin solution is very good, and is not subject to the artificial diffusion criticism.

3.4.4. Streamline upwind/Petrov-Galerkin transient multidimensional examples

The rotating cone problem [9, 11] has emerged as one of the standard test problems for advection algorithms. The problem statement is shown in Fig. 3.13. A mesh of 30-by-30 bilinear elements was employed. The Courant number at the peak of the cone was approximately $1/4$, corresponding to a full 360° rotation of the cone in 200 time steps.

The exact solution consists of a rigid rotation of the cone about the center of the mesh. Essential features of the numerical solutions are phase error, seen as spurious leading and trailing waves, and dissipation error, seen as a reduction in cone height.

The following methods were compared: G, SU2, and SU2 with inconsistent treatment of the

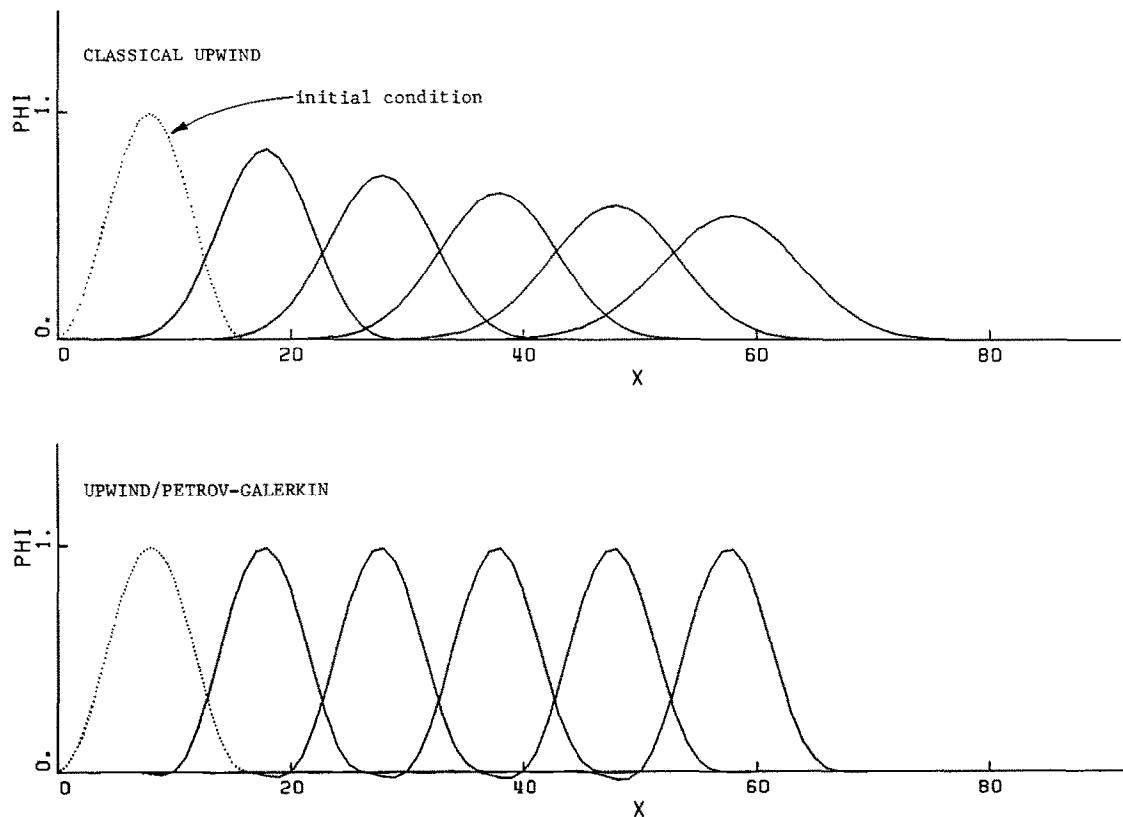


Fig. 3.12. Transient pure advection: Comparison of results at steps 20, 40, 60, 80, and 100.

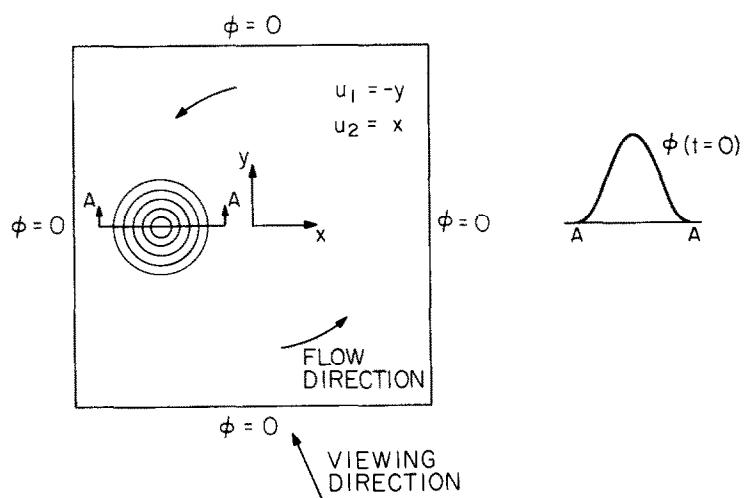


Fig. 3.13. Rotating cone: Problem statement.

time dependent term. That is, the streamline upwind method is implemented by adding artificial diffusion, instead of the consistent Petrov-Galerkin weighting of SU2.

Results for a full rotation are shown in Fig. 3.14. The Galerkin method exhibits no dissipation error, but trailing waves of about 5% of the original cone height indicate phase error. The inconsistent streamline method shows excessive damping of the cone in the direction of its travel. The streamline-upwind/Petrov-Galerkin results show much smaller phase error than Galerkin, but the cone is reduced to 88% of its original height.

This example once again clearly demonstrates that upwind finite element schemes should be developed within a consistent Petrov-Galerkin formulation.

Cone impinging on a mesh boundary. In environmental transport problems it is often necessary for cones or ‘puffs’ to leave the mesh as they are transported downstream. In this situation Galerkin methods often produce oscillations in the vicinity of the outflow boundaries.

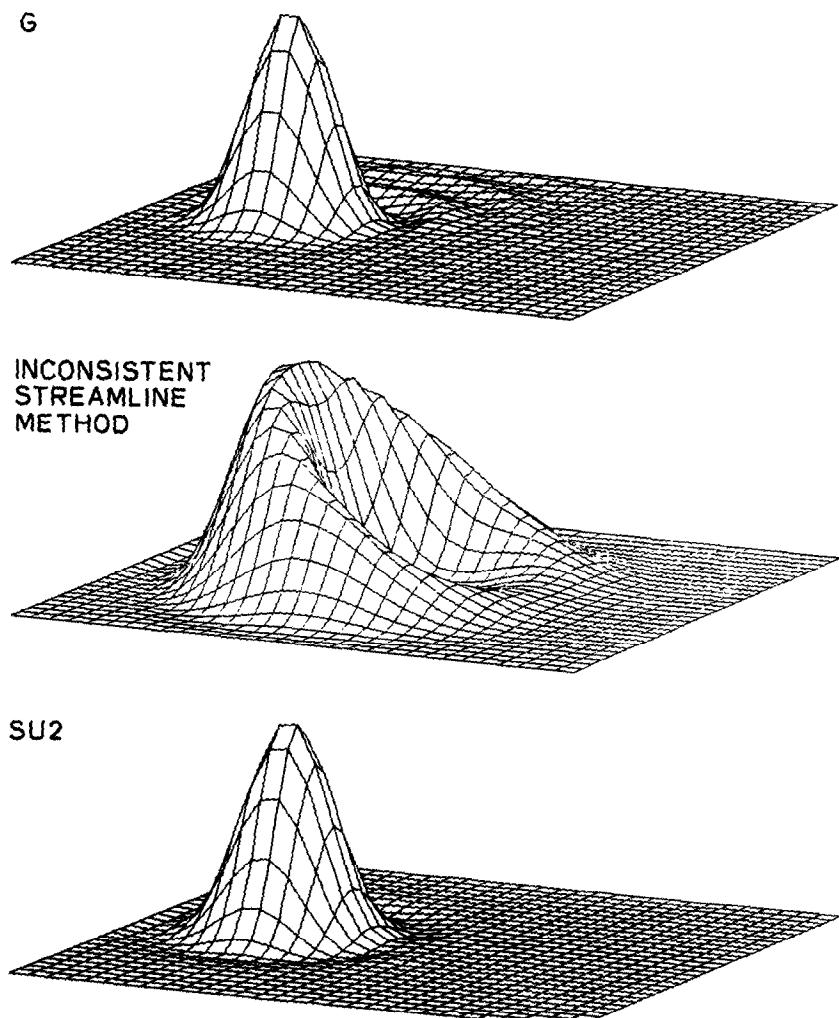


Fig. 3.14. Rotating cone: Elevations of ϕ after one rotation (200 time steps).

Such oscillations can be especially deleterious when the quantity being convected is coupled with sensitive chemical rate equations. To assess the performance of the streamline upwind/Petrov-Galerkin method in this case, tests were run using the mesh and initial conditions from the previous examples with a unidirectional velocity field; $u_1 = -1$, $u_2 = 0$. In the Galerkin method oscillations of about 3% appear as the cone leaves the mesh (Fig. 3.15). Due to the nondissipative nature of the system, these oscillations remain long after the cone has left. With the Petrov-Galerkin method, the cone leaves the mesh with a minimum of spurious oscillations. When the calculation was stopped after 100 time steps the largest values of φ were about 10^{-4} percent of the original cone height.

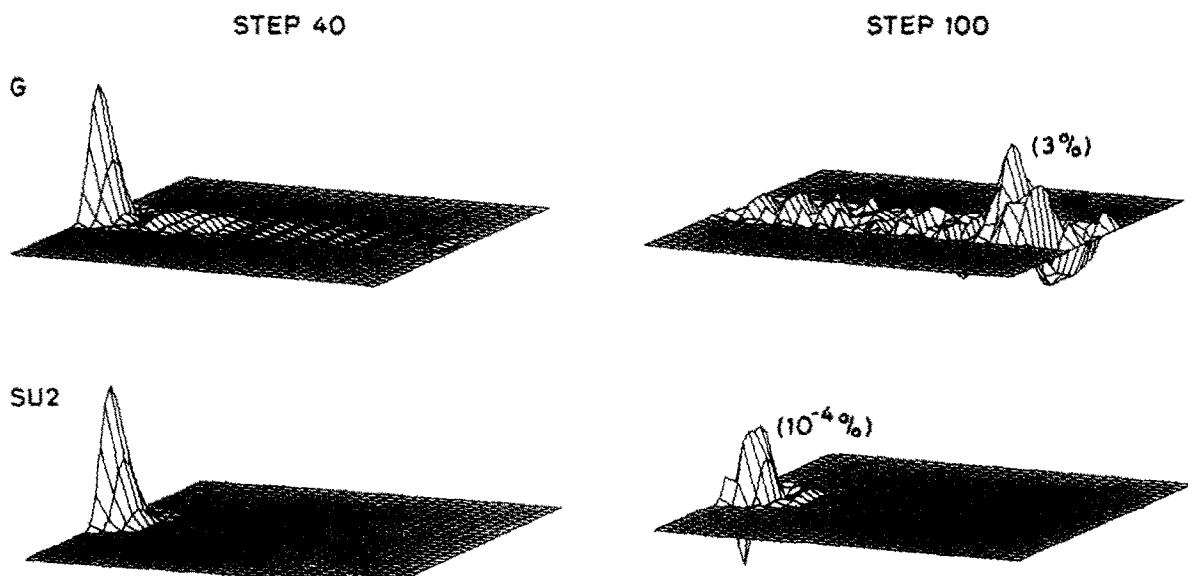


Fig. 3.15. Cone impinging on a natural-boundary-condition boundary. Amplitude of residual values, as a percentage of initial cone height, shown in parentheses.

4. Implicit-explicit transient algorithms: Applications to the streamline upwind method

4.1. Introduction

In this section, we consider a particular type of transient algorithm which is well suited to convective transport equations. Examples in the previous chapter were calculated with trapezoidal rule, a well known *implicit* algorithm. Implicit methods are generally unconditionally stable, but require solution of matrix equations at every time step. If the equations are linear, the stiffness matrix can be formed and factored in advance, leaving a (relatively fast) forward reduction and backsubstitution at each time step. If, on the other hand, the equations are nonlinear (e.g., Navier-Stokes equations) the matrix must be reformed and factored at least once every time step. In this situation an *explicit* method is often more efficient. Explicit methods are always conditionally stable, but require no solution of matrix equations. In explicit algorithms, the mass matrix is diagonalized, or ‘lumped’.

Unfortunately, mass lumping often seriously degrades phase accuracy in fluid mechanics (see e.g., Gresho et al. [11, 14]).

It will be shown that the predictor-multicorrector algorithm presented herein possesses the advantageous structure of an explicit algorithm, while maintaining the good phase accuracy associated with a consistent mass matrix. We were led to such a procedure by the success of the formulation of Donea et al. [9]. The streamline upwind/Petrov–Galerkin method interfaces naturally with the proposed algorithm and engenders time step restrictions which are much more favorable than those of typical Galerkin formulations (see e.g. [10]).

4.2. Predictor-multicorrector algorithm

Consistent finite element spatial discretization of the variational equation (3.2.17) leads to the following semi-discrete system of equations,

$$\mathbf{M}\dot{\boldsymbol{\varphi}} + \mathbf{C}\boldsymbol{\varphi} = \mathbf{F} \quad (4.2.1)$$

where \mathbf{M} , \mathbf{C} , and \mathbf{F} represent the ‘mass’ matrix, diffusion and convection matrix, and generalized force vector, respectively, and $\boldsymbol{\varphi}$ and $\dot{\boldsymbol{\varphi}}$ represent vectors of nodal values of φ and $\varphi_{,\alpha}$, respectively.

Note that the streamline upwind/Petrov–Galerkin formulation leads to a *nonsymmetric* consistent mass matrix.

The initial-value problem consists of finding a function $\boldsymbol{\varphi} = \boldsymbol{\varphi}(t)$ satisfying (4.2.1) and the initial condition

$$\boldsymbol{\varphi}(0) = \boldsymbol{\varphi}_0 \quad (4.2.2)$$

where $\boldsymbol{\varphi}_0$ is given.

The algorithm used to solve (4.2.1) is a version of the predictor-multicorrector algorithms proposed by Hughes [20, 21, 27]. The solution procedure is as follows:

(1) *Predictors.*

$$\boldsymbol{\varphi}_{n+1}^{(0)} = \boldsymbol{\varphi}_n + \Delta t(1 - \gamma)\dot{\boldsymbol{\varphi}}_n, \quad (4.2.3)$$

$$\dot{\boldsymbol{\varphi}}_{n+1}^{(0)} = \mathbf{0}. \quad (4.2.4)$$

Set $i = 0$.

(2) *Solution.*

$$\mathbf{M}^* \Delta \dot{\boldsymbol{\varphi}}_{n+1}^{(i)} = \mathbf{R}_{n+1}^{(i)}, \quad (4.2.5)$$

$$\mathbf{R}_{n+1}^{(i)} = \mathbf{A}(\mathbf{f}_{n+1}^e - m^e \dot{\boldsymbol{\varphi}}_{n+1}^{e(i)} - c^e \boldsymbol{\varphi}_{n+1}^{e(i)}) \quad (\text{residual force}). \quad (4.2.6)$$

(3) *Correctors.*

$$\boldsymbol{\varphi}_{n+1}^{(i+1)} = \boldsymbol{\varphi}_{n+1}^{(i)} + \Delta t\gamma \Delta \dot{\boldsymbol{\varphi}}_{n+1}^{(i)}, \quad (4.2.7)$$

$$\dot{\boldsymbol{\varphi}}_{n+1}^{(i+1)} = \dot{\boldsymbol{\varphi}}_{n+1}^{(i)} + \Delta \dot{\boldsymbol{\varphi}}_{n+1}^{(i)}. \quad (4.2.8)$$

(4) If an additional iteration is required, increment i , and go to step 2. If no additional iteration, increment n , and go to step 1.

