COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 45 (1984) 217–284
NORTH-HOLLAND

FINITE ELEMENT METHODS FOR FIRST-ORDER HYPERBOLIC SYSTEMS WITH PARTICULAR EMPHASIS ON THE COMPRESSIBLE EULER EQUATIONS*

T.J.R. HUGHES
Division of Applied Mechanics, Stanford University, Stanford, CA 94305, U.S.A.

T.E. TEZDUYAR
Department of Mechanical Engineering, University of Houston, Houston, TX 77004, U.S.A.

Received 31 May 1982
Revised manuscript received 31 May 1983

A Petrov–Galerkin finite element formulation is presented for first-order hyperbolic systems of conservation laws with particular emphasis on the compressible Euler equations. Applications of the methodology are made to one- and two-dimensional steady and unsteady flows with shocks. Results obtained suggest the potential of the type of methods developed.

1. Introduction

Recently, a considerable amount of interest has focused on the numerical solution of the equations of compressible flow and in particular the compressible Euler equations. Research in this area is currently dominated by finite difference methods, especially in the United States. Several innovations have been made in recent years and presently finite difference procedures are being effectively utilized on a number of problems of engineering interest. The literature in this area is so vast that we will make no attempt to review it here. However, we do wish to point out a few developments which we believe are responsible for the success of finite difference methodology. Firstly, the compressible Euler equations possess discontinuous solutions and these have traditionally caused problems for numerical schemes. Very 'robust' finite difference methods have been developed, such as one-sided difference flux-vector splitting algorithms [1, 56, 57], which are capable of obtaining solutions to these difficult problems. Furthermore, methodology whose aim is to combine precise resolution of discontinuities while maintaining formal accuracy and the robustness necessary to obtain solutions has recently been under intense development [14, 17, 19–21, 47–51]. (It should be pointed out for the reader unfamiliar with this area that traditional numerical approaches, which create oscillations in the presence of discontinuities, are virtually useless for compressible-flow problems because the oscillations feed back by way of nonlinearities and completely change the character of the equations, frequently resulting in catastrophic divergence. Some form of

*Final report of work performed under NASA-Ames University Consortium, Interchange No. NCA-OR745-104. The second author was partially supported by Office of Naval Research Contract N00014-82-K-0335.

0045-7825/84/$3.00 © 1984, Elsevier Science Publishers B.V. (North-Holland)

'stabilization' is necessary, such as by way of artificial viscosities, and the art is to introduce just the right amount so that stability is enhanced without sacrificing accuracy.) Another attribute of finite difference methods is their simplicity which facilitates implementation and the development of efficient computer programs. For example, 'approximate factorization' schemes have been developed which possess the stability and accuracy attributes of implicit time-stepping algorithms, without engendering the enormous data base and computational effort usually associated with such methods. On the other hand, finite difference methods have traditionally been geometrically limited. Considerable progress has been made recently in extending the applicability of these methods by the development of grid mapping techniques (see e.g. [7, 55]). This has permitted application to two-dimensional flows about airfoils and some three-dimensional configurations of aerodynamic interest. However, the development of grids for geometrically and topologically intricate domains is still at the very least laborious and is often a practical impossibility. Because there is great interest in flows about complicated shapes (such as e.g. an airplane) one would anticipate that finite element methodology, due to its inherent geometric flexibility, would ultimately have a significant role to play in this area of computational fluid dynamics. We believe this to be the case, but it is often pointed out to us that, presently, finite element methods are too slow, require too large a data base, cannot handle shocks, etc. It is our response that none of these drawbacks is intrinsic, all can be corrected to at least the degree of any finite difference methodology, and, at the same time, full geometrical modeling capabilities can be retained. We anticipate that the reader may wonder: why hasn't this already been achieved? The answer is simple: compared with the amount of finite difference research on compressible flows, the amount of finite element research has been virtually insignificant. Some procedures have been developed specifically for nuclear reactor applications, see for example [11], but their applicability to high-speed flows of aerodynamic interest is not apparent. For this latter class of problems, the following finite element works may be mentioned: the variational-principle based approach of Ecer and Akay [13]; the general interpolants method of Spradley et al. [54] which merges finite element and difference ideas in a novel way; and the work of Baker [2, 3].

In this paper we describe our initial work on the development of finite element methodology for the compressible Euler equations. We are especially interested in high-speed flows with shocks. Based upon previous research on the advection–diffusion and incompressible Navier–Stokes equations [5, 6, 26–28], it seems apparent that the Galerkin finite element method, by itself, would be incapable of handling the essentially discontinuous phenomena of interest herein. (Ample evidence will be presented to confirm this assertion.) Thus we are led to consider a class of procedures, termed Petrov–Galerkin formulations, which possesses improved robustness characteristics and has performed well for us in the context of incompressible flows [6]. The particular class of Petrov–Galerkin methods which we have used are of streamline-upwind type. Johnson [35] and Nävert [46] have analyzed the streamline-upwind/Petrov–Galerkin method in the context of the multi-dimensional advection–diffusion equation and have established optimal convergence rates and a strong discontinuity capturing property, even when the discontinuity is skew to the mesh. In the developments herein we generalize the streamline-upwind/Petrov–Galerkin procedure to hyperbolic systems of conservation laws, in the sense that the present formulation reduces to the one for the advection–diffusion equation presented in [6, 28]. (We qualify the use of the terminology 'generalize' in the foregoing manner because there are a number of possible modes of generalization, the present approach just representing one.) This enables a degree of robustness to be introduced beyond that of the standard Galerkin method, which is a crucial ingredient in the present applications. The basic weighted residual framework is described in Section 2. The semi-discrete matrix equations and transient algorithms are presented in Section 3. In the formulation developed several options are available in the definition of the weighting function. These are described in Section 4. A stability and accuracy analysis of several of the fully-discrete schemes is presented in Section 5 for the one-dimensional linear model problem of pure advection. In Section 6, a number of one-dimensional calculations are presented. These encompass linear and nonlinear, steady and unsteady hyperbolic systems. Some two-dimensional Euler flow calculations are presented in Section 7. In many respects the results obtained are gratifying given the initiatory character of this endeavor. However, we are of the opinion that significant improvement can still be made. Further amplification on this point is presented in the concluding remarks made in Section 8.

2. Multi-dimensional systems of conservation laws

2.1. Preliminaries

Let n_{sd} denote the number of space dimensions. Let Ω be an open region in $\mathbb{R}^{n_{\mathrm{sd}}}$ with piecewise smooth boundary Γ. Let $\boldsymbol{x} = \{x_i\}$, $i = 1, 2, \ldots, n_{\mathrm{sd}}$, denote a general point in Ω and let $\boldsymbol{n} = \{n_i\}$ be the unit outward normal vector to Γ. We assume Γ admits the following decomposition:

$$\Gamma = \overline{\Gamma_{\mathscr{G}} \cup \Gamma_{\mathscr{H}}}\,, \tag{2.1}$$

$$\emptyset = \Gamma_{\mathscr{G}} \cap \Gamma_{\mathscr{H}}\,, \tag{2.2}$$

where $\Gamma_{\mathscr{G}}$ and $\Gamma_{\mathscr{H}}$ are subsets of Γ. The superposed bar in (2.1) represents set closure and $\emptyset$, in (2.2), denotes the empty set. The significance of $\Gamma_{\mathscr{G}}$ and $\Gamma_{\mathscr{H}}$ will be made apparent in the sequel.

The summation convention on repeated indices is assumed in force and the subscript n is used to denote the normal component of a vector (e.g., if $n_{\mathrm{sd}} = 3$, then $\boldsymbol{F}_n = \boldsymbol{F}_j n_j = \boldsymbol{F}_1 n_1 + \boldsymbol{F}_2 n_2 + \boldsymbol{F}_3 n_3$). A comma is used to denote partial differentiation (e.g., $\boldsymbol{F}_{j,j} = \partial \boldsymbol{F}_j / \partial x_j$) and t denotes time. The Kronecker delta is denoted by δ_{ij}; if $i = j$, then $\delta_{ij} = 1$, otherwise $\delta_{ij} = 0$.

Consider a discretization of Ω into element subdomains Ω^e, $e = 1, 2, \ldots, n_{\mathrm{el}}$, where n_{el} is the number of elements. Each Ω^e is taken to be an open set and its boundary is denoted Γ^e. We assume

$$\bar{\Omega} = \bigcup_e \bar{\Omega}^e\,, \tag{2.3}$$

$$\Gamma \subset \bigcup_e \Gamma^e\,. \tag{2.4}$$

The set $\bigcup_e \Omega^e$ will be referred to as the *element interiors*. The element boundaries, modulo Γ, play an important role in which follows. We call this set the *interior boundary*, viz.,

$$\Gamma_{\text{int}} = \bigcup_e \Gamma^e - \Gamma . \tag{2.5}$$

Two classes of functions are important in the developments which follow. The classes are distinguished by their continuity properties across Γ_{int}. It suffices to assume that all functions considered herein are smooth on the element interior. Functions of the first class are continuous across Γ_{int}. These functions are denoted by C^0. Functions of the second class are allowed to be discontinuous across Γ_{int} and are denoted by C^{-1}. The C^0 functions may be recognized as containing the standard finite element interpolations.

2.2. Systems of conservation laws

We consider the following system of m partial differential equations:

$$\boldsymbol{U}_{,t} + \boldsymbol{F}_{j,j} + \boldsymbol{G} = \boldsymbol{0} \quad \text{on } \Omega \tag{2.6}$$

where

$$\boldsymbol{U} = \boldsymbol{U}(\boldsymbol{x}, t) , \tag{2.7}$$

$$\boldsymbol{F}_j = \boldsymbol{F}_j(\boldsymbol{U}, \nabla\boldsymbol{U}, \boldsymbol{x}, t) , \tag{2.8}$$

$$\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{U}, \boldsymbol{x}, t) . \tag{2.9}$$

The vector $\boldsymbol{F}_j$ is referred to as a flux vector and $\boldsymbol{G}$ is a source term. We assume that for each $\boldsymbol{k} = \{k_i\} \in \mathbb{R}^{n_{sd}}$ there exists a matrix $\boldsymbol{S}$ such that

$$\boldsymbol{S}^{-1}(k_j\boldsymbol{A}_j)\boldsymbol{S} = \boldsymbol{\Lambda} \tag{2.10}$$

where

$$\boldsymbol{A}_j = D_1\boldsymbol{F}_j = \partial\boldsymbol{F}_j/\partial\boldsymbol{U} \tag{2.11}$$

and $\boldsymbol{\Lambda}$ is a real, diagonal matrix. If, in addition

$$D_2\boldsymbol{F}_j = \partial\boldsymbol{F}_j/\partial(\nabla\boldsymbol{U}) = 0 , \tag{2.12}$$

then (2.6) is said to be a *first-order hyperbolic system*; if $\boldsymbol{A}_j = \boldsymbol{A}_j^{\text{t}}$, then (2.6) is called a *symmetric hyperbolic system*.

Let the total flux be decomposed into *partial fluxes*, $\boldsymbol{F}_j^{(1)}$ and $\boldsymbol{F}_j^{(2)}$, as follows:

$$\boldsymbol{F}_j(\boldsymbol{U}, \nabla\boldsymbol{U}, \boldsymbol{x}, t) = \boldsymbol{F}_j^{(1)}(\boldsymbol{U}, \boldsymbol{x}, t) + \boldsymbol{F}_j^{(2)}(\boldsymbol{U}, \nabla\boldsymbol{U}, \boldsymbol{x}, t) . \tag{2.13}$$

Thus, if dissipative mechanisms are present, as evidenced by the appearance of the argument $\nabla\boldsymbol{U}$, we assume they are confined to the partial flux $\boldsymbol{F}_j^{(2)}$.

The initial/boundary-value problem for (2.6) consists of finding a function $\boldsymbol{U}$ which satisfies (2.6), the initial condition

$$\boldsymbol{U}(\boldsymbol{x}, 0) = \boldsymbol{U}_0(\boldsymbol{x}) , \tag{2.14}$$

where U_0 is a given function of $\boldsymbol{x} \in \Omega$, and appropriately specified boundary conditions. In this paper, for sake of simplicity, we shall limit attention to boundary conditions of the following form:

(1) *Dirichlet type.* In this case we assume

$$\partial U = \mathcal{G} \quad \text{on } \Gamma_{\mathcal{G}} \tag{2.15}$$

where ∂ is a boundary operator and $\mathcal{G}$ is a prescribed function.

(2) *Neumann type.* In this case we assume

$$-\boldsymbol{F}^{(2)}_n = \mathcal{H} \quad \text{on } \Gamma_{\mathcal{H}} \tag{2.16}$$

where $\mathcal{H}$ is a prescribed function on $\Gamma_{\mathcal{H}}$. This can be interpreted as a partial flux boundary condition or, if $\boldsymbol{F}^{(1)}_j = \boldsymbol{0}$, a total flux boundary condition.

Finally, we allow for no boundary condition on $\Gamma_{\mathcal{H}}$. This is viewed as a special case of (2.16) in which both $\boldsymbol{F}^{(2)}_j$ and $\mathcal{H}$ are assumed to be identically zero.

Clearly, various combinations of the above boundary conditions may also be specified on portions of Γ, although this is not explicitly spelled out in the sequel.

2.3. Weighted residual formulation

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

$$[\boldsymbol{F}_n] = (\boldsymbol{F}^+_j - \boldsymbol{F}^-_j)n^+_j = \boldsymbol{F}^+_j n^+_j + \boldsymbol{F}^-_j n^-_j\,. \tag{2.17}$$

As may be readily verified from (2.17), the jump is invariant with respect to reversing the plus and minus designations.

Throughout, we shall assume that trial solutions, $\boldsymbol{U}$, satisfy $\partial \boldsymbol{U} = \mathcal{G}$ on $\Gamma_{\mathcal{G}}$ and weighting functions, $\boldsymbol{W}$, satisfy $\partial \boldsymbol{W} = \boldsymbol{0}$ on $\Gamma_{\mathcal{G}}$. Thus all Dirichlet type boundary conditions are treated as *essential boundary conditions* in the present formulation.

The variational equation is assumed to take the form:

$$\int_{\Omega} \{\boldsymbol{W}^{t}(\boldsymbol{U}_{,t} + \boldsymbol{F}^{(1)}_{j,j} + \boldsymbol{G}) - \boldsymbol{W}^{t}_{,j}\boldsymbol{F}^{(2)}_j\}\,\mathrm{d}\Omega$$
$$+ \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tilde{\boldsymbol{P}}^{t}(\boldsymbol{U}_{,t} + \boldsymbol{F}_{j,j} + \boldsymbol{G})\,\mathrm{d}\Omega = \int_{\Gamma_{\mathcal{H}}} \boldsymbol{W}^{t}\mathcal{H}\,\mathrm{d}\Gamma\,. \tag{2.18}$$

In (2.18), $\boldsymbol{U}$ and $\boldsymbol{W}$ are assumed to be taken from the same class of typical C^0 finite element interpolations; $\tilde{\boldsymbol{P}}$ is a C^{-1} perturbation to the weighting function.

The Euler–Lagrange conditions emanating from (2.18) may be deduced by way of integration by parts:

$$0 = \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \boldsymbol{W}^{\mathrm{t}}(\boldsymbol{U}_{,t} + \boldsymbol{F}_{i,i} + \boldsymbol{G})\,\mathrm{d}\Omega - \int_{\Gamma_h} \boldsymbol{W}^{\mathrm{t}}(\boldsymbol{F}_n^{(2)} + \mathcal{H})\,\mathrm{d}\Gamma - \int_{\Gamma_{\mathrm{int}}} \boldsymbol{W}^{\mathrm{t}}[\boldsymbol{F}_n^{(2)}]\,\mathrm{d}\Gamma . \tag{2.19}$$

From (2.19) we see that the Euler–Lagrange equations are (2.6) restricted to the element interiors, (2.16), and the (partial) flux continuity condition across interelement boundaries, namely,

$$[\boldsymbol{F}_n^{(2)}] = \boldsymbol{0} . \tag{2.20}$$

Note that (2.16) is a *natural boundary condition.*

REMARK 2.1. Note that if $\tilde{\boldsymbol{P}} = \boldsymbol{0}$ we have a *Galerkin* weighted residual formulation; if $\tilde{\boldsymbol{P}} \neq \boldsymbol{0}$ we have a *Petrov–Galerkin* formulation. The modified weighting function, that is

$$\tilde{\boldsymbol{W}} = \boldsymbol{W} + \tilde{\boldsymbol{P}} , \tag{2.21}$$

is confined to the element interiors and thus does not affect boundary or continuity conditions. In the present work $\tilde{\boldsymbol{P}}$ is assumed to take on the following form: $\tilde{\boldsymbol{P}} = \boldsymbol{T}_i \boldsymbol{W}_{,i}$ where $\boldsymbol{T}_i$ is an $m \times m$ matrix. Further elaboration of the structure of $\boldsymbol{T}_i$ is presented in Section 4.

REMARK 2.2. The preceding formulation generalizes [28] (which was restricted to the linear advection–diffusion equation) to systems of conservation laws. A related but somewhat different formulation for the incompressible Navier–Stokes equations is described in [6].

The following examples will illustrate potential applications of the preceding formulation.

2.4. Examples

EXAMPLE 2.3. The scalar linear advection–diffusion equation. In this case we have

$$\phi_{,t} + \sigma_{j,j} - f = 0 , \tag{2.22}$$

where

$$\sigma_j = \sigma_j^{\mathrm{a}} + \sigma_j^{\mathrm{d}} \qquad \text{(total flux)} , \tag{2.23}$$

$$\sigma_j^{\mathrm{a}} = \sigma_j^{\mathrm{a}}(\phi) = u_j \phi \qquad \text{(advective flux)} , \tag{2.24}$$

$$\sigma_j^{\mathrm{d}} = \sigma_j^{\mathrm{d}}(\nabla\phi) = -k_{jk}\phi_{,k} \qquad \text{(diffusive flux)} . \tag{2.25}$$

In the above, f is a source term, u_j is the flow velocity, and k_{jk} is the diffusivity. Each of f, u_j and k_{jk} is assumed to be a given function of $\boldsymbol{x}$ and t.

If Γ_g and Γ_h decompose the boundary such that

$$\Gamma = \overline{\Gamma_g \cup \Gamma_h} , \tag{2.26}$$

$$\emptyset = \Gamma_g \cap \Gamma_h , \tag{2.27}$$

then we shall assume Dirichlet conditions on $\Gamma_{\mathscr{g}}$, that is,

$$\phi = \mathscr{g} \quad \text{on } \Gamma_{\mathscr{g}} \tag{2.28}$$

where $\mathscr{g}$ is a given function defined on $\Gamma_{\mathscr{g}}$. The three possibilities on $\Gamma_{\mathscr{h}}$ are:

(1) *total flux boundary condition*:

$$-\sigma_n = \mathscr{h} \quad \text{on } \Gamma_{\mathscr{h}} \tag{2.29}$$

where $\mathscr{h}$ is a given function defined on $\Gamma_{\mathscr{h}}$;

(2) *diffusive flux boundary condition*:

$$-\sigma_n^{\mathrm{d}} = \mathscr{h} \quad \text{on } \Gamma_{\mathscr{h}}\ ; \tag{2.30}$$

and, finally,

(3) *no boundary condition on $\Gamma_{\mathscr{h}}$.*

This last condition occurs in cases of pure advection in which $\Gamma_{\mathscr{h}}$ is defined to be that part of Γ on which $u_n \geqslant 0$.

The initial condition is

$$\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x})\,, \tag{2.31}$$

where ϕ_0 is a given function of $\boldsymbol{x} \in \Omega$.

The linear advection–diffusion equation may be brought within the general framework by the following definitions:

$$m = 1\,, \tag{2.32}$$

$$\boldsymbol{U} = \phi\,, \tag{2.33}$$

$$\boldsymbol{W} = w\,, \tag{2.34}$$

$$\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{p}}\,, \tag{2.35}$$

$$\boldsymbol{F}_j = \sigma_j\,, \tag{2.36}$$

$$\boldsymbol{G} = -\mathscr{f}\,, \tag{2.37}$$

$$\partial = 1\,, \tag{2.38}$$

$$\mathscr{G} = \mathscr{g}\,, \tag{2.39}$$

$$\mathscr{H} = \mathscr{h}\,, \tag{2.40}$$

$$\boldsymbol{U}_0 = \phi_0\,. \tag{2.41}$$

To attain the various boundary conditions, $\boldsymbol{F}_j^{(1)}$ and $\boldsymbol{F}_j^{(2)}$ need to be set as follows:

(1) *total flux boundary condition*:

$$\boldsymbol{F}_j^{(1)} = \boldsymbol{0}\,, \qquad \boldsymbol{F}_j^{(2)} = \boldsymbol{F}_j = \sigma_j; \tag{2.42}$$

(2) *diffusive flux boundary condition*:

$$\boldsymbol{F}_j^{(1)} = \sigma_j^{\mathrm{a}}, \qquad \boldsymbol{F}_j^{(2)} = \sigma_j^{\mathrm{d}}; \tag{2.43}$$

(3) *no boundary condition on* Γ_h:

$$\boldsymbol{F}_j^{(1)} = \boldsymbol{F}_j = \sigma_j = \sigma_j^{\mathrm{a}}, \qquad \boldsymbol{F}_j^{(2)} = \sigma_j^{\mathrm{d}} = 0, \qquad \mathscr{H} = h = 0. \tag{2.44}$$

We have found the diffusive flux boundary condition to be very effective in numerically simulating outflow conditions (see, e.g., [6]). Cohen [10] has employed the total flux condition successfully in certain problems of oil reservoir simulation.

EXAMPLE 2.4. The compressible Euler equations (*inviscid gas dynamics*). The equations are:

$$\rho_{,t} + (\rho u_j)_{,j} = 0 \qquad \text{(continuity)}, \tag{2.45}$$

$$(\rho u_i)_{,t} + (\rho u_j u_i)_{,j} = \sigma_{ij,j} + f_i \qquad \text{(momentum)}, \tag{2.46}$$

$$(\rho e)_{,t} + (\rho u_j e)_{,j} = r + f_i u_i + (\sigma_{ij} u_i)_{,j} \qquad \text{(energy)}, \tag{2.47}$$

where

$$\sigma_{ij} = -p\delta_{ij}, \qquad p = \hat{p}(\rho, \varepsilon), \tag{2.48}$$

$$e = \varepsilon + \tfrac{1}{2}u^2, \qquad u = \|\boldsymbol{u}\| = (u_i u_i)^{1/2}, \tag{2.49}$$

in which ρ is the density, u_j is the velocity, σ_{ij} is the Cauchy stress tensor, f_i is the prescribed body force (per unit volume), e is the total energy density, r is the heat supply (per unit volume), p is the pressure, and ε is the internal energy density.

These equations may be put into the format of a hyperbolic system of conservation laws by employing the following definitions:

$$m = n_{\mathrm{sd}} + 2, \tag{2.50}$$

$$U_1 = \rho, \tag{2.51}$$

$$U_{j+1} = \rho u_j, \quad 1 \leq j \leq n_{\mathrm{sd}}, \tag{2.52}$$

$$U_m = \rho e, \tag{2.53}$$

$$\boldsymbol{F}_j = (U_{j+1}/U_1)\boldsymbol{U} + \left\{ \begin{matrix} 0 \\ p\boldsymbol{\delta}_j \\ pU_{j+1}/U_1 \end{matrix} \right\}, \tag{2.54}$$

$$p = \hat{p}\Big(U_1, U_m/U_1 - \tfrac{1}{2}\Big(\sum_{i=1}^{n_{\mathrm{sd}}} U_{i+1}^2\Big)\Big/ U_1^2\Big), \tag{2.55}$$

$$\boldsymbol{\delta}_j = \left\{ \begin{matrix} \delta_{j1} \\ \vdots \\ \delta_{jn_{\mathrm{sd}}} \end{matrix} \right\}, \tag{2.56}$$

$$\boldsymbol{G} = -\left\{ \begin{matrix} 0 \\ \boldsymbol{f} \\ r + f_i u_i \end{matrix} \right\}. \tag{2.57}$$

Consideration of the characteristics and particular physical situation lead to appropriate boundary condition specifications. A variety of Dirichlet type conditions (2.15) may be envisioned. Three potentially useful boundary conditions on $\Gamma_{\mathscr{H}}$ may be set up as follows:

(1) *total flux boundary condition*:

$$\boldsymbol{F}_j^{(1)} = \boldsymbol{0}\,, \qquad \boldsymbol{F}_j^{(2)} = \boldsymbol{F}_j\,; \tag{2.58}$$

(2) *pressure ('traction') boundary condition.* Pressure may be specified as a natural boundary condition by selecting

$$\boldsymbol{F}_j^{(1)} = (U_{j+1}/U_1)\boldsymbol{U}\,, \tag{2.59}$$

$$\boldsymbol{F}_j^{(2)} = \begin{Bmatrix} 0 \\ p\boldsymbol{\delta}_j \\ pU_{j+1}/U_1 \end{Bmatrix}, \tag{2.60}$$

and

$$\mathscr{H} = -\begin{Bmatrix} 0 \\ \boldsymbol{\mathscr{h}}\boldsymbol{n} \\ \mathscr{h}u_n \end{Bmatrix} \tag{2.61}$$

where $\mathscr{h}$ is the prescribed value of pressure defined on $\Gamma_{\mathscr{H}}$;

(3) *no boundary condition on* $\Gamma_{\mathscr{H}}$:

$$\boldsymbol{F}_j^{(1)} = \boldsymbol{F}_j\,, \qquad \boldsymbol{F}_j^{(2)} = \boldsymbol{0}\,, \qquad \mathscr{H} = 0\,. \tag{2.62}$$

In the present work we have employed the formulation defined by (2.62). In modelling outflow conditions, (2.62) appears to be effective although pressure specification may be more appropriate in some circumstances.

It is interesting to observe that if $\hat{p}(\rho, \varepsilon) = \rho f(\varepsilon)$, where f is an arbitrary function, then $\boldsymbol{F}_j(\boldsymbol{U})$ is a homogeneous function of degree 1, that is $\boldsymbol{F}_j(\alpha\boldsymbol{U}) = \alpha\boldsymbol{F}_j(\boldsymbol{U})$ for all $\alpha \in \mathbb{R}$. In this case it follows that $\boldsymbol{F}_j = \boldsymbol{A}_j\boldsymbol{U}$. The individual partial fluxes defined by (2.59) and (2.60) also are homogeneous functions of degree 1. For example, this is the case for a perfect gas defined by $\hat{p}(p, \varepsilon) = (\gamma - 1)\rho\varepsilon$, $\gamma \in \mathbb{R}$.

REMARK 2.5. The full, compressible Navier–Stokes equations, including thermal effects, can also be subsumed by the general weighted residual format of (2.18). Likewise, various natural boundary conditions can be built into the formulation. However, this lies outside the scope of the present paper.

REMARK 2.6. Discretization of (2.18) is carried out by expanding $\boldsymbol{U}$ and $\boldsymbol{W}$ in terms of a set of finite element basis, or shape, functions. For example, the expression for $\boldsymbol{U}$ might take the form

$$\boldsymbol{U}(\boldsymbol{x}, t) = \sum_B N_B(\boldsymbol{x})\boldsymbol{U}_B(t) \tag{2.63}$$

where B is a nodal index, N_B is the shape function associated with node B, and U_B is the value of U at node B. This leads to a weighted residual formulation in the strict sense. Certain modifications of the basic formulation—such as reduced integration or use of lower-order interpolants—are interesting from the standpoints of efficiency and, occasionally, lead to improved accuracy. We have been fond of the use of reduced integration techniques in our finite element work and have experimented with them in the present context.

The use of interpolants of $\boldsymbol{F}_j$ and $\boldsymbol{G}$ has an interesting consequence. It produces schemes which are reminiscent of, and at the same time generalize, classical 'conservative differencing schemes'. The basic idea is to approximate $\boldsymbol{F}_j$ and $\boldsymbol{G}$ by expansions in terms of shape functions and the *nodal values* of $\boldsymbol{F}_j$ and $\boldsymbol{G}$, respectively. For example, the expansion for $\boldsymbol{F}_j$ might take the following form:

$$\boldsymbol{F}_j(\boldsymbol{x}, t) = \sum_B N_B(\boldsymbol{x})\boldsymbol{F}_{jB}(t) \tag{2.64}$$

where

$$\boldsymbol{F}_{jB}(t) = \boldsymbol{F}_j(\boldsymbol{U}_B(t), (\nabla \boldsymbol{U})_B(t), \boldsymbol{x}_B, t)\,, \tag{2.65}$$

$$(\nabla \boldsymbol{U})_B(t) = \sum_C \nabla N_C(\boldsymbol{x}_B)\boldsymbol{U}_C(t)\,, \tag{2.66}$$

and likewise for $\boldsymbol{G}$. Christie et al. [9] have proposed and examined schemes of this kind. They term them 'product approximations'. Spradley et al. [54] have also adopted this idea in their 'general interpolants method'. Various other interesting finite element and finite difference concepts are synthesized in their approach. Fletcher [15] terms the use of interpolants the 'group approximation' and has demonstrated the efficiency of the procedure.

