OLD DOMINION UNIVERSITY RESEARCH FOUNDATION

DEPARTMENT OF MATHEMATICAL AND COMPUTING SCIENCES
SCHOOL OF SCIENCES AND HEALTH PROFESSIONS
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

A BLOCK ITERATIVE FINITE ELEMENT ALGORITHM FOR NUMERICAL SOLUTION OF THE STEADY-STATE, COMPRESSIBLE NAVIER-STOKES EQUATIONS

By

Charlie H. Cooke

(NASA-CR-148276) A BLOCK ITERATIVE FINITE ELEMENT ALGORITHM FOR NUMERICAL SOLUTION OF THE STEADY-STATE, COMPRESSIBLE NAVIER-STOKES EQUATIONS (Old Dominion Univ. Research Foundation) 34 p HC $4.00 CSCL 20D G3/34

N76-26422

Unclas
42324

Progress Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

Under
Research Grant NSG 1098
September 9, 1975 - June 30, 1976

July 1976

DEPARTMENT OF MATHEMATICAL AND COMPUTING SCIENCES
SCHOOL OF SCIENCES AND HEALTH PROFESSIONS
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

A BLOCK ITERATIVE FINITE ELEMENT ALGORITHM FOR NUMERICAL SOLUTION OF THE STEADY-STATE, COMPRESSIBLE NAVIER-STOKES EQUATIONS

By

Charlie H. Cooke

Progress Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Research Grant NSG 1098
September 9, 1975 - June 30, 1976
Dr. Julius E. Harris, Technical Monitor
High Speed Aerodynamics Division

Submitted by the
Old Dominion University Research Foundation
Norfolk, Virginia 23508

July 1976

CONTENTS

	Page
SUMMARY	1
INTRODUCTION	1
FLUID DYNAMICS MODEL OF A FREE SHEAR LAYER FLOW	4
Flow Field Configuration	4
Governing Equations	5
Boundary Conditions	6
FINITE ELEMENT APPROXIMATION	6
Trial Functions	6
Discretized Equations	7
Numerical Quadratures	9
A NONLINEAR BLOCK ITERATIVE GAUSS-SEIDEL SOLVER	10
NUMERICAL RESULTS FOR A FREE SHEAR LAYER FLOW	11
Case I. Time Independent Equations and Supersonic-Supersonic Inflow	11
Case II. Time Independent Equations and Subsonic-Supersonic Inflow	12
Case III. Time Dependent Equations and Subsonic-Supersonic Inflow	14
CONCLUSIONS	15
EPILOGUE	17
TABLES	18
FIGURES	21
REFERENCES	30

LIST OF TABLES

Table		Page
I.	Convergence Requirements for Various Computational Methods (Re = 1000 Supersonic-Supersonic Free Shear Layer Flow)	19
II.	Numerical Difference Between Finite Element and ADI Steady State Results (Supersonic-Supersonic Flow)	20
III.	Percent Differences Between Finite Element and ADI Steady State Results (Supersonic-Supersonic Flow)	21

LIST OF FIGURES

Figure		
1.	Flow Field Configuration, Supersonic-Supersonic Shear Layer Problem	22
2.	Boundary Conditions for Supersonic-Supersonic Flow	23
3.	Node Numbering Scheme for the C^{o}-Cubic Element	24
4.	Comparison of Finite Element and ADI Steady State Results; Supersonic-Supersonic Flow	25
5.	Boundary Conditions for Mixed Subsonic-Supersonic Flow	27
6.	Comparison of Time Transient Finite Element and ADI; Steady State Results; Mixed Subsonic-Supersonic Flow	28

A BLOCK ITERATIVE FINITE ELEMENT ALGORITHM FOR NUMERICAL SOLUTION OF THE STEADY-STATE, COMPRESSIBLE NAVIER-STOKES EQUATIONS

By

Charlie H. Cooke[1]

SUMMARY

An iterative method for numerically solving the time independent Navier-Stokes equations for viscous compressible flows is presented. The method is based upon partial application of the Gauss-Seidel principle in block form to the systems of nonlinear algebraic equations which arise in construction of finite element (Galerkin) models approximating solutions of fluid dynamic problems. The C^0-cubic element on triangles is employed for function approximation. Computational results for a free shear flow at Re = 1000 indicate signficant achievement of economy in iterative convergence rate over finite element and finite difference models which employ the customary time dependent equations and asymptotic time marching procedure to steady solution. Numerical results are in excellent agreement with those obtained for the same test problem employing time marching finite element and finite difference solution techniques.

INTRODUCTION

Historically, most numerical methods for obtaining steady-state solutions of the nonlinear Navier-Stokes equations of fluid dynamics are time-dependent methods in which the steady solution is approached asymptotically by time marching procedures. Prevalence of such methods derives from the confidence that is placed in eventual convergence to the steady solution (for stable and consistent algorithms), as well as

[1] Associate Professor, Department of Mathematical and Computing Sciences, Old Dominion University, Norfolk, Virginia 23508.

the ease and economy with which numerous finite difference methods may be formulated and implemented. However, in many situations the structure of the problem or stability restrictions of the algorithm can lead to the requirement of many time steps to convergence. Hence, it is desirable to devise methods in which iterative convergence is achieved as rapidly as possible.

In a series of investigations by Roache (refs. 1 and 2) several iterative methods have been devised for solving the incompressible 2D steady Navier-Stokes equations in stream functions and vorticity. These finite difference techniques are neither time dependent nor even timelike in their iterations. Instead, the nature of the iterations is such that otherwise nonlinear equations approximating time independent Navier-Stokes flows become linear in the independent variables over one iterative step. Recent advances in solving the resulting linear systems by direct methods (fast Poisson solvers and biharmonic solvers) are employed for economical equation solving. Improved iterative convergence rates have been experienced for the driven cavity problem and other low Reynolds number flows.

On the other hand, applications of the finite element method in fluid dynamics problems governed by the unsteady Navier-Stokes equations leaves one unconvinced such methods are at all competitive, in terms of economy of computer resources, with the standard finite difference techniques (refs. 3 and 4). A built-in cumbersomeness due to the general (gridwise) applicability of the method seems to indicate more complex program structure, more lengthy development and slower problem execution times are to be expected. Hence, in attempting to make the finite element method more nearly competitive with the state-of-the-art finite difference algorithms which now exist in rather streamlined form, one might logically consider iterative solution procedures for the steady equations, with expectations that order of magnitude improvements in the number of iterations to convergence could vastly improve the economy of the method.

Early investigations of this type on the 2D steady transonic flow equations (ref. 5) are encouraging. A reported comparison of the Galerkin method (with quadratic rectangular elements) versus finite difference results for the flow around a circular-arc airfoil with imbedded shock indicates finite element results which are superior to finite differences, while retaining a speed ratio of at least one order of magnitude. Here, of course, only one governing equation is involved, which greatly simplifies the numerical computation.

Attempted solutions by finite elements of steady problems involving the full Navier-Stokes equations have as yet been only sparsely reported. The case of viscous incompressible flow and laminar, steady, isothermal fluid motion in two dimensions has been investigated by Garling (ref. 3), using quadratic function approximation with isoparametric quadrilateral and triangular finite elements. The nonlinear algebraic equations resulting from the discretization process are solved with a Picard iteration of the form

$$A(\overline{X}_R)\ \overline{X}_{R+1} = f(\overline{X}_R) \quad ,$$

where $\overline{X}$ is a vector of all density and momentum variables at the nodes of the discretization. The initial flow field employed the solution to the creeping flow problem (zero Reynolds number) and iterative convergence was achieved in only a few iterations, for a broad class of viscous flow problems and a significant range of Reynolds number ($\leq$ 10,000). The matrix A and vector f were assembled by the frontal solution technique, generalized to the case of nonsymmetric matrices (essentially Gaussian elimination without pivoting). [Herein lies the weakness of this formulation, since in the present investigation for the case of mixed subsonic-supersonic compressible flow at Re = 1000 we have experienced significant round off when Gaussian elimination with no pivoting (Crouts Method) was used.]

Laskaris (ref. 6) has developed a finite element numerical technique whereby the steady state hydrodynamic equations for two-dimensional viscous compressible flows are solved, taking into full account the nonlinear convective terms, viscous terms, heat conduction terms, and variable fluid properties. The (Galerkin) method of weighted residuals is applied over distorted rectangular elements with cubic (Hermite, tensor product) function approximation. The resulting set of nonlinear algebraic equations for the nodal parameters are solved by means of a multi-dimensional Newton-Raphson scheme. For a heat transfer problem in a diverging channel with plane walls, with around 400 elements (roughly 2400 unknowns), convergence was achieved in 5 to 8 iterations and approximately two hours CPU time on the GE-600 computer. Direct Gaussian elimination and an out-of-core solver were used for matrix inversion at each iterative step.

In the present investigation viscous compressible flows governed by the two-dimensional steady Navier-Stokes equations in primitive variables form are considered. The goal of the investigation is to determine whether the finite element method becomes a more feasible tool for fluids computations (for which either time asymptotic or steady governing equation formulation is applicable) when the steady-state governing equations are adopted. Solutions of the same physical problem by both time transient finite element and finite difference methods affords a ready basis for comparison of these diverse techniques.

