# A Finite Element Algorithm for Computational Fluid Dynamics

A. J. Baker* and M. O. Soliman†
*The University of Tennessee, Knoxville, Tennessee*

**An implicit finite element numerical algorithm is derived for the compressible Navier-Stokes equations expressed in generalized coordinates. The theoretical basis utilizes a Galerkin weighted residuals formulation and extremization of approximation error within the context of a multipole expansion. A simplified von Neumann analysis indicates the algorithm fourth-order phase accurate with third-order dissipation for the elementary linear basis construction. Algorithm performance is nominally improved using a quadratic basis. Numerical experiments for shocked duct flows are employed to refine the several algorithm parameters.**

## Introduction

ASSESSMENT of the impact potential for finite element theory and practice on the construction of numerical solution algorithms for computational fluid dynamics (CFD) is under study. The formal elegance of the methodology has produced a sound theoretical basis for the comprehensive simulation capabilities now in existence throughout structural mechanics.[1] The potential for such an impact in fluid dynamics remains to be verified.

The past decade has witnessed greatly expanded development of numerical solution algorithms and techniques for problem classes in CFD, in particular aerodynamics. In 1969 MacCormack[2] published an explicit, predictor-corrector finite difference algorithm for the Euler equations, which has since enjoyed worldwide acceptance and use. Using a split-operator technique, this truly elementary construction yields a second-order accurate (in space and time) approximation to the Euler equations. The addition of artificial diffusion permits prediction of shocked flows, and the basic theoretical formulation has enjoyed many refinements.

For viscous or turbulent flows, the explicit Euler algorithm becomes severely penalized by the "stiffness" of the discrete Navier-Stokes approximation. Modifications have been formulated,[3,4] but interest has generally shifted to implicit formulations, exemplified by the finite difference algorithms of Beam and Warming[5] and Briley and McDonald,[6] for an "approximate factorization" formulation. This algorithm concept is specifically addressed to the steady-state, compressible flow equation system, but could be extended to the transient problem. The addition of artificial diffusion is required for stability and the algorithm can simulate shocked flows. The extension of this finite difference algorithm to a generalized coordinates description,[7] utilizing grid generation and solution concepts pioneered by Thompson and coworkers,[8,9] renders the algorithm applicable to practical aerodynamic configurations.

Most recently, the formality of finite element function theoretic concepts has been applied to algorithm constructions in CFD.[10-12] While in its infancy compared to the extensive development of difference algorithms, the methodology does appear to yield robust algorithm formulations. In its elementary interpretation, the finite element concept returns calculus and vector field theory to the construction of discrete simulation algorithms for any branch of mechanics. Of necessity, using Taylor series expansions, one must be able to verify the equivalent (finite difference) order of accuracy for elementary derivatives within the governing equation system. Specifically, linear (quadratic) finite element bases yield, respectively, second- (fourth-) order finite difference representations for linear spatial derivatives. For other than linear space derivatives, however, the finite element construction usually yields expressions that are not so familiar. However, upon dissection they always can be related to appropriate Taylor series expansions. The important issue is that the theory produces the discrete analog expressions completely independent of the a posteriori ability to construct an equivalent difference representation.

In fluid mechanics, confidence in the theoretical statement can be attained only through detailed numerical assessments of progressively more complicated and pertinent differential equation descriptions. Strict adherence to convergence theory has been verified for scalar parabolic partial differential equations[10] using linear, quadratic, and cubic Lagrange and Hermite finite element bases. Agreement with linear convergence theory concepts is documented for solutions of the mildly nonlinear, parabolic laminar boundary-layer equations[11] for both linear and quadratic bases. Similar results for the consequentially nonlinear parabolic turbulent boundary-layer equations is also reported.[12] Of primary theoretical interest here, the Sobolev norm used to quantize convergence was a strongly nonlinear function, yet the theory accurately predicted algorithm performance.

An implicit finite element numerical solution algorithm has been derived for the unsteady three-dimensional Navier-Stokes equations expressed in generalized coordinates. The solution statement is applicable to inviscid, viscous, and/or turbulent flowfield descriptions, upon specification of the stress tensor. The theoretical statement utilizes a Galerkin weighted residuals formulation and extremization of semidiscrete approximation error within a multipole expansion. A von Neumann linearized stability analysis indicates the basic algorithm fourth-order accurate in the large Reynolds number limit. Numerical experiments for shocked, ducted flows quantize solution accuracy and convergence with respect to several identified algorithm parameters.

## Problem Description

The partial differential equation set governing transient, three-dimensional aerodynamic flows is the familiar and very nonlinear Navier-Stokes system. In nondimensional conservation form, using Cartesian tensor summation notation, the equation system for a compressible, viscous, heat-conducting fluid is

$$L(\rho)=\frac{\partial\rho}{\partial t}+\frac{\partial}{\partial x_j}[u_j\rho]=0 \tag{1}$$

Presented as Paper 81-1031 at the AIAA Fifth Computational Fluid Dynamics Conference, Palo Alto, Calif., June 22-23, 1981; submitted July 6, 1981; revision received Aug. 30, 1982. Copyright © American Institute of Aeronautics and Astronautics, Inc., 1982. All rights reserved.

*Professor of Engineering Science and Mechanics. Associate Fellow AIAA.

†Assistant Professor of Engineering Science and Mechanics. Member AIAA.

$$L(\rho u_i) = \frac{\partial(\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}[u_j \rho u_i + p\delta_{ij} - \sigma_{ij}] = 0 \tag{2}$$

$$L(\rho e) = \frac{\partial(\rho e)}{\partial t} + \frac{\partial}{\partial x_j}[u_j \rho e + u_j p - \sigma_{ij} u_i - q_j] = 0 \tag{3}$$

In Eqs. (1-3), $\rho$ is density, $\rho u_i$ the momentum vector, $p$ the pressure, and $e$ the mass specific total energy. Assuming a polytropic gas, $p = (\gamma - 1)\rho\epsilon$, the equation of state is

$$p = (\gamma - 1)[\rho e - \tfrac{1}{2}\rho u_j u_j] \tag{4}$$

The Stokes stress tensor $\sigma_{ij}$, heat flux vector $q_j$, and specific internal energy $\epsilon$ are defined as

$$\sigma_{ij} = \frac{\mu}{Re}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] - \frac{2\mu}{3Re}\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{5}$$

$$q_j = -\kappa \frac{\partial e}{\partial x_j} \tag{6}$$

$$\epsilon = e - \tfrac{1}{2} u_i u_i \tag{7}$$

where $\mu$ is the absolute viscosity, $\kappa$ the coefficient of heat conductivity, and $\delta_{ij}$ the Kronecker delta.

The Euler equations are contained within Eqs. (1-4) upon specification that Eqs. (5) and (6) vanish identically. The form of Eqs. (1-4) is also representative of the mass-weighted formulation of the time-averaged Navier-Stokes equations for a turbulent flow.[13] In this instance, the variables are interpreted as appropriate descriptors of the mean flow, and $\sigma_{ij}$ and $q_j$ are generalized to include nonvanishing correlations of subgrid scale phenomena. For example, the total stress tensor becomes

$$\sigma_{ij} \equiv \bar{\sigma}_{ij} - \overline{\bar{\rho} u_i' u_j'} \tag{8}$$

where $-\overline{\bar{\rho} u_i' u_j'}$ is the Reynolds stress tensor and $\bar{\sigma}_{ij}$ denotes the time-averaged form for Eq. (5).

## Numerical Solution Algorithm

### Finite Element Formulation

The three-dimensional Navier-Stokes equation system is identified for the unsteady description. The direct steady-state algorithm will emerge as a special case of the transient finite element algorithm formulation. Denoting $\rho u_i \equiv m_i$ and $\rho e \equiv g$, Eqs. (1-6) describe evolution of the vector dependent variable with elements

$$\{q\}^T \equiv \{\rho, m_i, g, p, \sigma_{ij}, q_j\} \tag{9}$$

The initial-value, partial differential equations (1-3) are of the form

$$L(q_\alpha) = \frac{\partial q_\alpha}{\partial t} + \frac{\partial}{\partial x_j}[u_j q_\alpha + f_{\alpha j}] + s_\alpha = 0 \tag{10}$$

In Eq. (10) $f_{\alpha j}(q_\beta)$ and $s_\alpha(q_\beta)$ are specified nonlinear functions of their argument, as determined by inspection. The form for the remaining algebraic and partial differential equations (4-8) is

$$L(q_\alpha) = q_\alpha + f_\alpha(q_\beta) = 0 \tag{11}$$

The $n$-dimensional equation system [Eqs. (10) and (11)] is defined on the Euclidean space $R^n$ spanned by $x$ with scalar components $x_i$, $1 \le i \le n$. The solution domain $\Omega$ is the product of $R^n$ and $t$ for all elements of $x$ belonging to $R^n$ and all elements of $t$ belonging to the open interval measured from $t_0$, i.e.,

$$\Omega \equiv R^n \times t = \{(x,t): \; x \in R^n \text{ and } t \in [t_0, t)\} \tag{12}$$

The domain boundary $\partial\Omega$ is the product of the boundary $\partial R$ of $R^n$ spanned by $\bar{x}$ and $t$, i.e., $\partial\Omega = \partial R \times t$. Thereupon, the general form for the differential boundary constraint is

$$\ell(q_\alpha) = a_1^\alpha q_\alpha + a_2^\alpha \frac{\partial}{\partial x_j} q_\alpha \hat{n}_j + a_3^\alpha = 0 \tag{13}$$

where the $a_I^\alpha$ are specified coefficients and $\hat{n}_j$ is the unit normal vector. Finally, an initial distribution for $q_\alpha$ on $\Omega_0 = R^n \times t_0$ is required,

$$q_\alpha(x, t_0) = q_\alpha^0(x) \tag{14}$$

The functional requirement of the (any) numerical solution algorithm for Eqs. (10-13) is to construct a suitable approximation $q_\alpha^h(x,t)$ to the dependent variable set $q_\alpha(x,t)$ and to render the error in this approximation extremum in some norm. A finite element algorithm is the formal application of these basic requirements. The approximation $q_\alpha^h(x,t)$ is constructed from members of a convenient finite-dimensional subspace of $H_0^1(\Omega)$, the Hilbert space of all functions possessing square integrable first derivatives, and satisfying the boundary conditions of Eq. (13). While extremely flexible in theory, the usual practice is to employ polynomials truncated at degree $k$ and defined on disjoint interior subdomains $\Omega_e$, the union of which forms the discretization of $\Omega = U\Omega_e$. Hence, by definition

$$q_\alpha(x,t) \approx q_\alpha^h(x,t) = \mathop{U}_{e=1}^{M} q_\alpha^e(x,t) \tag{15}$$