The use of reduced integration and/or the use of interpolants simplifies element calculations. Their relative merits, from the standpoints of accuracy, stability, shock structure, etc., do not seem certain at this point and thus further investigations are warranted.

REMARK 2.7. One could also employ *different* shape functions for the individual components of $\boldsymbol{U}$. This idea becomes important in 'constrained' cases, such as incompressibility (see, e.g., [58] and references therein). Similar considerations need to be made if compressible flow algorithms are to be exercised at very low Mach numbers.

REMARK 2.8. The development of so-called finite volume techniques also emanates from integral forms of the conservation equations (see e.g. [8]). These techniques, which are often thought of as finite difference methods, have essential features in common with the integral difference methods originated at Lawrence Livermore National Laboratory (see e.g. [63]). More recently it has been shown how to derive the latter class of methods via finite element/reduced integration concepts (see [18] for a survey of these ideas). We thus anticipate that the finite volume method will also find a place within the finite element hierarchy.

3. Semi-discrete equations and transient algorithms

3.1. Semi-discrete equations

Spatial discretization of the weighted residual equation (2.18) via finite elements leads to the following semi-discrete system of ordinary differential equations:

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}\boldsymbol{v} = \boldsymbol{F} \tag{3.1}$$

where $\boldsymbol{M} = \boldsymbol{M}(\boldsymbol{v}, t)$ is the generalized 'mass' matrix, $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{v}, t)$ is the generalized convection matrix, $\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{v}, t)$ is the force vector, $\boldsymbol{v}$ is the vector of (unknown) nodal values of $\boldsymbol{U}$, and a superposed dot denotes time differentiation. The initial-value problem for (3.1) consists of finding a function $\boldsymbol{v} = \boldsymbol{v}(t)$ satisfying (3.1) and the initial condition

$$\boldsymbol{v}(0) = \boldsymbol{v}_0 \tag{3.2}$$

where $\boldsymbol{v}_0$ is determined from (2.14).

The arrays in (3.1) are assembled from element contributions:

$$\boldsymbol{M} = \operatorname*{\mathsf{A}}_{e=1}^{n_{\mathrm{el}}} (\boldsymbol{m}^e) , \tag{3.3}$$

$$\boldsymbol{m}^e = [\boldsymbol{m}^e_{ab}] , \tag{3.4}$$

$$\boldsymbol{m}^e_{ab} = \int_{\Omega^e} (N_a \boldsymbol{I} + N_{a,i}\boldsymbol{T}^{t}_{i}) N_b \, \mathrm{d}\Omega , \tag{3.5}$$

$$\boldsymbol{C} = \operatorname*{\mathsf{A}}_{e=1}^{n_{\mathrm{el}}} (\boldsymbol{c}^e) , \tag{3.6}$$

$$\boldsymbol{c}^e = [\boldsymbol{c}^e_{ab}] , \tag{3.7}$$

$$\boldsymbol{c}^e_{ab} = \int_{\Omega^e} (N_a \boldsymbol{I} + N_{a,i}\boldsymbol{T}^{t}_{i}) \boldsymbol{A}_j N_{b,j} \, \mathrm{d}\Omega , \tag{3.8}$$

$$\boldsymbol{F} = \operatorname*{\mathsf{A}}_{e=1}^{n_{\mathrm{el}}} (\boldsymbol{f}^e) , \tag{3.9}$$

$$\boldsymbol{f}^e = \{\boldsymbol{f}^e_a\} , \tag{3.10}$$

$$\boldsymbol{f}^e_a = \int_{\Omega^e} (N_a \boldsymbol{I} + N_{a,i}\boldsymbol{T}^{t}_{i}) \boldsymbol{G} \, \mathrm{d}\Omega - \sum_{b=1}^{n_{\mathrm{en}}} (\boldsymbol{m}^e_{ab}\dot{\boldsymbol{g}}^e_b + \boldsymbol{c}^e_{ab}\boldsymbol{g}^e_b) , \tag{3.11}$$

where A represents the finite element assembly operator; a and b are (local) element node numbers; $1 \leqslant a, b \leqslant n_{\mathrm{en}}$ where n_{en} is the number of nodes for the element under consideration; N_a is the element shape function associated with node a; $\boldsymbol{I}$ is the $m \times m$ identity matrix; and $\boldsymbol{g}^e_b$ is a vector which contains the boundary condition data emanating from (2.15). The dimensions of the nodal arrays $\boldsymbol{m}^e_{ab}$ and $\boldsymbol{c}^e_{ab}$ are $m \times m$, and the dimension of $\boldsymbol{f}^e_a$ and $\boldsymbol{g}^e_b$ are $m \times 1$.

3.2. Transient algorithms

We first consider a family of one-step implicit methods defined by

$$\boldsymbol{M}_{n+1}\boldsymbol{a}_{n+1} + \boldsymbol{C}_{n+1}\boldsymbol{v}_{n+1} = \boldsymbol{F}_{n+1} , \tag{3.12}$$

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \Delta t \boldsymbol{a}_{n+\alpha} \tag{3.13}$$

where

$$M_{n+1} = M(v_{n+1}, t_{n+1}), \tag{3.14}$$

$$C_{n+1} = C(v_{n+1}, t_{n+1}), \tag{3.15}$$

$$F_{n+1} = F(v_{n+1}, t_{n+1}), \tag{3.16}$$

$$a_{n+\alpha} = (1-\alpha)a_n + \alpha a_{n+1}. \tag{3.17}$$

In the above, Δt is the time step, n is the step number, and α is a parameter which determines stability and accuracy properties.

The starting value, a_0, may be determined from

$$M_0 a_0 = F_0 - C_0 v_0 \tag{3.18}$$

where

$$M_0 = M(v_0, 0), \tag{3.19}$$

$$C_0 = C(v_0, 0), \tag{3.20}$$

$$F_0 = F(v_0, 0). \tag{3.21}$$

The foregoing algorithm is sometimes referred to as the generalized trapezoidal method. Its behavior in nonlinear problems is discussed in [25].

A general family of predictor/multi-corrector algorithms, based on the preceding implicit methods, is implemented as follows:

Step 1. $i = 0$ (i is the iteration counter). (3.22)

Step 2. $v_{n+1}^{(0)} = v_n + \Delta t(1-\alpha)a_n$. (3.23)

Step 3. $a_{n+1}^{(0)} = 0$. (3.24)

} (predictor phase)

Step 4. $R = F_{n+1}^{(i)} - M_{n+1}^{(i)} a_{n+1}^{(i)} - C_{n+1}^{(i)} v_{n+1}^{(i)}$ (residual force). (3.25)

Step 5. $M^* \Delta a = R$ (M^* is the 'effective mass'). (3.26)

Step 6. $a_{n+1}^{(i+1)} = a_{n+1}^{(i)} + \Delta a$. (3.27)

Step 7. $v_{n+1}^{(i+1)} = v_{n+1}^{(i)} + \alpha\, \Delta t\, \Delta a$. (3.28)

} (corrector phase)

If additional iterations are to be performed, i is replaced by $i+1$, and calculations resume with Step 4. Either a fixed number of iterations may be performed or iterating may be continued until R satisfies a convergence condition. When the iterative phase is completed the solution at step $n+1$ is defined by the last iterates (i.e., $v_{n+1} = v_{n+1}^{(i+1)}$ and $a_{n+1} = a_{n+1}^{(i+1)}$). At this point n is replaced by $n+1$ and calculations for the next time step may begin.

The properties of the algorithm are strongly influenced by the choice of the effective mass. There are various possibilities. For example, a *fully implicit procedure* may be defined by taking

$$M^* = M_{n+1}^{(i)} + \alpha\, \Delta t\, C_{n+1}^{(i)} + \alpha\, \Delta t\, H_{n+1}^{(i)} \tag{3.29}$$

where

$$\boldsymbol{M}_{n+1}^{(i)} = \boldsymbol{M}(\boldsymbol{v}_{n+1}^{(i)}, t_{n+1}), \tag{3.30}$$

$$\boldsymbol{C}_{n+1}^{(i)} = \boldsymbol{C}(\boldsymbol{v}_{n+1}^{(i)}, t_{n+1}), \tag{3.31}$$

$$\boldsymbol{H}_{n+1}^{(i)} = \boldsymbol{H}(\boldsymbol{v}_{n+1}^{(i)}, t_{n+1}), \tag{3.32}$$

$$\boldsymbol{H} = \mathop{\mathrm{A}}_{e=1}^{n_{\mathrm{el}}} (\boldsymbol{h}^e), \tag{3.33}$$

$$\boldsymbol{h}^e = [\boldsymbol{h}_{ab}^e], \tag{3.34}$$

$$\boldsymbol{h}_{ab}^e = \int_{\Omega^e} (N_a \boldsymbol{I} + N_{a,i} \boldsymbol{T}_i^{\mathrm{t}}) \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} N_b \, \mathrm{d}\Omega, \tag{3.35}$$

and $\boldsymbol{h}_{ab}^e$ has dimensions $m \times m$. In general, this definition of $\boldsymbol{M}^*$ leads to a non-symmetric band-profile matrix.

An *explicit algorithm* may be constructed by taking $\boldsymbol{M}^*$ to be 'lumped' (i.e. diagonal):

$$\boldsymbol{M}^* = \boldsymbol{M}_{\mathrm{diag}}. \tag{3.36}$$

There are several schemes for obtaining suitable $\boldsymbol{M}_{\mathrm{diag}}$. In the present work we assume that the diagonal element array is defined by nodal quadrature [16]. This suffices for the simple elements employed in our applications. However, in more general cases the following approach is recommended [65]:

$$\boldsymbol{m}_{ab}^e = \begin{cases} m_a^e \boldsymbol{I} & \text{if } a = b, \\ \boldsymbol{0} & \text{if } a \neq b \end{cases} \tag{3.37}$$

where

$$m_a^e = \beta \int_{\Omega^e} N_a^2 \, \mathrm{d}\Omega, \tag{3.38}$$

$$\beta = \int_{\Omega^e} \mathrm{d}\Omega \bigg/ \left(\sum_{a=1}^{n_{\mathrm{en}}} \int_{\Omega^e} N_a^2 \, \mathrm{d}\Omega \right). \tag{3.39}$$

In the present work we confine our attention to the implicit and explicit schemes defined above. Stability and accuracy analyses for the scalar, one-dimensional linear case are presented in Section 5.

However, there are other possibilities. Implicit-explicit finite element mesh partitions [24, 31–33] may prove useful, for example. Additionally, to obtain the stability properties of implicit methods, while eliminating the equation-solving burden imposed by (3.29), approximate factorization schemes may be employed. We are presently experimenting with element-by-element factorizations which are very convenient from an implementational standpoint [29, 30, 34].

4. Selection of the perturbation to the weighting function

In our work so far we have assumed that the perturbation to the weighting function takes the form

$$\tilde{\boldsymbol{P}} = \boldsymbol{T}_i \boldsymbol{W}_{,i}, \tag{4.1}$$

in which $\boldsymbol{T}_i$ is either

$$\boldsymbol{T}_i = \tau_i \boldsymbol{A}_i \quad \text{(no sum)} \tag{4.2}$$

or

$$\boldsymbol{T}_i = \tau_i \boldsymbol{A}_i^{\mathrm{t}} \quad \text{(no sum)} \tag{4.3}$$

where τ_i is a parameter selected to optimize accuracy according to some criterion. (Recall, $\boldsymbol{A}_i = \partial \boldsymbol{F}_i / \partial \boldsymbol{U}$.)

Both choices of $\boldsymbol{T}_i$ have interesting consequences. For example, assume the one-dimensional, linear, constant-coefficient case in which $\boldsymbol{F}^{(1)} = \boldsymbol{F}$ and $\boldsymbol{G} = \boldsymbol{0}$. Choose $\boldsymbol{T} = \tau \boldsymbol{A}^{\mathrm{t}}$. Then (2.18) reduces to the canonical form

$$0 = \int_{\Omega} (\bar{\boldsymbol{W}} + \tau \boldsymbol{\Lambda} \bar{\boldsymbol{W}}_{,x})^{\mathrm{t}} (\bar{\boldsymbol{U}}_{,t} + \boldsymbol{\Lambda} \bar{\boldsymbol{U}}_{,x})\, \mathrm{d}\Omega \tag{4.4}$$

where $\bar{\boldsymbol{W}} = \boldsymbol{S}^{\mathrm{t}} \boldsymbol{W}$ and $\bar{\boldsymbol{U}} = \boldsymbol{S}^{-1} \boldsymbol{U}$. Thus (4.4) is equivalent to a system of *uncoupled* scalar equations. Scalar equations of this form are extensively analyzed in Section 5.

We have performed numerical experiments with both (4.2) and (4.3), and also with the definition of τ_i. Presently we favor (4.3) over (4.2) due to its superior behavior in nonlinear problems of interest. This will be illustrated in Sections 6 and 7.

Frequently, τ_i is selected independent of direction, that is $\tau_i = \tau$. This assumption has proved successful in application to the advection–diffusion equation for which (4.1)–(4.3) reduce to

$$\tilde{p} = \tau u_i w_{,i}\,. \tag{4.5}$$

Equation (4.5) leads to the so-called *streamline-upwind/Petrov–Galerkin* formulation originated by Hughes and Brooks [28]. This formulation has been analyzed by Johnson [35] and Nävert [46]. See also [12, 36, 45] for related developments.

4.1. Selection of τ_i

As yet no universal scheme has been formulated to select τ_i. The following choices have been investigated thus far:

(1) *Temporal criterion.* In conjunction with certain time-stepping algorithms, a criterion based upon the time step, Δt, proves effective:

$$\tau_i = \tau = F\alpha\, \Delta t\,, \quad 1 \leqslant i < n_{\mathrm{sd}}\,. \tag{4.6}$$

In (4.6), α is an algorithmic parameter and F is a parameter which enables us to adjust the magnitude of τ_i for various purposes (e.g., to adequately handle shock-wave phenomena). The value of α is usually set to $\frac{1}{2}$ in cases in which we are interested in time accuracy, whereas it is set to 1 in cases in which we wish to rapidly obtain a steady flow.

Since (4.6) pertains to all elements in a mesh it is a *global* criterion. Rationale in support of the temporal criterion is provided by the following examples:

EXAMPLE 4.1. Assume $\boldsymbol{H} = \boldsymbol{0}$. If $\boldsymbol{T}_i = \tau \boldsymbol{A}_i$, and $F = 1$, then (4.6) leads to symmetry of the implicit operator $\boldsymbol{M}^*$ (see (3.32)). This can be seen from the definitions of the element contributions to $\boldsymbol{M}^*$:

$$\boldsymbol{m}^e_{ab} + \alpha\, \Delta t\, \boldsymbol{c}^e_{ab} = \int_{\Omega^e} N_a N_b \, \mathrm{d}\Omega\, \boldsymbol{I} + \alpha\, \Delta t \int_{\Omega^e} (N_{a,i} N_b \boldsymbol{A}^{\mathrm{t}}_i + N_a N_{b,j} \boldsymbol{A}_j)\, \mathrm{d}\Omega$$

$$+ (\alpha\, \Delta t)^2 \int_{\Omega^e} N_{a,i} N_{b,j} \boldsymbol{A}^{\mathrm{t}}_i \boldsymbol{A}_j \, \mathrm{d}\Omega = (\boldsymbol{m}^e_{ba} + \alpha\, \Delta t\, \boldsymbol{c}^e_{ba})^{\mathrm{t}}\,. \tag{4.7}$$

The obvious advantage in this case is the decreased storage and factorization costs. Symmetric element arrays are also advantageous in implicit-explicit finite element mesh partitions [24]. This choice also leads to an optimality condition in that for a specified residual, $\boldsymbol{R}$, the increment $\Delta \boldsymbol{a}$ is optimal with respect to the norm defined by $\boldsymbol{M}^*$. Another way of putting this is to say that the increment of $\boldsymbol{U}$ is optimized with respect to the symmetric bilinear form which generates $\boldsymbol{M}^*$. This concept of optimality is related to the following optimal steady formulation:

$$\mathscr{a}(\boldsymbol{W}, \boldsymbol{U}) = -\int_{\Omega} (\boldsymbol{W} + \tau \boldsymbol{A}_i \boldsymbol{W}_{,i})^{\mathrm{t}} \boldsymbol{G} \, \mathrm{d}\Omega \tag{4.8}$$

where $\mathscr{a}(\cdot, \cdot)$ is a symmetric, bilinear form defined by

$$\mathscr{a}(\boldsymbol{W}, \boldsymbol{U}) = \int_{\Omega} \tau^{-1} (\boldsymbol{W} + \tau \boldsymbol{A}_i \boldsymbol{W}_{,i})^{\mathrm{t}} (\boldsymbol{U} + \tau \boldsymbol{A}_j \boldsymbol{U}_{,j})\, \mathrm{d}\Omega\,, \tag{4.9}$$

and $\boldsymbol{A}_i$ and $\boldsymbol{G}$ are assumed to be independent of $\boldsymbol{U}$, that is,

$$\boldsymbol{A}_i = \boldsymbol{A}_i(x)\,, \tag{4.10}$$

$$\boldsymbol{G} = \boldsymbol{G}(x)\,. \tag{4.11}$$

EXAMPLE 4.2. The choices $\boldsymbol{T}_i = \tau \boldsymbol{A}^{\mathrm{t}}_i$, $F = 1$, $\alpha = \frac{1}{2}$, and $\boldsymbol{M}^* = \boldsymbol{M}_{\mathrm{diag}}$, leads to an explicit Lax–Wendroff type method. We shall explore this point further subsequently.

(2) *Spatial criteria.* Two spatial criteria have been employed:

(a) $$\tau_i = \tau = F\alpha\xi h/a\,, \quad 1 \leq i \leq n_{\mathrm{sd}}\,, \tag{4.12}$$

(b) $$\tau_i = F\alpha\xi h_i/a_i\,, \quad 1 \leq i \leq n_{\mathrm{sd}} \text{ (no sum)} \tag{4.13}$$

where a_i is the spectral radius of $\boldsymbol{A}_i$,

$$a = \|\boldsymbol{a}\| = (a_i a_i)^{1/2}, \tag{4.14}$$

$$h_i = 2\|\nabla x_i\|, \tag{4.15}$$

$$h = h_i a_i / a \quad (= 0 \text{ if } a = 0), \tag{4.16}$$

$$\tilde{\xi} = (\coth \beta) - 1/\beta, \tag{4.17}$$

$$\beta = ah/(2d). \tag{4.18}$$

The definition of h_i was prompted by the work of Nakazawa [45].

In (4.15), ∇ denotes the gradient operator with respect to the canonical isoparametric coordinates. The factor of 2 appears since the isoparametric parent domain is usually scaled to have a length of 2 [65]. The following examples should make the definition clear:

$$h = 2|\partial x/\partial \xi| \qquad \text{(one dimension)}, \tag{4.19}$$

$$h_i = 2[(\partial x_i/\partial \xi)^2 + (\partial x_i/\partial \eta)^2]^{1/2} \quad \text{(two dimensions)}. \tag{4.20}$$

In (4.18), d is a diffusivity coefficient which takes on the following definitions:

$$d = k_{ij} a_i a_j / a^2 \quad \text{(advection–diffusion equation)}, \tag{4.21}$$

$$d = 0 \qquad \text{(hyperbolic systems, Euler equations).} \tag{4.22}$$

Note that $d = 0$ implies $\tilde{\xi} = 1$.

The values of τ_i given by (4.12) and (4.13) depend upon the geometry and state of an individual element. In this sense, (4.12) and (4.13) are *local* criteria.

Various model equations have been employed to select $F\alpha$ in (4.12) and (4.13).

EXAMPLE 4.3. Consider the scalar model equation

$$U_{,t} + aU_{,x} = 0, \tag{4.23}$$

where a is assumed constant. Raymond and Garder [52] have shown that if

$$F\alpha = 1/\sqrt{15}, \tag{4.24}$$

then the semi-discrete equations achieve fourth-order phase accuracy. We have employed this value in transient advection–diffusion and incompressible Navier–Stokes calculations [6].

EXAMPLE 4.4. Consider the steady analog of (4.23) regularized by a diffusion term,

$$aU_{,x} = \delta U_{,xx}. \tag{4.25}$$

As $\delta \to 0$, the choice

$$F\alpha = \tfrac{1}{2} \tag{4.26}$$

leads to nodally exact solutions. A boundary-layer investigation also points to this value [27]. The general case for the advection–diffusion equation is described in [5, 6, 26–28].

5. Stability and accuracy analysis of algorithms for the one-dimensional linear hyperbolic model problem

5.1. Finite difference equations for the one-dimensional case

The finite difference equations for the algorithms of Section 3 are needed for the stability and accuracy analyses and are of interest in their own right. In explicating the finite difference equations for an internal node we have made the following assumptions: (i) linear elements are employed; (ii) h, $\boldsymbol{A}$ and $\boldsymbol{H}$ are constant; and (iii) $\boldsymbol{G}$ varies linearly over each element. Furthermore, for notational clarity we have dropped the iteration superscript i and time-step subscript $n+1$. The equations are as follows:

Implicit case:

$$\left(\tfrac{1}{2}h(\boldsymbol{I} + \alpha\,\Delta t\,\boldsymbol{H})D_r + (-\boldsymbol{T}^{\mathrm{t}} + \alpha\,\Delta t\,\boldsymbol{A} - \alpha\,\Delta t\,\boldsymbol{T}^{\mathrm{t}}\boldsymbol{H})D_1 + \alpha\,\Delta t\,\frac{2}{h}\,\boldsymbol{T}^{\mathrm{t}}\boldsymbol{A}D_2\right)\Delta\boldsymbol{a}(j) =$$

$$= (-\tfrac{1}{2}h\boldsymbol{I}D_r + \boldsymbol{T}^{\mathrm{t}}D_1)\boldsymbol{a}(j) + \left(-\boldsymbol{A}D_1 - \frac{2}{h}\,\boldsymbol{T}^{\mathrm{t}}\boldsymbol{A}D_2\right)\boldsymbol{v}(j) + (-\tfrac{1}{2}h\boldsymbol{I}D_r + \boldsymbol{T}^{\mathrm{t}}D_1)\boldsymbol{G}(j) \tag{5.1}$$

where

$$D_r\boldsymbol{v}(j) = 2(r\boldsymbol{v}(j-1) + (1-2r)\boldsymbol{v}(j) + r\boldsymbol{v}(j+1))\,, \tag{5.2}$$

$$D_1\boldsymbol{v}(j) = \tfrac{1}{2}(-\boldsymbol{v}(j-1) + \boldsymbol{v}(j+1))\,, \tag{5.3}$$

$$D_2\boldsymbol{v}(j) = -\tfrac{1}{2}(\boldsymbol{v}(j-1) - 2\boldsymbol{v}(j) + \boldsymbol{v}(j+1))\,, \tag{5.4}$$

and $\boldsymbol{v}(j) = \boldsymbol{v}(x_j)$ is the subvector of $\boldsymbol{v}$ which is associated with node number j, etc. The value of r is determined by the element quadrature rule employed. The most important cases are listed in Table 1.

Explicit case. In the explicit case, the right-hand side of (5.1) is the same, but the left-hand side simplifies to

$$h\,\Delta\boldsymbol{a}(j)\,. \tag{5.5}$$

Table 1

r	Rule
1/4	1-point Gauss
1/6	2-point Gauss (exact)
0	trapezoidal

REMARK 5.1. It is interesting to observe that even though upwind influence has been introduced via the perturbation to the weighting function defined by (4.1), the resulting difference equations are centered about node j.

REMARK 5.2. Assume $\boldsymbol{G} = \boldsymbol{0}$, $\boldsymbol{T} = \tau \boldsymbol{A}'$, $\tau = \frac{1}{2}\Delta t$ and the explicit one-pass (i.e. one-iteration) case, then from (5.1) we get

$$\boldsymbol{v}_{n+1}(j) = \left(\boldsymbol{I} - \frac{\Delta t}{h}\boldsymbol{A}D_1 - \left(\frac{\Delta t}{h}\boldsymbol{A}\right)^2 D_2\right)\boldsymbol{v}_n(j)\,. \tag{5.6}$$

Equation (5.6) defines the Lax–Wendroff method.

5.2. Stability and accuracy analysis

To assess the accuracy and stability properties of the algorithms, we consider the following model problem:

$$U_{,t} + aU_{,x} = 0 \tag{5.7}$$

where $U = U(x, t)$, and a, the convection velocity, is assumed positive and constant. Equation (5.7) has solutions of the form

$$U = \mathrm{e}^{\nu t + \mathrm{i}kx}\,, \quad \mathrm{i} = \sqrt{-1} \tag{5.8}$$

where

$$\nu = -\mathrm{i}ak\,. \tag{5.9}$$

Dissipation and phase properties may be determined from ν as follows:

$$\xi + \mathrm{i}\omega = -\nu \tag{5.10}$$

where $\xi = 0$ is the *damping coefficient* and $\omega = ak$ is the *frequency*.

The algorithms under consideration, when applied to (5.7), possess solutions of the form

$$U^h(x_j, t_n) = \mathrm{e}^{\bar{\nu}t_n + \mathrm{i}kx_j} \tag{5.11}$$

where $x_j = jh$ and $t_n = n\,\Delta t$. Corresponding to (5.10), we write

$$\bar{\xi} + \mathrm{i}\bar{\omega} = -\bar{\nu} \tag{5.12}$$

where $\bar{\xi}$ is the *algorithmic damping coefficient* and $\bar{\omega}$ is the *algorithmic frequency*. As non-dimensional error measures, we take $\bar{\xi}/\bar{\omega}$, the *algorithmic damping ratio*, and $\bar{\omega}/\omega$, the *algorithmic frequency ratio*. The algorithmic damping ratio must be greater than or equal to zero for the algorithm to be stable. The quantities $\bar{\xi}/\bar{\omega}$ and $\bar{\omega}/\omega$ can be expressed in terms of $q = kh$, the *dimensionless wave number*, $C_{\Delta t} = \Delta t\, a/h$, the *Courant number*, and $C_{2\tau} = 2\tau a/h$, the '*algorithmic Courant number*'; if $C_{2\tau} = 0$, we have the usual Galerkin spatial discretization.

The dimensionless wave number is a measure of spatial refinement, for example, $q = \frac{1}{2}\pi$ translates to having 4 elements for one full wave length. The two main parameters $C_{2\tau}$ and r, which determine the type of mass matrix, characterize the spatial semi-discretization; q and $C_{\Delta t}$ are measures of resolution; and the designations 'implicit' and 'explicit' identify the temporal discretization. Within the explicit category we always assume that the left-hand side coefficient matrix is the diagonal (i.e. 'lumped') mass matrix (see (3.36)), whereas the right-hand side is treated 'consistently' (see (3.5)). By virtue of the predictor–corrector format in which $\boldsymbol{a}_{n+1}^{(0)} = \mathbf{0}$, in an explicit algorithm the effect of the consistent mass is not felt during the first pass through (3.22)–(3.28). Thus it takes at least a two-pass explicit scheme to sense the influence of the consistent mass matrix.

The following algorithms were compared:

(Bubnov–) Galerkin algorithms. In this group $C_{2\tau} = 0$. Two cases were considered:
- GC employs consistent mass (i.e., $r = \frac{1}{6}$ in (5.1)).
- GL employs lumped mass (i.e., $r = 0$ in (5.1)).

Petrov–Galerkin algorithms. The first three members of this group are defined by the value of $C_{2\tau}$, that is
- PG($C_{2\tau} = 1$) [28];
- PG($C_{2\tau} = 2/\sqrt{15}$) [52];
- PG($C_{2\tau} = C_{\Delta t}$), which is described in Section 4 under the heading 'temporal criterion'.

The fourth member of the group,
- PG(Padé), does not fit within the weighted residual framework previously presented, but can be derived from it by assuming that the element weighting functions place all weight on the upwind node (see Fig. 1). This method can be brought within the formalism of (5.1)–(5.4) by making the following replacements:

$$D_r \boldsymbol{v}(j) \leftarrow \boldsymbol{v}(j-1) + \boldsymbol{v}(j)\,, \tag{5.13}$$

$$D_1 \boldsymbol{v}(j) \leftarrow -\boldsymbol{v}(j-1) + \boldsymbol{v}(j)\,, \tag{5.14}$$

$$D_2 \boldsymbol{v}(j) \leftarrow 0\,. \tag{5.15}$$

The name of this method derives from the corresponding Padé finite difference approximation (see [61]).