FLUID DYNAMICS MODEL OF A FREE SHEAR LAYER FLOW

A computer code for numerical computation of 2D viscous fluid flows governed by the steady-state compressible Navier-Stokes equations in primitive variable form has been developed. Proof of concept for this finite element program is provided by the computation of the solution to a free shear flow generated by the parallel mixing of two supersonic jets, initially separated by a thin splitter plate. Solution of the problem does not require the full Navier-Stokes equations, since fairly accurate results can be obtained using the quasi-parallel assumptions of parabolic boundary layer theory. However, the availability of solutions generated by several computational methods (refs. 4 and 7) affords a ready basis for evaluation of the finite element method.

Flow Field Configuration

The flow field configuration of the test problem considered is shown in figure 1. The computational domain begins downstream from the base of the splitter plates. Numerical computations have been performed for flow at Reynolds number 1000.

Governing Equations

Steady-state flow is obtained through solution of the time independent Navier-Stokes equations. The assumption of constant total temperature (adiabatic mixing) and two-dimensional flow yields the following non-dimensional systems of governing equations (non-conservative form):

Continuity

$$\rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + v \frac{\partial \rho}{\partial y} + u \frac{\partial \rho}{\partial x} = 0 \tag{1}$$

y-momentum

$$\rho \left(v \frac{\partial v}{\partial y} + u \frac{\partial v}{\partial x}\right) + \frac{\partial P}{\partial y} + \frac{4}{3R_e} \frac{\partial}{\partial y} \left(\mu \frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial y} \left(\frac{2\mu}{3R_e} \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial x} \left[\frac{\mu}{R_e} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] = 0 \tag{2}$$

x-momentum

$$\rho \left(v \frac{\partial u}{\partial y} + u \frac{\partial u}{\partial x}\right) + \frac{\partial P}{\partial x} + \frac{4}{3R_e} \frac{\partial}{\partial x} \left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial x} \left(\frac{2\mu}{3R_e} \frac{\partial v}{\partial y}\right) - \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] = 0 \tag{3}$$

Temperature relation

$$T = 1 - u^2 - v^2 \tag{4}$$

Constitutive relationship

Sutherland's viscosity law: $$\mu = T^{3/2} \left[\frac{T_s + 198.6}{T_s T + 198.6}\right] \tag{5}$$

$$\text{Perfect gas law:} \quad P = \rho \left(\frac{\gamma - 1}{2\gamma}\right) T. \tag{6}$$

In equation (5), μ, T are dimensionless, although T_s and the constant 198.6 (Sutherland's constant) are expressed in degrees Rankine. The variables used to non-dimensionalize eqs. (1) to (6) are presented in reference 7.

Boundary Conditions

Boundary conditions for the problem are shown schematically in figure 2. Function specifications are given for all three variables on the inflow; symmetry conditions apply at the bottom, and on the top function specification is made for velocity, zero normal derivative for density. On the outflow a computational boundary condition must be applied; this was chosen to be quadratic extrapolation.

FINITE ELEMENT APPROXIMATION

Our approximation to the fluid dynamics problem [equations (1) to (7) and boundary conditions of figure 2] is obtained by applying the classical Galerkin (or method of weighted residuals) in conjunction with finite elements. The first step is to triangulate the computational domain Ω with boundary Γ, and then consider piecewise polynomial approximating (trial) functions on this grid.

Trial Functions

Function approximation in all independent variables is accomplished by means of piecewise cubic trial functions on triangular elements. For a precise description of the element used, see reference 8. For purposes of illustration, the trial functions for approximating density variations are of the form

$$\rho(x,y) = \sum_{j=1}^{N} \rho_J \; \phi_J(x,y) \quad . \tag{7}$$

Here N is the total number of nodes. The functions $\{\phi_J\}$ comprise a local and interpolating basis for functions of the form exhibited by eq. (7) which are continuous and piecewise cubic polynomial on Ω with sectionally continuous first partial derivatives, but which are infinitely differentiable cubic on the interior of each triangle.

The weights ρ_J are chosen by Galerkin's methods. Thus, the final approximating function satisfies all boundary and initial conditions at problem nodes, and approximately satisfies the governing equations over the domain. For each trial function (density and two velocity components) there are ten nodes per triangle; triple nodes at triangle vertices, and a single node at the centroid (see fig. 3). The parameters ρ_J each associate with a distinct node, and represent approximations to function and first partial derivative values $\left(\rho, \frac{\partial \rho}{\partial x}, \frac{\partial \rho}{\partial y}\right)$ at vertices, and function values alone at the centroid.

The trial functions for velocity, defined similarly to those of density, are of the form

$$u(x,y) = \sum_{J=1}^{N} u_J \, \phi_J(x,y)$$

$$v(x,y) = \sum_{J=1}^{N} v_J \, \phi_J(x,y) \quad . \tag{8}$$

Discretized Equations

Consider a node at which a density independent variable ρ_J is not restricted by any boundary condition specification. The corresponding discretized finite element equation associated with ρ_J is determined by setting to zero the weighted residual obtained upon multiplying eq. (1) by the basis function ϕ_J and integrating the result over Ω. After shifting derivatives where possible onto the basis function (integrating by parts) and letting $\overline{\rho},\overline{u},\overline{v}$ be the vectors whose components consist of all nodal variables not restricted by any boundary condition, there results the system of equations:

$$C(\bar{u},\bar{v})\bar{\rho} = \bar{F}(\bar{u},\bar{v}) \tag{9}$$

for determing the density of unknowns $\bar{\rho}$. Here the typical element of C is specified by

$$C_{JK} = \iint_{\Omega} \phi_K \left(u \frac{\partial \phi_J}{\partial x} + v \frac{\partial \phi_J}{\partial y}\right) dA + \int_{\Gamma} \left[\phi_J \phi_K (udx - udy)\right] \tag{10}$$

and

$$F_J = -\sum_L C_{JL}\, \rho_L \quad . \tag{11}$$

The sum defining F_J is over all nodes at which ρ_L is known. The system of discretized equations for the velocity vectors $\bar{u},\bar{v}$ may be shown to have the functional form

$$\left[\begin{array}{c|c} zz(\bar{u},\bar{v},\bar{\rho}) & zR(\bar{u},\bar{v},\bar{\rho}) \\ \hline Rz(\bar{u},\bar{v},\bar{\rho}) & RR(\bar{u},\bar{v},\bar{\rho}) \end{array}\right] \left[\begin{array}{c} \bar{u} \\ \hline \bar{v} \end{array}\right] = \left[\begin{array}{c} \bar{G}\ (\bar{u},\bar{v},\bar{\rho}) \\ \hline \bar{H}\ (\bar{u},\bar{v},\bar{\rho}) \end{array}\right] \tag{12}$$

These equations are the result of multiplying eqs. (2) and (3) by ϕ_J and setting to zero the weighted residuals obtained upon integrating the results over Ω, employing integration by parts to shift (where second derivatives occur) higher derivatives onto the basis functions.

The matrix elements of the u-momentum matrices are specified by the equations:

$$zz_{JK} = \iint_{\Omega} \left\{-\rho\phi_J \left(v \frac{\partial \phi_K}{\partial y} + u \frac{\partial \phi_K}{\partial x}\right) + \mu \left[\frac{\partial \phi_J}{\partial x}\left(\frac{4}{3R_e}\frac{\partial \phi_K}{\partial x} + \frac{\partial \phi_K}{\partial y}\right) + \frac{\partial \phi_J}{\partial y}\frac{\partial \phi_K}{\partial y}\right]\right\} dA + \oint_{\Gamma} \left[\mu\phi_J \frac{\partial \phi_K}{\partial y} dx - \frac{4\mu}{3R_e}\phi_J \frac{\partial \phi_K}{\partial x} dy\right] \cdot \tag{13}$$

$$zR_{JK} = \iint_{\Omega} \mu \left\{ \frac{\partial\phi_J}{\partial x} \left(\frac{\partial\phi_K}{\partial x} - \frac{2}{3R_e} \frac{\partial\phi_K}{\partial y} \right) + \frac{\partial\phi_J}{\partial y} \frac{\partial\phi_K}{\partial x} \right\} dA$$

$$+ \oint_{\Gamma} \left[\mu\phi_J \frac{\partial\phi_K}{\partial x} dx + \frac{2\mu}{3R_e} \phi_J \frac{\partial\phi_K}{\partial y} dy \right] , \qquad (14)$$

$$G_J = \iint_{\Omega} P \frac{\partial\phi_J}{\partial x} dA - \oint_{\Gamma} \phi_J P \, dy - \sum_L (zz_{JL} u_L + zR_{JL} v_L) . \qquad (15)$$

The sum in eq. (15) is over all nodes at which u,v are specified by boundary conditions. The v-momentum equations may be obtained from eqs. (13) to (15) by interchanging the variables x with y; u with v and reversing algebraic sign on boundary integral terms. The indices J,K in eqs. (10) to (15) range over all nodes at which ρ,u,v are unknown.

To simplify accounting in the equation solving process, it is assumed that u_K,v_K are either both known or both unknown at each node. When one but not both is specified by boundary conditions, both are treated as unknown in the equation assembly process, and a corrected equation is inserted for the known component prior to equation solution. This artifice produces momentum matrices characterized by identical dimension, bandwidth, profile, and intraband distribution of zero and non-zero elements. Descriptors of any one matrix which must be stored then suffice for all.

