JOURNAL OF COMPUTATIONAL PHYSICS 51, 159–188 (1983)

A Comparison of Finite Element and Finite Difference Solutions of the One- and Two-Dimensional Burgers' Equations

C. A. J. FLETCHER

Department of Mechanical Engineering,
University of Sydney, New South Wales 2006, Australia

Received July 6, 1982; revised December 7, 1982

Solutions to the one- and two-dimensional Burgers' equations with moderate to severe internal and boundary gradients have been used to compare minimum truncation error three, five-, and seven-point finite difference schemes with linear, quadratic, and cubic rectangular finite element schemes. The various schemes demonstrate the theoretically predicted convergence rates if the mesh is sufficiently refined. In one dimension the quadratic finite element scheme and the five-point finite difference scheme are computationally the most efficient. In multidimensions the finite element method is less economical than the finite difference method even if the *group representation* is used for the convective terms. In two dimensions the linear group finite element representation and the three-point finite difference scheme are computationally the most efficient on a coarse mesh or if a severe gradient is present.

1. INTRODUCTION

Comparisons of finite difference and finite element methods have been made previously for simple linear partial differential equations [1, 2] and for nonlinear one-dimensional problems [3, 4] of restricted application.

Also realistic fluid flow problems, but with little or no dissipation, have been considered. For example, Haidvogel *et al.* [5] have compared the spectral, finite difference and finite element methods applied to open ocean modelling. Orszag [6] and Gresho *et al.* [7] have used passive scalar convection (the rotating cone problem) to compare the spectral and finite element methods respectively with the finite difference method. However in these comparisons nonlinearities were either not present or played a minor role.

Boundary layer flows, both laminar and turbulent, have been used to compare the finite element and finite difference methods [8–10] and spectral methods [11].

We would like to be able to compare finite element and finite difference methods for dissipative flow problems that strictly require solution of the Navier–Stokes equations. However to make a precise comparison an easily computed exact solution is necessary. For interesting problems governed by the Navier–Stokes equations, exact solutions are not available. As a consequence comparisons have often been

0021-9991/83 $3.00

made [12] in relation to gross features of the flow pattern that are known from experiments.

Here we simplify the Navier–Stokes equations until an exact solution is available but adjust the boundary conditions so that the relatively severe gradients associated with the interaction of the nonlinear convective terms and the dissipative viscous terms are present. In this way it is anticipated that the more important behaviour determined by the Navier–Stokes equations is retained.

Burger's equations are an appropriate *simplified* form of the Navier–Stokes equations. In one and two dimensions these are

$$u_t + uu_x - u_{xx}/\mathrm{Re} = 0, \tag{1}$$

and

$$\begin{aligned} u_t + uu_x + vu_y - (u_{xx} + u_{yy})/\mathrm{Re} &= 0, \\ v_t + uv_x + vv_y - (v_{xx} + v_{yy})/\mathrm{Re} &= 0. \end{aligned} \tag{2}$$

Burger's equations have the same "convective" and "dissipative" form as the incompressible Navier–Stokes equations, although the pressure gradient terms are not retained. Also a solution to Burgers' equation would not, in general, satisfy the continuity equation.

However, for the present purpose, Burgers' equations possess the desirable attribute that *exact solutions can be constructed readily* by invoking the Cole–Hopf transformation [13]. This is true in both one [14] or more [15] dimensions.

We are employing Burgers' equations here as a qualitatively correct approximation of the Navier–Stokes equations. However, for specific flow problems, such as weak shock propagation, acoustic attenuation in fogs, compressible turbulence, and even continuum traffic simulation, Burger's equations are the appropriate governing equations [13].

Solutions to problems with moderate to severe internal and boundary gradients governed by the one- and two-dimensional Burgers' equations will be used to compare the accuracy, economy, and computational efficiency of linear, quadratic, and cubic finite element methods and minimum truncation error three-, five-, and seven-point finite difference methods. The "severe" gradients considered here are still relatively moderate in comparison with those associated with strong shocks, etc.