The notation is as follows: Δt is the time step; $\boldsymbol{\varphi}_n$ and $\dot{\boldsymbol{\varphi}}_n$ are the approximations to $\boldsymbol{\varphi}(t_n)$ and $\dot{\boldsymbol{\varphi}}(t_n)$, respectively; the superscript i is the iteration counter; the superscript e denotes an element level matrix or vector; and γ is a parameter governing the approximation of $\dot{\boldsymbol{\varphi}}$. The symbol \mathbf{A} represents an assembly operator which adds the element level force vectors to the global force vector. The elements vectors $\boldsymbol{\varphi}^e$ and $\dot{\boldsymbol{\varphi}}^e$ include the prescribed boundary conditions in addition to the local values of the global $\boldsymbol{\varphi}$ and $\dot{\boldsymbol{\varphi}}$ vectors. The vector \mathbf{f}^e represents the consistent nodal forces arising from the source term \mathbf{f} , and the boundary condition on Γ_k . $\dot{\boldsymbol{\varphi}}_0$ may be computed via (4.2.1) or read in as input data. (In the problems of the previous section, $\dot{\boldsymbol{\varphi}}_0$ was computed via (4.2.1).)

This algorithm is particularly powerful due to the residual (or ‘out-of-balance-force’) formulation, and the ability to make multiple iterations within a time step. Additionally, the option of implicit or explicit treatment of specific terms is naturally contained within the algorithmic structure.

Choice of \mathbf{M}^ .* Hughes [20] showed that the definition of the generalized mass matrix, \mathbf{M}^* in (4.2.5), determined what parts of the equation system were treated implicitly or explicitly. Specifically, terms to be treated implicitly are included in the definition of \mathbf{M}^* , and terms to be treated explicitly are simply left out.

Examples of \mathbf{M}^ .* (1) If we wish to treat all terms implicitly, \mathbf{M}^* is defined as follows,

$$\mathbf{M}^* = \mathbf{M} + \gamma \Delta t \mathbf{C}. \quad (4.2.9)$$

With $\gamma = 1/2$, this results in the trapezoidal rule, which was employed in the time-dependent calculations of Section 3.

(2) It is possible to split the operators. For example, the streamline upwind contributions may be treated explicitly,

$$\mathbf{M}^* = \mathbf{M}^g + \gamma \Delta t \mathbf{C}^g \quad (4.2.10)$$

where \mathbf{M}^g and \mathbf{C}^g denote the Galerkin contributions to \mathbf{M} and \mathbf{C} , respectively. In this case, \mathbf{M}^* is symmetric.

(3) To treat everything explicitly, set

$$\mathbf{M}^* = \mathbf{M}^L \quad (4.2.11)$$

where \mathbf{M}^L denotes a lumped mass matrix (see e.g., [26]).

The fully explicit algorithm defined by (4.2.3)–(4.2.8) and (4.2.11) is a model for the Navier–Stokes algorithm outlined in Section 5, and is the subject of the stability and accuracy analysis in the following section.

REMARKS. (1) Normally γ is set equal to $1/2$, resulting in the trapezoidal rule for implicitly treated terms. Note, however, that the first pass through (4.2.3)–(4.2.8) results in forward difference approximations for explicitly treated terms, *regardless of the value of γ* .

(2) Element contributions to the residual force are calculated directly as vectors, and

assembled to the global residual vector (4.2.6). Thus, when \mathbf{M}^* is diagonal, no matrix calculations at all are required.

(3) For explicit Petrov–Galerkin formulations it is necessary to perform at least one additional pass on (4.2.5)–(4.2.8) in order to include the effects of the Petrov–Galerkin mass matrix. This arises because the predicted value of ϕ is zero on the first pass (see (4.2.4)).

(4) The residual force, \mathbf{R} , is seen to be a measure of how close the solution is to the consistent-mass implicit solution. Thus as the residual force converges to zero during successive iterations, the solution converges to the implicit solution. Numerical experience has shown that one additional iteration is usually sufficient to achieve the implicit character.

(5) When \mathbf{M}^* is diagonal, and one additional pass is performed on (4.2.5)–(4.2.8), the current algorithm has essential features in common with the two-stage explicit technique proposed by Donea et al. [9].

4.3. Stability and accuracy

4.3.1. Introduction

In this section we examine the stability and accuracy properties of the explicit multicorrector ($\mathbf{M}^* = \mathbf{M}^L$) algorithm described in the previous section. It is assumed that $\gamma = 1/2$.

The results which follow hold for the one-dimensional constant coefficient advection diffusion equation

$$\dot{\phi} + u\phi_{,x} = k\phi_{,xx}. \quad (4.3.1)$$

4.3.2. Stability

For linear finite element discretization of (4.3.1) on a uniform grid with element length h , and no additional iterations on (4.2.5)–(4.2.8), the following stability conditions hold:

(1) *Diffusive limit* [10, 41].

(a) Galerkin,

$$Cr \leq \alpha, \quad (4.3.2)$$

(b) SU/PG ($\tilde{k} = (uh/2)\tilde{\xi}$),

$$Cr \leq \alpha_{\text{eff}} \stackrel{\text{def}}{=} \frac{1}{2}uh/(k + \tilde{k}). \quad (4.3.3)$$

(2) *Advection-diffusive limit*.

(a) Galerkin [10, 34],

$$Cr \leq 1/\alpha, \quad (4.3.4)$$

(b) SU/PG ($\tilde{k} = (uh/2)\tilde{\xi}$) [26],

$$Cr \leq 1, \quad (4.3.5)$$

where Cr is the Courant number ($Cr = \Delta t u/h$), and α is the element Peclet number ($\alpha = uh/2k$). These stability limits are depicted graphically in Fig. 4.1.

REMARKS. (1) The stability of the SU/PG algorithm is always governed by the diffusive limit. As seen in Fig. 4.1, the critical Courant number approaches unity asymptotically as the element Peclet number approaches infinity. The Galerkin stability limit, unlike the SU/PG limit, is seen to be severely restrictive for large element Peclet numbers. It is also seen from (4.3.4) that the Galerkin method is *unconditionally unstable* in the case of pure advection.

(2) When one additional iteration is performed on (4.2.5)–(4.2.8), numerical experience has indicated that the above stability conditions are sufficient.

(3) In the two-dimensional case, conditions (4.3.2)–(4.3.4) hold when the Courant and Peclet numbers are defined as follows [10].¹

$$Cr = \Delta t \sqrt{(\|\mathbf{u}\|/h_\xi)^2 + (\|\mathbf{u}\|/h_\eta)^2}, \quad (4.3.6)$$

$$\alpha = \|\mathbf{u}\|/(2k\sqrt{1/h_\xi^2 + 1/h_\eta^2}). \quad (4.3.7)$$

Analogous definitions can be made for the three-dimensional case.

(4) The stability conditions for the streamline upwind/Petrov–Galerkin method with $\tilde{k} = uh/\sqrt{15\xi}$ are more complicated than for the other cases. For one pass through the algorithm the conditions are based on the effective Peclet number:

Diffuse limit.

$$Cr \leq \alpha_{\text{eff}},$$

Advective-diffusive limit.

$$Cr \leq 1/\alpha_{\text{eff}}.$$

In the case of two passes, the effective Peclet number concept is no longer valid, and exact expressions are not available. The following approximate conditions were determined

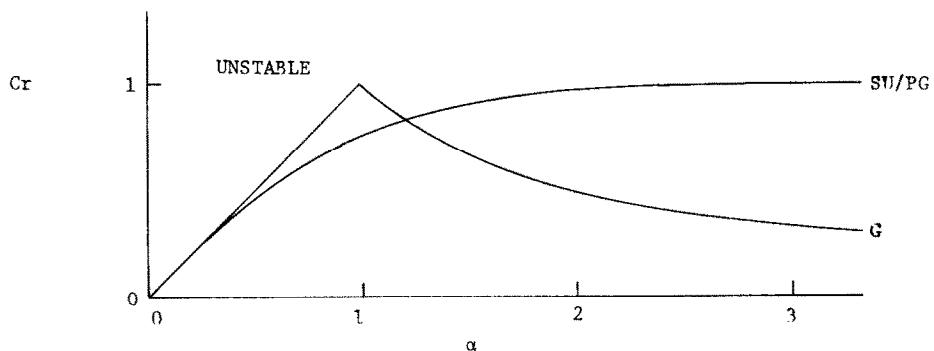


Fig. 4.1. Stability limits for Galerkin (G) and streamline upwind/Petrov–Galerkin (SU/PG) explicit algorithms.

¹These results hold for the standard central-difference stencils. We conjecture that finite-element equations may yield somewhat more generous restrictions.

graphically from the results of Tezduyar [48]:

$$\text{Cr} \leq 0.8, \quad \text{if } \alpha \geq 100,$$

$$\text{Cr} \leq \min(1, \alpha), \quad \text{if } \alpha < 100.$$

These approximate conditions have proven to be satisfactory in the example problem of Section 6.

4.3.3. Accuracy

A detailed analysis of phase accuracy and damping for the algorithms under consideration has been performed by Tezduyar [48]. Samples of these results are reproduced here to demonstrate the superiority of the streamline upwind/Petrov-Galerkin method over the Galerkin method.

In the analysis of [48], exact solutions to (4.3.1) are assumed of the form

$$\varphi(x, t) = e^{-(\xi+i\omega)t} e^{iKx}, \quad (4.3.8)$$

and approximate linear finite element solutions are of the form,

$$\varphi(x_A, t_n) = e^{-(\bar{\xi}+i\bar{\omega})n\Delta t} e^{iKx_A} \quad (4.3.9)$$

where ξ and $\bar{\xi}$ represent exact and approximate damping, respectively;² ω and $\bar{\omega}$ represent exact and approximate frequencies, respectively; and K denotes the wave number ($K = 2\pi/\lambda$ where λ is the wavelength).

An important parameter in determining the accuracy of the algorithm is the dimensionless wave number Kh . This number gives a measure of how many elements are used to resolve one wavelength,

$$\lambda/h = 2\pi/Kh. \quad (4.3.10)$$

For good resolution it is felt that a minimum of ten elements per wavelength are required, or $Kh \approx 0.6$. The maximum possible value for Kh is π , corresponding to node-to-node oscillations of φ .

The results to be discussed shortly are for a typical Courant number of 0.8, and for three values of the upwind parameter, \bar{k} , viz.:

- (1) $\bar{k} = (uh/2)\bar{\xi}$ (steady state optimization),
- (2) $\bar{k} = (uh/\sqrt{15})\bar{\xi}$ (phase optimization),
- (3) $\bar{k} = 0$ (Galerkin).

Additionally, it is assumed that the algorithm is explicit ($\mathbf{M}^* = \mathbf{M}^L$) unless otherwise noted.

Damping error—Pure advection. In the case of pure advection ($k = 0$), the numerical solution should exhibit a minimum amount of damping. A measure of this damping is the algorithmic damping ratio, $\bar{\xi}/\bar{\omega}$, which is related to the amplitude decay over one wavelength

²The damping parameters $\xi, \bar{\xi}$ should not be confused with the optimal quadrature point $\bar{\xi}$.

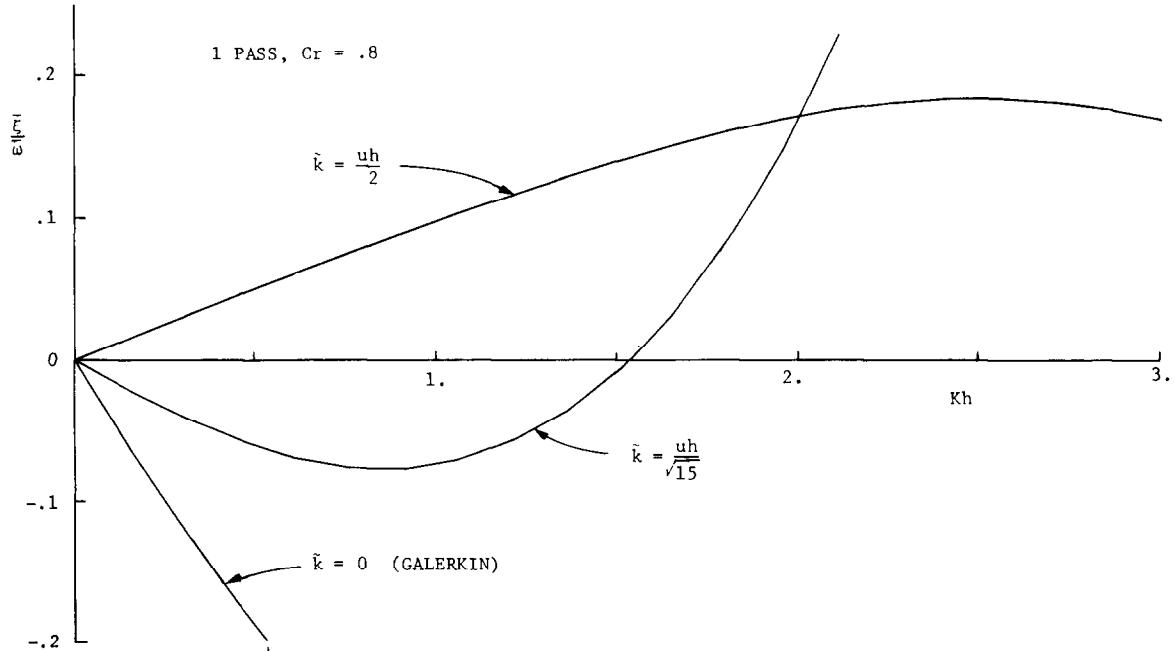


Fig. 4.2. Effect of Petrov-Galerkin parameter, \tilde{k} , on algorithmic damping ratio: Pure advection, one pass. (Negative values result in instability.)

of advection. Note that a negative algorithmic damping ratio gives an exponentially growing (hence unstable) solution. In Fig. 4.2 the algorithmic damping ratio is plotted against the dimensionless wave number for the case of only one pass through the algorithm (i.e., no additional iterations on (4.2.5)–(4.2.6)). It is seen that $\tilde{k} = uh/2$ is the only usable case, the other two being unstable. The case of $\tilde{k} = uh/2$ is not especially good, however, due to significant damping at all values of Kh . This damping is to be expected as the Petrov-Galerkin contribution is felt only in the advective term on the first pass, which can be interpreted as added artificial diffusion (cf. Section 3.4.3).

In Fig. 4.3, the algorithmic damping ratio is plotted as before, for the case of two passes through the algorithm (i.e., one additional iteration on (4.2.5)–(4.2.6)). The Galerkin method is again unstable, as expected, but the Petrov-Galerkin results show exceptionally low damping in the region of accurate resolution ($0 < Kh < 0.6$).

Damping error—Physical diffusion present. In the case when physical diffusion is present, the numerical solution should exhibit the amount of damping to be expected due to the physical diffusion. A measure of this accuracy is the ratio of numerical damping, $\bar{\xi}$, to the physical damping, ξ . Optimally, $\bar{\xi}/\xi$ should be equal to one. For the example to be considered we take the element Peclet number, α , to be 1.25.

This case was chosen to demonstrate how well the physical diffusion is manifested when the algorithm is run at or near the critical time step in an advection dominated situation (i.e., $\alpha > 1$). Note that since the *element* Peclet number is held constant here, the wave number, K , should be considered to be the varying part of Kh .