The quantities $\bar{\xi}/\bar{\omega}$ and $\bar{\omega}/\omega$ were computed and plotted for the values of $C_{\Delta t} = 0.2$, 0.4, 0.6, 0.8 and 1.0, and $q \in \,]0, \pi[$. In all cases the algorithmic parameter $\alpha = \frac{1}{2}$. The following concepts are utilized in describing the results:

Unit CFL condition: An algorithm satisfies the unit CFL condition if it produces nodally exact solutions for $C_{\Delta t} = 1$.

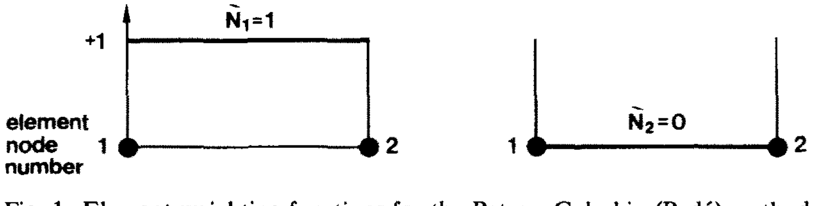

Fig. 1. Element weighting functions for the Petrov–Galerkin (Padé) method.

Order of accuracy: The behavior of the $\bar{\xi}/\bar{\omega}$ and $\bar{\omega}/\omega$ curves as $q \to 0$ reveals the order of accuracy. If either of these curves has a finite slope as $q \to 0$, then the algorithm is first-order accurate. If both curves have slopes approaching zero as $q \to 0$, then the algorithm is at least second-order accurate.

Unconditional stability: If no time step restriction is required to guarantee stability this is called unconditional stability. Otherwise, the stability is said to be conditional.

In the computer-plotted results, $C_{\Delta t}$ is written 'CDT' and $C_{2\tau}$ is written 'C2T'.

Implicit algorithms. Fig. 2 shows the frequency ratio for the implicit GC, GL and PG(Padé) algorithms. These algorithms are unconditionally stable, have no amplitude error (i.e. $\bar{\xi}/\bar{\omega} = 0$), and are second-order accurate. GC is more accurate than GL for finite q. PG(Padé) satisfies the unit CFL condition. For finite q, GC and GL become more accurate as $C_{\Delta t}$ decreases; the opposite is true for PG(Padé).

Fig. 3 shows the algorithmic damping ratio and the frequency ratio for the implicit

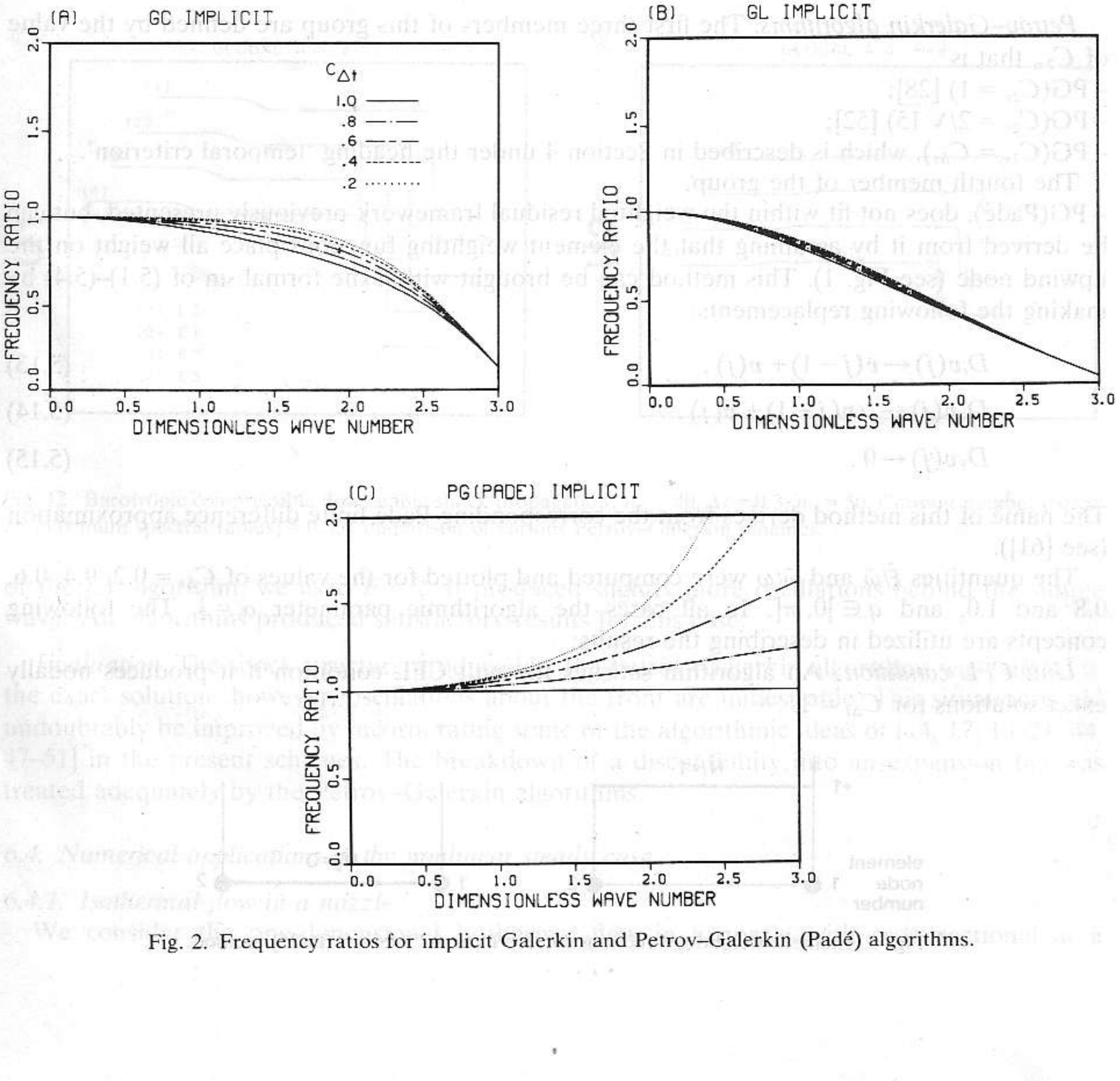

Fig. 2. Frequency ratios for implicit Galerkin and Petrov–Galerkin (Padé) algorithms.

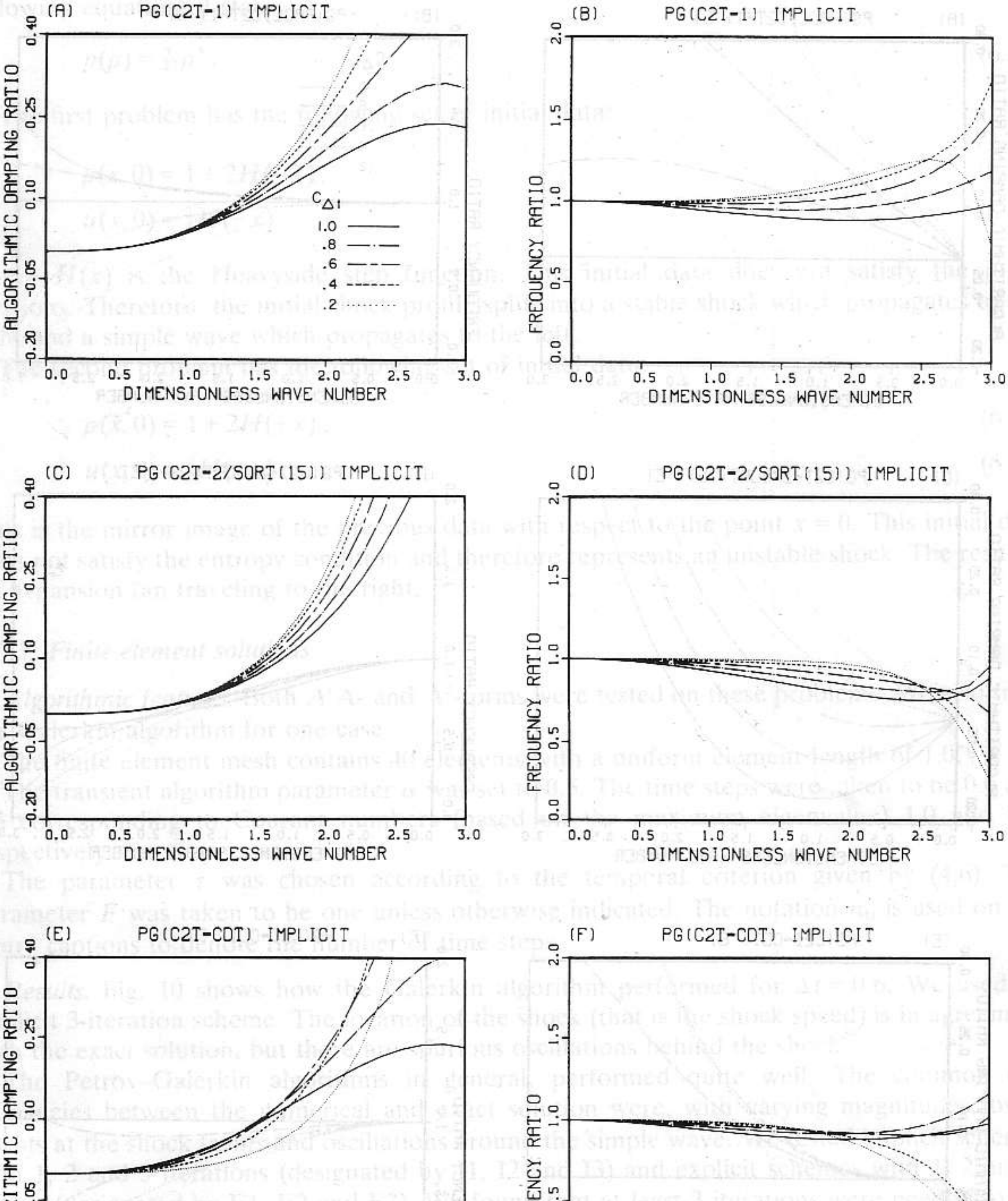

Fig. 3. Algorithmic damping and frequency ratios for Petrov–Galerkin algorithms.

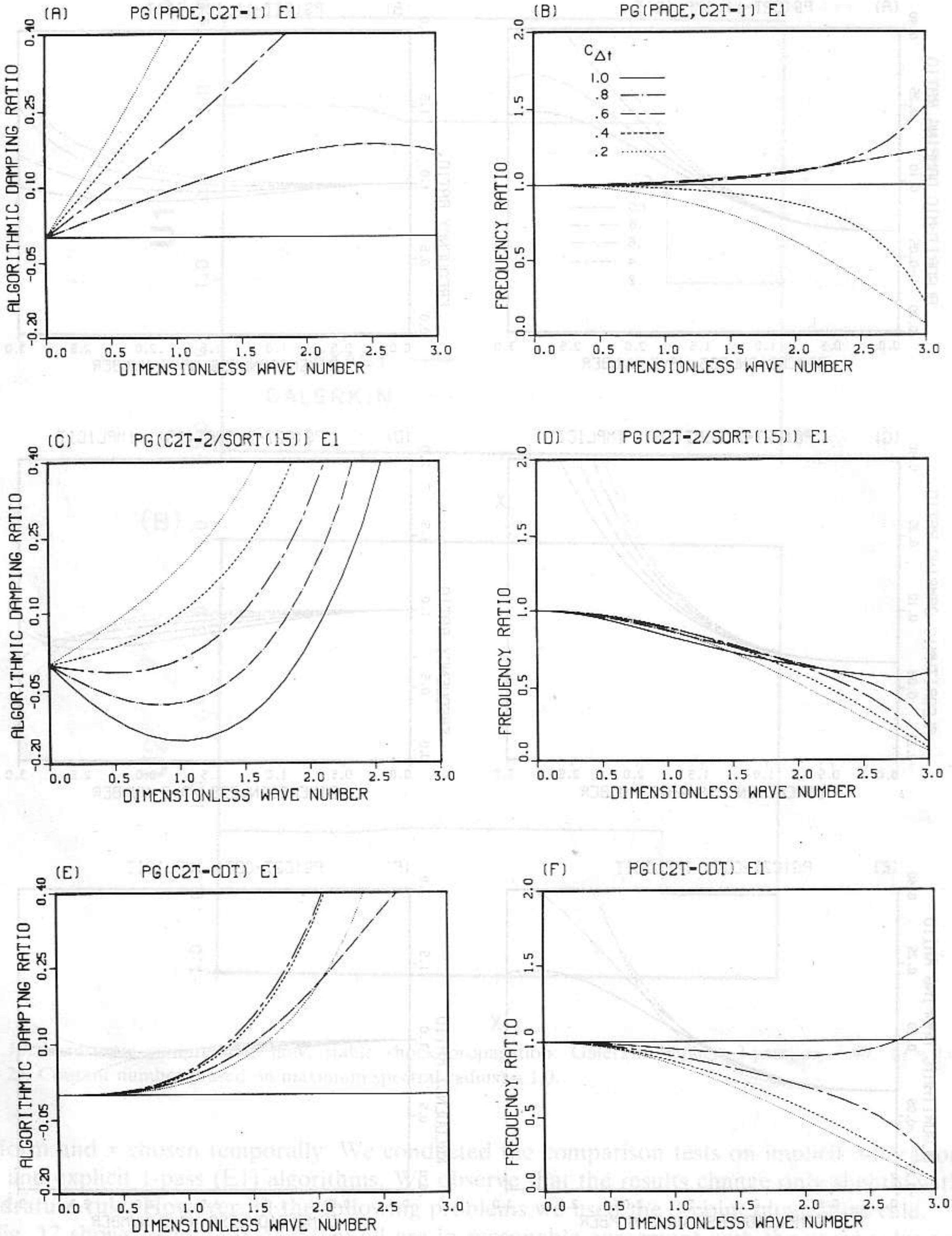

Fig. 4. Algorithmic damping and frequency ratios for explicit 1-pass algorithms.

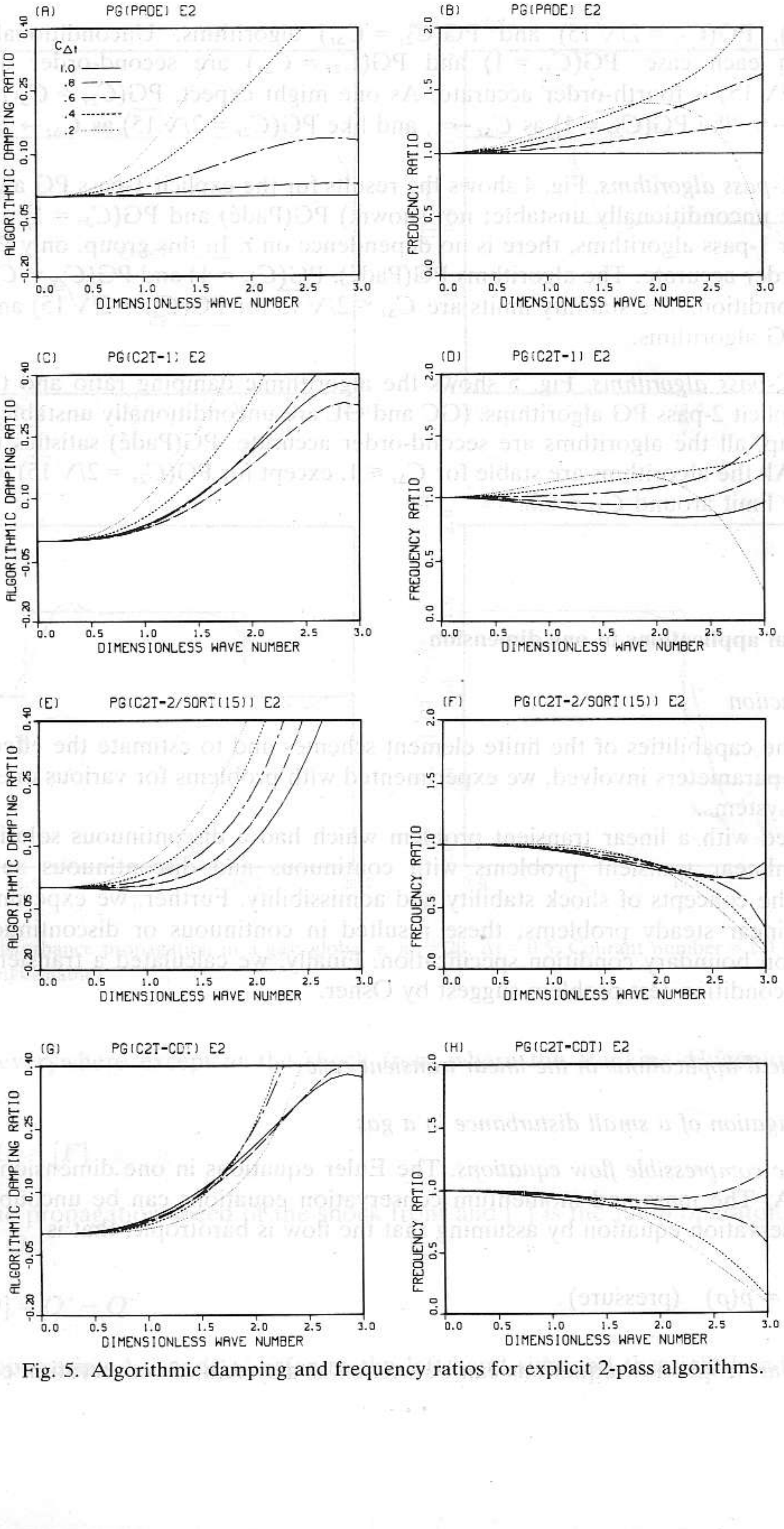

Fig. 5. Algorithmic damping and frequency ratios for explicit 2-pass algorithms.

PG($C_{2\tau} = 1$), PG($C_{2\tau} = 2/\sqrt{15}$) and PG($C_{2\tau} = C_{\Delta t}$) algorithms. Unconditional stability is achieved in each case. PG($C_{2\tau} = 1$) and PG($C_{2\tau} = C_{\Delta t}$) are second-order accurate, and PG($C_{2\tau} = 2/\sqrt{15}$) is fourth-order accurate. As one might expect, PG($C_{2\tau} = C_{\Delta t}$) behaves like GC as $C_{\Delta t} \to 0$, like PG($C_{2\tau} = 1$) as $C_{\Delta t} \to 1$, and like PG($C_{2\tau} = 2/\sqrt{15}$) as $C_{\Delta t} \to 2/\sqrt{15}$.

Explicit 1-pass algorithms. Fig. 4 shows the results for the explicit 1-pass PG algorithms (GC and GL are unconditionally unstable; not shown.) PG(Padé) and PG($C_{2\tau} = 1$) are equivalent because, for 1-pass algorithms, there is no dependence on r. In this group, only PG($C_{2\tau} = C_{\Delta t}$) is second-order accurate. The algorithms PG(Padé), PG($C_{2\tau} = 1$) and PG($C_{2\tau} = C_{\Delta t}$) satisfy the unit CFL condition. The stability limits are $C_{\Delta t} \leq 2/\sqrt{15}$ for PG($C_{2\tau} = 2/\sqrt{15}$) and $C_{\Delta t} \leq 1$ for the other PG algorithms.

Explicit 2-pass algorithms. Fig. 5 shows the algorithmic damping ratio and the frequency ratio for explicit 2-pass PG algorithms. (GC and GL are unconditionally unstable; not shown.) In this group, all the algorithms are second-order accurate. PG(Padé) satisfies the unit CFL condition. All the algorithms are stable for $C_{\Delta t} \leq 1$, except for PG($C_{2\tau} = 2/\sqrt{15}$) which exceeds the stability limit around $C_{\Delta t} = 0.8$.

6. Numerical applications in one dimension

6.1. Introduction

To test the capabilities of the finite element schemes and to estimate the effects of several algorithmic parameters involved, we experimented with problems for various one-dimensional hyperbolic systems.

We started with a linear transient problem which had a discontinuous solution. Then we studied nonlinear transient problems with continuous and discontinuous solution; these illustrated the concepts of shock stability and admissibility. Further, we experimented with a set of nonlinear steady problems; these resulted in continuous or discontinuous solutions depending on boundary condition specification. Finally, we calculated a transient solution to an entropy-condition test problem suggest by Osher.

6.2. Numerical applications in the linear transient case

6.2.1. Propagation of a small disturbance in a gas

Barotropic compressible flow equations. The Euler equations in one dimension are given in Appendix A. The mass and momentum conservation equations can be uncoupled from the energy conservation equation by assuming that the flow is barotropic, that is

$$p = p(\rho) \quad \text{(pressure)}. \tag{6.1}$$

Then, the barotropic flow equations can be written as a system of conservation equations with conservation variables and flux vector defined as

$$U = \rho\begin{Bmatrix}1\\u\end{Bmatrix}, \tag{6.2}$$

$$F = \begin{Bmatrix}\rho u\\\rho u^2 + p\end{Bmatrix}. \tag{6.3}$$

Small disturbance equations. We assume that the departures of ρ and u from constant values ρ_0 and u_0 are very small. Taking u_0 to be zero, this assumption can also be stated as ([62])

$$\left|\frac{\rho}{\rho_0} - 1\right| \ll 1, \tag{6.4}$$

$$\left|\frac{p'(\rho)}{p'(\rho_0)} - 1\right| \ll 1, \tag{6.5}$$

$$\left|\frac{u}{\sqrt{p'(\rho_0)}}\right| \ll 1 \tag{6.6}$$

where $p'(\rho)$ is the derivative of $p(\rho)$ with respect to its argument. These assumptions lead to the following linearized version of the original conservation equation system:

$$\rho_{,t} + \rho_0 u_{,x} = 0, \tag{6.7}$$

$$\rho_0 u_{,t} + p'(\rho_0)\rho_{,x} = 0. \tag{6.8}$$

Further, by introducing the acoustic-speed parameter c, defined by

$$c^2 = p'(\rho_0), \tag{6.9}$$

and the scaling

$$\rho^* = \rho/\rho_0 - 1, \tag{6.10}$$

$$u^* = -u/c, \tag{6.11}$$

we get

$$U_{,t} + AU_{,x} = 0 \tag{6.12}$$

where

$$U = \begin{Bmatrix}\rho^*\\u^*\end{Bmatrix} \tag{6.13}$$

and

$$A = \begin{bmatrix}0 & -c\\-c & 0\end{bmatrix}. \tag{6.14}$$

Note that $\boldsymbol{A}$ is symmetric. Its eigenvalues are

$$\lambda_{1,2} = \pm c\,. \tag{6.15}$$

Henceforth, for notational simplicity, we omit the asterisks on ρ and u.

6.2.2. *Initial/boundary-value problem*

The equation system of (6.12), together with the following initial/boundary-value data was studied numerically by Hughes in [23]:

$$\left.\begin{aligned}\rho(x,0) &= 1\,,\\ u(x,0) &= 0\,,\end{aligned}\right. \quad x \in \,]0, 10[\,, \tag{6.16}$$

$$\left.\begin{aligned}\rho(0,t) &= 0\,,\\ u(10,t) &= 0\,,\end{aligned}\right. \quad t \geqslant 0\,. \tag{6.17}$$

The acoustic speed was chosen to be unity.

6.3.3. *Finite element solutions*

Naming convention for the $\boldsymbol{T} = \tau\boldsymbol{A}$ and $\boldsymbol{T} = \tau\boldsymbol{A}^{t}$ forms. If we choose to define the operator $\boldsymbol{T}$ according to the criterion $\boldsymbol{T} = \tau\boldsymbol{A}$, then the resulting formulation would have a second-order term containing the product $\boldsymbol{A}^{t}\boldsymbol{A}$. Therefore, we name this the '$A^{t}A$-form'. If we choose $\boldsymbol{T} = \tau\boldsymbol{A}^{t}$ then the second-order term would contain the product $\boldsymbol{A}^{2}$. Therefore, we name this the 'A^{2}-form'. Thus, in the study of this all subsequent problems, we adopt the following naming convention:

$$\boldsymbol{T} = \tau\boldsymbol{A} \leftrightarrow A^{t}A\,,$$

$$\boldsymbol{T} = \tau\boldsymbol{A}^{t} \leftrightarrow A^{2}\,.$$

Algorithmic features. In this problem, both $A^{t}A$- and A^{2}-forms result in the same formulation due to the symmetry of the operator $\boldsymbol{A}$. Setting $\boldsymbol{T} = \boldsymbol{0}$ we obtain the usual Galerkin technique.

The finite element mesh contains 20 elements with uniform mesh spacing of 0.5. The quadrature rules chosen provide exact integration of the vectors and matrices involved. Unless otherwise indicated, $\alpha = 0.5$ and $F = 1.0$.

Implicit, explicit 1-pass (designated by E1), and explicit 2-pass (designated by E2) algorithms were tested. Two different time steps, 0.50 and 0.25, were used; these time steps correspond to Courant numbers 1.0 and 0.5, respectively. For both time steps, the temporal criterion for τ as given by (4.6) was used. (By virtue of the fact that τ is constant throughout the mesh in this case, equivalent results can be obtained by use of the spatial criterion. However, in some cases, a different value of F needs to be employed to obtain the same value of τ used in the temporal criterion.)

Results. In the figures, U1 and U2 refer to ρ and u, respectively. Fig. 6 shows the implicit Galerkin solution for Courant number 1.0. As can be seen this technique produces spurious oscillations.

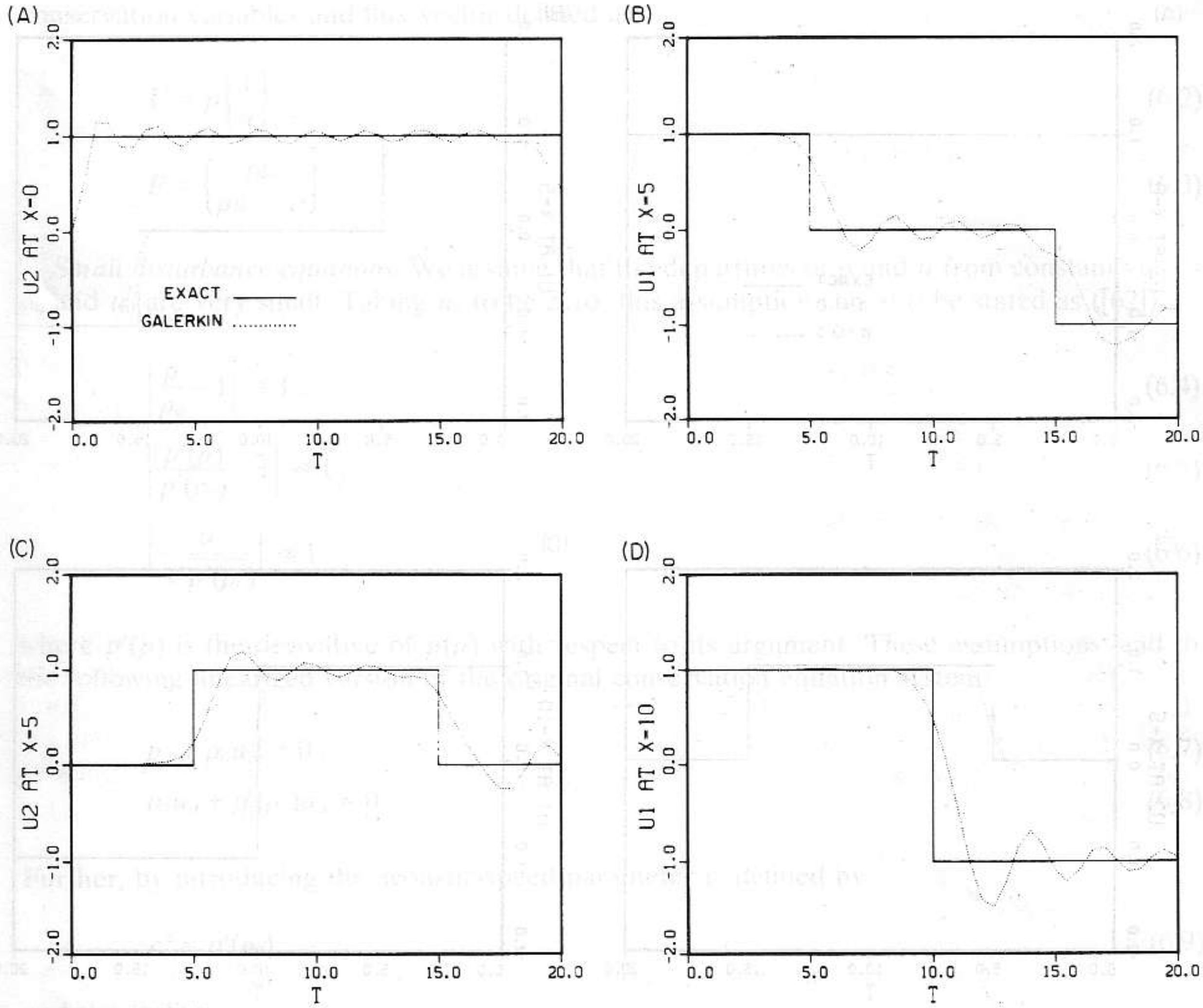

Fig. 6. Small disturbance propagation in a gas: Galerkin implicit, $n_{\mathrm{el}} = 20$, $\Delta t = 0.5$, Courant number = 1.0.