It would be possible to seek some quantitative measure of the computational efficiency (CE), e.g.,

$$\mathrm{CE} = k/\tau\varepsilon, \tag{3}$$

where τ is the execution (CPU) time or, perhaps, an operation count and ε is the error in the computed solution in some appropriate norm. However, here we will infer the computational efficiency by comparing the accuracy for nominally the same execution time or vice versa.

We expect that results from the present study will help in the choice of an

appropriate method and in the appropriate order of the computational scheme when solving "real" flow problems.

It is well known that if centered finite difference or conventional finite element formulae are used to represent the convective terms, e.g., uu_x, vu_y in Eqs. (1) and (2), then spatial oscillations will occur if the spatial mesh is too coarse for large values of Re (cell Reynolds number effect).

In the present investigation no attempt has been made to include asymmetric (or upwinded) formulae. Therefore the mesh sizes considered have generally been kept smaller than the cell Reynolds number limit. However, since the severity of the gradients increases with Reynolds number, this limit is also *necessary* to achieve reasonable accuracy.

The underlying theory of the finite element method, in particular, has received considerable attention so that it is possible to estimate convergence rates [16] with some precision. Often the theoretical results are restricted to linear problems that can be given a variational interpretation. A secondary purpose of this investigation is to directly measure the convergence rates of a *highly nonlinear* problem to see if the more restricted theoretical results can be extended to more complex governing equations.

It is generally accepted [17] that the finite element is *less* economical than the finite difference method in more than one dimension because of the greater number of connected nodes. This problem is aggravated in the treatment of nonlinearities, like the convective terms, by the conventional finite element method. It will be shown that, by considering the divergence form of the convective terms and by introducing a *group finite element representation* [18], it is possible to overcome this further lack of economy. In many situations the accuracy is increased as well.

2. One-Dimensional Comparison

Two problems will be utilised; first the propagating shock wave is characterised by a severe internal gradient in u. Both steady-state and unsteady solutions will be considered. Second the propagating sine wave is characterised by a severe gradient adjacent to the right-hand boundary. Only unsteady solutions of this problem will be examined.

2.1. *Propagating Shock Wave Problem*

Burgers' equation models, in the simplest manner possible, the balance between nonlinear convection and viscous dissipation. The evolution of this balance for a propagating shock wave is shown in Fig. 1. At $t = 0$ the solution is

$$u(x < 0, 0) = 1, \qquad u(x > 0, 0) = 0, \tag{4}$$

i.e., a "shock" centered at $x = 0$. Subsequently the shock convects to the right and the

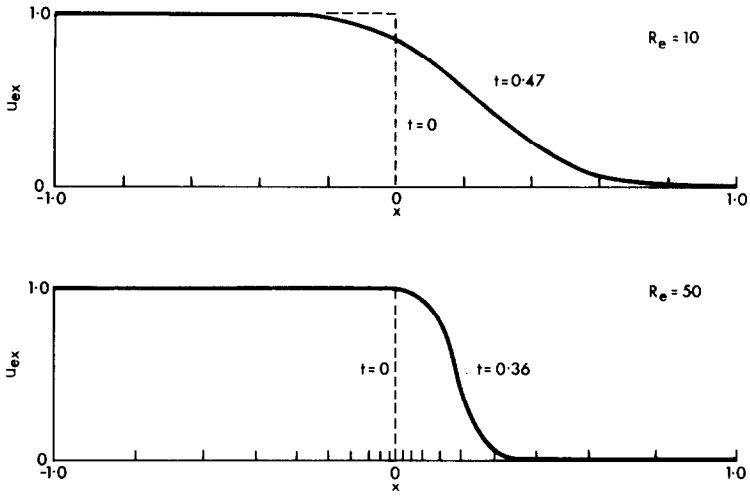

FIG. 1. Exact solutions of the one-dimensional propagating shock problem.

sharp front is smoothed by the dissipative term, u_{xx}/Re. At $\mathrm{Re} = 10$ the distinctive character of the shock is quickly obliterated. At larger values of Re (e.g., $\mathrm{Re} = 50$), the shock can still be identified as a region in which the solution changes rapidly.