The element semidiscrete approximation is the series

$$q_\alpha^e(x,t) \equiv \{N_k(x)\}^T \{QI(t)\}_e \tag{16}$$

In Eq. (16), the free index $I$ denotes members of $q_\alpha^h$ at nodes of $U\Omega_e$, and sub- or superscript $e$ denotes pertaining to the $e$th finite element, $\Omega_e = R_e^n \times t$. The elements of the row matrix $\{N_k(x)\}^T$ are polynomials written on $x_j$, $1 \le j \le n$, complete to degree $k$ and constructed to form a cardinal basis.[14]

To render the error in $q_\alpha^h$ extremum, a finite element algorithm requires the semidiscrete approximation error in Eqs. (10), (11), and (13), i.e., $L(q_\alpha^h)$ and $\ell(q_\alpha^h)$, to be orthogonal to the function space utilized to construct $q_\alpha^h$. A multipole expansion is also invoked to constrain select derivatives of the error. Specifically, since high-frequency dispersion is a dominant error mechanism, and since $L(q_\alpha^h)$ is therefore locally nonsmooth, the finite element algorithm requires that $\nabla L(q_\alpha^h)$ also be orthogonal to $\{N_k\}$. These constraints, which are linearly independent, are combined into a single theoretical statement by identification of the (Lagrange) multiplier set $\beta_i$ as

$$\int_{R^n} \{N_k\} L(q_\alpha^h)\,dx + \beta_1 \cdot \int_{R^n} \{N_k\} \nabla L(q_\alpha^h)\,dx + \beta_2 \int_{\partial R} \{N_k\} \ell(q_\alpha^h)\,dx \equiv \{0\} \tag{17}$$

Upon definition of $k$ in Eq. (16), Eq. (17) is a coupled system of ordinary differential and algebraic equations of the form

$$[C]\{QI\}' + [U]\{QI\} + [FIJ]\{QJ\} + \{SI\} = \{0\} \tag{18}$$

The specific form of the matrices in Eq. (18) are given in a later section. Since the $[C]$ matrix is nondiagonal, an implicit integration algorithm is suggested. The $\Theta$-implicit finite difference algorithm, where $\Theta = \frac{1}{2}$ yields the trapezoidal rule, can be written as

$$\{FI\} = \{QI\}_{j+1} - \{QI\}_j - \Delta t[\Theta\{QI\}'_{j+1} + (1-\Theta)\{QI\}'_j] = 0 \tag{19}$$

Equation (18) provides the definition of the derivative $\{QI\}'$ for Eq. (19). Upon substitution and proceeding through the algebra,[10] Eq. (19) becomes a nonlinear algebraic equation system written on $\{QI\}_{j+1}$. The Newton matrix iteration solution algorithm is

$$[J(FI)]^p_{j+1}\{\delta QI\}^{p+1}_{j+1} = -\{FI\}^p_{j+1} \tag{20}$$

The dependent variable $\{\delta QI\}$ yields the solution

$$\{QI\}^{p+1}_{j+1} = \{QI\}^p_{j+1} + \{\delta QI\}^{p+1}_{j+1} \tag{21}$$

The Jacobian for Eq. (20) is formed from Eq. (19) as

$$[J(FI)] = \frac{\partial\{FI\}}{\partial\{QJ\}} \tag{22}$$

## Generalized Coordinates

A principal requirement in computational aerodynamics is to accurately interpolate solution domain boundaries $\partial R$ that are nonregular and perhaps nonsmooth. The term "generalized coordinates" has gained acceptance to describe the algorithm statement appropriate for use with a regularizing, boundary-fitted coordinate transformation. Many procedures are available to construct such transformations.[15] The generated coordinate transformation is the mapping

$$x_i = x_i(\eta_j) \tag{23}$$

Since Eq. (16) defines a general interpolation, the local form for Eq. (23) is

$$x_i = \{N_k(\eta)\}^T\{XI\}_e, \qquad x_i \in R^n_e \tag{24}$$

The elements of $\{XI\}_e$ are the coordinates of the nodes of $R^n_e$, $1 \leq (i,I) \leq n$, hence $UR^n_e$ the global discretization. The elements of $\{N_k(\eta)\}$ for $R^2_e$ and $R^3_e$ are well known[1] for $1 \leq k \leq 3$.

The numerical algorithm requirement is transformation of the divergence operator in Eq. (10) to

$$\frac{\partial}{\partial x_j} = \frac{\partial}{\partial \eta_k}\frac{\partial \eta_k}{\partial x_j} \tag{25}$$

The elements of the inverse Jacobian, $J_e^{-1} \equiv [\partial\eta_k/\partial x_j]_e$, are

$$\left[\frac{\partial \eta_k}{\partial x_j}\right]_e \equiv J_e^{-1} = \frac{1}{\det[J]_e}[\text{transformed } J \text{ cofactor}]_e \tag{26}$$

where $[J]_e = [\partial x_i/\partial \eta_j]_e$ is directly evaluated from Eq. (24). The differential element for Eq. (17) is

$$\mathrm{d}x = \det[J]_e \mathrm{d}\eta \tag{27}$$

Then, for example, using the divergence theorem for the second term in Eq. (10), the first error constraint in Eq. (17) for $L(\rho u_i)$ becomes

$$\int_{R^n}\{N\}L(\rho u_i^h)\mathrm{d}x = \int_{R^n}\{N\}\frac{\partial \rho u_i^h}{\partial t}\det[J]\mathrm{d}\eta + \oint_{\partial R^n}\{N\}[u_j^h\rho u_i^h + p^h\delta_{ij} - \sigma_{ij}^h]\cdot\hat{n}_j\det[J]\mathrm{d}\eta - \int_{R^n}\frac{\partial\{N\}}{\partial\eta_k}\left(\frac{\partial\eta_k}{\partial x_j}\right)[u_j^h\rho u_i^h + p^h\delta_{ij} - \sigma_{ij}^h]\det[J]\mathrm{d}\eta \tag{28}$$

Define the contravariant components of convection velocity on $R^n_e$ as

$$\bar{u}_k^e = \det[J]_e\left(\frac{\partial\eta_k}{\partial x_j}\right)u_j^e = [\mathrm{Cof}J]_e u_j^e \tag{29}$$

where $[\mathrm{Cof}J]_e$ is the transformed cofactor matrix of $[J]_e$. Using Eq. (16) to interpolate $\bar{u}_k^e$ on $R^n_e$ and noting that the surface integral in Eq. (28) vanishes identically on all interior elemental boundaries $\partial R_e$, Eq. (28) becomes the matrix statement

$$\begin{aligned}\int_{R^n}\{N\}L(\rho u_i^h)\mathrm{d}x \Rightarrow\ & S_e\Bigg[\int_{R^n_e}\Big(\{DET\}_e^T\{N\}\{N\}\{N\}^T\{RHOUI\}'_e \\ & -\{UBARK\}_e^T\{N\}\frac{\partial}{\partial\eta_k}\{N\}\{N\}^T\{RHOUI\}_e \\ & -\{ETAKI\}_e^T\{N\}\frac{\partial}{\partial\eta_k}\{N\}\{N\}^T\{P\}_e \\ & +\{ETAKJ\}_e^T\{N\}\frac{\partial}{\partial\eta_k}\{N\}\{N\}^T\{SIGIJ\}_e\Big)\mathrm{d}\eta \\ & +\oint_{\partial R_e\cap\partial R}\Big(\{UBARK\}_e^T\{N\}\{N\}\{N\}^T\{RHOUI\}_e \\ & +\{DET\}_e^T\{N\}\{N\}\{N\}^T\{P\}_e\delta_{IK} \\ & -\{DET\}_e^T\{N\}\{N\}\{N\}^T\{SIGIK\}_e\Big)\cdot\hat{\eta}_k\mathrm{d}\eta\Bigg]\end{aligned} \tag{30}$$

In Eq. (30), since the $\{N_k\}$ for $q_i^h$ are orthonormalized on $R^n_e$, the limits for each integral become $\eta = \pm 1$. $S_e$ is the matrix operator[16] projecting computed element contributions onto the corresponding global matrix statement [Eq. (18)].

The $e$-subscripted terms in Eq. (30) are independent of $\eta_k$ and can be extracted from the integrand, leaving only products of the polynomials and derivatives in $\{N_k\}$. These are directly evaluable, using numerical quadrature, yielding Eq. (30) as the matrix statement

$$\begin{aligned}\int_{R^n}\{N\}L(\rho u_i^h)\mathrm{d}x \Rightarrow S_e\Big[&\{DET\}_e^T[M3000]\{RHOUI\}'_e \\ & -\{UBARK\}_e^T\Big([M30K0] - \hat{n}_K[N3000]\Big)\{RHOUI\}_e \\ & -\Big(\{ETAKI\}_e^T[M30K0] - \hat{n}_I\{DET\}_e^T[N3000]\Big)\{P\}_e \\ & +\Big(\{ETAKL\}_e^T[M30K0] \\ & -\hat{n}_L\{DET\}_e^T[N3000]\Big)\{SIGIL\}_e\Big]\end{aligned} \tag{31}$$

In Eq. (31) the indices $K$ and $L$ obey the tensor summation rule, $1 \leq (K,L) \leq n$, $I$ is the free index (for $\rho u_I^h$) and $[M30K0]$ is the hypermatrix equivalent of $\partial/\partial\eta_k$ (transformed), contracted with corresponding element distributions. The matrix $[N3000]$, premultiplied by unit outward pointing normal $\hat{n}_i$ to $\partial R$, results from the integration of the last term in Eq. (30). $\{DET\}_e$ is the nodal distribution of $\det[J]_e$ on $R_e^n$, while $\{ETAKI\}_e$ and $\{ETAKL\}_e$ are corresponding nodal distributions of components of $J^{-1}$ on $R_e^n$. Within the generalized coordinate framework, therefore, the grid and metric data required for a finite element numerical simulation reduces to nodal distributions of $J_e^{-1} = [\partial\eta_k/\partial x_j]_e$ and $\det[J]_e = \det[\partial x_i/\partial\eta_k]_e$.

