Finite Element Formulations for Hyperbolic Systems with Particular Emphasis on the Compressible Euler Equations

Thesis by

Tayfun Ersin Tezduyar

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology
Pasadena, California

1982

(Submitted May 10, 1982)

ACKNOWLEDGEMENTS

I wish to thank my advisor Thomas J.R. Hughes, for his guidance and patience throughout my graduate studies. I am especially grateful for his assistance and encouragement during the preparation of this thesis.

This research was made possible by the support of the California Institute of Technology, the Stanford University and Computational Fluid Dynamics branch of the NASA Ames Research Center. I would like to thank H. Lomax for guidance and encouragement and J. Barton, T. Holst and J. Steger for valuable suggestions.

Abstract

A Petrov-Galerkin finite element formulation for first-order hyperbolic systems is developed generalizing the streamline approach which has been successfully applied previously to convection-diffusion and incompressible Navier-Stokes equations. The formulation is shown to possess desirable stability and accuracy properties.

The algorithm is applied to the Euler equations in conservation-law form and is shown to be effective in all cases studied, including ones with discontinuous solutions. Accurate and crisp representation of shock fronts in transonic problems is achieved.

TABLE OF CONTENTS

CHAPTER 1

Introduction

Analysis of inviscid, compressible fluid flows, especially ones with discontinuities, has been an interesting and challenging part of the research done in the field of computational fluid dynamics (see [L6]).

Numerous workers in this field have employed finite difference techniques. The following represents a brief sampling of some recent works. Ballhaus et al. [B4] used implicit approximate factorization schemes to solve the transonic small disturbance equation. Holst and Ballhaus [H3] applied approximate factorization schemes to the full potential equation in conservation form. Holst and Brown [H4] utilized solution adaptive grids for the full potential equation in conservation form. Warming and Beam [W3] used approximate factorization schemes to solve the Euler equations in conservation law form. Steger [S3], and Steger and Warming [S2] applied flux vector splitting ideas to the solution of the Euler equations in conservation form.

Finite difference schemes of the above type are mostly limited to problems with simple geometries. Finite element methods, on the other hand, can easily handle arbitrary geometries.

In the finite element method, the problem domain is discretized into sub-domains (elements), and, via a weighted residual formulation, the governing differential equation system is translated into a system of ordinary differential equations.

In a weighted residual formulation, selecting the weighting functions from the same class that the interpolation functions are selected from, leads to a (Bubnov) Galerkin formulation. When applied to differential equation systems with symmetric operators (e.g. diffusion equations, most structural and solid mechanics problems) Galerkin formulations produce solutions with a "best approximation" property. That is, the error is minimized with respect to a certain norm.

For systems with non-symmetric operators (e.g. first-order hyperbolic systems), however, the Galerkin formulation does not possess a best approximation property. This, in some cases, may result in solutions with spurious node-to-node oscillations. In fact, this problem is not limited to Galerkin finite element formulations. It also arises for finite difference schemes when non-symmetric operators are approximated centrally.

Instead of using weighting functions which lead to a Galerkin formulation, one can employ a Petrov-Galerkin formulation by modifying those weighting functions according to an optimal rule. The basic idea is to minimize the spurious oscillations without introducing excessive diffusion to the solution.

An optimal streamline upwind/Petrov-Galerkin formulation for convection dominated flows was recently developed by Hughes and Brooks (see [B7, B8, H12, H14, H15]) and was successfully applied to the solution of advection-diffusion and incompressible Navier-Stokes equations. In this work we present a Petrov-Galerkin algorithm which is a generalization of the streamline upwind/Petrov-Galerkin algorithm to hyperbolic systems. The weighting functions (which would normally lead to a Galerkin formulation) are perturbed by the product

of the coefficient matrix of the hyperbolic system, the gradient of the weighting function and a time parameter. The alternatives of transposing or not transposing the coefficient matrix in the weighting function, and the selection of the time parameter are among the subjects discussed here. The algorithm presented, under some very special conditions, reduces to the Lax-Wendroff scheme, which is known as a shock capturing algorithm. By incorporating the coefficient matrix of the hyperbolic system into the weighting function we automatically inject the eigenvalue/eigenvector information of the system into our finite element formulation.

In chapter 2 we briefly review the properties of one-dimensional hyperbolic systems and introduce the Petrov-Galerkin algorithms. The selection of the weighting functions is discussed in detail. We also investigate under what circumstances the weighted residual formulation of a system can be reduced to that of uncoupled single degree-of-freedom equations. The procedure of finite element discretization, and the transient algorithm used for solving the semi-discrete equation are described. Further, for a special case, we write the finite difference equations for the Petrov-Galerkin formulation.

In chapter 3 we perform a detailed stability and accuracy analysis of algorithms for the linear one-dimensional hyperbolic equation. Several algorithms of interest are studied and compared.

Chapter 4 reports numerical results in one space dimension. Several linear and nonlinear, steady and transient problems are solved using various techniques. Special emphasis is placed on problems with discontinuous solutions (shocks).

In chapter 5 we introduce the multi-dimensional versions of the Petrov-Galerkin algorithms.

Chapter 6 covers numerical applications in two space dimensions. A biconvex thin airfoil problem is solved for subsonic and transonic cases. Several algorithms are tested.

In chapter 7, we draw our conclusions and make suggestions for future research.

Appendix I reviews the properties of the compressible Euler equations.

In appendix II, a stability and accuracy analysis of algorithms for the one-dimensional, linear parabolic equation is performed. In appendix III a similar analysis for the one-dimensional linear second-order hyperbolic equation is performed. The methods used in appendices II and III are essentially the same as that used in chapter 3.

CHAPTER 2

One-dimensional Hyperbolic Systems

2.1 Initial/Boundary-value Problem

Let $\Omega =]0,L[$ denote the open interval of length L, and let $\overline{\Omega} = [0, L]$ denote its closure. The boundary of Ω is $\Gamma = \{0, L\}$, that is, the points 0 and L. Spatial and temporal coordinates are denoted by $x \in \overline{\Omega}$ and $t \in [0, T]$, respectively.

Consider the following system of m partial differential equations:

$$\underset{\sim}{U}_{,t} + \underset{\sim\sim}{A}\underset{\sim}{U}_{,x} + \underset{\sim}{G} = \underset{\sim}{0} \tag{2.1.1}$$

where

$$\underset{\sim}{U} = \underset{\sim}{U}(x, t) \tag{2.1.2}$$

$$\underset{\sim}{A} = \underset{\sim}{A}(\underset{\sim}{U}, x, t) \tag{2.1.3}$$

$$\underset{\sim}{G} = \underset{\sim}{G}(\underset{\sim}{U}, x, t) \tag{2.1.4}$$

and a comma denotes partial differentiation.

We are concerned with the case in which (2.1.1) is hyperbolic, that is when $\underset{\sim}{A}$ has real eigenvalues and there exists a transformation matrix $\underset{\sim}{S}$ such that

$$\underset{\sim}{S}^{-1}\,\underset{\sim}{A}\,\underset{\sim}{S} = \underset{\sim}{\Lambda} \tag{2.1.5}$$

where $\underset{\sim}{\Lambda}$ is the diagonal matrix of eigenvalues of $\underset{\sim}{A}$.

(2.1.1) is called a balance law if there exists a vector

$$\underset{\sim}{\mathscr{F}} = \underset{\sim}{\mathscr{F}}(\underset{\sim}{U}, x, t) \tag{2.1.6}$$

such that

$$\underset{\sim}{A} = \partial \underset{\sim}{\mathscr{F}} / \partial \underset{\sim}{U} \tag{2.1.7}$$

A balance law in which $\underset{\sim}{G} = \underset{\sim}{0}$ is said to be a conservation law.

In the linear case

$$\underset{\sim}{A} = \underset{\sim}{A}(x, t) \tag{2.1.8}$$

and

$$\underset{\sim}{G} = \underset{\sim}{B}(x, t)\underset{\sim}{U} + \underset{\sim}{\mathscr{f}}(x, t) \tag{2.1.9}$$

In the constant-coefficient case $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are independent of x and t.

Classical references for the study of hyperbolic systems are Courant - Hilbert [C3] and Courant - Friedrichs [C2].

Consideration of the eigenstructure of $\underset{\sim}{A}$ enables the specification of appropriate boundary conditions. For a general treatment of this topic see Yee [Y1]. For the present purposes, it suffices to assume that the boundary conditions take the abstract form

$$\underset{\sim}{\partial}\ \underset{\sim}{U} = \underset{\sim}{g}(t) \tag{2.1.10}$$

where $\underset{\sim}{\partial}$ is a boundary operator and $\underset{\sim}{g}$ is a prescribed function.

The initial/boundary-value problem for (2.1.1) consists of finding a function $\underset{\sim}{U}$ which satisfies (2.1.1), the boundary conditions (2.1.10), and the following initial condition:

$$\underset{\sim}{U}(x, 0) = \underset{\sim}{U}_0(x) \tag{2.1.11}$$

where $\underset{\sim}{U}_0$ is a given function of $x \in \Omega$.

2.2 Weighted Residual Formulation

Consider a discretization of Ω into element subdomains Ω^e, $e = 1, 2, \ldots, n_{el}$, where n_{el} is the number of elements. We assume

$$\overline{\Omega} = \bigcup_{e=1}^{n_{el}} \overline{\Omega}^e \tag{2.2.1}$$

$$\emptyset = \bigcap_{e=1}^{n_{el}} \Omega^e \tag{2.2.2}$$

All functions considered in the finite element formulation will be smooth on the element interiors (i.e. Ω^e's). Two classes of functions are important in the developments which follow. The classes are distinguished by their continuity properties across the element boundaries.

Functions of the first class are assumed to be continuous across element boundaries. These functions are denoted by $C^0 = C^0(\overline{\Omega})$ and may be recognized as containing the standard finite element interpolations.

Functions of the second class are allowed to be discontinuous across element boundaries and are denoted by $C^{-1} = C^{-1}(\overline{\Omega})$.

A weighted residual formulation of (2.1.1) is given by

$$0 = \int_\Omega \tilde{\underset{\sim}{W}} \cdot (\underset{\sim}{U}_{,t} + \underset{\sim\sim}{A}\underset{\sim}{U}_{,x} + \underset{\sim}{G}) d\Omega \tag{2.2.3}$$

where $\tilde{\underset{\sim}{W}}$ is a weighting function and $\cdot$ denotes the dot product. In all cases we assume $\underset{\sim}{U}$ is approximated by standard, C^0, finite element interpolations. The weighting functions may be selected from a different set of functions than the trial solutions. Thus (2.2.3) gives rise to a Petrov-Galerkin formulation (see e.g. [B1, B7, B8, C1, D1, H1, H8, H12, H14 H15, M2, R1, W1]).

An important class of Petrov-Galerkin methods, which is emphasized in the sequel, is defined by the following expression for $\tilde{\underset{\sim}{W}}$:

$$\tilde{\underset{\sim}{W}} = \underset{\sim}{W} + \underset{\sim}{T}\,\underset{\sim}{W}_{,x} \tag{2.2.4}$$

where $\underset{\sim}{W}$ is a member of the same class of functions as the trial solutions and $\underset{\sim}{T}$ is either $\tau\underset{\sim}{A}$ or $\tau\underset{\sim}{A}^T$ where τ is a parameter which is chosen to optimize accuracy according to some criterion. This class of methods represents a generalization to hyperbolic systems of the streamline-upwind/ Petrov-Galerkin formulation which has been successfully applied heretofore to the advection-diffion and incompressible Navier-Stokes equations [B7, B8, H12, H14, H15].

Both choices of $\underset{\sim}{T}$ have interesting consequences. For example, assume the linear, constant-coefficient case in which $\underset{\sim}{G} = \underset{\sim}{0}$. Choose $\underset{\sim}{T} = \tau\underset{\sim}{A}^T$. Then (2.2.3) reduces to the canonical form

$$0 = \int_\Omega (\bar{\underset{\sim}{W}} + \tau\underset{\sim\sim}{\Lambda}\bar{\underset{\sim}{W}}_{,x}) \cdot (\bar{\underset{\sim}{U}}_{,t} + \underset{\sim\sim}{\Lambda}\bar{\underset{\sim}{U}}_{,x})\,d\Omega \tag{2.2.5}$$

where $\bar{\underset{\sim}{W}} = \underset{\sim}{S}^T\underset{\sim}{W}$ and $\bar{\underset{\sim}{U}} = \underset{\sim}{S}^{-1}\underset{\sim}{U}$. Thus (2.2.5) is equivalent to a system of uncoupled scalar equations. Scalar equations of this form are extensively analyzed in Chapter 3.

Furthermore, the choice $\underset{\sim}{T} = \tau\underset{\sim}{A}^T$ leads to difference equations which, under special circumstances, have essential features in common with the well-known Lax-Wendroff method [R2].

Under the circumstances which led to (2.2.5), choosing $\underset{\sim}{T} = \tau\underset{\sim}{A}$ does not result in the canonical form (2.2.5) unless the weighted residual formulation is generalized to

$$0 = \int_{\Omega} (\underset{\sim}{W} + \tau \underset{\sim\sim}{A} \underset{\sim}{W}_{,x}) \cdot \underset{\approx}{Q} (\underset{\sim}{U}_{,t} + \underset{\sim\sim}{A} \underset{\sim}{U}_{,x}) d\Omega \tag{2.2.6}$$

where

$$\underset{\approx}{Q} = (\underset{\sim}{S}\ \underset{\sim}{S}^{T})^{-1} \tag{2.2.7}$$

$$\underset{\sim}{W} = \underset{\sim}{S}\ \overline{\underset{\sim}{W}} \tag{2.2.8}$$

$$\underset{\sim}{U} = \underset{\sim}{S}\ \overline{\underset{\sim}{U}} \tag{2.2.9}$$

Computational experiences with generalizations of formulations of this type proved cumbersome and unreliable in the nonlinear regime when compared with (2.2.3) and (2.2.4), and thus were abandoned.

Despite the fact that $\underset{\sim}{T} = \tau \underset{\sim}{A}$ does not canonically reduce (2.2.3), it leads to another optimality property which will be described subsequently (see §2.5).

If τ is taken to be zero then we have the usual Galerkin method which possesses central-difference like character.

2.3 Semi-discrete Equations

Spatial discretization of the weighted residual equation (2.2.3) via finite elements leads to the following semi-discrete system of ordinary differential equations:

$$\underset{\sim}{M}\,\dot{\underset{\sim}{v}} + \underset{\sim}{C}\,\underset{\sim}{v} = \underset{\sim}{F} \tag{2.3.1}$$

where $\underset{\sim}{M} = \underset{\sim}{M}(\underset{\sim}{v}, t)$ is the generalized "mass" matrix, $\underset{\sim}{C} = \underset{\sim}{C}(\underset{\sim}{v}, t)$ is the generalized convection matrix, $\underset{\sim}{F} = \underset{\sim}{F}(\underset{\sim}{v}, t)$ is the force vector, $\underset{\sim}{v}$ is the vector of (unknown) nodal values of $\underset{\sim}{U}$, and a superposed dot denotes time differentiation. The initial-value problem for (2.3.1) consists of finding a function $\underset{\sim}{v} = \underset{\sim}{v}(t)$ satisfying (2.3.1) and the initial condition

$$\underset{\sim}{v}(0) = \underset{\sim}{v}_0 \tag{2.3.2}$$

where $\underset{\sim}{v}_0$ is determined from (2.1.11).

The arrays in (2.3.1) are assembled from element contributions:

$$\underset{\sim}{M} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\underset{\sim}{m}^e) \tag{2.3.3}$$

$$\underset{\sim}{m}^e = [\underset{\sim ab}{m}^e] \tag{2.3.4}$$

$$\underset{\sim ab}{m}^e = \int_{\Omega^e} (N_a \underset{\sim}{I} + N_{a,x} \underset{\sim}{T}^T) N_b \, d\Omega \tag{2.3.5}$$

$$\underset{\sim}{C} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\underset{\sim}{c}^e) \tag{2.3.6}$$

$$\underset{\sim}{c}^e = [\underset{\sim ab}{c}^e] \tag{2.3.7}$$

$$\underset{\sim}{c}^{e}_{ab} = \int_{\Omega^e} (N_a \underset{\sim}{I} + N_{a,x} \underset{\sim}{T}^T) \underset{\sim}{A} N_{b,x} \, d\Omega \tag{2.3.8}$$

$$\underset{\sim}{F} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\underset{\sim}{f}^e) \tag{2.3.9}$$

$$\underset{\sim}{f}^e = \{\underset{\sim a}{f}^e\} \tag{2.3.10}$$

$$\underset{\sim a}{f}^e = \int_{\Omega^e} N_a \underset{\sim}{G} \, d\Omega - \sum_{b=1}^{n_{en}} (\underset{\sim ab}{m}^e \dot{\underset{\sim b}{g}}^e + \underset{\sim ab}{c}^e \underset{\sim b}{g}^e) \tag{2.3.11}$$

where $\mathbf{A}$ represents the finite element assembly operator; a and b are (local) element node numbers; $1 \leq a$, $b \leq n_{en}$ where n_{en} is the number of nodes for the element under consideration; N_a is the element shape function associated with node a; $\underset{\sim}{I}$ is the $m \times m$ identity matrix; and $\underset{\sim b}{g}^e$ is a vector which contains the boundary condition data emanating from (2.1.10). The dimensions of the nodal arrays $\underset{\sim ab}{m}^e$ and $\underset{\sim ab}{c}^e$ are $m \times m$, and the dimension of $\underset{\sim a}{f}^e$ and $\underset{\sim b}{g}^e$ are $m \times 1$.

The reader is reminded that

$$\underset{\sim}{U}(\underset{\sim}{x}, t) = \sum_{A=1}^{n_{np}} N_A(x) \underset{\sim}{v}_A(t) \tag{2.3.12}$$

where A is the global nodal index, n_{np} is the total number of nodes and $\underset{\sim}{v}_A$ refers to the components of $\underset{\sim}{v}$ associated with node A.

2.4 Transient Algorithms

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

$$\underset{\sim}{M}_{n+\gamma}\ \underset{\sim}{a}_{n+\gamma} + \underset{\sim}{C}_{n+\gamma}\ \underset{\sim}{v}_{n+\gamma} = \underset{\sim}{F}_{n+\gamma} \tag{2.4.1}$$

$$\underset{\sim}{v}_{n+1} = \underset{\sim}{v}_{n} + \Delta t\ \underset{\sim}{a}_{n+\alpha} \tag{2.4.2}$$

where

$$\underset{\sim}{M}_{n+\gamma} = \underset{\sim}{M}(\underset{\sim}{v}_{n+\gamma}\ ,\ t_{n+\gamma}) \tag{2.4.3}$$

$$\underset{\sim}{C}_{n+\gamma} = \underset{\sim}{C}(\underset{\sim}{v}_{n+\gamma}\ ,\ t_{n+\gamma}) \tag{2.4.4}$$

$$\underset{\sim}{F}_{n+\gamma} = \underset{\sim}{F}(\underset{\sim}{v}_{n+\gamma}\ ,\ t_{n+\gamma}) \tag{2.4.5}$$

$$a_{n+\gamma} = (1-\gamma)\underset{\sim}{a}_{n} + \gamma \underset{\sim}{a}_{n+1} \tag{2.4.6}$$

$$\underset{\sim}{v}_{n+\gamma} = (1-\gamma)\underset{\sim}{v}_{n} + \gamma \underset{\sim}{v}_{n+1} \tag{2.4.7}$$

$$\underset{\sim}{a}_{n+\alpha} = (1-\alpha)\underset{\sim}{a}_{n} + \alpha \underset{\sim}{a}_{n+1} \tag{2.4.8}$$

$$t_{n+\gamma} = (1-\gamma)t_{n} + \gamma t_{n+1} \tag{2.4.9}$$

In the above, Δt is the time step, n is the step number, and α and γ are parameters which determine stability and accuracy properties.

The starting value, $\underset{\sim}{a}_{0}$, may be determined from

$$\underset{\sim}{M}_{0}\ \underset{\sim}{a}_{0} = \underset{\sim}{F}_{0} - \underset{\sim}{C}_{0}\ \underset{\sim}{v}_{0} \tag{2.4.10}$$

where

$$\underset{\sim}{M}_0 = \underset{\sim}{M}(\underset{\sim}{v}_0, 0) \tag{2.4.11}$$

$$\underset{\sim}{C}_0 = \underset{\sim}{C}(\underset{\sim}{v}_0, 0) \tag{2.4.12}$$

$$\underset{\sim}{F}_0 = \underset{\sim}{F}(\underset{\sim}{v}_0, 0) \tag{2.4.13}$$

If $\gamma = 1$ the above algorithm reduces to the generalized trapezoidal method, whereas if $\gamma = \alpha$, it reduces to the generalized midpoint method. These methods have been contrasted in [H7, H16].

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

1. $i = 0$ (i is the iteration counter) (2.4.14)

2. $\underset{\sim}{v}_{n+1}^{(0)} = \underset{\sim}{v}_n + \Delta t(1 - \alpha)\underset{\sim}{a}_n$ (2.4.15)

3. $\underset{\sim}{a}_{n+1}^{(0)} = \underset{\sim}{0}$ (2.4.16)

(predictor phase)

4. $\underset{\sim}{R} = \underset{\sim}{F}_{n+\gamma}^{(i)} - \underset{\sim}{M}_{n+\gamma}^{(i)} \underset{\sim}{a}_{n+\gamma}^{(i)} - \underset{\sim}{C}_{n+\gamma}^{(i)} \underset{\sim}{v}_{n+\gamma}^{(i)}$ (residual force) (2.4.17)

5. $\underset{\sim}{M}^* \Delta\underset{\sim}{a} = \underset{\sim}{R}$ ($\underset{\sim}{M}^*$ is the "effective mass") (2.4.18)

6. $\underset{\sim}{a}_{n+1}^{(i+1)} = \underset{\sim}{a}_{n+1}^{(i)} + \Delta\underset{\sim}{a}$ (2.4.19)

7. $\underset{\sim}{v}_{n+1}^{(i+1)} = \underset{\sim}{v}_{n+1}^{(i)} + \alpha\,\Delta t\,\Delta\underset{\sim}{a}$ (2.4.20)

(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 $\underset{\sim}{R}$ satisfies a convergence condition. When the iterative phase is completed the solution at step $n + 1$ is defined by the last iterates (i.e. $\underset{\sim}{v}_{n+1} = \underset{\sim}{v}_{n+1}^{(i+1)}$ and $\underset{\sim}{a}_{n+1} = \underset{\sim}{a}_{n+1}^{(i+1)}$). 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

$$\underset{\sim}{M}^{*} = \underset{\sim}{M}_{n+\gamma}^{(i)} + \alpha\Delta t\, \underset{\sim}{C}_{n+\gamma}^{(i)} + \alpha\Delta t\, \underset{\sim}{H}_{n+\gamma}^{(i)} \tag{2.4.21}$$

where

$$\underset{\sim}{M}_{n+\gamma}^{(i)} = \underset{\sim}{M}(\underset{\sim}{v}_{n+\gamma}^{(i)}, t_{n+\gamma}) \tag{2.4.22}$$

$$\underset{\sim}{C}_{n+\gamma}^{(i)} = \underset{\sim}{C}(\underset{\sim}{v}_{n+\gamma}^{(i)}, t_{n+\gamma}) \tag{2.4.23}$$

$$\underset{\sim}{H}_{n+\gamma}^{(i)} = \underset{\sim}{H}(\underset{\sim}{v}_{n+\gamma}^{(i)}, t_{n+\gamma}) \tag{2.4.24}$$

$$\underset{\sim}{H} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} (\underset{\sim}{h}^{e}) \tag{2.4.25}$$

$$\underset{\sim}{h}^{e} = [\underset{\sim}{h}_{ab}^{e}] \tag{2.4.26}$$

$$\underset{\sim}{h}_{ab}^{e} = \int_{\Omega^e} (N_a \underset{\sim}{I} + N_{a,x} \underset{\sim}{T}^{T}) \frac{\partial \underset{\sim}{G}}{\partial \underset{\sim}{U}} N_b \, d\Omega \tag{2.4.27}$$

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

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

$$\underset{\sim}{M}^* = \underset{\sim}{M}_{diag} \tag{2.4.28}$$

There are several schemes for obtaining suitable $\underset{\sim}{M}_{diag}$. In the present work we assume that the diagonal element array is defined by [Z1]

$$\underset{\sim}{m}^e_{ab} = \begin{cases} m^e_a \, \underset{\sim}{I} & \text{if} \quad a = b \\ \underset{\sim}{0} & \text{if} \quad a \neq b \end{cases} \tag{2.4.29}$$

where

$$m^e_a = \beta \int_{\Omega^e} N_a^2 \, d\Omega \tag{2.4.30}$$

$$\beta = \int_{\Omega^e} d\Omega \Big/ \left(\sum_{a=1}^{n_{en}} \int_{\Omega^e} N_a^2 \, d\Omega \right) \tag{2.4.31}$$

In the present work we confine our attention to the implicit and explicit schemes defined above. Stability and accuracy analyses are presented in Chapter 3.

However, there are other possibilities. Implicit-explicit finite element mesh partitions [H9, H10, H11, H13] may prove useful, for example. Additionally, to obtain the stability properties of implicit methods, while eliminating the equation-solving burden imposed by (2.4.21), approximate factorization schemes may be employed. We are presently experimenting with element-by-element factorizations which are very convenient from an implementational standpoint [H17].

2.5 Selection of τ

Two expressions for τ have been employed in the numerical calculations. One is based upon spatial discretization and the other upon temporal discretization. They are given as follows:

spatial criterion

In this case we assume

$$\tau = F \alpha h/\rho \tag{2.5.1}$$

where F is a non-dimensional parameter, h is the element length, ρ is the spectral radius of $\underset{\sim}{A}$, that is

$$\rho = \max_{1 \leq i \leq m} |\lambda_i(\underset{\sim}{A})| \tag{2.5.2}$$

and the $\lambda_i(\underset{\sim}{A})$'s are the eigenvalues of $\underset{\sim}{A}$. Note that (2.5.1) is a local specification of τ in that it depends upon the element lengths and eigenvalues of $\underset{\sim}{A}$ which vary from point to point. Rationale for this form of τ is provided by the following examples:

Examples

1. Consider the scalar model equation

$$U_{,t} + \lambda U_{,x} = 0 \tag{2.5.3}$$

where λ is assumed constant. Raymond and Garder [R1] have shown that if

$$F \alpha = 1/\sqrt{15} \tag{2.5.4}$$

then the semi-discrete equations achieve fourth-order phase accuracy.

2. Consider the steady analog of (2.5.3) regularized by a diffusion term,

$$\lambda U_{,x} = \varepsilon U_{,xx} \tag{2.5.5}$$

As $\varepsilon \to 0$, the choice

$$F\,\alpha = {}^{1}/_{2} \tag{2.5.6}$$

leads to nodally exact solutions. The general case for the advection-diffusion equation is described in Hughes-Brooks [B7, B8, H12, H14, H15].

Remarks

1. The preceding optimality conditions, (2.5.4) and (2.5.6), need to be altered for higher-order elements. For example, in the case of three-node quadratic elements (2.5.6) should be changed to $F\,\alpha = {}^{1}/_{4}$ [N1] Throughout this work only low-order elements are employed.

2. A weighted residual formulation of a linear, constant-coefficient, hyperbolic system with $\underset{\sim}{G} = \underset{\sim}{0}$ can be given in which each uncoupled scalar equation is treated optimally, viz.

$$0 = \int_{\Omega} (\overline{\underset{\sim}{W}} + F\,\alpha\, h\, \operatorname{sgn} \underset{\sim}{\Lambda}\, \overline{\underset{\sim}{W}}_{,x}) \cdot (\overline{\underset{\sim}{U}}_{,t} + \underset{\sim}{\Lambda}\, \overline{\underset{\sim}{U}}_{,x}) d\Omega \tag{2.5.7}$$

where $\overline{\underset{\sim}{W}} = \underset{\sim}{S}^{T}\, \underset{\sim}{W}$ and $\overline{\underset{\sim}{U}} = \underset{\sim}{S}^{-1}\, \underset{\sim}{U}$; cf. (2.2.5). Unfortunately, there does not appear to be a nonlinear or multidimensional generalization of (2.5.7) and consequently, we have not explored the subject further.

temporal criterion

In this case we assume

$$\tau = F \alpha \Delta t \tag{2.5.8}$$

Note that (2.5.8) is a global specification in that Δt is the same for all elements. Rationale for this choice is provided by the following examples.

Examples

1. Assume $\underset{\sim}{H} = \underset{\sim}{0}$. If $\underset{\sim}{T} = \tau \underset{\sim}{A}$, and $F = 1$, then (2.5.8) leads to symmetry of the implicit operator $\underset{\sim}{M}^*$ (see Eq. 2.4.21). This can be seen from the definitions of the element contributions to $\underset{\sim}{M}^*$:

$$\underset{\sim}{m}^e_{ab} + \alpha\Delta t \, \underset{\sim}{c}^e_{ab} = \int_{\Omega^e} N_a \, N_b \, d\Omega \, \underset{\sim}{I} + \alpha\Delta t \int_{\Omega^e} (N_{a,x} \, N_b \, \underset{\sim}{A}^T + N_a \, N_{b,x} \, \underset{\sim}{A}) d\Omega$$

$$+ (\alpha\Delta t)^2 \int_{\Omega^e} N_{a,x} \, N_{b,x} \, \underset{\sim}{A}^T \underset{\sim}{A} \, d\Omega = (\underset{\sim}{m}^e_{ba} + \alpha\Delta t \, \underset{\sim}{c}^e_{ba})^T \tag{2.5.9}$$

The obvious advantage in this case is the decreased storage and factorzation costs. Symmetric element arrays are also advantageous in implicit-explicit finite element mesh partitions [H13]. This choice also leads to an optimality condition in that for a specified residual, $\underset{\sim}{R}$, the increment $\Delta \underset{\sim}{a}$ is optimal with respect to the norm defined by $\underset{\sim}{M}^*$. Another way of putting this is to say that the increment of $\underset{\sim}{U}$ is optimized with respect to the symmetric bilinear form which generates $\underset{\sim}{M}^*$. This concept of optimality is related to the following optimal steady formulation

$$\alpha(\underset{\sim}{W},\ \underset{\sim}{U}) = -\int_{\Omega} (\underset{\sim}{W} + \tau\ \underset{\sim}{A}\ \underset{\sim}{W}_{,x}) \cdot \underset{\sim}{G}\ d\Omega \tag{2.5.10}$$

where $\alpha(\cdot\ ,\ \cdot)$ is a symmetric, bilinear form defined by

$$\alpha(\underset{\sim}{W},\ \underset{\sim}{U}) = \int_{\Omega} \tau^{-1} (\underset{\sim}{W} + \tau\ \underset{\sim}{A}\ \underset{\sim}{W}_{,x}) \cdot (\underset{\sim}{U} + \tau\ \underset{\sim}{A}\ \underset{\sim}{U}_{,x}) d\Omega \tag{2.5.11}$$

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

$$\underset{\sim}{A} = \underset{\sim}{A}(x) \tag{2.5.12}$$

$$\underset{\sim}{G} = \underset{\sim}{G}(x) \tag{2.5.13}$$

2. The choices $\underset{\sim}{T} = \tau\ \underset{\sim}{A}^{T}$, $F = 1$, $\alpha = {}^{1}/_{2}$, and $\underset{\sim}{M}^{*} = \underset{\sim}{M}_{diag}$, leads to an explicit Lax-Wendroff type method. We shall explore this point further subsequently.

Remark

The factor, F, in (2.5.1) and (2.5.8) has been included to account for nonlinear effects. It has been our experience that a value of F greater than one needs to be employed to adequately handle shock-wave phenomena.

2.6 Finite Difference Equations

The finite difference equations for the preceding algorithms are needed subsequently 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; and (ii) h, $\underset{\sim}{A}$ and $\underset{\sim}{H}$ are constant; and (iii) $\underset{\sim}{G}$ varies linearly over each element. Furthermore, for notational clarity we have dropped the superscript i and subscript $n+\gamma$. The equations are as follows:

implicit case

$$\left(\frac{h}{2}(\underset{\sim}{I} + \alpha\Delta t\, \underset{\sim}{H})D_r + (-\underset{\sim}{T}^T + \alpha\Delta t\, \underset{\sim}{A} - \alpha\Delta t\, \underset{\sim}{T}^T \underset{\sim}{H})D_1 + \alpha\Delta t \frac{2}{h}\, \underset{\sim}{T}^T \underset{\sim}{A}\, D_2\right)\Delta\underset{\sim}{a}(j) = \left(-\frac{h}{2}\underset{\sim}{I}\, D_r + \underset{\sim}{T}^T D_1\right)\underset{\sim}{a}(j) + \left(-\underset{\sim}{A}\, D_1 - \frac{2}{h}\, \underset{\sim}{T}^T \underset{\sim}{A}\, D_2\right)\underset{\sim}{v}(j) + \left(-\frac{h}{2}\underset{\sim}{I}\, D_r + \underset{\sim}{T}^T D_1\right)\underset{\sim}{G}(j) \qquad (2.6.1)$$

where

$$D_r\, \underset{\sim}{v}(j) = 2(r\, \underset{\sim}{v}_{(j-1)} + (1-2r)\underset{\sim}{v}(j) + r\, \underset{\sim}{v}_{(j+1)}) \qquad (2.6.2)$$

$$D_1\, \underset{\sim}{v}(j) = \frac{1}{2}(-\underset{\sim}{v}_{(j-1)} + \underset{\sim}{v}_{(j+1)}) \qquad (2.6.3)$$

$$D_2\, \underset{\sim}{v}(j) = -\frac{1}{2}(\underset{\sim}{v}_{(j-1)} - 2\,\underset{\sim}{v}(j) + \underset{\sim}{v}_{(j+1)}) \qquad (2.6.4)$$

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

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

explicit case

In the explicit case, the right-hand side of (2.6.1) is the same, but the left-hand side simplifies to

$$h\,\Delta\underset{\sim}{a}(j) \tag{2.6.5}$$

Remarks

1. It is interesting to observe that even though upwind influence has been introduced via the weighting function defined by (2.2.4), the resulting difference equations are centered about node j.

2. Assume $\underset{\sim}{G} = \underset{\sim}{0}$, $\alpha = 1$, $T = \tau\underset{\sim}{A}^T$, $\tau = \Delta t/2$ and the explicit one-pass (i.e. one-iteration) case, then from (2.4.15), (2.4.16), (2.4.20) and (2.6.1) we get

$$\underset{\sim}{v}_{n+1}(j) = \left(\underset{\sim}{I} - \frac{\Delta t}{h}\,\underset{\sim}{A}\,D_1 - \left(\frac{\Delta t}{h}\,\underset{\sim}{A}\right)^2 D_2\right)\underset{\sim}{v}_n(j) \tag{2.6.5}$$

Eq. (2.6.5) defines the Lax-Wendroff method.

CHAPTER 3

Stability and Accuracy Analysis of Algorithms for the One-dimensional Linear Hyperbolic Equation

3.1 Development of the Tools for the Analysis

3.1.1 Introduction

Model Problem: Convection Equation

One-dimensional convection of a function $U(x,t)$ is governed by the following hyperbolic equation:

$$U_{,t} + \lambda U_{,x} = 0 \qquad (3.1.1)$$

where λ is the convection velocity. An initial condition of the following form is assumed:

$$U(x,0) = e^{ikx} \qquad (3.1.2)$$

where $i = (-1)^{\frac{1}{2}}$ and k is the wave number. We note that the function e^{ikx} is an element of the set of Fourier functions. This is a complete set and any piecewise regular function in the range of $(0,2\pi)$ can be represented as an expansion in this set.