The solutions shown in Fig. 1 are exact solutions of Eq. (1) that satisfy Eq. (4) and $u(-\infty, 0) = 1$, $u(\infty, 0) = 0$. To obtain computational solutions of Eq. (1) the following boundary conditions have been applied,

$$u(x_L, t) = 1, \qquad u(x_R, t) = 0, \tag{5}$$

where x_L and x_R are chosen to be sufficiently large so that Eq. (5) is satisfied, e.g., $-x_L = x_R = 2.0$ for $\mathrm{Re} = 10$ and $t \leqslant 0.5$.

Although the method of setting x_L and x_R is satisfactory for unsteady solutions of Eq. (1) close to $t = 0$, the computational region becomes excessive if t_{max} is large. This problem can be avoided by considering a *modified* form of Eq. (1),

$$u_t + (u - \alpha)\, u_x - u_{xx}/\mathrm{Re} = 0, \tag{6}$$

where α is an adjustable constant.

The parameter α can be interpreted as the speed of propagation of a coordinate system imposed on the solution of Eq. (1). If $\alpha = 0.5$ the coordinate system is propagating at the same speed as the shock solution of Eq. (1). Consequently the *average convective motion is frozen* and the shock remains centered at $x = 0$ for all time. As a result computational steady-state solutions of Eq. (6) can be obtained without the need for excessive values of x_L and x_R. In addition the exact steady-state solution of Eq. (6), with $\alpha = 0.5$, is particularly simple,

$$u = 0.5 - 0.5 \tanh(0.25\ \mathrm{Re}\, x). \tag{7}$$

A conventional Galerkin finite element spatial representation of Eq. (6) can be written

$$\mathbf{M}\mathbf{u}_t = \mathbf{S} = (\mathbf{C}/\mathrm{Re} + \alpha\mathbf{E} - \mathbf{B})\mathbf{u}, \tag{8}$$

where elements of the various matrices are given by

$$m_{kj} = (N_k, N_j), \qquad c_{kj} = (dN_k/dx, dN_j/dx),$$

$$e_{kj} = (N_k, dN_j/dx), \qquad \text{and} \qquad b_{kj} = \sum_i u_i(N_k, N_i\, dN_j/dx). \tag{9}$$

In Eq. (9) $N_j(x)$ and $N_k(x)$ are linear, quadratic, or cubic one-dimensional elements. The inner product (a, b) is evaluated as

$$(a, b) = \int_{x_L}^{x_R} ab\, dx.$$

It is apparent that **B**, in Eq. (8), depends on the solution. Consequently **B** must be reevaluated *at every time step.* In contrast the other matrices **M**, **C**, and **E** can be evaluated once and for all.

Also the matrix elements b_{kj} depend on a summation over all contributing nodes in the finite element. This additional algebraic complexity will introduce a substantial increase in execution time in more than one dimension (Section 3). The additional algebraic complexity and consequent increase in execution time can be avoided by adopting a *group representation* [18].

This consists of two parts. First Eq. (6) is put into divergence form, i.e.,

$$u_t + 0.5u_x^2 - \alpha u_x - u_{xx}/\mathrm{Re} = 0. \tag{10}$$

Second a supplementary trial solution is introduced to represent the group of terms u^2, i.e.,

$$u^2 = \sum_j N_j(x)\, u_j^2. \tag{11}$$

After application of a Galerkin finite element formulation, the following equation is obtained in lieu of Eq. (8),

$$\mathbf{M}\mathbf{u}_t = \mathbf{S} = (\mathbf{C}/\mathrm{Re} + \alpha\mathbf{E})\,\mathbf{u} - 0.5\mathbf{E}\mathbf{u}^2. \tag{12}$$

The improved economy of using the group representation in one dimension is relatively minor; a *much larger improvement* will be demonstrated when solving the two-dimensional Burgers' equation (Section 3).