**Tensor Matrix Product Jacobian**

For efficiency, a tensor matrix product approximation to Eq. (22) can be constructed using the tensor product cardinal basis $\{N_k(\eta)\}$ spanning quadrilateral and hexahedron domains on $R^2$ and $R^3$, respectively. The Newton algorithm Jacobian $[J(FI)]$ is replaced by the tensor (outer) product

$$[J(FI)] \Rightarrow [J_1] \otimes [J_2] \otimes [J_3] \tag{32}$$

Each component $[J_\eta]$ is constructed from its definition [Eq. (22)], assuming interpolation and differentiation are one-dimensional. The solution statement, [Eq. (20)], then becomes

$$[J_1] \otimes [J_2] \otimes [J_3]\{\delta QI\}_{j+1}^{p+1} = -\{FI\}_{j+4}^{p} \tag{33}$$

Defining

$$\{P1\}_{j+1}^{p+1} \equiv [J_2] \otimes [J_3]\{\delta QI\}_{j+1}^{p+1}$$
$$\{P2\}_{j+1}^{p+1} \equiv [J_3]\{\delta QI\}_{j+1}^{p+1} \tag{34}$$

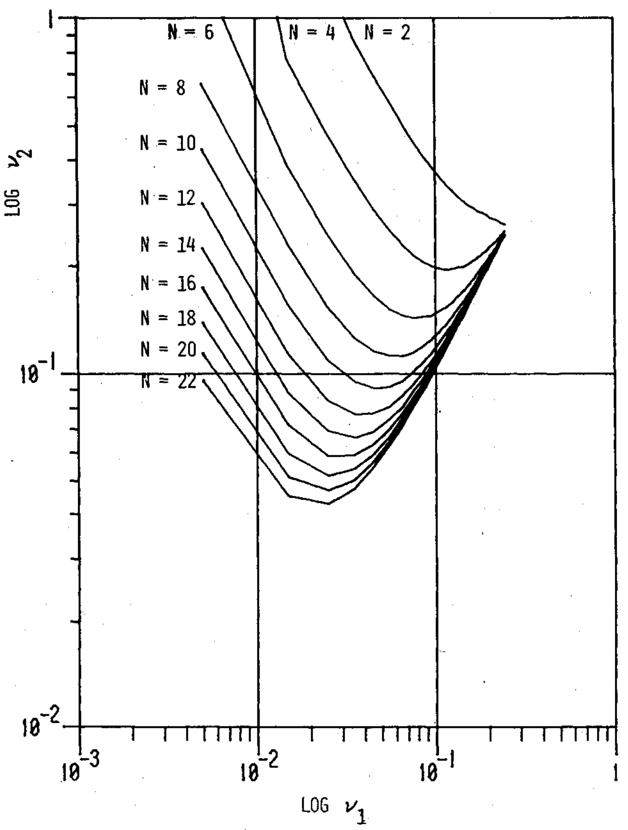


**Fig. 1 Solution to Eq. (39).**

the operational sequence for Eq. (33) becomes

$$[J_1]\{P1\}_{j+1}^{p+1} = -\{FI\}_{j+1}^{p}$$
$$[J_2]\{P2\}_{j+1}^{p+1} = \{P1\}_{j+1}^{p+1}$$
$$[J_3]\{\delta QI\}_{j+1}^{p+1} = \{P2\}_{j+1}^{p+1} \tag{35}$$

Other permutations of the index sequence for $[J_\eta]$ could be utilized. The key aspect is replacement of the very large (albeit sparse) matrix $[J]$, with $\alpha$-block-tridiagonal [pentadiagonal for $k=2$ in Eq. (16)] matrices $[J_\eta]$, yielding a large reduction in both storage for the Jacobian and CPU to construct the $LU$ decomposition and perform the back substitution. This procedure in no way affects the formation of $\{FI\}$, wherein lie the accuracy features intrinsic to the finite element algorithm statement. Compromises in the construction of $\{FI\}$ will invariably produce inferior results. A complete description of the algorithm construction [Eqs. (32-35)] is given in Ref. 16.

## Theoretical Analysis

The estimation of an "optimum" constraint set $\beta_I$ [Eq. (17)] is provided by a semidiscrete stability analysis of Eq. (1) on $R^1$. Assuming a constant convection velocity $U_0$, the semidiscrete Fourier solution is

$$q_I^h(j\Delta x,t) = V\exp[i\omega(j\Delta x - \Gamma t)] \tag{36}$$

Here, $\Gamma \equiv U_0[\sigma + i\delta]$, where $\sigma$ and $\delta$ are real numbers, $i=\sqrt{-1}$, and $\omega = 2\pi/\lambda$ is the wave number for wavelength $\lambda$. Substituting Eq. (36) into Eq. (17), expanding $\beta_I$ as $\nu\det[J]_e$, where $\nu \equiv \{\nu_1,\nu_2\}\hat{i}$, setting $k=1$ in Eq. (16), and proceeding through the algebra[16] yields the solutions

$$\sigma = \left[1 - \nu_1(\nu_1-\nu_2)d^2 + \left(\frac{-1}{180} + \frac{\nu_1\nu_2}{12} + \nu_1^3(\nu_1-\nu_2)\right)d^4 + 0(d^6)\right] \tag{37}$$

$$\delta = \left[(\nu_1-\nu_2)d - \left(\frac{\nu_2}{12} - \nu_1^2(\nu_1-\nu_2)\right)d^3 + 0(d^5)\right] \tag{38}$$

where $d=\omega\Delta x$ and $0(\cdot)$ indicates order. Without the multipole term, i.e., $\beta_I = \mathbf{0}$, Eqs. (37) and (38) indicate the basic algorithm is spatially fourth-order accurate and free of artificial diffusion. Setting $\nu_1 \equiv 1/\sqrt{15} \equiv \nu_2$ yields $\sigma \approx 1 + 0(d^6)$ [Eq. (37)], hence $\delta \approx -d^3/12$, which is documented[16] to produce excessive artificial diffusion. Defining $\nu_1$ and $\nu_2$ distinct and enforcing sixth-order accuracy in Eq. (37) yields the constraint

$$\nu_2 = \frac{\nu_1^2 + d^2/180 - \nu_1^4 d^2}{\nu_1(1 + d^2/12 - \nu_1^2 d^2)} \tag{39}$$

The semidiscrete solution resolves discrete wavelengths $\lambda_n = n\Delta x$, where $d = 2\pi/n$. The solution of Eq. (39) for a range of $\nu_1 > 0$ and $n > 2$ is graphed in Fig. 1. The solutions for each $n$ converge at $\nu_1 = 1/\sqrt{15} = \nu_2$, and generally $\nu_2(n) > \nu_1(n) > 10^{-2}$ for sixth-order accuracy. For the smallest $n$, $\nu_2$ is an order of magnitude or more larger than $\nu_1$.

## Two-Dimensional, Bilinear Element Algorithm

**Metric Definitions**

Restrict consideration to problems on $R^2$ and $k=1$ in Eq. (16), i.e., $\{N_1(\eta)\}$. In this case, $R_e^2$ exhibits only vertex nodes and hence,

$$\left(\frac{\partial \eta_k}{\partial x_j}\right)_e \equiv \frac{1}{\det J_e}\{N_1\}^T\{ETAKJ\}_e \qquad 1\leq(K,J)\leq 2 \quad (40)$$

Noting that $(\det J)_e^{-1}$ cancels in the algebra [Eq. (26)], the elements of $\{ETAKJ\}_e$ are

$$\{ETAKJ\}_e = \frac{1}{2}\left\{\begin{array}{ll} \{Y4-Y1,\ Y3-Y2,\ Y3-Y2,\ Y4-Y1\}_e & (1,1) \\ \{X1-X4,\ X2-X3,\ X2-X3,\ X1-X4\}_e & (1,2) \\ \{Y1-Y2,\ Y1-Y2,\ Y4-Y3,\ Y4-Y3\}_e & (2,1) \\ \{X2-X1,\ X2-X1,\ X3-X4,\ X3-X4\}_e & (2,2) \end{array}\right\} \quad (41)$$

In Eq. (41), the indices in parentheses indicate $(K,J)$. Each array contains only two distinct entries; therefore, the storage required is 8M. The corresponding definition of interpolation for $\det[J]_e$ yields

$$\{DET\}_e = \frac{1}{4}\left\{\begin{array}{l} (X2-X1)(Y4-Y1) - (X4-X1)(Y2-Y1) \\ (X2-X1)(Y3-Y2) - (X3-X2)(Y2-Y1) \\ (X3-X4)(Y3-Y2) - (X3-X2)(Y3-Y4) \\ (X3-X4)(Y4-Y1) - (X4-X1)(Y3-Y4) \end{array}\right\}_e \quad (42)$$

The storage requirement for Eq. (42) is 4M. Therefore, the total metric storage requirement is 12M, if performed as a preprocessor operation. In Eqs. (41) and (42), the element nodal coordinates are $\{XI\}_e=\{XJ,YJ,1\leq J\leq 4\}_e$. The remaining metric-related operation is the contravariant convection velocity approximation $\bar{u}_k^e$. From Eq. (29) and for the definition,

$$\bar{u}_k^e \equiv \{N_1\}^T\{UBARK\}_e \quad (43)$$

the algebra operations yield

$$\{UBARK\}_e = \{ETAKJ\}_e\{N_1\}^T\{UJ\}_e \quad (44)$$

The evaluation of Eq. (44) is on a nodal basis, whereupon $\{N_1\}$ reduces to the Kronecker delta. Summation is implied over $J$, at the nodes $UJ\equiv MJ/R$ (i.e., $u_j\equiv m_j/\rho$), and no additional storage is required.