Fig. 7 shows solutions for the explicit 1-pass Petrov–Galerkin algorithm. Both α and F were varied to produce a constant $F\alpha = 0.25$. The unit CFL condition is satisfied in this case, however, by varying α, slightly different results are produced due to the dependence of the first time step's calculations upon α. As may be seen, $\alpha = 1.0$ produces sharper fronts and thus is preferable for the explicit 1-pass algorithm.

Fig. 8 shows the solutions (for Courant number 1.0) produced by Petrov–Galerkin implicit, explicit 1-pass and, explicit 2-pass algorithms. The explicit 2-pass results are not distinguishable from the implicit results. This implies that, for this problem, as the number of passes increase, the explicit algorithm converges quite rapidly to the implicit one.

Fig. 9 shows the results for Courant number 0.5 produced by the same set of algorithms. The implicit and explicit 2-pass algorithms are, again, indistinguishable. The explicit 1-pass algorithm produces slightly greater oscillations.

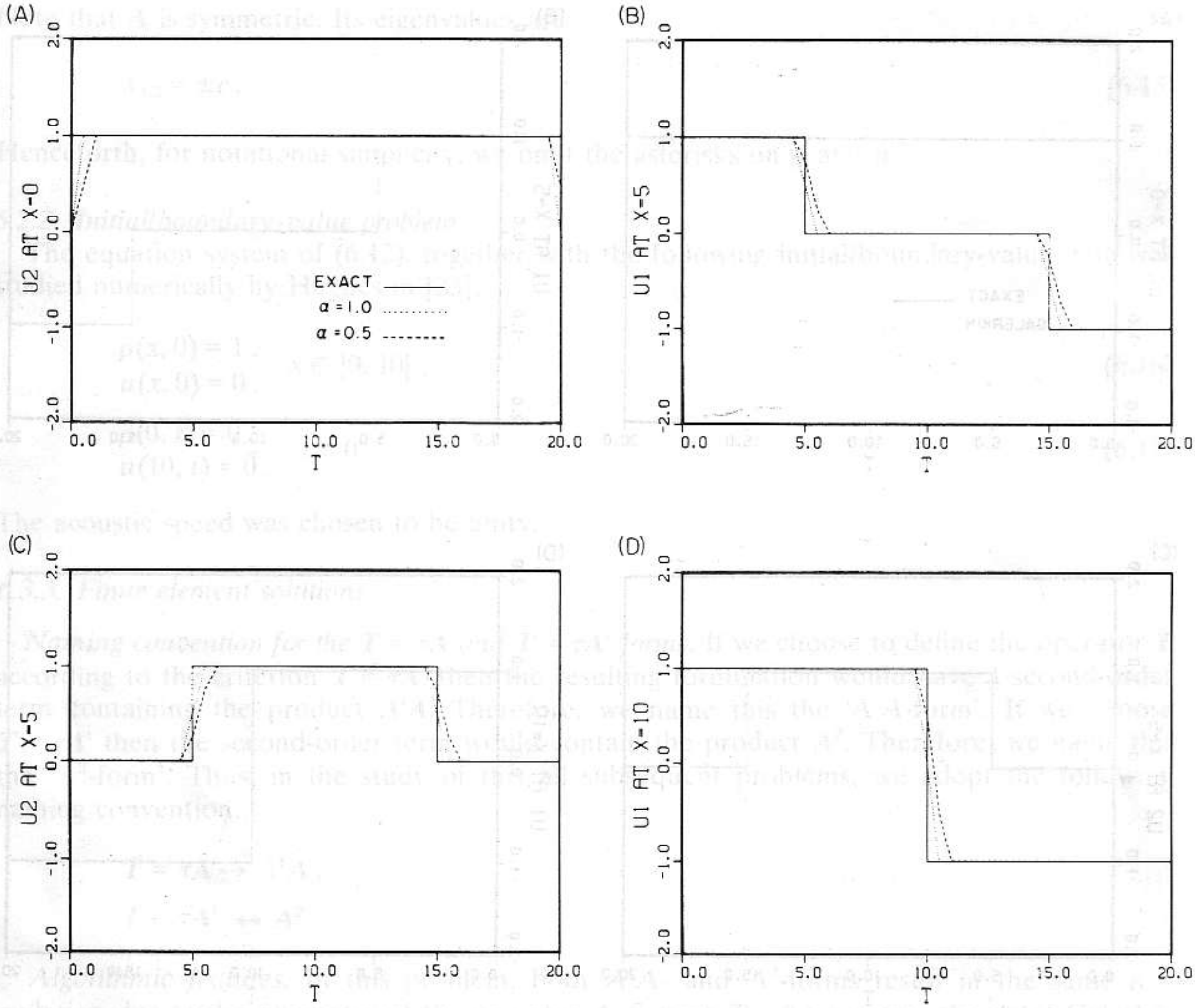

Fig. 7. Small disturbance propagation in a gas: global τ, explicit 1-pass, $n_{\mathrm{el}} = 20$, $\Delta t = 0.5$, Courant number = 1.0. (Note $\tau = 0.25$ for both cases presented.)

Evaluation. The results for the Petrov–Galerkin calculations tend to spread-out fronts and produce some mild oscillations about fronts. Results such as these are indicative of the performance of the algorithms on contact discontinuities in nonlinear problems in which characteristics are approximately parallel. (Shocks, due to convergence of characteristics, tend to be propagated with less smearing; this will be illustrated in some of the calculations which follow.) Significant improvement in resolving contact discontinuities can be achieved by employing algorithmic ideas described in [19–21].

6.3. Numerical applications in the nonlinear transient case

6.3.1. Barotropic compressible flow

Barotropic compressible flow was defined in Section 6.2. The differential equations need to

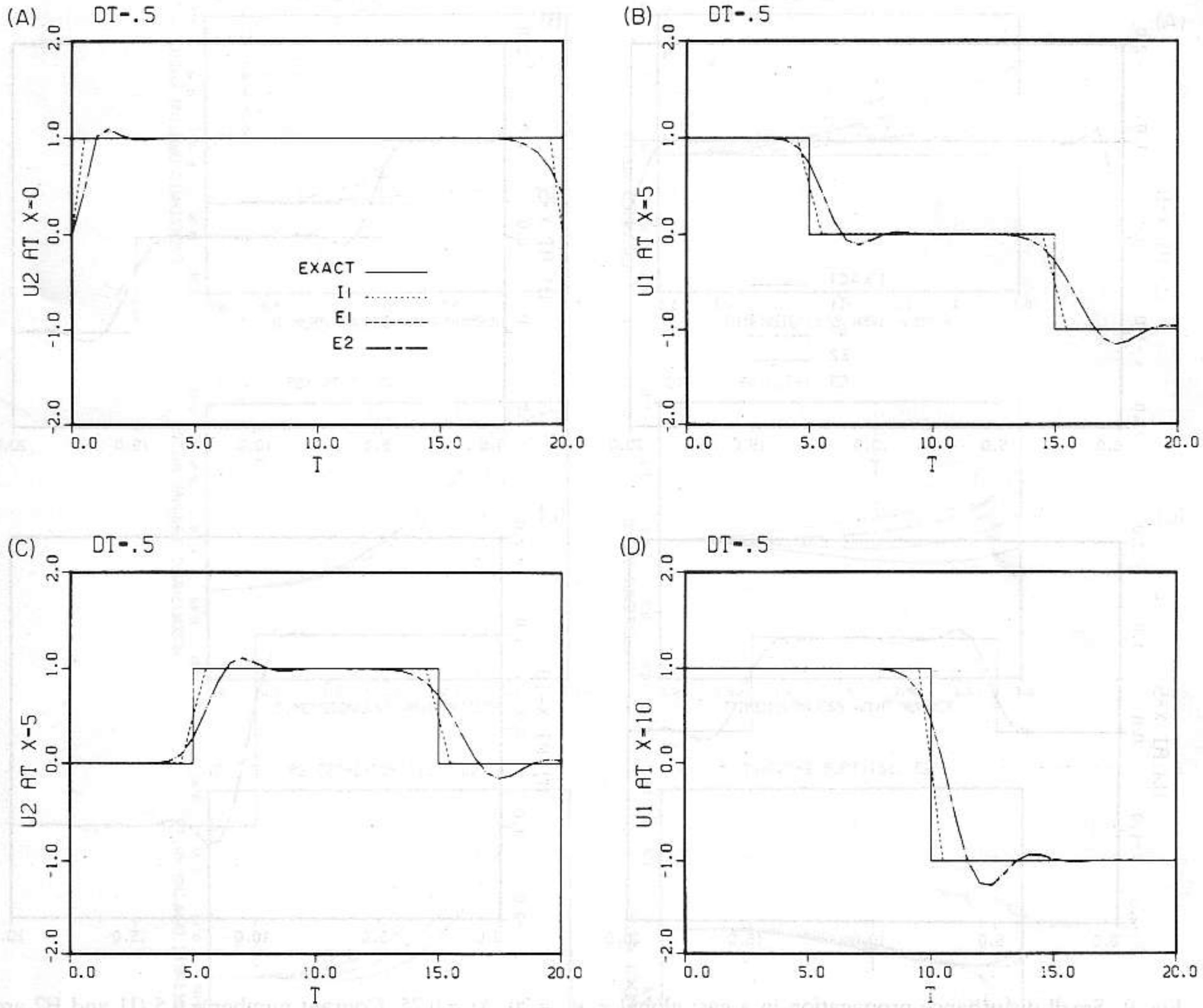

Fig. 8. Small disturbance propagation in a gas: global τ, $n_{\mathrm{el}} = 20$, $\Delta t = 0.5$, Courant number = 1.0 (I1 and E2 are virtually indistinguishable).

be satisfied everywhere except at the shock front where the Rankine–Hugoniot conditions hold:

$$s[\boldsymbol{U}] = [\boldsymbol{F}] \tag{6.18}$$

where s is the propagation speed of the shock front and [ ] is the jump operator. That is, for any variable Q,

$$[Q] = Q^{+} - Q^{-} \tag{6.19}$$

where the superscripts '−' and '+' refer to the left-hand side and the right-hand side of the

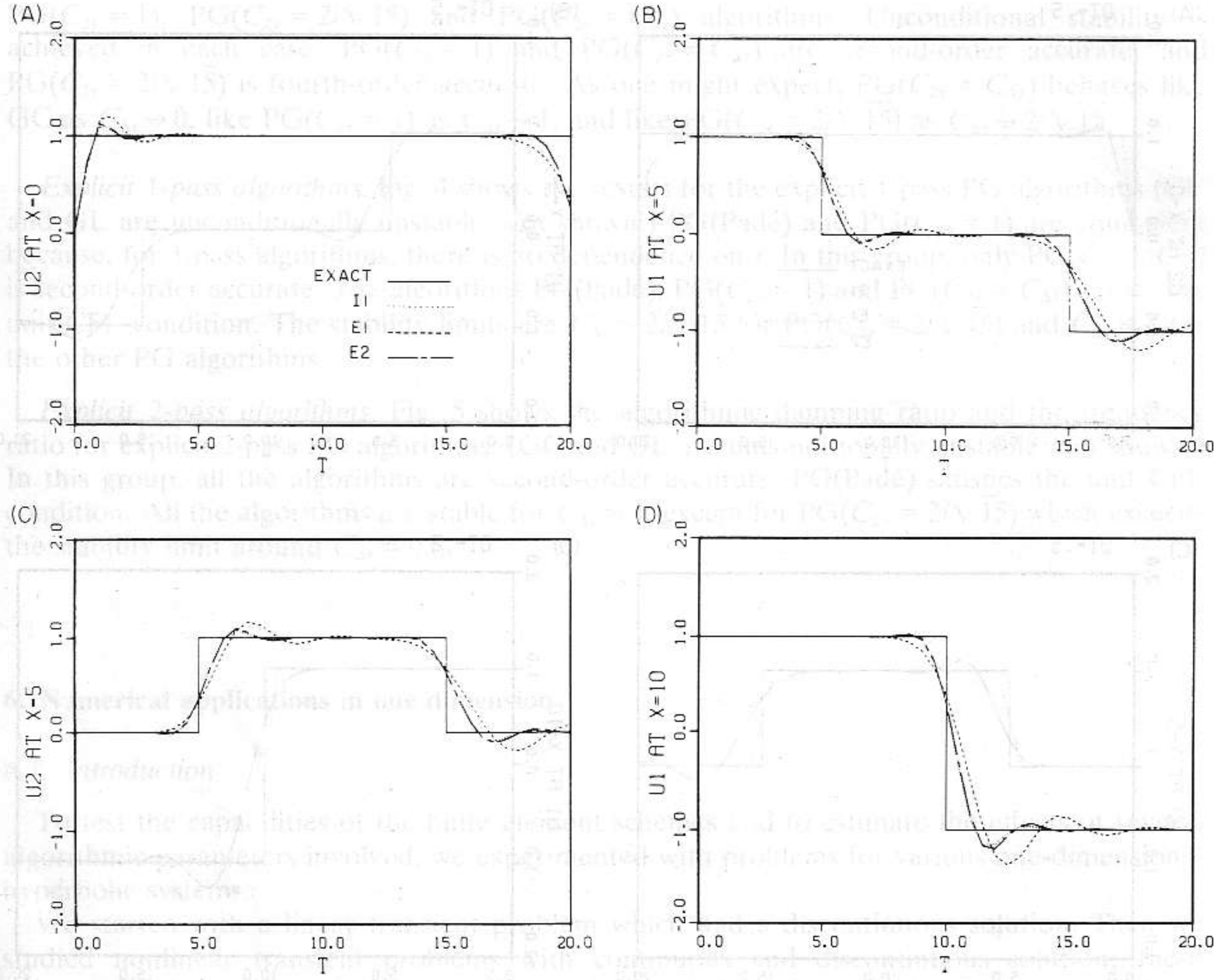

Fig. 9. Small disturbance propagation in a gas: global τ, $n_{el} = 20$, $\Delta t = 0.25$, Courant number = 0.5 (I1 and E2 are virtually indistinguishable).

shock front, respectively. For barotropic flow, these conditions are

$$s[\rho] = [\rho u], \tag{6.20}$$

$$s[\rho u] = [\rho u^2 + p]. \tag{6.21}$$

For shock profiles to be stable, the entropy condition must also be satisfied (see [22, 37–39]). The entropy condition is given in the form of inequalities in terms of s and the eigenvalues of the Jacobian matrix A. In the present case, the eigenvalues are

$$\lambda_{1,2} = u \pm c, \quad c^2 = p'(\rho). \tag{6.22}$$

6.3.2. Initial-value problems

We considered two initial-value problems, both studied by Hughes in [23], with the following equation of state:

$$p(\rho) = \tfrac{1}{27}\rho^3 . \tag{6.23}$$

The first problem has the following set of initial data:

$$\rho(x, 0) = 1 + 2H(-x) , \tag{6.24}$$

$$u(x, 0) = \tfrac{4}{3}H(-x) \tag{6.25}$$

where $H(x)$ is the Heavyside step function. This initial data does not satisfy the jump relations. Therefore, the initial shock profile splits into a stable shock which propagates to the right and a simple wave which propagates to the left.

The second problem has the following set of initial data:

$$\rho(x, 0) = 1 + 2H(+x) , \tag{6.26}$$

$$u(x, 0) = \tfrac{4}{3}H(+x) . \tag{6.27}$$

This is the mirror image of the previous data with respect to the point $x = 0$. This initial data does not satisfy the entropy condition and therefore represents an unstable shock. The result is an expansion fan traveling to the right.

6.3.3. *Finite element solutions*

Algorithmic features. Both A^tA- and A^2-forms were tested on these problems. We also tried the Galerkin algorithm for one case.

The finite element mesh contains 40 elements with a uniform element length of 1.0.

The transient algorithm parameter α was set to 0.5. The time steps were taken to be 0.6 and 0.3 corresponding to Courant numbers (based on the maximum eigenvalue) 1.0 and 0.5, respectively.

The parameter τ was chosen according to the temporal criterion given by (4.6). The parameter F was taken to be one unless otherwise indicated. The notation n_{ts} is used on the figure captions to denote the number of time steps.

Results. Fig. 10 shows how the Galerkin algorithm performed for $\Delta t = 0.6$. We used an implicit 3-iteration scheme. The location of the shock (that is the shock speed) is in agreement with the exact solution, but there are spurious oscillations behind the shock.

The Petrov–Galerkin algorithms in general, performed quite well. The common discrepancies between the numerical and exact solution were, with varying magnitudes, overshoots at the shock fronts and oscillations around the simple wave. We tested implicit schemes with 1, 2 and 3 iterations (designated by I1, I2 and I3) and explicit schemes with 1, 2 and 3 passes (designated by E1, E2 and E3). We found that at least 3 iterations were needed to get the correct shock structures when an implicit scheme was used. This observation is in agreement with the findings of Baker [3].

The element level 'mass' matrices and the vectors corresponding to them were integrated exactly. For the integration of all the other matrices we tested 1-, 2- and 3-point Gaussian quadrature rules. Fig. 11 shows the comparison of the integration rules for $\Delta t = 0.6$ with the

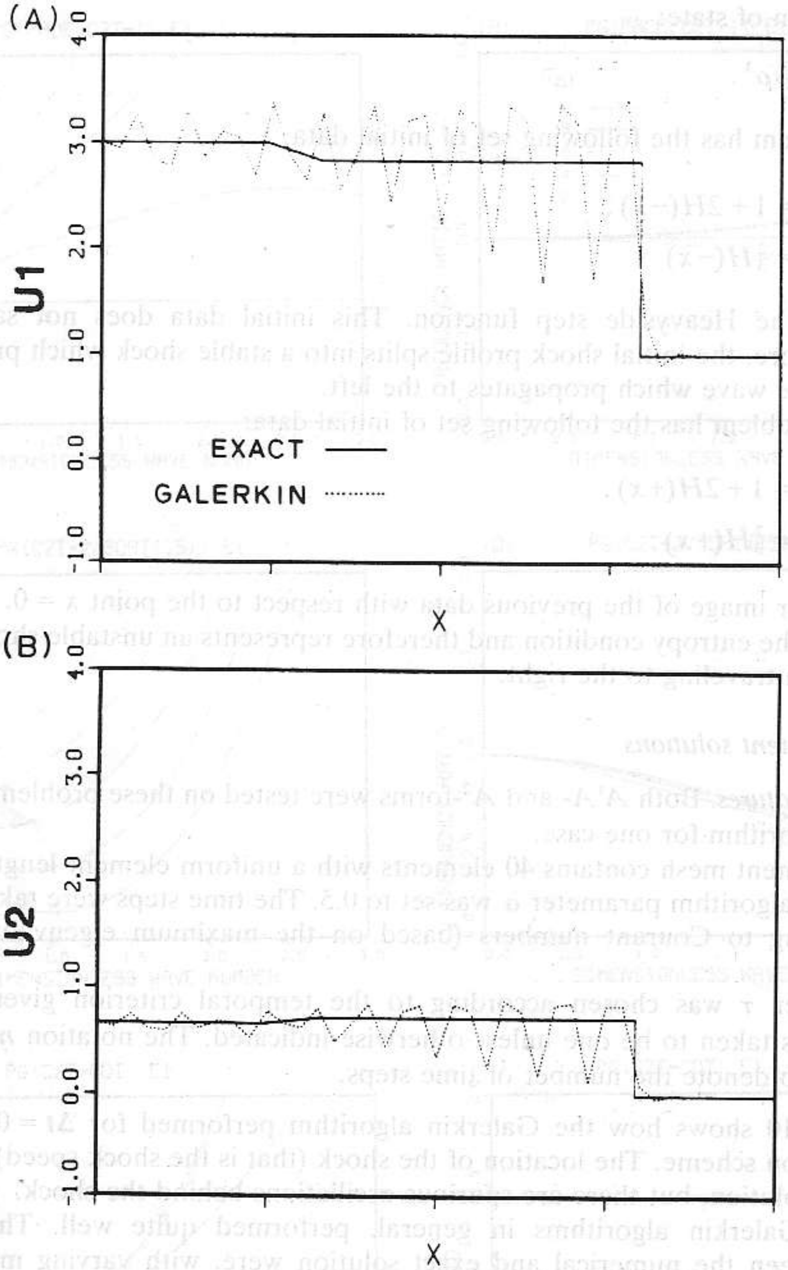

Fig. 10. Barotropic compressible flow, stable shock propagation: Galerkin implicit 3-pass, $n_{el} = 40$, $\Delta t = 0.6$, $n_{ts} = 25$, Courant number (based on maximum spectral radius) = 1.0.

A^2-form and τ chosen temporally. We conducted the comparison tests on implicit 3-iteration (I3) and explicit 1-pass (E1) algorithms. We observe that the results change only slightly with quadrature rule. However, in the following problems we used the 3-point quadrature rule.

Fig. 12 shows results for $\Delta t = 0.6$; all are in reasonable agreement with the exact solution, but exhibit, with comparable magnitudes, overshoots at the shock front and oscillations behind

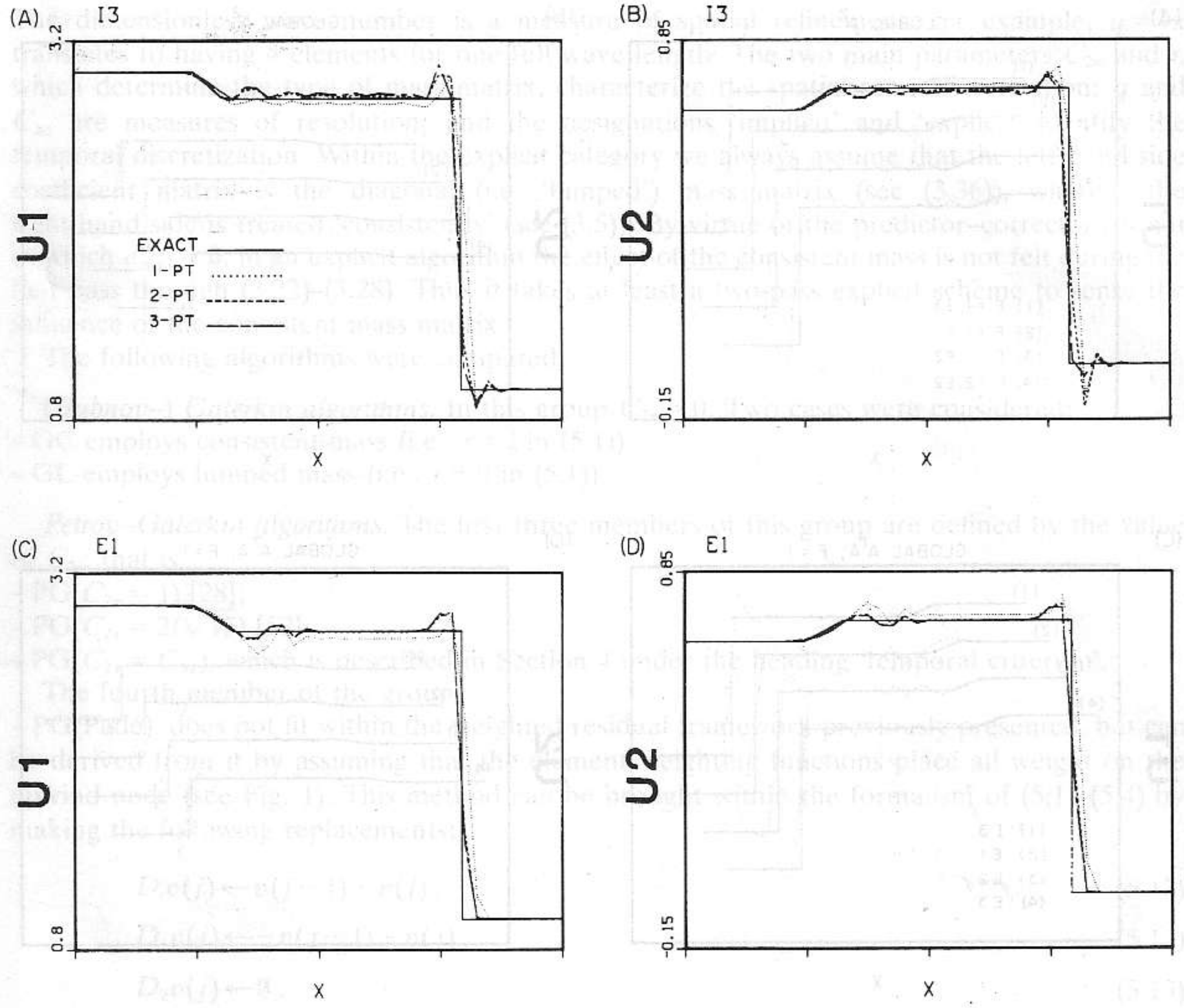

Fig. 11. Barotropic compressible flow, stable shock propagation: global τ, A^2-form, $n_{el} = 40$, $\Delta t = 0.6$, $n_{ts} = 25$, Courant number (based on maximum spectral radius) = 1.0. Comparison of different element quadrature rules.

the simple wave. We observe that explicit 2- and 3-pass results are very similar. For the A^tA-form the explicit 1-pass algorithm became unstable: for this form we also tested the implicit 3-iteration scheme with $F = 2$. This slightly reduced the oscillations and the magnitude of the overshoot to the left of the shock front.

Fig. 13 shows the results for $\Delta t = 0.3$. The solutions are in agreement with the exact solution with slightly more oscillations behind the shock front compared to the $\Delta t = 0.6$ case. For the A^2-form, there was virtually no difference between explicit 2- and 3-pass algorithms. We also tested the explicit 2-pass algorithm with $F = 2$. This algorithm reduced the oscillations and the magnitude of the overshoot to the left of the shock front.

Fig. 14 shows the results for the unstable shock problem. We used the A^2-form with the temporal choice of τ; the time step was 0.6. We tested the implicit 3-iteration and explicit 1-pass algorithms. Both solutions were in close agreement with the exact solution. In the case

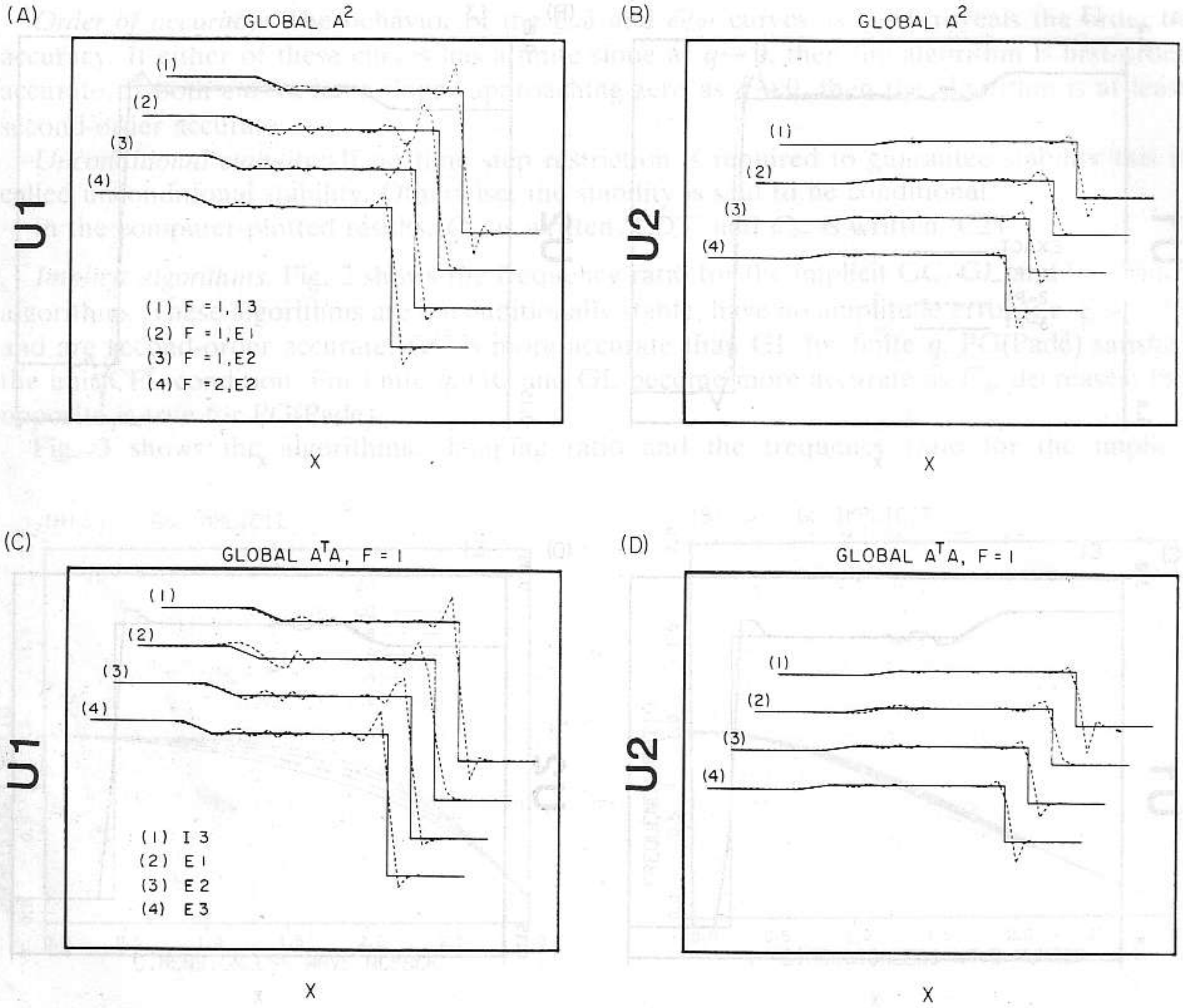

Fig. 12. Barotropic compressible flow, stable shock propagation: $n_{el} = 40$, $\Delta t = 0.3$, $n_{ts} = 50$, Courant number (based on maximum spectral radius) = 0.5. Comparison of various Petrov–Galerkin schemes.

of the E1 algorithm, we used $F = 2$. It produced slightly more oscillations behind the simple wave. All algorithms produced satisfactory results for this case.