In Fig. 4.4, $\bar{\xi}/\xi$ is plotted against Kh for one pass through the algorithm. The results show that the Galerkin method has almost no damping in the region of accuracy. The Petrov-

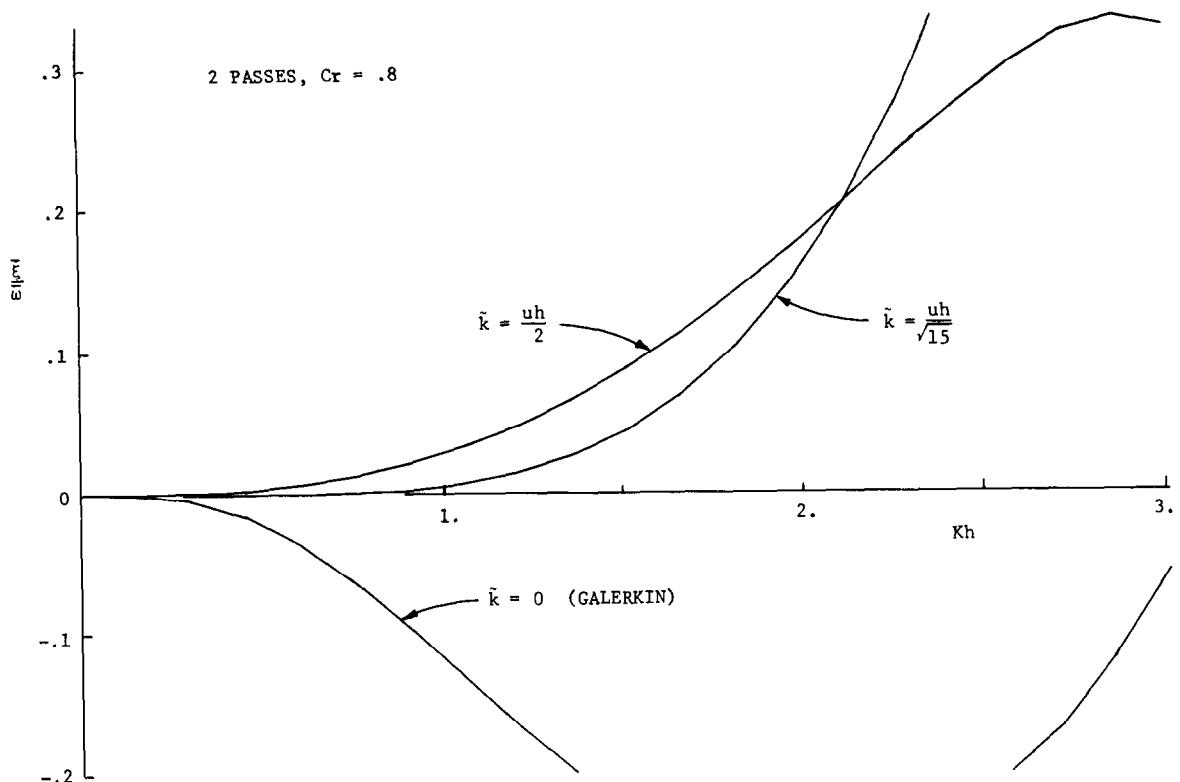


Fig. 4.3. Effect of Petrov-Galerkin parameter, \tilde{k} , on algorithmic damping ratio: Pure advection, two passes. (Negative values result in instability.)

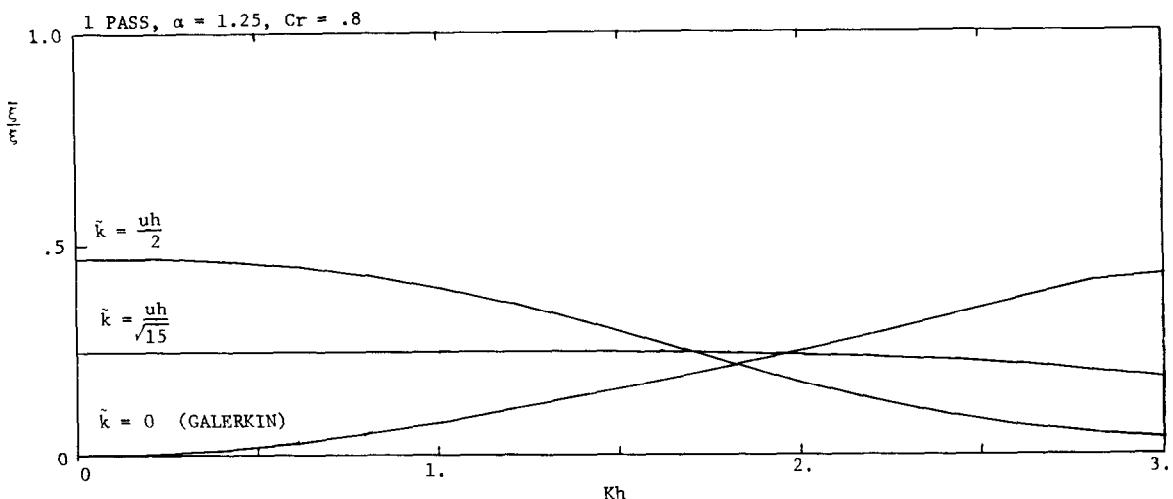


Fig. 4.4. Ratio of numerical damping, $\bar{\xi}$, to physical damping, ξ , for different values of Petrov-Galerkin parameter, \tilde{k} : One pass.

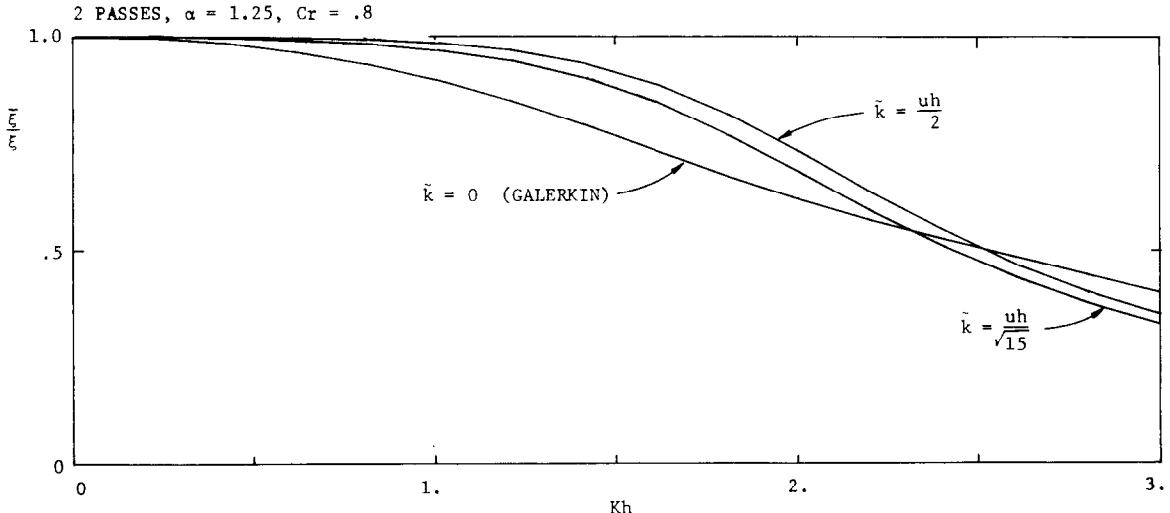


Fig. 4.5. Ratio of numerical damping, $\bar{\xi}$, to physical damping, ξ , for different values of Petrov–Galerkin parameter. \tilde{k} : Two passes.

Galerkin results are not extremely good, but at least some damping is present. The results for two passes are shown in Fig. 4.5. In this case, the Galerkin results are better, but still do not exhibit enough damping. The Petrov–Galerkin results, on the other hand, are very accurate in the region of accuracy ($0 < Kh < 0.6$).

REMARK. Upwind methods have long been criticized as being too diffuse, effectively decreasing the apparent Peclet number. It is seen that the streamline upwind/Petrov–Galerkin method is not subject to such criticisms, and furthermore, the Galerkin method is *underdiffuse*, effectively *increasing* the apparent Peclet number. These results raise questions about the suitability of explicit one pass Galerkin algorithms (see e.g., Gresho et al. [10]).

Phase error. In this case, the frequency, $\bar{\omega}$, of the numerical solution should be as close as possible to the exact frequency, ω . It is well known that explicit lumped mass methods are particularly poor in this regard. In Fig. 4.6, $\bar{\omega}/\omega$ is plotted against Kh for one pass, two passes, and for the implicit consistent mass algorithm. The upwind parameter \tilde{k} is $uh/\sqrt{15}$ and the physical diffusion is absent (pure advection). The one pass results are quite poor, as expected, but the two pass results are quite good, exhibiting the higher-order accuracy of the implicit solution. Results for other values of \tilde{k} are similar: the important result demonstrated here is the beneficial effect of the second pass.

5. Application of the streamline upwind method to incompressible viscous fluid flows

5.1. Preliminaries

5.1.1. Introduction

Numerical solution of the imcompressible Navier–Stokes equations is subject to the types of

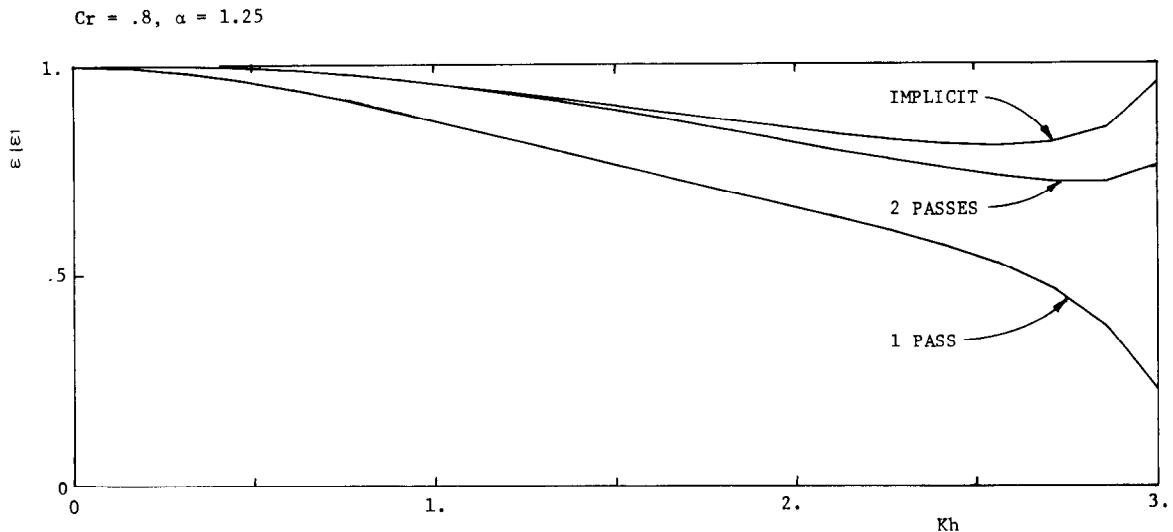


Fig. 4.6. Phase errors for the Petrov-Galerkin transient algorithms.

problems noted for the linear advection-diffusion case. In situations where there are 'hard' downstream boundary conditions, spurious oscillations can develop (see e.g., Fig. 1.1).

The streamline upwind/Petrov-Galerkin Navier-Stokes formulation presented herein precludes spurious oscillations and does not exhibit the loss of accuracy associated with the classical upwind schemes.

5.1.2. Problem statement

Assume the preliminary remarks of Section 3.2.2 are in force. The incompressible transient Navier-Stokes equations can be written as

$$\rho u_{i,t} + \rho u_j u_{i,j} = t_{ij,j} + f_i, \quad (5.1.1)$$

$$u_{i,i} = 0 \quad (5.1.2)$$

where

$$t_{ij} = -P\delta_{ij} + 2\mu u_{(i,j)}. \quad (5.1.3)$$

In the above, u_i is the flow velocity; ρ is the density; f_i is the body force density; t_{ij} is the Cauchy stress; P is the pressure; μ is the dynamic viscosity; and $u_{(i,j)}$ is the symmetrical part of the velocity gradient.

The initial-boundary value problem consists of finding a function $u(x, t)$ which satisfies (5.1.1) and (5.1.2) on Ω , and

$$u_i = g_i \quad \text{on } \Gamma_s, \quad (5.1.4)$$

$$t_i \stackrel{\text{def}}{=} t_{ij}n_j = h_i \quad \text{on } \Gamma_k, \quad (5.1.5)$$

$$u_i(x, 0) = u_{0i} \quad (5.1.6)$$

where g_i and h_i are given functions of x and t , and the initial velocity is a function of x and assumed dilatation free.

REMARK. In the case of $\Gamma_\kappa = \emptyset$ there is a consistency condition on g_i , viz.

$$0 = \int_{\Omega} u_{i,i} d\Omega = \int_{\Gamma} u_i n_i d\Gamma = \int_{\Gamma} g_i n_i d\Gamma. \quad (5.1.7)$$

In this case the pressure P is determined up to an arbitrary constant.

5.1.3. The incompressibility constraint

Treatment of the incompressibility constraint is one of the more difficult aspects of numerical Navier–Stokes calculations. A procedure that is gaining popularity is to develop an auxiliary Poisson equation for the pressure field. From this equation, the pressure is determined such that the continuity equation is satisfied on the average in each element.

This approach is successful only if the pressure field is interpolated with functions at least one order lower than those the velocity field. For example, if the velocity field is interpolated with the bilinear isoparametric shape functions, the pressure must be assumed constant within each element. There are still problems, however, when the pressure is interpolated in this fashion. Under certain types of boundary conditions, the pressure can exhibit a singular ‘checkerboard’ mode. The checkerboard mode is characterized by large pressure oscillations, with positive and negative pressures arranged in a checkerboard pattern.

The incompressibility constraint and its relation to the pressure field have recently been the subject of extensive research [8, 36, 40, 43, 44]. It should be noted, however, that for virtually all practical problems the checkerboard mode is either absent, or can be filtered out by smoothing techniques (see e.g., [26, 32, 44]).

5.1.4. The penalty formulation

It is possible to handle the incompressibility constraint without solving an auxiliary pressure equation through the use of the penalty function formulation (see e.g., Hughes et al. [26, 28]). In this method, the continuity eq. (5.1.2) is replaced by

$$u_{i,i} = -(1/\lambda)P \quad (5.1.8)$$

where λ is the penalty parameter. The modified continuity equation, (5.1.8), is then incorporated into the constitutive equation, (5.1.3), leaving only a modified momentum equation. It was shown in [47] that, under appropriate hypotheses, the solution to the penalty problem converges to that of the fully incompressible problem as $\lambda \rightarrow \infty$. An advantage of the penalty formulation is that the pressure and the continuity equation are eliminated (they are contained implicitly in the modified momentum equation). Physically, the penalty parameter λ may be interpreted as a large dilatational viscosity.

5.1.5. The slightly compressible formulation

The penalty function formulation can often lead to somewhat unphysical results. For example, consider the problem of fluid in a tank under simple gravity loading. The penalty formulation requires that the pressure in the fluid be balanced by a small $O(1/\lambda)$ dilatation. Thus there will be small residual velocities. In most cases, this effect is insignificant, but an alternative formulation may be employed to eliminate potential difficulties.

The alternate makes use of the fluid's bulk modulus, and is referred to as a 'slightly compressible formulation'. The pressure is assumed to be a function of the density, viz.

$$P - P_{\text{ref}} = \beta(\rho - \rho_{\text{ref}}) \quad (5.1.9)$$

where β is the bulk modulus, and P_{ref} and ρ_{ref} are constant reference values of the pressure and density, respectively. In addition, it is assumed that β is large, so that the density is almost constant. This assumption allows the full continuity equation,

$$\rho_{,t} + (\rho u_i)_{,i} = 0, \quad (5.1.10)$$

to be simplified to

$$\rho_{,t} = -\rho u_{i,i}. \quad (5.1.11)$$

The time derivatives of (5.1.9) is then used along with (5.1.11) to arrive at the modified continuity equation, viz.

$$u_{i,i} = -(1/\rho\beta)P_{,t}. \quad (5.1.12)$$

Finally, (5.1.12) is adjointed to the momentum equations, and the pressure is calculated via a time-stepping algorithm. This approach amounts to incorporating a bulk elastic effect.

5.1.6. Remarks

(1) The penalty and slightly compressible formulations require that the λ and β terms in the momentum equation be integrated with a quadrature rule at least one order lower than that used for the viscous term (see [26]). For example, if the viscous term were integrated with two-by-two Gaussian quadrature, then the λ or β terms would have to be integrated with the one point rule. This condition is the counterpart of the lower-order pressure interpolation requirement for the fully incompressible case.