The idea of directly representing a group or product, like u^2, seems to have been proposed first by Swartz and Wendroff [19]. The present application follows the independent use of the same idea by Fletcher and Holt [20] as an adjunct to the

orthonormal (spectral) method of integral relations. More recently the group representation has been used to solve a first-order hyperbolic model equation [21] for inviscid [22] and viscous [23] compressible flow using a least-squares and an ADI Galerkin finite element formulation, respectively.

Christie *et al.* [24] have shown theoretically and numerically that the greater economy of the group representation can be obtained *without a reduction in accuracy*. For the one-dimensional model equation

$$u_t + uu_x = 0$$

with linear elements they show that the truncation error for a uniform grid is *fourth order* whereas the truncation error for the conventional Galerkin formulation is second order. For two-dimensional problems the group representation of first derivatives with linear rectangular elements on a uniform grid also produces a *fourth-order* truncation error [23].

A possible disadvantage of the group representation occurs in the treatment of unsteady flow problems with little or no natural dissipation, i.e., the weather prediction problem. The conventional Galerkin formulation *conserves quadratic properties* like enstrophy, etc. [25] and this tends to suppress the onset of nonlinear instabilities which limit the temporal extent of the solution. In contrast the group representation does not preserve quadratic properties, typically. However, for problems with significant dissipation, like the present situation, the lack of quadratic conservation causes no difficulty.

Three-, five-, and seven-point finite difference schemes have been obtained by minimising the truncation error of the spatial terms. For centered differences on a uniform mesh the algebraic formulae that represent u_x and u_{xx} are shown in Table I.

TABLE I

Finite Difference Formulae on a Uniform Grid

Finite Difference Formula	u_x	u_{xx}
Three point	$(-u_{k-1} + u_{k+1})/2\Delta x$	$(u_{k-1} - 2u_k + u_{k+1})/\Delta x^2$
Five point	$(u_{k-2} - 8u_{k-1} + 8u_{k+1} - u_{k-2})/12\Delta x$	$(-u_{k-2} + 16u_{k-1} - 30u_k + 16u_{k+1} - u_{k+2})/12\Delta x^2$
Seven point	$(-u_{k-3} + 9u_{k-2} - 45u_{k-1} + 45u_{k+1} - 9u_{k+2} + u_{k+3})/60\Delta x$	$(2u_{k-3} - 27u_{k-2} + 270u_{k-1} - 490u_k + 270u_{k+1} - 27u_{k+2} + 2u_{k+3})/180\Delta x^2$

To solve the system of ordinary differential equations represented by Eq. (8) or (12), the following implicit scheme has been employed,

$$\mathbf{M}(\mathbf{u}^{n+1} - \mathbf{u}^n) = \Delta t(1-\theta)\,\mathbf{S}^n + \Delta t\theta \mathbf{S}^{n+1}. \tag{13}$$

The choice $\theta = 0.5$ gives the Crank–Nicolson scheme.

Steady-state solutions of Eqs. (8) and (12) have also been obtained [26] using second- and fourth-order Runge–Kutta schemes. However, the stability restriction on the time step caused the overall execution times (to reach the steady state) to be of the order of 30 to 50 times those produced by Eq. (13). Even for the transient propagating shock problem the Crank–Nicolson scheme *was more economical for the same accuracy* than either of the Runge–Kutta schemes. This was true whether the mass matrix **M** was present (finite element) or not (finite difference).

2.1.1. *Convergence Properties*

The theoretical spatial convergence rates for the various finite element and finite difference schemes are shown in Table II. The theoretical convergence rates for the finite element schemes have been obtained from the corresponding results for *linear* elliptic boundary value problems [16]. In contrast the current problem is highly *nonlinear* and parabolic. The convergence rates for the finite difference schemes are obtained from the truncation errors.