## Finite Element Algorithm

For a two-dimensional problem, the dependent variable set includes $\rho$, $m_i=\rho u_i$, $i=1,2$, $p$, and $q=\rho e$, and the parameters $\sigma_{ij}$ and $q_j$. The algorithm Lagrange multiplier $\beta_I$ is a two-dimensional vector with components $\det[J]\nu^i_{\alpha j}$. The subscript $\alpha$ signifies the appropriate variable of the dependent variable set $q_\alpha=\{\rho,m_k,g,p,\sigma_{k\ell},q_k\}$, where $1\leq(k,\ell)\leq 2$. The superscript $i=1,2$ signifies the two distinct components previously identified as $\nu_1$ and $\nu_2$ in the theoretical analysis. The subscript $j$ is a tensor index.

Order the discrete dependent-variable set as $\{QI\}^T=\{R,M(I),G,P,S(I,J),Q(J)\}$. The respective algorithm statements $\{FI\}$, [Eq. (19)] are the assembly of the elemental evaluations, i.e., $\{FI\}\equiv S_e\{FI\}_e$. In the generalized coordinate description, the element column matrices $\{FI\}_e$ are formed as the matrix inner products,

$$\{FR\}_e = \Big[\{DET\}^T[B3000] + \nu^1_{RJ}\{ETAKJ\}^T[B40K00]\{DET\}\Big]\{R\}'_{j+1} + \frac{\Delta t}{2}\Big[-\{ETAKI\}^T[B30K0]\{MI\} + \nu^2_{RJ}\{UBARL\}^T[B40KL0]\{ETAKJ\}\{R\}\Big]_{j+1,j} \quad (45)$$

$$\{FMI\}_e = \Big[\{DET\}^T[B3000] + \nu^1_{IJ}\{ETAKJ\}^T[B40K00]\{DET\}\Big]\{MI\}'_{j+1} + \frac{\Delta t}{2}\Big[-\{UBARK\}^T[B30K0]\{MI\} - \{ETAKI\}^T[B30K0]\{P\} + \{ETAKJ\}^T[B30K0]\{SIGIJ\} + \nu^2_{IJ}\{UBARL\}^T[B40KL0]\{ETAKJ\}\{MI\}\Big]_{j+1,j} \quad (46)$$

$$\{FG\}_e = \Big[\{DET\}^T[B3000] + \nu^1_{GJ}\{ETAKJ\}^T[B40K00]\{DET\}\Big]\{G\}'_{j+1} + \frac{\Delta t}{2}\Big[-\{UBARK\}^T[B30K0](\{G\} + \{P\}) + \{ETAKJ\}^T[B30K0]\{QJ\} + \{ETAKJ\}^T[B40K00]\{UL\}\{SIGLJ\} + \nu^2_{GJ}\{UBARL\}^T[B40KL0]\{ETAKJ\}\{G\}\Big]_{j+1,j} \quad (47)$$

$$\{FP\}_e = \{DET\}^T[B3000]\{P\} - (\gamma-1)\Big[\{DET\}^T[B3000]\{G\} + \tfrac{1}{2}\{DET\}^T[B40000]\{UJ\}\{MJ\}\Big]_{j+1} \quad (48)$$

$$\{FSIJ\}_e = \{DET\}^T[B3000]\{SIGIJ\} - \mu\Big[\{ETAKJ\}^T[B30K0]\{UI\} + \{ETAKI\}^T[B30K0]\{UJ\} - \tfrac{2}{3}\delta_{IJ}\{ETAKL\}^T[B30K0]\{UL\}\Big]_{j+1} \quad (49)$$

$$\{FQJ\}_e = \{DET\}^T[B3000]\{QJ\} - \kappa\{ETAKI\}^T[B30K0](\{GSR\} - \{ULUL\})_{j+1} \quad (50)$$

The general elemental hypermatrices $[M...]$ [Eq. (31)], are named $[B...]$ for the two-dimensional problem. The matrices $[B30K0]$, premultiplied by $\{UBARK\}^T$ and $\{ETAKJ\}^T$, are augmented on $\partial R_e \cap \partial R$ by the surface integral terms [see Eq. (31)]. Each integral is evaluated once over a master-element domain $R_e^2$. [Each element matrix defined in Eqs. (45-50), for $k=1$ in Eq. (16), is listed in Appendix B of Ref. 16.] The Boolean indices $(0,K)$ in $[B30K0]$ indicate the basis $\{N_k\}$ is not differentiated, or is differentiated once into the scalar components parallel to the $\eta_k$ coordinate system. In Eqs. (45-48), the discrete indices $J$, $K$, $L$, occurring in both matrix and variable names, are tensor summation indices with a range of $1 \le (J,K,L) \le 2$. The free index $I$ in Eq. (46) denotes Cartesian scalar components of $m_i$, i.e., $\{MI\}$, while $I$ and $J$ are both free indices in Eqs. (49) and (50).

The matrices $\{DET\}$ and $\{ETAKJ\}$ contain elements equal to the nodal values of determinant $J_e$ and elements of $(\partial\eta_k/\partial x_j)_e$. The nodal values of the contravariant scalar components of the solution parameter $\bar{u}_k$ are denoted $\{UBARK\}$. The elements of $\{SIGIJ\}$ are nodal values of the stress tensor computed in principal coordinates in terms of $u_i = m_i/\rho$. In Eq. (48), $\delta_{IJ}$ is the Kronecker delta; in Eq. (50), the elements of $(GSR)$ are nodal values of $e \equiv g/\rho$, while those of $\{ULUL\}$ are nodal values of specific kinetic energy $\frac{1}{2}u_\ell u_\ell$. In all equations, the notation $\{\cdot\}'_{j+1}$ denotes $\{\cdot\}^p_{j+1} - \{\cdot\}_j$ and $[\cdot]_{j+1,j}$ denotes evaluation of the argument at $t_{j+1}$ and $t_j$ followed by addition after multiplication by $\Theta$ and $(1-\Theta)$ [Eq. (19)]. Finally, note that Eqs. (45-50) completely express the three-dimensional algorithm by the exchange of $B$ prefix matrices with those of $C$ prefix, and extension of the discrete tensor index range to $1 \le (I,J,K,L) \le 3$.

### Tensor Matrix Product Jacobian

The remaining step in the algorithm formulation is construction of the tensor matrix product form of the Newton Jacobian, see Eq. (22). This is a calculus operation since Eqs. (45-50) are in a functional form and $[J] \equiv S_e[J]_e$. The lead term in each equation is $\{DET\}^T[B3000]\{QI\}_{j+1}$. Therefore, the first term in $[J(FI)]_e$, which accounts for self-coupling, is

$$\frac{\partial\{FI\}_e}{\partial\{QI\}_e} \equiv [JQQ]_e = \{DET\}_e^T[B3000] \tag{51}$$

Referring to Eq. (30) for the definition, the elemental operation in forming Eq. (51) is

$$[JQQ]_e \equiv \{DET\}_e^T[B3000] \equiv \int_{R_e^2} \{DET\}_e^T\{N_k\}\{N_k\}\{N_k\}^T \mathrm{d}\eta \tag{52}$$

Assuming for simplicity a rectangular element domain $R_e^2$, described by measures $\ell$ and $\omega$ and spanned by $\{N_I\}$, the evaluation of Eq. (52) yields

$$[JQQ]_e \equiv \{DET\}_e^T[B3000] = \Delta_e[B200]$$

$$= \ell\omega[B200] = \frac{\ell\omega}{36}\begin{bmatrix} 4 & 2 & 1 & 2 \\ & 4 & 2 & 1 \\ & & 4 & 2 \\ (\mathrm{sym}) & & & 4 \end{bmatrix} \tag{53}$$

The tensor product construction for Eq. (53) involves evaluation of Eq. (52) constrained to one dimension and definition of the outer product. Denoting $M \equiv A$ for $n=1$,

$$[JQQ_\eta]_e \equiv \Delta_\eta[A200] = \int_{R_e^1} \{N_I\}\{N_I\}^T \mathrm{d}x \tag{54}$$

Performing the integrals,

$$[JQQ_1]_e = \frac{\ell}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad [JQQ_2]_e = \frac{\omega}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{55}$$

assuming $\Delta_1 = \ell$ and $\Delta_2 = \omega$. Accounting for entry locations in $[JQQ]_e$, as determined by the node numbering convention, and letting $\otimes$ denote the outer product, it is readily verified that

$$[JQQ]_e = [JQQ_1]_e \otimes [JQQ_2]_e \tag{56}$$

The operations defined by Eqs. (53-56) can be extended to all terms in the Jacobian, cf. Ref. 16. The construction of certain terms involves differentiation with respect to the parameter $\bar{u}_k = \bar{m}_k/\rho$. Using the chain rule,

$$\frac{\partial}{\partial\{MI\}} = \frac{\partial}{\partial\{MI\}} + \frac{\partial}{\partial\{\bar{M}K\}}\frac{\partial\{\bar{M}K\}}{\partial\{MI\}} + \frac{\partial}{\partial\{\bar{U}K\}}\frac{\partial\{\bar{U}K\}}{\partial\{MI\}}$$
$$= \frac{\partial}{\partial\{MI\}} + \det J\frac{\partial\eta_k}{\partial x_i}\left[\frac{\partial}{\partial\{\bar{M}K\}} + \frac{1}{\bar{\rho}}\frac{\partial}{\partial\{\bar{U}K\}}\right] \tag{57}$$

$$\frac{\partial}{\partial\{R\}} = \frac{\partial}{\partial\{R\}} - \left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\frac{\partial}{\partial\{\bar{U}K\}} \tag{58}$$