(2) When using the fully incompressible continuity condition (5.1.2), a Poisson equation may be employed to solve for the pressure. Penalty and slightly compressible formations require only the solution of the modified momentum equations, although they too can be cast in a way that includes the pressure equation. The algorithm proposed herein makes use of the pressure equation, and allows for any combination of fully-incompressible, penalty, and slightly-compressible continuity conditions. It will be argued that this approach is considerably more efficient computationally than solving only the momentum equations with penalty or slightly-compressible formulations [9, 15].

(3) In most problems the fully incompressible formulation is the simplest and most accurate

of the three possibilities. However, in cases where the velocity is prescribed on the entire boundary (i.e., $\Gamma_\kappa = \emptyset$), the pressure is determined only up to an arbitrary constant, resulting in a singular mode in the numerical solution. Use of the penalty or slightly compressible formulation in this case removes the undetermined pressure mode.

5.2. Weighted residual formulation

The streamline upwind/Petrov–Galerkin weighted residual formulation for the Navier–Stokes equations is presented in this section. The continuity condition to be employed is a combination of the fully-incompressible (5.1.2), penalty (5.1.8), and slightly-compressible (5.1.12) formulations, viz.

$$u_{i,i} = -(1/\lambda)P - (1/\rho\beta)P_{,t}. \quad (5.2.1)$$

Any combination of the three conditions may be constructed simply by leaving out unwanted terms from (5.2.1).

The streamline upwind/Petrov–Galerkin weighting functions to be considered for the momentum equations are of the form

$$\tilde{w}_i = w_i + p_i \quad (5.2.2)$$

where w_i is continuous across interelement boundaries and p_i is the discontinuous streamline upwind contribution. For the continuity equation, the weighting function is denoted by q , and is assumed to be discontinuous across interelement boundaries. Each of w_i , p_i , and q are assumed to be smooth on element interiors.

Throughout, we shall assume that the trial solutions, u_i , satisfy $u_i = g_i$ on Γ_s , and the weighting functions, w_i , satisfy $w_i = 0$ on Γ_s .

A variational equation for the initial-boundary value problem of Section 5.1 is:

Momentum,

$$\begin{aligned} & \int_{\Omega} w_i (\rho \dot{u}_i + \rho u_j u_{i,j}) d\Omega + \int_{\Omega} t_{ij} w_{(i,j)} d\Omega - \int_{\Omega} w_i f_i d\Omega \\ & + \sum_e \int_{\Omega^e} p_i (\rho \dot{u}_i + \rho u_j u_{i,j} - t_{ij,j} - f_i) d\Omega = \int_{\Gamma_k} w_i h_i d\Gamma. \end{aligned} \quad (5.2.3)$$

Continuity,

$$\sum_e \int_{\Omega^e} q (u_{i,i} + (1/\lambda)P + (1/\rho\beta)\dot{P}) d\Omega = 0. \quad (5.2.4)$$

An equivalent form of (5.2.3) is

$$\sum_e \int_{\Omega^e} \tilde{w}_i (\rho \dot{u}_i + \rho u_i u_{i,j} - t_{ij,j} - f_i) d\Omega + \int_{\Gamma_A} w_i (t_{ij} n_j - \kappa_i) d\Gamma + \int_{\Gamma_{int}} w_i [t_{ij} n_j] d\Gamma = 0, \quad (5.2.5)$$

where $[t_{ij} n_j]$ represents the jump in traction across interelement boundaries (cf., (3.2.16)).

From (5.2.5) and (5.2.4), the Euler-Lagrange equations are seen to be (5.1.1) and (5.2.1) *restricted to the element interiors*, (5.1.5), and the traction continuity condition across interelement boundaries, viz.

$$[t_{ij} n_j] = 0 \quad \text{across } \Gamma_{int}. \quad (5.2.6)$$

REMARKS. (1) Note that the streamline upwind contribution to the weighting function, p_i , weights only on the element interiors, and therefore does not affect continuity conditions, or the weighting of κ_i .

(2) Assume the following conditions hold:

- (i) u_i is interpolated with multilinear isoparametric interpolation functions (e.g., the bilinear functions in two dimensions).
- (ii) P is interpolated with piecewise constant functions (i.e., P is constant in each element).
- (iii) The element domains are rectangular.
- (iv) $u_{i,i} = 0$.

The above assumptions imply that on the interior of each element

$$t_{ij,j} = 0. \quad (5.2.7)$$

It is apparent from (5.2.3) that the streamline upwind contribution does not affect the weighting of the stress divergence term in this case.

(3) Although, in general, assumption (iv) will not hold pointwise within an element, and in any event $t_{ij,j}$ will not vanish identically for non-rectangular elements, it is conjectured that $p_i t_{ij,j}$ can usually be neglected. With higher order elements, the $p_i t_{ij,j}$ term appears to be non-negligible.

Streamline upwind/Petrov-Galerkin weighting function. Analogous to the advection-diffusion case, the streamline upwind weighting function modification is

$$p_i = \bar{k} \hat{u}_j w_{i,j} / \|u\|. \quad (5.2.8)$$

In the above, \bar{k} is defined as in Section 3.3., with α defined to be the element Reynolds number, viz.

$$\alpha = \rho u h / (2\mu). \quad (5.2.9)$$

Similar definitions are made for α_ϵ and α_η .

5.3. Transient algorithm

5.3.1. Introduction

The weighted residual formulation, (5.2.3) and (5.2.4), along with appropriate definitions of the finite element weighting and interpolation functions, defines a system of semi-discrete equations. To complete the definition of the numerical solution procedure, a transient algorithm is required. The algorithm to be used here is based upon the predictor-multicorrector method discussed in the previous chapter. All terms except the pressure will be treated explicitly. Due to stability considerations the pressure term must be treated implicitly.

We have so far employed four-node quadrilateral elements with bilinear velocity interpolation and constant pressures. In this case, the weight functions, q , for the continuity equation (5.2.4) are simply constant on each element. These simple elements have proven to be completely satisfactory in all test cases. We can therefore see no compelling reasons to extend the method to costly and complicated higher-order elements.

The semi-discrete equations emanating from (5.2.3) and (5.2.4) are of the form

$$\mathbf{M}\mathbf{a} + \mathbf{C}\mathbf{v} + \mathbf{N}(\mathbf{v}) - \mathbf{GP} = \mathbf{F}, \quad (5.3.1)$$

$$\mathbf{G}^t\mathbf{v} = \mathbf{D} - (1/\lambda)\mathbf{M}^P\mathbf{P} - (1/\rho\beta)\mathbf{M}^P\dot{\mathbf{P}}. \quad (5.3.2)$$

In the above, \mathbf{M} is the consistent mass matrix; \mathbf{C} is the viscous matrix; $\mathbf{N}(\mathbf{v})$ is the nonlinear convective force vector; \mathbf{G} is the gradient operator; \mathbf{F} is the generalized force vector; \mathbf{D} is a generalized force vector representing the effect of the prescribed boundary velocities on (5.3.2); and \mathbf{M}^P is the generalized ‘mass’ matrix for the pressure terms in (5.2.4). The vectors \mathbf{v} and \mathbf{a} are nodal values of u_i and \dot{u}_i , respectively.

The initial value problem consists of finding functions $\mathbf{v} = \mathbf{v}(t)$ and $\mathbf{P} = \mathbf{P}(t)$ satisfying (5.3.1) and (5.3.2) and the initial conditions

$$\mathbf{v}(0) = \mathbf{v}_0, \quad (5.3.3)$$

$$\mathbf{P}(0) = \mathbf{P}_0 \quad (5.3.4)$$

where \mathbf{v}_0 and \mathbf{P}_0 are given.

5.3.2. Solution procedure

At this point it is useful to introduce some additional notation:

- \mathbf{M}^* lumped counterpart of \mathbf{M} (see e.g., [26]);
- I the number of corrector passes through the algorithm ($0 \leq i \leq I - 1$);
- γ_v parameter governing the approximation of \mathbf{a} ;
- γ_p parameter governing the approximation of \mathbf{P} .

The first step in the numerical solution procedure is to calculate the predicted values ($i = 0$) of \mathbf{v} , \mathbf{a} , \mathbf{P} , and $\dot{\mathbf{P}}$, viz.

$$\mathbf{v}_{n+1}^{(0)} = \mathbf{v}_n + \Delta t(1 - \gamma_v)\mathbf{a}_n, \quad (5.3.5)$$

$$\mathbf{a}_{n+1}^{(0)} = \mathbf{0}, \quad (5.3.6)$$

$$\mathbf{P}_{n+1}^{(0)} = \mathbf{P}_n + \Delta t(1 - \gamma_p)\dot{\mathbf{P}}_n, \quad (5.3.7)$$

$$\mathbf{P}_{n+1}^{(0)} = \mathbf{0}. \quad (5.3.8)$$

During the first time step ($n = 0$), (5.3.5) and (5.3.7) are replaced by

$$\mathbf{v}_{n+1}^{(0)} = \mathbf{v}_0, \quad (5.3.9)$$

$$\mathbf{P}_{n+1}^{(0)} = \mathbf{P}_0. \quad (5.3.10)$$

Next, the discretized momentum equation is written in the ‘residual’ or ‘out-of-balance force’ formulation (equations that follow are all at time t_{n+1} , so the $(n + 1)$ subscript is deleted for clarity).

$$\mathbf{M}^* \Delta \mathbf{a}^{(i)} - \mathbf{G} \Delta \mathbf{P}^{(i)} = \mathbf{R}^{(i)} \quad (5.3.11)$$

where

$$\mathbf{R}^{(i)} = \mathbf{A}_e (\mathbf{f}^e - \mathbf{m}^e \mathbf{a}^{e(i)} - \mathbf{c}^e \mathbf{v}^{e(i)} - \mathbf{n}^e (\mathbf{v}^{e(i)}) + \mathbf{g}^e \mathbf{p}^{e(i)}). \quad (5.3.12)$$

The element vectors $\mathbf{v}^{e(i)}$ and $\mathbf{a}^{e(i)}$ contain the prescribed boundary conditions as well as the local values of the global \mathbf{v} and \mathbf{a} vectors. The vector \mathbf{f}^e represents the consistent nodal forces arising from the body force term, f_i , and from the traction boundary condition on Γ_e .

Note that due to the implicit treatment of the pressure, the term $\mathbf{G} \Delta \mathbf{P}^{(i)}$ appears on the left-hand side of (5.3.1), (cf. Section 4.2). All other terms are treated explicitly.

Eq. (5.3.1) is now modified to solve for the acceleration increment without the implicit pressure contribution. Specifically, let

$$\Delta \mathbf{a}^{*(i)} = \Delta \mathbf{a}^{(i)} - \mathbf{M}^{*-1} \mathbf{G} \Delta \mathbf{P}^{(i)}. \quad (5.3.13)$$

Using (5.3.13) in (5.3.11), the modified equation is now

$$\mathbf{M}^* \Delta \mathbf{a}^{*(i)} = \mathbf{R}^{(i)}. \quad (5.3.14)$$

Note that since \mathbf{M}^* is diagonal, the solution of (5.3.14) is trivial.

The next step is the calculation of the pressure increment from the consistent Poisson equation,

$$\mathbf{K} \Delta \mathbf{P}^{(i)} = \mathbf{B}^{(i)} \quad (5.3.15)$$

where \mathbf{K} and $\mathbf{B}^{(i)}$ will be defined subsequently.

The corrector phase then completes the iteration, viz.

$$\Delta \mathbf{a}^{(i)} = \Delta \mathbf{a}^{*(i)} + \mathbf{M}^{*-1} \mathbf{G} \Delta \mathbf{P}^{(i)}, \quad (5.3.16)$$

$$\mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} + \gamma_v \Delta t \Delta \mathbf{a}^{(i)}, \quad (5.3.17)$$

$$\mathbf{a}^{(i+1)} = \mathbf{a}^{(i)} + \Delta \mathbf{a}^{(i)}, \quad (5.3.18)$$

$$\mathbf{P}^{(i+1)} = \mathbf{P}^{(i)} + \Delta \mathbf{P}^{(i)}, \quad (5.3.19)$$

$$\dot{\mathbf{P}}^{(i+1)} = \dot{\mathbf{P}}^{(i)} + (1/\gamma_p \Delta t) \Delta \mathbf{P}^{(i)}. \quad (5.3.20)$$

5.3.3. Derivation of the consistent Poisson equation

First, we define \mathbf{v}^* as the velocity corrected only by \mathbf{a}^* (pressure effect not yet included),

$$\mathbf{v}^{*(i)} = \mathbf{v}^{(i)} + \gamma_v \Delta t \Delta \mathbf{a}^{*(i)}. \quad (5.3.21)$$

The final corrected velocity, then can be written

$$\mathbf{v}^{(i+1)} = \mathbf{v}^{*(i)} + \gamma_v \Delta t \mathbf{M}^{*-1} \mathbf{G} \Delta \mathbf{P}^{(i)}. \quad (5.3.22)$$

Multiplying (5.3.22) through by \mathbf{G}^t , we obtain

$$\mathbf{G}^t \mathbf{v}^{(i+1)} = \mathbf{G}^t \mathbf{v}^{*(i)} + \gamma_v \Delta t \mathbf{G}^t \mathbf{M}^{*-1} \mathbf{G} \Delta \mathbf{P}^{(i)} \quad (5.3.23)$$

We shall require that the discrete continuity eq. (5.3.2) be satisfied at every iterate. Making use of the pressure corrector equations, (5.3.2) may be written for the $(i+1)$ iterate as,

$$\mathbf{G}^t \mathbf{v}^{(i+1)} = \mathbf{D} - \frac{1}{\lambda} \mathbf{M}^P \mathbf{P}^{(i)} - \frac{1}{\lambda} \mathbf{M}^P \Delta \mathbf{P}^{(i)} - \frac{1}{\rho \beta} \mathbf{M}^P \dot{\mathbf{P}}^{(i)} - \frac{1}{\Delta t \gamma_p \rho \beta} \mathbf{M}^P \Delta \mathbf{P}^{(i)}. \quad (5.3.24)$$

After subtracting (5.3.24) from (5.3.23), the definitions of \mathbf{K} and $\mathbf{B}^{(i)}$ are seen to be

$$\mathbf{K} = \mathbf{G}^t \mathbf{M}^{*-1} \mathbf{G} + \frac{1}{\gamma_v \Delta t} \left(\frac{1}{\lambda} + \frac{1}{\rho \beta \gamma_p \Delta t} \right) \mathbf{M}^P, \quad (5.3.25)$$

$$\mathbf{B}^{(i)} = -\frac{1}{\gamma_v \Delta t} \left(\mathbf{G}^t \mathbf{v}^{*(i)} + \frac{1}{\lambda} \mathbf{M}^P \mathbf{P}^{(i)} + \frac{1}{\rho \beta} \mathbf{M}^P \dot{\mathbf{P}}^{(i)} - \mathbf{D} \right). \quad (5.3.26)$$

REMARKS. (1) \mathbf{K} is symmetric and positive definite, and usually is formed and factored only once. If penalty or slightly compressible terms are present, \mathbf{K} is seen to be a function of Δt . In this case, \mathbf{K} must be reformed and factored every time Δt is changed. Procedures are available which reduce the frequency of reforming \mathbf{K} by keeping the generalized time steps, $\gamma_v \Delta t$ and $\gamma_p \Delta t$, constant (see e.g., [26]).

(2) In practice, the right-hand side vector $\mathbf{B}^{(i)}$ is formed element by element. On the element level, $\mathbf{v}^{*(i)}$ is modified to include the prescribed boundary velocities, which incorporates the calculation of \mathbf{D} into the $\mathbf{G}^t \mathbf{v}^{*(i)}$ term.

5.3.4. Flow chart

To clarify the algorithm described in the previous section, a flow chart is presented in Table 5.1.