The variation of the error in the L_2 norm with mesh size for steady-state solutions of Eq. (6) with $\alpha = 0.5$ is shown in Fig. 2. The results were generated with the conventional Galerkin finite element formulation, Eq. (8), on a uniform grid. The error in the L_2 norm is defined as follows:

$$\|u - u_{ex}\|_2 = \left[\int_{x_L}^{x_R} (u - u_{ex})^2\, dx\right]^{1/2}. \tag{14}$$

The results presented in Fig. 2 indicate that the theoretical convergence rates are achieved if the mesh is *sufficiently refined.* The coarsest mesh corresponds to a cell Reynolds number of 4. For this mesh quadratic and cubic elements produce a slightly higher accuracy, in relation to the more refined mesh, than predicted theoretically.

TABLE II

Theoretical Spatial Convergence Rates

Finite Difference Method	Spatial Convergence Rate	Finite Element Method	Spatial Convergence Rate
Three point	Δx^2	Linear	Δx^2
Five point	Δx^4	Quadratic	Δx^3
Seven point	Δx^6	Cubic	Δx^4

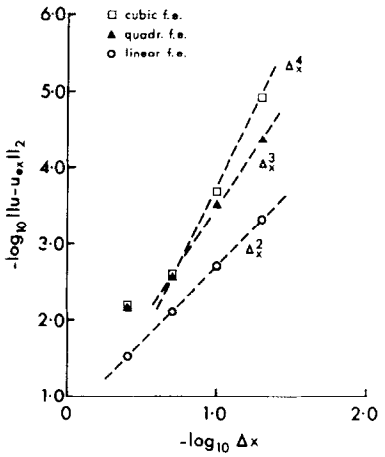

FIG. 2. Spatial convergence properties for conventional finite element steady-state solutions of Eq. (8); Re = 10.

It is of interest that cubic elements only produce more accurate results than quadratic elements on a very refined mesh. For most practical calculations we would expect an error of 1%, i.e., $-\log_{10}\|u-u_{ex}\|_2 = 2$, to be acceptable. In contrast the superior accuracy of cubic elements is only occurring *for errors one to two orders of magnitude less than this.*

Convergence results at Re = 1 and 100 (not shown), although displaced to different absolute error levels, demonstrate the same trend as in Fig. 2 and confirm the theoretically predicted convergence rates on a sufficiently refined mesh.

Convergence results for the group finite element representation, Eqs. (10)–(12), are shown in Fig. 3. The conditions for Fig. 3 are the same as for Fig. 2. It is clear that

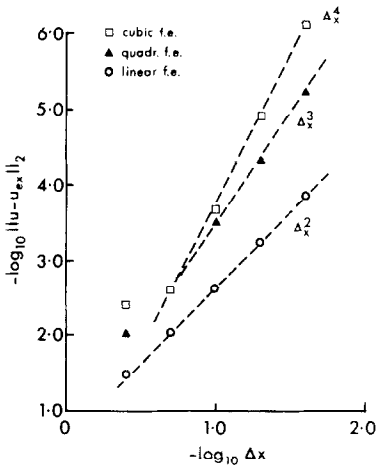

FIG. 3. Spatial convergence properties for group representation finite element steady-state solutions of Eq. (12); Re = 10.

the group representation also achieves the theoretical convergence rates on a refined mesh.

The group representation is as accurate as the conventional finite element formulation except for linear elements on a coarse mesh. Since the group representation is *more economical* than the conventional finite element formulation, it will be used in the rest of Section 2.

Convergence results for the finite difference schemes, applied to Eq. (10), are shown in Fig. 4. The rms error is defined by

$$\|u - u_{ex}\|_{\text{rms}} = \left[\sum_j (u_j - u_{exj})^2\right]^{1/2} \Big/ N_a^{1/2}, \tag{15}$$

where N_a is the number of mesh points at which a computational solution is sought (i.e., all internal points). As with the finite element results the theoretically predicted convergence rates (Table II) were achieved.