Evaluation. The shock structure produced by the Petrov–Galerkin algorithms was faithful to the exact solution; however, oscillations about the front are undesirable. This situation could undoubtably be improved by incorporating some of the algorithmic ideas of [14, 17, 19–21, 44, 47–51] in the present schemes. The breakdown of a discontinuity into an expansion fan was treated adequately by the Petrov–Galerkin algorithms.

6.4. Numerical applications in the nonlinear steady case

6.4.1. Isothermal flow in a nozzle

We consider the one-dimensional isothermal flow in a nozzle with cross-sectional area

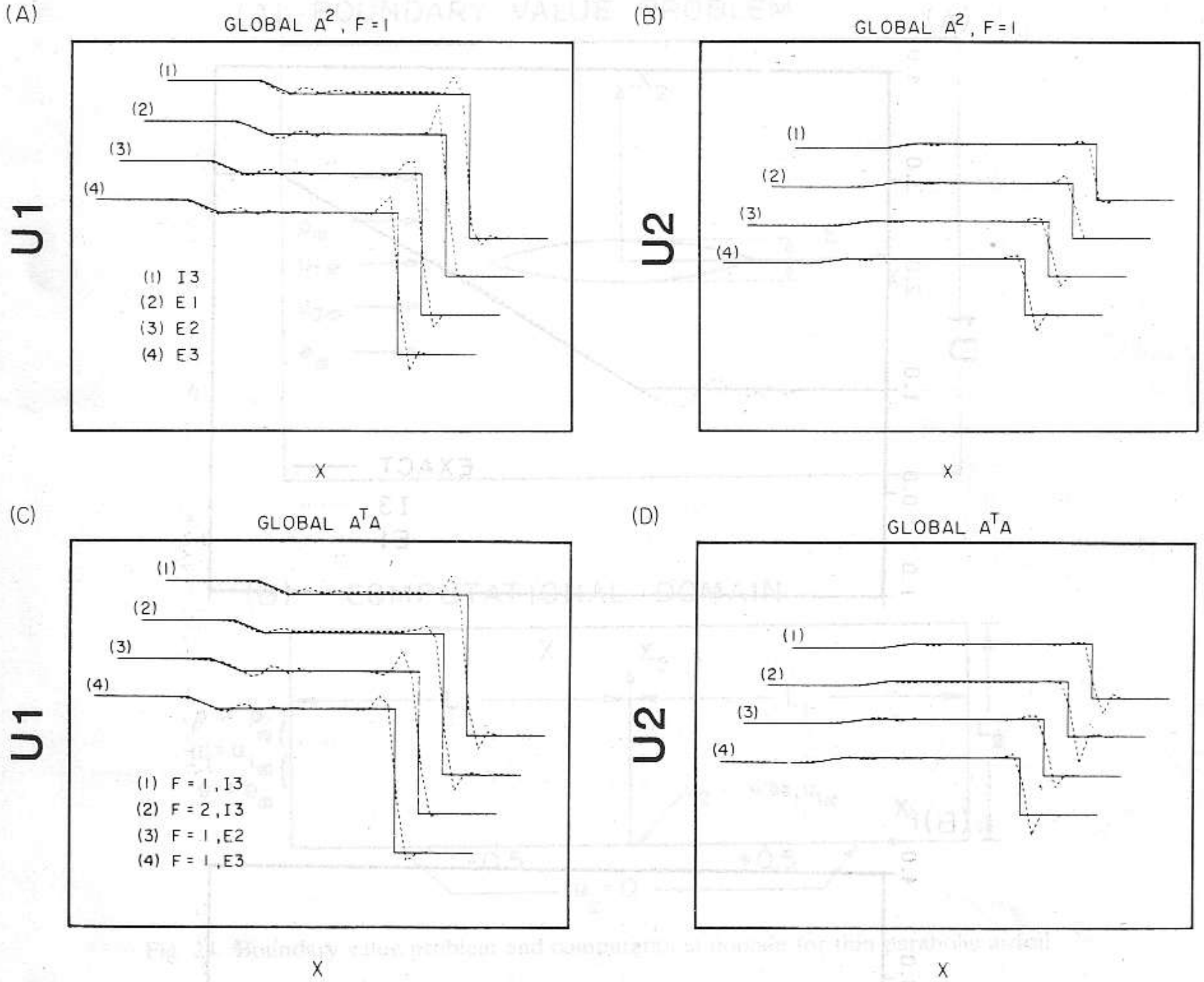

Fig. 13. Barotropic compressible flow, stable shock propagation: $n_{\mathrm{el}} = 40$, $\Delta t = 0.6$, $n_{\mathrm{ts}} = 25$, Courant number (based on maximum spectral radius) = 1.0. Comparison of various Petrov–Galerkin schemes.

varying along the axis. The governing equations, provided by Lomax et al. [42], possess the following conservation variable, flux and source vectors:

$$\boldsymbol{U} = \rho A \begin{Bmatrix} 1 \\ u \end{Bmatrix}, \tag{6.28}$$

$$\boldsymbol{F} = A \begin{Bmatrix} \rho u \\ \rho u^2 + \rho c^2 \end{Bmatrix}, \tag{6.29}$$

$$\boldsymbol{G} = \begin{Bmatrix} 0 \\ -\rho c^2 A_{,x} \end{Bmatrix} \tag{6.30}$$

where the acoustic speed is constant and the cross-sectional area is

$$A = A(x) \,. \tag{6.31}$$

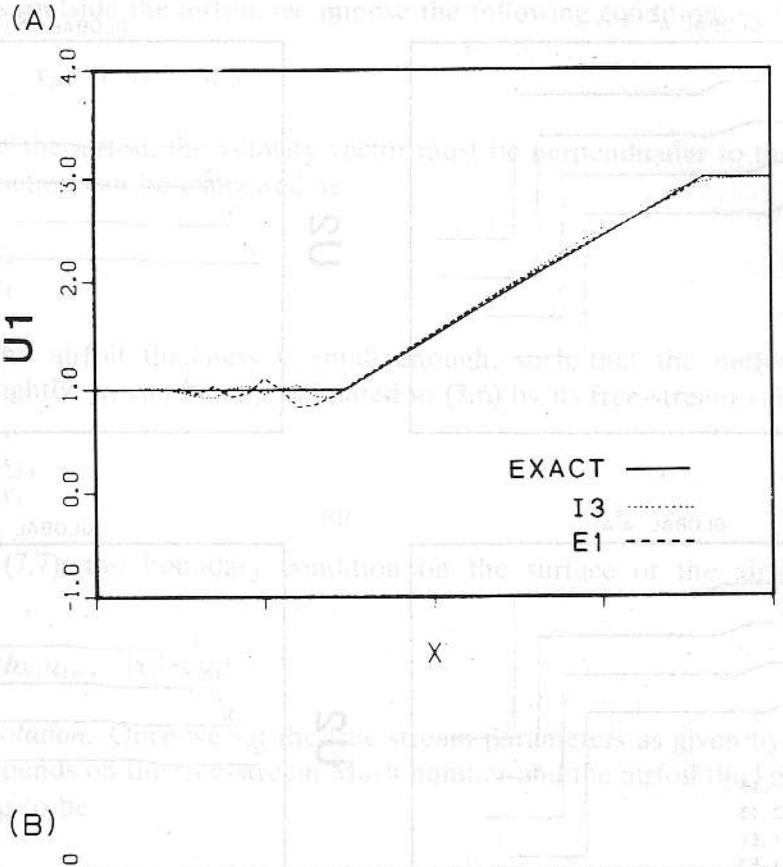

Fig. 14. Barotropic compressible flow, rarefaction wave: global τ, A^2-form, $n_{el} = 40$, $\Delta t = 0.6$, $n_{ts} = 25$. Courant number (based on maximum spectral radius) = 1.0.

The Jacobian matrices are

$$A = \partial F/\partial U = \begin{bmatrix} 0 & 1 \\ -u^2 + c^2 & 2u \end{bmatrix}, \tag{6.32}$$

$$\partial G/\partial U = \begin{bmatrix} 0 & 0 \\ -c^2 A_{,x}/A & 0 \end{bmatrix}. \tag{6.33}$$

The eigenvalues of A are

$$\lambda_{1,2} = u \pm c. \tag{6.34}$$

Assuming that $A(x)$ is a continuous function of x, the Rankine–Hugoniot conditions for steady flow reduce to

$$[\rho u] = 0, \tag{6.35}$$

$$[\rho u^2 + \rho c^2] = 0. \tag{6.36}$$

6.4.2. Boundary-value problem

We studied steady flows suggested by Lomax et al. [42]. The cross-sectional area and the acoustical speed were given by

$$A(x) = 1.0 + \frac{(x - 2.5)^2}{12.5}, \quad 0 \leqslant x \leqslant 5, \tag{6.37}$$

$$c = 1.0. \tag{6.38}$$

The problems considered were:

(1) Subsonic inflow–subsonic outflow with no shock.
(2) Subsonic inflow–supersonic outflow with no shock.
(3) Subsonic inflow–subsonic outflow with shock.

The exact solutions, which can be obtained by the integration of the square of the Mach number (in this case u^2), were provided by Lomax et al. [42].

6.4.3. Finite element solutions

For the boundary conditions of these problems, we set the values of the conservation variables as given by the exact solution. The number of variables to be specified at each boundary depends on the nature of the flow at that boundary. For supersonic inflow, two variables are set; for subsonic inflow or outflow, one variable; and for supersonic outflow, no variable is specified.

Considering all the combinations of possible boundary conditions in terms of conservation variables, we have the following cases for each problem:

For subsonic inflow–subsonic outflow problems:

U1U1: U_1 at the inflow/U_1 at the outflow;
U1U2: U_1 at the inflow/U_2 at the outflow;
U2U1: U_2 at the inflow/U_1 at the outflow;

U2U2: U_2 at the inflow/U_2 at the outflow.
For the subsonic inflow–supersonic outflow problem:
U1: U_1 at the inflow;
U2: U_2 at the inflow.

Transient introduction of the source term. In these problems, we introduced the source term into the equation system in a transient fashion. This is done by taking $A_{,x}$ as a linear function of time during an initial time interval. The numerical $A_{,x}$ can be expressed as follows:

$$\frac{(A_{,x})_{\text{numerical}}}{A_{,x}} = \begin{cases} n/n_{ti}\,, & n < n_{ti} \\ 1\,, & n \geq n_{ti} \end{cases} \tag{6.39}$$

where n_{ti} denotes the number of time steps in the transition interval. For the problems solved, $n_{ti} = 10$.

Algorithmic features. Both A^tA- and A^2-forms were employed. We also tested the Galerkin algorithm.

The finite element mesh has 40 elements with uniform element length 0.125. The element level 'mass' matrices were integrated exactly; all the other matrices and vectors were integrated by the 3-point Gaussian quadrature rule. We set the transient-algorithm parameter α to unity and employed implicit schemes with 2 iterations. The parameter τ was chosen according to both spatial and temporal criteria. The time step for each problem was usually chosen to be ten times the estimated critical time step for that problem. We define the critical time step as the time step for which the Courant number, based on the maximum spectral radius, is unity.

While full convergence to the steady state solutions was attained in about 100 steps, the 50-step solutions were close enough to the steady state solutions for practical purposes.

Results for the subsonic inflow–subsonic outflow problem with no shock. For this problem the time step was chosen to be 0.735.

We attempted to solve the problem with all possible boundary-condition types. Of the four types tried, only one, U1U2, failed to give the expected solution. For each boundary condition type, we tested the usual Galerkin algorithm, A^tA- and A^2-forms, the latter two with temporal choice of τ. For the type U1U1 we also tested the A^2-form with spatial choice of τ. In all cases $F = 1$.

For each boundary-condition type, there was no difference between the solutions produced by different algorithms. However, the solutions differed slightly from one boundary condition type to another. For all types, the agreement with the exact solution was very close. Fig. 15 shows the results.

Results for the subsonic inflow–supersonic outflow problem with no shock. The time step was chosen to be 0.460.

We solved the problem with both boundary-condition types U1 and U2. For each type, we tested the Galerkin algorithm and A^tA- and A^2-forms, the latter two with temporal choice of τ. In all cases $F = 1$.

The results are shown in Fig. 16. For each boundary-condition type, there was no difference between the solutions produced by different algorithms. However, the solutions differed

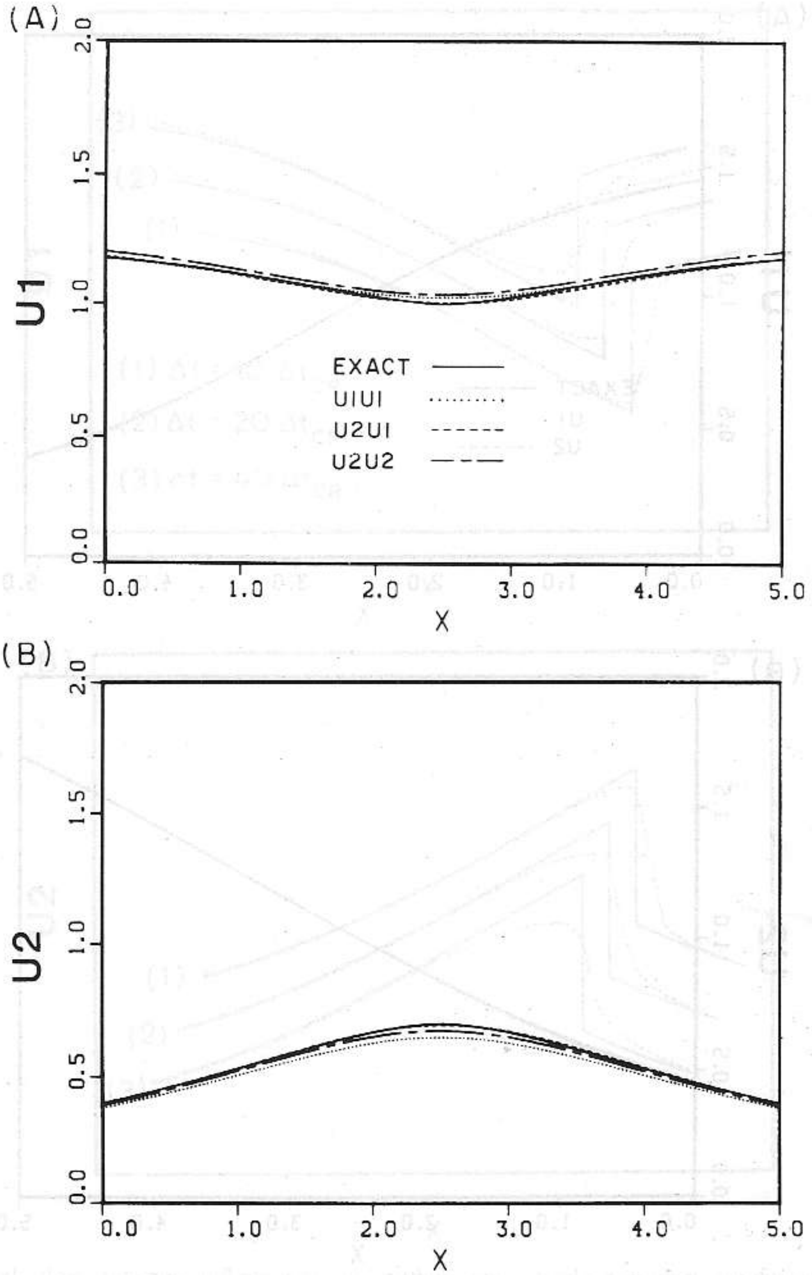

Fig. 15. Steady nozzle flow, subsonic inflow–subsonic outflow, with no shock: $n_{el} = 40$, $\Delta t = 0.735$. Comparison of different boundary conditions. (All methods give essentially the same results.)

slightly from one boundary-condition type to another. For all types, the numerical solutions were in close agreement with the exact solution.

Results for the subsonic inflow–subsonic outflow problem with shock. Unless specified otherwise, the time step was taken to be 0.5.

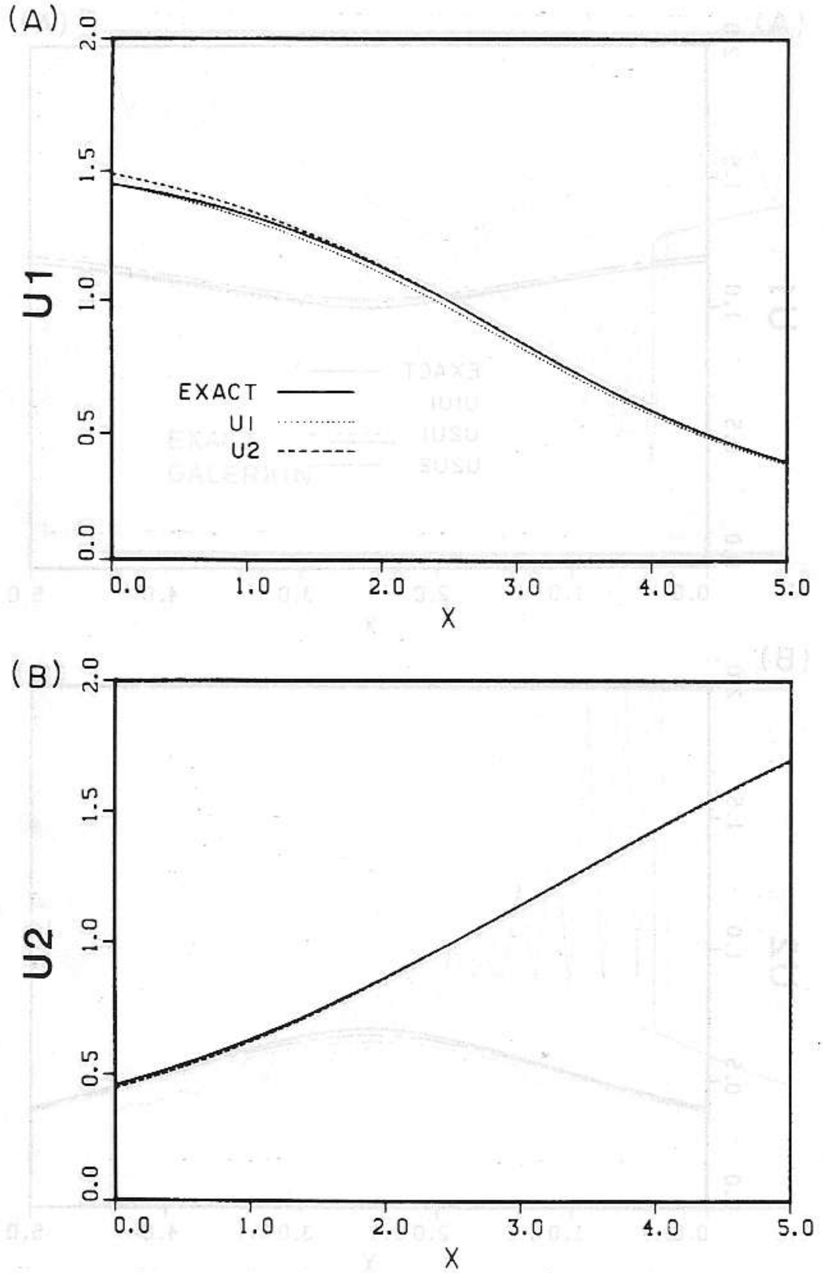

Fig. 16. Steady nozzle flow, subsonic inflow–supersonic outflow, with no shock: $n_{\mathrm{el}} = 40$, $\Delta t = 0.460$. Comparison of different boundary conditions. (All methods give essentially the same results.)

We also attempted to solve this problem with all possible boundary condition types. Types U1U1 and U2U1 gave the expected solutions.

We first describe the solutions obtained for type U1U1: Fig. 17 shows the results for the $A^{\mathrm{t}}A$- and A^2-forms with the temporal choice of τ. The solutions are in close agreement with the exact solution everywhere except at the shock front where the shock front is not very crisp.

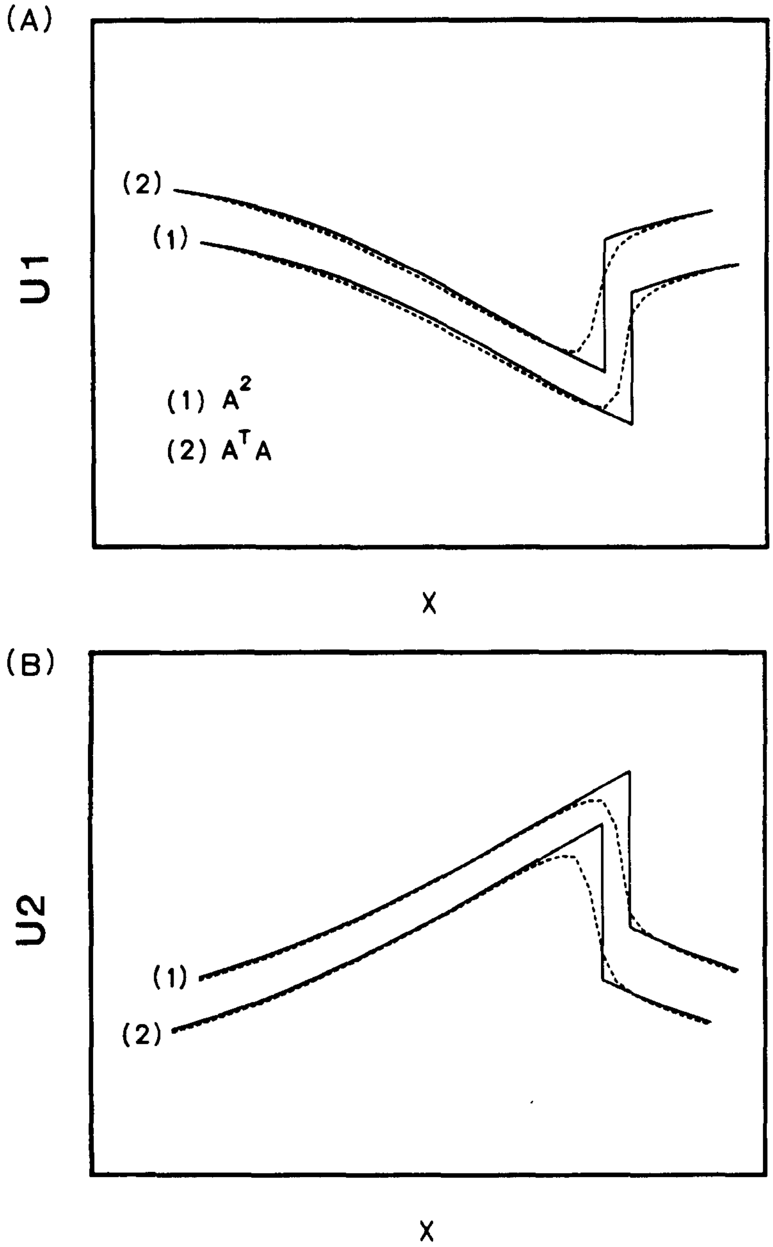

Fig. 17. Steady nozzle flow, subsonic inflow–subsonic outflow, with shock: boundary conditions U1U1, global τ, $n_{el} = 40$, $\Delta t = 0.500$.

Fig. 18 shows the results for the A^2-form with spatial choice of τ. The parameter F assumes values 1, 2, 5 and 10. The solutions are in close agreement with the exact solution. There are very slight oscillations near the shock front for low F. For $F = 1$, the shock front is across one element. As F increases, the shock front becomes smeared.

Fig. 19 shows the results for the A^tA-form with spatial choice of τ. The results are similar to that of Fig. 18. The only differences are: the shock fronts are slightly less crisp; for $F = 1$, we

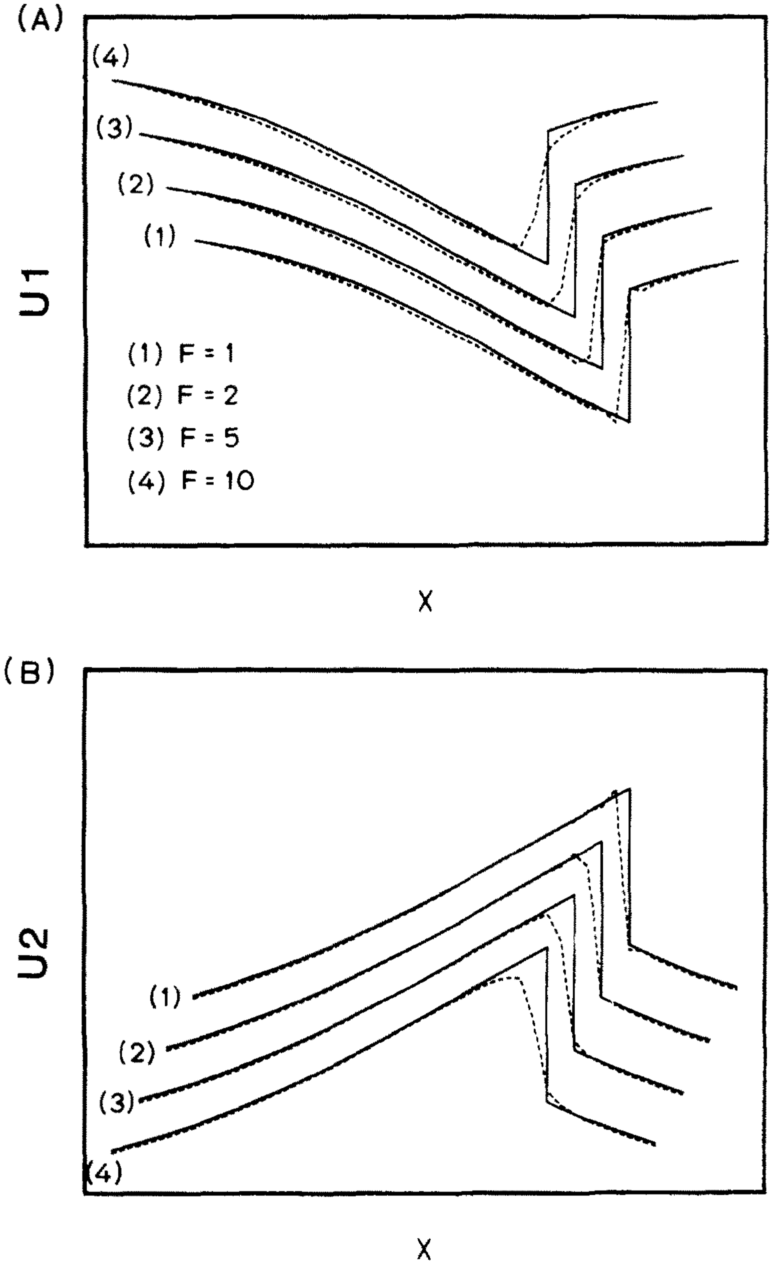

Fig. 18. Steady nozzle flow, subsonic inflow–subsonic outflow, with shock: boundary conditions U1U1, local τ, A^2, $n_{el} = 40$, $\Delta t = 0.500$.

observe oscillations behind the shock front. It is interesting to note that the oscillations are located in the region between the shock front and the sonic point.