Table 5.1
Flow chart for streamline upwind Navier–Stokes algorithm

-
- | | |
|----------|---|
| Step 1. | Form \mathbf{G} , \mathbf{M}^* , \mathbf{M}^P ; form and factorize \mathbf{K} . |
| Step 2. | Initialize n , t_n , \mathbf{v}_n , and \mathbf{P}_n . |
| Step 3. | Set iterate counter $i = 0$. |
| Step 4. | Form predictors $\mathbf{v}_{n+1}^{(i)}$, $\mathbf{a}_{n+1}^{(i)}$, $\mathbf{P}_{n+1}^{(i)}$, and $\dot{\mathbf{P}}_{n+1}^{(i)}$. |
| Step 5. | Solve $\mathbf{M}^* \Delta \mathbf{a}^{*(i)} = \mathbf{R}^{(i)}$ for $\Delta \mathbf{a}^{*(i)}$. |
| Step 6. | Calculate time increment for the next step. |
| Step 7. | Solve $\mathbf{K} \Delta \mathbf{P}^{(i)} = \mathbf{B}^{(i)}$ for $\Delta \mathbf{P}^{(i)}$. |
| Step 8. | Form correctors $\mathbf{v}_{n+1}^{(i+1)}$, $\mathbf{a}_{n+1}^{(i+1)}$, $\mathbf{P}_{n+1}^{(i+1)}$, $\dot{\mathbf{P}}_{n+1}^{(i+1)}$. |
| Step 9. | Increment iterate counter, $i = i + 1$, |
| | if $i > I$, go to Step 5. |
| | if $i = I$, continue. |
| Step 10. | Define $\mathbf{v}_{n+1} = \mathbf{v}_{n+1}^{(i+1)}$, etc. |
| Step 11. | Update time and increment n : $t_{n+1} = t_n + \Delta t$, $n = n + 1$. |
| Step 12. | If additional time steps are required go to Step 3. If not, stop. |
-

5.3.5. Remarks

(1) The critical time step we have used is based on the linear advection-diffusion equation time-step limits discussed in Section 4.3. We have, however, employed a simplified definition of the Courant number (4.3.5) in the two-dimensional case, viz.

$$Cr = \Delta t (|u_\xi|/h_\xi + |u_\eta|/h_\eta)$$

(2) At least two iterations ($I = 2$) are required in order to achieve the beneficial effects of the consistent mass contribution to the residual force vector. (Recall that the predicted acceleration is zero on the first iteration, hence there is no mass contribution to $\mathbf{R}^{(i)}$ until the second iteration.)

(3) To avoid algorithmic damping, γ_v and γ_p are usually set equal to 1/2. In the fully incompressible case (i.e. λ and β terms are absent), the time derivative of P does not enter into the semi-discrete equations, hence the calculated pressure is independent of the choice of γ_p .

(4) The basic structure of the present algorithmic treatment has essential features in common with the methods proposed by Chorin [4] and Temam [47]. Variants of this structure are now gaining popularity in finite element fluid flow applications (see e.g., [10, 9]). The contributions to this structure made herein are: (i) the incorporation of a residual formulation, which naturally allows for multiple iterations; (ii) the streamline upwind/Petrov–Galerkin weighted residual formulation; and (iii) the option of using penalty and/or slightly compressible formulations.

(5) For convenience, the initial velocity and pressure are often set to zero. When there are prescribed non-zero velocity boundary conditions, these initial conditions violate the con-

tinuity equation in the boundary elements. In spite of this, the first time step produces a smooth incompressible velocity field (potential-flow like) which becomes the effective initial condition. Although several investigators [13, 42] have expressed concerns about this procedure, no problems have been encountered in practice when the streamline upwind/Petrov–Galerkin formulation is employed.

5.4. Remarks on computational aspects

5.4.1. Introduction

The streamline upwind/Petrov–Galerkin Navier–Stokes algorithm has been implemented in a compact research code. For simplicity and efficiency, the four-node bilinear velocity, constant pressure element is employed. With modest mesh generation and plotting capability, the code contains fewer than 2600 FORTRAN statements. Additional plotting and data reduction is handled by a separate postprocessor code.

This section will detail some aspects of the code which may be different than standard finite element methodology.

5.4.2. Formation of consistent Poisson matrix (\mathbf{K})

In most finite element equation systems, the unknowns represent *nodal* values. However, in the case of the Poisson equation used in the present algorithm, the unknowns are the constant *element* pressures. As a result, it is not possible to define an element level Poisson matrix. This requires that \mathbf{K} be formed globally, rather than in the usual element-by-element fashion.

Recall from (5.3.25), the definition of \mathbf{K} is

$$\mathbf{K} = \mathbf{G}^t \mathbf{M}^{*-1} \mathbf{G} + \frac{1}{\gamma_v \Delta t} \left(\frac{1}{\lambda} + \frac{1}{\rho \beta \gamma_p \Delta t} \right) \mathbf{M}^P. \quad (5.4.1)$$

The gradient operator, \mathbf{G} , has dimensions of N_{eq}^v rows by N_{eq}^P columns, where N_{eq}^v and N_{eq}^P are the number of velocity equations and the number of pressure equations, respectively. The lumped mass, \mathbf{M}^* , has dimensions of N_{eq}^v by N_{eq}^v , and the generalized pressure ‘mass’ matrix, \mathbf{M}^P , has dimensions of N_{eq}^P by N_{eq}^P . (Both \mathbf{M}^* and \mathbf{M}^P are diagonal, and are stored as vectors.)

Conceptually, \mathbf{K} is formed with the global matrix products in (5.4.1), but in practice this is not possible as the \mathbf{G} matrix is not stored globally. Instead, each entry of \mathbf{K} is calculated individually, making use of the element level gradient matrix \mathbf{g}^e and mappings which project element equations into global equations.

Another unusual feature of \mathbf{K} is that its band-profile structure is a function of element, rather than nodal, ordering. For efficient operation, the elements should be numbered to minimize the bandwidth of \mathbf{K} .

5.4.3. Computational efficiency

In Section 5.1 it was noted that the continuity condition could be incorporated into the momentum equations with the use of either the penalty or slightly compressible formulations. While such a procedure may simplify the algorithm, the overall efficiency is reduced compared with the formulation in which pressure is segregated [4, 9, 15, 47].

Computational effort at each time step may be split into two major parts: (i) the implicit

solution for the pressure, and (ii) the formation of the residual force vector $\mathbf{R}^{(i)}$. It is in the implicit pressure solution that the momentum equation penalty formulation is less efficient. In this case the penalty term is treated implicitly in an equation system with N_{eq}^{v} unknowns. In the fully incompressible formulation, the pressure equation system has N_{eq}^{p} unknowns.

In a large two dimensional mesh there are approximately two velocity equations for every pressure equation. Thus the momentum equation penalty formulation requires implicit solution of twice as many equations. In addition, the mean-half bandwidth, M_b , of the implicit matrix is twice as large as that for the Poisson equation system used for the fully incompressible formulation.

An estimate for the forward reduction/back substitution time, T_{eq} , for the 'active column' equation solver [46] used in the code is,

$$T_{\text{eq}} = C_{\text{eq}} N_{\text{eq}} M_b , \quad (5.4.2)$$

where C_{eq} is a function of computer speed and N_{eq} is the number of equations. Clearly, the implicit pressure solution for the momentum equation penalty formulation is four times slower than for the auxilliary Poisson equation formulation.

The formation of the residual force vector, $\mathbf{R}^{(i)}$, is performed element-by-element, and

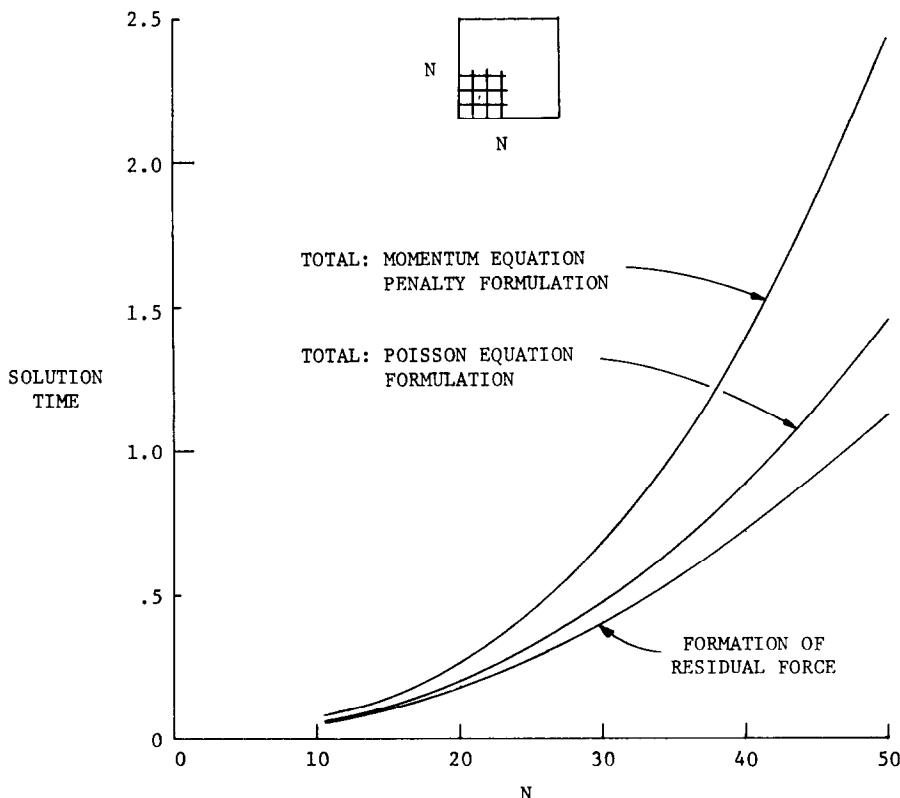


Fig. 5.1. Comparison of solution times: Two dimensions. (Time is given in CPU seconds per time step per iteration.) The difference between the total and the formation of the residual force represents the time for equation solving.

requires about the same amount of effort for either formulation. An estimate for the formation time, T_r , of $\mathbf{R}^{(i)}$ is

$$T_r = C_r n_{el} \quad (5.4.3)$$

where C_r is dependent on computer speed and the level of code optimization, and n_{el} is the number of elements. Empirical values for C_{eq} and C_r (one point quadrature $\mathbf{R}^{(i)}$) for the CDC 7600 computer are,

$$C_{eq} = 2.6 \times 10^{-6}, \quad (5.4.4)$$

$$C_r = 4.5 \times 10^{-4}. \quad (5.4.5)$$

Values for the IBM 3033 computer are about twice as great.

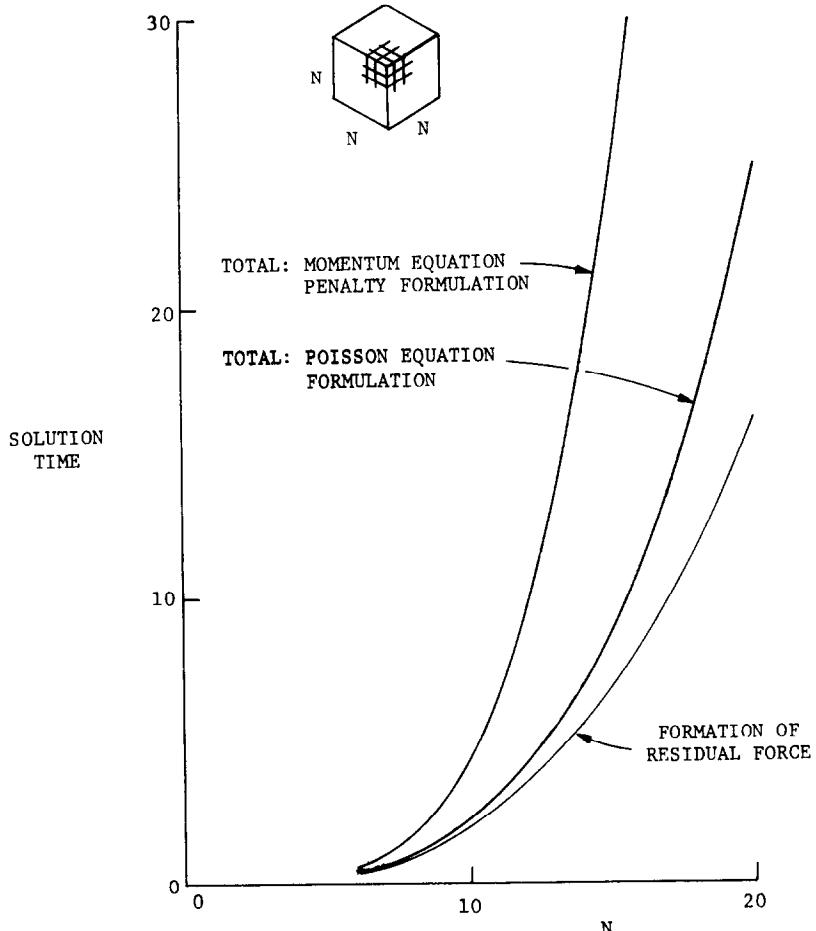


Fig. 5.2. Comparison of solution times: Projected values for three dimensions. (Time is given in CPU seconds per time step per iteration.)

The sum of T_{eq} and T_r gives an estimate of the solution time (in seconds) per time step per iteration,

$$T = 2.6 \times 10^{-6} N_{\text{eq}} M_b + 4.5 \times 10^{-4} n_{\text{el}} .$$

For clarification, it is useful to consider an example of an N -element by N -element square mesh. The time estimates for the momentum equation penalty (T_{pen}) and the auxiliary Poisson equation (T_{pois}) formulations are then,

$$T_{\text{pen}} = 4(2.6 \times 10^{-6})N^3 + (4.5 \times 10^{-4})N^2 , \quad (5.4.7)$$

$$T_{\text{pois}} = (2.6 \times 10^{-6})N^3 + (4.5 \times 10^{-4})N^2 . \quad (5.4.8)$$

Eqs. (5.4.7) and (5.4.8) are plotted in Fig. 5.1. It is seen that for small to medium sized problems, the solution of the Poisson equation is a small part of the total solution time. For example, in the example problem of Section 6, the solution of the Poisson equation required only 20% of the total solution time. However, for the momentum equation penalty formulation, the implicit equation solving time rapidly becomes a large portion of the solution time.

Projected solution times for a three dimensional $N \times N \times N$ mesh are plotted in Fig. 5.2. In this case, the Poisson equation formulation has a very pronounced advantage, even for relatively small meshes.

6. Navier–Stokes numerical example: Flow past a circular cylinder

6.1. Introduction

Simulation of flow past a circular cylinder is one of the most challenging problems for numerical solution methods. Unlike many other typical example problems, all of the terms in the governing equations are significant in this case, requiring across-the-board accuracy from the numerical method for a successful simulation.

The problem consists of a circular cylinder immersed in a flowing viscous fluid. At Reynolds numbers below about 40, a pair of symmetrical eddies form on the downstream side of the cylinder. At higher Reynolds numbers, the symmetrical eddies become unstable and periodic vortex shedding occurs. The eddies are transported downstream, resulting in the well-known Karman vortex street.

This problem is of engineering interest, as vortex shedding can induce significant structural vibrations. These practical engineering problems generally have high Reynolds numbers (over 10^6), and have fine scale turbulence in addition to the large scale vortex structures. It is usually not possible to numerically calculate the fine scale details, so turbulence, or ‘subgrid-scale’ models are introduced. These models generally use some form of additional diffusivity to account for the turbulence that cannot be resolved numerically. Before attempting solutions at high Reynolds numbers with turbulence models, it is important to verify that the method is accurate at moderate Reynolds numbers, where it is possible to resolve all flow details. In fact,

Leonard [33] has suggested that inaccurate (e.g., full upwind) numerical methods are hindering the development of accurate turbulence models.

A Reynolds number of 100 is considered to be the standard for testing numerical methods on the cylinder problem. It is high enough for vortex shedding to occur, but low enough that boundary layers can be easily resolved. The reader may consult [13, 14, 45] for further background on this problem.