Exact Solution

The exact solution of (3.1.1) and (3.1.2), for constant λ, is straightforward. Assuming a solution of the form:

$$U(x,t) = X(x)T(t) \quad , \qquad (3.1.3)$$

leads to:

$$X(x) = e^{ikx} \tag{3.1.4}$$

$$T(t) = e^{\nu t} \tag{3.1.5}$$

$$\nu = - i \lambda k \tag{3.1.6}$$

We define the damping coefficient, ξ, and the frequency, ω, as the components of the complex parameter ν:

$$-(\xi,\omega) = \nu \tag{3.1.7}$$

Thus:

$$\xi = 0 \tag{3.1.8}$$

$$\omega = \lambda k \tag{3.1.9}$$

3.1.2 Finite Element Solution

Semi-Discrete Equation

The Petrov-Galerkin formulations described in chapter 2 leads to the following semi-discrete equation:

$$\underset{\sim}{M}\,\dot{\underset{\sim}{v}} + \underset{\sim}{C}\,\underset{\sim}{v} = \underset{\sim}{0} \tag{3.1.10}$$

where

$$\underset{\sim}{v} = \{v_j\} \tag{3.1.11}$$

$$v_j = U(x_j) \tag{3.1.12}$$

Here j stands for an interior node, j, and x_j is the coordinate of that node. The matrices $\underset{\sim}{M}$ and $\underset{\sim}{C}$ were previously defined by (2.3.3) and (2.3.6).

For the purpose of analysis, we assume constant λ and constant mesh spacing h. Further, we assume that the finite element solution, also has a separable form:

$$\underset{\sim}{v} = \underset{\sim}{X}\, T \tag{3.1.13}$$

where

$$T = T(t) \tag{3.1.14}$$

$$\underset{\sim}{X} = \{X_j\} \tag{3.1.15}$$

$$X_j = X(x_j) \tag{3.1.16}$$

Consequently, we get the following semi-discrete form:

$$\underset{\sim}{M}\,\underset{\sim}{X}\,\dot{T} + \underset{\sim}{C}\,\underset{\sim}{X}\,T = \underset{\sim}{0} \tag{3.1.17}$$

The spatial component of the numerical solution is determined by the nodal interpolation of the imposed initial condition (3.1.2):

$$X_j = e^{ikx_j} = e^{iqj} \tag{3.1.18}$$

where the dimensionless wave number,

$$q = k\,h\ , \tag{3.1.19}$$

is a measure of the spatial refinement of the numerical method.
The j^{th} equation of the system of equations of (3.1.17) is

$$\left\{\left(\frac{h}{2}\,\underset{\sim}{D}_r - \tau\,\lambda\,\underset{\sim}{D}_1\right)\dot{T} + \left(\lambda\,\underset{\sim}{D}_1 + \tau\lambda^2\,\frac{2}{h}\,\underset{\sim}{D}_2\right)T\right\}\begin{Bmatrix} X_{j-1} \\ X_j \\ X_{j+1} \end{Bmatrix} = 0 \tag{3.1.20}$$

where τ is the parameter that appeared in section 2.2, and $\underset{\sim}{D}_r$, $\underset{\sim}{D}_1$, $\underset{\sim}{D}_2$ are the stencils for node j corresponding to the assembly of the element level matrices:

$$\int_{-1}^{+1} N_a N_b \, d\xi \quad \longleftrightarrow \quad \underset{\sim}{D}_r \tag{3.1.21}$$

$$\int_{-1}^{+1} N_a N_{b,\xi} \, d\xi \quad \longleftrightarrow \quad \underset{\sim}{D}_1 \tag{3.1.22}$$

$$\int_{-1}^{+1} N_{a,\xi} N_{b,\xi} \, d\xi \quad \longleftrightarrow \quad \underset{\sim}{D}_2 \tag{3.1.23}$$

These stencils are directly related to the difference operators given by (2.6.2)-(2.6.4). Depending on the numerical integration technique used, they can assume several forms. For example,

$$\underset{\sim}{D}_r = 2[1/6,\ 4/6,\ 1/6] \quad \text{(exact integration)} \tag{3.1.24}$$

$$\underset{\sim}{D}_1 = [-1/2,\ 0,\ 1/2] \quad \text{(exact integration)} \tag{3.1.25}$$

$$\underset{\sim}{D}_2 = [-1/2,\ 1,\ -1/2] \tag{3.1.26}$$

Further, we define the following array:

$$\underset{\sim}{E} = [e^{-iq},\ 1,\ e^{+iq}] \tag{3.1.27}$$

Then

$$[X_{j-1},\ X_j,\ X_{j+1}] = [e^{iq(j-1)},\ e^{iqj},\ e^{iq(j+1)}]$$

$$= e^{iqj} \underset{\sim}{E} \tag{3.1.28}$$

Substituting (3.1.28) into (3.1.20) leads to

$$(\frac{h}{2} \underset{\sim}{D}_r - \tau \lambda \underset{\sim}{D}_1)\underset{\sim}{E} \dot{T} + (\lambda \underset{\sim}{D}_1 + \tau \lambda^2 \frac{2}{h} \underset{\sim}{D}_2)\underset{\sim}{E}\, T = 0 \qquad (3.1.29)$$

Now define the complex scalars M and C corresponding to $\underset{\sim}{M}$ and $\underset{\sim}{C}$ respectively:

$$M = \frac{1}{2}(\underset{\sim}{D}_r - \frac{2\tau\lambda}{h} \underset{\sim}{D}_1)\underset{\sim}{E} \qquad (3.1.30)$$

$$\Delta t C = \frac{\lambda \Delta t}{h} (\underset{\sim}{D}_1 + \frac{2\tau\lambda}{h} \underset{\sim}{D}_2)\underset{\sim}{E} \qquad (3.1.31)$$

where Δt is the time step of the time integration algorithm. We define the following non-dimensional parameters.

$$C_{\Delta t} = \frac{\lambda \Delta t}{h} \qquad (3.1.32)$$

$$C_{2\tau} = \frac{2\tau\lambda}{h} \qquad (3.1.33)$$

$C_{\Delta t}$ is called the Courant number. Considering that τ has units of time, we can view $C_{2\tau}$ as an algorithmic "Courant number" based on 2τ. If we set $C_{2\tau} = 0$, we obtain the usual Galerkin formulation.

With the definitions of (3.1.30) and (3.1.31), (3.1.29) reduces to the following ordinary differential equation:

$$M \dot{T} + C T = 0 \qquad (3.1.34)$$

Transient Algorithm

Transient algorithms were described in section 2.4. We adopt the representation:

$$\underset{\sim}{Y} = [T, \dot{T}] \qquad (3.1.35)$$

and let $\underset{\sim}{Y}_n$ denote the approximation of $\underset{\sim}{Y}$ at the n^{th}-time step.

Given $\underset{\sim}{Y}_n$, we go through a predictor phase and an iterative phase to calculate $\underset{\sim}{Y}_{n+1}$.

In the predictor phase we calculate $\underset{\sim}{Y}^{(0)}_{n+1}$ by the following operation:

$$\underset{\sim}{Y}^{(0)}_{n+1} = \underset{\sim}{P}\, \underset{\sim}{Y}_n \tag{3.1.36}$$

where $\underset{\sim}{P}$ is the predictor matrix defined by

$$\underset{\sim}{P} = \begin{bmatrix} 1 & (1-\alpha)\Delta t \\ 0 & 0 \end{bmatrix} \tag{3.1.37}$$

The iterative phase starts with the zeroth-iteration value, $\underset{\sim}{Y}^{(0)}_{n+1}$, and continues according to the recurrence rule below:

Given $\underset{\sim}{Y}^{(i)}_{n+1}$ solve the following system for $\underset{\sim}{Y}^{(i+1)}_{n+1}$:

$$M^L\, \Delta\dot{T}^{(i)}_{n+1} + C^L\, \Delta T^{(i)}_{n+1} = -\left(M^R\, \dot{T}^{(i)}_{n+1} + C^R\, T^{(i)}_{n+1}\right) \tag{3.1.38}$$

$$\Delta T^{(i)}_{n+1} = \alpha\Delta t\, \Delta\dot{T}^{(i)}_{n+1} \tag{3.1.39}$$

$$\dot{T}^{(i+1)}_{n+1} = \dot{T}^{(i)}_{n+1} + \Delta\dot{T}^{(i)}_{n+1} \tag{3.1.40}$$

$$T^{(i+1)}_{n+1} = T^{(i)}_{n+1} + \Delta T^{(i)}_{n+1} \tag{3.1.41}$$

The superscripts L and R refer to the left and right-hand sides of (3.1.38). We reserve the option of having different evaluations for M and C on different sides, so that we can accomodate all of the algorithms described in chapter 2.

The recurrence rule defined above can be expressed as:

$$\underset{\sim}{Y}_{n+1}^{(i+1)} = \underset{\sim}{J}\ \underset{\sim}{Y}_{n+1}^{(i)} \tag{3.1.42}$$

$\underset{\sim}{J}$ is the iteration matrix emanating from the recurrence rule:

$$\underset{\sim}{J} = \begin{bmatrix} 1-\alpha \dfrac{\Delta t C^R}{\overline{M}} & -\alpha\Delta t \dfrac{M^R}{\overline{M}} \\ -\dfrac{C^R}{\overline{M}} & 1 - \dfrac{M^R}{\overline{M}} \end{bmatrix} \tag{3.1.43}$$

where

$$\overline{M} = M^L + \alpha\Delta t\ C^L \tag{3.1.44}$$

Combining the predictor and iterative phases, we have

$$\underset{\sim}{Y}_{n+1} = \underset{\sim}{A}\ \underset{\sim}{Y}_n \tag{3.1.45}$$

$$\underset{\sim}{A} = \underset{\sim}{J}^S\ \underset{\sim}{P} \tag{3.1.46}$$

Here S is the number of iterations. Exploiting the fact that $\det \underset{\sim}{P} = 0$, that is

$$\det \underset{\sim}{A} = \det \underset{\sim}{J}^S \det \underset{\sim}{P} = 0 \tag{3.1.47}$$

we can write

$$\frac{A_{21}}{A_{11}} = \frac{A_{22}}{A_{12}} = \mu \tag{3.1.48}$$

Then, from (3.1.45):

$$\dot{T}_n = \mu\ T_n \quad , \quad \dot{T}_{n+1} = \mu\ T_{n+1} \tag{3.1.49}$$

Substituting (3.1.49) into (3.1.45):

$$T_{n+1} = (\text{tr } \underset{\sim}{A}) T_n \tag{3.1.50}$$

Numerical Frequency and Damping Coefficients

For the temporal component, we assume a solution of the form:

$$T_n = e^{\tilde{\nu} \Delta t \, n} \tag{3.1.51}$$

where $\tilde{\nu}$ is the numerical counterpart of the exact ν defined by (3.1.6). From (3.1.50) and (3.1.51):

$$e^{\tilde{\nu} \Delta t} = \text{tr } \underset{\sim}{A} \tag{3.1.52}$$

Now, we need to calculate $\text{tr } \underset{\sim}{A}$:

$$\text{tr } \underset{\sim}{A} = \text{tr}(\underset{\sim}{J}^S \underset{\sim}{P}) = J^S_{11} + (1 - \alpha)\Delta t \, J^S_{21} \tag{3.1.53}$$

where J^S_{11} and J^S_{21} are components of $\underset{\sim}{J}^S$. $\underset{\sim}{J}^S$ can be calculated by way of the Cayley-Hamilton theorem:

$$\underset{\sim}{J}^S = a \, \underset{\sim}{I} + b \, \underset{\sim}{J} \tag{3.1.54}$$

The coefficients a and b are functions of the eigenvalues, λ_1 and λ_2 , of the matrix $\underset{\sim}{J}$:

$$a = (\lambda^S_1 - \lambda^S_2)/(\lambda_1 - \lambda_2) \tag{3.1.55}$$

$$b = (\lambda_1 \lambda^S_2 - \lambda_2 \lambda^S_1)/(\lambda_1 - \lambda_2) \tag{3.1.56}$$

The eigenvalues are given by

$$\lambda_1 = 1 \tag{3.1.57}$$

$$\lambda_2 = 1 - (M^R + \alpha\, \Delta t\, C^R)/\overline{M} \tag{3.1.58}$$

From (3.1.53)-(3.1.58):

$$\text{tr}\, \underset{\sim}{A} = 1 + \frac{Q}{R}\left((1 - R)^S - 1\right) \tag{3.1.59}$$

where

$$R = (M^R + \alpha\, \Delta t\, C^R)/\overline{M} \tag{3.1.60}$$

$$Q = \Delta t C^R/\overline{M} \tag{3.1.61}$$

One can note that, from (3.1.30)-(3.1.33), the terms M and ΔtC can be expressed in terms of the dimensionless parameters q, $C_{\Delta t}$ and $C_{2\tau}$:

$$M = \frac{1}{2}\,(\underset{\sim}{D}_r - C_{2\tau}\, \underset{\sim}{D}_1)\underset{\sim}{E} \tag{3.1.62}$$

$$\Delta tC = C_{\Delta t}\,(\underset{\sim}{D}_1 + C_{2\tau}\, \underset{\sim}{D}_2)\underset{\sim}{E} \tag{3.1.63}$$

From (3.1.52) and (3.1.59)-(3.1.63), we can express the complex parameter $\tilde{\nu}\Delta t$ in terms of the dimensionless parameters q, $C_{\Delta t}$ and $C_{2\tau}$, that is,

$$\tilde{\nu}\Delta t = \ln(\text{tr}\, \underset{\sim}{A}) \tag{3.1.64}$$

For comparison, we need to express $\nu\Delta t$ in terms of the same dimensionless parameter set. From (3.1.6):

$$\nu\Delta t = -\, i\lambda k\, \Delta t = -\, i\, \frac{\lambda \Delta t}{h}\, kh$$

$$= -\, i\, C_{\Delta t}\, q \tag{3.1.65}$$

By means of (3.1.64) and (3.1.65) we can study the stability and accuracy of a wide variety of algorithms for solving the convection problem.

We define the analytical and numerical amplification factors Z and $\tilde{Z}$ by

$$\tilde{Z} = e^{\nu\Delta t} \tag{3.1.66}$$

$$\tilde{Z} = e^{\tilde{\nu}\Delta t} \tag{3.1.67}$$

3.2 Unified Analysis of Algorithms

Introduction

In section 3.1 we developed the tools for stability and accuracy analysis of a large class of algorithms. The algorithms can be of implicit or explicit type. Unification of algorithms into one class enables us to perform one general stability and accuracy analysis for the entire class and then study individual cases within the framework of this analysis.

For the purpose of analysis, we classify the algorithms we study into two groups: the implicit types are the base algorithms and the explicit types are the ones derived from the base algorithms. To each implicit algorithm defined, we can (at least in principle) associate an explicit algorithm. This concept was described in section 2.4.

Further, we unify all the algorithms we study into one general Petrov-Galerkin class. Two main parameters $C_{2\tau}$ and r (defined in section 2.6) determine the particular algorithm in this class.

We obtain closed form expressions for the modulus ($|\tilde{z}| = e^{-\tilde{\xi}\Delta t}$) and the frequency ($\tilde{\omega}\,\Delta t$). These expressions are simple for the implicit and explicit 1-pass algorithms, and somewhat more complex for the explicit 2-pass algorithms.

Unconditional-stability proofs can easily be made for the implicit algorithms; stability limits can be determined for the explicit 1-pass algorithms with the same ease.

Expressions for the exact and numerical frequencies are not in easily comparable forms. One can expand these expressions in q or $C_{\Delta t}$ and compare them in series form; this requires extra caution and patience in the algebraic bookkeeping. Alternately, one can, with a general-purpose program, compute

the ratio $\tilde{\omega}/\omega$ for the desired ranges of $C_{2\tau}$, $C_{\Delta t}$ and q. The latter approach is adapted herein.

3.2.1 Description and Classification of Algorithms

In the following sections we briefly describe the (implicit) algorithms and place them in the general class.

(Bubnov-) Galerkin Algorithms

In this group we study two Galerkin ($C_{2\tau} = 0$) algorithms; we call these GC and GL.

GC is a galerkin algorithm with consistent (exactly integrated) mass.

GL, on the other hand, has a lumped mass (integrated with nodal trapezoidal rule).

The stencils $\underset{\sim}{D}_r$ and $\underset{\sim}{D}_l$ are

$$\underset{\sim}{D}_r = 2\,[r,\ 1 - 2r,\ r] \tag{3.2.1}$$

$$\underset{\sim}{D}_l = [\ -\tfrac{1}{2},\ 0,\ \tfrac{1}{2}\ ] \tag{3.2.2}$$

The parameter r, which was described in section 2.6, is set to 1/6 for GC and to 0 for GL. The form of the stencil $\underset{\sim}{D}_l$ corresponds to exact integration and is equivalent to central differences.

Petrov-Galerkin Algorithms

We study four Petrov-Galerkin algorithms. They are PG(Padé), PG($C_{2\tau} = 1$), PG($C_{2\tau} = 2/\sqrt{15}$) and PG($C_{2\tau} = C_{\Delta t}$).

PG(Padé) has weighting function $\tilde{W}$ constructed from the following element shape functions:

$$\tilde{N}_1(\xi) = 0 \tag{3.2.3}$$

$$\tilde{N}_2(\xi) = 1 \tag{3.2.4}$$

where

$$\xi \in [-1, +1] \tag{3.2.5}$$

is the usual isoparametric coordinate. In this case we need to replace $\underset{\sim}{D}_r$ and $\underset{\sim}{D}_1$ by

$$2[\tfrac{1}{2}, \tfrac{1}{2}, 0] \tag{3.2.6}$$

and

$$[-1, +1, 0] , \tag{3.2.7}$$

respectively.

This method corresponds to the Padé finite difference approximation (see [W3]).

The other methods, that is

PG($C_{2\tau} = 1$) [H15] ,

PG($C_{2\tau} = 2/\sqrt{15}$) [R1] and

PG($C_{2\tau} = C_{\Delta t}$) (described in chapter 2, section 5)

employ (3.1.24), (3.1.25) and (3.1.26), corresponding to exact integration rules. The name of each algorithm implies the way $C_{2\tau}$ is chosen.

3.2.2 Implicit Algorithms

From (3.1.62)-(3.1.63), the scalars M and ΔtC become

$$M = 1 - 2r\,(1 - \cos q) - i\,\frac{C_{2\tau}}{2}\sin q \tag{3.2.8}$$

$$\Delta tC = C_{\Delta t}C_{2\tau}\,(1 - \cos q) + i\,C_{\Delta t}\sin q \tag{3.2.9}$$

The stencils of the Padé approximation, (3.2.6) and (3.2.7), lead to,

$$M = \frac{1}{2}\,(1 + e^{-iq}) \tag{3.2.10}$$

$$\Delta tC = C_{\Delta t}\,(1 - e^{-iq}) \tag{3.2.11}$$

Remark: The same expressions can also be obtained by setting $r = 1/4$ and $C_{2\tau} = 1$ in (3.2.8)-(3.2.9). Therefore, for the purpose of analysis, we can include the PG(Padé) algorithm in the two-parameter $(C_{2\tau}, r)$ family of Galerkin/Petrov-Galerkin algorithms.

For the implicit class, (3.1.52), (3.1.59) and (3.1.67) reduce to the following expression for the numerical amplification factor:

$$\tilde{Z} = 1 - Q \tag{3.2.12}$$

where Q is given by (3.1.61).

Frequency Analysis

The following expression for $\tilde{\omega}\,\Delta t$ is found:

$$\tan(\tilde{\omega}\,\Delta t) = C_{\Delta t}\,W(G + C_{2\tau}^2\,V/2)/$$

$$[(G + \alpha\,C_{\Delta t}C_{2\tau}V)(G - (1 - \alpha)C_{\Delta t}C_{2\tau}V) +$$

$$(H + \alpha\,C_{\Delta t}\,W)(H - (1 - \alpha)C_{\Delta t}\,W)] \tag{3.2.13}$$

where

$$V = 1 - \cos q \tag{3.2.14}$$

$$W = \sin q \tag{3.2.15}$$

$$G = 1 - 2\, r\, V \tag{3.2.16}$$

$$H = - C_{2\tau}\, W/2 \tag{3.2.17}$$

Eq. (3.2.13) is a general expression for any combination of r, $C_{2\tau}$ and α. For example, for the trapezoidal rule ($\alpha = \frac{1}{2}$) the expression reduces to

$$\tan(\tilde{\omega}\,\Delta t) = C_{\Delta t}\, W(G + C_{2\tau}^2\, V/2)/ [G^2 - (\tfrac{1}{2}\, C_{\Delta t} C_{2\tau}\, V)^2 + H^2 - (\tfrac{1}{2}\, C_{\Delta t}\, W)^2] \tag{3.2.18}$$

Further, if we have a Galerkin algorithm ($C_{2\tau} = 0$), then (3.2.18) becomes:

$$\tan(\tilde{\omega}\,\Delta t) = C_{\Delta t}\, W\, G/[G^2 - (\tfrac{1}{2}\, C_{\Delta t}\, W)^2] \tag{3.2.19}$$

Lumping the mass term reduces this expression to:

$$\tan(\tilde{\omega}\,\Delta t) = C_{\Delta t}\, W\, /\, [1 - (\tfrac{1}{2}\, C_{\Delta t}\, W)^2] \tag{3.2.20}$$

Modulus/Stability Analysis

The following inequality needs to be satisfied for the stability of an algorithm:

$$|\tilde{z}|^2 \leq 1 \tag{3.2.21}$$

This inequality translates to:

$$-(1 - 4\, r) C_{2\tau}\, V - (2\,\alpha - 1)(C_{2\tau}^2\, V + (2 - V)) \leq 0 \tag{3.2.22}$$

As we can easily observe, for $\alpha \geq \frac{1}{2}$, all the algorithms considered are unconditionally stable, provided that $r \leq 1/4$. In particular, GC($C_{2\tau} = 0$, $r = 1/6$) , GL($C_{2\tau} = 0$, $r = 0$) and PG(Padé) ($C_{2\tau} = 1$, $r = 1/4$) have no modulus error, that is, $|\tilde{Z}| = 1$.

3.2.3 Explicit Algorithms

For explicit algorithms, the scalar quantities M^R , ΔtC^R and $\overline{M}$ are:

$$M^R = G + i\,H \tag{3.2.23}$$

$$\Delta tC^R = C_{\Delta t}C_{2\tau}\,V + i\,C_{\Delta t}\,W \tag{3.2.24}$$

$$\overline{M} = 1 . \tag{3.2.25}$$

Explicit 1-Pass Algorithms

For 1-pass algorithms, the numerical amplification factor takes the form

$$\tilde{Z} = 1 - \Delta tC^R \tag{3.2.26}$$

that is

$$\tilde{Z} = (1 - C_{\Delta t}C_{2\tau}V\ ,\ -C_{\Delta t}W) \tag{3.2.27}$$

For the frequency analysis, we get a relatively simple expression:

$$\tan(\tilde{\omega}\,\Delta t) = \frac{C_{\Delta t}\,W}{1 - C_{\Delta t}C_{2\tau}W} \tag{3.2.28}$$

Clearly, for 1-pass algorithms, r and α have no effect.

For modulus/stability analysis, the following inequality is considered:

$$|\tilde{Z}|^2 = (1 - C_{\Delta t}C_{2\tau}V)^2 + (C_{\Delta t}\,W)^2 \leq 1 \tag{3.2.29}$$

Utilizing the identity $W^2 = 2V - V^2$, this condition can be written as:

$$2(C_{2\tau} - C_{\Delta t}) + C_{\Delta t}\, V(1 - C_{2\tau})^2 \geq 0 \qquad (3.2.30)$$

We can further write this inequality in two other forms:

The first form,

$$\frac{1}{C_{\Delta t}V} - \left(1 - \frac{2}{V} + \frac{1}{(C_{\Delta t}V)^2}\right)^{\frac{1}{2}} \leq C_{2\tau} \leq \frac{1}{C_{\Delta t}V} + \left(1 - \frac{2}{V} + \frac{1}{(C_{\Delta t}V)^2}\right)^{\frac{1}{2}} \qquad (3.2.31)$$

provides us with the stability limits on the algorithmic parameter $C_{2\tau}$.

The second form,

$$C_{\Delta t} \leq \frac{2C_{2\tau}}{\left(2 - V(1 - C_{2\tau}^2)\right)} \qquad (3.2.32)$$

provides the stability limit on the Courant number once we select the parameter $C_{2\tau}$.

Consider the following cases:

If $C_{2\tau} = 0$, then, for the stability condition we get $V - 2 \geq 0$. This implies unconditional unstability.

If we set $C_{2\tau} = 1$, then, the stability condition is $C_{\Delta t} \leq 1$.

By setting $C_{2\tau} = C_{\Delta t}$, we get the same stability condition: $C_{\Delta t} \leq 1$.

Explicit 2-Pass Algorithms

For the 2-pass algorithms, the numerical amplification factor is

$$\tilde{Z} = 1 + \Delta t C^R(M^R + \alpha\, \Delta t C^R - 2) \qquad (3.2.33)$$

With this form of $\tilde{Z}$, the expressions for $\tan(\tilde{\omega}\,\Delta t)$ and the stability condition become rather complicated. Defining the variables

$$(A_1,\ A_2) \;=\; \Delta t C^R \tag{3.2.34}$$

$$(B_1,\ B_2) \;=\; M^R + \alpha\,\Delta t C^R - 2 \tag{3.2.35}$$

we can, without any further algebraic elaboration, write the expressions for $\tan(\tilde{\omega}\,\Delta t)$, that is,

$$\tan(\tilde{\omega}\,\Delta t) \;=\; -\;\frac{A_1B_2 + A_2B_1}{1 + A_1B_1 - A_2B_2} \tag{3.2.36}$$

and for the stability condition

$$2(A_1B_1 - A_2B_2) + (A_1^2 + A_2^2)(B_1^2 + B_2^2) \;\leq\; 0 \quad . \tag{3.2.37}$$

3.2.4 Stability and Accuracy Studies

As the algebra of the analysis gets lengthier, especially for the 2-pass algorithms, we find it more convenient to conduct the analysis numerically for the desired ranges of the parameters involved.

For the algorithms previously considered we utilized a general purpose program to determine the stability and accuracy properties. For each algorithm, the parameters r and $C_{2\tau}$ were given. The quantities $\tilde{\xi}/\tilde{\omega}$ and $\tilde{\omega}/\omega$ were computed and plotted for the values of $C_{\Delta t} = 0.2, 0.4, 0.6, 0.8, 1.0$ and $q \in]0, \pi[$.

For future reference, we describe the following concepts:

Unit CFL condition [M3]: An algorithm satisfies the unit CFL condition if it produces nodally exact solutions for $C_{\Delta t} = 1$.

Order of accuracy: The behaviors of the $\tilde{\xi}/\tilde{\omega}$ and $\tilde{\omega}/\omega$ curves for an algorithm as $q \to 0$, reveal the order of accuracy of that algorithm. 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.

The quantity $\tilde{\xi}/\tilde{\omega}$ is called the algorithmic damping ratio (see [H2]) and is related to the logarithmic decrement $\tilde{\delta}$ [H2] via the following expression

$$\tilde{\delta} = \ln \frac{\tilde{T}(t_n)}{\tilde{T}\left(t_n + \frac{2\pi}{\tilde{\omega}}\right)} = 2\pi(\tilde{\xi}/\tilde{\omega}) \qquad (3.2.38)$$

where $\tilde{T}$ is the numerical prediction for the dependent variable.

The dimensionless wave number q is a measure of spatial refinement. Scaling q by 2π , we get:

$$q^* = \frac{q}{2\pi} = \frac{h}{\lambda} \tag{3.2.46}$$

which represents the number of elements per wave length. It is reasonable to limit study to the range $q \in]0, \pi[$. For example $q = \pi/2$ translates to having 4 elements for one full wave form.

For the entire graphical analysis $\alpha = \frac{1}{2}$ (trapezoidal).

3.3 Comparison of Algorithms

3.3.1 Implicit Algorithm

Fig. 3.1 shows the frequency ratio $(\tilde{\omega}/\omega)$ for the implicit GC, GL and PG(Padé) algorithms. They have no modulus error, and are second-order accurate. GC is more accurate than GL for finite q . This is due to the term $2\,r(1 - \cos q)$ which vanishes for GL. 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. 3.2 shows the algorithmic damping ratio $(\tilde{\xi}/\tilde{\omega})$ and the frequency ratio for the implicit $PG(C_{2\tau} = 1)$, $PG(C_{2\tau} = 2/\sqrt{15})$ and $PG(C_{2\tau} = C_{\Delta t})$ algorithms. They are all second-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 $\Delta t \to 2/\sqrt{15}$.

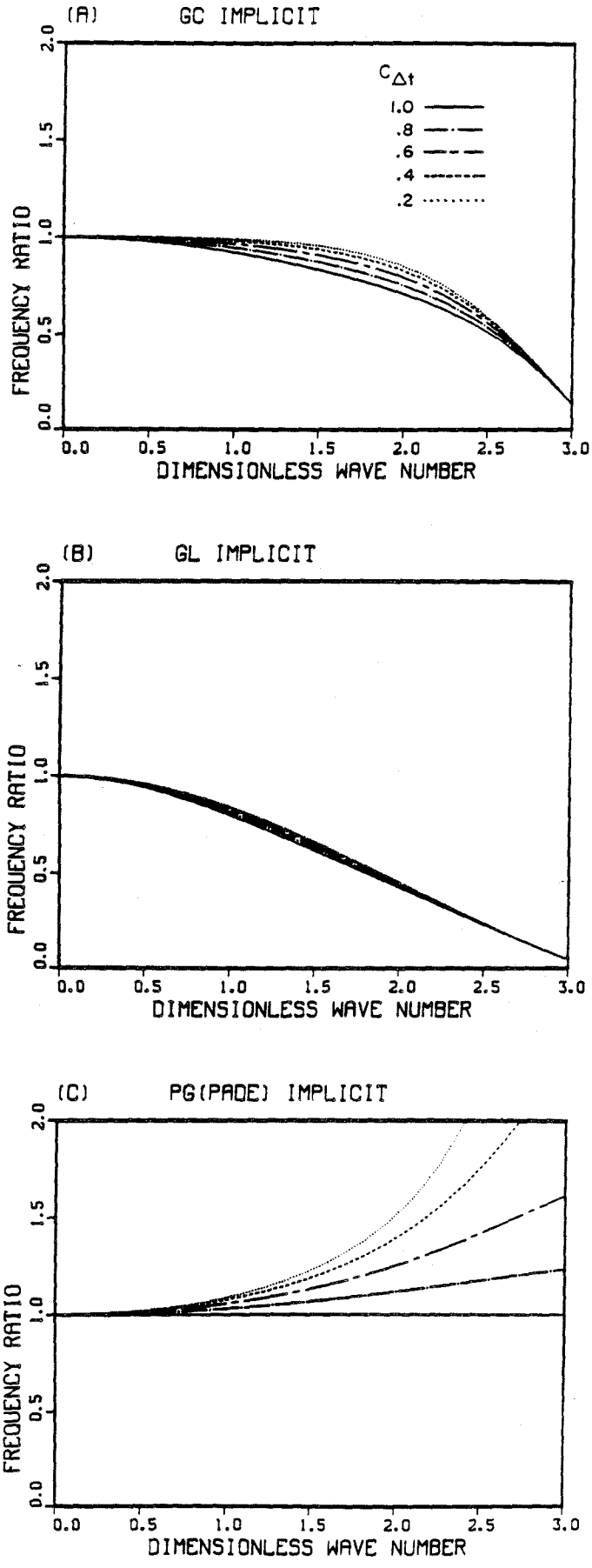


Figure 3.1 Frequency ratios for implicit Galerkin and Petrov-Galerkin (Padé) algorithms

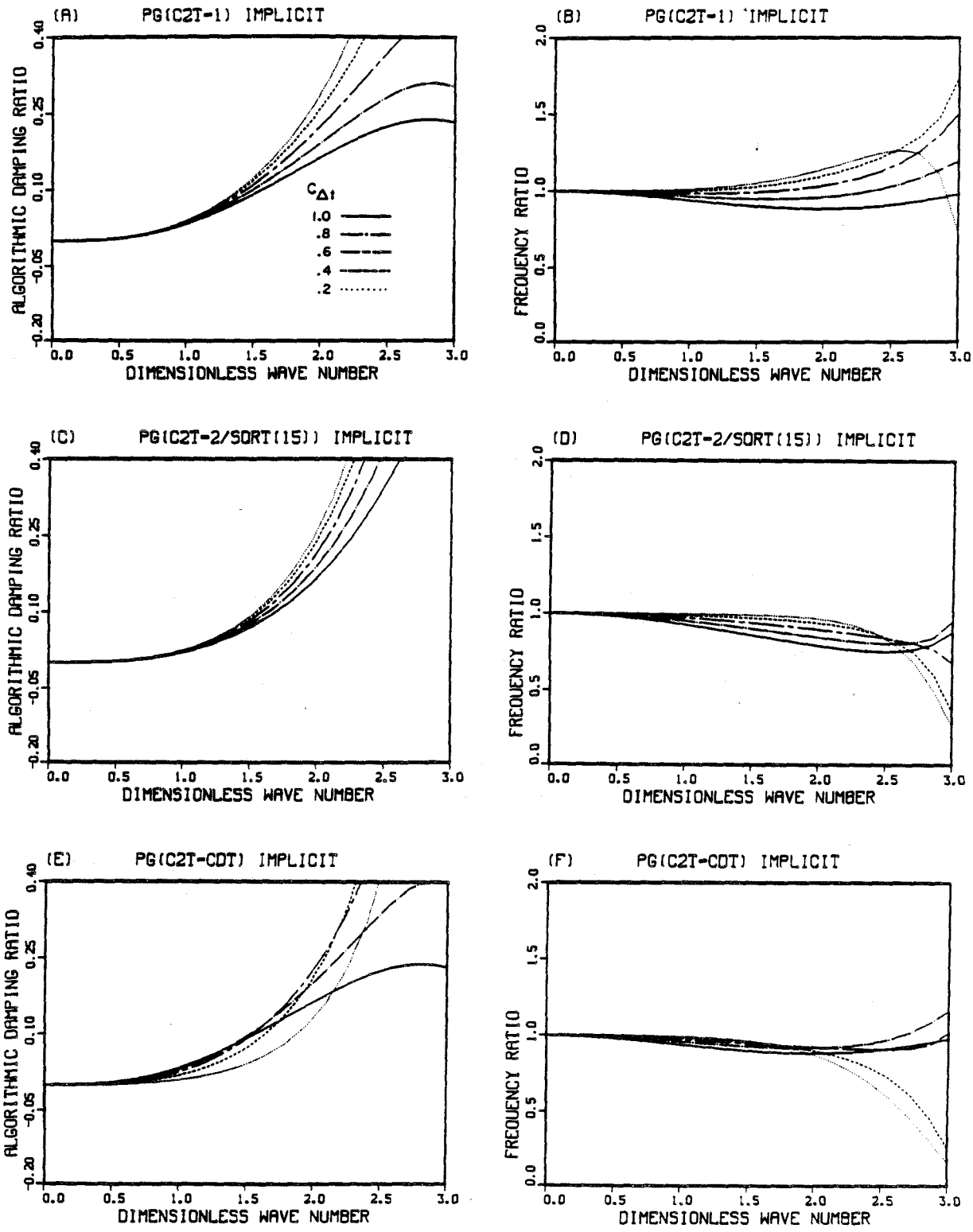


Figure 3.2 Algorithmic damping and frequency ratios for Petrov-Galerkin algorithms

3.3.2 Explicit 1-Pass Algorithms

Fig. 3.3 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. This superiority with respect to order-of-accuracy can easily be seen from (3.2.29), which can be written as:

$$|\tilde{z}|^2 = 1 - C_{\Delta t} V(2(C_{2\tau} - C_{\Delta t}) + C_{\Delta t} V(1 - C_{2\tau}^2)) \qquad (3.3.1)$$

Clearly, the departure of $|\tilde{z}|^2$ from unity is first-order in $C_{\Delta t}$ and V. However, if $C_{2\tau} = C_{\Delta t}$, then

$$|\tilde{z}|^2 = 1 - (C_{\Delta t} V)^2 (1 - C_{\Delta t}^2) \qquad (3.3.2)$$

and thus, the departure of $|\tilde{z}|^2$ from unity is now second-order in $C_{\Delta t}$ and V.

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.