Recalling that operations are performed on the element matrices, denoting $\bar{D}_{KI}$ as the element average of $\det J_e(\partial\eta_k/\partial x_i)_e$ and letting $K$ signify the discrete free index corresponding to $\partial/\partial\eta_k$, the nonempty tensor product element Jacobians for Eqs. (45-48) are

$$\begin{aligned}
[JRR]_e &= \{DETK\}^T[A3000] \\
&\quad + \nu_{RJ}^1\{ETAKJ\}^T[A40K00]\{DETK\} \\
&\quad + \frac{\Delta t}{2}\nu_{RJ}^2\Big[\{ETAKJ\}^T[M40KL0]\{UBARL\} \\
&\quad - \left(\frac{\bar{m}_L}{\bar{\rho}^2}\right)\{ETAKJ\}^T[M40K0L]\{R\}\Big] \\
[JRMI]_e &= \frac{\Delta t}{2}\Big[-\{ETAKI\}^T[A30K0] \\
&\quad + \nu_{RJ}^2\bar{D}_{KI}\{ETAKJ\}^T[A30KK]\Big] \\
[JMIR]_e &= \frac{\Delta t}{2}\left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\Big[\{MI\}^T[A30K0] \\
&\quad + \nu_{IJ}^2\{MI\}^T[A4KK00]\{ETAKJ\}\Big] \\
[JMIMI]_e &= \{DETK\}^T[A3000] \\
&\quad + \nu_{IJ}^1\{ETAKJ\}^T[A40K00]\{DETK\} \\
&\quad + \frac{\Delta t}{2}\Big[-\{UBARK\}^T[A30K0] \\
&\quad - \bar{D}_{KI}\{MI\}^T[A30K0]\frac{1}{\bar{\rho}} \\
&\quad + \nu_{IJ}^2(\{UBARK\}^T[A40KK0]\{ETAKJ\} \\
&\quad + \frac{1}{\bar{\rho}}\{MI\}^T[A4KK00]\{ETAKJ\})\Big]
\end{aligned} \tag{59}$$

$$[JMIMJ]_e = \frac{\Delta t}{2}\bar{D}_{KJ}\left(\frac{1}{\bar{\rho}}\right)\Big[-\{MJ\}^T[A30K0]$$

$$+\nu^2_{JL}\{MJ\}^T[A4KK00]\{ETAKL\}\Big]$$

$$[JMIP]_e = -\frac{\Delta t}{2}\{ETAKI\}^T[A30K0]$$

$$[JMISIJ]_e = \frac{\Delta t}{2}\{ETAKJ\}^T[A40K00]\{DETK\} \qquad (60)$$

$$[JGR]_e = \frac{\Delta t}{2}\left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\Big[\{G+P\}^T[A30K0]$$

$$+\nu^2_{GJ}\{G\}^T[A4KK00]\{ETAKJ\}\Big]$$

$$[JGMI]_e = \frac{\Delta t}{2}\bar{D}_{KI}\left(\frac{1}{\bar{\rho}}\right)\Big[-\{G+P\}^T[A30K0]$$

$$+\nu^2_{GL}\{G\}^T[A4KK00]\{ETAKJ\}\Big]$$

$$[JGG]_e = \{DETK\}^T[A3000]$$

$$+\nu^1_{GJ}\{ETAKJ\}^T[A40K00]\{DETK\}$$

$$+\frac{\Delta t}{2}\Big[-\{UBARK\}^T[A30K0]$$

$$+\nu^2_{GJ}\{UBARK\}^T[A40KK0]\{ETAKJ\}\Big]$$

$$[JGP]_e = -\frac{\Delta t}{2}\{UBARK\}^T[A30K0]$$

$$[JGSIJ]_e = \frac{\Delta t}{2}\{ETAKJ\}^T[A40K00]\{UI\}$$

$$[JGQJ]_e = \frac{\Delta t}{2}\bar{D}_{LJ}\{ETAKL\}^T[A30K0] \qquad (61)$$

$$[JPR]_e = -\left(\frac{\gamma-1}{2}\right)\left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\{DETK\}^T([A40000]\{MK\})$$

$$[JPMK]_e = \left(\frac{\gamma-1}{2}\right)\Big[\{DETK\}^T([A40000]\{UK\})$$

$$+\left(\frac{1}{\bar{\rho}}\right)\{DETK\}^T([A40000]\{MK\})\Big]$$

$$[JPG]_e = -(\gamma-1)\{DETK\}^T[A3000]$$

$$[JPP]_e = \{DETK\}^T[A3000] \qquad (62)$$

$$[JSIJR]_e = \frac{\mu}{\bar{\rho}^2}\Big[\bar{m}_i\{ETAKJ\}^T[A30K0]$$

$$+\bar{m}_j\{ETAKI\}^T[A30K0]$$

$$-\tfrac{2}{3}\delta_{IJ}\bar{m}_\ell\{ETAKL\}^T[A30K0]\Big]$$

$$[JSIJMI]_e = -\frac{\mu}{\bar{\rho}}\Big[\{ETAKJ\}^T[A30K0]$$

$$-\tfrac{2}{3}\delta_{IJ}\{ETAKL\}^T[A30K0]\Big]$$

$$[JSIJSIJ]_e = \{DETK\}^T[A3000] \qquad (63)$$

$$[JQIR]_e = \frac{\kappa}{\bar{\rho}^2}\Big[\bar{g}\{ETAKI\}^T[A30K0]$$

$$-\frac{\bar{m}_i^2}{\bar{\rho}}\{ETAKI\}^T[A30K0]\Big]$$

$$[JQIMI]_e = \frac{\kappa\bar{m}_i}{\bar{\rho}^2}\{ETAKI\}^T[A30K0]$$

$$[JQIG]_e = -\frac{\kappa}{\bar{\rho}}\{ETAKI\}^T[A30K0]$$

$$[JQIQI]_e = \{DETK\}^T[A3000] \qquad (64)$$

**Noniterative and Direct Steady State**

The noniterative and direct steady-state algorithms are special cases of the developed formulation. The noniterative form simply constitutes acceptance of the first solution, i.e., $\{\delta QI\}^2_{j+1}$ from the Newton algorithm, using only evaluation of $\{FI\}^1_{j+1} \equiv \{FI\}_j$. By definition $\{QI\}^1_{j+1} \equiv \{QI\}_j$, the dependent variable vector at time step $t_j$. Therefore, $\{QI\}'_{j+1} = \{0\}$ in Eqs. (45-47) and the corresponding expression in brackets is not evaluated. Furthermore, for the second bracketed expressions, $\Delta t/2 \Rightarrow \Delta t$ and the evaluation $[\cdot]_{j+1,j}$ reduces to $[\cdot]_j$. This procedure eliminates the error control parameter $\nu^1_{qj}$ from the algorithm. The tensor product Jacobians are unaltered, except that since the terms involving $\nu^1$ have been eliminated from $\{FI\}$ the corresponding terms in $[J_\alpha]$ are omitted. The multiplier $\Delta t/2$ remains appropriate as the time step factor and $\{QI\}^1_{j+1} \equiv \{QI\}_j$ in Eqs. (48-50).

The direct steady-state algorithm is identical to the noniterative formulation, except that $\Delta t/2 \Rightarrow \Delta t$ in the Jacobians. This multiplier, common to all elements of $\{FI\}$, can be removed by placing $(\Delta t)^{-1}$ as a scalar multiplier on $\{DET\}^T[A3000]$ in the Jacobians $[JQQ]$ in the differential equations exhibiting initial-value character, i.e., $\rho$, $m_i$, and $g$. Hence, this yields a nondiagonal tensor matrix generalization of the "approximate factorization" procedures devised for finite difference algorithms.[5]

## Results and Discussion

To validate, and refine as appropriate, the linearized theoretical analysis estimate of the parameter set $\nu^i_\alpha$, the inviscid Riemann shock tube problem of Sod[17] was executed. The problem specification discretizes the unit duct length into $M=100$ uniform finite elements for the $k=1$ algorithm, with the diaphragm located between elements 49 and 50. The initial condition specification is $m_i=0$, $p=1.0=\rho$ on $x_1 \le 0.5$, $p=0.1$ and $\rho=0.125$ on $x_1>0.5$, and $\gamma=1.4$.

Use of the linearized theoretical estimates for $\nu^\alpha_i$ yields a solution with excessive overshoot in all variables immediately downstream of the shock.[16] An optimization study was completed to refine the $\nu^i_\alpha$ with the criterion of sharp resolution of the shock with negligible overshoot. This "optimum" set, $\nu^1_\alpha \equiv \{\tfrac{3}{8}, 0, \tfrac{1}{4}\}/\sqrt{15}$ and $\nu^2_\alpha \equiv \{\tfrac{3}{4}, 2, 1\}/\sqrt{15}$ for $q_i = \{\rho, m_j, q\}$ and the $k=1$ algorithm, yields the solution shown in Fig. 2 for convergence defined as $\{|\delta QI|\} < 10^{-4}$. The dashed lines are the initial condition and each symbol corresponds to a node of $UR^1_e$. The results are in excellent agreement with linearized theory, the shock is sharply defined in each dependent-variable solution, and the various plateaus exhibit excellent flatness.