Numerical Quadratures

The appearance of nonlinearities in the integrands of eqs. (10) to (15) requires numerical quadratures. In order to maintain the degree of accuracy naturally achievable with cubic function approximation, the analysis of Fix (ref. 9) which predicts a quadrature scheme exact for polynomials of total degree five is needed. These requirements are conservatively met by the 16-point scheme (ref. 10) used by the finite element program, which possesses seventh order accuracy. A 7-point fifth order scheme was also tested and found to yield approximately eight seconds per iterative step decrease in CPU time, with equivalent algorithm performance.

A NONLINEAR BLOCK ITERATIVE GAUSS-SEIDEL SOLVER

Consider now the problem of solving the systems (9) and (12) of nonlinear algebraic equations governing the approximating finite element solution. Although more rapid convergence might be expected, it is clear that a full Newton iteration on these equations would be characterized by coupling of both continuity and momentum variables, implying excessive core requirements for in-core equation solving. As well, this method would lead to a very time consuming equation assembly process, since the Jacobian matrix would have rather complicated equations describing its elements. Ordinary nonlinear point SOR is not readily applicable if a triangle by triangle assembly is desired; nor could convergence be readily guaranteed.

However, a partial block Gauss-Seidel iteration* which is linear at each iterative step and which uncouples the density and momentum solutions is readily designed. The equations proposed for this iteration are

$$C_n \bar{\rho}_{n+1} = \bar{F}_n \tag{16}$$

$$zz_n \bar{u}_{n+1} = G_{n,n+1} - zR_n \bar{v}_n \tag{17}$$

$$RR_n v_{n+1} = H_{n,n+1} - Rz_n u_n \quad . \tag{18}$$

Here $\bar{\rho}_{n+1}$ is used in assembling G,H, in order to update the momentum solution. One merit of this iteration is that only one system matrix at a time need be in memory during the equation solving. This greatly reduces core demands and avoids out-of-core solvers.

---

* A full block Gauss-Seidel iteration cannot be applied, assuming one wishes for purposes of economy to simultaneously assemble both momentum equations.

In the present program eq. (16) is assembled first, with the matrix C in core. The resulting solution vector $\bar{\rho}_{n+1}$ is input to the (simultaneous) assembly of eqs. (17) and (18) with RR in core and element matrices for zz on disk. Equation (18) is solved, then zz is assembled in core and eq. (17) is solved. The matrix inversions, for the supersonic free shear flow problem reported, were accomplished by Crout's method of LU decomposition.

NUMERICAL RESULTS FOR A FREE SHEAR LAYER FLOW

Case I. Time Independent Equations and Supersonic-Supersonic Inflow

Steady-state results for computational solution of the supersonic free shear flow problem [eqs. (1) to (7) and boundary conditions of fig. 2] are now presented. The test case computed employed Re = 1000. For a mesh consisting of 225 elements (696 nodes per independent variable) iterative convergence was achieved with 15 iterations of eqs. (16) to (18) and 2079 seconds of CPU time on the CDC-6600 computer. Program core storage requirements were 162 K_8. For the 16-point quadrature scheme approximately 120 seconds per iterative step of CPU time were required, with approximately 8 seconds per step decrease when the 7-point scheme was employed. Total dollar costs for this computer run were $462, as determined by the computer systems' accounting scheme. The convergence criterion used is

$$\text{Max}\left|\frac{\Delta f_n}{f_n}\right| < .001 \tag{19}$$

where $\Delta f_n = f_{n+1} - f_n$, and f is a function value of u, v, or ρ. (Generally, the derivative values are less accurately modelled and lag function values in convergence.)

For purposes of comparison these results may be considered in relation to those obtained for the same problem by various finite difference (ref. 7) and finite element approximations (ref. 4) applied to the time dependent Navier-Stokes equations. Each method was initialized with the same starting flow field, and accuracy of the results appears equivalent. Several indicators of computational efficiency for these methods are presented in table I. (The number of steps to convergence for the time transient finite element code is not the best obtainable since the maximum permissible step was not consistently applied, as for the finite difference runs. However, due to the expense of this method no attempts to rerun with maximum step were made.)

Figure 4 shows typical comparisons of steady state density and velocity variations at stations $x_1 = 0.75$ and $x_2 = .175$ in the flow (the streamwise extent of the computational domain is $0 \leq x \leq .225.$), for steady finite elements versus time transient ADI finite differences. Table II exhibits actual numerical differences between these computations. Table III presents percent differences with the ADI computations as base. Since the normal component of velocity is zero over much of the field, percent differences for this component are normalized with respect to the maximum value.

Case II. Time Independent Equations and Subsonic-Supersonic Inflow

The steady finite element code has also been applied to a free shear flow problem resulting from the mixing of a subsonic and a supersonic flow. For a description of this problem, refer to figure 1 with $M_1 = 0.11$, $M_2 = 3.00$, and the boundary conditions of figure 5.

Catastrophic failure of the method for this case resulted from the equation solving, arising from very ill-conditioned matrices and Gaussian elimination without pivoting. For example, a standard debugging procedure is to apply the code to a constant flow problem (say, $\rho = .07625$, $u = .8018$, $v = 0$; or $v = .8018$, $u = 0$, for y-direction flow) to see if this flow reproduces itself. With large meshes (around 300 triangles) thought necessary for the subsonic-supersonic case the flow did not reproduce in the u-component, for y-direction flow. It was determined that the system matrices possessed determinants of order of magnitude

$$\begin{aligned} \det(zz) &= 10^{-3081} \\ \det(RR) &= 10^{-3031} \qquad (20) \\ \det(C) &= 10^{-1775} \end{aligned}$$

This seems to indicate ill-conditioning of all, but with a much more severe occurrence for the u-momentum equations (observe that the ratio of det(zz) to det(RR) is $0(10^{-50})$.

As a further test for the ill-conditioning, the equation

$$zz\bar{u} = \bar{f} \tag{21}$$

was, by proper choice of $\bar{f}$, set to have a solution all of whose components assumed the value of 1. This system was then solved by Crout's method (standard finite element solver) and by a band solver which used partial pivoting. Partial pivoting gave the best results, but each solution was characterized by numerous values with no significant figures of accuracy. Neither solver produced diagnostics peculiar to a singular matrix, although the eqs. (20) indicate near singularity. Best results, although still unacceptable, were obtained from Crout's method with double precision inner product accumulation on the decomposition, with matrix rows scaled so as to have unity diagonal elements.

As a remark on how nearly singular a matrix can be and the systems (16) to (18) still be solvable by ordinary techniques, the computational success for the supersonic-supersonic case prevailed in spite of the characteristics of near singularity indicated by

$$\det(zz) \simeq \det(RR) \simeq 10^{-1700} \tag{22}$$

To make the steady finite element formulation a generally applicable tool, these ill-conditioning problems must be overcome, and the feasibility of the resulting method evaluated.

Case III. Time Dependent Equations and Subsonic-Supersonic Inflow

During the present work period the finite element code developed previously (ref. 4) for numerical solution of the time dependent Navier-Stokes equations has also been applied to the free shear flow resulting from the mixing of subsonic and supersonic streams.

Figures 6 shows comparison of the finite element results (after 600 steps, $\Delta t \simeq .01$) and steady state results for the central difference ADI code. Data output from the two codes compare well with the exception of the normal component of velocity, which exhibits a maximum difference of 4 to 6 percent (relative percent difference) near the top right corner of the flow field. Here the finite element computation had not completely converged. This local slow convergence is attributed to boundary condition errors on the top; the finite element flow domain was obtained by truncating a region 20 finite difference mesh increments in width from the top of the ADI domain. Boundary conditions were then supplied by ADI steady results at the finite element domain. Initially, a significant bulge in finite element data occurred in this region, with the rest of the field converged to steady state. After the ADI code had been run further in time with a more solid convergence check instituted, the top boundary conditions then supplied resulted in a rapid and significant decrease in the bulge occurring in finite element output, to the level now indicated. It is conjectured mesh refinement of the ADI domain would be required in order to produce boundary condition data sufficiently accurate to produce totally satisfactory global convergence of the finite element calculation. Further support for this conclusion is furnished by figure 8 of reference 7, which shows disagreement in the inviscid region between various finite difference solutions, implying inaccuracy near the top of the finite element domain.

The time transient finite element code was also applied to a high Reynolds number (Re = 80,625) supersonic-supersonic mixing problem (see figs. 1 and 2). At a time step of .01 catastrophic failure emerged rapidly (negative temperatures in 30 steps). At a time step of .001 the computation was running smoothly at 150 steps, with the usual convergence criteria satisfied. However, these convergence criteria appeared satisfied through most of the run; it was concluded the initial flow was also steady state or else the time span insufficient for significant changes to develop. From this result it appears high Reynolds number flows without shocks could probably be calculated with this code.

CONCLUSIONS

A finite element code for numerical solution of fluid flow problems characterized by the two-dimensional time independent Navier-Stokes equations has been developed. Proof of concept was provided by the calculation of the primitive flow variables for a free shear flow problem. Excellent numerical results were obtained in comparison to ADI and various other finite difference methods.