6.2. Problem statement and finite element mesh

The domain and boundary conditions are shown in Fig. 6.1. The Reynolds number based on the inlet velocity and the cylinder diameter is 100. The finite element mesh is shown in Figs. 6.2 and 6.3. In designing the mesh, every effort was made to assure adequate resolution of all flow details. At the cylinder, element thicknesses were graded to efficiently resolve the developing boundary layer. In the downstream region, elements are sized to capture the vortex street. This is an area that many investigators have not treated properly. The wavelength of the vortex street is about 6 cylinder diameters, but in [14, 45] the length of the downstream elements is greater than one diameter. This results in fewer than 6 elements to resolve one wave-length. In the present mesh, the downstream elements are sized at one-half of the cylinder diameter, giving 12 elements per wavelength.

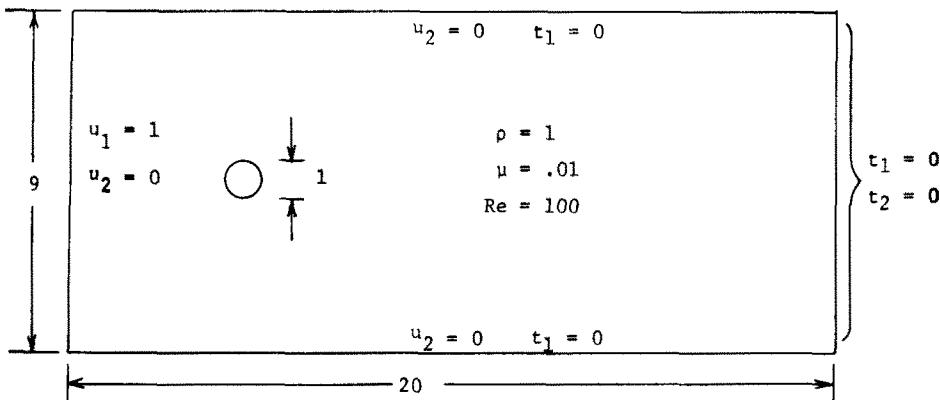


Fig. 6.1. Flow past a cylinder: Problem statement.

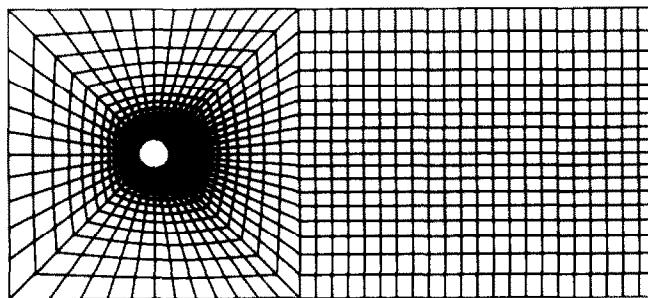


Fig. 6.2. Finite element mesh; 1510 nodes, 1436 elements.

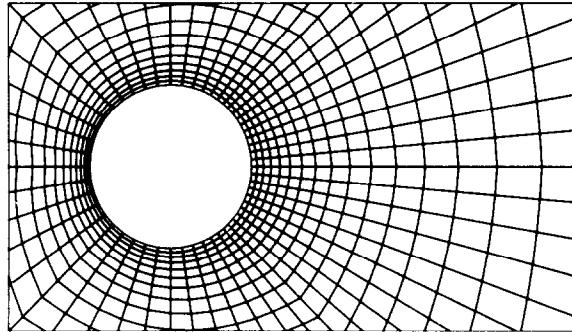


Fig. 6.3. Details of mesh near cylinder.

The solution method employed is the fully incompressible version of the algorithm described in Section 5. The Petrov-Galerkin parameter was selected to optimize phase accuracy, (3.3.2) and (3.3.11), and one additional iteration was performed on (5.3.14)–(5.3.20) every time step. To minimize computation time, one-point Gaussian quadrature was employed throughout. The time step, governed by the small elements near the cylinder, was constant at 0.03. This results in about 33 time steps per diameter of freestream movement, and about 200 time steps per vortex shedding cycle.

This initial condition was zero velocity everywhere. This is, of course, inconsistent with the inlet boundary condition of unit velocity. This does not, however, present any problems at all for the algorithm. The first time step produces a smooth incompressible velocity field which becomes the effective initial condition.

These calculations were performed in single precision (60 bits per word) on the CDC 7600 computer at the NASA Ames Research Center.

6.3. Results

The problem was run a total of 4800 time steps, corresponding to 144 time units. Note that by virtue of the unit diameter of the cylinder and the unit freestream velocity, each time unit represents one diameter of freestream movement.

Initially, a pair of symmetric attached eddies grew behind the cylinder, reaching a steady state by about $t = 36$. Velocity vectors of this development are shown in Fig. 6.4, and stationary streamlines are shown in Fig. 6.5. (Stationary streamlines are those seen by an observer moving with the flow.) Pressure and vorticity contours at $t = 45$ are shown in Fig. 6.6. Distributions of pressure and skin-friction coefficients around the cylinder are plotted in Fig. 6.7(a) and (b). No effort has been made to compare these values with other computed or experimental results due to the significant effects of blockage in the relatively narrow channel used in this study.

The results after 1800 time steps showed a very small amplitude ($\sim 10^{-11}$) vertical oscillation of the symmetric eddies. Although the oscillation amplitude was growing slowly with time, a perturbation was added in an attempt to hasten vortex shedding. Small forces were added to boundary layer nodes, as shown in Fig. 6.8, for 150 time steps starting at step 1801. The perturbation had little noticeable effect, as the oscillations merely continued growing slowly for an additional 1400 time steps. At about $t = 96$ (3200 time steps) vortex shedding began.

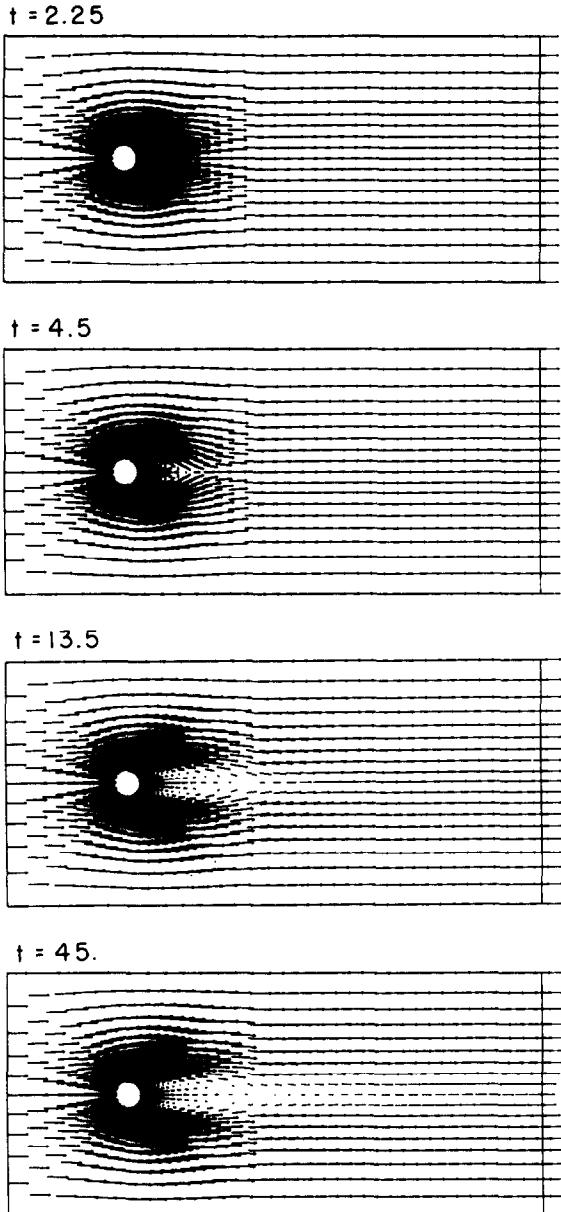


Fig. 6.4. Development of symmetric solution: Velocity vectors.

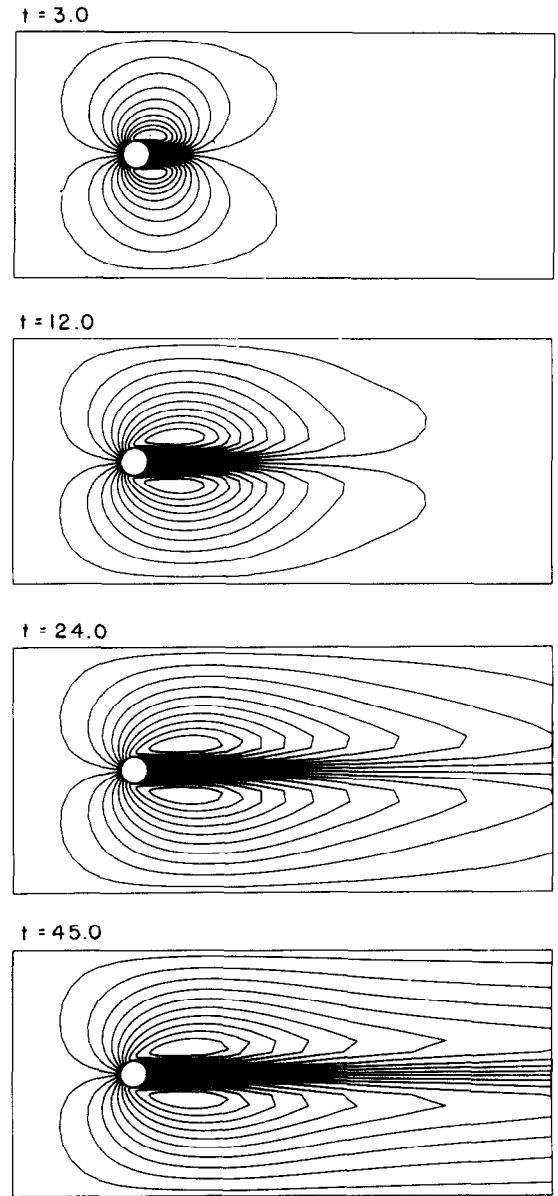


Fig. 6.5. Development of symmetric solution: Stationary streamlines.

Steady periodic shedding was achieved after about 6 shedding cycles at $t \approx 132$. The complete history of this simulation is shown in Fig. 6.9.

The observed shedding period, τ , was 6 time units (200 time steps), giving a dimensionless shedding frequency, or Strouhal number ($S = D/u_0\tau$), of 0.167. This compares well with results given in [14, 45].

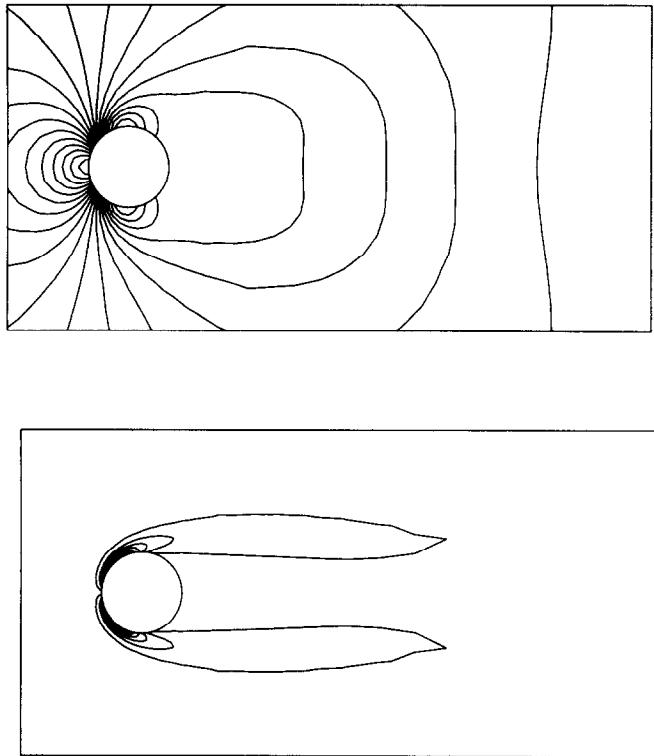


Fig. 6.6. Pressure and vorticity contours for the symmetric solution; $t = 45$.

Velocity vectors tracing the development of periodic shedding are shown in Fig. 6.10. Streamlines and stationary streamlines are shown in Figs. 6.11 and 6.12. Streamlines, pressure contours and vorticity contours one-half cycle apart are presented in Fig. 6.13.

6.4. Discussion

In general, the results look very good, and compare favorably with the implicit quadratic element calculations of Gresho et al. [13]. None of the poor vortex behavior reported on by Gresho et al. [14] for bilinear elements is seen in this case. This is attributed to the use of a finer mesh and to the superior phase speed properties of the streamline upwind/Petrov-Galerkin method.

It is of interest to compare the computational efficiencies of implicit methods to explicit methods. With an implicit method, the time step is limited only by accuracy considerations, while for explicit methods the time step is governed by the stability of the algorithm. The critical time step is usually much smaller than the time step required for accuracy alone. Thus, while explicit methods are faster per time step, the increase in the number of time steps needed may offset any speed advantages. A useful comparison of the present explicit calculation can be made with the implicit calculation of Gresho et al. [13]. Both were run on CDC 7600 computers. Table 6.1 lists the relevant data for comparison.

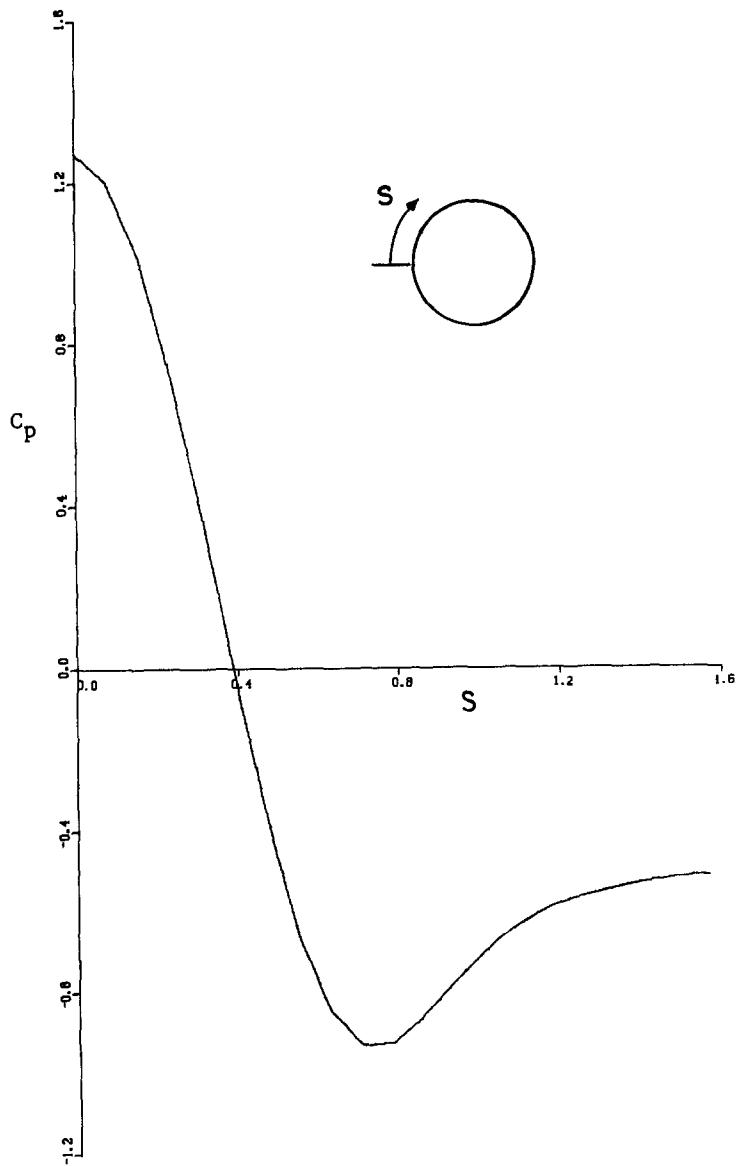


Fig. 6.7(a). Pressure coefficient on cylinder: symmetric solution.

It is seen that the explicit calculation, which has more than twice the number of total equations, and ten times as many time steps per shedding cycle, is still faster than the implicit calculation. For both cases, the computer codes employed were unoptimized research versions, and it is expected that optimization would at least double the efficiency.