U1U2: Both A^tA- and A^2-forms, with temporal choice of τ, produced smooth solutions with no shock.

U2U1: The time step was taken to be 10, 20, and 40 times the estimated critical time step.

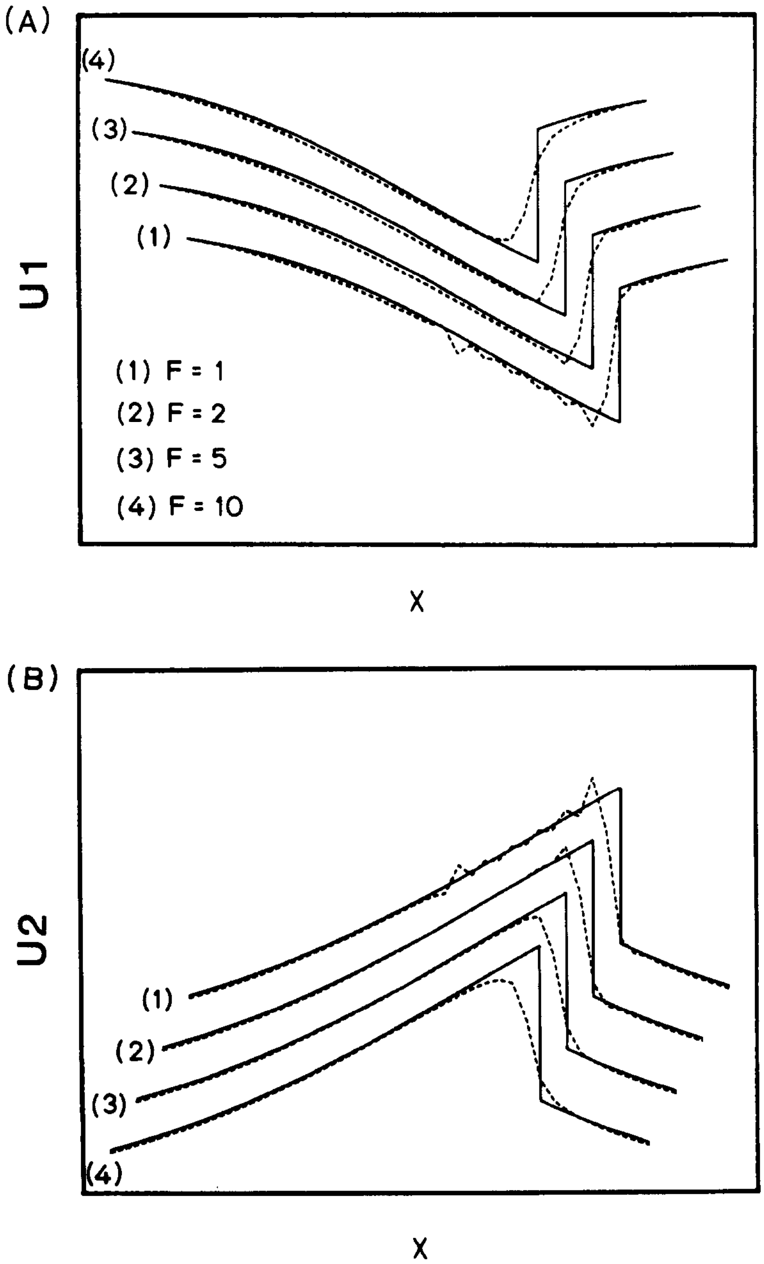

Fig. 19. Steady nozzle flow, subsonic inflow–subsonic outflow, with shock: boundary conditions U1U1, local τ, $A^{t}A$, $n_{el} = 40$, $\Delta t = 0.500$.

The results for $A^{t}A$- and A^2-forms and temporal choice of τ are very similar and are shown in Figs. 20 and 21, respectively. The shock fronts are shifted to the right about one element length and are considerably smeared due to the large value of τ engendered by the large time steps.

U2U2: Both $A^{t}A$ and A^2, with temporal choice of τ, produced smooth symmetric solutions.

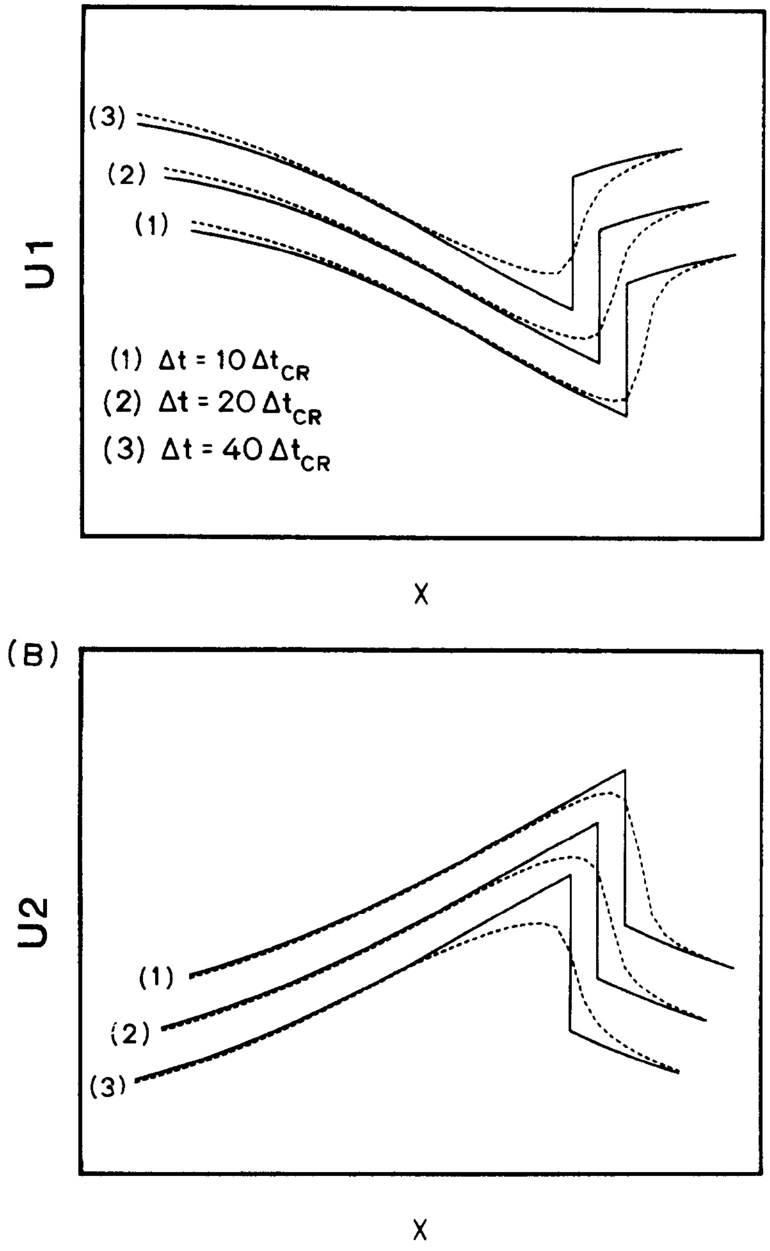

Fig. 20. Steady nozzle flow, subsonic inflow–subsonic outflow, with shock: boundary conditions U2U1, global τ, A^2, $n_{el} = 40$.

For this case, it was only the Galerkin algorithms which sensed the shock and located it almost at the exact location, but with severe oscillations. Fig. 22 shows the result produced by the Galerkin algorithm.

Remark. We observed tht the location of the shock front was shifted about half an element length to the left for the boundary condition type U1U1 and about one element length to the

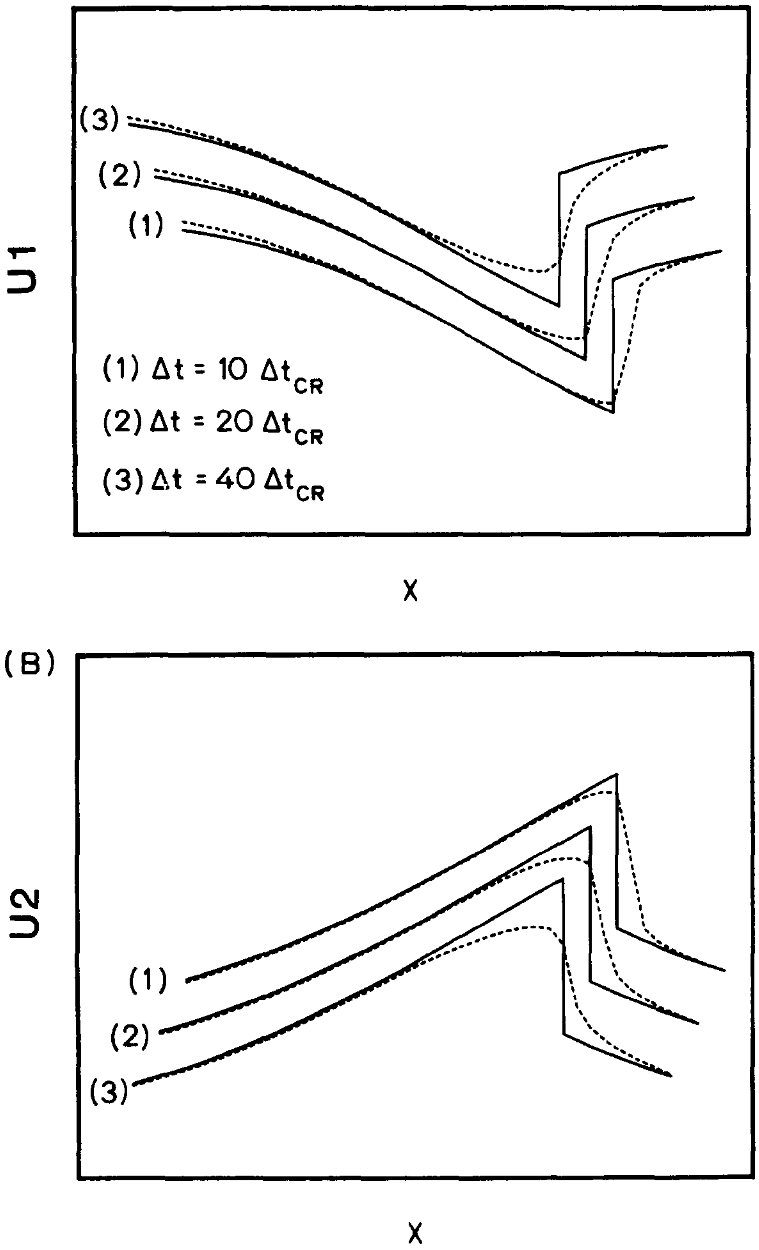

Fig. 21. Steady nozzle flow, subsonic inflow–subsonic outflow, with shock: boundary conditions U2U1, global τ, A^tA, $n_{el} = 40$.

right for the boundary condition type U2U1. This implies that the location of the shock front is dependent to some extent on the type of boundary conditions specified. In particular, for two boundary condition types (U1U2 and U2U2) the exact solution was not obtained. Proper specification of the boundary conditions in problems of this type is a very important subject which has attracted the attention of several researchers (see [4, 43, 64]), but does not yet seem to be fully understood.

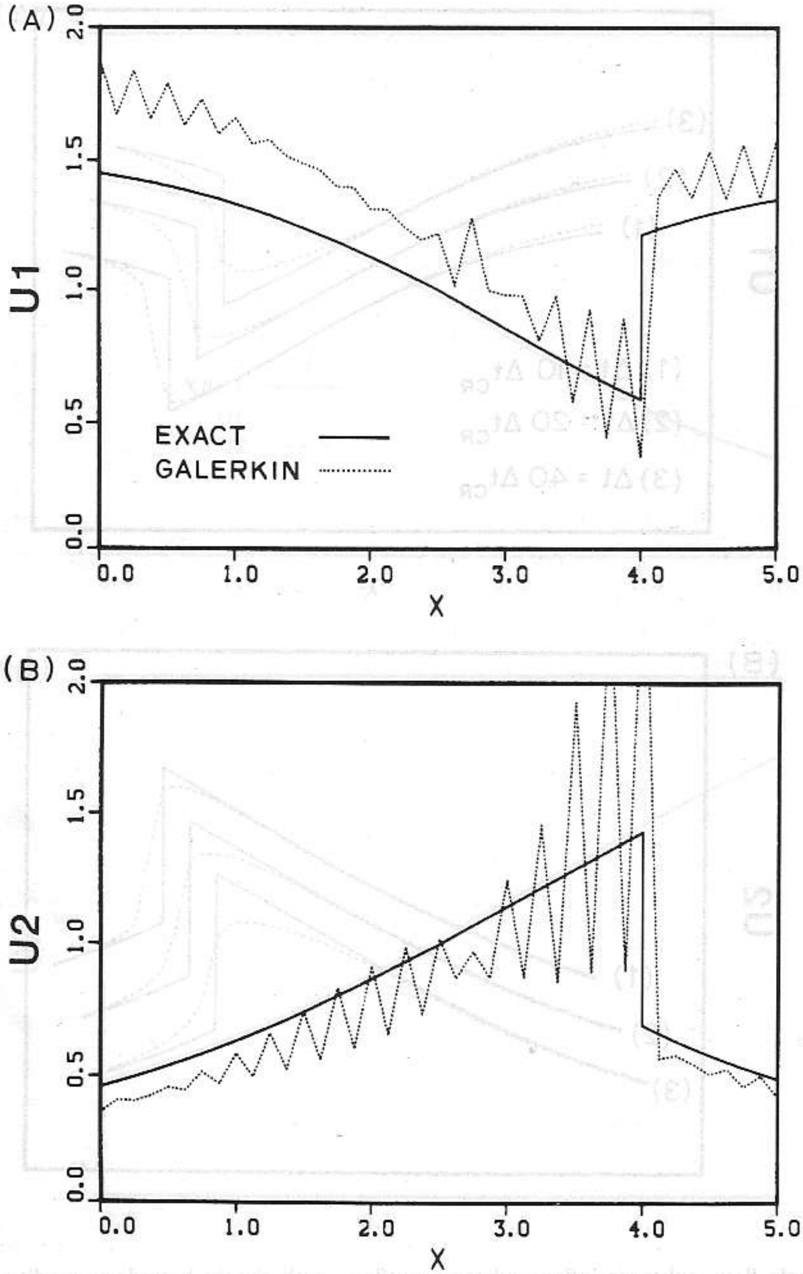

Fig. 22. Steady nozzle flow, subsonic inflow–subsonic outflow, with shock: boundary conditions U2U2, $n_{el} = 40$, $\Delta t = 0.500$.

Evaluation. The Petrov–Galerkin algorithm based on the A^2-form with spatial choice of τ is capable of very good shock resolution in the present steady flow cases. This may be seen from case (1) of Fig. 18. As will be confirmed subsequently, resolution of steady shocks is generally quite good. The oscillations exhibited in propagating cases are not seen in steady cases.

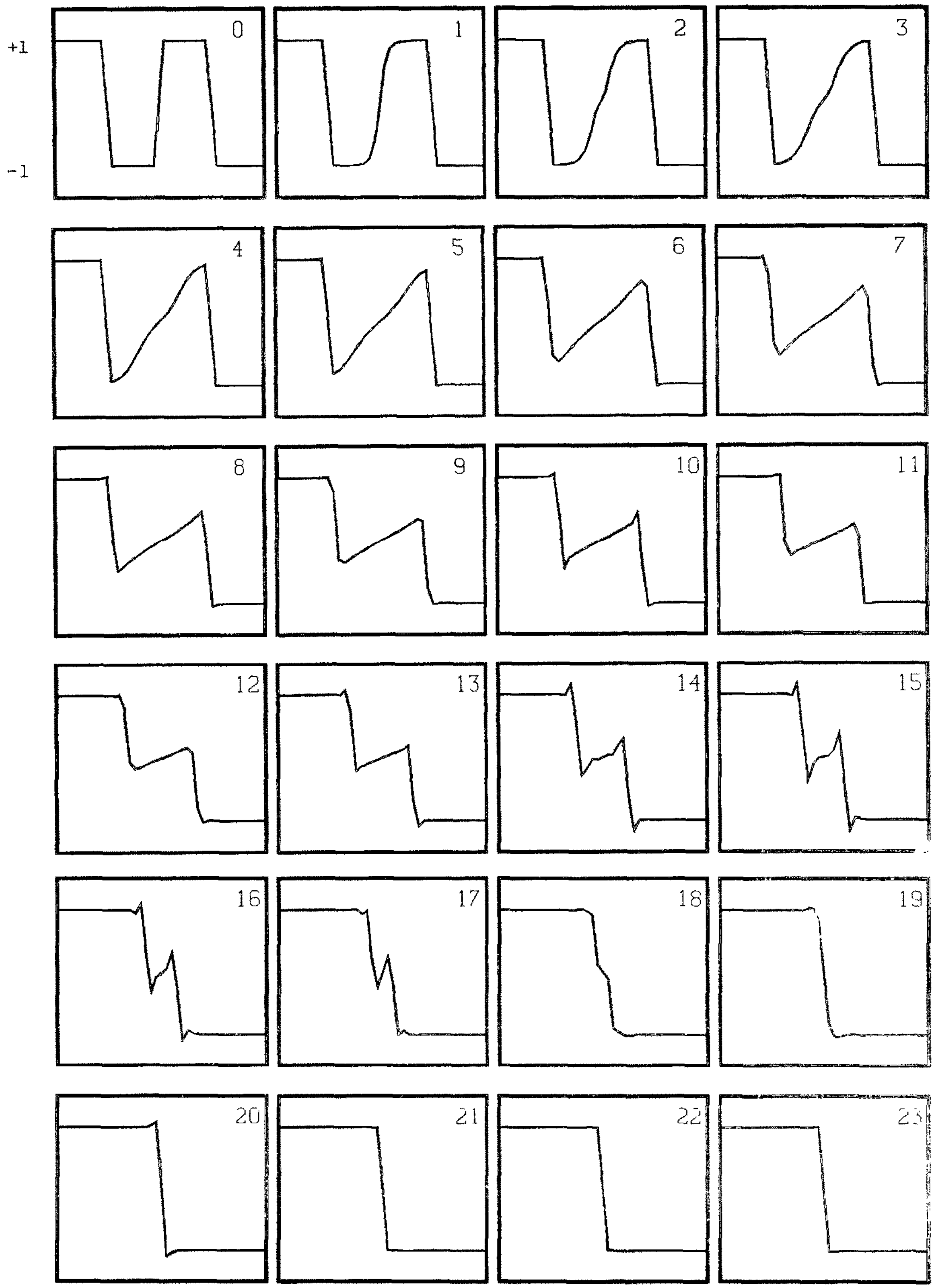

Fig. 23.

The present examples also provide some preliminary evidence of the superiority of the A^2-form over the A^tA-form (cf. curves (1) in Figs. 18 and 19). In two-dimensional calculations presented in Section 7, this superiority will become even more manifest.

Fig. 22 illustrates conclusively that Galerkin algorithms are inappropriate for shock problems (see also Fig. 10). However, by introduction of appropriate artificial viscosity mechanisms satisfactory behavior may be attainable.

6.5. *Shock structure/entropy condition test problem*

This problem was suggested to us by Osher. The equation is

$$u_{,t} + (\tfrac{1}{2}u^2)_{,x} = 0\,, \tag{6.40}$$

and the initial condition is shown in frame 0 of Fig. 23. Note that the initial condition involves two stable shocks and one unstable shock. Each successive frame corresponds to a time step ($\Delta t = 2.174$). As may be seen, the unstable shock immediately breaks down and eventually the stable shocks coalesce to form a steady shock (frame 23). The finite element scheme nodally interpolates the exact steady state. However, some overshoots are noted during the propagation of the shocks prior to reaching a steady state (see e.g. frames 13–16).

Equation (6.40) is a special case of the hyperbolic system formulation described in Section 2. In the calculation shown in Fig. 23, 40 linear elements were employed, the spatial criterion was employed, $F = 1$ and $\alpha = \frac{1}{2}$.

Evaluation. This problem confirms several observations made previously about the Petrov–Galerkin formulation. Firstly, unstable shocks are inadmissible[1]; secondly, correct stable shock structure is achieved; thirdly, steady shocks are resolved better than propagating shocks.

7. Numerical applications in two dimensions

7.1. *Thin biconvex airfoil*

7.1.1. *Problem geometry and boundary conditions*

We consider the problem of a thin biconvex airfoil placed in a uniform flow field. The axis of the parabolic arc is aligned with the direction of flow (non-lifting case). Fig. 24(a) shows the configuration; b denotes the ratio of the maximum airfoil thickness to the cord length. The subscript '∞' refers to the free stream.

We consider the half plane $x_2 \geq 0$ as our problem domain. The parabolic arc bounding the airfoil is described by the following expression:

$$x_2 = \tfrac{1}{2}b[1 - (2x_1)^2]\,. \tag{7.1}$$

The governing equations are the Euler equations described in Section 2 and Appendix A.

[1] It can be shown analytically that an unstable shock solution of (6.40) is inadmissable numerically as long as $\tau > 0$. Thus $\tau > 0$ represents an entropy condition in the present case.

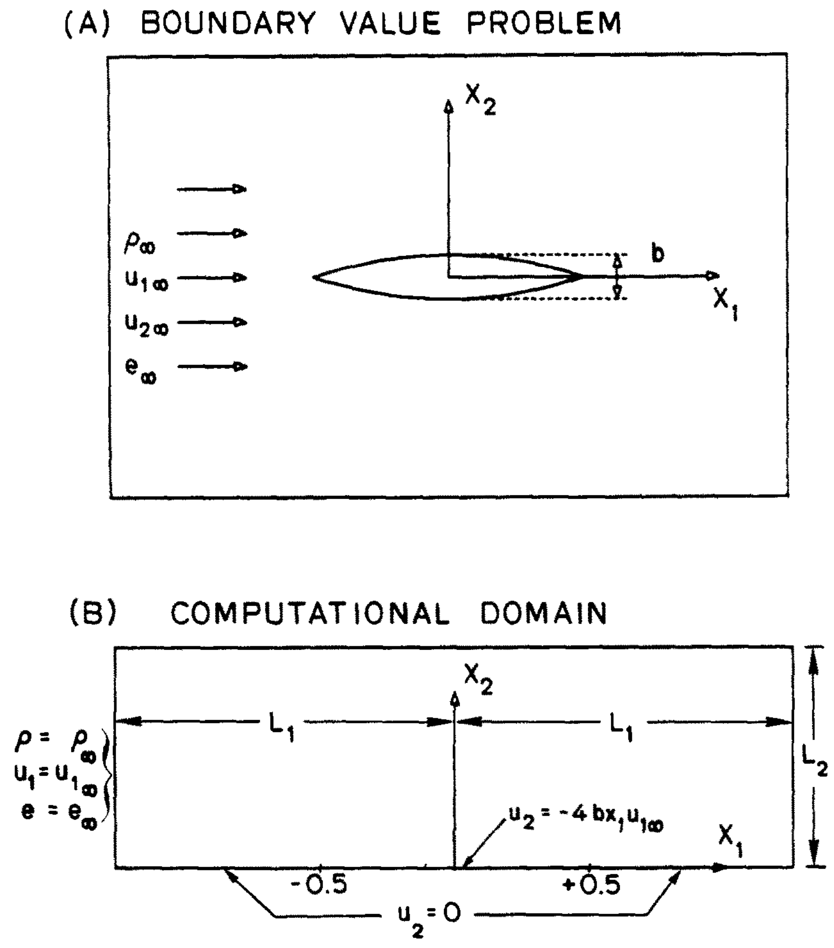

Fig. 24. Boundary value problem and computational domain for thin parabolic airfoil.

The ratio of the specific heats is

$$\gamma = 1.4. \tag{7.2}$$

The eigenvalues of the coefficient matrices A_1 and A_2 can be obtained from (A.37) by setting $(k_1, k_2) = (1, 0)$ and $(k_1, k_2) = (0, 1)$, respectively.

Boundary conditions. The free-stream parameters are taken to be

$$\rho_\infty = 1.0\,, \qquad u_{2\infty} = 0.0\,, \qquad e_\infty = 1.0\,. \tag{7.3}$$

The value of $u_{1\infty}$ will be set according to the following formula, which depends on the specified value of the free-stream Mach number M_∞:

$$u_{1\infty}^2 = \frac{M_\infty^2\gamma(\gamma-1)e}{\frac{1}{2}M_\infty^2\gamma(\gamma-1)+1}\,. \tag{7.4}$$

Along the x_1-axis, outside the airfoil, we impose the following condition on u_2:

$$u_2 = 0\,, \quad x_2 = 0\,, \; |x_1| > 0.5\,. \tag{7.5}$$

On the surface of the airfoil, the velocity vector must be perpendicular to the surface normal vector. This restriction can be expressed as

$$\frac{u_2}{u_1} = \frac{\mathrm{d}x_2}{\mathrm{d}x_1}\,. \tag{7.6}$$

Assuming that the airfoil thickness is small enough, such that the uniform flow field is perturbed only slightly, u_1 can be approximated in (7.6) by its free-stream value (see [40]),

$$\frac{u_2}{u_{1\infty}} = \frac{\mathrm{d}x_2}{\mathrm{d}x_1}\,. \tag{7.7}$$

From (7.1) and (7.7), the boundary condition on the surface of the airfoil can then be expressed as

$$u_2 = -4bx_1u_{1\infty}\,, \quad |x_1| \leq 0.5\,, \tag{7.8}$$

Nature of the solution. Once we set the free-stream parameters as given by (7.3) the nature of the solution depends on the free-stream Mach number and the airfoil thickness ratio. We fix the thickness ratio to be

$$b = 0.10\,, \tag{7.9}$$

and study the problem for two different values of the free-stream Mach number. The subcritical value $M_\infty = 0.5$ results in a symmetric subsonic solution, while the supercritical value $M_\infty = 0.84$ gives a solution with a shock around $x_1 = 0.3$.

7.1.2. Computational domain and algorithmic features

Finite element mesh and boundary conditions. The computational domain is shown in Fig. 24(b). We utilize three finite element meshes with different overall sizes: the medium and fine meshes with $L_1 = 3.5$, $L_2 = 3.0$, and the coarse mesh with $L_1 = 2.0$, $L_2 = 1.5$. Each mesh has $4N$ elements in the x_2-direction; in the x_1-direction, there are $8N$ elements across the airfoil and $4N$ elements each upstream and downstream. The number of nodal points are: $(16N + 1) \times (4N + 1)$, total, and $(8N + 1)$ on the airfoil. For the coarse mesh, $N = 1$, for the medium mesh $N = 2$, and for the fine mesh $N = 4$.

The meshes are shown in Fig. 25; they are symmetric with respect to the x_2-axis.