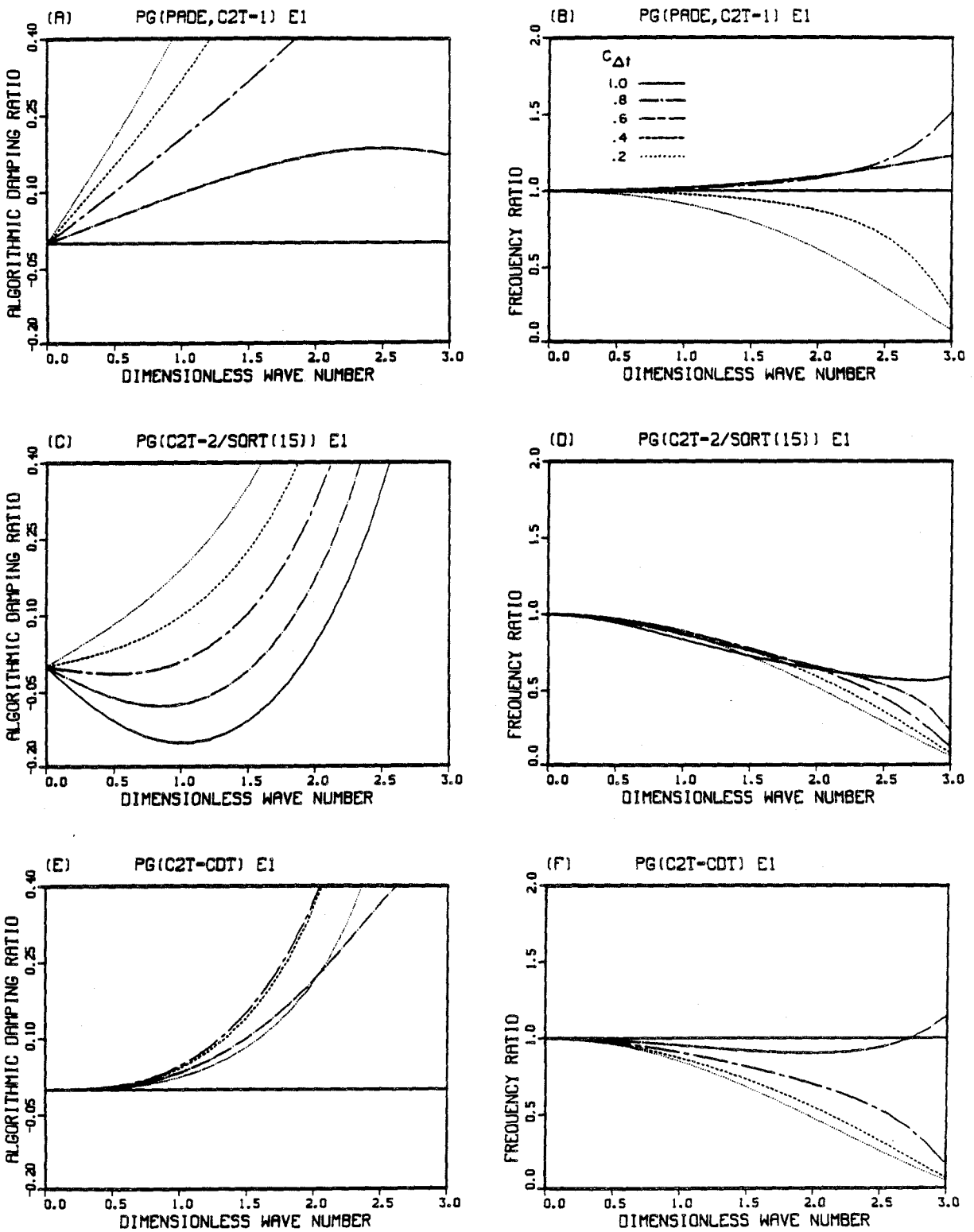


Figure 3.3 Algorithmic damping and frequency ratios for explicit 1-pass algorithms

3.3.3 Explicit 2-Pass Algorithms

Fig. 3.4 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 $C_{\Delta t} = 0.8$.

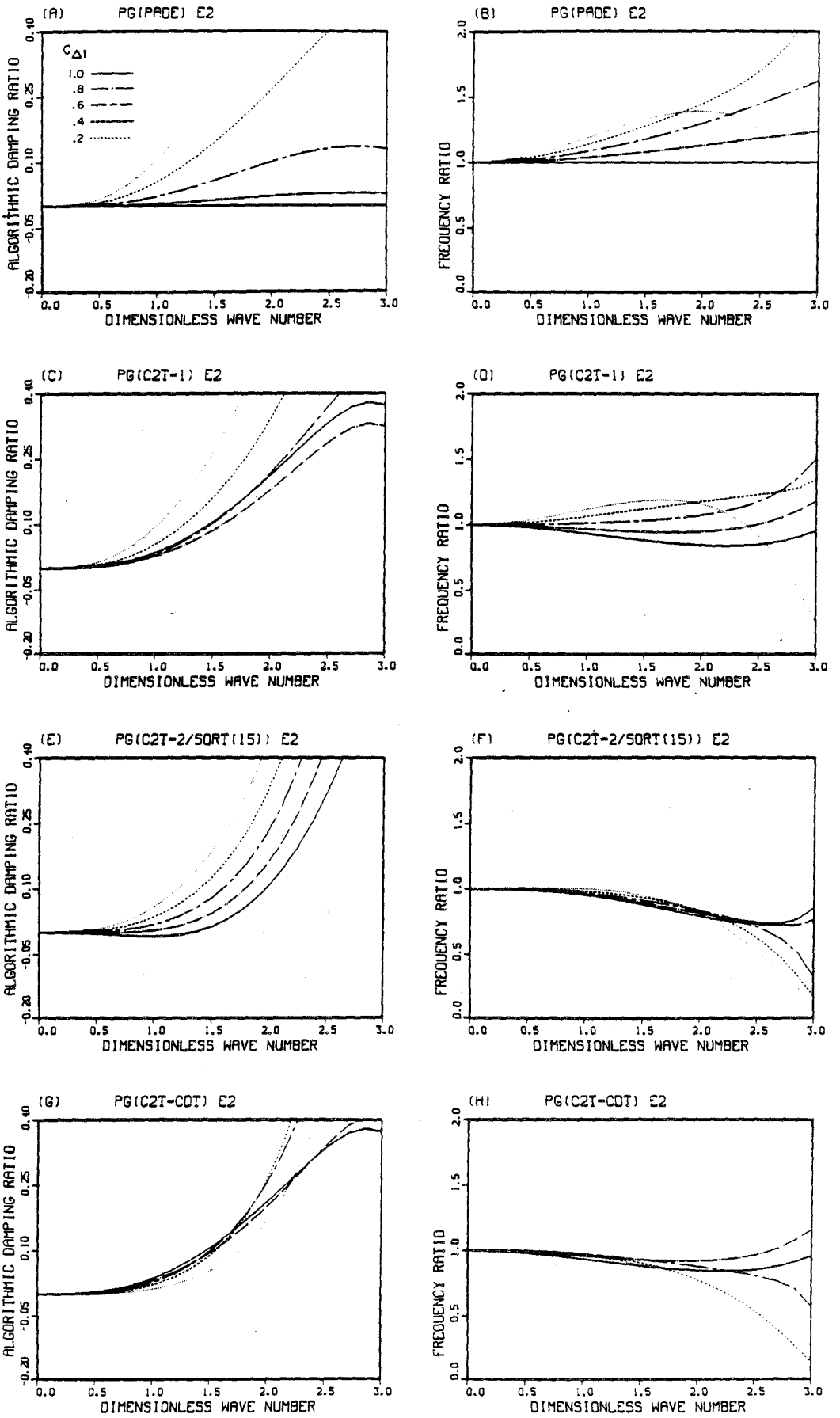


Figure 3.4 Algorithmic damping and frequency ratios for explicit 2-pass algorithms

CHAPTER 4

Numerical Applications in One Dimension

4.1 Introduction

To test the capabilities of our finite element schemes and to estimate the effects of several algorithmic parameters involved, we experimented with problems from various classes of 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 solutions; 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 the way the boundary conditions were specified.

4.2 Numerical Applications in the Linear Transient Case

4.2.1 Propagation of a Small Disturbance in a Gas

Barotropic Compressible Flow Equations

The Euler equations in one dimension are given in appendix I. The mass and momentum conservation equations can be uncoupled from the energy conservation equation by assuming that the flow is barotropic, that is:

$$p = p(\rho) \qquad \text{(pressure)} \tag{4.2.1}$$

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

$$\underset{\sim}{U} = \rho \begin{Bmatrix} 1 \\ u \end{Bmatrix} \tag{4.2.2}$$

$$\underset{\sim}{\mathcal{F}} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \end{Bmatrix} \tag{4.2.3}$$

where ρ and u are density and velocity, respectively.

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 (see [W4]):

$$\left| \frac{\rho}{\rho_0} - 1 \right| \ll 1 \tag{4.2.4}$$

$$\left| \frac{p'(\rho)}{p'(\rho_0)} - 1 \right| \ll 1 \tag{4.2.5}$$

$$\left| \frac{u}{\sqrt{p'(\rho_0)}} \right| \ll 1 \tag{4.2.6}$$

where $p'(\rho)$ is the derivative of $p(\rho)$ with respect to its argument. With these assumptions, we get the following linearized version of the original conservation equation system:

$$\rho_{,t} + \rho_0 u_{,x} = 0 \tag{4.2.7}$$

$$\rho_0 u_{,t} + p'(\rho_0)\rho_{,x} = 0 \tag{4.2.8}$$

Further, by introducing the parameter c ,

$$c^2 = p'(\rho_0) \tag{4.2.9}$$

and the scaling

$$\rho^* = \rho/\rho_0 - 1 \tag{4.2.10}$$

$$u^* = - u/c \tag{4.2.11}$$

we get

$$\underset{\sim}{U}_{,t} + \underset{\sim}{A}\,\underset{\sim}{U}_{,x} = 0 \tag{4.2.12}$$

where

$$\underset{\sim}{U} = \begin{Bmatrix} \rho^* \\ u^* \end{Bmatrix} \tag{4.2.13}$$

and

$$\underset{\sim}{A} = \begin{bmatrix} 0 & -c \\ -c & 0 \end{bmatrix} \tag{4.2.14}$$

Clearly, the eigenvalues of $\underset{\sim}{A}$ are:

$$\lambda_{1,2} = \pm c \tag{4.2.15}$$

4.2.2 Initial/Boundary-value Problem

The equation system of (4.2.12), together with the following initial/boundary-value data was studied numerically by Hughes in [H6]:

$$\left.\begin{aligned} \rho(x, 0) &= 0 \\ u(x, 0) &= 1 \end{aligned}\right\} \quad x \in]0, 10[\tag{4.2.16}$$

$$\left.\begin{aligned} \rho(0,\ t) &= 0 \\ u(10,\ t) &= 0 \end{aligned}\right\} \quad t \geq 0 \tag{4.2.17}$$

(We drop the asterisks for notational simplicity.) The parameter c was chosen to be unity.

4.2.3 Finite Element Solutions

Naming Convention for the $\underset{\sim}{T} = \tau\, \underset{\sim}{A}$ and $\underset{\sim}{T} = \tau\, \underset{\sim}{A}^T$

If we choose to define the operator $\underset{\sim}{T}$ according to the criterion $\underset{\sim}{T} = \tau\, \underset{\sim}{A}$, then the resulting formulation would have a second-order term containing the product $\underset{\sim}{A}^T\underset{\sim}{A}$. Therefore, we name this criterion the "A^TA-form." If we choose the criterion $\underset{\sim}{T} = \tau\, \underset{\sim}{A}^T$, on the other hand, then the second-order term would contain the product $\underset{\sim}{A}^2$. Therefore we name this criterion the "A^2-form". Thus, in the study of this and all the other problems, we adopt the following naming convention:

$$\underset{\sim}{T} = \tau\, \underset{\sim}{A} \longleftrightarrow A^TA \tag{4.2.18}$$

$$\underset{\sim}{T} = \tau\, \underset{\sim}{A}^T \longleftrightarrow A^2 \tag{4.2.19}$$

This naming convention will also be used for the multi-dimensional cases.

Algorithmic Features

In this problem, both A^TA and A^2-forms result in the same formulation due to the symmetry of the operator $\underset{\sim}{A}$. We can also set $\underset{\sim}{T} = \underset{\sim}{0}$ and obtain the usual Galerkin technique.

The finite element mesh contains 20 elements with uniform mesh spacing

of .5 . The quadrature rules chosen provide exact integration of the vectors and matrices involved.

The transient algorithm parameters γ and α were set to be 1.0 and 0.5 respectively. Implicit, explicit 1-pass (designated by E1), and explicit 1-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, we used the temporal criterion for τ as given by (2.5.8). The parameter F was set to unity. The spatial criterion for τ , (2.5.1), would result in the same formulation for both time steps, provided that we set $F = 0.5$ when $\Delta t = 0.25$.

Results

Fig. 4.1 shows the implicit Galerkin solution for Courant number 1.0 . As can be seen, this technique produces spurious oscillations.

Fig. 4.2 shows the solutions produced by the explicit 1-pass Petrov-Galerkin algorithm with $\alpha = 1.0$ and $\alpha = 0.5$. One would expect the solutions to be very similar because the explicit 1-pass algorithm is independent of α . Slight differences occur because the initial start-up conditions are handled differently by each algorithm.

Both algorithms satisfy the unit CFL condition defined in chapter 3 yet, in this problem, $\alpha = 0.5$ produced results inferior to $\alpha = 1.0$ due to start-up conditions. Therefore, for the explicit 1-pass algorithm, when it comes to a choice between $\alpha = 1.0$ and $\alpha = 0.5$, the former value is preferred.

Fig. 4.3 shows the solutions (for Courant number 1.0) produced by Petrov-

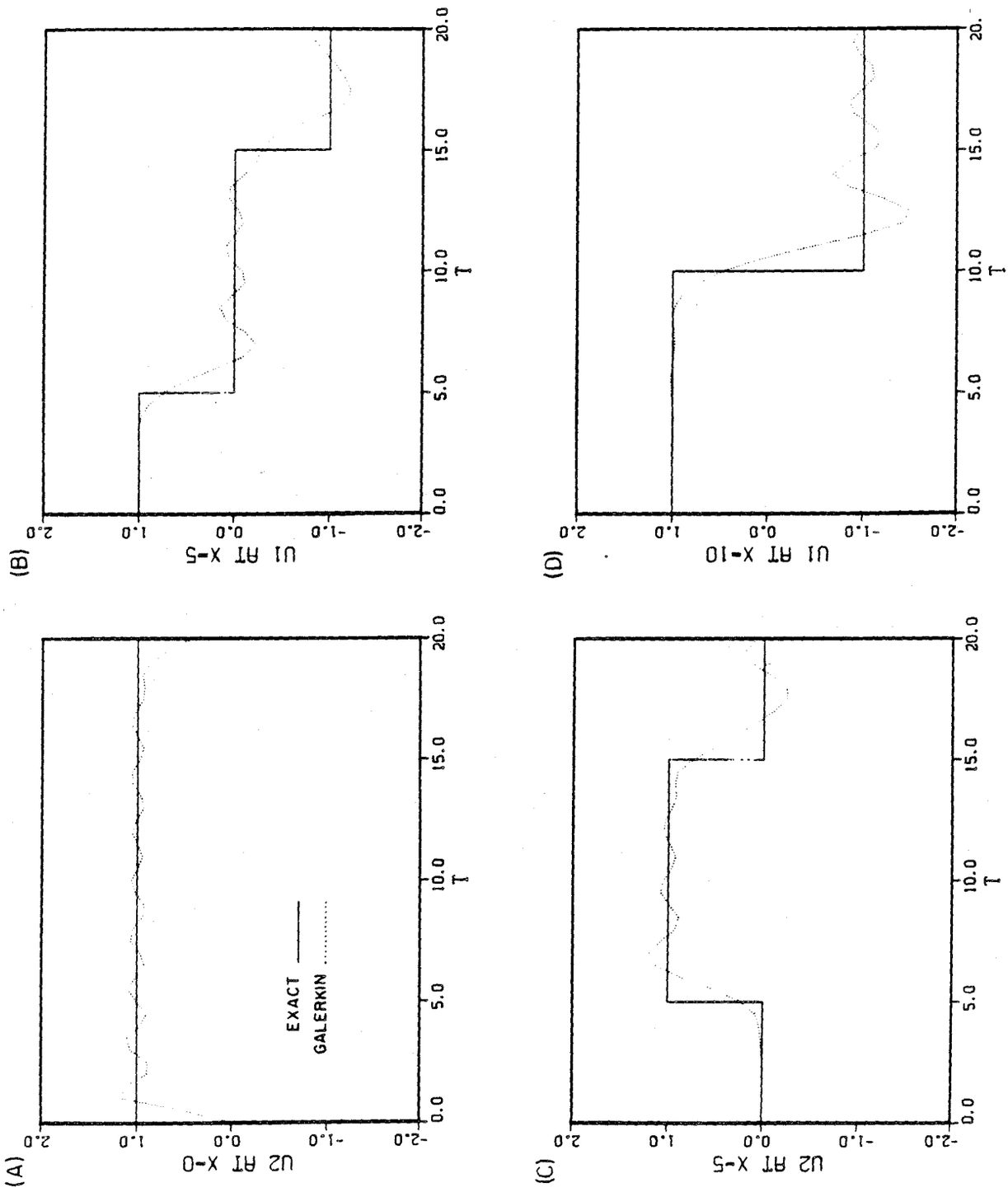


Figure 4.1 Small disturbance propagation in a gas: Galerkin implicit, $n_{e\ell} = 20$, $\Delta t = .5$, Courant number = 1.0.

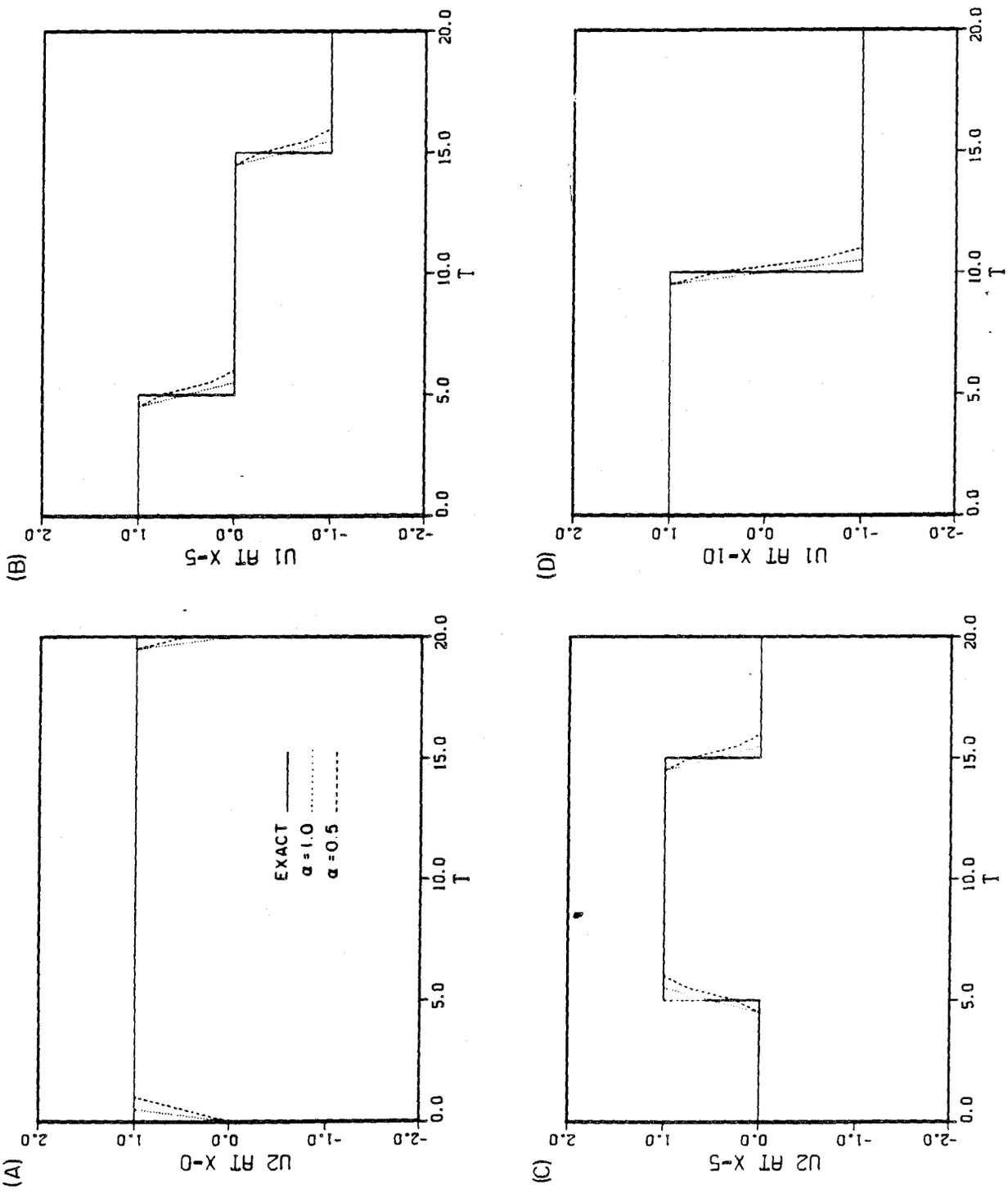


Figure 4.2 Small disturbance propagation in a gas: global τ , explicit 1-pass, $n_{e\ell} = 20$, $\Delta t = .5$, Courant number = 1.0 .

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. 4.4 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.

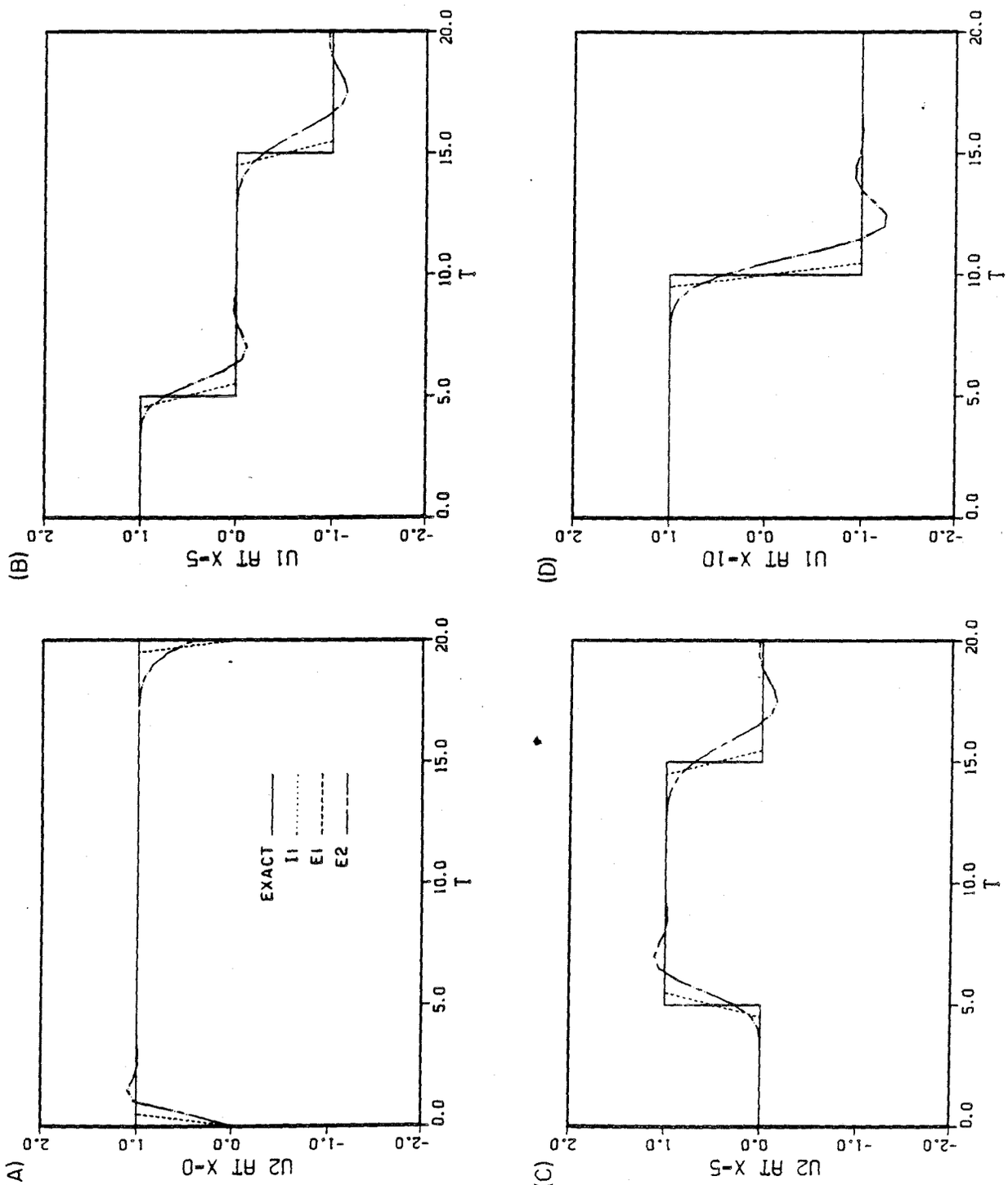


Figure 4.3 Small disturbance propagation in a gas: global τ , $n_{e\ell} = 20$, $\Delta t = .5$, Courant number = 1.0 (I1 and E2 are virtually indistinguishable).

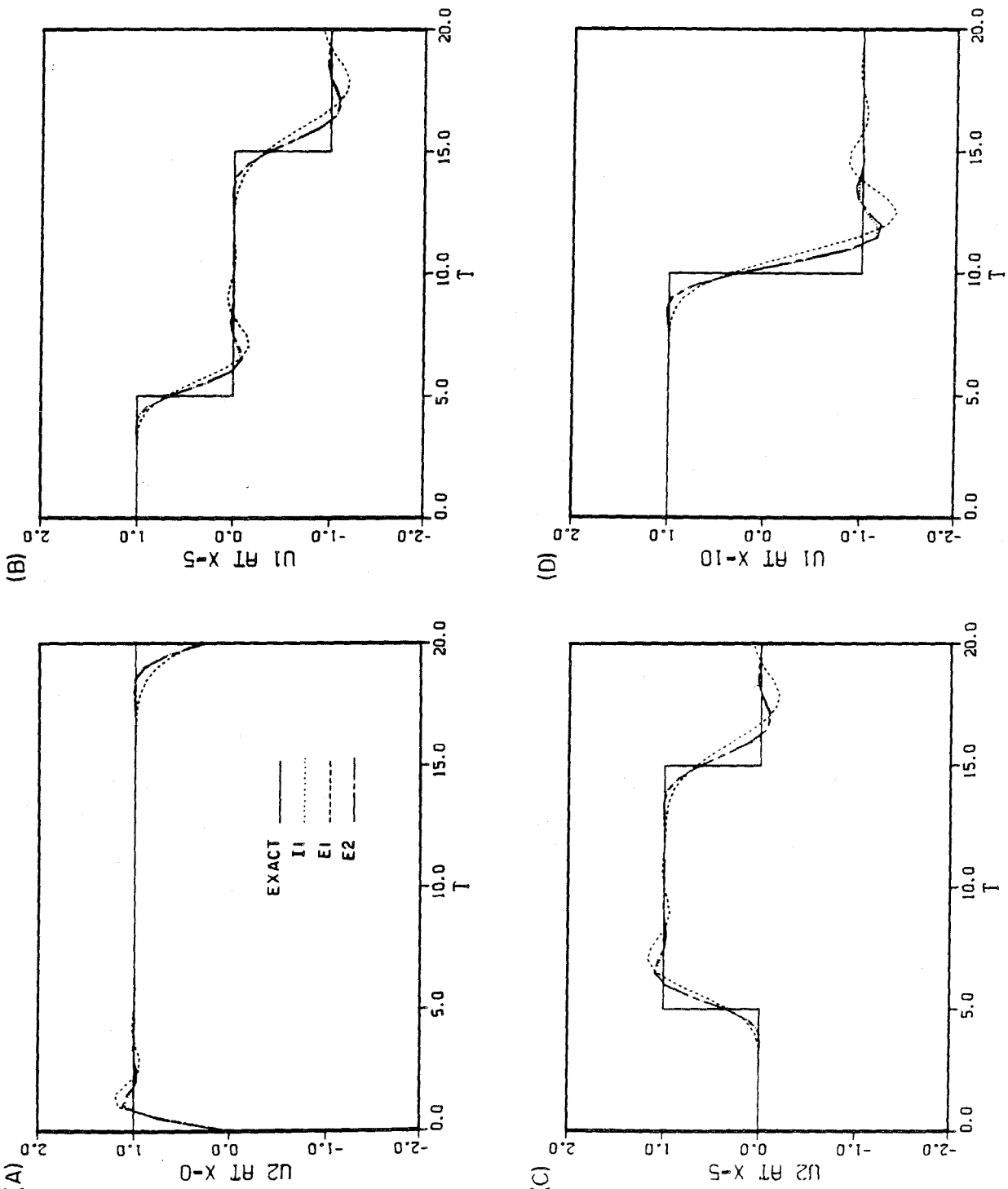


Figure 4.4 Small disturbance propagation in a gas: global τ , $n_{e\ell} = 20$, $\Delta t = 2.5$, Courant number = 0.5 (I1 and E2 are virtually indistinguishable).

4.3 Numerical Applications in the Nonlinear Transient Case

4.3.1 Barotropic Compressible Flow

Barotropic compressible flow was defined in section 4.2. The differential equations need to be satisfied everywhere except at the shock front where the Rankine-Hugoniot conditions (see [L1, L2, L3]),

$$s[\underset{\sim}{U}] = [\underset{\sim}{\mathcal{F}}] \tag{4.3.1}$$

have to be satisfied. Here s is the propagation speed of the shock front and $[\quad]$ is the jump operator. That is, for any variable Q :

$$[Q] = Q^{+} - Q^{-} \tag{4.3.2}$$

where the superscripts "-" and "+" refer to the left and the right of the shock front, respectively. For barotropic flow these conditions are:

$$s[\rho] = [\rho\, u] \tag{4.3.3}$$

$$s[\rho u] = [\rho u^2 + p] \tag{4.3.4}$$

For shock profile to be stable, the entropy condition also must be satisfied (see [L1, L2, L3]). The entropy condition is given in the form of inequalities in terms of s and the eigenvalues of the jacobian matrix $\underset{\sim}{A}$. In the present case, the eigenvalues are

$$\lambda_{1,2} = u \pm \left(p'(\rho)\right)^{\frac{1}{2}} \tag{4.3.5}$$

4.3.2 Initial-value Problems

We considered two initial-value problems, both studied by Hughes in [H6], with the following equation of state:

$$p(\rho) = \frac{1}{27} \rho^3 \tag{4.3.6}$$

The first problem has the following set of initial data:

$$\rho(x, 0) = 1 + 2H(- x) \tag{4.3.7}$$

$$u(x, 0) = \frac{2}{3} H(- x) \tag{4.3.8}$$

where $H(x)$ is the Heaviside 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:

$$\rho(x, 0) = 1 + 2H(+ x) \tag{4.3.9}$$

$$\rho(x, 0) = \frac{2}{3} H(+ x) \tag{4.3.10}$$

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 a rarefaction wave traveling to the right.

4.3.3 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 only.

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

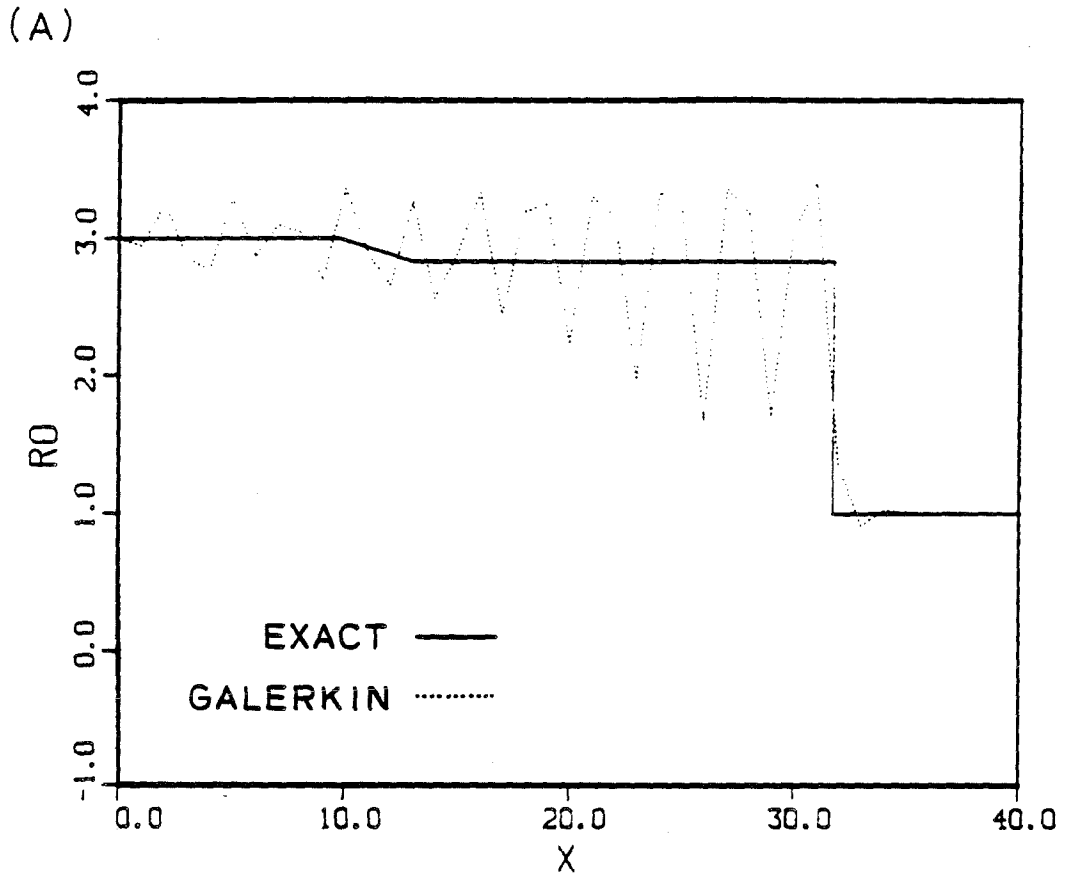
The transient algorithm parameters γ and α were set to 1.0 and 0.5, respectively. The time steps were taken to be 0.6 and 0.3 corresponding to Courant numbers (based on the maximum eigenvalue) 1.0 and 0.5, respectively.

The parameter τ was chosen according to the temporal criterion given by (2.5.8). The parameter F was usually taken to be one, however we tested cases where it was greater than one.

Results

Fig. 4.5 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 solutions were, with varying magnitudes, overshoots at the shock fronts and oscillations behind 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 is used. This observation is in agreement with the findings of Baker [B2, B3].



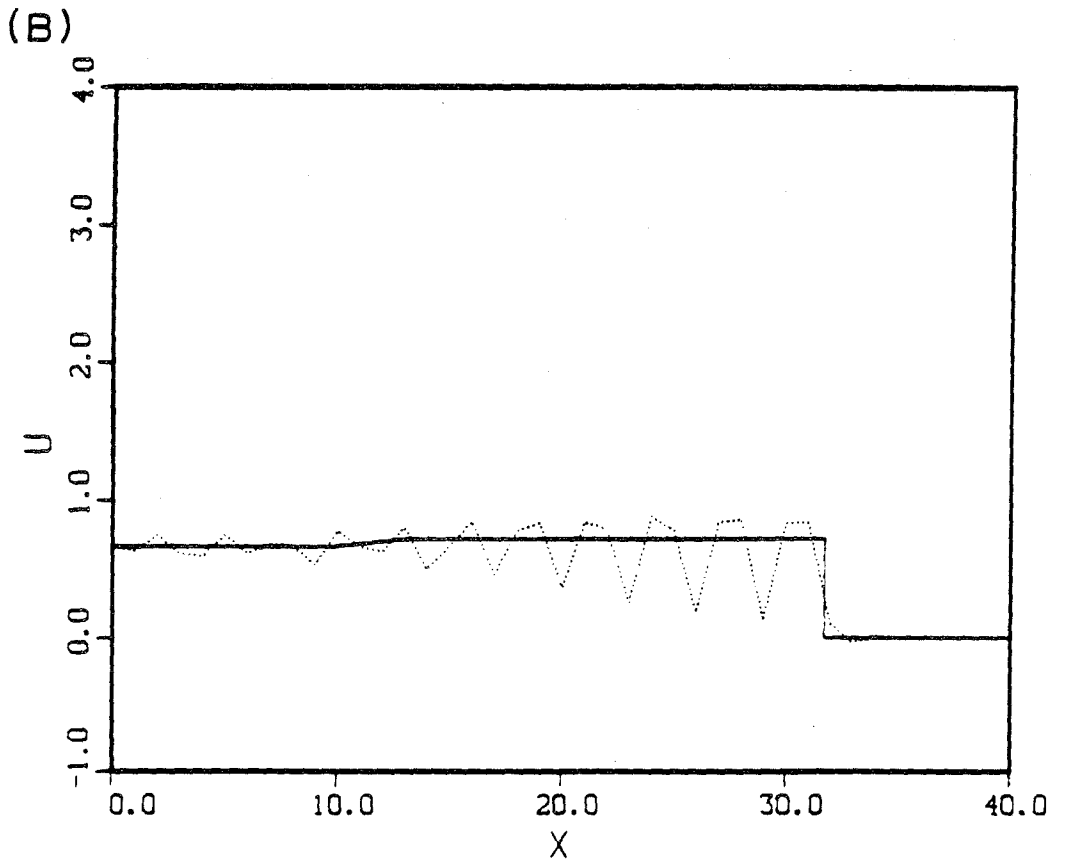


Figure 4.5 Barotropic compressible flow, stable shock propagation: Galerkin implicit 3-pass, $n_{el} = 40$, $\Delta t = .6$, $n_{ts} = 25$, Courant number (based on maximum spectral radius) = 1.0 .