For the supersonic-supersonic free shear layer problem, order of magnitude improvement in iterative convergence rate (15 steps compared to over 100 steps) was achieved, in comparison to the previously developed time transient finite element code (ref. 4), with some improvement in storage (236 K_8 down to 162 K_8) over the most feasible version of this code. Moreover, reduction in number of equation terms due to the steady form of the governing equation, as well as the diverse natures of the two numerical processes, produced reductions in CPU time (154 seconds/iterative step down to 120 seconds/iterative step) and O/S calls (7166/step down to 1025/step). From all these factors there resulted a reduction in machine total dollar cost (as calculated by the CDC-6600 operating system's accounting routine) for obtaining the converged solution on the order of 20 to 1, not to mention the significant reduction in man-hours necessary for processing the multiplicity of computer runs required by the time transient code. Thus we may readily conclude that the steady-state finite element approach is far more efficient than the time transient formulation.

On the other hand, at this stage of the investigation, there apparently exists the drawback of a less general applicability of the steady formulation; the weakness exhibited (ill-conditioning) for the 300-element mesh and the subsonic-supersonic flow problem. For broad general applicability of the code, the problem of ill-conditioning has yet to be overcome, and the resulting feasibility of the method then evaluated. This problem appears to be related to the mesh size; certainly the system matrices become more nearly singular as the number of nodes increases, and roundoff effects have further room to propagate.

Moreover, even with the success of the steady finite element code in evidence when compared with the time transient finite element code, at this state of development this method is still not quite competitive with the better finite difference techniques. However, the gap has been significantly narrowed to the point where the finite element method can almost be considered a feasible alternative.

As regards the time transient finite element code, it appears to have provided a reliable computational tool, for all test problems to which it has been applied, at the expense of much too heavy demands on computer resources.

EPILOGUE

Several factors contribute unnecessarily to a broadening of the gap between finite element and finite difference results. For example, it has been determined that equation assembly time in comparison to equation solving time per step is highly unbalanced on the side of equation assembly time. This imbalance could be lessened several ways:

1. Triangular elements were employed solely for general applicability of the finished code to other than rectangular regions. However, the code is being measured for competitiveness using a test problem whose domain is a rectangular region. For such a case, the (tensor product) Hermite cubic shape functions on rectangular elements would lead to more sufficient equation assembly time in at least two ways--fewer* (one-half) as many elements to process for the same mesh accuracy; and the existence of more efficient quadrature schemes for rectangles than for triangles (the seventh order accuracy afforded by the 16-point triangle scheme is afforded for a rectangle by an 8- or 9-point scheme). Finally, significant finite element results thus far (refs. 3, 5, and 6) have been with rectangular elements.

2. A scheme originally intended to improve the efficiency of the method actually degrades it, on the momentum equation assembly--for example, integrals like

$$\iint_{\Omega} \rho uv \, dA \qquad (23)$$

can be evaluated each iterative step from ρ,u,v values at the quadrature points (which must be computed anyway to determine viscosity) by applying the quadrature scheme directly to eq. (23), or the integrals

* Two triangles composing one rectangle.

$$\iint_{\Omega} \phi_J \phi_K \phi_L \, dA \tag{24}$$

may be computed once and stored, for future computations at each step of the form

$$\iint_{\Omega} \rho uv \, dA = \sum_J \rho_J \left(\sum_K u_K \left(\sum_L v_L \cdot \iint_{\Omega} \phi_J \phi_K \phi_L \, dA \right) \right) \tag{25}$$

For an integrand with an independent variable product of degree higher than two the second scheme is less efficient than the first, assuming the 16-point quadrature scheme. Consequently, in terms of number of multiplications necessary the momentum equations assembly suffers from inefficiency, possibly a significant amount.

3. It is conjectured that a linear element code would be more efficient on the equation assembly. Here only a 1-point quadrature scheme is necessary for the accuracy needed (ref. 9), a great simplification. However, more (triangles) elements would be required. One could only guess whether the iteration scheme would converge as well and what effect alternate boundary condition implementations would cause.

TABLE I. - CONVERGENCE REQUIREMENTS FOR VARIOUS COMPUTATIONAL METHODS
(Re = 1000, SUPERSONIC-SUPERSONIC FREE SHEAR LAYER FLOW)

Method		Best Number of Steps to Convergence	Core Storage (K_8)	CPU Time Seconds/step/node
Steady Equations	Cubic Finite Elements	15	162	0.172
Time Transient Equations	Cubic Finite Elements *	150 *	236	.229
	ADI with Centered Space Differences	62	107	.00374
	ADI with Upwind Space Differences	30	107	.00374
	Hopscotch	238	55	.00104

* Not best number of steps, since maximum allowable step was not consistently used.

TABLE II - NUMERICAL DIFFERENCE BETWEEN FINITE ELEMENT AND ADI STEADY STATE RESULTS (SUPERSONIC-SUPERSONIC FLOW)

$\rho_{FEM}-\rho_{ADI}$				$u_{FEM}-u_{ADI}$				$v_{FEM}-v_{ADI}$			
Density, $\Delta\rho$				Streamwise velocity, Δu				Normal velocity, Δv			
x = 0	.075	.175	.225	x = 0	.075	.175	.225	x = 0	.075	.175	.225
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
.000000	.000000	.000002	.000003	.000000	.000003	-.000007	.000002	.000000	-.000006	-.000002	.000000
.000000	-.000001	.000001	.000002	.000000	-.000004	.000001	-.000008	.000000	-.000004	-.000005	-.000003
.000000	-.000001	.000000	.000001	.000000	-.000001	-.000001	-.000007	.000000	-.000003	.000001	-.000005
.000000	-.000001	.000000	.000000	.000000	.000000	.000004	-.000008	.000000	-.000002	-.000005	.000002
.000000	-.000001	-.000001	-.000001	.000000	.000001	-.000001	-.000003	.000000	-.000002	-.000004	-.000005
.000000	-.000001	-.000001	-.000001	.000000	.000002	.000001	.000001	.000000	-.000002	-.000003	-.000004
.000000	-.000001	-.000001	-.000001	.000000	.000002	.000003	.000003	.000000	-.000002	-.000003	-.000004
.000000	-.000001	-.000001	-.000001	.000000	.000003	.000005	.000007	.000000	-.000002	-.000003	-.000005
.000000	-.000001	-.000002	-.000002	.000000	.000004	.000008	.000002	.000000	-.000002	-.000005	-.000007
.000000	-.000001	-.000002	-.000003	.000000	.000005	.000004	.000012	.000000	-.000002	-.000003	-.000002
.000000	-.000001	-.000003	-.000004	.000000	-.000003	.000005	.000022	.000000	-.000003	-.000002	-.000003
.000000	-.000002	-.000004	.000002	.000000	-.000001	.000008	.000028	.000000	-.000004	.000004	-.000003
.000000	-.000002	-.000007	-.000004	.000000	-.000004	.000005	.000024	.000000	.000002	.000003	-.000009
.000000	-.000002	-.000002	-.000004	.000000	.000004	.000003	.000000	.000000	-.000002	.000007	-.000013
.000000	-.000002	-.000003	.000003	.000000	.000000	-.000011	-.000024	.000000	.000000	.000007	-.000023
.000000	-.000007	-.000003	.000000	.000000	-.000002	-.000015	-.000073	.000000	.000009	.000006	-.000042
.000000	.000003	.000005	.000009	.000000	.000011	-.000009	-.000096	.000000	.000011	-.000017	-.000070
.000000	-.000001	-.000002	-.000001	.000000	.000022	.000023	-.000053	.000000	.000003	-.000058	-.000103
.000000	.000008	-.000002	-.000013	.000000	.000033	.000078	.000105	.000000	-.000030	-.000123	-.000090
.000000	.000004	-.000019	-.000042	.000000	.000028	.000119	.000336	.000000	-.000086	-.000162	-.000038
.000000	-.000018	-.000034	-.000063	.000000	-.000013	.000107	.000444	.000000	-.000100	-.000101	.000014
.000000	-.000031	-.000038	-.000053	.000000	-.000027	.000059	.000356	.000000	-.000001	.000044	.000048
.000000	-.000022	-.000033	-.000042	.000000	-.000005	.000041	.000144	.000000	.000130	.000188	.000103
.000000	.000007	-.000008	-.000002	.000000	.000034	.000063	-.000001	.000000	.000106	.000206	.000169
.000000	.000036	.000049	.000049	.000000	.000038	.000074	.000000	.000000	-.000017	.000073	.000117
.000000	.000020	.000050	.000062	.000000	-.000007	.000022	.000034	.000000	-.000052	-.000074	-.000030
.000000	-.000003	.000014	.000023	.000000	-.000027	-.000040	.000000	.000000	-.000020	-.000069	-.000552
.000000	-.000005	-.000013	-.000008	.000000	-.000016	-.000033	-.000031	.000000	.000003	-.000016	-.000028
.000000	-.000004	-.000010	-.000008	.000000	-.000009	-.000020	-.000021	.000000	.000003	.000007	.000004
.000000	-.000005	-.000007	-.000001	.000000	-.000002	-.000007	-.000008	.000000	.000006	.000011	.000009
.000000	-.000004	-.000001	.000003	.000000	.000000	-.000004	-.000001	.000000	-.000004	.000004	.000001
.000000	-.000004	-.000004	-.000003	.000000	-.000004	-.000003	-.000003	.000000	.000000	-.000002	.000006
.000000	-.000004	-.000004	-.000004	.000000	-.000004	-.000004	-.000004	.000000	.000000	.000000	.000000
.000000	-.000004	-.000004	-.000004	.000000	-.000004	-.000004	-.000004	.000000	.000000	.000000	.000000
.000000	-.000004	-.000004	-.000004	.000000	-.000004	-.000004	-.000004	.000000	.000000	.000000	.000000
.000000	-.000004	-.000004	-.000004	.000000	-.000004	-.000004	-.000004	.000000	.000000	.000000	.000000
.000000	-.000004	-.000004	-.000004	.000000	-.000004	-.000004	-.000004	.000000	.000000	.000000	.000000
.000000	-.000004	-.000004	-.000004	.000000	-.000004	-.000004	-.000004	.000000	.000000	.000000	.000000