At the left boundary we set ρ, u_1 and e to their free-stream values. Along the x_1-axis we take the boundary conditions of (7.5) and (7.8). Imposing the boundary condition of (7.8) along the x_1-axis instead of on the airfoil surface is a standard thin-airfoil approximation. At the upper boundary, we impose the condition $u_2 = 0$, which can physically be interpreted as a

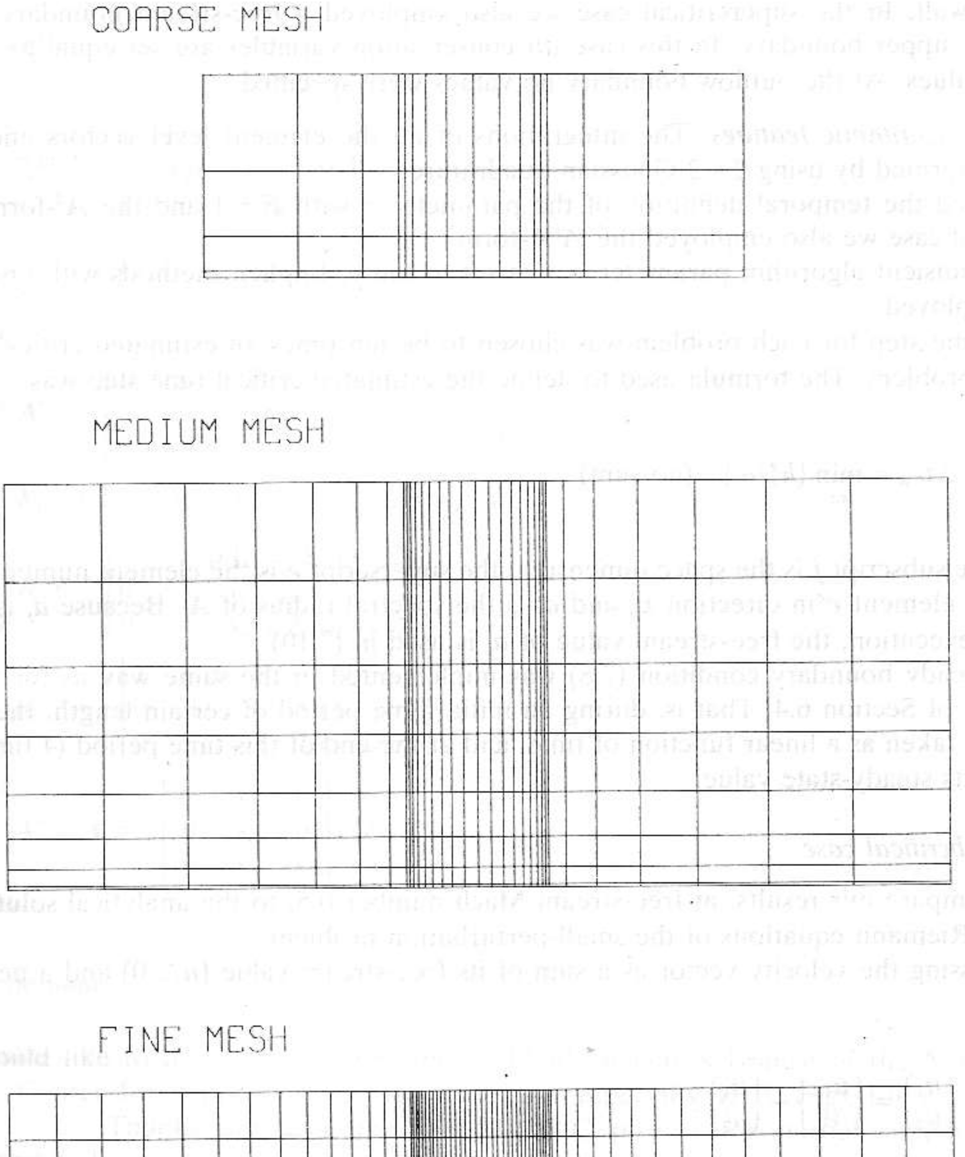

Fig. 25. Finite element meshes. Coarse mesh, 64 elements; medium mesh, 256 elements; fine mesh, 1016 elements.

channel wall. In the supercritical case we also employed a free-stream boundary condition along the upper boundary. In this case *all* conservation variables are set equal to their free-stream values. At the outflow boundary no values were specified.

Other algorithmic features. The integrations of all the element level vectors and matrices were performed by using 2×2 Gaussian quadrature.

We used the temporal definition of the parameter τ with $F=1$ and the A^2-form. For the subcritical case we also employed the $A^{\mathrm{t}}A$-form.

The transient algorithm parameter α was set to unity. Implicit methods with one iteration were employed.

The time step for each problem was chosen to be ten times an estimated critical time step for that problem. The formula used to define the estimated critical time step was

$$\Delta t_{\mathrm{CR}} = \min_{e,j} \{h_j^e/a_j\} \quad \text{(no sum)} \tag{7.10}$$

where the subscript j is the space dimension, the superscript e is the element number, h_j^e is the length of element e in direction x_j, and a_j is the spectral radius of $\boldsymbol{A}_j$. Because a_j is unknown prior to execution, the free-stream value of a_j is used in (7.10).

The steady boundary condition (7.8) was implemented in the same way as for the nozzle problems of Section 6.4. That is, during an initial time period of certain length, the thickness ratio was taken as a linear function of time, and at the end of this time period (4 time steps) it reached its steady-state value.

7.1.3. Subcritical case

We compare our results, at free-stream Mach number 0.5, to the analytical solution of the Cauchy–Riemann equations of the small-perturbation problem.

Expressing the velocity vector as a sum of its free-stream value $(u_{1\infty}, 0)$ and a perturbation (u_1', u_2'),

$$\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} u_{1\infty} \\ 0 \end{Bmatrix} + \begin{Bmatrix} u_1' \\ u_2' \end{Bmatrix}, \tag{7.11}$$

the analytical solution (see [41]) can be written as

$$\beta u_1' - \mathrm{i} u_2' = u_{1\infty} b \frac{4}{\pi}\left(1 - Z \ln\left(\frac{Z+0.5}{Z-0.5}\right)\right) \tag{7.12}$$

where $Z = x_1 + \mathrm{i}\beta x_2$, and

$$\beta = (1 - M_\infty)^{1/2}. \tag{7.13}$$

One can observe from (7.12) that, along the x_1-axis, the boundary condition for u_2 has been satisfied, while, for u_1', the following expression is obtained:

$$u_1' = \frac{1}{\beta} u_{1\infty} b \frac{4}{\pi} \left(1 - x_1 \ln\left|\frac{x_1 + 0.5}{x_1 - 0.5}\right|\right). \tag{7.14}$$

The pressure coefficient C_P, defined as

$$C_P = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty (u_{1\infty})^2}, \tag{7.15}$$

can be obtained by the following equation (see [40]):

$$C_P = -\left[2\left(\frac{u_1'}{u_{1\infty}}\right) + \left(\beta \frac{u_1'}{u_{1\infty}}\right)^2 + \left(\frac{u_2'}{u_{1\infty}}\right)^2\right]. \tag{7.16}$$

If the second-order terms are neglected, then we get

$$C_P = -2\left(\frac{u_1'}{u_{1\infty}}\right). \tag{7.17}$$

Finite element solutions. Fig. 26 shows the analytical and finite element (medium and coarse mesh) solution for subcritical flow at $M_\infty = 0.5$.

The time steps for the medium and coarse meshes were set to 0.23 and 0.46, respectively. The solutions at the end of 30 steps were taken as steady state solutions. For the medium mesh, convergence was achieved in about 20 steps.

The finite element solutions for A^2 and A^tA-forms are shown in Figs. 26(a), 26(b) and 26(c), 26(d), respectively. The values of C_P('CP') and u_1('U') are plotted along the airfoil. The numerical solutions are in good agreement with the analytical solution except for the variable u_1 (i.e. U) when the A^tA-form is used. This considerable discrepancy is the main empirical reason we have for favoring the A^2-form. When it comes to other variables, such as ρ and e, similar discrepancies were observed between the solutions produced by A^2- and A^tA-forms. In subsequent calculations only the A^2-form was employed.

REMARK 7.1. The analytical solution predicts infinite, and thus discontinuous, values at the leading and trailing edges.

REMARK 7.2. The computational boundaries did not seem to notably influence the solution near the airfoil.

REMARK 7.3. It is interesting to observe that the results for the coarse mesh are in close agreement with the analytical solution.

REMARK 7.4. The Galerkin algorithm produced highly oscillatory results which fed back into the operators and caused divergence.

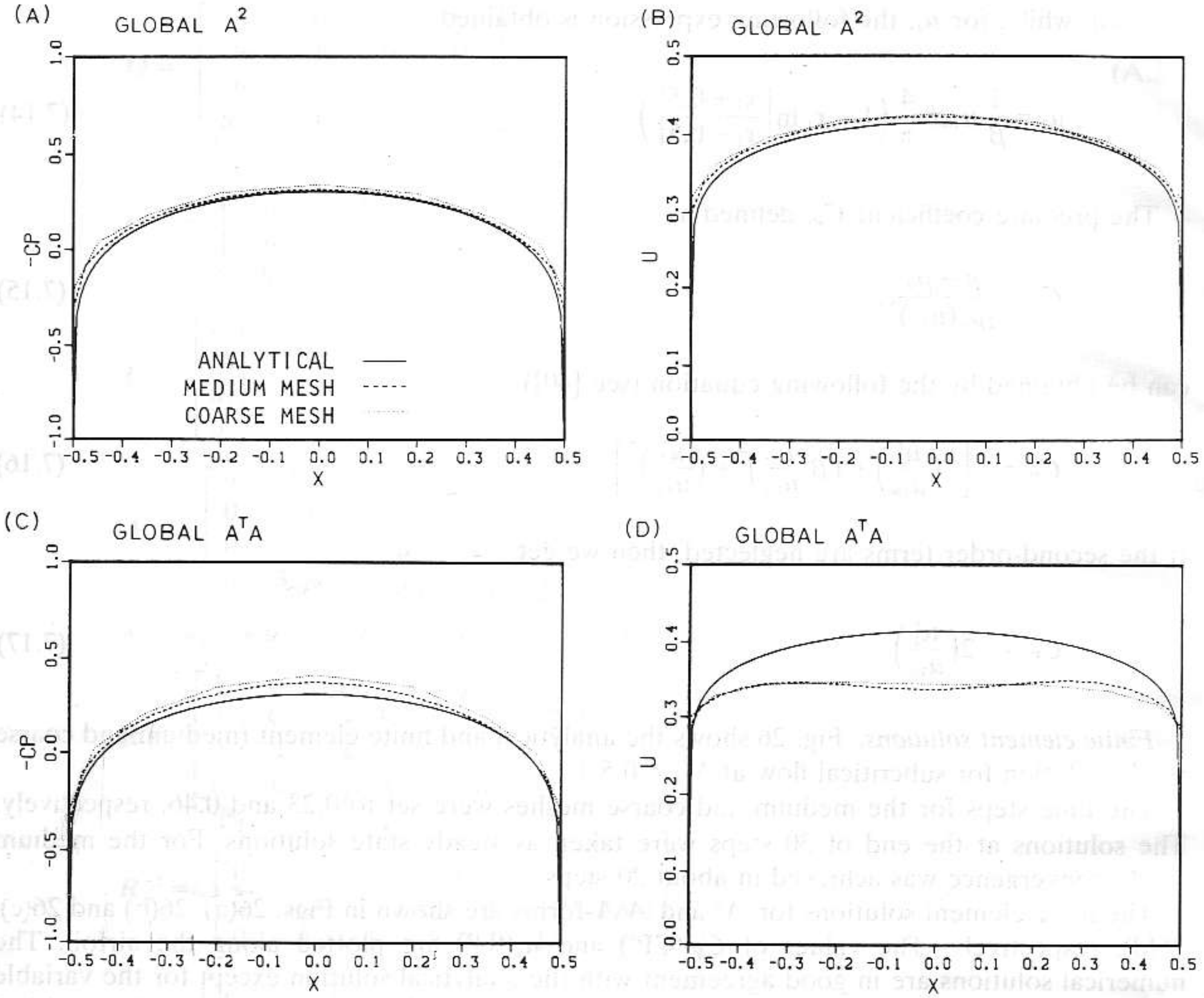

Fig. 26. Steady state solutions for 10% thickness airfoil, subcritical flow, $M_\infty = 0.50$: Δt (medium mesh) = 0.23, Δt (coarse mesh) = 0.46.

7.1.4. Supercritical case

Finite element solutions. The time step was set to 0.20 for the medium and fine meshes. The 120-step solutions were taken as the steady states. Pressure coefficient plots are presented in Fig. 27 for channel and free-stream boundary conditions along the upper edge. The results show a shift in the shock location of about 6% for the two boundary condition cases. This is to be expected because the wall boundary conditions produce a Mach number higher than the free-stream value along the upper boundary (about 0.85–0.86). It is known that as the free-stream Mach number gets higher, the shock front moves downstream.

Continuous-tone color results are presented in Fig. 31.

Evaluation. The results for the thin airfoil problem are viewed as very good. Both the medium and fine meshes produced non-oscillatory shocks resolved over one or, at most, two

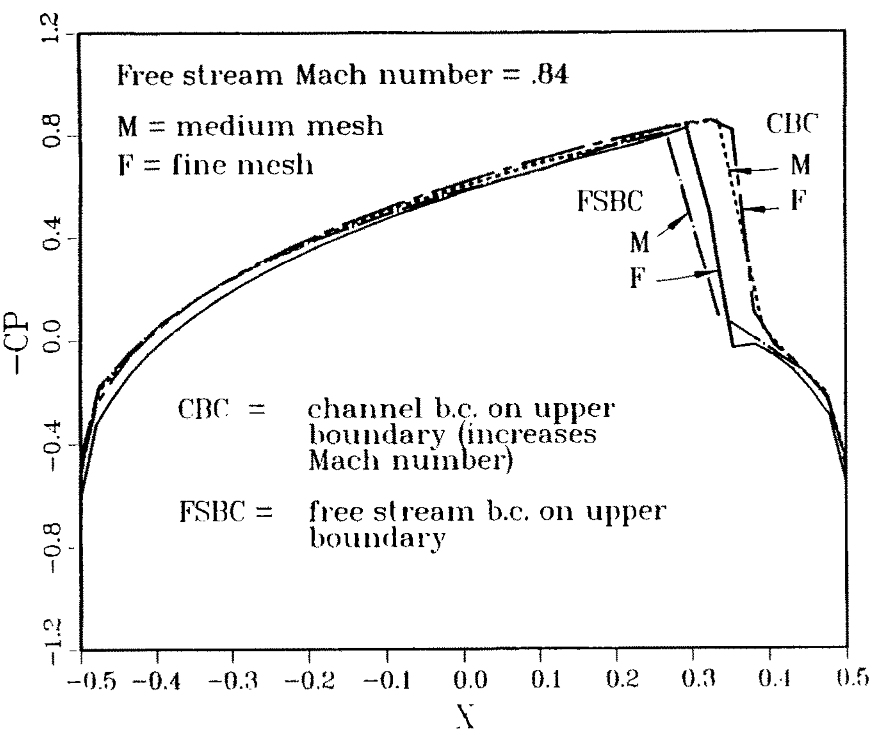

Fig. 27. Pressure coefficient for biconvex airfoil.

element lengths. The medium mesh is considered quite coarse for a shock problem of this type, yet very good results were obtained. The present results again confirm the ability of the Petrov–Galerkin formulation to accurately capture steady state shock-wave phenomena.

7.2. *Unsteady flow about a cylinder*

The computational domain and boundary conditions are shown in Fig. 28. The free-stream parameters are given as follows:

$$\rho_\infty = 1\,, \qquad u_{1\infty} = M_\infty = 0.5\,, \qquad u_{2\infty} = 0\,, \qquad c_\infty = 1\,.$$

With these four free-stream parameters one can calculate the free-stream value of the total specific energy e_∞.

The free-stream values are taken as initial conditions. The finite element mesh employed is shown in Figs. 29 and 30. It involves a total of 2928 elements. About the cylinder there are 29 elements in the radial direction and 96 elements in the circumferential direction. The total number of nodes is 3030. The parameter τ was chosen according to the global criterion with $F = 1$. A 1-pass explicit transient algorithm was employed with

$$\alpha = 1.0\,, \qquad \Delta t = 0.02\,.$$

Thus $\tau = F\alpha\,\Delta t = 0.02$.

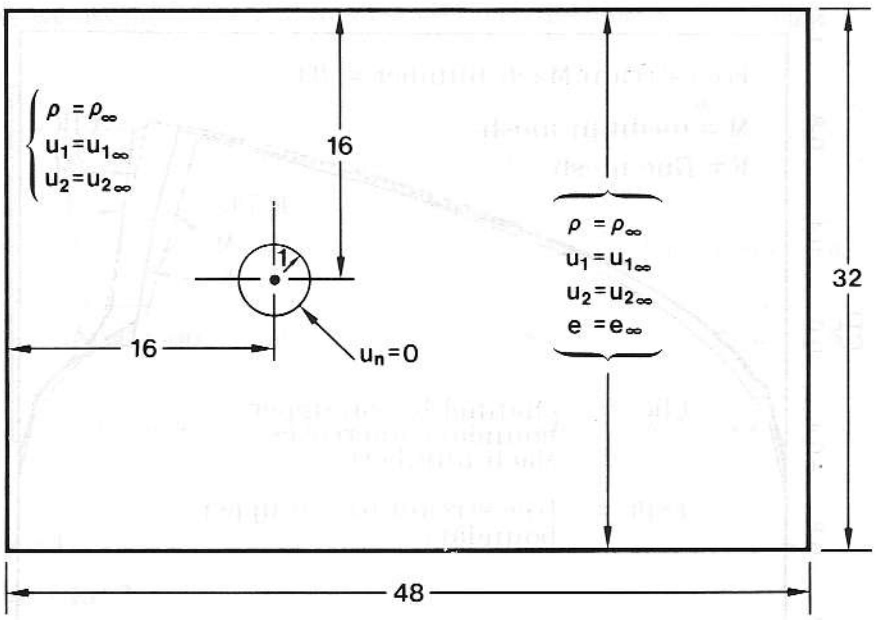

Fig. 28. Computational domain and boundary conditions.

An estimate of the critical time step may be deduced as follows: For a length scale, take the edge length of the elements about the cylinder. These are the smallest elements in the mesh, they are approximately square, with edge length approximately 0.065. For a characteristic velocity, take the free-stream acoustic speed (i.e. 1.0) and add to it an estimated maximum particle velocity (say 2.0). This leads to a critical time step estimate of $\Delta t_{\mathrm{CR}} = 0.065/3.0 = 0.0216$.

Fig. 29. Finite element mesh.

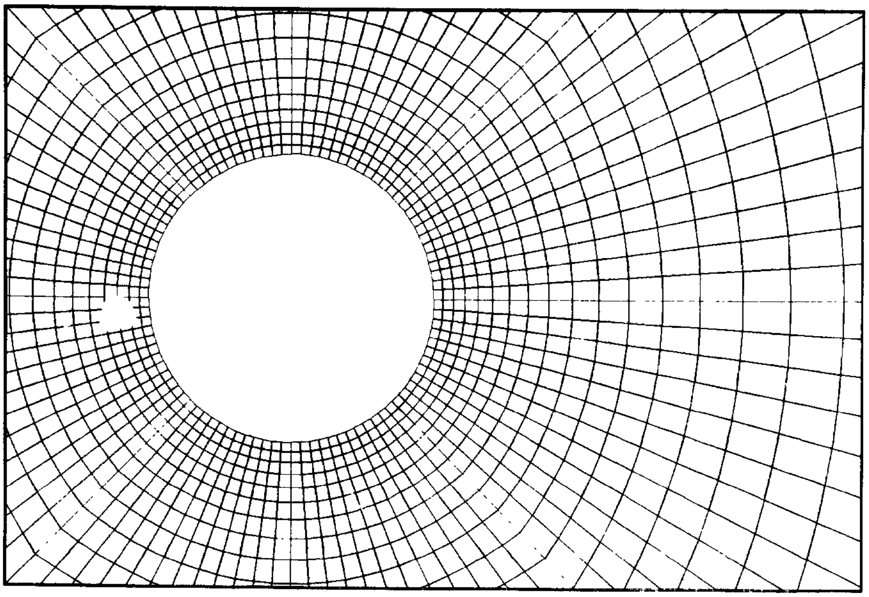

Fig. 30. Detail of finite element mesh.

In the numerical results, four distinct phases are noticeable: (i) initially shocks form symmetrically about the cylinder; (ii) thereafter, symmetrical eddies form behind the cylinder; (iii) the eddies are shed symmetrically in pairs; (iv) finally, the symmetry breaks down, first in the wake, then eddies are shed unsymmetrically and periodically in a fashion similar to the Karman vortex street. No effort was made to induce unsymmetrical behavior, its cause is the unsymmetrical propagation of round-off errors. Needless to say, this mode of breaking down of what appears to be a physically unstable symmetric solution is very time-consuming. What constitutes a correct solution to this problem is not known. However, a number of investigators, employing very different approaches, domains, meshes, boundary conditions, etc., have all calculated unsymmetrical vortex shedding (see [7, 53] for a summary of calculations to date).

Results are presented in Figs. 31 and 32. Some early-time results on a crude mesh are presented in [34].

Evaluation. From a numerical standpoint, the present problem is viewed as difficult in the sense that even obtaining a solution is something of an achievement. However, we are not completely satisfied with the results because they tend to be more oscillatory about shocks than we believe they should be. This is consistent with our previous observations about the behavior of the methods developed when unsteady shock phenomena is present. On the other hand, the unsymmetrical shedding of vortices is at least qualitatively consistent with the results of other investigators. Because so little is presently known about the exact solution to the problem of unsteady Euler flow about a cylinder, it is unfortunately impossible to evaluate the results on this basis.

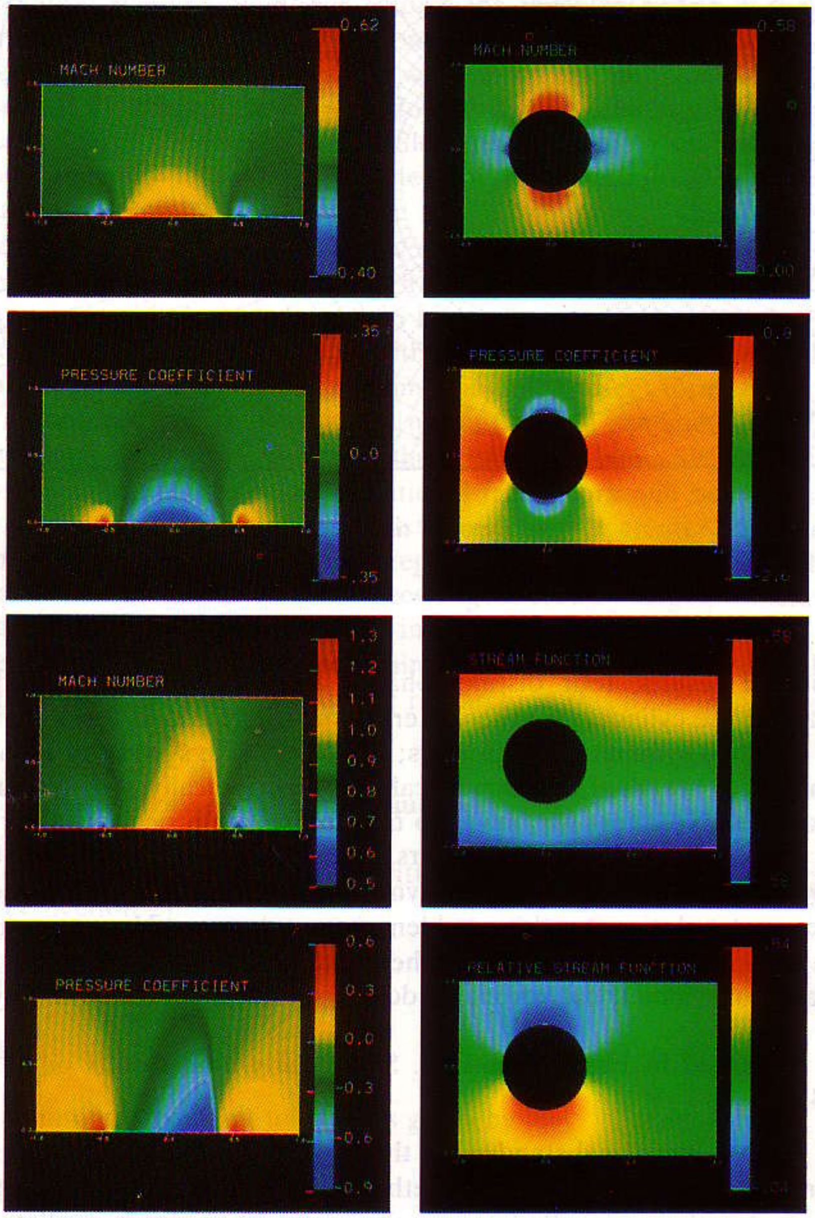

Fig. 31. First column: Mach number and pressure for the thin airfoil problem described in Section 7.1. The first two frames are for the subsonic case and the second two frames are for the transonic case. *Second column*: subcritical flow about a cylinder. A 336 element mesh (not shown) was used to obtain a steady flow for a free stream Mach number of 0.3.

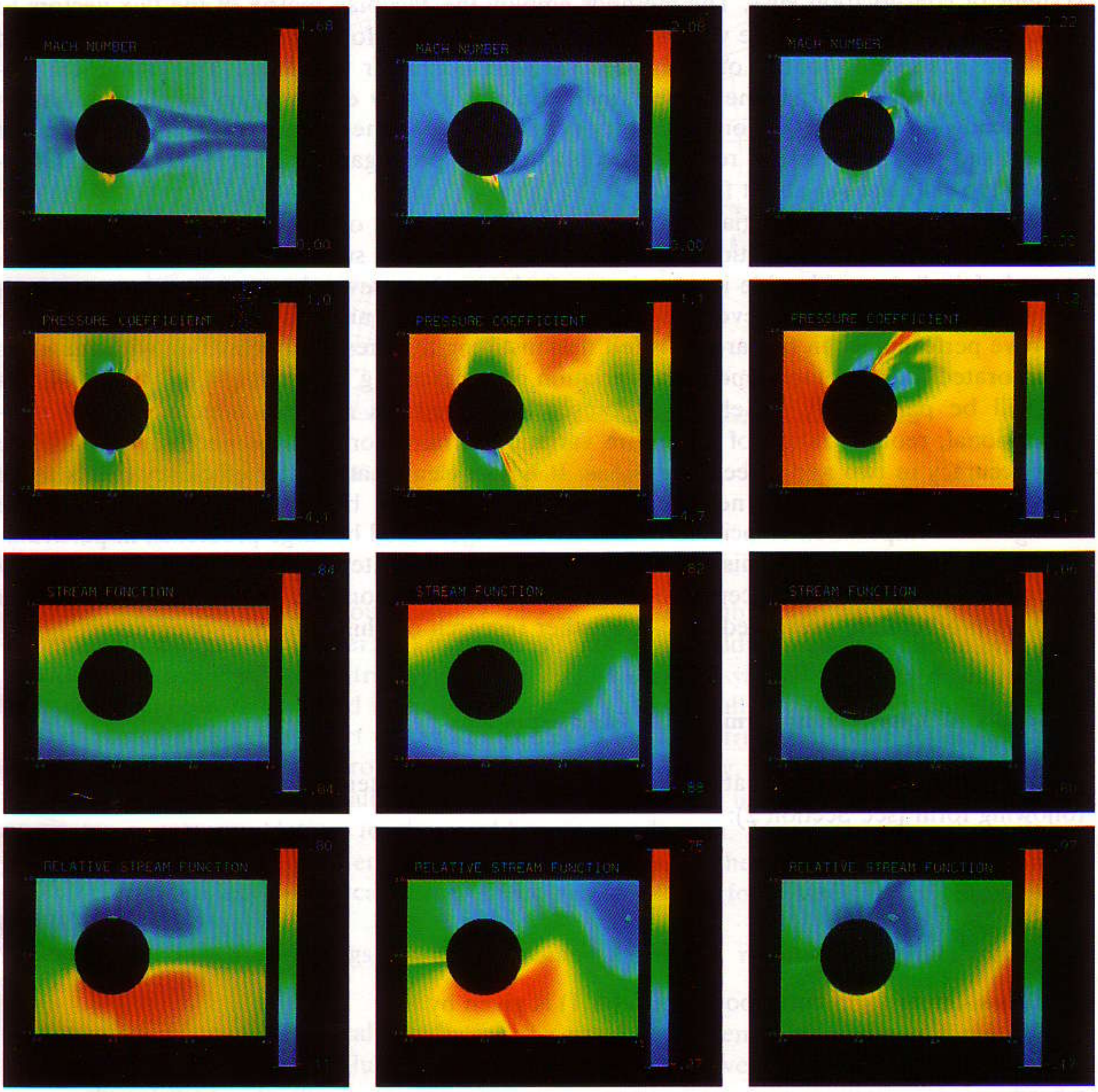

Fig. 32. Unsteady results for the problem described in Section 7.2. Results just prior to the breakdown of the symmetrical solution are presented in the first column (step 3660). Slight asymmetries in the wake may be seen. Unsymmetrical shedding is exhibited in the second and third columns (steps 4400 and 4900, respectively).

8. Conclusions

In this paper we have investigated a class of Petrov–Galerkin methods for hyperbolic systems of conservation laws. The methods employ the Jacobian matrix of the flux vectors to define the perturbation of the weighting function. Two basic forms are studied and it is found that one is superior to the other in dealing with nonlinear problems. The numerical calculations demonstrate that the present methodology is fairly effective for handling solutions with steady shocks. Sharp, non-oscillatory profiles are obtained in several examples. On the other hand, less satisfactory results are obtained for propagating shock phenomena. Oscillations typically appear about fronts.

The present work is an initiatory effort in the development of finite element procedures for the compressible Euler equations and, although a degree of success has been achieved, it is strongly felt that considerable improvement can be made in several areas. A prime concern is the level of robustness achieved, especially with respect to unsteady shock phenomena. To improve performance in this area it is felt that instilling the present formulation with the ideas incorporated in 'monotone' and 'total-variation non-increasing' schemes [14, 17, 19–21, 44, 47–51] will be particularly beneficial. Because these methods are still effectively only one-dimensional, reorganization of the entire variational framework will be necessary to achieve high accuracy in two and three dimensions. It is also clear that greater efficiency needs to be attained and fundamentally new implicit strategies need to be developed which avoid the storage and computation associated with matrices engendered by large problems, in particular three-dimensional ones. In this regard, we are investigating iterative strategies which employ the element-by-element concept. Some preliminary results on the transonic flow around a cylinder problem are presented in [34]. Further research on this topic is in progress.