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. 4.6 shows the comparison of the integration rules for $\Delta t = 0.6$ with 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 the quadrature rule. One can therefore use 1-point quadrature for economy reasons without much decrease in accuracy. However, in the following problems we used the 3-point quadrature rule.

Fig. 4.7 shows the results for $\Delta t = 0.6$; all are in close agreement with the exact solution. All have, with comparable magnitudes, overshoots at the shock front and oscillations behind 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. 4.8 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. 4.9 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 2-pass algorithms. Both solutions were in close agreement with the exact solution. The E1 algorithm

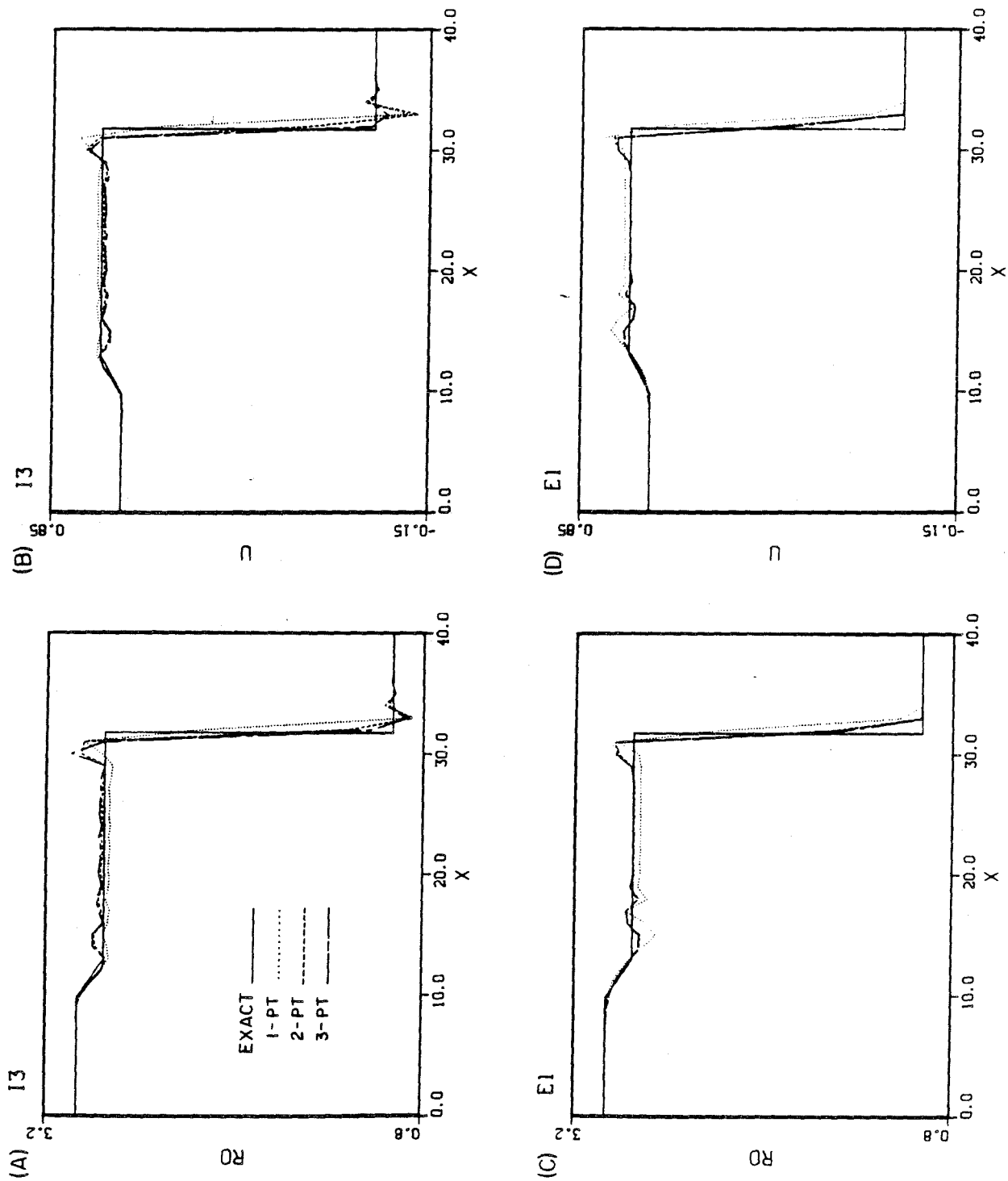


Figure 4.6 Barotropic compressible flow, stable shock propagation: global τ , A^2-form, n_{el} = 40, Δt = .6, n_{ts} = 25, Courant number (based on maximum spectral radius) = 1.0. Comparison of different element quadrature rules.

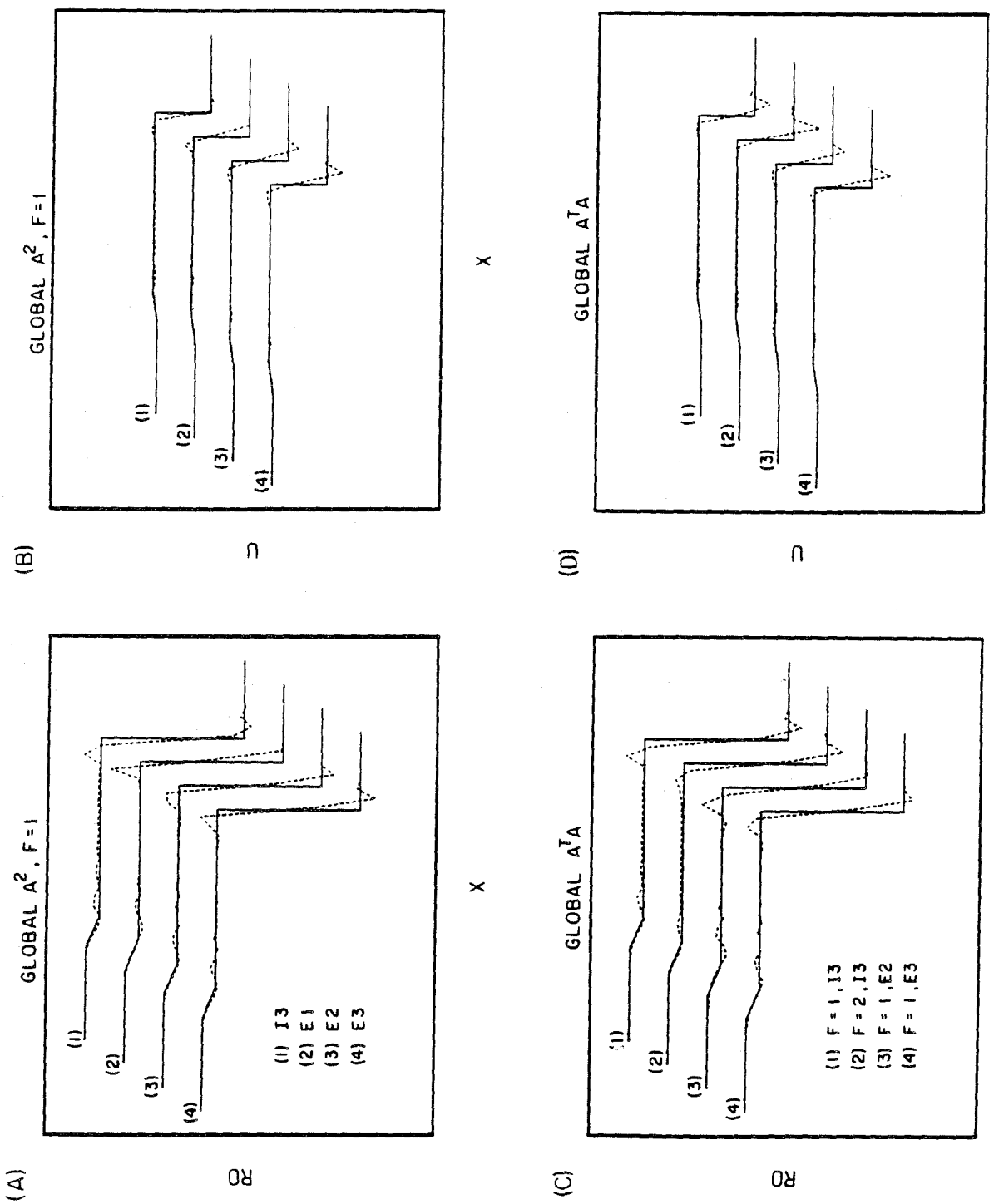


Figure 4.7 Barotropic compressible flow, stable shock propagation: $n_{e\ell} = 40$, $\Delta t = .6$, $n_{ts} = 25$, Courant number (based on maximum spectral radius) = 1.0. Comparison of various SU/PG schemes.

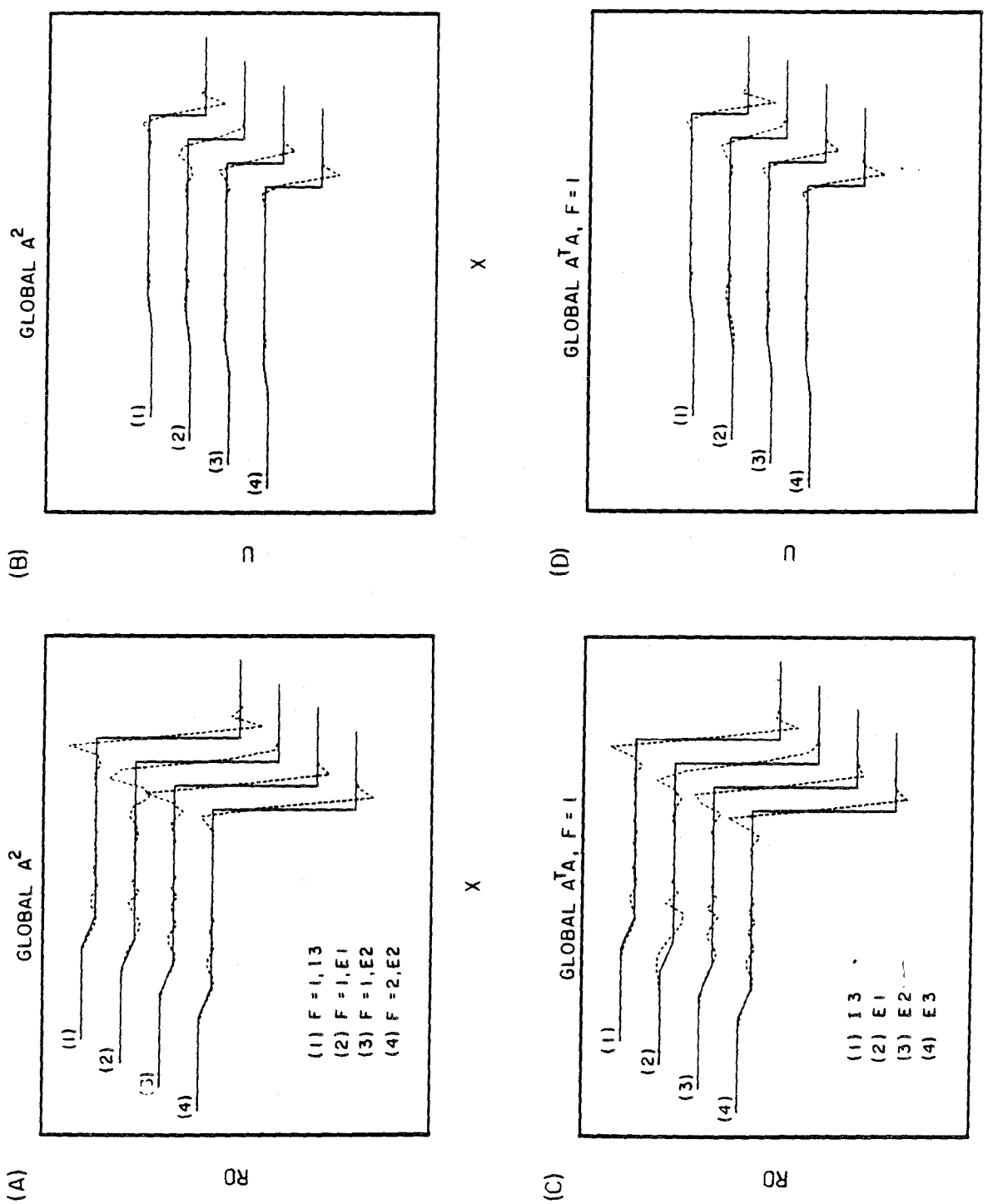
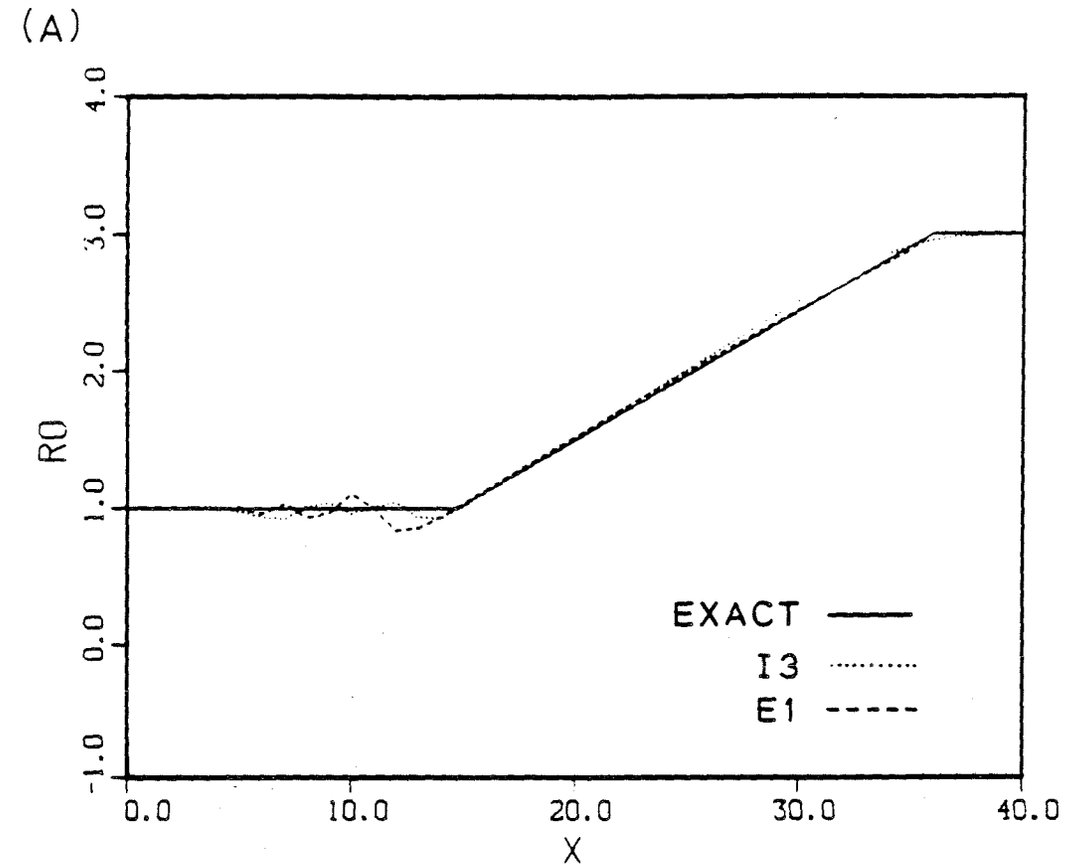


Figure 4.8 Barotropic compressible flow, stable shock propagation: $n_{e\ell} = 40$, $\Delta t = .3$, $n_{ts} = 50$, Courant number (based on maximum spectral radius) = 0.5. Comparison of various SU/PG schemes.



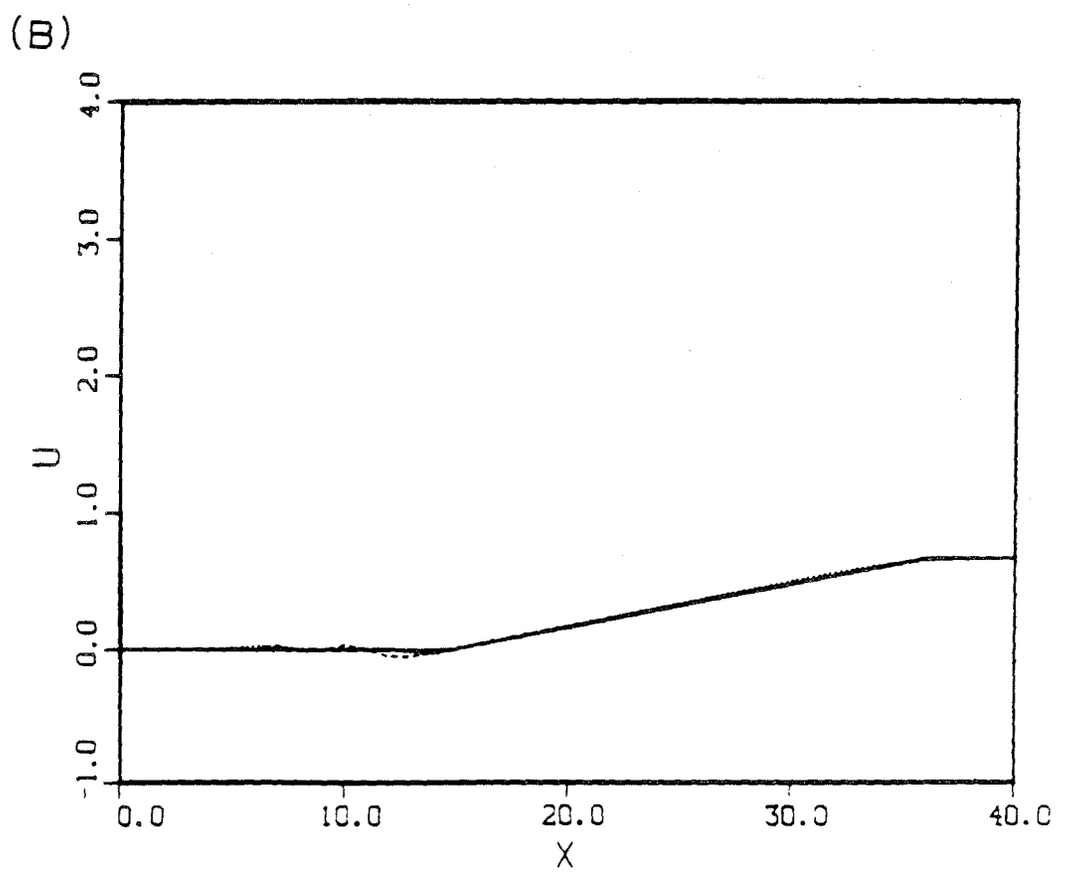


Figure 4.9 Barotropic compressible flow, rarefaction wave: global τ , A^2-form, $n_{el} = 40$, $\Delta t = .6$, $n_{ts} = 25$. Courant number (based on maximum spectral radius) = 1.0 .

produced slightly more oscillations behind the simple wave. All algorithms produced good results for this case.

4.4 Numerical Applications in the Nonlinear Steady Case

4.4.1 Isothermal Flow in a Nozzle

We consider the one-dimensional isothermal flow in a nozzle with cross-sectional area varying along the axis. The governing balance law equations, provided by Lomax et al. [L7], possesses the following conservation variable, flux and source vectors:

$$\underset{\sim}{U} = \rho A \begin{Bmatrix} 1 \\ u \end{Bmatrix} \tag{4.4.1}$$

$$\underset{\sim}{\mathscr{F}} = A \begin{Bmatrix} \rho u \\ \rho u^2 + \rho c^2 \end{Bmatrix} \tag{4.4.2}$$

$$G = \begin{Bmatrix} 0 \\ -\rho c^2 A_{,x} \end{Bmatrix} \tag{4.4.3}$$

where the acoustic speed c^2 is constant and the cross-sectional area is

$$A = A(x) \tag{4.4.4}$$

The jacobian matrices are:

$$\underset{\sim}{A} = \partial \underset{\sim}{\mathscr{F}} / \partial \underset{\sim}{U} = \begin{bmatrix} 0 & 1 \\ -u^2 + c^2 & 2u \end{bmatrix} \tag{4.4.5}$$

$$\partial \underset{\sim}{G}/\partial \underset{\sim}{U} = \begin{bmatrix} 0 & 0 \\ -c^2 A_{,x}/A & 0 \end{bmatrix} \tag{4.4.6}$$

The eigenvalues of $\underset{\sim}{A}$ are

$$\lambda_{1,2} = u \pm c \tag{4.4.7}$$

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

$$[\rho\ u] = 0 \tag{4.4.8}$$

$$[\rho u^2 + \rho c^2] = 0 \tag{4.4.9}$$

4.4.2 Boundary-value Problem

We studied steady flows suggested by Lomax et al. [L7]. The cross-sectional area and the acoustical speed were given by

$$A(x) = 1.0 + \frac{(x - 2.5)^2}{12.5} \qquad 0 \leq x \leq 5. \qquad (4.4.16)$$

$$c = 1.0 \qquad (4.4.17)$$

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. [L7].

4.4.3 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 outflow

Transient Introduction of the Source Term

In these problems, we introduced the source term into the equation system in a transient fashion. That is, instead of having the full value of the term $A_{,x}$ right at the beginning, we let it reach its full value gradually. This is done by taking $A_{,x}$ as a linear function of time during

an initial time interval at the end of which $A_{,x}$ reaches its full value. The numerical $A_{,x}$ can be expressed as follows:

$$\frac{(A_{,x})_{\text{NUMERICAL}}}{A_{,x}} = \begin{cases} n/n_{ti} & n < n_{ti} \\ 1 & n \geq n_{ti} \end{cases} \tag{4.4.18}$$

where n_{ti} denotes the number of time steps marking the end of 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 of 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 parameters γ and α to unity and employed implicit schemes with 2 iterations.

The parameter τ was chosen according to both spatial and temporal criteria given by (2.5.1) and (2.5.8), respectively.

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 Δt_{CR} as the time step for which the Courant number, based on the maximum spectral radius, is unity. That is

$$\Delta t_{CR} = h/\max \rho(\underset{\sim}{A}) \tag{4.4.19}$$

where $\rho(\underset{\sim}{A})$ is the spectral radius of $\underset{\sim}{A}$.

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 $F = 1$.

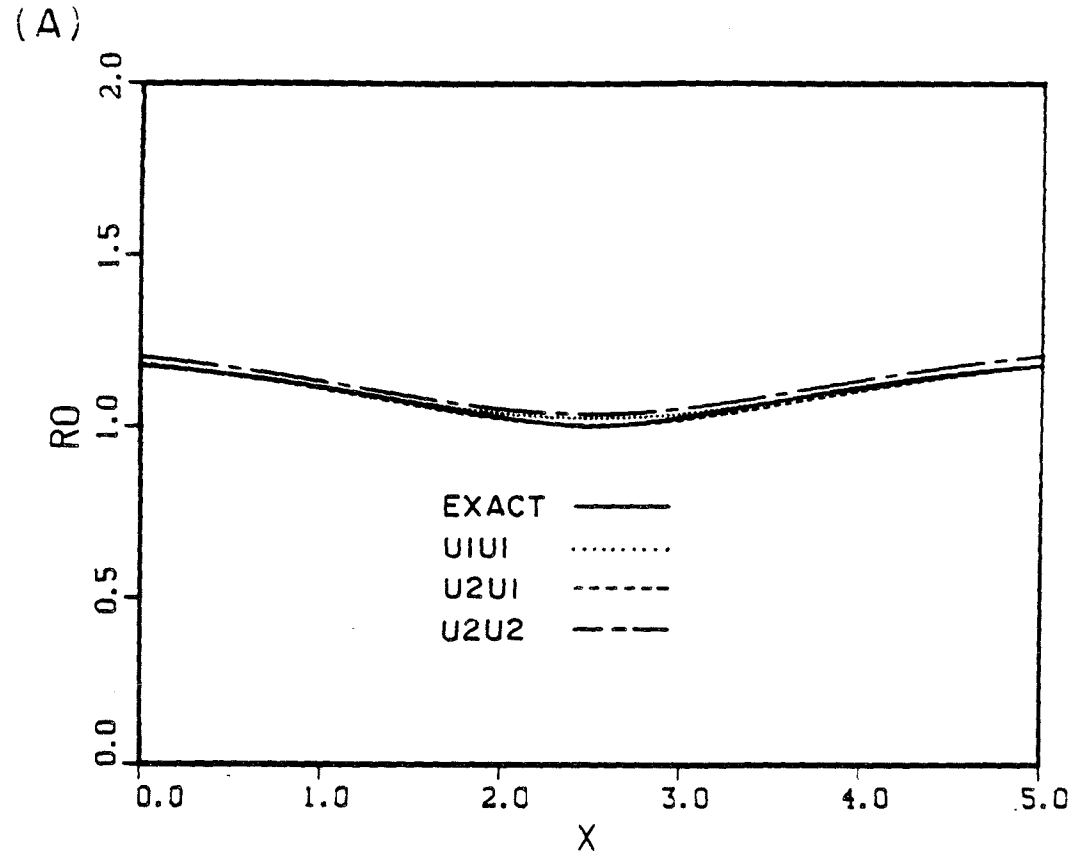
For each boundary-condition type, there was no difference between the solutions produced by different algorithms. However, the solutions differed slightly from one boundary condition type to another. For all types, the agreement with the exact solution was very close. Fig. 4.10 shows the results.

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. 4.11. 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 numerical solutions were in close agreement with



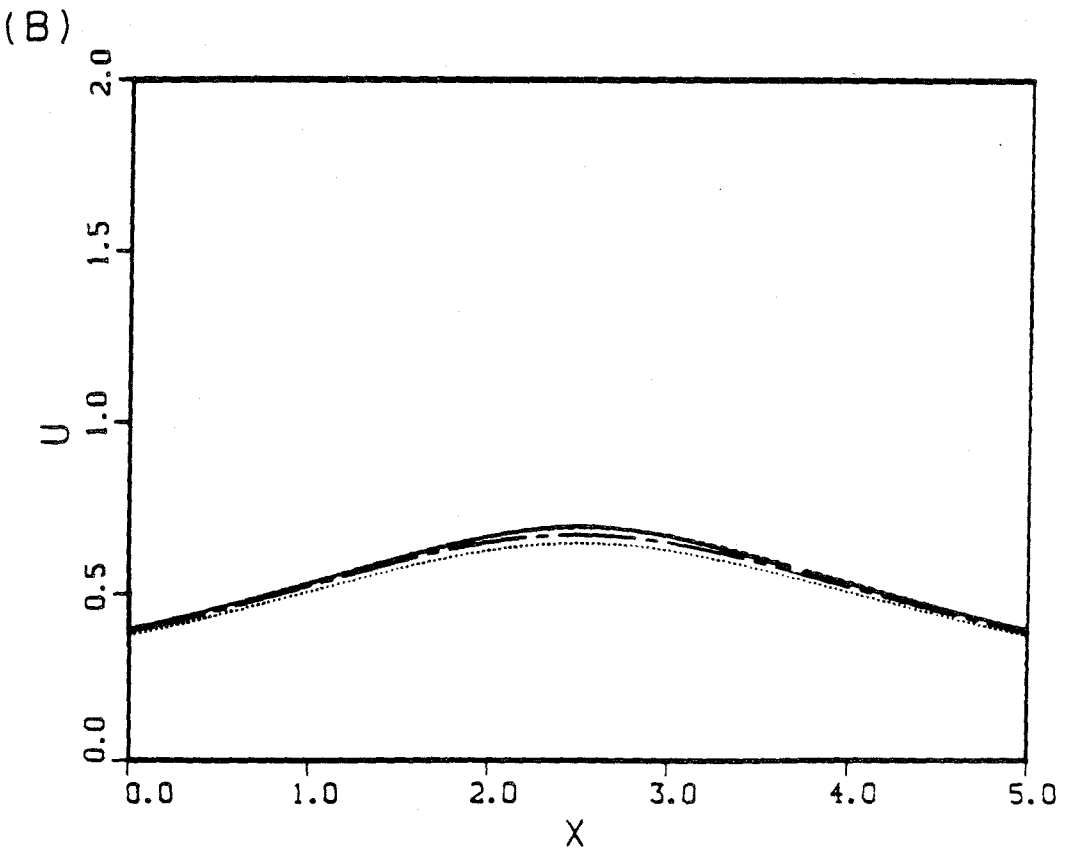
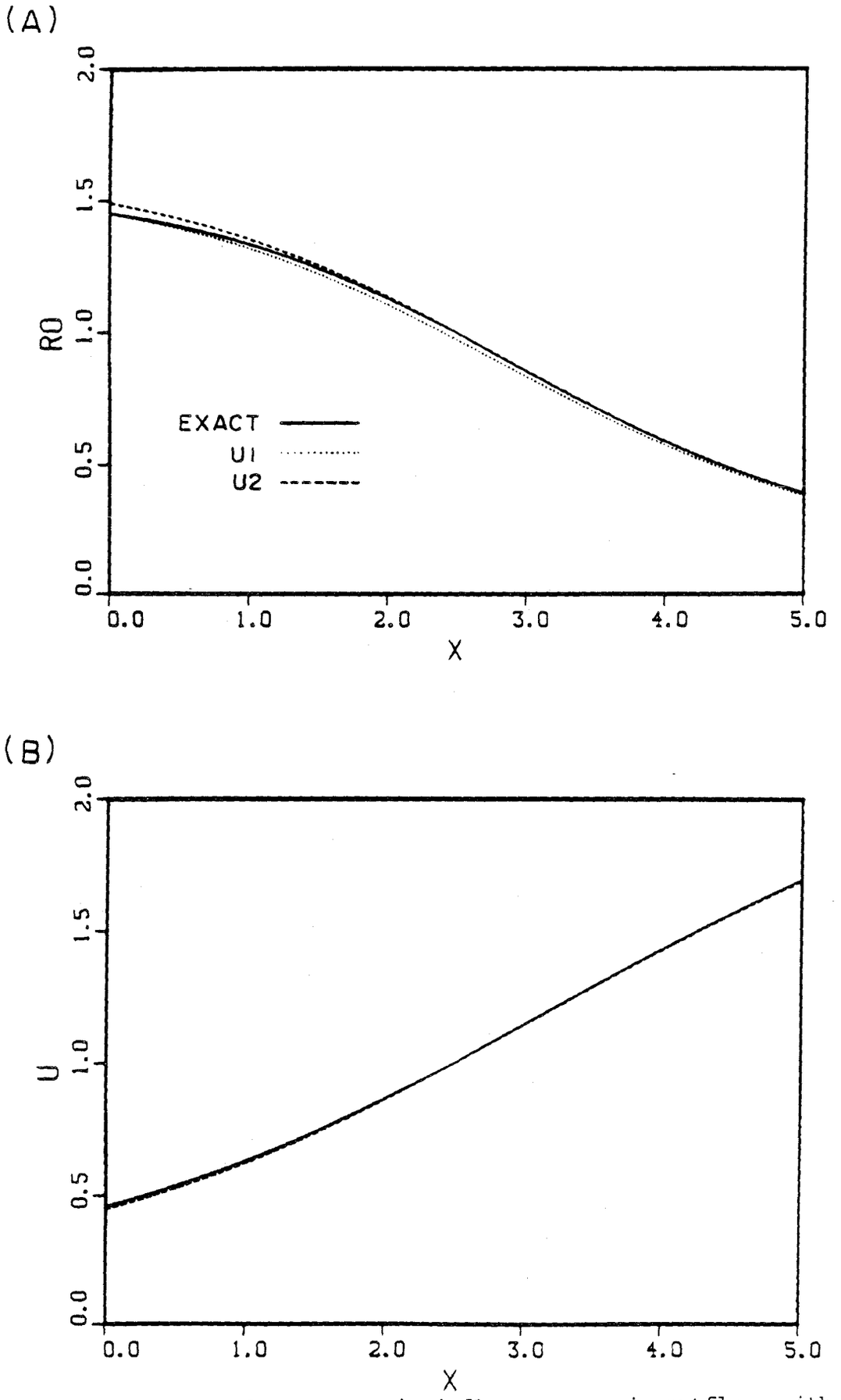


Figure 4.10 Steady nozzle flow, subsonic inflow-subsonic outflow, with no shock: $n_{e\ell} = 40$, $\Delta t = 0.735$. Comparison of different boundary conditions. (All methods give essentially the same results.)



gure 4.11 Steady nozzle flow, subsonic inflow-supersonic outflow, with no shock: $n_{e\ell} = 40$, $\Delta t = 0.460$. Comparison of different boundary conditions. (All methods give essentially the same results.)

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.

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. 4.12 shows the results for the A^TA 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 and shifted to the left by half an element length.

Fig. 4.13 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 very close agreement with the exact solution. There are very slight oscillations near the shock front for low F. For $F = 1$ and $F = 2$ the shock front is across one element only. The error in the shock location is about half an element length. As F increases, the shock front becomes smeared.