TABLE III. - PERCENT DIFFERENCE BETWEEN FINITE ELEMENT AND ADI STEADY STATE RESULTS (SUPERSONIC-SUPERSONIC FLOW)

Density, ρ				Streamwise velocity, u				Normal velocity, v			
x = 0	.075	.175	.225	x = 0	.075	.175	.225	x = 0	.075	.175	.225
0.000	0.000	0.007	0.009	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
.000	.000	.005	.007	.000	.000	.001	.000	.000	.008	.003	.000
.000	.002	.002	.005	.000	.001	.000	.001	.000	.005	.006	.004
.000	.002	.000	.002	.000	.000	.000	.001	.000	.004	.001	.006
.000	.002	.000	.000	.000	.000	.001	.001	.000	.003	.006	.003
.000	.002	.002	.002	.000	.000	.000	.000	.000	.003	.005	.006
.000	.002	.002	.002	.000	.000	.000	.000	.000	.003	.004	.005
.000	.002	.002	.002	.000	.000	.000	.000	.000	.003	.004	.005
.000	.002	.002	.002	.000	.000	.001	.001	.000	.003	.004	.006
.000	.002	.005	.005	.000	.001	.001	.000	.000	.003	.006	.009
.000	.002	.005	.007	.000	.001	.001	.002	.000	.003	.004	.003
.000	.002	.007	.009	.000	.000	.001	.004	.000	.004	.003	.004
.000	.005	.009	.005	.000	.000	.001	.005	.000	.005	.005	.004
.000	.005	.016	.009	.000	.001	.001	.004	.000	.003	.004	.012
.000	.005	.005	.009	.000	.001	.000	.000	.000	.003	.009	.017
.000	.005	.007	.007	.000	.000	.002	.004	.000	.000	.009	.030
.000	.016	.007	.000	.000	.000	.002	.012	.000	.012	.008	.054
.000	.007	.012	.021	.000	.002	.001	.016	.000	.014	.022	.090
.000	.002	.005	.002	.000	.004	.004	.009	.000	.004	.075	.132
.000	.019	.005	.030	.000	.005	.013	.017	.000	.039	.158	.116
.000	.009	.043	.094	.000	.005	.019	.054	.000	.111	.208	.049
.000	.040	.074	.136	.000	.002	.017	.069	.000	.129	.130	.018
.000	.065	.078	.108	.000	.004	.009	.054	.000	.001	.057	.062
.000	.043	.063	.080	.000	.001	.006	.021	.000	.167	.242	.132
.000	.012	.014	.003	.000	.005	.009	.000	.000	.136	.265	.217
.000	.056	.077	.077	.000	.005	.010	.000	.000	.022	.094	.150
.000	.028	.072	.089	.000	.001	.003	.004	.000	.067	.095	.039
.000	.004	.019	.031	.000	.003	.005	.000	.000	.026	.089	.710
.000	.007	.017	.011	.000	.002	.004	.004	.000	.004	.021	.036
.000	.005	.013	.011	.000	.001	.002	.003	.000	.004	.009	.005
.000	.007	.009	.001	.000	.000	.001	.001	.000	.008	.014	.012
.000	.005	.001	.004	.000	.000	.000	.000	.000	.005	.005	.001
.000	.005	.005	.004	.000	.000	.000	.000	.000	.000	.003	.008
.000	.005	.005	.005	.000	.000	.000	.000	.000	.000	.000	.000
.000	.005	.005	.005	.000	.000	.000	.000	.000	.000	.000	.000
.000	.005	.005	.005	.000	.000	.000	.000	.000	.000	.000	.000
.000	.005	.005	.005	.000	.000	.000	.000	.000	.000	.000	.000
.000	.005	.005	.005	.000	.000	.000	.000	.000	.000	.000	.000
.000	.005	.005	.005	.000	.000	.000	.000	.000	.000	.000	.000

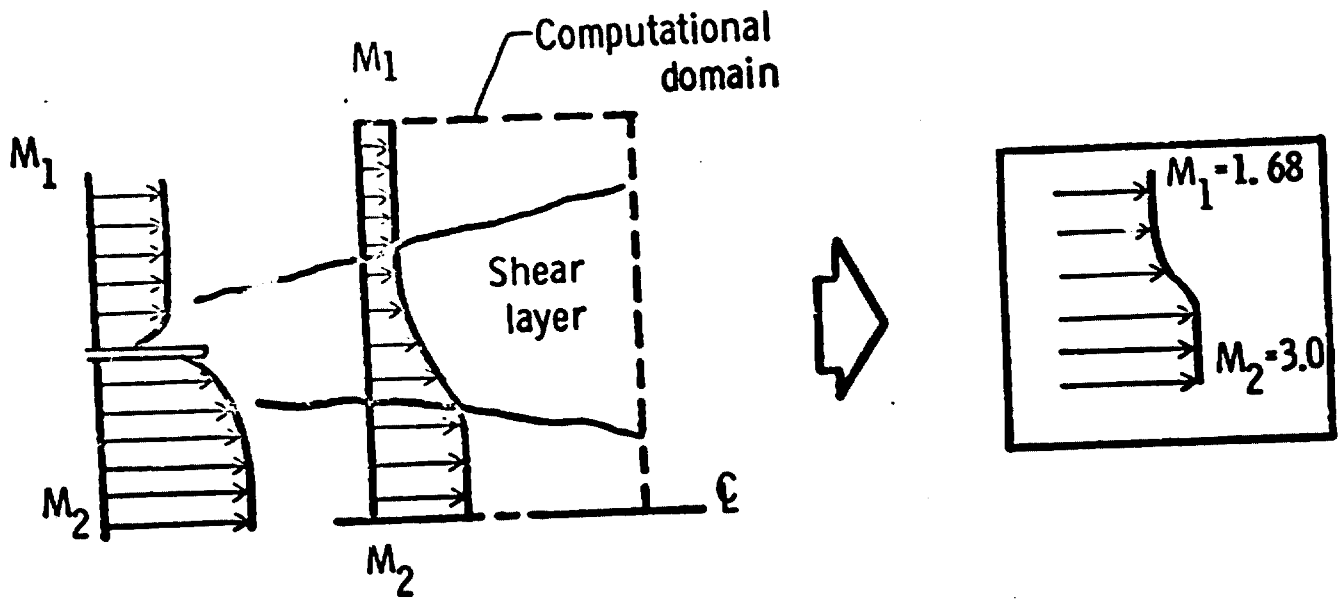


Figure 1. - Flow field configuration, supersonic-supersonic shear layer problem.