The present calculation could have been performed using the one pass Galerkin formulation, rather than two pass Petrov-Galerkin, with the following consequences:

- (1) The critical time step would be three times smaller. In this convection dominated flow,

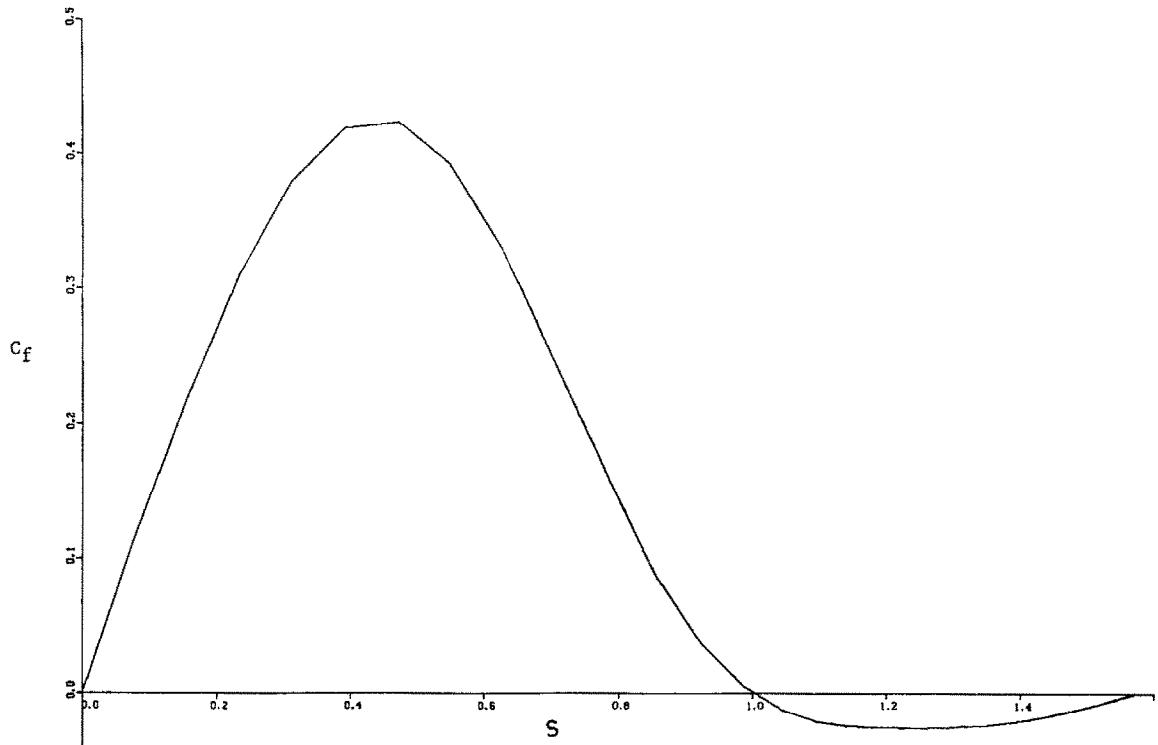


Fig. 6.7(b). Skin friction coefficient on cylinder: symmetric solution.

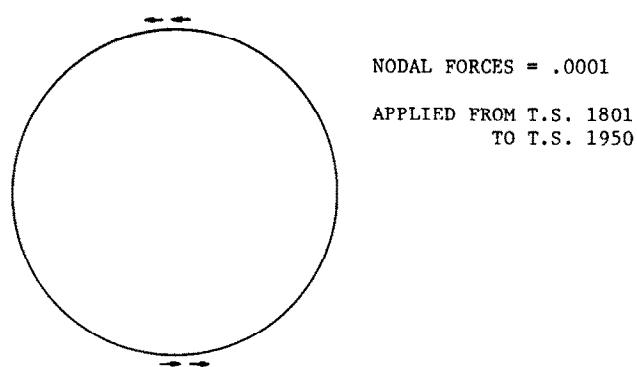


Fig. 6.8. Boundary layer perturbation.

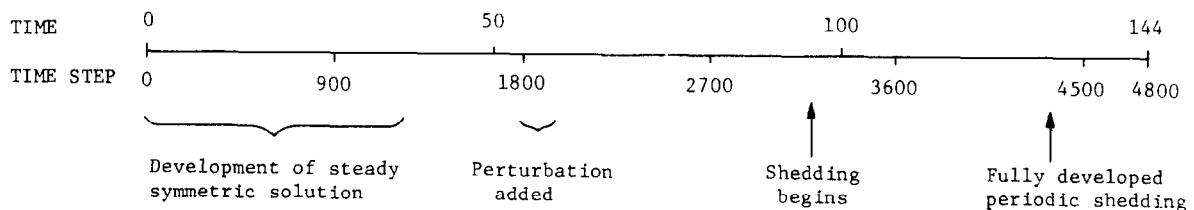


Fig. 6.9. History of important flow features.

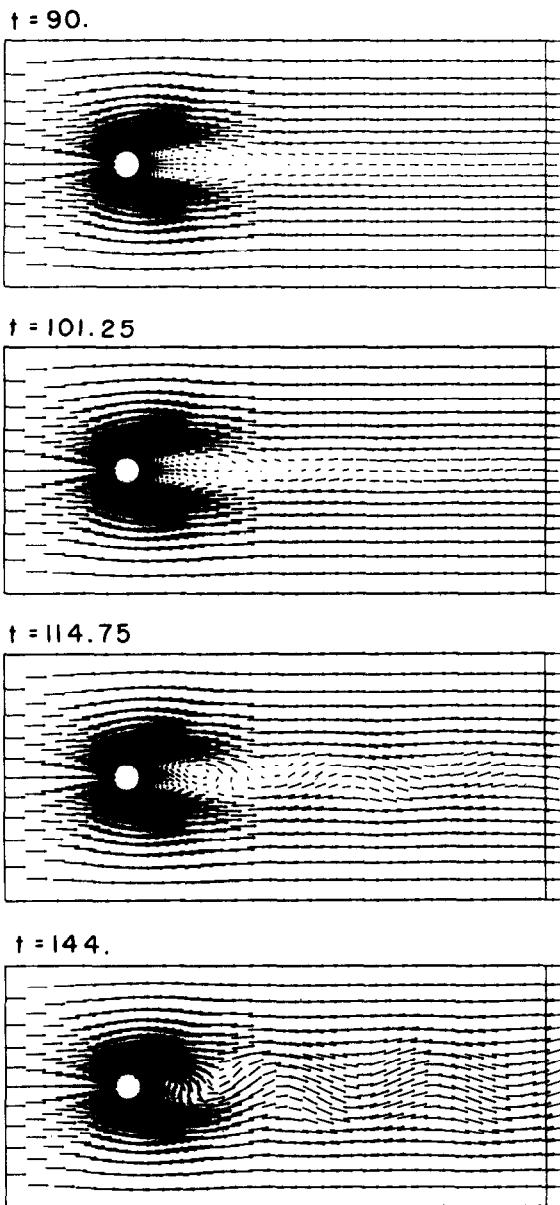


Fig. 6.10. Developing vortex shedding: Velocity vectors.

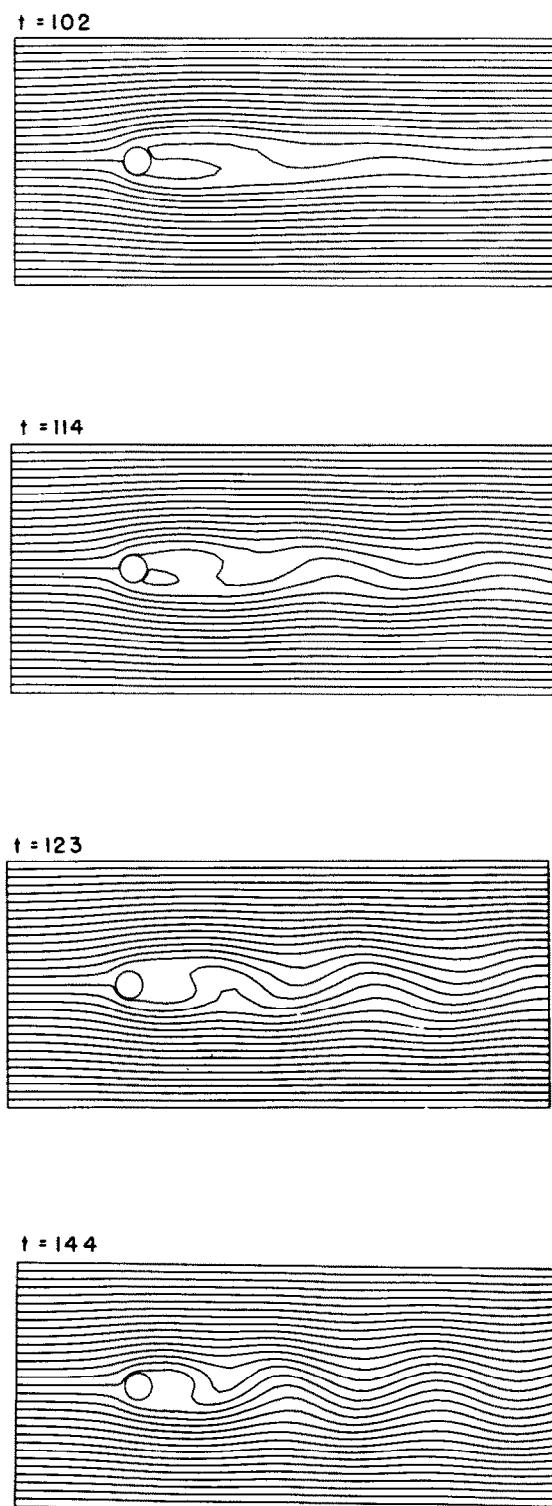


Fig. 6.11. Developing vortex shedding: Streamlines.

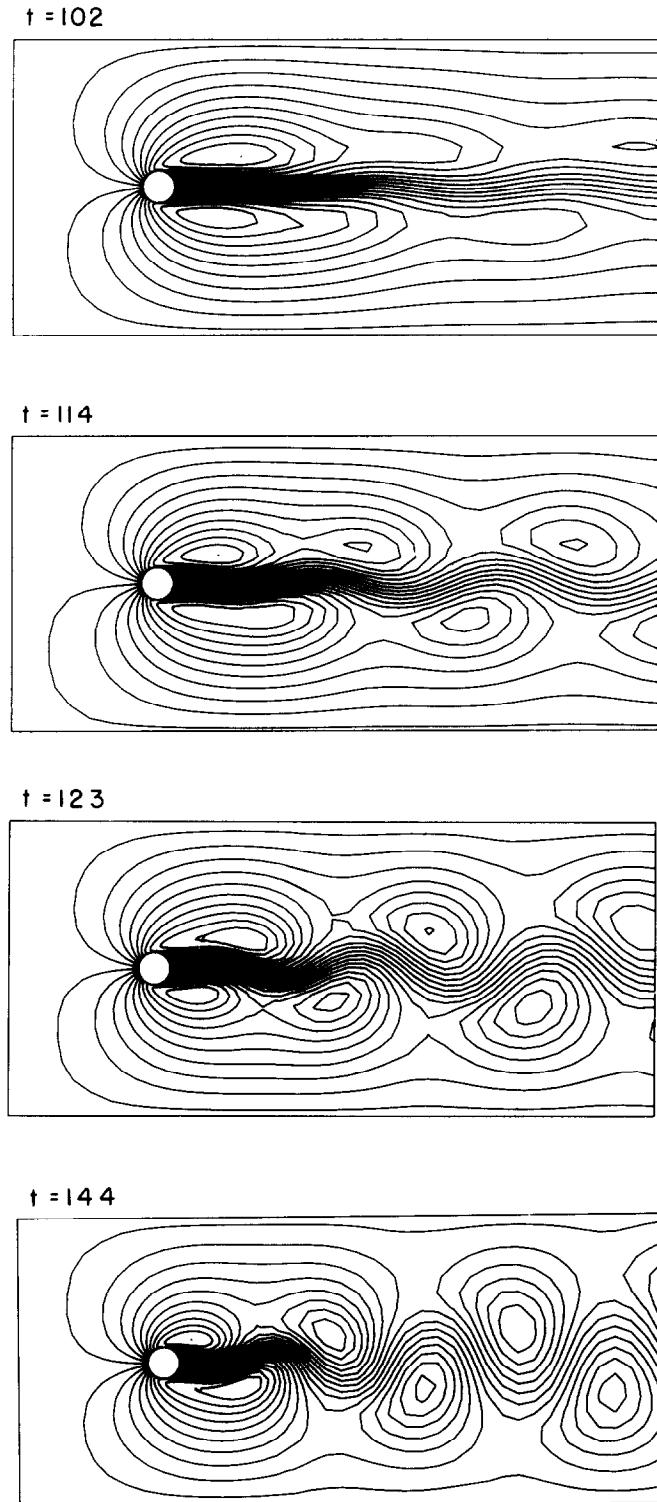


Fig. 6.12. Developing vortex shedding: Stationary streamlines.

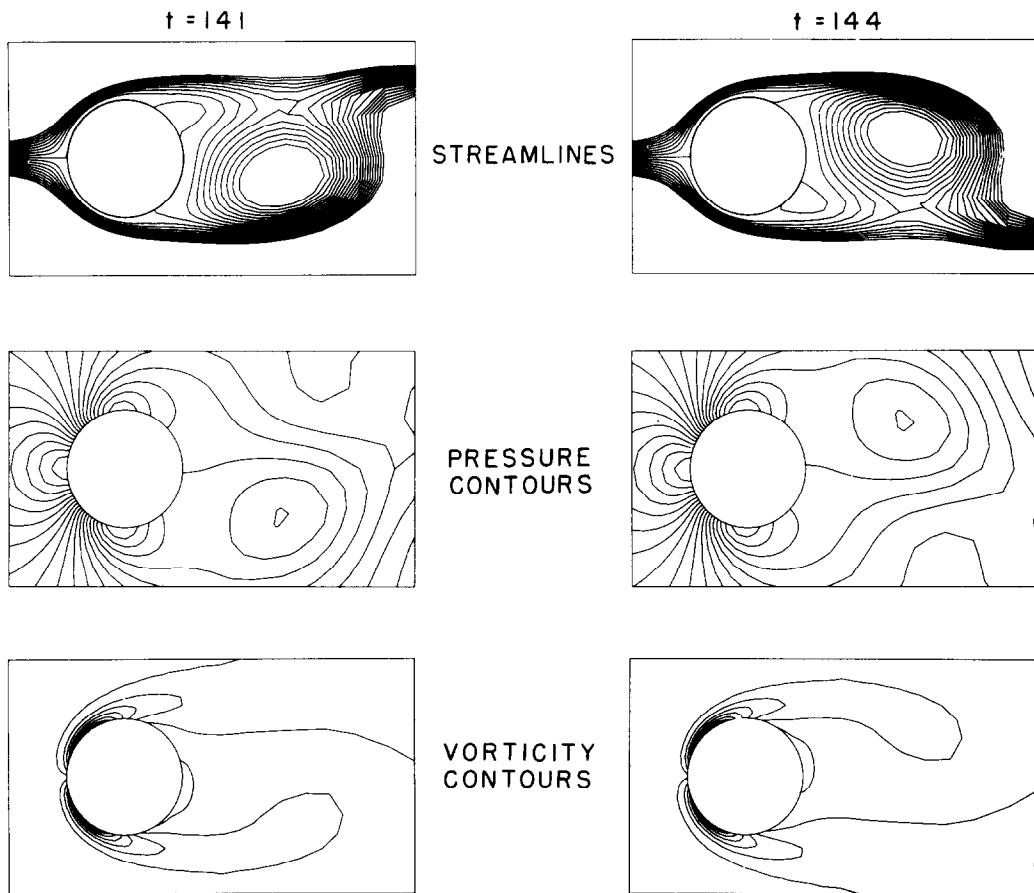


Fig. 6.13. Streamlines, pressure contours and vorticity contours 1/2 cycle apart.