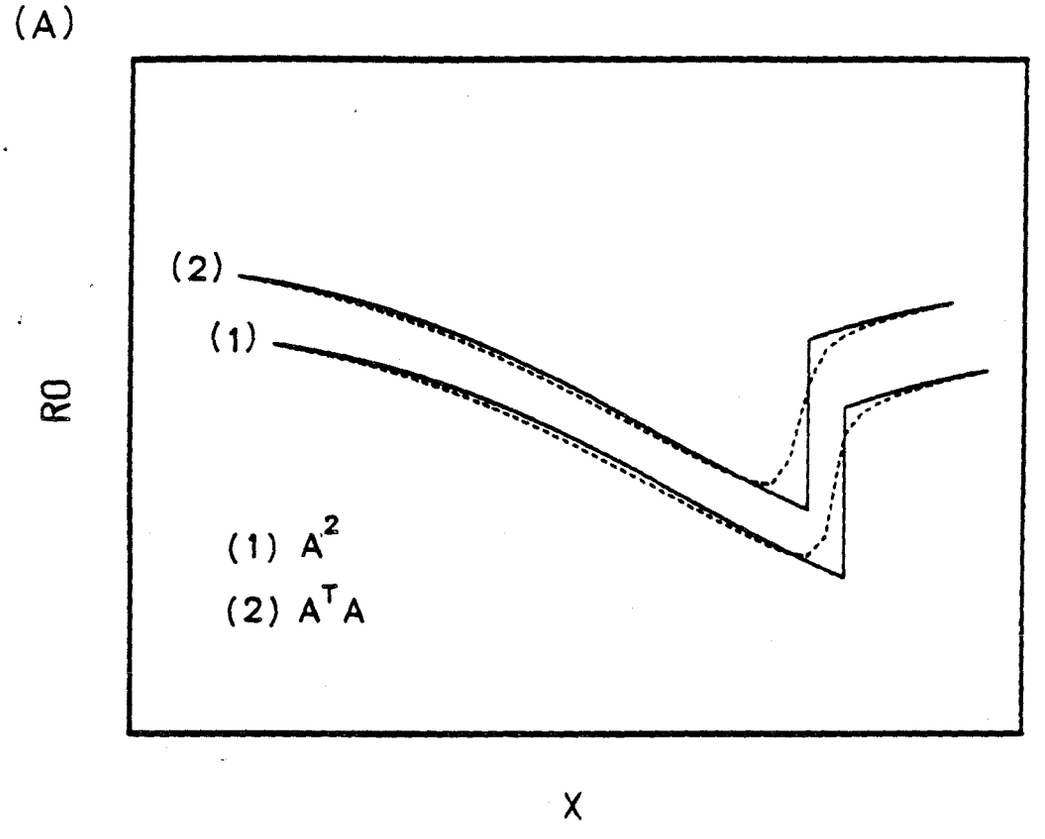
Fig. 4.14 shows the results for the A^TA-form with spatial choice of τ. The results are similar to that of Fig. 4.13. The only differences are:

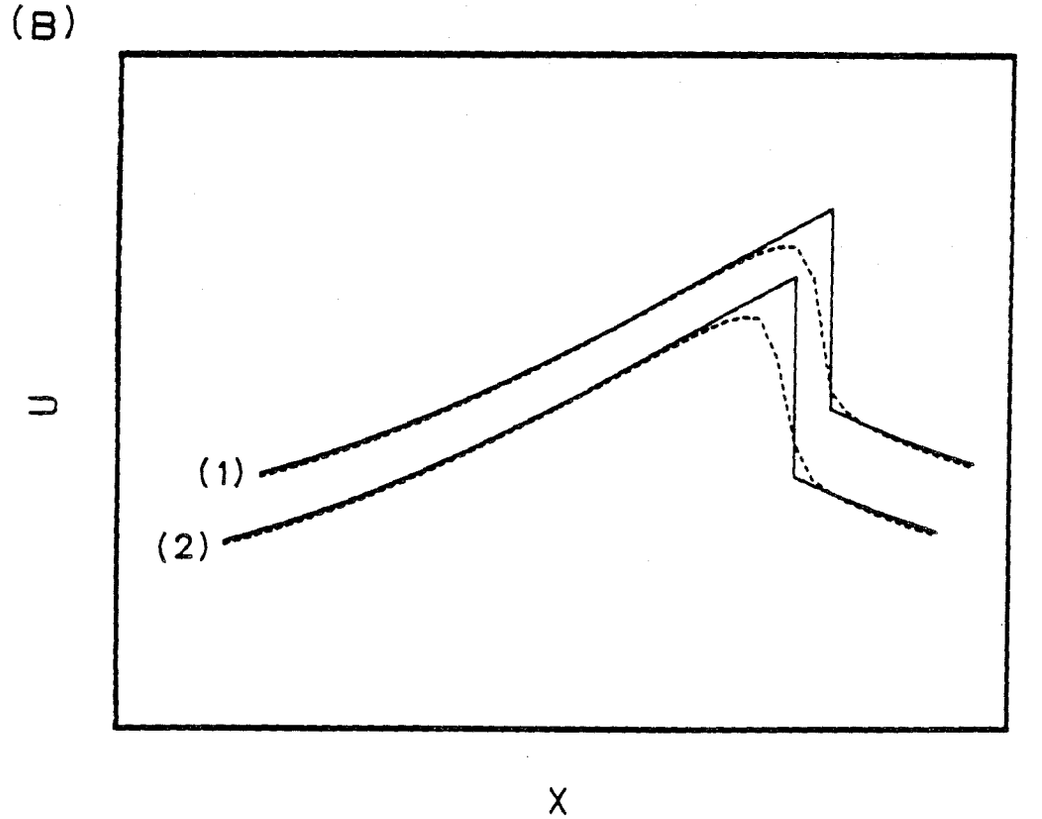
a. The shock fronts are slightly less crisp.

b. For $F = 1$, we 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 point where the flow velocity reaches the speed of sound.

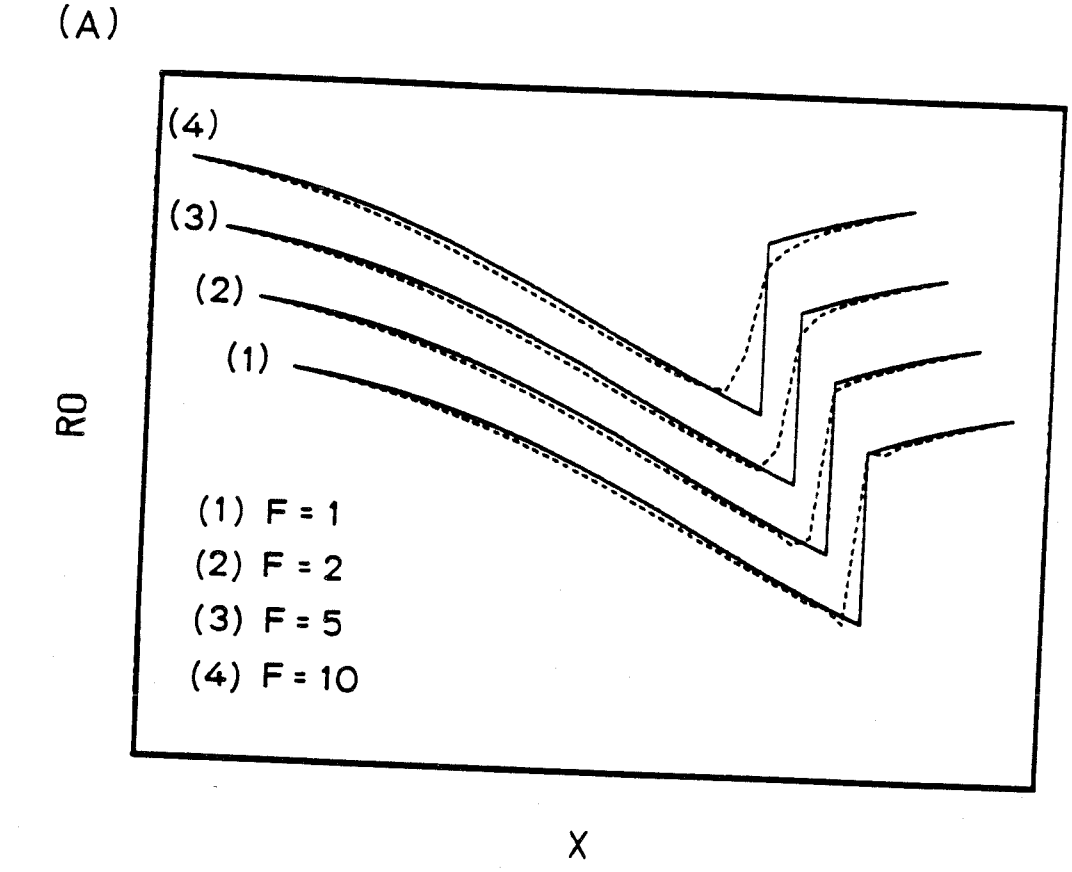
U1U2:

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





ure 4.12 Steady nozzle flow, subsonic inflow-subsonic outflow, with shock: boundary conditions U1U1, global τ, $n_{e\ell} = 40$, $\Delta t = 0.500$.



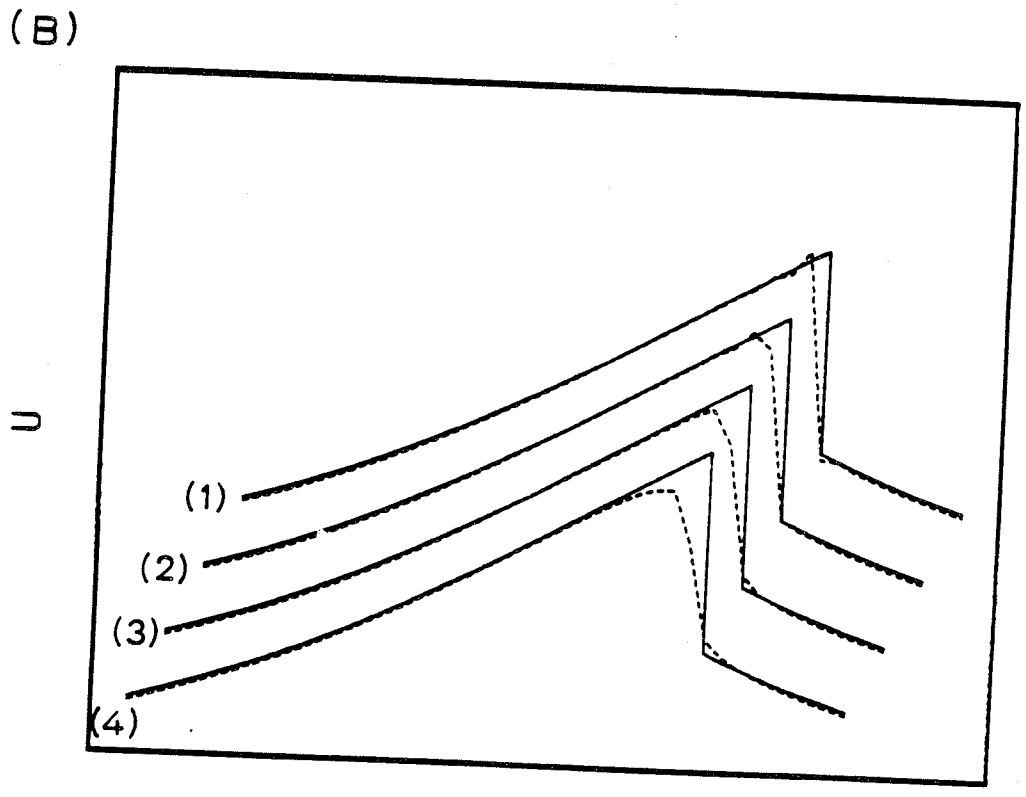
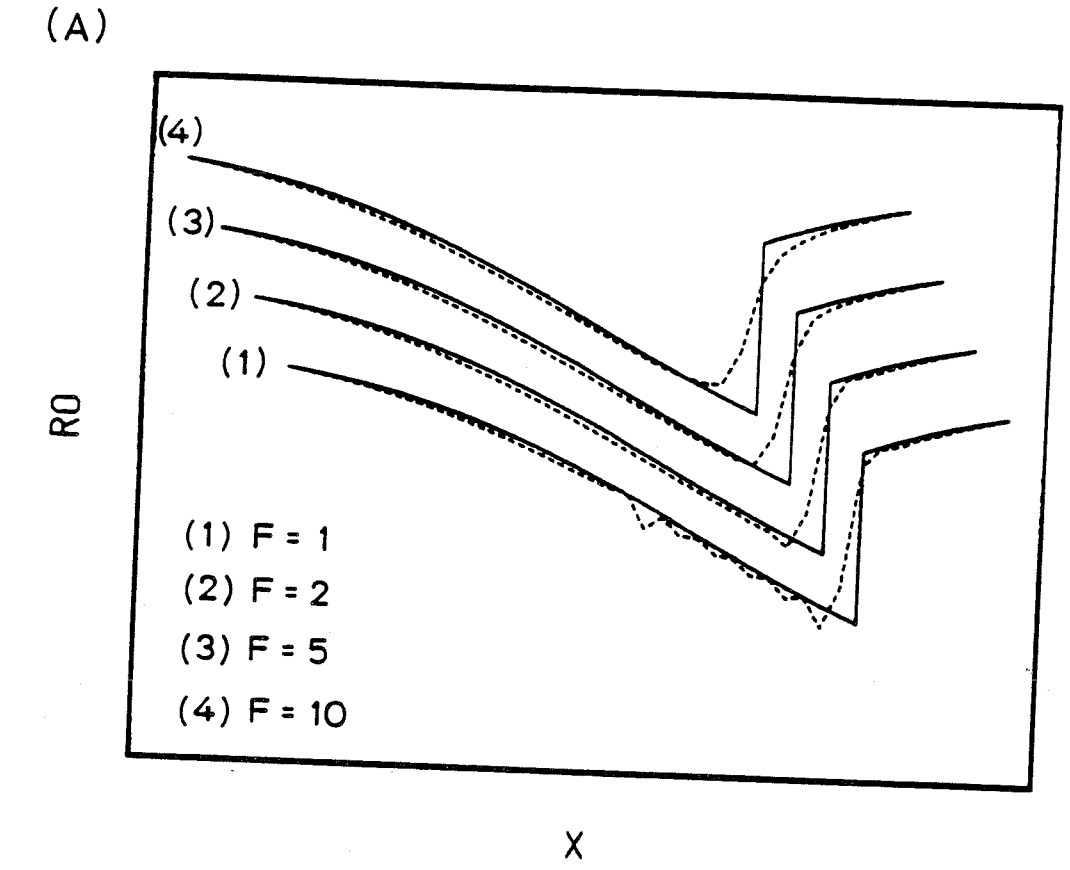
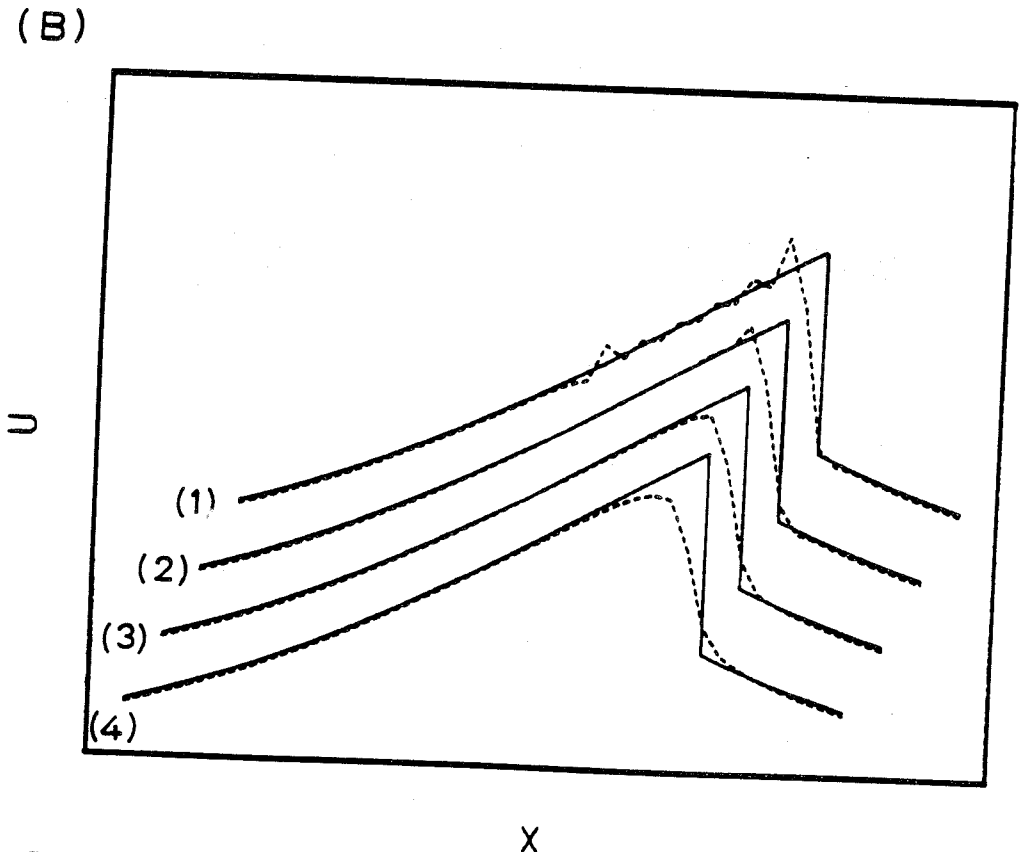


Figure 4.13 Steady nozzle flow, subsonic inflow-subsonic outflow, with shock: boundary conditions UlUl, local τ , A^2 , $n_{e\ell} = 40$, $\Delta t = 0.500$.





igure 4.14 Steady nozzle flow, subsonic inflow-subsonic outflow, with shock: boundary conditions UlUl, local τ , A^TA, $n_{e\ell} = 40$, $\Delta t = 0.500$.

U2U1:

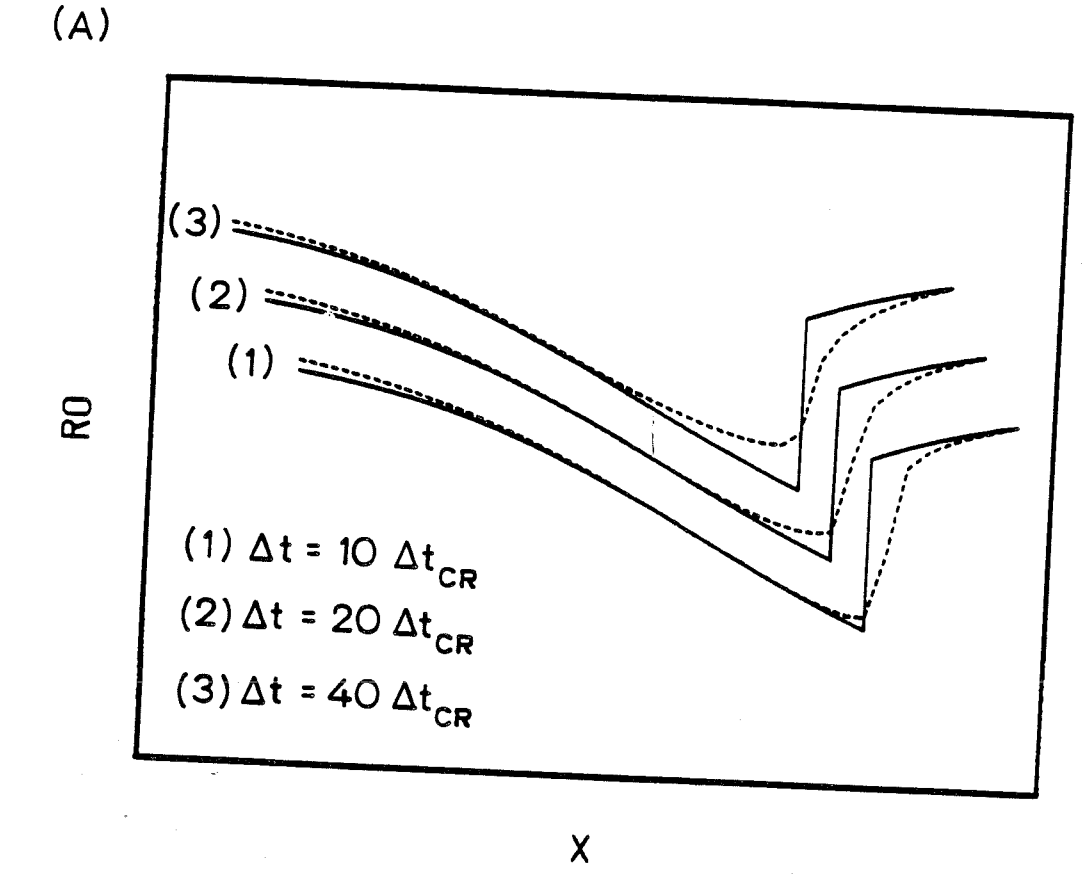
The time step was taken to be 10, 20, and 40 times the estimated critical time step. The results for A^TA and A^2 forms and temporal choice of τ are very similar and are shown in Figs. 4.15 and 4.16 respectively. The shock fronts are shifted to the right about one element length.

U2U2:

Both A^TA and A^2 with temporal choice of τ, produced smooth symmetric solutions. For this case, it was only the Galerkin algorithm which sensed the shock and located it almost at the exact location, but with severe oscillations. Fig. 4.17 shows the result produced by the Galerkin algorithm.

Remark

We observed that 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 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 several researchers (see [B6, M1, Y1]), but does not yet seem to be fully understood.



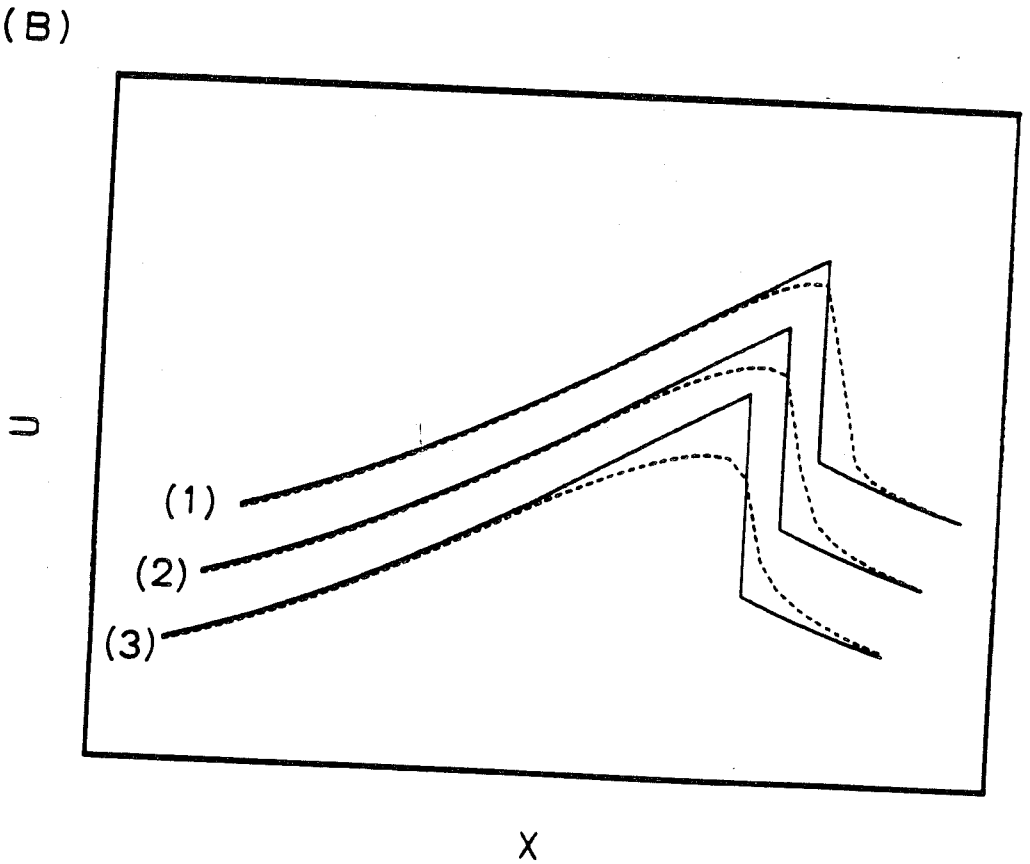
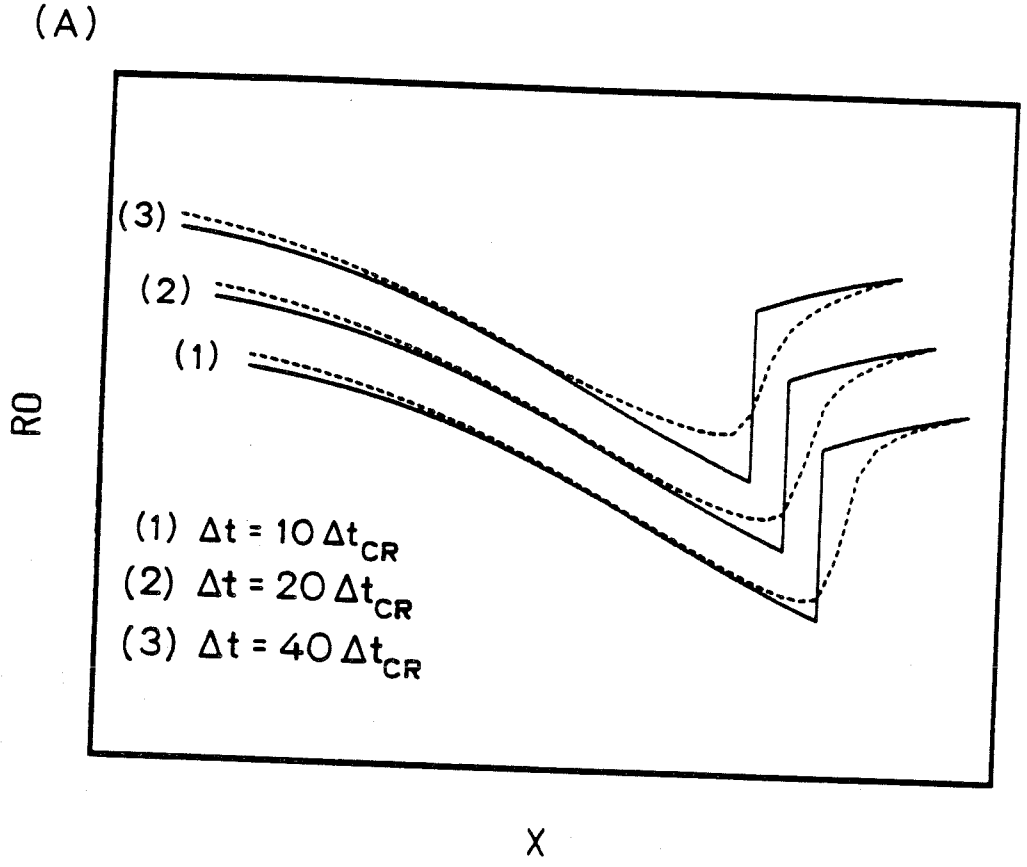


Figure 4.15 Steady nozzle flow, subsonic inflow-subsonic outflow, with shock: boundary conditions U2U1, global τ , A^2, $n_{e\ell}$ = 40.



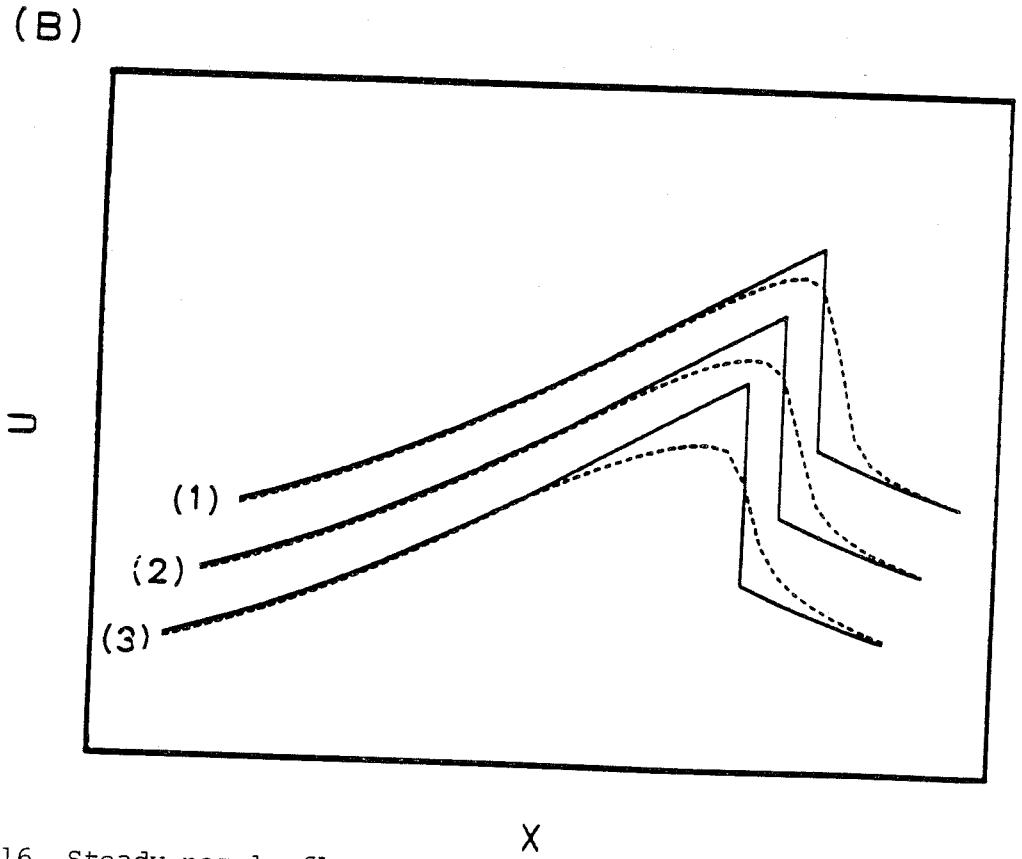


Figure 4.16 Steady nozzle flow, subsonic inflow-subsonic outflow, with shock: boundary conditions U2U1, global τ , A^TA, $n_{e\ell} = 40$.

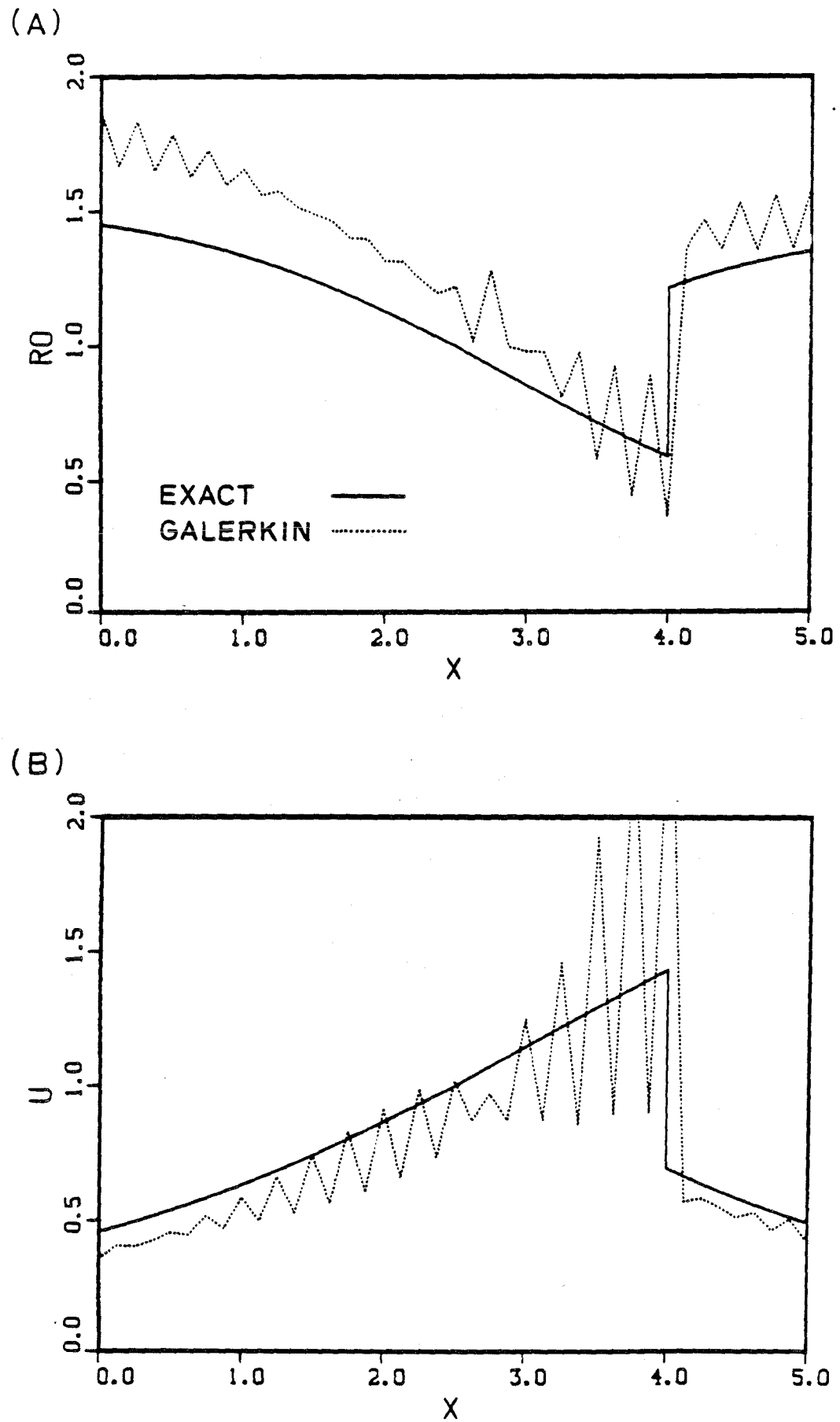


Figure 4.17 Steady nozzle flow, subsonic inflow-subsonic outflow, with shock: boundary conditions U2U2, $n_{el} = 40$, $\Delta t = 0.500$.

CHAPTER 5

Multi-dimensional Hyperbolic Systems

In this chapter the presentation of Chapter 2 is generalized to the multi-dimensional case.

5.1 Initial/Boundary-value Problem

Let Ω be an open region of $\mathbb{R}^{n_{sd}}$, where n_{sd} is the number of space dimensions. The boundary of Ω is denoted by Γ. Spatial coordinates are denoted be $\underset{\sim}{x} = \{x_j\}$.

Consider the following system of m partial differential equations:

$$\underset{\sim}{U}_{,t} + \underset{\sim}{A}_j \underset{\sim}{U}_{,j} + \underset{\sim}{G} = \underset{\sim}{0} \tag{5.1.1}$$

where

$$\underset{\sim}{U} = \underset{\sim}{U}(\underset{\sim}{x}, t) \tag{5.1.2}$$

$$\underset{\sim}{A}_j = \underset{\sim}{A}_j(\underset{\sim}{U}, \underset{\sim}{x}, t) \qquad 1 \le j \le n_{sd} \tag{5.1.3}$$

$$\underset{\sim}{G} = \underset{\sim}{G}(\underset{\sim}{U}, \underset{\sim}{x}, t) \tag{5.1.4}$$

$$\underset{\sim}{U}_{,j} = \partial \underset{\sim}{U}/\partial x_j \tag{5.1.5}$$

$$\underset{\sim}{A}_j \underset{\sim}{U}_{,j} = \sum_{j=1}^{n_{sd}} \underset{\sim}{A}_j \underset{\sim}{U}_{,j} \qquad \text{(summation convention)} \tag{5.1.6}$$

Eq. (5.1.1) is the multi-dimensional analog of (2.1.1).

Eq. (5.1.1) is said to be hyperbolic if for each $\underset{\sim}{k} = \{k_i\} \in \mathbb{R}^{n_{sd}}$ there exists a transformation matrix $\underset{\sim}{S}$ such that

$$\underset{\sim}{S}^{-1}(k_j \underset{\sim}{A}_j)\underset{\sim}{S} = \underset{\sim}{\Lambda} \tag{5.1.7}$$

where $\underset{\sim}{\Lambda}$ is a real, diagonal matrix.

Eq. (5.1.1) is called a balance law if there exist vectors $\underset{\sim}{\mathcal{F}}_j$ such that

$$\underset{\sim}{A}_j = \partial \underset{\sim}{\mathcal{F}}_j / \partial \underset{\sim}{U} \qquad 1 \leq j \leq n_{sd} \tag{5.1.8}$$

If, in addition to (5.1.8), we have that $\underset{\sim}{G} = \underset{\sim}{0}$, (5.1.1) is called a conservation law.

(5.1.1) is called a symmetric hyperbolic system if

$$\underset{\sim}{A}_j = \underset{\sim}{A}_j^T \qquad 1 \leq j \leq n_{sd} \tag{5.1.8}$$

The initial condition for (5.1.1) is

$$\underset{\sim}{U}(\underset{\sim}{x}, 0) = \underset{\sim}{U}_0(\underset{\sim}{x}), \quad \underset{\sim}{x} \in \Omega \tag{5.1.9}$$

and boundary conditions are assumed to take the abstract form (2.1.10).

5.2 Weighted Residual Formulation

In the present case the weighted residual formulation is given by

$$0 = \int_\Omega \underset{\sim}{\tilde{W}} \cdot (\underset{\sim}{U}_{,t} + \underset{\sim}{A}_j \underset{\sim}{U}_{,j} + \underset{\sim}{G}) d\Omega \tag{5.2.1}$$

where $\tilde{\underset{\sim}{W}}$ is typically assumed to have the following form:

$$\tilde{\underset{\sim}{W}} = \underset{\sim}{W} + \underset{\sim i}{T} \underset{\sim}{W}_{,i} \tag{5.2.2}$$

and

$$\underset{\sim i}{T} = \tau_i \underset{\sim i}{A} \qquad \text{(no sum)} \tag{5.2.3}$$

or

$$\underset{\sim i}{T} = \tau_i \underset{\sim i}{A}^T \qquad \text{(no sum)} \tag{5.2.4}$$

Eqs. (5.2.1) and (5.2.2) are the multi-dimensional analogs of (2.2.3) and (2.2.4), respectively.