The spatial convergence results shown in Figs. 2–4 have been for the steady-state solutions. Since Burgers' equation is parabolic it is of interest to utilise the solution at small time to see if the theoretical convergence rates can still be obtained.

Lack of smoothness of the initial data is expected to limit the convergence rate [16, 17]. To avoid this effect the discontinuous initial condition given by Eq. (4) was replaced by the exact solution at $t = 0.01$. The time step in Eq. (13) was kept sufficiently small so that only spatial discretisation contributed to the errors in the solution. It was found [26] that theoretical convergence rates were substantially achieved although with *less precision* than shown in Figs. 2–4.

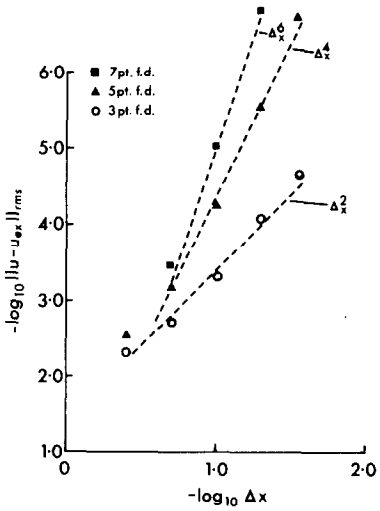

FIG. 4. Spatial convergence properties for finite difference steady-state solutions of Eq. (12): $\text{Re} = 10$.

When the discontinuous initial conditions, Eq. (4) and $u(0) = 0.5$, were used there was a significant effect on the convergence properties of the higher order schemes. This is shown in Fig. 5. It can be seen that the convergence rate for cubic elements is *reduced to second order*. This was also found to be the case for the five- and seven-point finite difference schemes. This result is not unexpected since Majda and Osher [27] have established that discontinuous initial data determine the rate of convergence of a model hyperbolic equation *independently* of the order of the difference scheme. Although the accuracy for cubic elements is higher than for linear elements (Fig. 5), the accuracy of five- and seven-point finite difference schemes was *no greater than* for the three point finite difference scheme.

The convergence results presented in Figs. 2–5 also indicate the accuracy achieved by the various schemes.

2.1.2. *Computational Efficiency*

In determining the computational efficiency the execution time of the various schemes is important. A comparison of the relative execution time per time step is provided by Table III. The results were obtained by integrating for 300 time steps on a uniform grid of 121 points at $\mathrm{Re} = 100$. The comparison is equally valid for the transient and steady-state solutions. The results shown in Table III were obtained on a Perkin–Elmer 3220 computer.

Explicit finite difference schemes are the most economical group, mainly due to the replacement of **M** in Eq. (8) by **I**. Explicit finite element schemes (2R–K) require a once-only factorisation of **M** and two evaluations of $\mathbf{M}^{-1}\mathbf{S}$ (see Eq. 12) per time step. Explicit schemes are typically twice as economical as implicit schemes.

For the implicit schemes, Eq. (13) is manipulated to give an augmented matrix **MA** on the left-hand side. An element of **MA** is given by

$$ma_{kj} = m_{kj} - \Delta t\theta\, \partial S_k/\partial u_j. \tag{16}$$

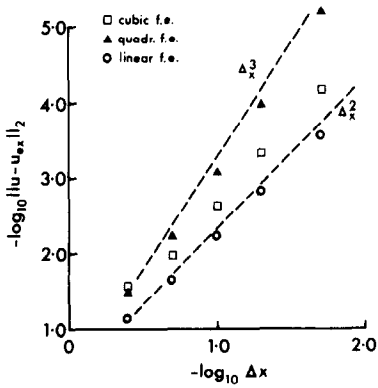

FIG. 5. Spatial convergence properties for group finite element representation with discontinuous initial data; $t = 0.50$, $\mathrm{Re} = 10$.