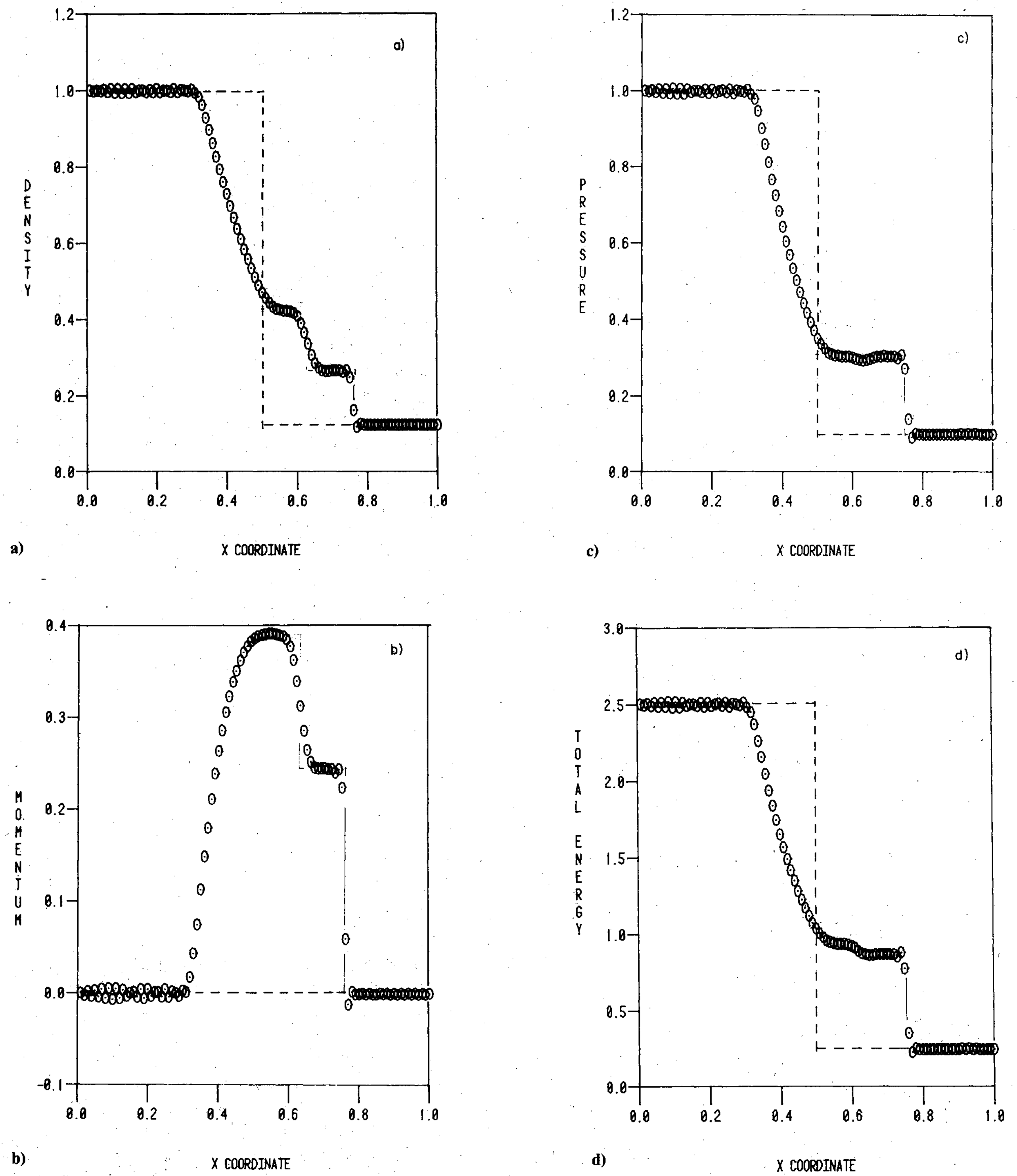
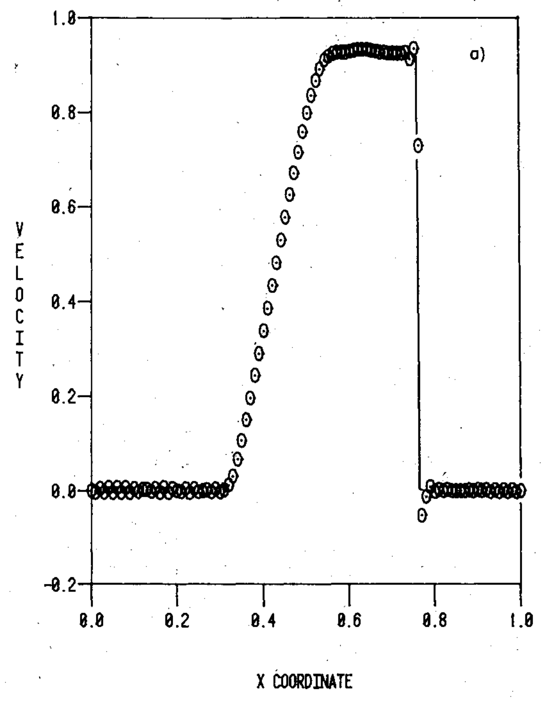


**Fig. 2 Finite element algorithm solution, Riemann shock tube, $t=0.14154$ s, $M=100$, $k=1$, $\nu^1_\alpha = \{3/8,0,1/4\}$, $\nu^2_\alpha = \{3/4,2,1\}$: a) density, b) momentum, c) energy, d) pressure.**

Sensitive measures for accuracy comparisons are convection velocity $\bar{u}_l = m_l/\rho$ and temperature. Figures 3 and 4 compare the iterative $k=1$ algorithm results to those obtained using a modification to the finite element algorithm that mimics the iterative, time-accurate form of the approximate factorization finite difference algorithms.[5,6] The finite element algorithm smears the shock over only two elements, while the pseudo-finite difference results are smeared over five elements. This loss in accuracy is directly attributable to the reduction in algorithm order of accuracy by the finite difference modification. Since there are no CPU time or core storage distinctions in obtaining these two solutions, the ad hoc modifications appear unwarranted. Additional detail, and $k=2$ finite element algorithm solutions are given in Ref. 16. The favorable "eyeball-norm" accuracy of the $k=1$ algorithm solutions are retained on the progression to coarser grids (Fig. 5), using $M=50$ and 25 element discretizations. The results of rigorous validation of accuracy and convergence in Sobolev norms are reported in Ref. 16.

The two-dimensional inviscid simulation of the Riemann problem defines $m_i = 0$ as the initial condition and applies a vanishing normal derivative in the $x_2$ direction as the boundary condition for $\{QI\}$, $1 \le I \le 5$. This is the default specification for the finite element algorithm and no dif-



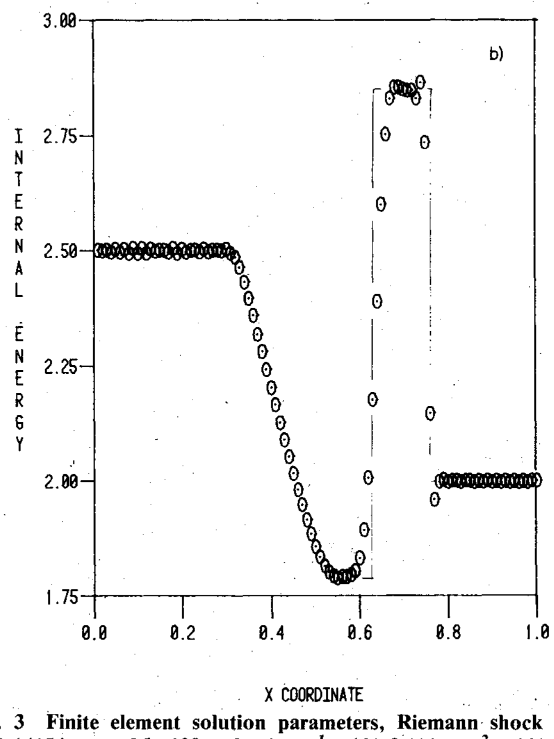
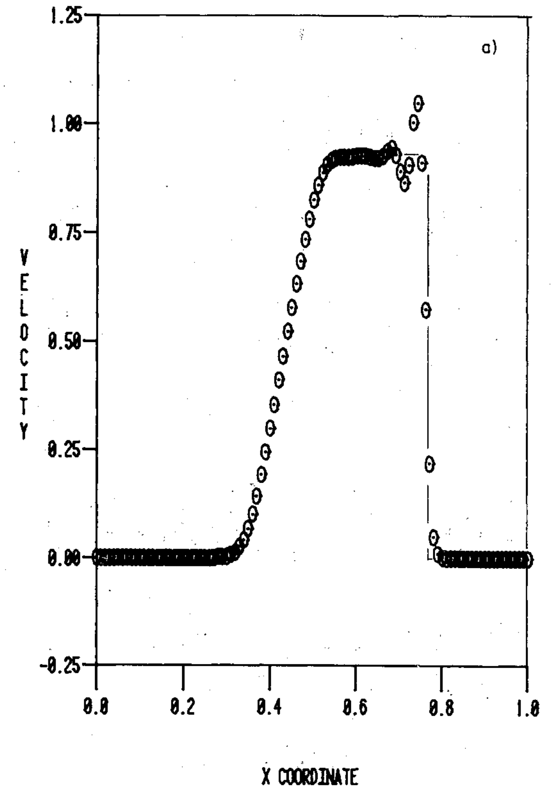


Fig. 3 Finite element solution parameters, Riemann shock tube, $t=0.14154$ s, $M=100$, $k=1$, $\nu_\alpha^I=\{3/8,0,1/4\}$, $\nu_\alpha^2=\{3/4,2,1\}$: a) velocity, b) internal energy.