Appendix A. Quasi-linear forms of the Euler equations

Recall that the Euler equations may be written as a system of conservation laws of the following form (see Section 2):

$$\boldsymbol{U}_{,t} + \boldsymbol{F}_{j,j} = \boldsymbol{0} \tag{A.1}$$

where

$$\boldsymbol{F}_j = \boldsymbol{F}_j(\boldsymbol{U})\,. \tag{A.2}$$

The quasi-linear form corresponding to (A.1) is given by

$$\boldsymbol{U}_{,t} + \boldsymbol{A}_j\boldsymbol{U}_{,j} = \boldsymbol{0} \tag{A.3}$$

where

$$\boldsymbol{A}_j = \partial\boldsymbol{F}_j/\partial\boldsymbol{U}\,. \tag{A.4}$$

Various other quasi-linear forms are also of interest, for example, consider the linear combination of coefficient matrices

$$\boldsymbol{A} = k_j\boldsymbol{A}_j \tag{A.5}$$

where $\boldsymbol{k} = \{k_j\} \in \mathbb{R}^{n_{sd}}$. (We shall assume $\boldsymbol{k}$ is a unit vector.) There exists a matrix $\boldsymbol{S}$ such that

$$\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S} = \boldsymbol{\Lambda} \tag{A.6}$$

where $\boldsymbol{\Lambda}$ is a real diagonal matrix (i.e., the eigenvalues of $\boldsymbol{A}$). *For the Euler equations, the similarity transformation defined by* (A.6) *simultaneously symmetrizes the individual* $\boldsymbol{A}_j$. (This, of course, does not hold for hyperbolic systems in general.) To determine $\boldsymbol{S}$, it is necessary to perform an eigenvalue–eigenvector calculation for $\boldsymbol{A}$ (see [59–61]). It is instructive to perform this calculation in two phases.

The first phase consists of a transformation into *primitive variables form*:

$$\boldsymbol{U}'_{,t} + \boldsymbol{A}'_j\boldsymbol{U}'_{,j} = \boldsymbol{0} \tag{A.7}$$

where

$$\boldsymbol{A}'_j = \boldsymbol{Q}^{-1}\boldsymbol{A}_j\boldsymbol{Q}, \tag{A.8}$$

$$\boldsymbol{Q} = \partial\boldsymbol{U}/\partial\boldsymbol{U}' \tag{A.9}$$

and

$$\boldsymbol{A}' = k_j\boldsymbol{A}'_j. \tag{A.10}$$

Here '′' refers to the frame of primitive variables. In this frame, the coefficient matrices have simpler forms, thus, it is easier to find the eigenvalues and eigenvectors.

In the second phase, operators which diagonalize $\boldsymbol{A}'$ are constructed and another transformation is performed:

$$\boldsymbol{U}''_{,t} + \boldsymbol{A}''_j\boldsymbol{U}''_{,j} = \boldsymbol{0} \tag{A.11}$$

where

$$\boldsymbol{A}''_j = \boldsymbol{R}^{-1}\boldsymbol{A}'_j\boldsymbol{R} = (\boldsymbol{Q}\boldsymbol{R})^{-1}\boldsymbol{A}_j(\boldsymbol{Q}\boldsymbol{R}), \tag{A.12}$$

$$\boldsymbol{R} = \partial\boldsymbol{U}'/\partial\boldsymbol{U}'' \tag{A.13}$$

and

$$\boldsymbol{A}'' = k_j\boldsymbol{A}''_j. \tag{A.14}$$

Here '″' refers to the frame in which $\boldsymbol{A}''$ is a diagonal matrix and the $\boldsymbol{A}''_j$ are symmetric. The transformation matrix $\boldsymbol{S}$ is, then, equal to the product $\boldsymbol{QR}$.

In the following sections we list explicit formulae for the arrays described above.

A.1. Three-dimensional case

$$\boldsymbol{U} = \rho\begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e \end{Bmatrix}, \tag{A.15}$$

$$\boldsymbol{F}_j = \begin{Bmatrix} u_j\rho \\ u_j\rho u_1 + \delta_{j1}p \\ u_j\rho u_2 + \delta_{j2}p \\ u_j\rho u_3 + \delta_{j3}p \\ u_j(\rho e + p) \end{Bmatrix}. \tag{A.16}$$

In what follows we assume the equation of state takes the form $p = (\gamma - 1)\rho\varepsilon$.

$$\boldsymbol{A}_j = \left[\begin{array}{c|c|c|c|c} 0 & \delta_{j1} & \delta_{j2} & \delta_{j3} & 0 \\ \hline \tfrac{1}{2}\delta_{j1}\bar{\gamma}u^2 - u_j u_1 & \begin{array}{c}\delta_{j1}u_1 - \delta_{j1}\bar{\gamma}u_1 \\ +u_j\end{array} & \delta_{j2}u_1 - \delta_{j1}\bar{\gamma}u_2 & \delta_{j3}u_1 - \delta_{j1}\bar{\gamma}u_3 & \delta_{j1}\bar{\gamma} \\ \hline \tfrac{1}{2}\delta_{j2}\bar{\gamma}u^2 - u_j u_2 & \delta_{j1}u_2 - \delta_{j2}\bar{\gamma}u_1 & \begin{array}{c}\delta_{j2}u_2 - \delta_{j2}\bar{\gamma}u_2 \\ +u_j\end{array} & \delta_{j3}u_2 - \delta_{j2}\bar{\gamma}u_3 & \delta_{j2}\bar{\gamma} \\ \hline \tfrac{1}{2}\delta_{j3}\bar{\gamma}u^2 - u_j u_3 & \delta_{j1}u_3 - \delta_{j3}\bar{\gamma}u_1 & \delta_{j2}u_3 - \delta_{j3}\bar{\gamma}u_2 & \begin{array}{c}\delta_{j3}u_3 - \delta_{j3}\bar{\gamma}u_3 \\ +u_j\end{array} & \delta_{j3}\bar{\gamma} \\ \hline (\bar{\gamma}u^2 - \gamma e)u_j & \delta_{j1}\bar{\varepsilon} - \bar{\gamma}u_j u_1 & \delta_{j2}\bar{\varepsilon} - \bar{\gamma}u_j u_2 & \delta_{j3}\bar{\varepsilon} - \bar{\gamma}u_j u_3 & \gamma u_j \end{array}\right] \tag{A.17}$$

where

$$\bar{\gamma} = \gamma - 1\,, \tag{A.18}$$

$$\bar{\varepsilon} = \gamma e - \bar{\gamma}u^2/2\,. \tag{A.19}$$

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u_1 & \rho & 0 & 0 & 0 \\ u_2 & 0 & \rho & 0 & 0 \\ u_3 & 0 & 0 & \rho & 0 \\ \tfrac{1}{2}u^2 & \rho u_1 & \rho u_2 & \rho u_3 & 1/\bar{\gamma} \end{bmatrix}, \tag{A.20}$$

$$\boldsymbol{Q}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -u_1/\rho & 1/\rho & 0 & 0 & 0 \\ -u_2/\rho & 0 & 1/\rho & 0 & 0 \\ -u_3/\rho & 0 & 0 & 1/\rho & 0 \\ \tfrac{1}{2}\bar{\gamma}u^2 & -\bar{\gamma}u_1 & -\bar{\gamma}u_2 & -\bar{\gamma}u_3 & \bar{\gamma} \end{bmatrix}, \tag{A.21}$$

$$\boldsymbol{U}' = \begin{Bmatrix} \rho \\ u_1 \\ u_2 \\ u_3 \\ p \end{Bmatrix}, \tag{A.22}$$

$$\boldsymbol{A}'_j = \begin{bmatrix} u_j & \delta_{j1}\rho & \delta_{j2}\rho & \delta_{j3}\rho & 0 \\ 0 & u_j & 0 & 0 & \delta_{j1}/\rho \\ 0 & 0 & u_j & 0 & \delta_{j2}/\rho \\ 0 & 0 & 0 & u_j & \delta_{j3}/\rho \\ 0 & \delta_{j1}\rho c^2 & \delta_{j2}\rho c^2 & \delta_{j3}\rho c^2 & u_j \end{bmatrix}, \tag{A.23}$$

where c, the acoustic speed, is defined by

$$c^2 = \gamma p/\rho\,. \tag{A.24}$$

The eigenvalues of $\boldsymbol{A}$ are

$$\lambda_1 = \lambda_2 = \lambda_3 = k_j u_j\,, \qquad \lambda_4 = \lambda_1 + c\,, \qquad \lambda_5 = \lambda_1 - c\,, \tag{A.25}$$

$$\boldsymbol{R} = \begin{bmatrix} k_1 & k_2 & k_3 & \rho/(\sqrt{2}c) & \rho/(\sqrt{2}c) \\ 0 & -k_3 & k_2 & k_1/\sqrt{2} & -k_1/\sqrt{2} \\ k_3 & 0 & -k_1 & k_2/\sqrt{2} & -k_2/\sqrt{2} \\ -k_2 & k_1 & 0 & k_3/\sqrt{2} & -k_3/\sqrt{2} \\ 0 & 0 & 0 & \rho c/\sqrt{2} & \rho c/\sqrt{2} \end{bmatrix}, \tag{A.26}$$

$$\boldsymbol{R}^{-1} = \begin{bmatrix} k_1 & 0 & k_3 & -k_2 & -k_1/c^2 \\ k_2 & -k_3 & 0 & k_1 & -k_2/c^2 \\ k_3 & k_2 & -k_1 & 0 & -k_3/c^2 \\ 0 & k_1/\sqrt{2} & k_2/\sqrt{2} & k_3/\sqrt{2} & 1/(\sqrt{2}\rho c) \\ 0 & -k_1/\sqrt{2} & -k_2/\sqrt{2} & -k_3/\sqrt{2} & 1/(\sqrt{2}\rho c) \end{bmatrix}, \tag{A.27}$$

$$\boldsymbol{A}''_j = \begin{bmatrix} u_j & 0 & 0 & \delta_{j2}\bar{k}_3 - \delta_{j3}\bar{k}_2 & \delta_{j2}\bar{k}_3 - \delta_{j3}\bar{k}_2 \\ 0 & u_j & 0 & \delta_{j3}\bar{k}_1 - \delta_{j1}\bar{k}_3 & \delta_{j3}\bar{k}_1 - \delta_{j1}\bar{k}_3 \\ 0 & 0 & u_j & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & \delta_{j1}k_2 - \delta_{j2}\bar{k}_1 \\ & & & & \\ \delta_{j2}\bar{k}_3 - \delta_{j3}\bar{k}_2 & \delta_{j3}\bar{k}_1 - \delta_{j1}\bar{k}_3 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & u_j + ck_j & 0 \\ \delta_{j2}\bar{k}_3 - \delta_{j3}\bar{k}_2 & \delta_{j3}\bar{k}_1 - \delta_{j1}\bar{k}_3 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & 0 & u_j - ck_j \end{bmatrix}, \tag{A.28}$$

where

$$\bar{k}_j = ck_j/\sqrt{2}\,. \tag{A.29}$$

A.2. Two-dimensional case

$$\boldsymbol{U} = \rho\begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ e \end{Bmatrix}, \tag{A.30}$$

$$\boldsymbol{F}_j = \begin{Bmatrix} u_j\rho \\ u_j\rho u_1 + \delta_{j1}p \\ u_j\rho u_2 + \delta_{j2}p \\ u_j(\rho e + p) \end{Bmatrix}, \tag{A.31}$$

$$\boldsymbol{A}_j = \left[\begin{array}{c|c|c|c} 0 & \delta_{j1} & \delta_{j2} & 0 \\ \hline \frac{1}{2}\delta_{j1}\bar{\gamma}u^2 - u_j u_1 & \begin{array}{c}\delta_{j1}u_1 - \delta_{j1}\bar{\gamma}u_1 \\ +u_j\end{array} & \delta_{j2}u_1 - \delta_{j1}\bar{\gamma}u_2 & \delta_{j1}\bar{\gamma} \\ \hline \frac{1}{2}\delta_{j2}\bar{\gamma}u^2 - u_j u_2 & \delta_{j1}u_2 - \delta_{j2}\bar{\gamma}u_1 & \begin{array}{c}\delta_{j2}u_2 - \delta_{j2}\bar{\gamma}u_2 \\ +u_j\end{array} & \delta_{j2}\bar{\gamma} \\ \hline (\bar{\gamma}u^2 - \gamma e)u_j & \delta_{j1}\bar{\varepsilon} - \bar{\gamma}u_j u_1 & \delta_{j2}\bar{\varepsilon} - \bar{\gamma}u_j u_2 & \gamma u_j \end{array}\right], \tag{A.32}$$

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ u_1 & \rho & 0 & 0 \\ u_2 & 0 & \rho & 0 \\ u^2/2 & \rho u_1 & \rho u_2 & 1/\bar{\gamma} \end{bmatrix}, \tag{A.33}$$

$$\boldsymbol{Q}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -u_1/\rho & 1/\rho & 0 & 0 \\ -u_2/\rho & 0 & 1/\rho & 0 \\ \bar{\gamma}u^2/2 & -\bar{\gamma}u_1 & -\bar{\gamma}u_2 & \bar{\gamma} \end{bmatrix}, \tag{A.34}$$

$$\boldsymbol{U}' = \begin{Bmatrix} \rho \\ u_1 \\ u_2 \\ p \end{Bmatrix}, \tag{A.35}$$

$$\boldsymbol{A}'_j = \begin{bmatrix} u_j & \delta_{j1}\rho & \delta_{j2}\rho & 0 \\ 0 & u_j & 0 & \delta_{j1}/\rho \\ 0 & 0 & u_j & \delta_{j2}/\rho \\ 0 & \delta_{j1}\rho c^2 & \delta_{j2}\rho c^2 & u_j \end{bmatrix}, \tag{A.36}$$

$$\lambda_1 = \lambda_2 = k_j u_j, \qquad \lambda_3 = \lambda_1 + c, \qquad \lambda_4 = \lambda_1 - c, \tag{A.37}$$

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & \rho/(\sqrt{2}c) & \rho/(\sqrt{2}c) \\ 0 & k_2 & k_1/\sqrt{2} & -k_1/\sqrt{2} \\ 0 & -k_1 & k_2/\sqrt{2} & -k_2/\sqrt{2} \\ 0 & 0 & \rho c/\sqrt{2} & \rho c/\sqrt{2} \end{bmatrix}, \tag{A.38}$$

$$\boldsymbol{R}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -1/c^2 \\ 0 & k_2 & -k_1 & 0 \\ 0 & k_1/\sqrt{2} & k_2/\sqrt{2} & 1/(\sqrt{2}\rho c) \\ 0 & -k_1/\sqrt{2} & -k_2/\sqrt{2} & 1/(\sqrt{2}\rho c) \end{bmatrix}, \tag{A.39}$$

$$\boldsymbol{A}''_j = \begin{bmatrix} u_j & 0 & 0 & 0 \\ 0 & u_j & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 \\ 0 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & u_j + ck_j & 0 \\ 0 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & 0 & u_j - ck_j \end{bmatrix}. \tag{A.40}$$

A.3. One-dimensional case

$$\boldsymbol{U} = \rho \begin{Bmatrix} 1 \\ u \\ e \end{Bmatrix}, \tag{A.41}$$

$$\boldsymbol{F} = \begin{Bmatrix} u\rho \\ u\rho u + p \\ u(\rho e + p) \end{Bmatrix}, \tag{A.42}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma - 3)u^2 & -(\gamma - 3)u & \bar{\gamma} \\ (\bar{\gamma}u^2 - \gamma e)u & \gamma e - \frac{3}{2}\bar{\gamma}u^2 & \gamma u \end{bmatrix}, \tag{A.43}$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ u & \rho & 0 \\ \frac{1}{2}u^2 & \rho u & 1/\bar{\gamma} \end{bmatrix}, \tag{A.44}$$

$$Q^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -u/\rho & 1/\rho & 0 \\ \frac{1}{2}\bar{\gamma}u^2 & -\bar{\gamma}u & \bar{\gamma} \end{bmatrix}, \tag{A.45}$$

$$U' = \begin{Bmatrix} \rho \\ u \\ p \end{Bmatrix}, \tag{A.46}$$

$$A' = \begin{bmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \rho c^2 & u \end{bmatrix}, \tag{A.47}$$

$$\lambda_1 = u, \qquad \lambda_2 = \lambda_1 + c, \qquad \lambda_3 = \lambda_1 - c, \tag{A.48}$$

$$R = \begin{bmatrix} 1 & \rho/(\sqrt{2}c) & \rho/(\sqrt{2}c) \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & \rho c/\sqrt{2} & \rho c/\sqrt{2} \end{bmatrix}, \tag{A.49}$$

$$R^{-1} = \begin{bmatrix} 1 & 0 & -1/c^2 \\ 0 & 1/\sqrt{2} & 1/(\sqrt{2}\rho c) \\ 0 & -1/\sqrt{2} & 1/(\sqrt{2}\rho c) \end{bmatrix}, \tag{A.50}$$

$$A'' = \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}. \tag{A.51}$$

Acknowledgment

We would like to thank the Computational Fluid Dynamics Branch of the NASA Ames Research Center for supporting our research, and especially H. Lomax for guidance and encouragement. Thanks are also due the CLaSSiC project for partial support of T.E. Tezduyar while he was a post-doctoral student at Stanford University.

References

[1] N.N. Anucina, Some difference schemes for hyperbolic systems, in: N.N. Yanenko, ed., Difference Methods for Solutions of Problems of Mathematical Physics I (American Mathematical Society, Providence, RI, 1967) 1–13.

[2] A.J. Baker, Research on numerical algorithms for the three-dimensional Navier–Stokes equations, I. Accuracy, convergence and efficiency, Tech. Rep. AFFDL-TR-79-3141, Wright–Patterson Air Force Base, OH, 1979.

[3] A.J. Baker and M.O. Soliman, On the utility of finite element theory for computational fluid dynamics, Proc. 5th AIAA Computational Fluid Dynamics Conference, Palo Alto, CA, 1981.

[4] R.M. Beam, R.F. Warming and H.C. Yee, Stability analysis of numerical boundary conditions and implicit difference approximations for hyperbolic equations, Proc. Numerical Boundary Condition Procedures Symposium, NASA Ames Research Center, Moffet Field, CA, 1981.

[5] A. Brooks and T.J.R. Hughes, Streamline upwind/Petrov–Galerkin methods for advection dominated flow, Proc. Third International Conference on Finite Element Methods in Fluid Flow, Banff, Canada, 1980.

[6] A. Brooks and T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Meths. Appl. Mech. Engrg. 32 (1982) 199–259.

[7] P.G. Buning and J.L. Steger, Solution of the two-dimensional Euler equations with generalized coordinate transformation using flux vector splitting, AIAA-82-0971 AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, MO, 1982.

[8] D.A. Caughey and A. Jameson, Numerical calculation of transonic potential flow about wing-body combinations, AIAA J. 17 (1979) 175–181.

[9] I. Christie, D.F. Griffiths, A.R. Mitchell and J.M. Sanz-Serna, Product approximation for non-linear problems in the finite element method, IMA J. Numer. Anal. 1 (1981) 253–266.

[10] M. Cohen, Private communication, 1982.

[11] J. Donéa, Finite element analysis of transient dynamic fluid-structure interaction, in: J. Donéa, ed., Advanced Structural Dynamics (Applied Science Publishers, London, 1980) 255–290.

[12] J.K. Dukowicz and J.D. Ramshaw, Tensor viscosity method for convection in numerical fluid dynamics, J. Comput. Phys. 32 (1979) 71–79.

[13] A. Ecer and H.U. Akay, A finite element formulation of Euler equations for the solution of steady transonic flows, AIAA-82-0062, AIAA 20th Aerospace Sciences Meeting, Orlando, FL, 1982.

[14] B. Engquist and S. Osher, One-sided difference schemes and transonic flow, Proc. Nat. Acad. Sci. U.S.A. 77 (1980) 3071–3074.

[15] C.A.J. Fletcher, The group finite element formulation, Comput. Meths. Appl. Mech. Engrg. 37 (2) (1983) 225–244.

[16] I. Fried and D.S. Malkus, Finite element mass matrix lumping by numerical integration with no convergence rate loss, Internat. J. Solids and Structures 11 (1975) 461–465.

[17] P.M. Goorjian and R. Van Buskirk, Implicit calculations of transonic flows using monotone methods, AIAA-81-0331, AIAA 19th Aerospace Sciences Meeting, St. Louis, MO, 1981.

[18] G.L. Goudreau and J.O. Hallquist, Recent developments in large scale finite element Lagrangian hydrocode technology, Comput. Meths. Appl. Mech. Engrg. 33 (1982) 725–757.

[19] A. Harten, The artificial compression method for computation of shocks and contact discontinuities: I. Single conservation laws, Comm. Pure Appl. Math. XXX (1977) 611–638.

[20] A. Harten, The artificial compression method for computation of shocks and contact discontinuities: III. Self-adjusting hybrid schemes, Math. Comput. 32 (142) (1978) 363–389.

[21] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.

[22] A. Harten, J.M. Hyman and P.D. Lax, On finite difference approximations and entropy conditions for shocks, Comm. Pure Appl. Math. XXIX (1976) 297–322.

[23] T.J.R. Hughes, A study of the one-dimensional theory of arterial pulse propagation, Rep. No. UC SESM 74-13, Structural Engineering Laboratory, University of California, Berkeley, CA, 1974.

[24] T.J.R. Hughes, Implicit–explicit finite element techniques for symmetric and non-symmetric systems, in: E. Hinton, D.R.J. Owen and C. Taylor, eds., Recent Advances in Non-linear Computational Mechanics (Pineridge, Swansea, 1982) 255–267.

[25] T.J.R. Hughes, Analysis of transient algorithms with particular reference to stability behavior, in T. Belytschko and T.J.R. Hughes, eds., Computational Methods in Transient Analysis (North-Holland, Amsterdam, 1982) Chapter 2.

[26] T.J.R. Hughes and A. Brooks, A multidimensional upwind scheme with no crosswind diffusion, in: T.J.R. Hughes et al., eds., Finite Element Methods for Convection Dominated Flows, AMD Vol. 34 (ASME, New York, 1979).

[27] 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) 103–112.

[28] T.J.R. Hughes and A. Brooks, A theoretical framework for Petrov–Galerkin methods with discontinuous weighting functions: Application to the streamline upwind procedure, in: R.H. Gallagher, ed., Finite Elements in Fluids, Vol. 4 (Wiley, New York, 1982).

[29] T.J.R. Hughes, I. Levit and J. Winget, Unconditionally stable element-by-element implicit algorithm for heat conduction analysis, J. Engrg. Mech., ASCE 109 (2) (1983) 576–585.

[30] T.J.R. Hughes, I. Levit and J. Winget, An element-by-element solution algorithm for problems of structural and solid mechanics, Comput. Meths. Appl. Mech. Engrg. 36 (2) (1983) 241–254.

[31] T.J.R. Hughes and W.K. Liu, Implicit–explicit finite elements in transient analysis: Stability theory, J. Appl. Mech. 45 (1978) 371–374.

[32] T.J.R. Hughes and W.K. Liu, Implicit–explicit finite elements in transient analysis: Implementation and numerical examples, J. Appl. Mech. 45 (1978) 374–378.

[33] 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.

[34] T.J.R. Hughes, J. Winget, I. Levit and T.E. Tezduyar, New alternating direction procedures in finite element analysis based upon EBE approximate factorizations, in: S.N. Atluri and N. Perrone, eds., Computer Methods for Nonlinear Solids and Structural Mechanics, AMD Vol. 54 (ASME, New York, 1983) 75–109.

[35] C. Johnson, Finite element methods for convection–diffusion problems, Fifth International Symposium on Computing Methods in Engineering and Applied Sciences, INRIA, Versailles, 1981.

[36] D.W. Kelly, S. Nakazawa, O.C. Zienkiewicz and J.C. Heinrich, A note on upwinding and anisotropic balancing dissipation in finite element approximations to convective diffusion problems, Internat. J. Numer. Meths. Engrg. 15 (1980) 1705–1711.

[37] P.D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Comm. Pure Appl. Math. 7 (1954) 159–193.

[38] P.D. Lax, Hyperbolic systems of conservation laws II, Comm. Pure Appl. Math. 10 (1957) 537–566.

[39] P.D. Lax, Nonlinear hyperbolic systems of conservation laws, in: R.E. Langer, ed., Nonlinear Problems (University of Wisconsin Press, Madison, WI, 1963).

[40] H.W. Liepmann and A. Roshko, Elements of Gasdynamics (Wiley, New York, 1957).

[41] H. Lomax and E.D. Martin, Fast direct numerical solution of the nonhomogenous Cauchy–Riemann equations, J. Comput. Phys. 15 (1) (1974) 55–80.

[42] H. Lomax et al., Private communications, 1981–1982.

[43] G. Moretti, A physical approach to the numerical treatment of boundaries in gas dynamics, Proc. Numerical Boundary Condition Procedures Symposium, NASA Ames Research Center, Moffet Field, CA, 1981.

[44] K.W. Morton, Shock capturing, fitting and recovery, Department of Mathematics, University of Reading, 1982.

[45] S. Nakazawa, Finite element analysis applied to polymer processing, Ph.D. Thesis, University of Swansea, 1982.

[46] U. Nävert, A finite element method for convection–diffusion problems Ph.D. Thesis, Department of Computer Sciences, Chalmers University of Technology, Göteborg, Sweden, 1982.

[47] S. Osher, Approximation par éléments finis avec décentrage pour des lois de conservation hyperboliques non linéaires multi-dimensionelles, C.R. Acad. Sci. Paris, Sér. A290 (1980) 819–821.

[48] S. Osher, Approximation par elements finis avec decentrage de problemes de perturbations singulières quasi linéaries et multidimensionnels, C.R. Acad. Sci. Paris, Sér. I292 (1981) 99–101.

[49] S. Osher, Nonlinear singular perturbation problems and one sided difference schemes, SIAM J. Numer. Anal. 18 (1981) 129–144.

[50] S. Osher, Numerical solution of singular perturbation problems and hyperbolic systems of conservation laws, in: O. Axelsson et al. eds., North-Holland Math Studies 47 (North-Holland, Amsterdam, 1982).

[51] S. Osher, Shock modelling in aeronautics, in: K.W. Morton and M.J. Baines, eds., Numerical Methods for Fluid Dynamics (Academic Press, London, 1982).

[52] 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.

[53] M.D. Salas, Recent developments in transonic Euler flow over a circular cylinder, 10th IMACS World Congress on System Simulation and Scientific Computation, Montreal, Canada, 1982.

[54] L.W. Spradley, J.F. Stalnaker and A.W. Ratliff, Computation of three-dimensional viscous flows with the

Navier–Stokes equations, AIAA-80-1348, AIAA 13th Fluid and Plasma Dynamics Conference, Snowmass, CO, 1980.

[55] J.L. Steger, Implicit finite difference simulation of flow about arbitrary geometries with application to airfoils, AIAA Paper 77-665, 1977.

[56] J.L. Steger, A preliminary study of relaxation methods for the inviscid conservative gasdynamics equations using flux splitting, NASA Contractor Rept. 3415, 1981.

[57] J.L. Steger and R.F. Warming, Flux vector splitting of the inviscid gasdynamics equations with application to finite difference methods, J. Comput. Phys. 40 (2) (1981) 263–293.

[58] F. Thomasset, Implementation of Finite Element Methods for Navier–Stokes Equations (Springer, New York, 1981).

[59] E. Turkel, Symmetrization of the fluid dynamic matrices with applications, Math. Comput. 27 (124) (1973) 729–736.

[60] R.F. Warming, R.M. Beam and B.J. Hyett, Diagonalization and simultaneous symmetrization of the gas-dynamic matrices, Math. Comput. 29 (132) (1975) 1037–1045.

[61] R.F. Warming and R.M. Beam, On the construction and application of implicit factored schemes for conservation laws, SIAM-AMS Proc. 11 (1978).

[62] H.F. Weinberger, A First Course in Partial Differential Equations (Ginn Waltham, MA, 1965).

[63] M.L. Wilkins, Calculation of elastic-plastic flows, Lawrence Livermore Laboratory, Rept. UCRL-7322, Rev. 1, Livermore, CA, 1969.

[64] H.C. Yee, Numerical approximation of boundary conditions with applications to inviscid equations of gas dynamics, NASA TM 81265, 1981.

[65] O.C. Zienkiewicz, The Finite Element Method (McGraw–Hill, London, 1977).