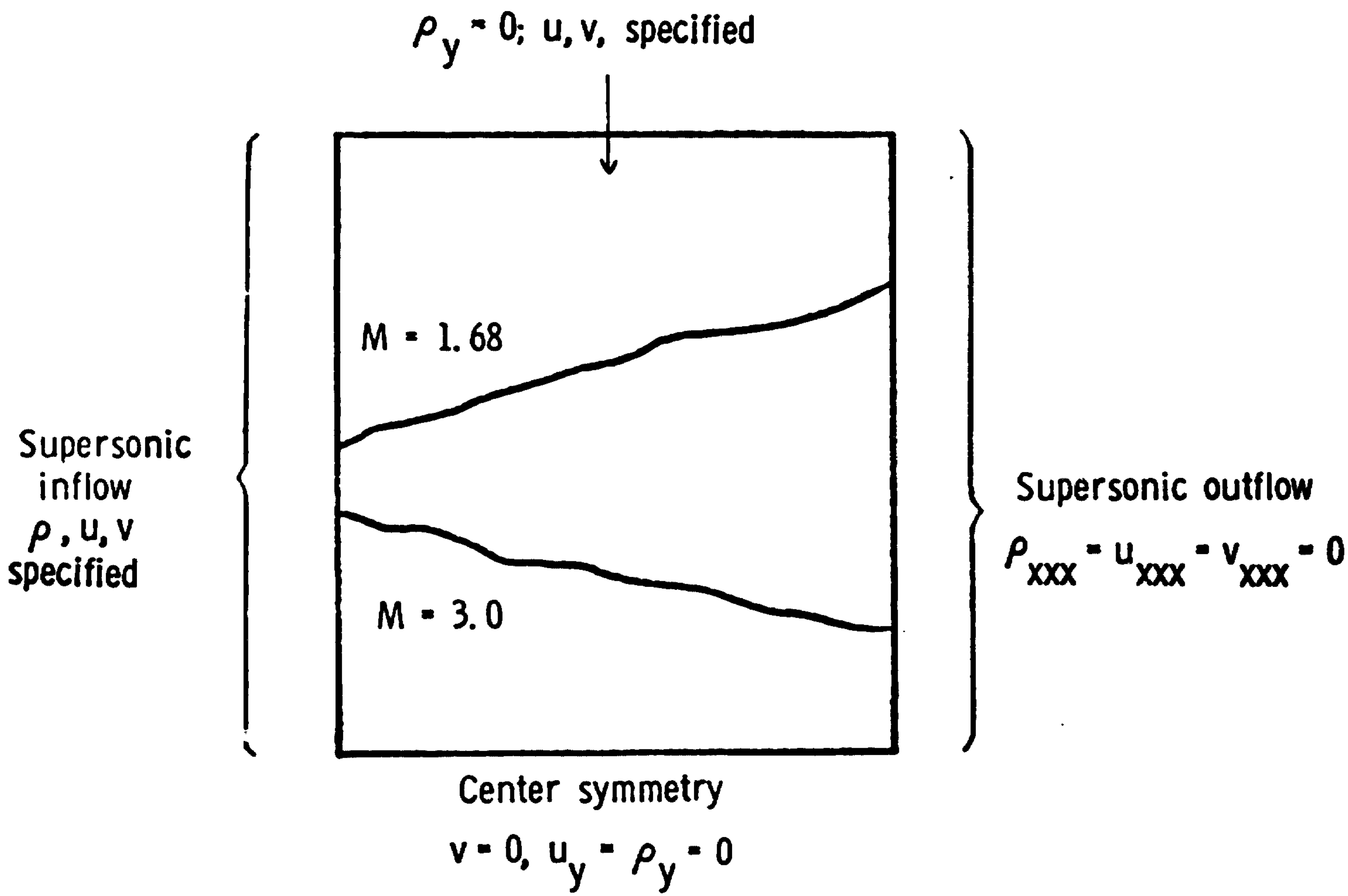


Figure 2. - Boundary conditions for supersonic-supersonic flow.

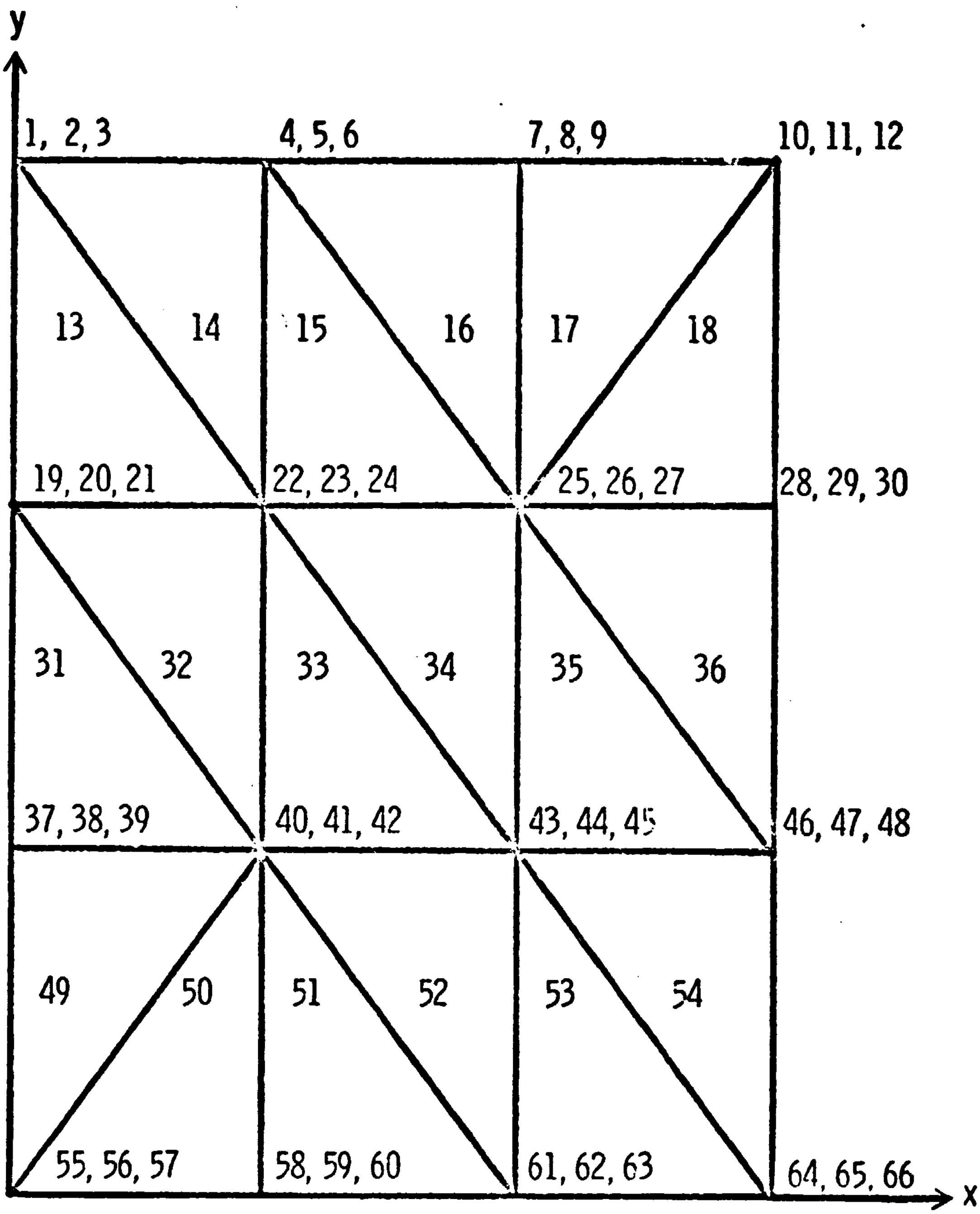
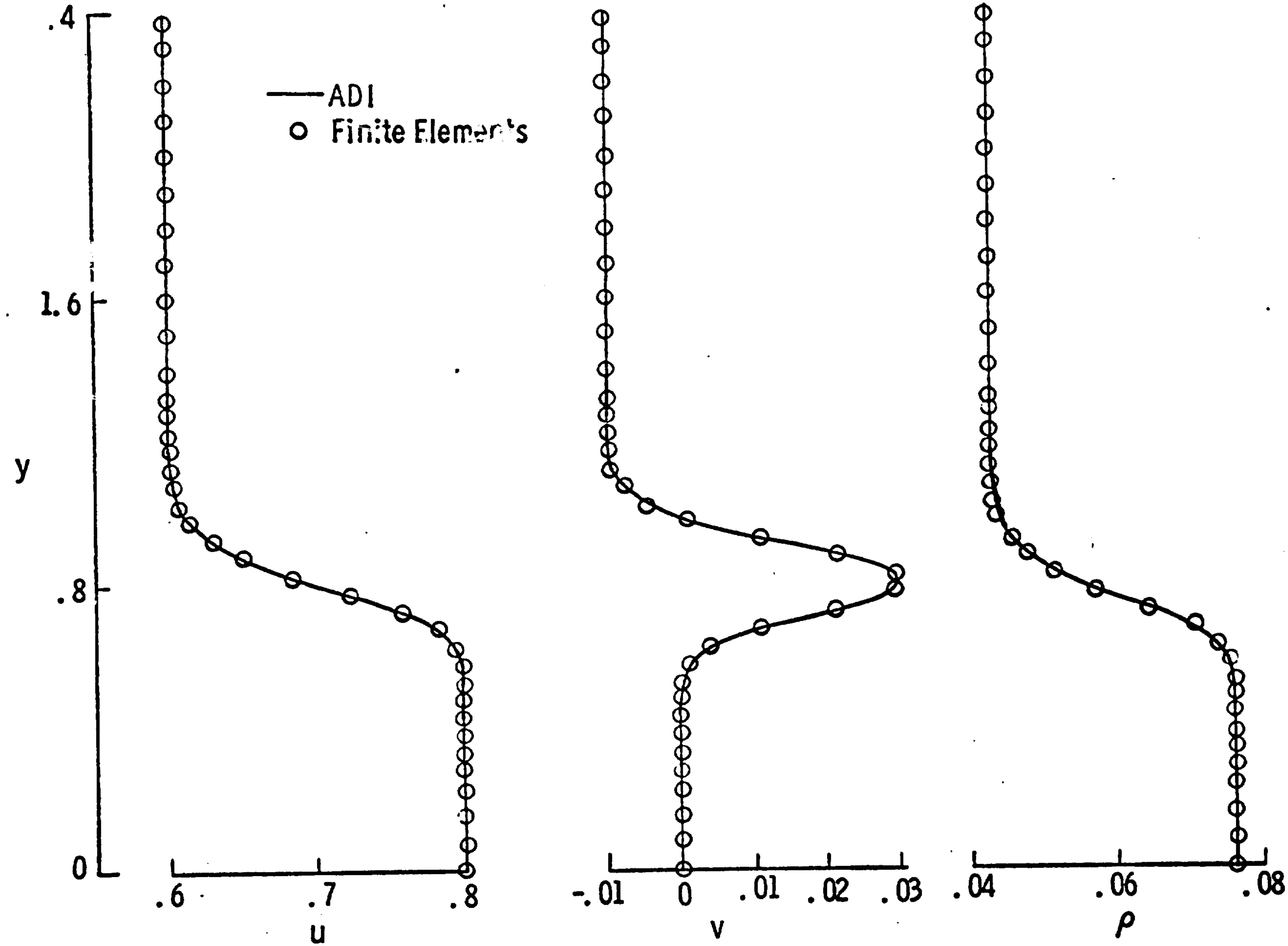
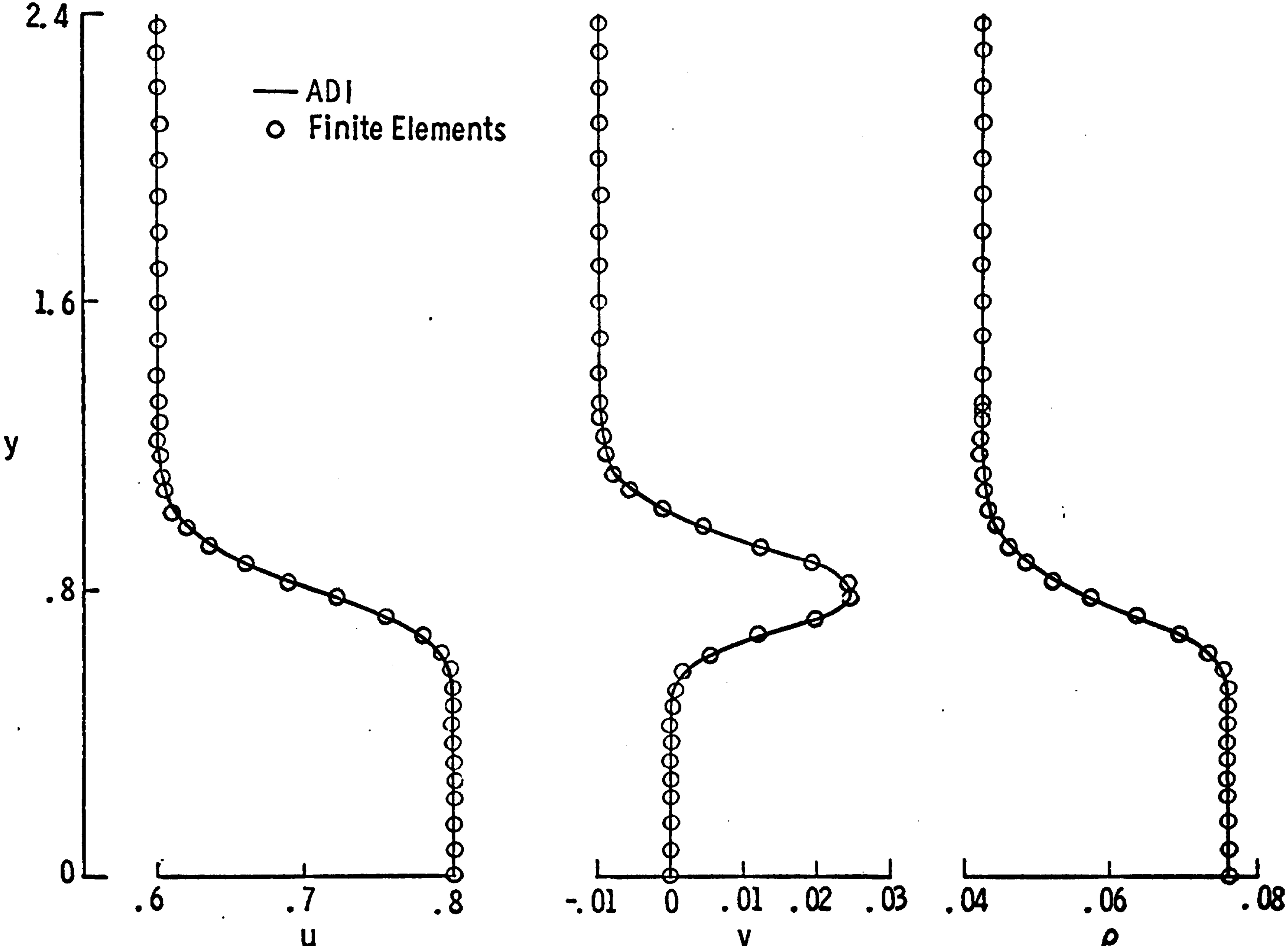