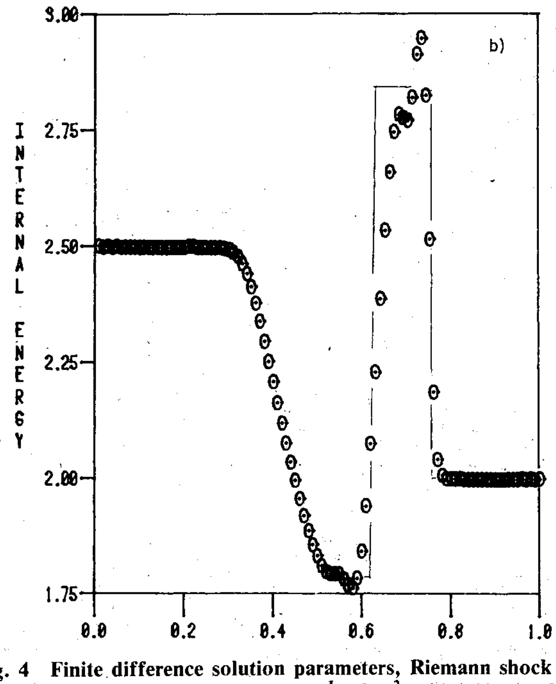


Fig. 4 Finite difference solution parameters, Riemann shock tube, $t=0.14154$ s, $M=100$, second-order, $\nu_\alpha^I=0$, $\nu_\alpha^2=\{1,1,1\}$: a) velocity, b) internal energy.

ference approximations are required for its enforcement. Figure 6 summarizes the $k=1$ algorithm solution for $\{QI\}=\{MI,P,R,G\}$ at $t=0.14154$ s using a $M=32\times6$ uniform discretization of $R^2$ and the predetermined optimum levels of $\nu_\alpha^2$ with $\nu_\alpha^I\equiv\{0\}$. The enforcement of vanishing gradient is essentially exact, and the shock is defined across the span of two elements in the $x_I$ direction. The maximum level computed for $\{M2\}$ was smaller than the convergence limit $\epsilon=10^{-3}$.

The Riemann solution was repeated for various orientations of the shock tube in the fixed Cartesian reference frame $x_i$. Figure 7 summarizes the noniterative solution for a 26 deg rotation, using $M=32\times6$ and the $\nu_\alpha^2$ of Fig. 6. No difficulty is encountered enforcing the vanishing derivative boundary condition on surfaces not aligned with $x_i$. The computed extremum of $\{M2\}$ is about one-third that of $\{M1\}$; squaring and adding to form $\bar{u}$ produces essential agreement with the solution of Fig. 5. Figure 8 summarizes the solution for the diaphragm oriented at an extremum angle (45 deg) to the mesh. The result is propagation of an oblique shock with large cross-mesh gradients. The shock position is sharply defined in all variables, as is the trailing plateau. No

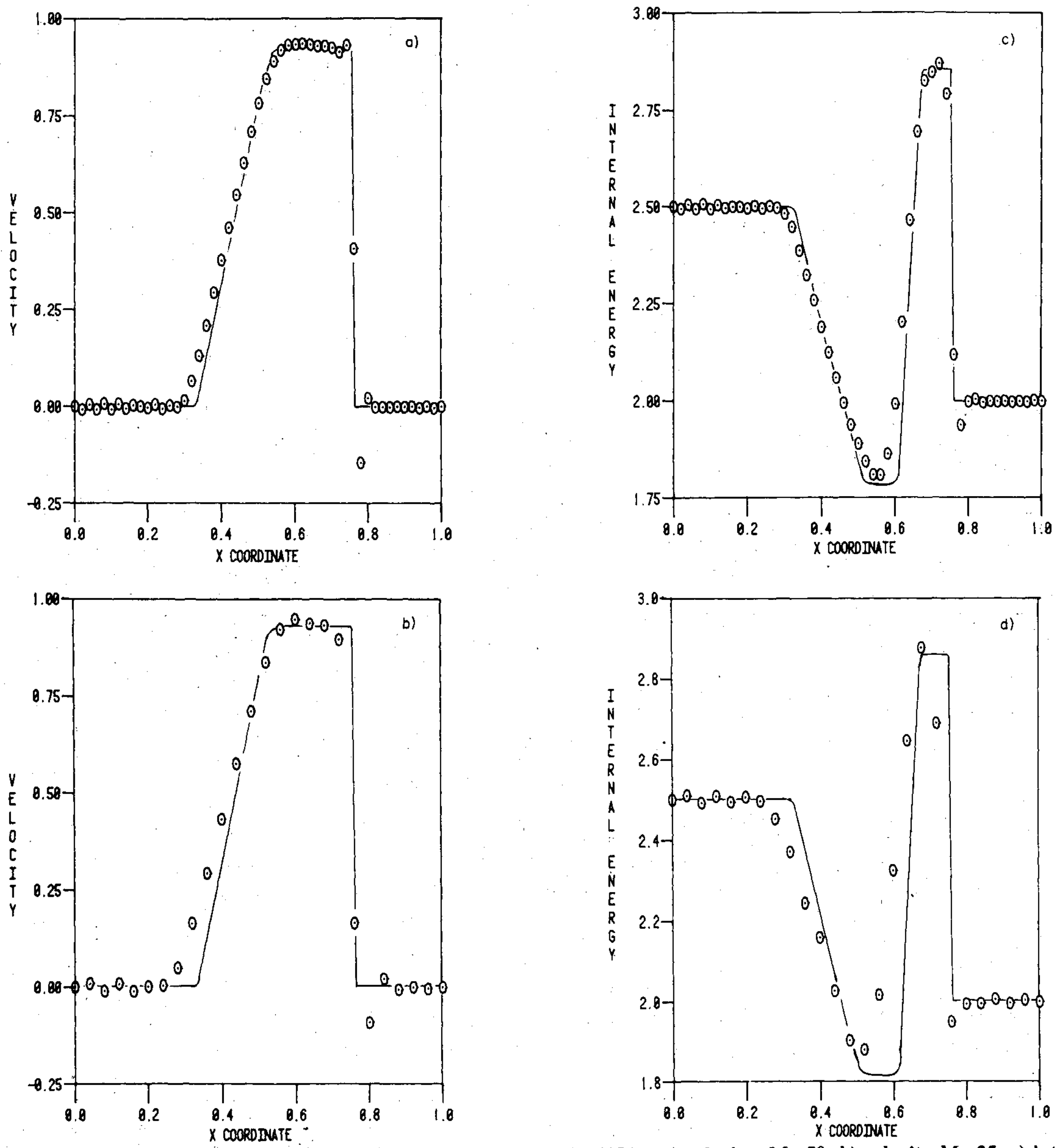


Fig. 5 Finite element solution parameters, Riemann shock tube, $k=1$, $t=0.14154$ s: a) velocity, $M=50$, b) velocity $M=25$, c) internal energy, $M=50$, d) internal energy, $M=25$.

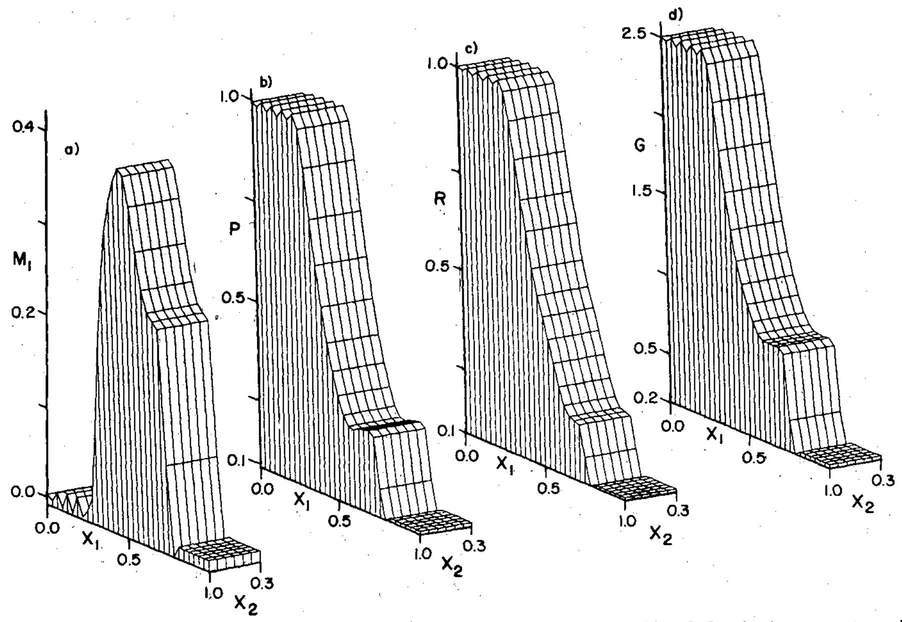


Fig. 6 Finite element algorithm solution, two-dimensional Riemann shock tube, iterative, $M=32\times 6$, $k=1$: a) momentum, b) pressure, c) density, d) energy.

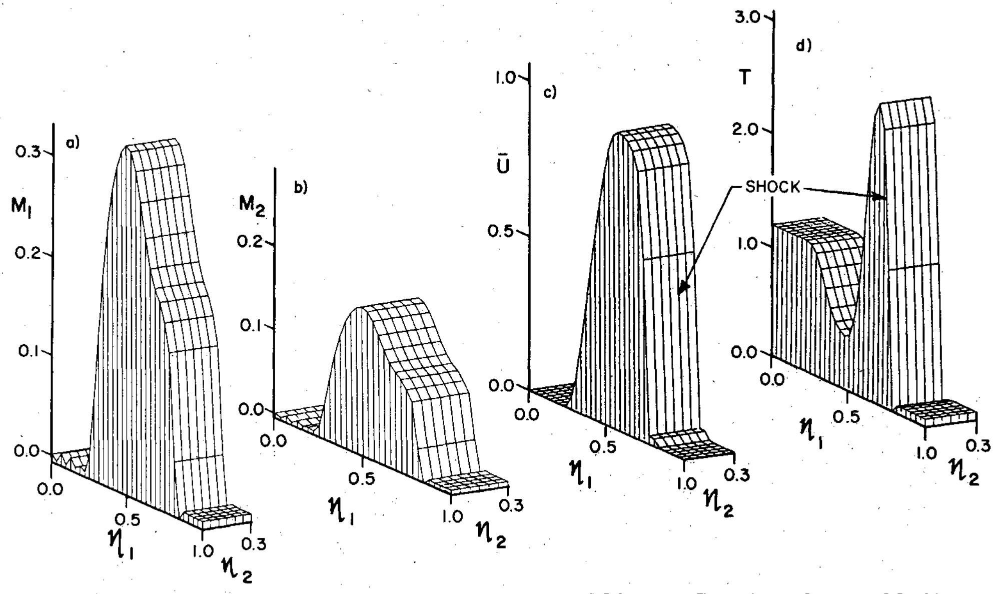


Fig. 7 Finite element algorithm solution, two-dimensional Riemann tube oriented 26 deg to Cartesian reference, $M=32\times6$, $k=1$: a) momentum $m_1$, b) momentum $m_2$, c) speed, d) temperature.