5.3 Semi-discrete Equations

The semi-discrete equations of Section 2.3 remain in force except for the definitions of the element arrays $\underset{\sim}{m}^e_{ab}$ and $\underset{\sim}{c}^e_{ab}$ which need to be redefined as follows (cf. (2.3.5) and (2.3.8), resp.):

$$\underset{\sim}{m}^e_{ab} = \int_{\Omega^e} (N_a \underset{\sim}{I} + N_{a,i} \underset{\sim i}{T}^T) N_b \, d\Omega \tag{5.3.1}$$

$$\underset{\sim}{c}^e_{ab} = \int_{\Omega^e} (N_a \underset{\sim}{I} + N_{a,i} \underset{\sim i}{T}^T) \underset{\sim j}{A} N_{b,j} \, d\Omega \tag{5.3.2}$$

5.4 Transient Algorithms

The transient algorithms of Section 2.4 also pertain to the present case. The only change necessary is to the definition of the element array $\underset{\sim}{h}^e_{ab}$ which now takes the form (cf. (2.4.27)):

$$\underset{\sim}{h}^e_{ab} = \int_{\Omega^e} (N_a \underset{\sim}{I} + N_{a,i} \underset{\sim}{T}_i^T) \frac{\partial \underset{\sim}{G}}{\partial \underset{\sim}{U}} N_b \, d\Omega \tag{5.4.1}$$

5.5 Selection of τ_i

spatial criteria

We consider two multi-dimensional generalizations of the local criterion, (2.5.1):

$$\tau_i = F \alpha \, h/\rho \qquad 1 \leq i \leq n_{sd} \tag{5.5.1}$$

and

$$\tau_i = F \alpha \, h_i/\rho_i \qquad \text{(no sum)} \qquad 1 \leq i \leq n_{sd} \tag{5.5.2}$$

where ρ_i is the spectral radius of $\underset{\sim}{A}_i$, that is,

$$\rho_i = \max_{1 \leq j \leq m} |\lambda_j(\underset{\sim}{A}_i)| \tag{5.5.3}$$

and

$$\rho = || \underset{\sim}{\rho} || = (\rho_i \, \rho_i)^{\frac{1}{2}} \tag{5.5.4}$$

$$h = h_i \, \rho_i/\rho \tag{5.5.5}$$

$$h_i = 2 \, ||\underset{\sim}{\nabla} x_i|| \tag{5.5.6}$$

Eq. (5.5.6) holds for isoparametric mappings. The gradient operator, $\underset{\sim}{\nabla}$, is taken in terms of the natural Cartesian coordinates of the bi-unit n_{sd}-cube. For example, (5.5.6) yields the following formulas:

$$n_{sd} = 1 : \qquad h = 2\,|\partial x/\partial \xi| \tag{5.5.7}$$

$$n_{sd} = 2 : \qquad h_i = 2\left(\left(\frac{\partial x_i}{\partial \xi}\right)^2 + \left(\frac{\partial x_i}{\partial \eta}\right)^2\right)^{1/2} \tag{5.5.8}$$

If other types of finite elements are employed, (5.5.6) needs to be suitably modified.

temporal criterion

The generalization of the global criterion, (2.5.8), is

$$\tau_i = F\,\alpha\,\Delta t \qquad 1 \leq i \leq n_{sd} \tag{5.5.9}$$

The examples which follow (2.5.8) concerning symmetric implicit operators, incremental optimality, and Lax-Wendroff type methods, may be generalized to the multi-dimension case in straightforward fashion.

Remark

The formulas presented in this section represent esthetic improvements of ones used previously [B7, B8, H12, H14, H15].

CHAPTER 6

Numerical Applications in Two Dimensions

6.1 Introduction

Problem Geometry and Governing Equations

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 the uniform flow (non-lifting case). Fig. 6.1(A) shows the configuration, where b denotes the ratio of the maximum airfoil thickness to the cord length. The notations used for flow variables are defined in appendix I. The subscript " ∞ " refers to the free stream.

Since we know that the solution will be symmetric with respect to the x_1 - axis, we need only consider the half plane $x_2 \geq 0$ as our problem domain. The parabolic arc bounding the airfoil in this plane is described by the following expression:

$$x_2 = \frac{b}{2}\left[1 - (2\,x_1)^2\right] \tag{6.1.1}$$

The governing equations are the Euler equations described in section I.1 and I.3. The ratio of the specific heats is

$$\gamma = 1.4 \tag{6.1.2}$$

(The reader should not be confused by the use of γ for the transient algorithm parameter of chapter 2 and the present usage.)

The eigenvalues of the coefficient matrices $\underset{\sim}{A}_1$ and $\underset{\sim}{A}_2$ can be obtained from (I.3.8) by setting $(k_1, k_2) = (1, 0)$ and $(k_1, k_2) = 0.1$ respectively.

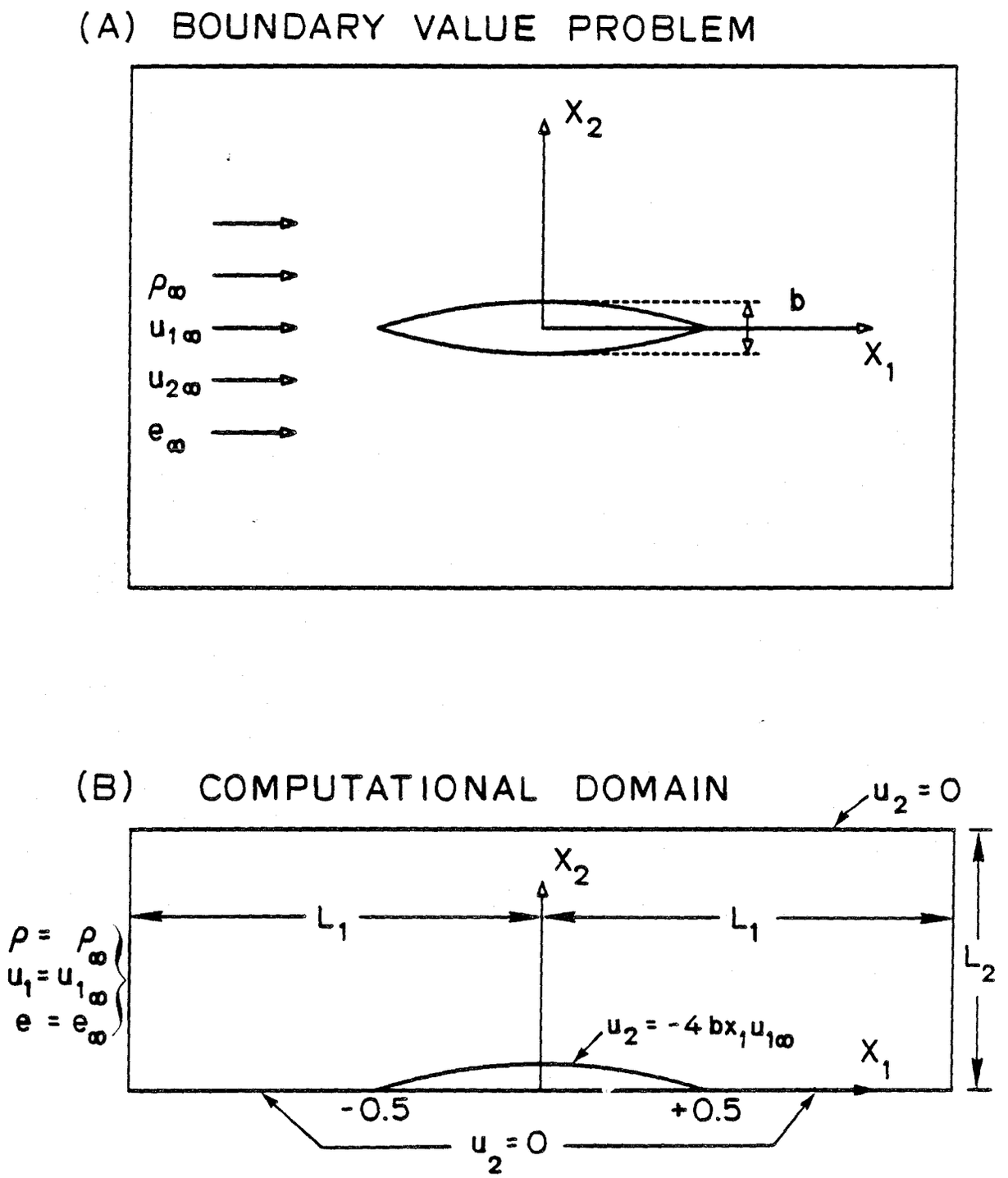


Figure 6.1 Boundary value problem and computational domain for thin parabolic are airfoil.

Boundary Conditions

The free stream parameters are taken to be

$$\rho_\infty = 1. \quad , \quad u_{2\infty} = 0. \quad , \quad e_\infty = 1. \tag{6.1.3}$$

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_∞ :

$$u_{1\infty}^2 = \frac{M_\infty^2 \gamma(\gamma - 1) e_\infty}{M_\infty^2 \gamma(\gamma - 1)/2 + 1} \tag{6.1.4}$$

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

$$u_2 = 0 \quad , \quad x_2 = 0 \ , \ |x_1| > .5 \tag{6.1.5}$$

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

$$\frac{u_2}{u_1} = \frac{dx_2}{dx_1} \tag{6.1.6}$$

Assuming that the airfoil thickness is small enough, such that the uniform flow field is perturbed only slightly, u_1 can be approximated in (6.1.6) by its free stream value (see [L4]):

$$\frac{u_2}{u_{1\infty}} = \frac{dx_2}{dx_1} \tag{6.1.7}$$

From (6.1.1) and (6.1.7), the boundary condition on the surface of the airfoil can then be expressed as

$$u_2 = -4\, b\, x_1 u_{1\infty} \qquad |x_1| \leq .5 \tag{6.1.8}$$

Nature of the Solution

Once we set the free stream parameters as given by (6.1.3) the nature of the solution depends on the free stream Mach number and the airfoil thickness ratio. We fix the thickness ratio to be

$$b = 0.10 \tag{6.1.9}$$

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 $x_1 = .3$.

6.2 Computational Domain and Algorithmic Features

Finite Element Mesh and Boundary Conditions

The computational domain is shown in Fig. 6.1(B). We utilize three finite element meshes with different overall sizes: the medium and fine meshes with $L_1 = L_2 = 3.5$, and the coarse mesh with $L_1 = L_2 = 2.0$. 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. 6.2; 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. At the upper boundary, we impose the condition $u_2 = 0$, which can physically be interpreted as a channel wall. Along the x_1 - axis we take the boundary conditions of (6.1.5) and (6.1.8). Imposing the boundary condition of (6.1.8) along the x_1 - axis instead of on the airfoil surface is a standard thin-airfoil approximation.

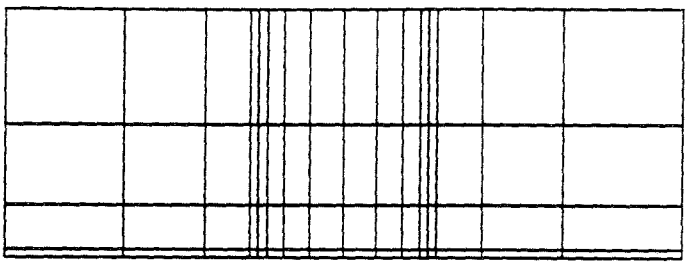
Other Algorithmic Features

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

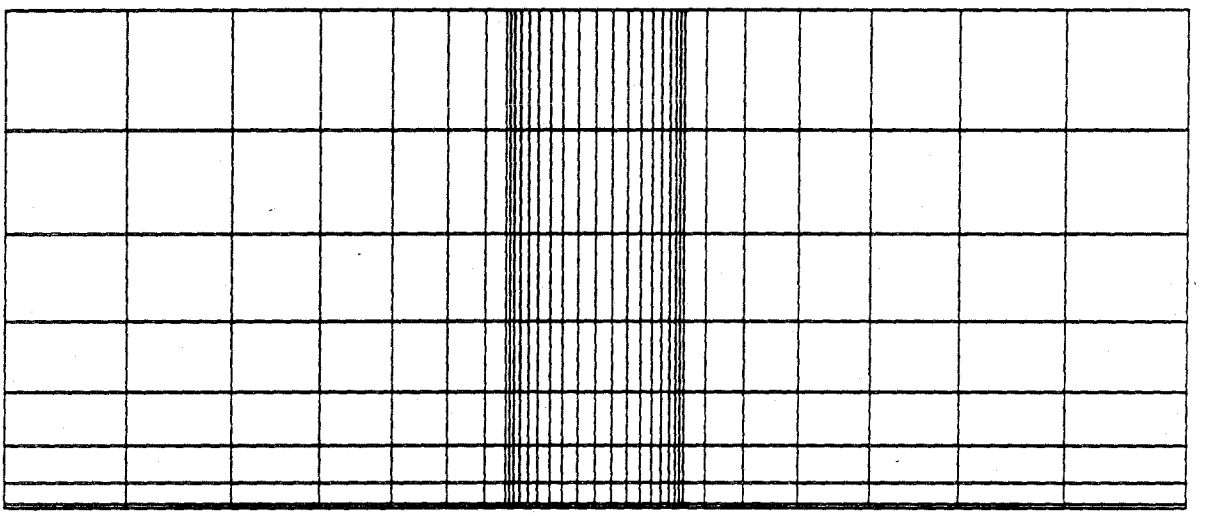
We used the temporal definition of the parameter τ_i with $F = 1$ and the A^2-form. For the subcritical case we also employed the A^TA-form.

The transient algorithm parameters γ and α were both set to unity.

COARSE MESH



MEDIUM MESH



FINE MESH

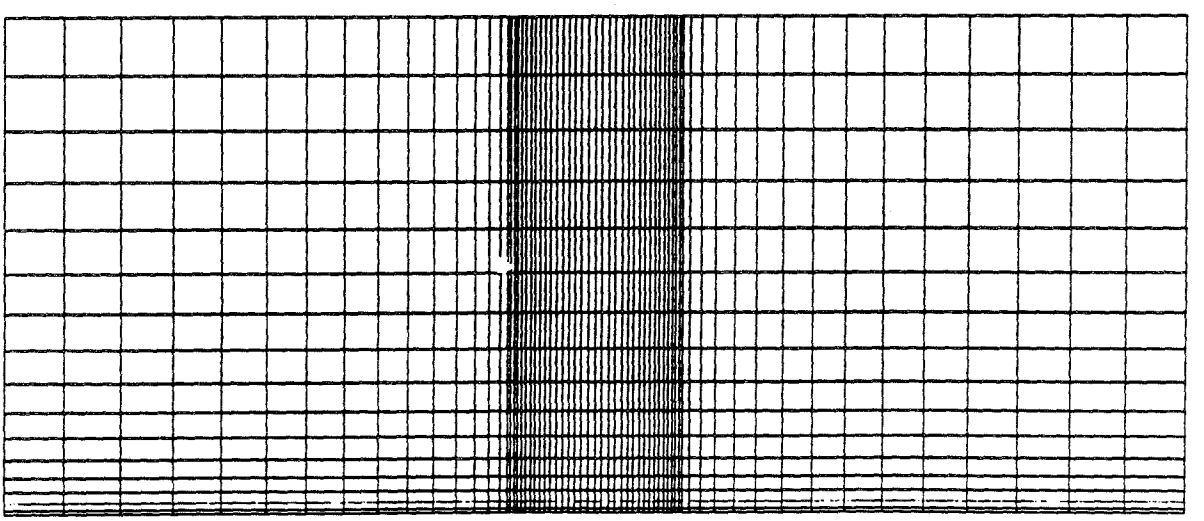


Figure 6.2 Finite element meshes. Coarse mesh, 64 elements; medium mesh, 256 elements; fine mesh, 1016 elements.

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. Similar to the definition of (4.4.19), the critical time step is defined here as:

$$\Delta t_{CR} = \min_{e,j} \left(h_j^e / \rho(\underset{\sim}{A}_j) \right) \qquad \text{(no sum)} \tag{6.2.1}$$

where the subscript j is the space dimension and the superscript e is the element number. Since $\rho(\underset{\sim}{A}_j)$ is unknown prior to execution, for the estimation of Δt_{CR} we use the free stream value of $\rho(\underset{\sim}{A}_j)$.

The steady boundary condition of (6.1.8) was implemented in the same way as for the nozzle problems of chapter 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.

6.3 Subcritical Case

We compare our results, at free stream Mach number .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'):

$$\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} u_{1_\infty} \\ 0 \end{Bmatrix} + \begin{Bmatrix} u_1' \\ u_2' \end{Bmatrix} \tag{6.3.1}$$

the analytical solution (see [L5]) can be written as:

$$\beta u_1' - i u_2' = u_{1_\infty} b \frac{4}{\pi}\left(1 - Z \ln\left(\frac{Z + 0.5}{Z - 0.5}\right)\right) \tag{6.3.2}$$

where $Z = x_1 + i\beta x_2$, $i = (-1)^{\frac{1}{2}}$ and

$$\beta = (1 - M_\infty)^{\frac{1}{2}} \tag{6.3.3}$$

One can observe from (6.3.2) that, along the x_1 - axis, the boundary condition for u_2 has been satisfied, while, for u_1' , the following expression is obtained:

$$u_1' = \frac{1}{\beta} u_{1_\infty} b \frac{4}{\pi}\left(1 - x_1 \ln\left|\frac{x_1 + 0.5}{x_1 - 0.5}\right|\right) \tag{6.3.4}$$

The pressure coefficient C_p , defined as

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty (u_{1_\infty})^2} \tag{6.3.5}$$

can be obtained by the following equation (see [L4]):

$$C_p = -\left[2\left(\frac{u_1'}{u_{1_\infty}}\right) + \left(\beta \frac{u_1'}{u_{1_\infty}}\right)^2 + \left(\frac{u_2'}{u_{1_\infty}}\right)^2\right] \tag{6.3.6}$$

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

$$C_p = -2\left(\frac{u_1'}{u_{1_\infty}}\right) \tag{6.3.7}$$

Finite Element Solutions

Fig. 6.3 shows the analytical and finite element (medium and coarse mesh) solutions for subcritical flow at $M_\infty = .5$.

The time steps for the regular 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

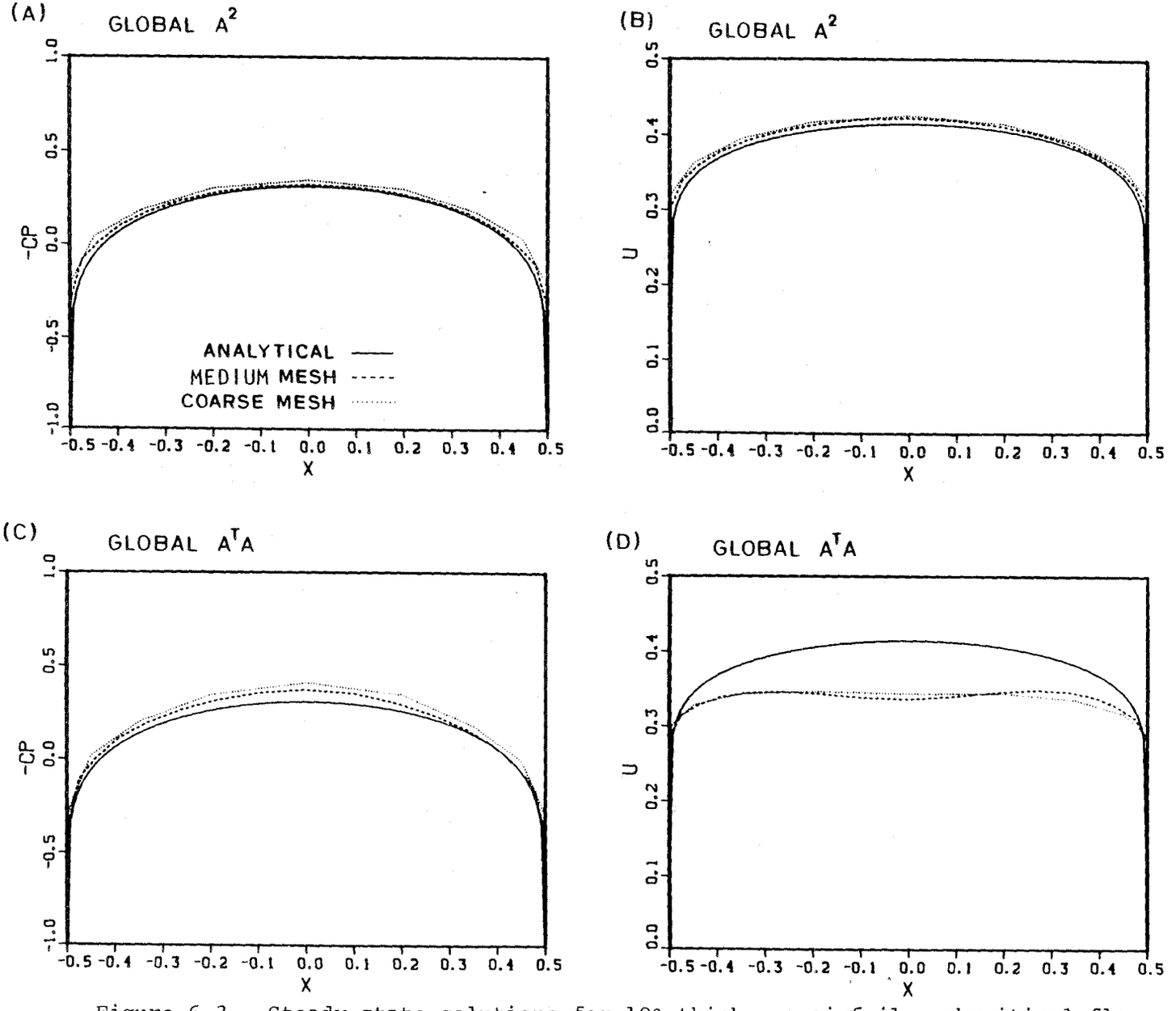


Figure 6.3. Steady state solutions for 10% thickness airfoil, subcritical flow, M_∞ = .50: Δt (medium mesh) = 0.23, Δt (coarse mesh) = 0.46.

20 steps.

The finite element solutions for A^2 and A^TA-forms are shown in Figs. 6.3(A), (B) and 6.3(C), (D) respectively.

The values of C_p(CP) and u_1(U) are plotted along the airfoil.

The numerical solutions are in close agreement with the analytical solution except for the variable u_1(U) when the A^TA-form is used. This 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.

Remarks

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

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

3. It is interesting to observe that the results for the coarsest mesh are in close agreement with the analytical solution.

4. The Galerkin algorithm, on the other hand, produced highly oscillatory results which fed back into the operators of the problem and caused the results to diverge.

6.4 Supercritical Case

We compare our solutions, at free stream Mach number 0.84, to the finite-difference solution of Barton [B5] who used the flux vector splitting scheme of Steger [S2, S3].

The finite difference grid has 97 × 33 points with 75 points along the airfoil. The boundary condition on the airfoil was taken to be the same as

ours, as given by (6.1.8). However, at the outer boundaries of the computational domain, free stream boundary conditions were imposed in Barton's calculations.

Finite Element Solutions

Fig. 6.4 shows the 800-iteration solution of Barton together with the finite element solutions for coarse, medium and fine meshes.

The time steps were set to 0.40 for the coarse mesh and to 0.20 for the medium and fine meshes. For the coarse mesh, the 60-step solution, and for the medium and fine meshes, the 120-step solutions, were taken as the steady-states.

We observe that the medium and fine meshes produced very similar results and they are in agreement with the Barton solution, except for a 3-4% shift in the location of the shock front. It is known that the way the boundary conditions are imposed can change the solution considerably [B5, H5, S1]. Because we imposed different boundary conditions at the outer boundaries of the computational domain, and especially, because we imposed a "wall" boundary condition, rather than a free stream boundary condition at the upper boundary, we are not surprised that there is some difference between the results of Barton and ours. This may be explained as follows:

It is known that as the free stream Mach number gets higher, the shock front moves downstream. When we check our Mach number at the points along the upper computational boundary, we see that it is not equal to the free stream Mach number 0.84, but higher (about 0.85-0.86). This is consistent with the downstream shift of the shock front we obtain.

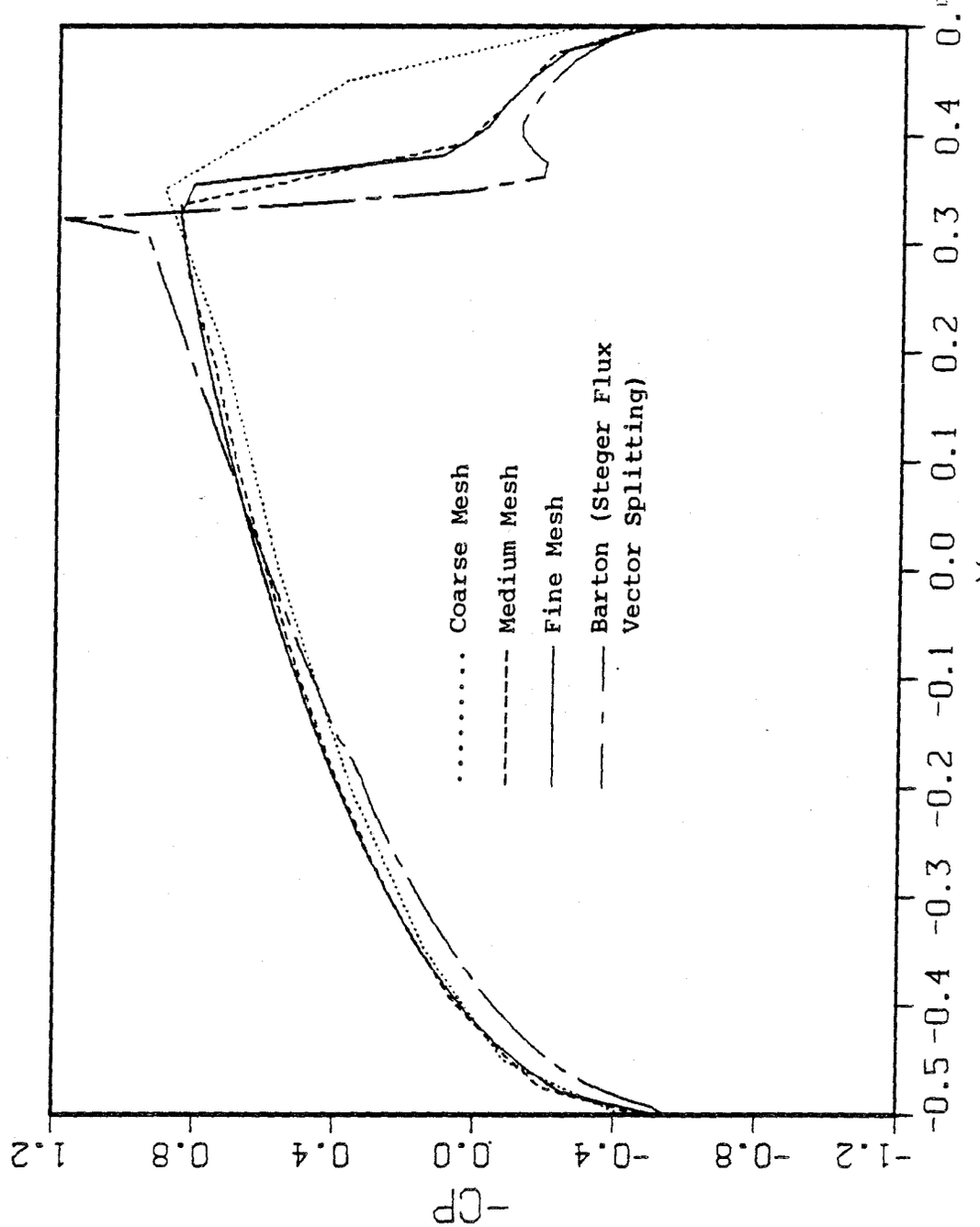


Figure 6.4 Steady state solutions for 10% thickness airfoil, supercritical flow, $M_\infty = .84$: Δt (medium and fine mesh) = 0.20, Δt(coarse mesh) = 0.40.

The coarse mesh (which is really much too coarse for this type of problem) did the best that could be expected. The peak point of the shock is about at the same location found by the medium and fine meshes. The coarse mesh, however, has only two elements between the peak and the trailing edge, and this is not enough for a distinct representation of the pressure profile to the right of the shock front.

The medium and fine finite element meshes, compared with Barton's difference grid, are also very coarse. Thus we can conclude that, even with relatively crude meshes, the finite element algorithm performed very well on this problem.

CHAPTER 7

Conclusions

In this work we presented a Petrov-Galerkin finite element algorithm for first-order hyperbolic equation systems. The immediate purpose was to solve fluid dynamics problems governed by the conservation-law form of the compressible Euler equations. Finite element algorithms, inherently, can be easily applied to problems with arbitrary geometries and boundary conditions. This permits us to utilize the algorithms developed for complicated domains that finite difference algorithms would, normally, have difficulty with.

The Petrov-Galerkin algorithm presented here is a generalization to hyperbolic systems of the streamline upwind/Petrov-Galerkin algorithm developed by Hughes and Brooks [B7, B8, H12, H14, H15].

We conducted an extensive stability and accuracy analysis on a linear model problem and observed that the algorithms suggested have desirable properties. Compared to the usual Galerkin algorithms they minimize spurious oscillations without loss of accuracy.

We tested our algorithms on several problems with governing equations in conservation law form. Particular attention was paid to the cases with shocks.

In one dimension, numerical experiments were made on linear transient, nonlinear transient, and nonlinear steady problems. We compared the numerical solutions to analytical ones and observed that they were generally in very close agreement. The algorithms handled shock fronts very satisfactorily;

the shock fronts were, for the most part, very crisp, with minimal spurious oscillations. The usual Galerkin algorithm on the other hand, was ineffective for problems with discontinuous solutions, while for problems with smooth solutions it performed satisfactorily.

In two space dimensions, we tested the algorithms on a thin biconvex airfoil problem. For both subsonic and transonic cases, the algorithms proved to be successful. For the transonic problem, the location and magnitude of the shock front was in good agreement with Barton (Steger flux-vector splitting) solution [B5, S2, S3]. The subsonic case results were in close agreement with a linear analytical solution. For the subsonic case, we obtained a good solution even with a very coarse mesh. For the transonic case, the coarse mesh solution was not as good as the medium and fine mesh solutions, yet, was qualitatively satisfactory.

Overall, the finite element algorithms suggested here performed very well for problems with smooth and discontinuous solution. The optimal selection of the time parameter, τ , which appears as a factor in the perturbation part of the weighting functions, needs further investigation. This needs to be pursued from the standpoint of nonlinearities and shocks which are, of course, prime concerns in solving the compressible Euler equations.

We believe that, with the recent advances in the development of Petrov-Galerkin algorithms, the finite element method has now become a viable alternative in computational fluid dynamics. However, the efficiency of finite element algorithms still needs to be improved, especially with respect to decreasing storage requirements. Recently, an "element-by-element" approch to the finite element formulation has been proposed by Hughes, Levit and Winget

(see [H17]). The preliminary results seem to be promising, particularly for problems with symmetric operators. Eventually, with the help of such new concepts, the finite element method can be expected to become an economically competitive and powerful analysis tool in the field of computational fluid dynamics.

APPENDIX I

The Euler Equations

I.1 General Principles

The compressible Navier-Stokes equations, with no source terms, can be written as a system of conservation equations:

$$\underset{\sim}{U}_{,t} + \underset{\sim}{\mathscr{F}}_{j,j} = \underset{\sim}{0} \quad , \quad 1 \leq j \leq n_{sd} \qquad \text{(I.1.1)}$$

where $\underset{\sim}{U}$ is the vector of conservation variables and the $\mathscr{F}_j$'s are flux vectors, which are, in general, functions of $\underset{\sim}{U}$ and its spatial derivatives:

$$\underset{\sim}{\mathscr{F}}_j = \underset{\sim}{\mathscr{F}}_j(\underset{\sim}{U}, \underset{\sim}{U}_{,k}) \qquad \text{(I.1.2)}$$

If we neglect dissipative effects (i.e. conduction, viscosity, etc.) then the flux vectors are functions of $\underset{\sim}{U}$ only:

$$\underset{\sim}{\mathscr{F}}_j = \underset{\sim}{\mathscr{F}}_j(\underset{\sim}{U}) \qquad \text{(I.1.3)}$$

and (I.1.1) is called the Euler equations (or the inviscid gas dynamics equations).

Let us write the Euler equations in a quasi-linear form:

$$\underset{\sim}{U}_{,t} + \underset{\sim}{A}_j \underset{\sim}{U}_{,j} = \underset{\sim}{0} \qquad \text{(I.1.4)}$$

where

$$\underset{\sim}{A}_j = \partial \underset{\sim}{\mathscr{F}}_j / \partial \underset{\sim}{U} \qquad \text{(I.1.5)}$$

Consider the linear combination of coefficient matrices:

$$\underset{\sim}{A} = k_j \underset{\sim}{A}_j \tag{I.1.6}$$

where $\underset{\sim}{k}$ is a real vector. Without loss of generality, we take $\underset{\sim}{k}$ to be a unit vector. The equation system of (I.1.4) is hyperbolic if for each $\underset{\sim}{k}$ there exists a non-singular transformation matrix $\underset{\sim}{S}$ such that the similarity transformation

$$\underset{\sim}{S}^{-1} \underset{\sim}{A} \underset{\sim}{S} = \underset{\sim}{\Lambda} \tag{I.1.7}$$

diagonalizes the matrix $\underset{\sim}{A}$. Here $\underset{\sim}{\Lambda}$ is a real diagonal matrix. It turns out that, for the Euler equations, this similarity transformation also symmetrizes the individual coefficient matrices simultaneously. This, in general, cannot be expected for all hyperbolic systems.

To determine the transformation matrix $\underset{\sim}{S}$ we need to go through the usual procedure of finding the eigenvalues and eigenvectors of the matrix $\underset{\sim}{A}$ (see [T1, W2, W3]). We go through this procedure in two phases. The first phase consists of a transformation into the primitive variables form:

$$\underset{\sim}{U}'_{,t} + \underset{\sim}{A}'_j \underset{\sim}{U}'_{,j} = \underset{\sim}{0} \tag{I.1.8}$$

where

$$\underset{\sim}{A}'_j = \underset{\sim}{Q}^{-1} \underset{\sim}{A}_j \underset{\sim}{Q} \tag{I.1.9}$$

$$\underset{\sim}{Q} = \partial \underset{\sim}{U} / \partial \underset{\sim}{U}' \tag{I.1.10}$$

and

$$\underset{\sim}{A}' = k_j \underset{\sim}{A}'_j \tag{I.1.11}$$

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, the operators $\underset{\sim}{R}$ and $\underset{\sim}{R}^{-1}$, which diagonalize $\underset{\sim}{A}'$, are constructed and another transformation is performed:

$$\underset{\sim}{U}''_{,t} + \underset{\sim}{A}''_j \underset{\sim}{U}''_{,j} = \underset{\sim}{0} \tag{I.1.12}$$

where

$$\underset{\sim}{A}''_j = \underset{\sim}{R}^{-1} \underset{\sim}{A}'_j \underset{\sim}{R} = (\underset{\sim}{Q}\ \underset{\sim}{R})^{-1} \underset{\sim}{A}_j (\underset{\sim}{Q}\ \underset{\sim}{R}) \tag{I.1.13}$$

$$\underset{\sim}{R} = \partial \underset{\sim}{U}'/\partial \underset{\sim}{U}'' \tag{I.1.14}$$

and

$$\underset{\sim}{A}'' = k_j \underset{\sim}{A}''_j \tag{I.1.15}$$

Here " ˜ " refers to the frame in which $\underset{\sim}{A}''$ is a diagonal matrix. The transformation matrix $\underset{\sim}{S}$ of (I.1.7) is, then, equal to the product $\underset{\sim}{Q}\ \underset{\sim}{R}$. Further, in this last frame, the individual coefficient matrices are symmetric:

$$(\underset{\sim}{A}''_j)^T = \underset{\sim}{A}''_j \tag{I.1.16}$$

In the following sections, we define all the arrays involved in three, two and one space dimensions.