Figure 3. - Node numbering scheme for the C^0 cubic element.



(a) x = 0.075

Figure 4. - Comparison of Finite Element and ADI steady state results; supersonic-supersonic flow.



(b) x = 0.175

Figure 4. - Concluded.

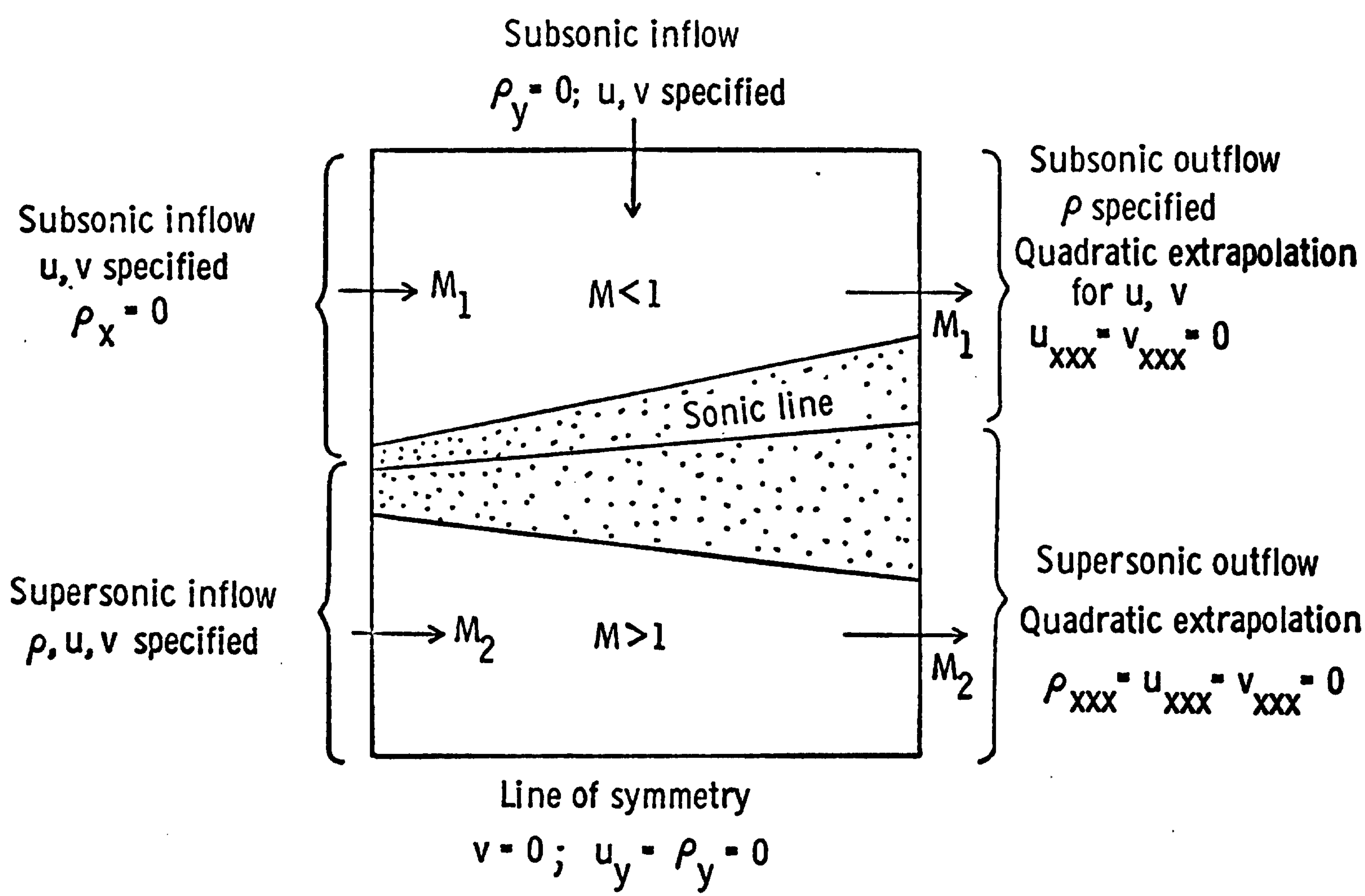


Figure 5. - Boundary conditions for mixed subsonic-supersonic flow.