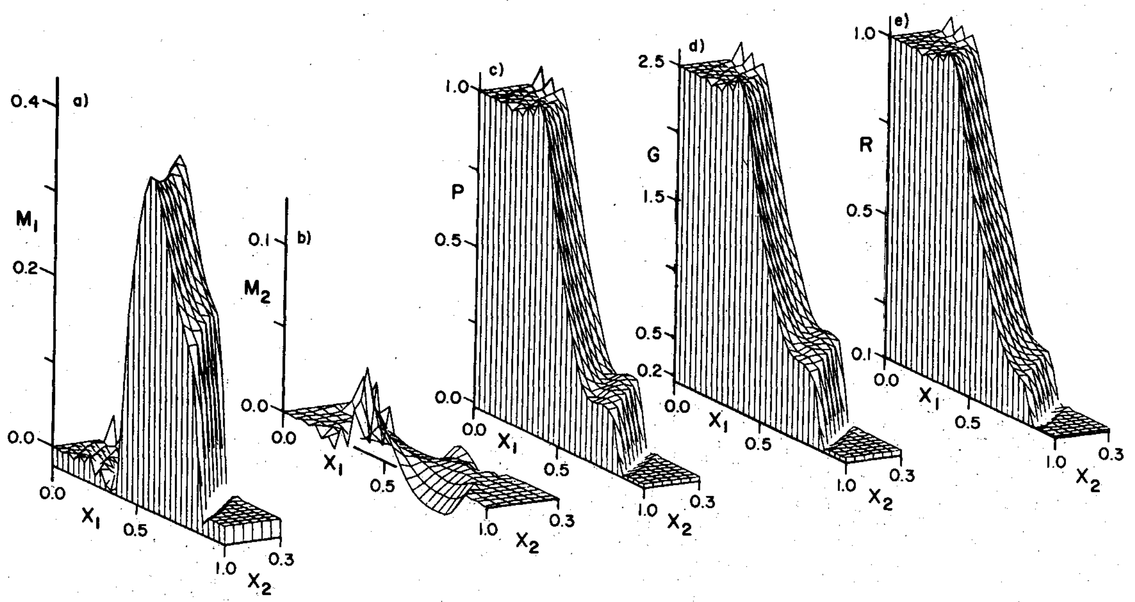


Fig. 8 Finite element algorithm solution, two-dimensional Riemann tube, oblique diaphragm and shock, $M=32\times6$, $k=1$: a) momentum $m_1$, b) momentum $m_2$, c) pressure, d) energy, e) density.

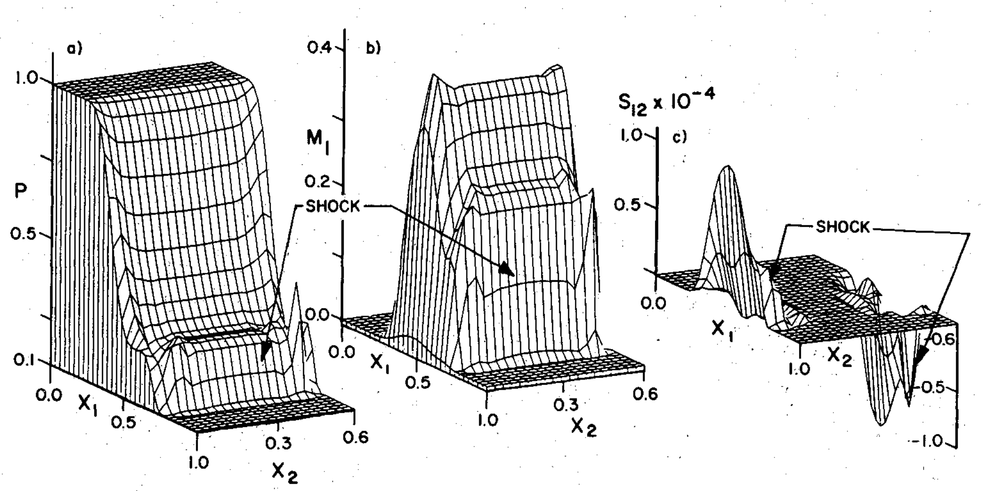


Fig. 9 Finite element algorithm solution, two-dimensional viscous Riemann problem, $Re=10^5$, $k=1$, $M=32\times20$, $\epsilon=10^{-3}$: a) pressure, b) momentum $m_1$, c) shear stress $\sigma_{12}$.

difficulty is indicated in enforcing the homogeneous von Neumann boundary constraint for arbitrary mesh intersection with the domain boundary. The solution trashiness near the upstream left corner is the result of an inadequate domain extension.

The two-dimensional inviscid Riemann problem is redefined as viscous by setting $m_i \equiv 0$ at the walls and the cold-wall boundary condition for the energy equation. Selecting $Re = 10^5$ and using a uniform $M = 32 \times 20$ grid aligned with the principal coordinates, Fig. 9 summarizes the $k=1$ algorithm solution at $t = 0.142$ s using the inviscid-optimized dissipation levels for $\nu_\alpha^2$ and $\nu_\alpha^I \equiv 0$. The shock front is sharply defined in the pressure solution (Fig. 9a). The growth of a boundary layer is visible in $m_I$ behind the shock. Furthermore, away from the walls, the centroidal nodal distribution of $m_I$ is in exact agreement with the inviscid solution. In the solution for the dominant shear stress $\sigma_{12}$ (Fig. 9c), the shock intersection with the wall is well defined and its passage has induced a residual large local level with steep gradients. Away from these actions, the solution is smooth and generally devoid of under- or overshoot. This solution used the same time step as the inviscid solution and averaged three iterations/time step at $\epsilon = 10^{-3}$.

## Summary and Conclusions

An implicit finite element solution algorithm is derived for problems in computational fluid dynamics. The Galerkin weighted residuals formalism has been augmented using a multipole expansion concept to induce a highly phase-selective dissipation of nonlinearly induced dispersion error. The algorithm, cast in a generalized coordinates framework, is expressed in a FORTRAN-appearing notation using a tensor summation index convention for addressing multidimensional problem descriptions. Numerical results document solution accuracy, grid sensitivity, and boundary condition application for compressible flows with shocks. Although still in a very early stage of development and evaluation, the theoretical and programmatic structures of the finite element algorithm do produce an accurate and robust CFD procedure for application to large Reynolds number flows. Further quantization of performance is required and is under way.

## Acknowledgments

This work was supported by the U.S. Air Force, under Grant AFOSR-79-0005, Air Force Wright Aeronautical Laboratory and the National Aeronautics and Space Administration, Langley Research Center, under Grant NAG1-32. The authors wish to acknowledge the considerable generosity of The University of Tennessee Computer Center in providing computer time for the numerical studies.

## References

[1] Zienkiewicz, O. C., *The Finite Element Method,* McGraw Hill Book Co., London, 1977.

[2] MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, 1969.

[3] MacCormack, R. W., "An Efficient Explicit-Implicit-Characteristic Method for Solving the Compressible Navier-Stokes Equations," *Proceedings of SIAM-AMS Symposium on Computational Fluid Dynamics,* New York, 1977.

[4] MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA Paper 81-0110, 1981.

[5] Beam, R. M. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal,* Vol. 16, 1978, pp. 393-402.

[6] Briley, W. R. and McDonald, H., "Solution of the Multi-Dimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," *Journal of Computational Physics,* Vol. 24, 1977, p. 372.

[7] Steger, J. L. and Pulliam, T. H., "An Implicit Finite Difference Code for Inviscid and Viscous Cascade Flow," AIAA Paper 80-1427, 1980.

[8] Thames, F. C., Thompson, J. F., Mastin, C. W., and Walker, R. L., "Numerical Solutions for Viscous and Potential Flow About Arbitrary Two-Dimensional Bodies Using Body-Fitted Coordinate Systems," *Journal of Computational Physics,* Vol. 24, No. 1, 1977, pp. 245-273.

[9] Thompson, J. F. and Mastin, C. W., "Grid Generation Using Differential Systems Techniques," *Numerical Grid Generation Techniques,* NASA CP 2166, 1980, pp. 37-72.

[10] Baker, A. J. and Soliman, M. O., "Utility of a Finite Element Solution Algorithm for Initial-Value Problems," *Journal of Computational Physics,* Vol. 32, No. 3, 1979, pp. 289-324.

[11] Soliman, M. O. and Baker, A. J., "Accuracy and Convergence of a Finite Element Algorithm for Laminar Boundary Layer Flow," *Computers and Fluids,* Vol. 9, 1981, pp. 43-62.

[12] Soliman, M. O. and Baker, A. J., "Accuracy and Convergence of a Finite Element Algorithm for Turbulent Boundary Layer Flow," *Computer Methods in Applied Mechanics and Engineering,* Vol. 28, 1981, pp. 81-102.

[13] Cebeci, T. and Smith, A. M. O., *Analysis of Turbulent Boundary Layers,* Academic Press, New York, 1974.

[14] Prenter, P. M., *Splines and Variational Methods,* John Wiley & Sons, New York, 1975.

[15] *Proceedings of NASA Workshop on Numerical Grid Generation Techniques for Partial Differential Equations,* NASA CP-2166, 1980.

[16] Baker, A. J., "Research on a Finite Element Numerical Algorithm for the Three-Dimensional Navier-Stokes Equations," AFWAL-TR-82-3012, 1982.

[17] Sod, G. A., "A Survey of Several Finite Difference Methods for Systems of Non-Linear Hyperbolic Conservation Laws," *Journal of Computational Physics,* Vol. 27, 1978, pp. 1-31.