I.2 Three-dimensional Case

The conservation variables vector and flux vectors are

$$\underset{\sim}{U} = \rho \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e \end{Bmatrix} \tag{I.2.1}$$

$$\underset{\sim}{\mathcal{F}}_j = \begin{Bmatrix} \mu_j \rho \\ u_j \rho u_1 + \delta_{j1} p \\ u_j \rho u_2 + \delta_{j2} p \\ u_j \rho u_3 + \delta_{j3} p \\ u_j (\rho e + p) \end{Bmatrix} \qquad \text{(I.2.2)}$$

where ρ , $\underset{\sim}{u}$ and p are density, velocity and pressure respectively; and δ_{ij} is the Kronecker delta. The total energy per unit mass, e , is the sum of the internal and kinetic energies per unit mass. An equation of state relates the pressure to the other variables. That is:

$$p = p(\rho, i) \qquad \text{(I.2.3)}$$

$$i = e - \frac{1}{2} |\underset{\sim}{u}|^2 \qquad \text{(I.2.4)}$$

where i is the internal energy per unit mass. If we have an ideal gas then the equation of state becomes:

$$p = (\gamma - 1)\, \rho\, i \qquad \text{(I.2.5)}$$

Here γ is the ratio of the specific heats.

The coefficients matrices are:

$\underset{\sim}{A}_j$ =

$$
\begin{array}{|c|c|c|c|c|}
\hline
0 & \delta_{j1} & \delta_{j2} & \delta_{j3} & 0 \\
\hline
\delta_{j1}\bar{\gamma}u^2/2 - u_j u_1 & \delta_{j1}u_1 - \delta_{j1}\bar{\gamma}u_1 + u_j & \delta_{j2}u_1 - \delta_{j1}\bar{\gamma}u_2 & \delta_{j3}u_1 - \delta_{j1}\bar{\gamma}u_3 & \delta_{j1}\bar{\gamma} \\
\hline
\delta_{j2}\bar{\gamma}u^2/2 - u_j u_2 & \delta_{j1}u_2 - \delta_{j2}\bar{\gamma}u_1 & \delta_{j2}u_2 - \delta_{j2}\bar{\gamma}u_2 + u_j & \delta_{j3}u_2 - \delta_{j2}\bar{\gamma}u_3 & \delta_{j2}\bar{\gamma} \\
\hline
\delta_{j3}\bar{\gamma}u^2/2 - u_j u_3 & \delta_{j1}u_3 - \delta_{j3}\bar{\gamma}u_1 & \delta_{j2}u_3 - \delta_{j3}\bar{\gamma}u_2 & \delta_{j3}u_3 - \delta_{j3}\bar{\gamma}u_3 + u_j & \delta_{j3}\bar{\gamma} \\
\hline
(\bar{\gamma}u^2 - \gamma e)u_j & \delta_{j1}\varepsilon - \bar{\gamma}u_j u_1 & \delta_{j2}\varepsilon - \bar{\gamma}u_j u_2 & \delta_{j3}\varepsilon - \bar{\gamma}u_j u_3 & \bar{\gamma}u_j \\
\hline
\end{array}
\qquad \text{(I.2.6)}
$$

where

$$\bar{\gamma} = \gamma - 1 \tag{I.2.7}$$

$$\varepsilon = \gamma e - \bar{\gamma} u^2/2 \tag{I.2.8}$$

Matrices for the transformation to the frame of primitive variables are:

$$\underset{\approx}{Q} = \begin{bmatrix} 1 & & & & \\ u_1 & \rho & & & \\ u_2 & & \rho & & \\ u_3 & & & \rho & \\ u^2/2 & \rho u_1 & \rho u_2 & \rho u_3 & 1/\bar{\gamma} \end{bmatrix} \tag{I.2.9}$$

and

$$\underset{\approx}{Q}^{-1} = \begin{bmatrix} 1 & & & & \\ -u_1/\rho & 1/\rho & & & \\ -u_2/\rho & & 1/\rho & & \\ -u_3/\rho & & & 1/\rho & \\ \bar{\gamma} u^2/2 & -\bar{\gamma} u_1 & -\bar{\gamma} u_2 & -\bar{\gamma} u_3 & \bar{\gamma} \end{bmatrix} \tag{I.2.10}$$

where a blank slot indicates a zero term.

The primitive variables vector is:

$$\underset{\sim}{U}' = \begin{Bmatrix} \rho \\ u_1 \\ u_2 \\ u_3 \\ p \end{Bmatrix} \tag{I.2.11}$$

The coefficient matrices in the frame of primitive variables are defined by

$$\underset{\sim}{A}'_j = \begin{bmatrix} u_j & \delta_{j1}\rho & \delta_{j2}\rho & \delta_{j3}\rho & 0 \\ & u_j & & & \delta_{j1}/\rho \\ & & u_j & & \delta_{j2}/\rho \\ & & & u_j & \delta_{j3}/\rho \\ 0 & \delta_{j1}\rho c^2 & \delta_{j2}\rho c^2 & \delta_{j3}\rho c^2 & u_j \end{bmatrix} \qquad \text{(I.2.12)}$$

where c is the acoustic speed:

$$c^2 = \gamma p/\rho \qquad \text{(I.2.13)}$$

The eigenvalues are:

$$\lambda_1 = \lambda_2 = \lambda_3 = k_j u_j$$

$$\lambda_4 = \lambda_1 + c \quad , \quad \lambda_5 = \lambda_1 - c \qquad \text{(I.2.14)}$$

The matrices for diagonalizing $\underset{\sim}{A}'$ (see [W2, W3]) are

$$\underset{\sim}{R} = \begin{bmatrix} k_1 & k_2 & k_3 & \rho/(\sqrt{2}\,c) & \rho/(\sqrt{2}\,c) \\ 0 & -k_3 & k_2 & k_1/\sqrt{2} & -k_1/\sqrt{2} \\ k_3 & 0 & -k_1 & k_2/\sqrt{2} & -k_2/\sqrt{2} \\ -k_2 & k_1 & 0 & k_3/\sqrt{2} & -k_3/\sqrt{2} \\ 0 & 0 & 0 & \rho c/\sqrt{2} & \rho c/\sqrt{2} \end{bmatrix} \qquad \text{(I.2.15)}$$

$$\underset{\sim}{R}^{-1} = \begin{bmatrix} k_1 & 0 & k_3 & -k_2 & -k_1/c^2 \\ k_2 & -k_3 & 0 & k_1 & -k_2/c^2 \\ k_3 & k_2 & -k_1 & 0 & -k_3/c^2 \\ 0 & k_1/\sqrt{2} & k_2/\sqrt{2} & k_3/\sqrt{2} & 1/(\sqrt{2}\,\rho c) \\ 0 & -k_1/\sqrt{2} & -k_2/\sqrt{2} & -k_3/\sqrt{2} & 1/(\sqrt{2}\,\rho c) \end{bmatrix} \qquad \text{(I.2.16)}$$

The symmetrized coefficient matrices are [W2, W3]

$$\underset{\sim}{A}_j =$$

$$\begin{bmatrix}
u_j & & & \delta_{j2}\bar{k}_3 - \delta_{j3}\bar{k}_2 & \delta_{j2}\bar{k}_3 - \delta_{j3}\bar{k}_2 \\
& u_j & & \delta_{j3}\bar{k}_1 - \delta_{j1}\bar{k}_3 & \delta_{j3}k_1 - \delta_{j1}\bar{k}_3 \\
& & u_j & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 \\
\delta_{j2}\bar{k}_3 - \delta_{j3}\bar{k}_2 & \delta_{j3}\bar{k}_1 - \delta_{j1}\bar{k}_3 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & u_j + ck_j & \\
\delta_{j2}\bar{k}_3 - \delta_{j3}k_2 & \delta_{j3}\bar{k}_1 - \delta_{j1}\bar{k}_3 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & & u_j - ck_j
\end{bmatrix} \qquad \text{(I.2.17)}$$

where

$$\bar{k}_j = ck_j/\sqrt{2} \qquad \text{(I.2.18)}$$

I.3 Two-dimensional Case

The conservation variables vector and flux vectors are defined by

$$\underset{\sim}{U} = \rho \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ e \end{Bmatrix} \qquad \text{(I.3.1)}$$

$$\mathcal{F}_j = \begin{Bmatrix} u_j\rho \\ u_j\rho u_1 + \delta_{j1}\, p \\ u_j\rho u_2 + \delta_{j2}\, p \\ u_j(\rho e + p) \end{Bmatrix} \qquad \text{(I.3.2)}$$

The coefficient matrices are defined by:

$$\underset{\sim}{A}_j'' =$$

$$\left[\begin{array}{c|c|c|c} 0 & \delta_{j1} & \delta_{j2} & 0 \\ \hline \delta_{j1}\bar{\gamma}u^2/2 - u_j u_1 & \delta_{j1}u_1 - \delta_{j1}\bar{\gamma}u_1 + u_j & \delta_{j2}u_1 - \delta_{j1}\bar{\gamma}u_2 & \delta_{j1}\bar{\gamma} \\ \hline \delta_{j2}\bar{\gamma}u^2/2 - u_j u_2 & \delta_{j1}u_2 - \delta_{j2}\bar{\gamma}u_1 & \delta_{j2}u_2 - \delta_{j2}\bar{\gamma}u_2 + u_j & \delta_{j2}\bar{\gamma} \\ \hline (\bar{\gamma}u^2 - \gamma e)u_j & \delta_{j1}\varepsilon - \bar{\gamma}u_j u_1 & \delta_{j2}\varepsilon - \bar{\gamma}u_j u_2 & \bar{\gamma}u_j \end{array}\right] \qquad \text{(I.3.3)}$$

The matrices for transformation to the frame of primitive variables are:

$$\underset{\sim}{Q} = \begin{bmatrix} 1 & & & \\ u_1 & \rho & & \\ u_2 & & \rho & \\ u^2/2 & \rho u_1 & u_2 & 1/\bar{\gamma} \end{bmatrix} \qquad \text{(I.3.4)}$$

and

$$\underset{\sim}{Q}^{-1} = \begin{bmatrix} 1 & & & \\ -u_1/\rho & 1/\rho & & \\ -u_2/\rho & & 1/\rho & \\ \bar{\gamma}u^2/2 & -\bar{\gamma}u_1 & -\bar{\gamma}u_2 & \bar{\gamma} \end{bmatrix} \tag{I.3.5}$$

The primitive variables vector is

$$\underset{\sim}{U}' = \begin{Bmatrix} \rho \\ u_1 \\ u_2 \\ p \end{Bmatrix} \tag{I.3.6}$$

The coefficient matrices in the frame of primitive variables are defined by

$$\underset{\sim}{A}_j' = \begin{bmatrix} u_j & \delta_{j1}\rho & \delta_{j2}\rho & 0 \\ & u_j & & \delta_{j1}/\rho \\ & & u_j & \delta_{j2}/\rho \\ 0 & \delta_{j1}\rho c^2 & \delta_{j2}\rho c^2 & u_j \end{bmatrix} \tag{I.3.7}$$

The eigenvalues are:

$$\lambda_1 = \lambda_2 = k_j u_j$$

$$\lambda_3 = \lambda_1 + c \quad , \quad \lambda_4 = \lambda_1 - c \tag{I.3.8}$$

The matrices for diagonalizing $\underset{\sim}{A}'$ are [W2, W3]:

$$\underset{\sim}{R} = \begin{bmatrix} 1 & 0 & \rho/(\sqrt{2}\,c) & \rho/(\sqrt{2}\,c) \\ 0 & k_2 & k_1/\sqrt{2} & -k_1/\sqrt{2} \\ 0 & -k_1 & k_2/\sqrt{2} & -k_2/\sqrt{2} \\ 0 & 0 & \rho c/\sqrt{2} & \rho c/\sqrt{2} \end{bmatrix} \tag{I.3.9}$$

and

$$\underset{\sim}{R}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -1/c^2 \\ 0 & k_2 & -k_1 & 0 \\ 0 & k_1/\sqrt{2} & k_2/\sqrt{2} & 1/(\sqrt{2}\,\rho c) \\ 0 & -k_1/\sqrt{2} & -k_2/\sqrt{2} & 1/(\sqrt{2}\,\rho c) \end{bmatrix} \qquad \text{(I.3.10)}$$

The symmetrized coefficient matrices are defined by

$$\underset{\sim}{\bar{A}}_j =$$

$$\begin{bmatrix} u_j & 0 & 0 & 0 \\ 0 & u_j & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 \\ 0 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & u_j + c\,k_j & 0 \\ 0 & \delta_{j1}\bar{k}_2 - \delta_{j2}\bar{k}_1 & 0 & u_j - c\,k_j \end{bmatrix} \qquad \text{(I.3.11)}$$

I.4 One-dimensional Case:

The conservation variables vector and flux vector are

$$\underset{\sim}{U} = \rho \begin{Bmatrix} 1 \\ u \\ e \end{Bmatrix} \qquad \text{(I.4.1)}$$

$$\underset{\sim}{F} = \begin{Bmatrix} u\rho \\ u\rho u + p \\ u(\rho e + p) \end{Bmatrix} \qquad \text{(I.4.2)}$$

The coefficient matrix is

$$\underset{\sim}{A} = \begin{bmatrix} 0 & 1 & 0 \\ (\gamma - 3)u^2/2 & -(\gamma - 3)u & \bar{\gamma} \\ (\bar{\gamma}u^2 - \gamma e)u & \gamma e - 3\bar{\gamma}u^2/2 & \gamma u \end{bmatrix} \qquad \text{(I.4.3)}$$

Matrices for transformation to the frame of primitive variables are

$$\underset{\sim}{Q} = \begin{bmatrix} 1 & & \\ u & \rho & \\ u^2/2 & \rho\, u & 1/\bar{\gamma} \end{bmatrix} \qquad \text{(I.4.4)}$$

and

$$\underset{\sim}{Q}^{-1} = \begin{bmatrix} 1 & & \\ -u/\rho & 1/\rho & \\ \bar{\gamma}u^2/2 & -\bar{\gamma}u & \bar{\gamma} \end{bmatrix} \qquad \text{(I.4.5)}$$

The primitive variables vector is

$$\underset{\sim}{U}' = \begin{Bmatrix} \rho \\ u \\ p \end{Bmatrix} \qquad \text{(I.4.6)}$$

The coefficient matrix in the frame of primitive variables is

$$\underset{\sim}{A}' = \begin{bmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \rho c^2 & u \end{bmatrix} \qquad \text{(I.4.7)}$$

The eigenvalues are:

$$\lambda_1 = u$$

$$\lambda_2 = \lambda_1 + c \quad , \quad \lambda_3 = \lambda_1 - c \qquad \text{(I.4.8)}$$

The matrices for diagonalizing $\underset{\sim}{A}'$ are [W2, W3]:

$$\underset{\sim}{R} = \begin{bmatrix} 1 & \rho/(\sqrt{2}\,c) & \rho/(\sqrt{2}\,c \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & \rho c/\sqrt{2} & \rho c/\sqrt{2} \end{bmatrix} \tag{I.4.9}$$

and

$$R^{-1} = \begin{bmatrix} 1 & 0 & -1/c^2 \\ 0 & 1/\sqrt{2} & 1/(\sqrt{2}\,\rho c) \\ 0 & -1/\sqrt{2} & 1/(\sqrt{2}\,\rho c) \end{bmatrix} \tag{I.4.10}$$

The diagonalized coefficient matrix is

$$\underset{\sim}{A}' = \underset{\sim}{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} \tag{I.4.11}$$

APPENDIX II

Stability and Accuracy Analysis of Algorithms for the One-dimensional Linear Parabolic Equation

II.1 Development of the Tools for the Analysis

II.1.1 Introduction

Model Problem: Diffusion Equation

One-dimensional diffusion of a function $U(x,t)$ is governed by the following parabolic equation:

$$U_{,t} - \kappa U_{,xx} = 0 \qquad \text{(II.1.1)}$$

where κ is the diffusion coefficient. An initial condition of the following form is assumed:

$$U(x, 0) = e^{ikx} \qquad \text{(II.1.2)}$$

The analysis, arguments, parameters, etc., defined in this appendix are mostly the same, or very similar to, those of chapter 3; we will just state state the ones which are different.

Exact Solution

The exact solution is obtained by separation of variables, as was done in chapter 3. The spatial component of the solution is given by (II.1.2). The damping coefficient and the frequency of the temporal component are:

$$-(\xi, \omega) = \nu = -(\kappa k^2, 0) \qquad \text{(II.1.3)}$$

II.1.2 Finite Element Solution

Spatial Discretization

For the weighted residual formulation of the problem we use the usual weighting function which results in the (Bubnov-) Galerkin formulation. That is

$$\tilde{W} = W \tag{II.1.4}$$

The resulting semi-discrete equation

$$\underset{\sim}{M}\,\dot{\underset{\sim}{v}} + \underset{\sim}{C}\,\underset{\sim}{v} = \underset{\sim}{0} \tag{II.1.5}$$

is obtained by following a procedure which is very similar to the one we followed in chapter 2. The element level matrices $\underset{\sim}{m}^e$ and $\underset{\sim}{c}^e$ are defined differently:

$$m^e_{ab} = \int_{\Omega^e} N_a N_b \, d\Omega \tag{II.1.6}$$

$$c^e_{ab} = \int_{\Omega^e} \kappa \, N_{a,x} N_{b,x} \, d\Omega \tag{II.1.7}$$

Assuming constant κ and mesh spacing h, and going through the same arguments that we went through in the corresponding parts of chapter 3, the j^{th} equation of the system of equations can be written as:

$$\left(\frac{h}{2}\,\underset{\sim}{D}_r\,\dot{T} + \kappa\,\frac{2}{h}\,\underset{\sim}{D}_2 T\right) \begin{Bmatrix} x_{j-1} \\ x_j \\ x_{j+1} \end{Bmatrix} = 0 \tag{II.1.8}$$

thus:

$$\frac{h}{2}\,\underset{\sim}{D}_r\,\underset{\sim}{E}\,\dot{T} + \kappa\,\frac{2}{h}\,\underset{\sim}{D}_2\,\underset{\sim}{E}\,T = 0 \tag{II.1.9}$$

We define the scalars M and C corresponding to $\underset{\sim}{M}$ and $\underset{\sim}{C}$ as:

$$M = \frac{1}{2} \underset{\sim}{D}_r \underset{\sim}{E} \tag{II.1.10}$$

$$\Delta t\, C = \frac{2 \kappa \Delta t}{h^2} \underset{\sim}{D}_2 \underset{\sim}{E} \tag{II.1.11}$$

The non-dimensional parameter C_κ is defined as:

$$C_\kappa = \frac{2 \kappa \Delta t}{h^2} \tag{II.1.12}$$

We note that κ/h has units of velocity; thus C_κ can be regarded as a "diffusion Courant number".

The problem is reduced to solving the following ordinary differential equation

$$M \dot{T} + C\, T = 0 \tag{II.1.13}$$

Numerical Frequency and Damping Coefficient

Going through the same steps we went through in chapter 3, we obtain

$$e^{\tilde{\nu}\Delta t} = \operatorname{tr} \underset{\sim}{A} = 1 + \frac{Q}{R}\left((1 - R)^S - 1\right) \tag{II.1.14}$$

where

$$R = (M^R + \alpha\, \Delta t\, C^R)/\overline{M} \tag{II.1.15}$$

$$Q = \Delta t\, C^R/\overline{M} \tag{II.1.16}$$

$$\Delta tC = C_\kappa \underset{\sim}{D}_2 \underset{\sim}{E} \tag{II.1.17}$$

$$\overline{M} = M^L + \alpha\, \Delta t\, C^L \tag{II.1.18}$$

Then

$$\tilde{\nu}\Delta t = \ln(\mathrm{tr}\ \underset{\sim}{A}) \qquad \text{(II.1.19)}$$

For comparison, we need to express $\nu\Delta t$ in terms of the same dimensionless parameters; from (II.1.3):

$$\nu\Delta t = -\kappa k^2 \Delta t = -\frac{1}{2} C_\kappa q^2 \qquad \text{(II.1.20)}$$

We define the analytical and numerical amplification factors, Z and $\tilde{Z}$, as:

$$Z = e^{\nu\Delta t} \qquad \text{(II.1.21)}$$

$$\tilde{Z} = e^{\tilde{\nu}\Delta t} \qquad \text{(II.1.22)}$$

II.2 Unified Analysis of Algorithms

II.2.1 Introduction

When we study the stability and accuracy of algorithms for the diffusion equation, we classify the algorithms considered as implicit and explicit types.

This time we do not use Petrov-Galerkin algorithms, therefore, the type of implicit algorithm will solely depend on the parameter r (defined in section 2.6) which specifies the way mass terms are integrated.

In the diffusion problem, we observe from (II.1.21) that the exact amplification factor has no imaginary part and $0 \leq Z \leq 1$. The numerical amplification factor has no imaginary part either. However, we also have to make sure that the numerical solution is stable.

Graphically, we compare the exact and numerical amplification factors and inspect the ratio of the numerical damping coefficient to the exact damping coefficient for several algorithms.

Obtaining closed form expressions for $\tilde{Z}$ and the damping ratio $\tilde{\xi}/\xi$ is relatively easy, yet, these expressions are not in a simple form. One can expand the damping ratio in Δt and h, and observe the temporal and spatial accuracies of the algorithms.

The Galerkin algorithms use the integration stencils:

$$\underset{\sim}{D}_r = 2(r,\ 1-2r,\ r) \tag{II.2.1}$$

$$\underset{\sim}{D}_2 = \left(-\frac{1}{2},\ 1,\ -\frac{1}{2}\right) \tag{II.2.2}$$

Then the scalars M and ΔtC become:

$$M = 1 - 2rV \tag{II.2.3}$$

$$\Delta tC = C_K V \tag{II.2.4}$$

when

$$V = 1 - \cos q \tag{II.2.5}$$

II.2.2 Implicit Algorithms

For implicit algorithms, the amplification factor from (II.1.14) is

$$\tilde{Z} = 1 - Q = 1 - \frac{C_K V}{1 + V(-2r + \alpha C_K)} \tag{II.2.6}$$

The stability limits are determined by the condition:

$$|\tilde{Z}| \leq 1 \tag{II.2.7}$$

The inequality $\tilde{Z} \leq 1$ dictates that

$$1 + 2\alpha C_K \geq 4r \tag{II.2.8}$$

This condition is automatically satisfied as long as $r \leq \frac{1}{4}$. The second inequality $\tilde{Z} \geq -1$ requires that:

$$C_K V(2\alpha - 1) \geq 4rV - 2 \tag{II.2.9}$$

Since $r \leq \frac{1}{4}$ from the first inequality, this condition is satisfied as long as $\alpha \geq \frac{1}{2}$. We summarize the unconditional stability conditions for the implicit algorithms as:

$$r \leq \frac{1}{4} \quad \text{and} \quad \alpha \geq \frac{1}{2} \tag{II.2.10}$$

For the accuracy analysis, we can employ asymptotic expansions of the damping ratio in Δt and h. We expand $\tilde{\xi}/\xi$ in Δt after setting $h = 0$ while, for expansion in h we first set $\Delta t = 0$. These expansions are:

$$\left(\frac{\tilde{\xi}}{\xi}\right)\Big|_{h=0} = 1 + (\kappa k^2 \Delta t)\left(\frac{1}{2} - \alpha\right) + (\kappa k^2 \Delta t)^2\left(\frac{1}{3} + \alpha^2 - \alpha\right) + O(\Delta t^3) \tag{II.2.11}$$

$$\left(\frac{\tilde{\xi}}{\xi}\right)\Big|_{\Delta t=0} = 1 + (k h)^2\left(r - \frac{1}{12}\right) + O(h^4) \tag{II.2.12}$$

Observe that from (II.2.11), $\alpha = \frac{1}{2}$ is the condition for second-order accuracy is time. From (II.2.12), we see that second-order accuracy in space is automatic, and that the value $r = 1/12$ results in fourth-order accuracy.

II.2.3 Explicit Algorithms

Explicit 1-pass Algorithm

For this algorithm, the amplification factor is:

$$\tilde{Z} = 1 - Q = 1 - C_K V \qquad \text{(II.2.13)}$$

The inequality $\tilde{Z} \leq 1$ is automatically satisfied since $C_K V \geq 0$. The second inequality $\tilde{Z} \geq -1$ dictates that $C_K V \geq 2$. The worse case occurs when $V = 2$; then, $C_K \leq 1$ is the stability limit.

For the accuracy analysis, the expansions of the damping ratio in Δt and h are:

$$\left(\frac{\tilde{\xi}}{\xi}\right)\Bigg|_{h=0} = 1 + \frac{1}{2}(\kappa k^2 \Delta t) + \frac{1}{3}(\kappa k^2 \Delta t)^2 + O(\Delta t^3) \qquad \text{(II.2.14)}$$

$$\left(\frac{\tilde{\xi}}{\xi}\right)\Bigg|_{\Delta t=0} = 1 - \frac{1}{12}(k h)^2 + O(h^4) \qquad \text{(II.2.15)}$$

Thus the 1-pass algorithm is first-order accurate in time and second-order accurate in space.

Explicit 2-pass Algorithms

The amplification factor for these algorithms is

$$\tilde{Z} = 1 + Q(R - 2) = 1 + C_K V(-1 + V(-2r + \alpha C_K)) \qquad \text{(II.2.16)}$$

The inequality $\tilde{Z} \leq 1$ requires that $C_K V(-1 - 2rV + \alpha C_K V) \leq 0$; this leads to

$$C_K \leq \frac{1 + 4r}{2\alpha} \qquad \text{(II.2.17)}$$

The inequality $\tilde{Z} \geq -1$ dictates that $C_K V(1 + 2rV - \alpha C_K V) - 2 \leq 0$; one can show that this is satisfied as long as $\alpha \geq \frac{1}{2}$ and $r \leq \frac{1}{4}$. Combining these inequalities with the inequality of (II.2.17), we can state that the maximum value C_K can assume is 2, corresponding to $\alpha = \frac{1}{2}$ and $r = \frac{1}{4}$.

The expansions in Δt and h are:

$$\left(\frac{\tilde{\xi}}{\xi}\right)\Bigg|_{h = 0} = 1 + (\kappa k^2 \Delta t)\left(\frac{1}{2} - \alpha\right) + (\kappa k^2 \Delta t)^2 \left(\frac{1}{3} - \alpha\right) + 0(\Delta t^3) \qquad \text{(II.2.18)}$$

$$\left(\frac{\tilde{\xi}}{\xi}\right)\Bigg|_{\Delta t = 0} = 1 + (k h)^2 \left(r - \frac{1}{12}\right) + 0(h^4) \qquad \text{(II.2.19)}$$

Thus, with $\alpha = \frac{1}{2}$ second-order accuracy is achieved in time; and, as in the implicit case, $r = \frac{1}{12}$ can raise the spatial accuracy from second to fourth-order.

II.3 Summary

Asymptotic expansions, stability limits determined analytically, and graphical representations of $\tilde{Z}$ (figure II.1) and $\tilde{\xi}/\xi$ (figure II.2) are utilized for the stability and accuracy analysis of the algorithms considered.

Stability

We observe from Fig. II.1 that for $\alpha = \frac{1}{2}$, $C_K = 0.2, 0.4, 0.6, 0.8, 1.0$ and $q \in]0, \pi[$, all the algorithms considered had amplification factors which remained in the interval $[-1, +1]$; that is, all were stable. This can also be deduced from the stability limits determined analytically.

For the implicit algorithms (GC and GL) we have stability as long as

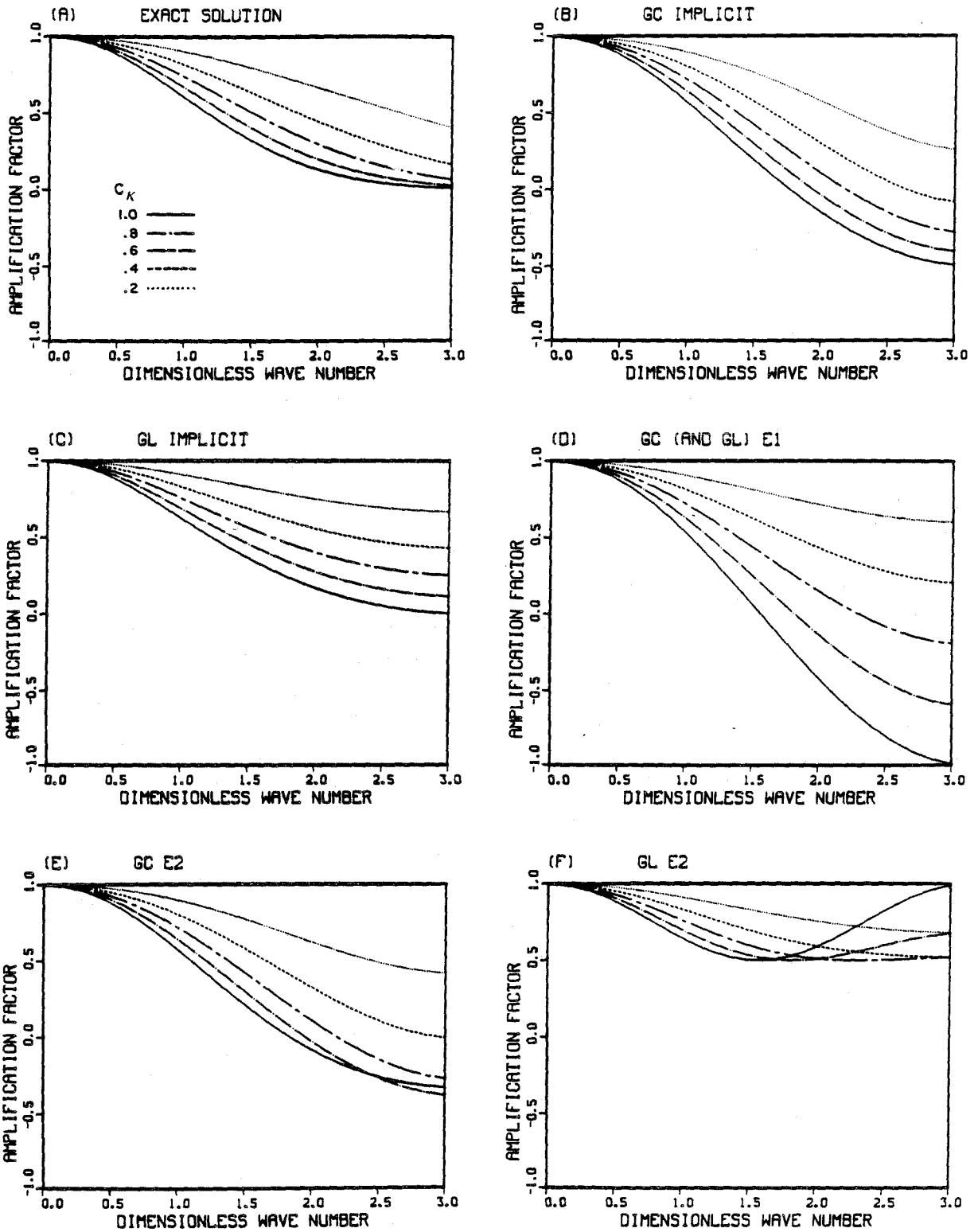


Figure II.1 Amplification factors for heat equation algorithms.

$r \leq \frac{1}{4}$ and $\alpha \geq \frac{1}{2}$. With $\alpha = \frac{1}{2}$, both the consistent mass $(r = \frac{1}{6})$ algorithm (GC) and the lumped-mass $(r = 0)$ algorithm (GL) are therefore stable.

For the explicit 1-pass algorithm (E1), we found that the stability condition is $C_K \leq 1$ (independent of r and α); Fig. II.1 (D) also shows this clearly. We recall that $C_K \leq 1$ is the condition for $\tilde{Z}$ not to drop below - 1.0; in Fig. II.1 (D) we observe that the curve for $C_K = 1$ just touches the stability limit at $q = \pi$.

For the explicit 2-pass algorithms (E2), the stability requirements are $C_K \leq (1 + 4r)$, $\alpha \geq \frac{1}{2}$, $r \leq \frac{1}{4}$. The consistent mass $(r = \frac{1}{6})$ algorithm with $\alpha = \frac{1}{2}$ satisfies the restrictions on the values of α and r; as a restriction on C_K, we get $C_K \leq \frac{5}{3}$. The lumped mass $(r = 0)$ algorithm with $\alpha = \frac{1}{2}$ has a restriction $C_K \leq 1.0$.

Accuracy

For the accuracy of the algorithms, Fig. II.2 provides information for $\alpha = \frac{1}{2}$, $C_K = 0.2, 0.4, 0.6, 0.8, 1.0$ and points away from $q = 0$. The asymptotic expansions on the other hand provide information needed in the neighborhood of $q = 0$.

We observe from Fig. II.2 that the GC implicit algorithm overdamps while GL implicit underdamps. Similarly, GC - E2 overdamps while GL - E2 underdamps. The explicit 1-pass algorithm, on the other hand, overdamps or underdamps depending on the value of C_K. The crossover point above which overdamping occurs, is about $C_K = 0.4$.

The asymptotic expansions show that all the algorithms considered have second-order spatial accuracy for $r \neq \frac{1}{12}$. For $r = \frac{1}{12}$, the implicit

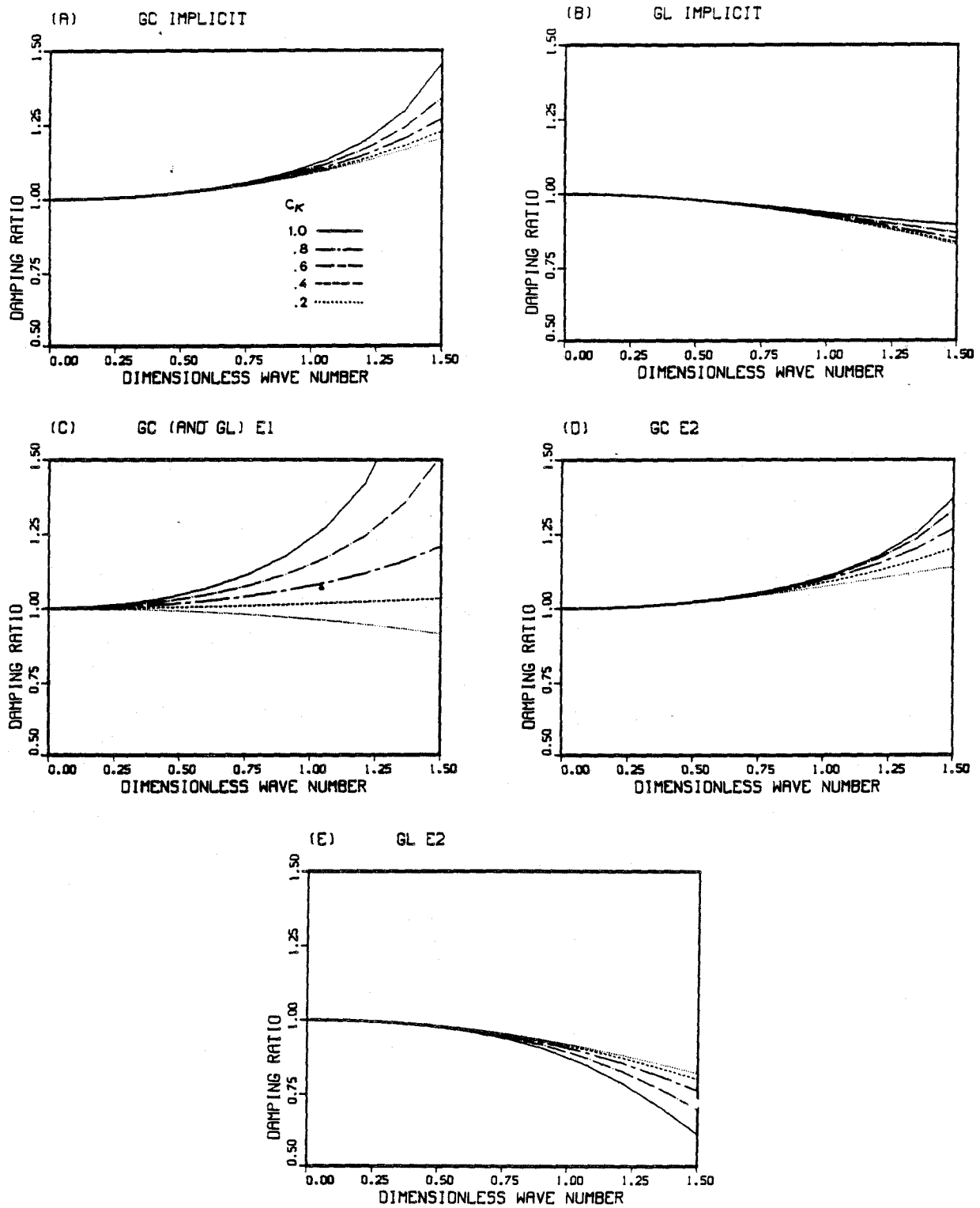


Figure II.2 Damping Ratios for heat equation algorithms.

and explicit 2-pass algorithms attain fourth-order accuracy. The algorithms considered have first-order temporal accuracy for $\alpha \neq \frac{1}{2}$. For $\alpha = \frac{1}{2}$, the implicit and explicit 2-pass algorithms attain second-order accuracy.