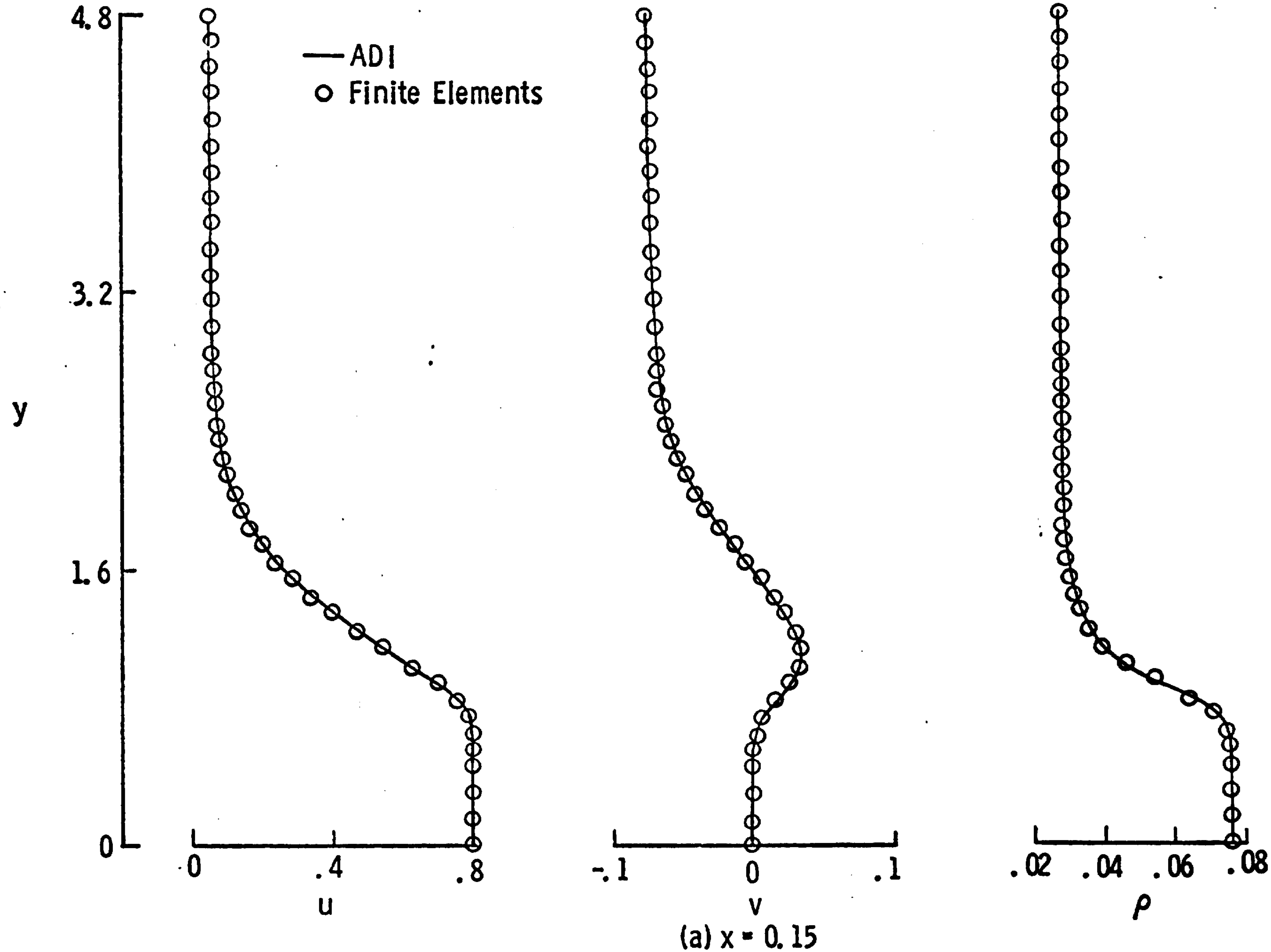
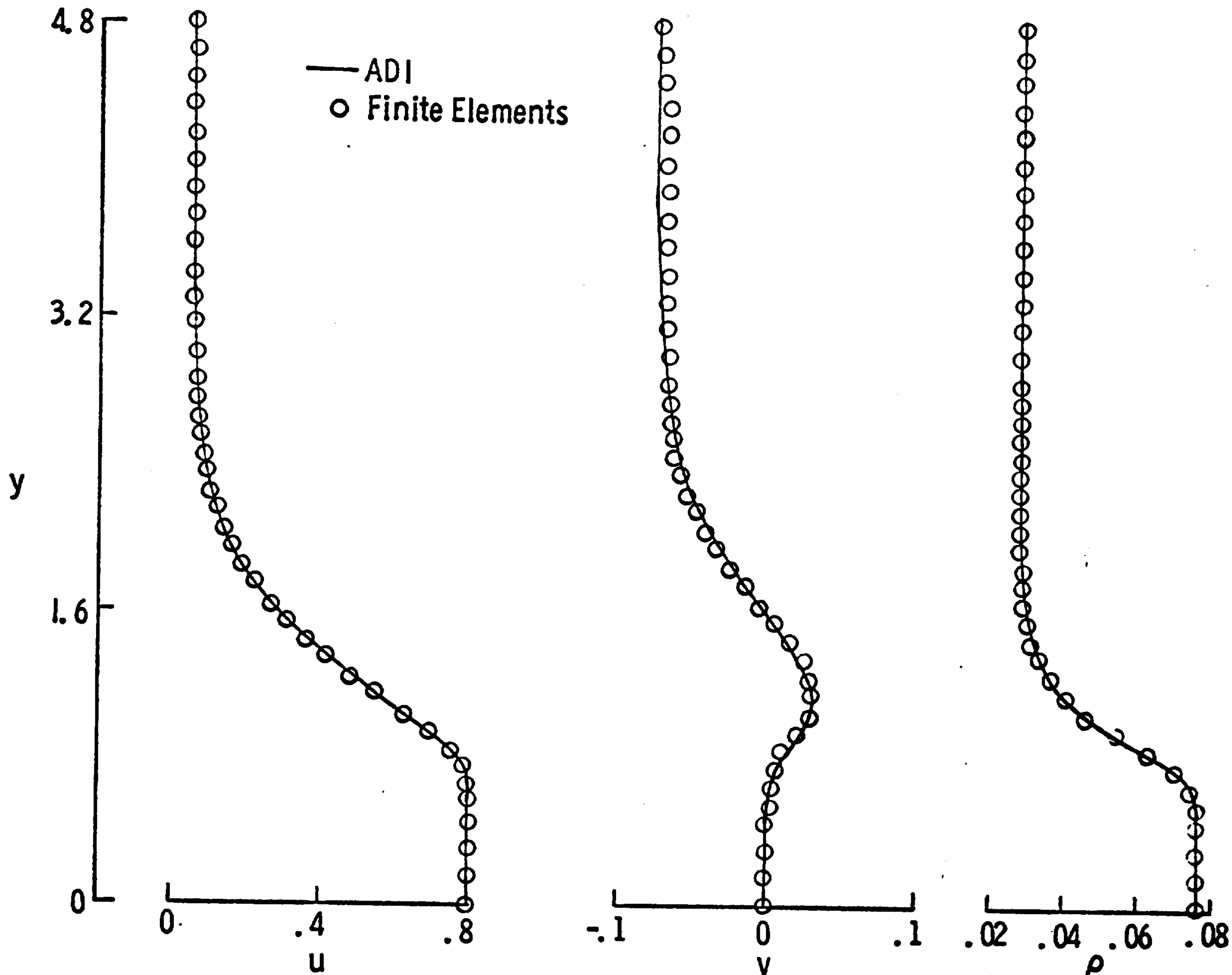


Figure 6. - Comparison of time transient Finite Element and ADI steady state results; mixed subsonic-supersonic flow.



(b) x = 0.35

Figure 6. - Concluded.

REFERENCES

1. Roache, Patrick J., "The LAD, NOS, and Split NOS Methods for the Steady-State Navier-Stokes Equations", Computers and Fluids, Vol. 3, pp. 179-195. Pergamon Press, 1975.

2. Roache, Patrick J., "The BID Method for the Steady State Navier-Stokes Equations", Computers and Fluids, Vol. 3, pp. 305-320. Pergamon Press, 1975.

3. Gartling, David K., "Finite Element Analysis of Viscous Incompressible Fluid Flow", Ph.D. dissertation, Department of Aerospace Engineering, University of Texas at Austin, December 1974 (TICØM Report 74-8).

4. Cooke, C.H. and Blanchard, D.K., "Finite Element Computation of a Viscous Compressible Free Shear Flow Governed by the Time Dependent Navier-Stokes Equations", Technical Report prepared for NASA-Langley Research Center under grant NSG 1098, Department of Mathematical and Computing Sciences, Old Dominion University, December 1975.

5. Brashears, M.R., Chan, S.T.K., and Young, V.Y.C., "Finite Element Analysis of Transonic Flow", Paper No. L-6, Proceedings, International Conference on Computational Methods in Nonlinear Mechanics, Huntsville, Alabama, September 1974.

6. Laskaris, Trifòn E., "Finite Element Analysis of Compressible and Incompressible Viscous Flow and Heat Transfer Problems", The Physics of Fluids, Vol. 18, No. 12, December 1975, pp. 1639-1648.

7. Rudy, David H. and Morris, Dana J., "Time Asymptotic Solutions of the Navier-Stokes Equations for Free Shear Flows Using an ADI Method", National Aeronautics and Space Administration, Langley Research Center, Technical Note TN D-8217, Hampton, Virginia, November 1975.

8. Fix, G.J. and Strang, G., An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs, NJ, 1973.

9. Fix, George J., "Effects of Quadrature Errors in Finite Element Approximations of Steady State, Eigenvalue, and Parabolic Problems", Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations, Academic Press, New York, NY, 1972, pp. 525-556.

10. Stroud, A.H., Approximate Calculation of Numerical Integrals, Prentice-Hall, Englewood Cliffs, NJ, 1971.