Table 6.1
Implicit-explicit comparison, flow past a cylinder

Method	Ref. [13], 9-node implicit Galerkin, bilinear continuous pressure	Present method, 4-node explicit 2-pass Petrov-Galerkin, implicit constant pressure
Number of velocity DOF	1700	2826
Number of pressure DOF	229	1436
Time step	~ 0.3	0.03
CPU sec/step	~ 20	1.6
Steps/vortex cycle	~ 20	200
CPU sec/vortex cycle	400	320

the Galerkin stability limit is inversely proportional to the element Reynolds number, while the petrov–Galerkin limit is only a convection condition, which is independent of Reynolds number.

(2) The effective Reynolds number of the downstream vortices would be greatly increased (cf. Fig. 4.4).

(3) The phase accuracy would suffer (cf. Fig. 4.6).

7. Conclusions

This study has focused on the development of an upwind finite element formulation which does not exhibit any of the shortcomings that have heretofore been associated with upwind techniques. The streamline upwind/Petrov–Galerkin method presented herein possesses the desirable features of both classical upwind methods and the Galerkin method.

It has the robustness of an upwind method, in that spurious wiggles are not generated, and it has the accuracy often associated with wiggle-free Galerkin solutions. The method is in no way degraded by ‘artificial diffusion’ which often afflicts other upwind schemes. It has also been shown that, in many circumstances, the Galerkin method exhibits *negative* artificial diffusion.

The success of the new method is due to two main features: (i) the streamline upwind concept, which precludes the possibility of excessive crosswind diffusion, and (ii) the consistent Petrov–Galerkin weighted residual formulation, which eliminates the artificial diffusion that plagues many classical upwind schemes. Additionally, the method is quite easy to implement, and does not require the use of higher-order, or exotic, weighting functions.

In transient analysis, it has been shown that the streamline upwind/Petrov–Galerkin method is capable of high accuracy. The multiple-iteration algorithm proposed herein exhibits the excellent phase accuracy characteristics of a consistent-mass implicit algorithm, within an explicit lumped-mass framework. The critical time step for the algorithm in convection dominated cases is based solely on a convection condition, and as a result, is independent of Peclet (or Reynolds) number. This is a considerable improvement over explicit Galerkin algorithms, for which the critical time step is inversely proportional to the Peclet number.

A new Navier–Stokes algorithm is presented, employing the streamline upwind/Petrov–Galerkin method. Several different treatments of incompressibility are incorporated into the formulation. The proposed algorithm is seen to be more efficient computationally than penalty formulations, especially in the three-dimensional case.

The example problem of flow past a cylinder demonstrates that the method is quite effective, and definitely not over-diffuse. Compared with an implicit calculation of the same problem, the present explicit velocity/implicit pressure algorithm was seen to be more economical.

The main thrust of future research should focus on improving the efficiency of the algorithm. It is hoped that finite element methods, which handle complicated geometry with relative ease, can eventually match the good finite difference methods in speed and storage requirements. The computational speed of the present Navier–Stokes algorithm is significantly faster than many previous finite element formulations, but is still somewhat slower than the best finite difference methods.

It is believed that optimized coding will significantly improve, perhaps by a factor of two or more, the speed of the present algorithm. Preliminary results of Hughes and Tezduyar [29] indicate that it may be possible to retain high accuracy without an extra iteration if the upwind parameter, \tilde{k} , is selected properly. If this approach proves successful, the computation speed is again doubled.³

The implicit Poisson matrix, \mathbf{K} , severely limits the viability of the present algorithm in large two-dimensional problems, and in almost all three-dimensional problems. In these situations, the storage requirements for the factored \mathbf{K} can be extremely large, and the time required for the forward reduction/back substitution can be very significant (cf. Figs. 5.1 and 5.2).⁴ To alleviate this problem it is necessary to develop either a matrix-split algorithm or an alternating direction method for the solution of the Poisson equation. With these improvements, large-scale three-dimensional simulations with complicated geometry will finally be possible.

Acknowledgement

We wish to gratefully acknowledge the financial support of the National Science Foundation, the Electric Power Research Institute, the California Institute of Technology and Stanford University. We are especially grateful for the computing resources made available by the Computational Fluid Dynamics Branch of the NASA Ames Research Center.

References

- [1] J.D. Atkinson and T.J.R. Hughes, Upwind finite element schemes for convective-diffusive equations, Charles Kolling Laboratory Tech. Note C-2, The University of Sydney, Sydney, N.S.W. (1977).
- [2] A.J. Baker, Research on numerical algorithms for the three-dimensional Navier–Stokes equations, Part I. Accuracy, convergence and efficiency, Tech. Rep. AFFDL-TR-79-3141, Wright-Patterson Air Force Base, OH (1979).
- [3] A. Brooks and T.J.R. Hughes, Streamline upwind/Petrov–Galerkin methods for advection dominated flows, Proc. Third Internat. Conf. on Finite Element Methods in Fluid Flow, Banff, Canada, 1980.
- [4] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comp. 22 (1968) 745.
- [5] I. Christie, D.F. Griffiths, A.R. Mitchell and O.C. Zienkiewicz, Finite element methods for second order differential equations with significant first derivatives, Internat. J. Numer. Methods Engrg. 10 (1976) 1389–1396.
- [6] J.E. Dendy, Two methods of Galerkin type achieving optimum L^2 rates of convergence for first order hyperbolics, SIAM J. Numer. Anal. 11 (1974) 637–653.
- [7] G. DeVahl Davis and G. Mallinson, An evaluation of upwind and central difference approximations by a study of recirculating flow, Computers and Fluids 4 (1976) 29–43.
- [8] J. Donea, Private Communication, 1980.
- [9] J. Donea, S. Giuliani and H. Laval, Accurate explicit finite element schemes for convective-conductive heat

³In this case the streamline upwind/Petrov–Galerkin formulation degenerates to an approach which has essential features in common with that proposed by J.K. Dukowicz and J.D. Ramshaw, Tensor viscosity method for convection in numerical fluid dynamics, J. Comput. Phys. Vol. 32 (1979) 71–79. We thank P.M. Gresho for pointing this reference out to us.

⁴P.M. Gresho has suggested solving the Poisson equation every few time steps to decrease computational cost further (private communication).

- transfer problems, in: T.J.R. Hughes, ed., Finite Element Methods for Convection Dominated Flows, AMD Vol. 34 (ASME, New York, 1979).
- [10] P.M. Gresho, S.T. Chan, R.L. Lee, C.D. Upson, Solution of the time dependent, three-dimensional incompressible Navier–Stokes equations via FEM, Lawrence Livermore Laboratory Report UCRL-85337, 1981.
 - [11] P. Gresho, R. Lee and R. Sani, Advection-dominated flows, with emphasis on the consequences of mass lumping, in: R.H. Gallagher et al., eds., Finite Elements in Fluids, Vol. 3 (Wiley, Chichester, England, 1978).
 - [12] P.M. Gresho and R.L. Lee, Don't suppress the wiggles—They're telling you something!, in: T.J.R. Hughes, ed., Finite Element Methods for Convection Dominated Flows, AMD Vol. 34 (ASME, New York, 1979).
 - [13] P.M. Gresho, R.L. Lee, S.T. Chan and R.L. Sani, Solution of the time dependent incompressible Navier–Stokes and Boussinesq equations using the Galerkin finite element method, Lawrence Livermore Laboratory Rep. UCRL-82899, 1979. Also, Lecture Notes in Mathematics, No. 771, Approximation Methods for Navier–Stokes Problems (Springer, Berlin, 1980).
 - [14] P.M. Gresho, R.L. Lee and C.D. Upson, FEM solution of the Navier–Stokes equations for vortex shedding behind a cylinder: Experiments with the four-node element, in: Proc. Third Internat. Conf. on Finite Elements in Water Resources, University of Mississippi, U.S.A., 1980.
 - [15] P.M. Gresho, R.L. Lee and R.L. Sani, On the time-dependent solution of the Navier–Stokes equations in two and three dimensions, in: C. Taylor and K. Morgan, eds., Recent Advances in Numerical Methods in Fluids, Vol. 1 (Pineridge, Swansea, 1980) pp. 27–81.
 - [16] D.F. Griffiths and A.R. Mitchell, On generating upwind finite element methods, in: T.J.R. Hughes, ed., Finite Element Methods for Convection Dominated Flows, AMD Vol. 34 (ASME, New York, 1979).
 - [17] J.C. Heinrich, P.S. Huyakorn, O.C. Zienkiewicz and A.R. Mitchell, An 'upwind' finite element scheme for two-dimensional convective transport equation, Intern. J. Numer. Methods Engrg. 11 (1977) 134–143.
 - [18] J. Heinrich and O.C. Zienkiewicz, The finite element method and 'upwinding' techniques in the numerical solution of convection dominated flow problems, in: T.J.R. Hughes, ed., Finite Element Methods for Convection Dominated Flows, AMD Vol. 34 (ASME, New York, 1979).
 - [19] T.J.R. Hughes, A simple scheme for developing 'upwind' finite elements, Internat. J. Numer. Methods Engrg. 12 (1978) 1359–1365.
 - [20] T.J.R. Hughes, Implicit-explicit finite element techniques for symmetric and nonsymmetric systems, Proc. First Internat. Conf. on Numerical Methods for Non-linear Problems, Swansea, U.K., 1980.
 - [21] T.J.R. Hughes, Recent developments in computer methods for structural analysis, Nucl. Engrg. Des. 57 (1980) 427–439.
 - [22] T.J.R. Hughes and J. Atkinson, A variational basis for 'upwind' finite elements, IUTAM Symposium on Variational Methods in the Mechanics of Solids, Northwestern University, Evanston, IL, 1978.
 - [23] T.J.R. Hughes and A. Brooks, A multidimensional upwind scheme with no crosswind diffusion, in: T.J.R. Hughes, ed., Finite Element Methods for Convection Dominated Flows, AMD Vol. 34 (ASME, New York, 1979).
 - [24] T.J.R. Hughes and A. Brooks, Galerkin/upwind finite element mesh partitions in fluid mechanics, in: J.J.H. Miller, ed., Boundary and Interior Layers—Computational and Asymptotic Methods (Boole, Dublin, 1980) pp. 103–112.
 - [25] T.J.R. Hughes and A. Brooks, A theoretical framework for Petrov–Galerkin methods with discontinuous weighting functions: Application to the streamline upwind procedure, to appear in: R.H. Gallagher, ed., Finite Elements in Fluids, Vol. 4 (Wiley, London).
 - [26] T.J.R. Hughes, W.K. Liu and A. Brooks, Review of finite element analysis of incompressible viscous flows by the penalty function formulation, J. Comput. Phys. 30 (1979) 1–60.
 - [27] T.J.R. Hughes, K.S. Pister and R.L. Taylor, Implicit-explicit finite elements in nonlinear transient analysis, Comput. Meths. Appl. Mech. Engrg. 17/18 (1979) 159–182.
 - [28] T.J.R. Hughes, R.L. Taylor and J.F. Levy, High Reynolds number, steady, incompressible flows by a finite element method, Finite Elements in Fluids, Vol. 3 (Wiley, London, 1978).
 - [29] T.J.R. Hughes and T.E. Tezduyar, Private Communication, 1981.
 - [30] C. Johnson and U. Nävert, Analysis of some finite element methods for advection-diffusion problems, Res. Rept. 80.01R, Dept. of Computer Sciences, Chalmers University of Technology and the University of Göteborg, Göteborg, Sweden, 1980.

- [31] D.W. Kelly, S. Nakazawa, O.C. Zienkiewicz and J.C. Heinrich, A note of upwinding and anisotropic balancing dissipation in finite element approximations to convective diffusion problems, *Internat. J. Numer. Methods Engrg.* 15 (1980) 1705–1711.
- [32] R.L. Lee, P.M. Gresho and R.L. Sani, Smoothing techniques for certain primitive variable solutions of the Navier–Stokes equations, *Internat. J. Numer. Methods Engrg.* 14 (1979) 1785–1804.
- [33] B.P. Leonard, A survey of finite differences of opinion on numerical muddling of the incompressible defective convection equation, in: T.J.R. Hughes, ed., *Finite Element Methods for Convection Dominated Flows*, AMD Vol. 34 (ASME, New York, 1979).
- [34] B.P. Leonard, Note on the Von Neumann stability of the explicit FTCS convective diffusion equation, *Appl. Math. Modelling* 4 (1980) 401.
- [35] K.W. Morton and J.W. Barrett, Optimal finite element methods for diffusion-convection problems, in: J.J.H. Miller, ed., *Boundary and Interior Layers—Computational and Asymptotic Methods* (Boole, Dublin, 1980) pp. 134–148.
- [36] J.T. Oden, Penalty methods and selective reduced integration for Stokesian flows, *Proc. Third Internat. Conf. on Finite Elements in Flow Problems*, Banff, Canada, 1980.
- [37] G.D. Raithby, A critical evaluation of upstream differencing applied to problems involving fluid flow, *Comput. Meths. Appl. Mech. Engrg.* (1976) 75–103.
- [38] G.D. Raithby and K.E. Torrance, Upstream-weighted differencing schemes and their application to elliptic problems involving fluid flow, *Comput. and Fluids* 2 (1974) 191–206.
- [39] W.H. Raymond and A. Garder, Selective damping in a Galerkin method for solving wave problems with variable grids, *Monthly Weather Rev.* 104 (1976) 1583–1590.
- [40] J.N. Reddy, On the mathematical theory of penalty-finite elements for Navier–Stokes equations, *Proc. Third Internat. Conf. on Finite Elements in Flow Problems*, Banff, Canada, 1980.
- [41] P.J. Roache, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1976).
- [42] R.L. Sani, B.E. Eaton, P.M. Gresho, R.L. Lee and S.T. Chan, On the solution of the time-dependent incompressible Navier–Stokes equations via a penalty Galerkin finite element method, *Lawrance Livermore Laboratory Rept. UCRL-85354*, 1981.
- [43] R.L. Sani, P.M. Gresho and R.L. Lee, On the spurious pressures generated by certain GFEM solutions of the incompressible Navier–Stokes equations, *Proc. Third Internat. Conf. on Finite Elements in Flow Problems*, Banff, Canada, 1980.
- [44] R.L. Sani, P.M. Gresho, R.L. Lee and D.F. Griffiths, The cause and cure of the spurious pressures generated by certain GFEM solutions of the incompressible Navier–Stokes equations, *Internat. J. Numer. Meths. Fluids* (1981), to appear.
- [45] S.L. Smith and C.A. Brebbia, Improved stability techniques for the solution of Navier–Stokes equations, *Appl. Math. Modelling* 1 (1977) 226–234.
- [46] R.L. Taylor, Computer procedures for finite element analysis, in: O.C. Zienkiewicz, ed., *The Finite Element Method* (McGraw-Hill, London, 3rd ed., 1977) Ch. 24.
- [47] R. Temam, *On the Theory and Numerical Analysis of the Navier–Stokes Equations* (North-Holland, Amsterdam, 1977).
- [48] T.E. Tezduyar, Ph.D. Thesis, California Institute of Technology, in preparation.
- [49] E.L. Wachspress, Isojacobian crosswind differencing, in: *Numerical Analysis*, Proc. Biennial Conf., Dundee 1977, Lecture Notes in Mathematics, Vol. 630 (Springer, Berlin, 1978) pp. 190–199.
- [50] L.B. Wahlbin, A dissipative Galerkin method for the numerical solution of first order hyperbolic equations, in: C. de Boor, ed., *Mathematical Aspects of Finite Elements in Partial Differential Equations* (Academic Press, New York, 1974) pp. 147–169.
- [51] O.C. Zienkiewicz and J.C. Heinrich, The finite element method and convection problems in fluid mechanics, in: R.H. Gallagher, O.C. Zienkiewicz, J.T. Oden, M. Morandi Cecchi and C. Taylor, eds., *Finite Elements in Fluids*, Vol. 3 (Wiley, Chichester, England, 1978) pp. 1–22.