APPENDIX III

Stability and Accuracy Analysis of Algorithms for One-dimensional Linear Second-order Hyperbolic Equation

III.1 Development of the Tools for the Analysis

III.1.1 Introduction

Model Problem: Wave Equation

One-dimensional, undamped wave motion is governed by the following second-order hyperbolic equation for $U(x, t)$:

$$U_{,tt} - \kappa U_{,xx} = 0 \qquad \text{(III.1.1)}$$

where $(\kappa)^{\frac{1}{2}}$ is the wave-propagation speed.

The initial conditions associated with the problem are assumed to be of the form:

$$U(x, 0) = U_0 e^{ikx} \qquad \text{(III.1.2)}$$

$$\dot{U}(x, 0) = V_0 e^{ikx} \qquad \text{(III.1.3)}$$

We note that these could be viewed as Fourier components of a general set of initial conditions for U and $\dot{U}$.

Exact Solution

For constant propagation speed, assuming a solution of the form

$$U(x, t) = X(x)T(t) \qquad \text{(III.1.4)}$$

leads to the following spatial and temporal components for the function

$U(x, t)$:

$$X(x) = e^{ikx} \tag{III.1.5}$$

$$T(t) = A_1 e^{+\nu t} + A_2 e^{-\nu t} \tag{III.1.6}$$

where the constants A_1 and A_2 depend on the constants U_0 and V_0 and

$$\nu^2 = -\kappa k^2 \tag{III.1.7}$$

We define the damping coefficient ξ and the frequency ω as

$$(-\xi, \omega) = \nu = i\sqrt{\kappa}\, k \tag{III.1.8}$$

The exact solution is seen to have no damping.

III.1.2 Finite Element Solution

Spatial Discretization

For the weighted residual formulation of the problem, we use the usual weighting functions ($\tilde{W} = W$) which lead to the (Bubnov-) Galerkin formulation. The resulting semi-discrete equation

$$\underset{\sim}{M}\,\ddot{\underset{\sim}{v}} + \underset{\sim}{C}\,\underset{\sim}{v} = \underset{\sim}{0} \tag{III.1.9}$$

is obtained by following a procedure very similar to the one we followed in chapter 2 and appendix II. The element level matrices $\underset{\sim}{M}^e$ and $\underset{\sim}{C}^e$ are defined as:

$$m^e_{ab} = \int_{\Omega^e} N_a N_b \, d\Omega \tag{III.1.10}$$

$$c^e_{ab} = \int_{\Omega^e} \kappa \, N_{a,x} N_{b,x} \, d\Omega \tag{III.1.11}$$

Going over a sequence of steps as we did in chapter 3, we end up with the following ordinary differential equation

$$M \ddot{T} + C T = 0 \qquad \text{(III.1.12)}$$

Here the scalars M and C are

$$M = \frac{1}{2} \underset{\sim}{D}_r \underset{\approx}{E} \qquad \text{(III.1.13)}$$

$$\Delta t^2 C = \frac{2\kappa \Delta t^2}{h^2} \underset{\sim}{D}_2 \underset{\approx}{E} \qquad \text{(III.1.14)}$$

For the purpose of analysis, we assume constant κ and h. The dimensionless parameter C_κ is defined as

$$\frac{2\kappa \Delta t^2}{h^2} = 2 C_\kappa^2 = 2\left(\frac{\sqrt{\kappa}\, \Delta t}{h}\right)^2 \qquad \text{(III.1.15)}$$

Here, C_κ is a Courant number based on the propagation speed $\sqrt{\kappa}$.

Time Integration

The ordinary differential equation of (III.1.12) can be solved by a family of time integration schemes described in [H11]. We adopt the representation

$$\underset{\sim}{Y} = [T, \dot{T}, \ddot{T}] \qquad \text{(III.1.16)}$$

and let $\underset{\sim}{Y}_n$ denote the approximation to $\underset{\sim}{Y}$ at the nth time step.

Given $\underset{\sim}{Y}_n$, we go through a predictor phase and an iterative phase to calculate $\underset{\sim}{Y}_{n+1}$.

In the predictor phase, we calculate the zeroth iteration value, $\underset{\sim}{Y}_{n+1}^{(0)}$, by the following operation:

$$\underset{\sim}{Y}_{n+1}^{(0)} = \underset{\sim}{P}\ \underset{\sim}{Y}_{n} \tag{III.1.17}$$

Here $\underset{\sim}{P}$ is the predictor matrix defined as:

$$\underset{\sim}{P} = \begin{bmatrix} 1 & \Delta t & (1 - 2\beta)\ \frac{\Delta t^2}{2} \\ 0 & 1 & (1 - \alpha)\ \Delta t \\ 0 & 0 & 0 \end{bmatrix} \tag{III.1.18}$$

where Δt is the time step and α , β are the Newmark parameters which control the stability and accuracy of the algorithm.

The iterative phase starts with the zeroth-iteration value $\underset{\sim}{Y}_{n+1}^{(0)}$ and continues according to the recurrence rule below:

Given $\underset{\sim}{Y}_{n+1}^{(i)}$ solve the following system for $\underset{\sim}{Y}_{n+1}^{(i+1)}$:

$$M^{L}\ \Delta\ddot{T}_{n+1}^{(i)} + C^{L}\ \Delta T_{n+1}^{(i)} = -\left(M^{R}\ddot{T}_{n+1}^{(i)} + C^{R}T_{n+1}^{(i)}\right) \tag{III.1.19}$$

$$\Delta\dot{T}_{n+1}^{(i)} = \alpha\ \Delta t\ \Delta\ddot{T}_{n+1}^{(i)} \tag{III.1.20}$$

$$\Delta T_{n+1}^{(i)} = \beta\ \Delta t^2\ \Delta\ddot{T}_{n+1}^{(i)} \tag{III.1.21}$$

$$\ddot{T}_{n+1}^{(i+1)} = \ddot{T}_{n+1}^{(i)} + \Delta\ddot{T}_{n+1}^{(i)} \tag{III.1.22}$$

$$\dot{T}_{n+1}^{(i+1)} = \dot{T}_{n+1}^{(i)} + \Delta\dot{T}_{n+1}^{(i)} \tag{III.1.23}$$

$$T_{n+1}^{(i+1)} = T_{n+1}^{(i)} + \Delta T_{n+1}^{(i)} \tag{III.1.24}$$

The superscripts L and R refer to the left and right-hand sides of

(III.1.19). We reserve the option of having different evaluations for M and C on different sides, as in chapter 3.

The recurrence rule defined above can be expressed as:

$$\underset{\sim}{Y}_{n+1}^{(i+1)} = \underset{\sim}{J}\, \underset{\sim}{Y}_{n+1}^{(i)} \qquad \text{(III.1.25)}$$

$\underset{\sim}{J}$ is the iteration matrix:

$$\underset{\sim}{J} = \frac{1}{\overline{M}} \begin{bmatrix} \overline{M} - \beta\, \Delta t^2 C^R & 0 & -\beta\, \Delta t^2 M^R \\ -\alpha\, \Delta t\, C^R & 1 & -\alpha\, \Delta t\, M^R \\ -C^R & 0 & \overline{M} - M^R \end{bmatrix} \qquad \text{(III.1.26)}$$

where

$$\overline{M} = M^L + \beta\, \Delta t^2\, C^L \qquad \text{(III.1.27)}$$

Combining the predictor and iterative phases:

$$\underset{\sim}{Y}_{n+1} = \underset{\sim}{A}\, \underset{\sim}{Y}_n \qquad \text{(III.1.28)}$$

$$\underset{\sim}{A} = \underset{\sim}{J}^S \underset{\sim}{P} \qquad \text{(III.1.29)}$$

Here S is the number of iterations performed in the iterative phase. Exploiting the fact that $\det \underset{\sim}{P} = 0$,

$$\det \underset{\sim}{A} = \det \underset{\sim}{J}^S \det \underset{\sim}{P} = 0\,, \qquad \text{(III.1.30)}$$

we determine that at any time step, $n+1$, $\ddot{T}_{n+1}$ can be expressed in terms of T_{n+1} and $\dot{T}_{n+1}$. That is:

$$\ddot{T}_{n+1} = \mu_1\, T_{n+1} + \mu_2\, \dot{T}_{n+1} \qquad \text{(III.1.31)}$$

We further observe from the structure of the matrix $\underset{\sim}{J}$ that the iterates $T_{n+1}^{(i+1)}$ and $\ddot{T}_{n+1}^{(i+1)}$ have no dependence on $\dot{T}_{n+1}^{(i)}$. This is, of course, a consequence of the absence of the $\dot{T}$ term in the ordinary differential equation of (III.1.12). This observation suggests that we extract the submatrices $\overline{\underset{\sim}{J}}$ and $\overline{\underset{\sim}{P}}$ from the matrices $\underset{\sim}{J}$ and $\underset{\sim}{P}$ respectively:

$$\overline{\underset{\sim}{J}} = \frac{1}{\overline{M}} \begin{bmatrix} \overline{M} - \beta \Delta t^2 C^R & - \beta \Delta t^2 M^R \\ - C^R & \overline{M} - M^R \end{bmatrix} \tag{III.1.32}$$

$$\overline{\underset{\sim}{P}} = \begin{bmatrix} 1 & \Delta t & (1 - 2\beta) \frac{\Delta t^2}{2} \\ 0 & 0 & 0 \end{bmatrix} \tag{III.1.33}$$

and rewrite (III.1.28) in a different form:

$$\begin{Bmatrix} T_{n+1}^{(i+1)} \\ \ddot{T}_{n+1}^{(i+1)} \end{Bmatrix} = \overline{\underset{\sim}{J}} \begin{Bmatrix} T_{n+1}^{(i)} \\ \ddot{T}_{n+1}^{(i)} \end{Bmatrix} \tag{III.1.34}$$

$$\begin{Bmatrix} T_{n+1} \\ \ddot{T}_{n+1} \end{Bmatrix} = \overline{\underset{\sim}{J}}^{S} \overline{\underset{\sim}{P}} \underset{\sim}{Y}_n \tag{III.1.35}$$

$$\dot{T}_{n+1} = \dot{T}_n + (1 - \alpha)\Delta t \ \ddot{T}_n + \alpha \Delta t \ \dot{T}_{n+1} \tag{III.1.36}$$

Eq. (III.1.35) reveals that the relation of (III.1.31) degenerates to the form:

$$\ddot{T}_{n+1} = \mu \ T_{n+1} \tag{III.1.37}$$

with μ being

$$\mu = \frac{\bar{J}^S_{21}}{\bar{J}^S_{11}} \qquad \text{(III.1.38)}$$

where $\bar{J}^S_{21}$ and $\bar{J}^S_{11}$ are the components of $\bar{\underset{\sim}{J}}^S$.

By adopting the equivalents form of (III.1.35)-(III.1.36), we need only to calculate the Sth power of a 2×2 matrix instead of a 3×3 matrix. Besides, the only difference between the matrix $\bar{\underset{\sim}{J}}$ here and the matrix $\underset{\sim}{J}$ of chapter 3 is that the term $\beta\Delta t^2$ replaces the term $\alpha\Delta t$. With this in mind, by analogy, we write the eigenvalues of $\bar{\underset{\sim}{J}}$ from (3.1.57)-(3.1.58):

$$\lambda_1 = 1 \qquad \text{(III.1.39)}$$

$$\lambda_2 = 1 - (M^R + \beta\,\Delta t^2 C^R)/\bar{M} \qquad \text{(III.1.40)}$$

By way of a procedure similar to the one in chapter 3;

$$\bar{\underset{\sim}{J}}^S = \frac{1}{\bar{M}} \begin{bmatrix} \bar{M} - \beta\,\bar{b}\,\Delta t^2 C^R & -\beta\,\bar{b}\,\Delta t^2 M^R \\ -\bar{b}\,C^R & \bar{M} - \bar{b}\,M^R \end{bmatrix} \qquad \text{(III.1.41)}$$

where

$$\bar{b} = \frac{1 - (1 - R)^S}{R} \qquad \text{(III.1.42)}$$

$$R = (M^R + \beta\,\Delta t^2 C^R)/\bar{M} \qquad \text{(III.1.43)}$$

It is interesting to note that the only difference between the matrix $\bar{\underset{\sim}{J}}$ and its Sth power is the insertion of a $\bar{b}$ term which accounts for the number of iterations. From (III.1.38):

$$\mu = \frac{-\bar{b}\,C^R}{\bar{M} - \beta\,\bar{b}\,\Delta t^2 C^R} \qquad \text{(III.1.44)}$$

and from (III. 1.35)-(III.1.37)

$$\begin{Bmatrix} T_{n+1} \\ \dot{T}_{n+1} \end{Bmatrix} = \underset{\sim}{A^*} \begin{Bmatrix} T_n \\ \dot{T}_n \end{Bmatrix} \tag{III.1.45}$$

with $\underset{\sim}{A^*}$ given as

$$\underset{\sim}{A^*} = \begin{bmatrix} \bar{r}\,\rho & \bar{r}\,\Delta t \\ \Delta t\mu[(1-\alpha) + \alpha\bar{r}\rho] & (1+\alpha)\bar{r}\,\bar{\mu} \end{bmatrix} \tag{III.1.46}$$

where

$$\bar{\mu} = \Delta t^2\mu \quad , \quad \rho = 1 + (\Delta t^2/2)(1 - 2\beta)\mu$$

and

$$\bar{r} = (\bar{M} - \beta\,\bar{b}\,\Delta t^2 C^R)/\bar{M} \tag{III.1.47}$$

Numerical Frequency

We are mainly interested in the invariants of the matrix $\underset{\sim}{A^*}$, which are:

$$\text{tr}\ \underset{\sim}{A^*} = 2 - E(1 + 2\alpha) \tag{III.1.48}$$

$$\det \underset{\sim}{A^*} = 1 + E(1 - 2\alpha) \tag{III.1.49}$$

where

$$E = \frac{\bar{b}}{2}\,\frac{\Delta t^2 C^R}{\bar{M}} \tag{III.1.50}$$

The eigenvalues $\tilde{Z}_1$ and $\tilde{Z}_2$ of the matrix $\underset{\sim}{A^*}$ are found by solving the quadratic equation

$$\tilde{Z}^2 - \text{tr}\ \underset{\sim}{A^*}\tilde{Z} + \det \underset{\sim}{A^*} = 0 \tag{III.1.51}$$

Assuming a solution of the form:

$$T_n = A_1 e^{\tilde{\nu}_1 \Delta t n} + A_2 e^{\tilde{\nu}_2 \Delta t n} \qquad \text{(III.1.52)}$$

one can show that the eigenvalues of $\underset{\sim}{A}^*$ are related to $\tilde{\nu}_1 \Delta t$ and $\tilde{\nu}_2 \Delta t$ by

$$e^{\tilde{\nu}_1 \Delta t} = \tilde{z}_1 \quad , \quad e^{\tilde{\nu}_2 \Delta t} = \tilde{z}_2 \qquad \text{(III.1.53)}$$

The eigenvalues can be used for comparing the numerical $\tilde{\nu}$ to the exact ν.

We now have $\tilde{z}_1$ and $\tilde{z}_2$ calculated in terms of two dimensionless parameters q and C_K. We also need to express ν in terms of the same parameters. From (III.1.8):

$$\nu \Delta t = (0, \Delta t \sqrt{\kappa}\, k) = (0, C_K\, q) \qquad \text{(III.1.54)}$$

while

$$\tilde{\nu}_{1,2}\, \Delta t = \ln(\tilde{z}_{1,2}) \qquad \text{(III.1.55)}$$

III.2 Unified Analysis of Algorithms

III.2.1 Introduction

Stability and accuracy analyses of algorithms for the wave equation are made for the implicit and explicit cases. Petrov-Galerkin algorithms are not introduced. The type of implicit algorithm depends only on the r term (defined in section 2.6).

The Galerkin algorithms utilize the integration stencils

$$\underset{\sim}{D}_r = 2[r,\ 1 - 2r,\ r] \qquad \text{(III.2.1)}$$

$$\underset{\sim}{D}_2 = \left[-\frac{1}{2},\ 1,\ -\frac{1}{2}\right] \qquad \text{(III.2.2)}$$

The scalars M and $\Delta t^2 C$ then become:

$$M = 1 - 2rV \tag{III.2.3}$$

$$\Delta t^2 C = 2C_K^2 V \tag{III.2.4}$$

where

$$V = 1 - \cos q \tag{III.2.5}$$

For consistency with the exact solution, we need to have $\tilde{\nu}_2 \Delta t = - \tilde{\nu}_1 \Delta t$; that is

$$\tilde{Z}_1 \tilde{Z}_2 = \det \underset{\sim}{A}^* = 1 \ . \tag{III.2.6}$$

This condition is satisfied if $\alpha = \frac{1}{2}$.

For stability, we require that $|\tilde{Z}_1|$, $|\tilde{Z}_2| \leq 1$; but this is possible only when $|\tilde{Z}_1| = |\tilde{Z}_2| = 1$. The eigenvalues $\tilde{Z}_1$ and $\tilde{Z}_2$ are given by the expression:

$$\tilde{Z}_{1,2} = \frac{\text{tr } \underset{\sim}{A}^*}{2} \pm \sqrt{\frac{\Delta^2}{4}} \tag{III.2.7}$$

where

$$\frac{\text{tr } \underset{\sim}{A}^*}{2} = 1 - E \tag{III.2.8}$$

$$\frac{\Delta^2}{4} = E(E - 2) \tag{III.2.9}$$

Clearly $\frac{\Delta^2}{4} \leq 0$, if and only if

$$0 \leq E \leq 2 \tag{III.2.10}$$

This is the stability condition, because, if $\frac{\Delta^2}{4} \leq 0$ then $|\tilde{z}_1| = |\tilde{z}_2| = 1$. Otherwise, we would have either $|\tilde{z}_1|$ or $|\tilde{z}_2|$ greater than unity.

III.2.2 Implicit and Explicit 1-pass Algorithms

For an implicit or explicit 1-pass algorithm, the term E is

$$E = \frac{c_K^2 V}{(1 - 2rV) + 2\beta c_K^2 V} \qquad \text{(III.2.11)}$$

In this expression, if we set $r = \beta = 0$, we get the value of E for the explicit 1-pass algorithm (Warning: This applies to (II.2.11) only, but not the algorithm in general). From the inequality of III.2.10 we determine the stability limits.

One can show that $E \geq 0$, if and only if

$$c_K^2 \geq \frac{4r - 1}{4\beta} \qquad \text{(III.2.12)}$$

It is quite clear that if $r \leq \frac{1}{4}$, then this condition is satisfied unconditionally.

The inequality $E \leq 2$ implies the following condition:

$$\left(4r - c_K^2(4\beta - 1)\right)V \leq 2 \qquad \text{(III.2.13)}$$

If $\beta \geq \frac{1}{4}$, this inequality dictates that:

$$c_K^2 \geq \frac{4r - 1}{4\beta - 1} \qquad \text{(III.2.14)}$$

Clearly, if $r \leq \frac{1}{4}$, this condition is satisfied unconditionally. If not, then we need to satisfy both (III.2.12) and (III.2.14). One can show that (III.2.14) implies (III.2.12). Therefore (III.2.14) is the key condition.

If $\beta < \frac{1}{4}$, we cannot satisfy (III.2.13) unless $r \leq \frac{1}{4}$. Provided that this happens, the restriction on C_K is given as:

$$C_K^2 \leq \frac{1 - 4r}{1 - 4\beta} \qquad \text{(III.2.15)}$$

For the implicit algorithm with $r = \frac{1}{6}$ and $\beta = \frac{1}{4}$ we attain unconditional stability. For the explicit 1-pass algorithm with $r = 0$ and $\beta = 0$, the condition of III.2.15 becomes

$$C_K^2 \leq 1 . \qquad \text{(III.2.16)}$$

III.2.3 Explicit 2-pass Algorithms

For 2-pass algorithms

$$E = C_K^2 V(1 + 2rV - 2\beta C_K^2 V) \qquad \text{(III.2.17)}$$

One can show that $E \geq 0$ if and only if

$$C_K^2 \leq \frac{4r + 1}{4\beta} \qquad \text{(III.2.18)}$$

For $E \leq 2$, the following condition must be met:

$$\beta \geq \frac{(1 + 4r)^2}{16} \qquad \text{(III.2.19)}$$

With $\beta = \frac{1}{4}$, both $r = \frac{1}{6}$ and $r = 0$ satisfy this condition. Then the inequality of (III.2.18) implies that

$$C_K^2 \leq \frac{5}{3} \quad \text{for} \quad r = \frac{1}{6} \qquad \text{(III.2.20)}$$

$$C_K^2 \leq 1 . \quad \text{for} \quad r = 0 \qquad \text{(III.2.21)}$$

III.3 Summary

According to the stability guidelines of the previous sections, with $\alpha = \frac{1}{2}$ and $\beta = \frac{1}{4}$, all the algorithms considered are stable (with no modulus error) for the ranges of $C_K \in [0, 1]$ and $q \in [0, \pi]$. The accuracy information is provided by the graphs of Fig. III.1.

The implicit, lumped mass algorithm (GL) becomes less accurate as C_K increases. The implicit, consistent mass algorithm (GC), on the other hand, maximizes its accuracy around $C_K = 0.6$.

The explicit 1-pass algorithm (E1) satisfies the unit CFL condition defined in section 3.2.4. This is as we would expect, because if $C_K = 1$;

$$\text{tr}\,\frac{\underset{\sim}{A^*}}{2} = 1 - E = 1 - C_K^2 V$$
$$= 1 - V \qquad \text{(III.3.1)}$$

$$\frac{\Delta^2}{4} = E(E - 2) = C_K^2 V(C_K^2 V - 2)$$
$$= V(V - 2) \qquad \text{(III.3.2)}$$

and therefore the eigenvalues are

$$\tilde{z}_{1,2} = (1 - V) \pm \sqrt{V(V - 2)}$$
$$= \cos q \pm i \sin q$$
$$= e^{\pm iq} \qquad \text{(III.3.3)}$$

This is the same expression as for the exact solution. The accuracy of E1 increases as C_K increases within the stability regime.

The explicit 2-pass (E2) version of GC is more accurate than the

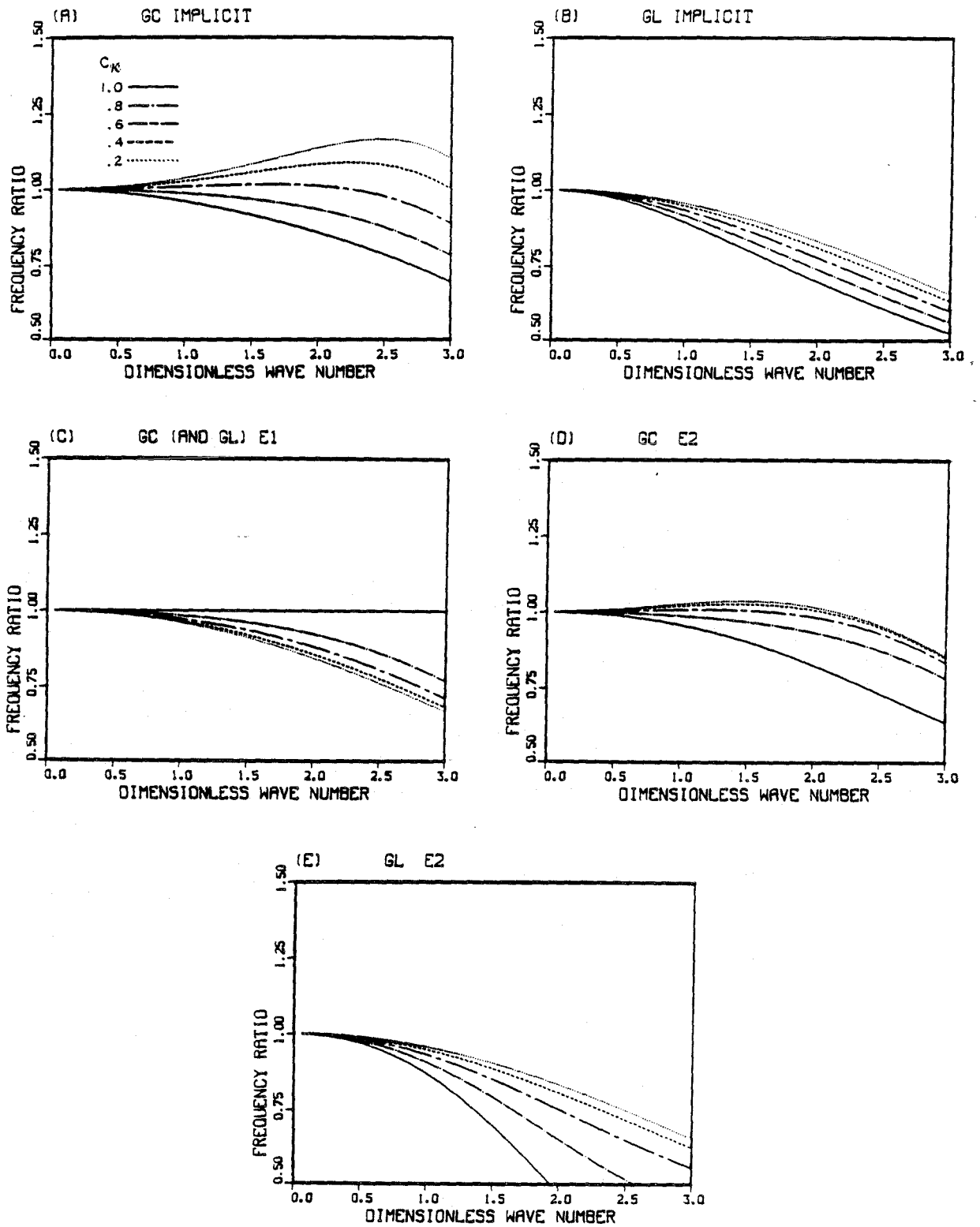


Figure III.1 Frequency ratios for wave equation algorithms.

implicit version until about $C_K = 0.8$. At this point they almost have the same accuracy, and after that the implicit version is more accurate. GC-E2 maximizes its accuracy between $C_K = 0.6$ and 0.8 .

REFERENCES

B1. A.J. Baker, Research on Numerical Algorithms for the Three-Dimensional Navier-Stokes Equations, I. Accuracy, Convergence and Efficiency, Technical Report AFFDL-TR-79-3141, Wright-Patterson Air Force Base, Ohio, December 1979.

B2. A.J. Baker and M.O. Soliman, "On the Utility of Finite Element Theory for Computational Fluid Dynamics,", Proceedings of the 5th AIAA Computational Fluid Dynamics Conference, June 1981, Palo Alto, California.

B3. A.J. Baker, Private Communication, 1981

B4. W.F. Ballhaus, A. Jameson and J. Albert, "Implicit Approximate-Factorization Schemes for Steady Transonic Flow Problems", AIAA Journal, Vol. 16, No. 6, 1978, pp. 573-579.

B5. J. Barton, Private Communications, 1982.

B6. R.M. Beam, R.F. Warming, and H.C. Yee, "Stability Analysis of Numerical Boundary Conditions and Implicit Difference Approximations for Hyperbolic Equations", Proceedings of Numerical Boundary Condition Procedures Symposium, October 19-20, 1981, Ames Research Center, NASA, Moffet Field, California.

B7. A. Brooks and T.J.R. Hughes, "Streamline Upwind/Petrov-Galerkin Methods for Advection Dominated Flows", Proceedings of the Third International Conference on Finite Element Methods in Fluid Flow, Banff, Canada, 1980.

B8. 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", to appear in the Proceedings of FENOMECH '81 and Computer Methods in Applied Mechanics and Engineering.

C1. I. Christie, D.F. Griffiths, A.R. Mitchell and O.C. Zienkiewicz, "Finite Element Methods for Second Order Differential Equations with Significant First Derivatives", International Journal for Numerical Methods in Engineering, Vol. 10, 1389-1396, 1976.

C2. R. Courant and K.O. Friedrich, Supersonic Flow and Shock Waves, Interscience Publishers, New York, 1948.

C3. R. Courant and D. Hilbert, Methods of Mathematical Physics, Interscience Publishers, New York, 1962.

D1. J.E. Dendy, "Two Methods of Galerkin Type Achieving Optimum L^2 Rates of Convergence for First Order Hyperbolics", SIAM Journal of Numerical Analysis, Vol. 11, pp. 637-653, 1974.

H1. J.C. Heinrich, P.S. Huyakorn, O.C. Zienkiewicz and A.R. Mitchell, "An 'Upwind' Finite Element Scheme for Two-Dimensional Convective Transport Equation", International Journal for Numerical Methods in Engineering, Vol. 11, pp. 134-143, 1977.

H2. H.M. Hilber, T.J.R. Hughes and R.L. Taylor, "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics", Earthquake Engineering and Structural Dynamics, Vol. 5, 283-292, (1977).

H3. T.L. Holst and W.F. Ballhaus, "Fast, Conservative Schemes for the Full Potential Equation Applied to Transonic Flows", AIAA Journal, Vol. 17, No. 2, February 1979, pp. 145-152.

H4. T.L. Holst and D. Brown, "Transonic Airfoil Calculations Using Solution-Adaptive Grids", Proceedings of the 5th AIAA Computational Fluid Dynamics Conference, June 1981, Palo Alto, California.

H5. T.L. Holst, Private Communications, 1982.

H6. T.J.R. Hughes, A study of the One-Dimensional Theory of Arterial Pulse Propagation, Report No. UC SESM 74-13, Structural Engineering Laboratory, University of California, Berkeley, 1974.

H7. T.J.R. Hughes, "Stability of One-Step Methods in Transient Nonlinear Heat Conduction", Transactions of the Fourth International Conference on Structural Mechanics in Reactor Technology, San Francisco, California, August 1977.

H8. T.J.R. Hughes and J. Atkinson, "A Variational Basis for 'Upwind' Finite Elements", IUTAM Symposium on Variational Methods in the Mechanics of Solids, Northwestern University, Evanston, Illinois, September, 1978.

H9. T.J.R. Hughes and W.K. Liu, "Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory", Journal of Applied Mechanics, Vol. 45, (1978), 371-374.

H10. T.J.R. Hughes and W.K. Liu, "Implicit-Explicit Finite Elements in Transient Analysis: Implementation and Numerical Examples", Journal of Applied Mechanics, Vol. 45, (1978), 375-378.

H11. T.J.R. Hughes, K.S. Pister and R.L. Taylor, "Implicit-Explicit Finite Elements in Nonlinear Transient Analysis", Computer Methods in Applied Mechanics and Engineering, 17/18 (1979) 159-182.

H12. T.J.R. Hughes and A. Brooks, "A Multidimensional Upwind Scheme with no Crosswind Diffusion", in AMD Vol. 34, Finite Element Methods for Convection Dominated Flows, T.J.R. Hughes (ed.), ASME, New York, 1979.

H13. T.J.R. Hughes, "Implicit-Explicit Finite Element Techniques for Symmetric and Nonsymmetric Systems", Proceedings, First International Conference on Numerical Methods for Non-Linear Problems, Swansea, U.K., 1980.

H14. T.J.R. Hughes and A. Brooks, "Galerkin/Upwind Finite Element Mesh Partitions in Fluid Mechanics", pp. 103-112, in Boundary and Interior Layers - Computational and Asymptotic Methods, J.J.H. Miller (ed.), Boole Press, Dublin, 1980.

H15. T.J.R. Hughes and A. Brooks, "A Theoretical Framework for Petrov-Galerkin Methods with Discontinuous Weighting Functions: Application to the Streamline Upwind Procedure", to appear in Finite Elements in Fluids, Vol. 4, R.H. Gallagher (ed.), J. Wiley and Sons.

H16. T.J.R. Hughes, "Analysis of Transient Algorithms with Particular Reference to Stability Behavior", to appear in Computational Methods in Transient Analysis, T. Belytschko and T.J.R. Hughes (eds.), North-Holland Publishing Company, Amsterdam.

H17. T.J.R. Hughes, I. Levit and J. Winget, "An Implicit Unconditionally Stable Element-by-Element Algorithm for Heat Conduction Analysis", to appear in Journal of the Engineering Mechanics Division, ASCE.

L1. P.D. Lax, "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation", Communications on Pure and Applied Mathematics, Vol. 7, pp. 159-193, 1954.

L2. P.D. Lax, "Hyperbolic Systems of Conservation Laws II", Communications on Pure and Applied Mathematics, Vol. 10, pp. 537-566, 1957.

L3. P.D. Lax "Nonlinear Hyperbolic Systems of Conservation Laws", Nonlinear Problems, R.E. Langer (ed.), pp. 3-12, Madison, The University of Wisconsin Press, 1963.

L4. H.W. Liepmann and A. Roshko, Elements of Gasdynamics, J. Wiley and Sons. New York, 1957.

L5. H. Lomax and E.D. Martin, "Fast Direct Numerical Solution of the Nonhomogenous Cauchy-Riemann Equations", Journal of Computational Physics, Vol. 15, No. 1, May 1974, pp. 55-80.

L6. H. Lomax, "Some Prospects for the Future of Computational Fluid Dynamics", Proceedings of the 5th AIAA Computational Fluid Dynamics Conference, June 1981, Palo Alto, California.

L7. H. Lomax et al., Private Communications, 1981-1982.

M1. G. Moretti, "A Physical Approach to the Numerical Treatment of Boundaries in Gas Dynamics", Proceedings of Numerical Boundary Condition Procedures Symposium, October 19-20, 1981 Ames Research Center, NASA, Moffet Field, California.

M2. K.W. Morton and J.W. Barrett, "Optimal Finite Element Methods for Diffusion-Convection Problems", in Boundary and Interior Layers-Computational and Asymptotic Methods, J.J.H. Miller (ed.) Boole Press, Dublin, pp. 134-148, 1980.

M3. K.W. Morton and A.K. Parrott, "Generalized Galerkin Methods for First-Order Hyperbolic Equations", Journal of Computational Physics, Vol. 36, No. 2, 1980.

N1. S. Nakazawa, Ph.D. Thesis, University of Swansea, to appear.

R1. W.H. Raymond and A. Garder, "Selective Damping in a Galerkin Method for Solving Wave Problems with Variable Grids", Monthly Weather Review, Vol. 104, pp. 1583-1590, 1976.

R2. R.D. Richtmyer and K.W. Morton, Difference Methods for Initial-Value Problems, Interscience Publishers, New York, 1967.

S1. J.O. Steger, Private Communications, 1981-1982.

S2. J.L. Steger and R.F. Warming, "Flux Vector Splitting of the Inviscid Gasdynamics Equations with Application to Finite Difference Methods", Journal of Computational Physics, Vol. 40, No. 2, April 1981.

S3. J.L. Steger, "A Preliminary Study of Relaxation Methods for the Inviscid Conservative Gasdynamics Equations Using Flux Splitting", NASA Contractor Report 3415, March 1981.

T1. E. Turkel, "Symmetrization of the Fluid Dynamic Matrices with Applications", Mathematics of Computation, Vol. 27, No. 124, October 1973, pp. 729-736.

W1. L.B. Wahlbin, "A Dissipative Galerkin Method for the Numerical Solution of First Order Hyperbolic Equations", pp. 147-169 in Mathematical Aspects of Finite Elements in Partial Differential Equations, Carl de Boor, (ed.), Academic Press, New York, 1974.

W2. R.F. Warming, R.M. Beam and B.J. Hyett, "Diagonalization and Simultaneous Symmetrization of the Gas-Dynamic Matrices", Mathematics of Computation, Vol. 29, No. 132, October 1975, pp. 1037-1045.

W3. R.F. Warming and R.M. Beam, "On the Costruction and Application of Implicit Factored Schemes for Conservation Laws", SIAM-AMS Proceedings, Vol. 11, 1978.

W4. H.F. Weinberger, A First Course in Partial Differential Equations, Ginn and Company, Waltham, Massachusetts, 1965.

Y1. H.C. Yee, "Numerical Approximation of Boundary Conditions with Applications to Inviscid Equations of Gas Dynamics", NASA TM 81265, March 1981.

Z1. O.C. Zienkiewicz, The Finite Element Method, McGraw-Hill, London, 1977.