TABLE III

Relative Execution Time Comparison

Method	Relative Execution Time per Time Step	
	Explicit (2R–K)[a]	Implicit (C–N)[b]
Finite element		
Linear	1.15	2.12
Quadratic	1.50	2.35
Cubic	1.65	2.62
Finite difference		
Three point	1.00	2.27
Five point	1.31	2.65
Seven point	1.62	3.27

[a] 2R–K = second-order Runge–Kutta.
[b] C–N = Crank–Nicolson ($\theta = 0.5$ in Eq. 13).

Since **MA** depends on the solution, through $\partial S_k/\partial u_j$, it must be factorised and $\mathbf{MA}^{-1}\mathbf{S}$ evaluated at every time step. These two steps are executed by a generalised Thomas algorithm which takes advantage of the narrower bandwidth associated with mid-side nodes of quadratic and cubic finite elements. Consequently the higher order, implicit finite element schemes are *more economical* than higher order, implicit finite difference schemes.

The computational efficiency of a given scheme will be interpreted as the accuracy that can be achieved per unit of execution time. This will depend on the severity of any gradients in the solution and on whether the problem is inherently steady or unsteady.

For the steady-state problem, Eq. (6) with $\alpha = 0.5$, the computational efficiency has been assessed using a 121 node variable mesh. The mesh size varied geometrically between $\Delta x = 0.02$ at the shock and $\Delta x = 0.20$ adjacent to the boundaries. Equation (13) has been integrated numerically using a variable time step, implicit scheme from $t = 0.01$ to $t = 8.00$ with $\mathrm{Re} = 100$. At $t = 8.00$ both the computational and exact solutions have reached the steady state. Results are shown in Table IV. To permit a comparison of the finite element and finite difference schemes the nodal point rms error has been utilised.

For the propagating shock problem (Fig. 1) at high Reynolds number the accuracy, and hence the computational efficiency, is dominated by the solution *adjacent to the shock*. Generally the higher order schemes are more accurate (and more efficient) than the three-point schemes. For the particular mesh and problem considered *the five-point finite difference scheme is the most efficient.*

To compare the computational efficiency of various schemes for a parabolic problem, Eq. (13) with $\alpha = 0$ has been integrated from $t = 0.01$ to $t = 0.50$ on a

TABLE IV

Summary of Results for 121 Node Variable Grid "Steady" Solution at Re = 100

Method	Δt_{max}	Number of Time Steps	Relative Exec. Time	$\|u - u_{ex}\|_{rms}$
Finite element				
Linear	0.32	32	1.0	0.94×10^{-3}
Quadratic	0.32	32	1.11	0.50×10^{-3}
Cubic	0.16	56	2.11	0.32×10^{-3}
Finite difference				
Three point	0.32	32	1.06	0.93×10^{-3}
Five point	0.32	32	1.22	0.30×10^{-3}
Seven point	0.32	32	1.50	0.32×10^{-3}

uniform grid at Re = 100. The accuracy that can be achieved for a given execution time depends on the choice for both the temporal and spatial step sizes. For the results shown in Table V, these have been chosen so that the accuracy is maximised for an execution time of approximately the same magnitude as for the linear finite element method.

The results indicate that higher order schemes produce more accurate results than lower order schemes. *The quadratic finite element scheme is the most efficient for this case.* An essential difference between the finite element results shown in Tables IV and V is that **M** plays no part in the accuracy of the steady-state results (Table IV). In contrast the superior accuracy of the finite element results shown in Table V

TABLE V

Summary of Results for a Uniform Mesh "Unsteady" Solution of the Propagating Shock Problem at Re = 100

Method	Δt_{max}	No. of Time Steps	No. of Points	Relative Execution Time	$\|u - u_{ex}\|_{rms}$
Finite element					
Linear	0.008	67	181	1.00	2.03×10^{-4}
Quadratic	0.004	125	91	1.09	0.83×10^{-4}
Cubic	0.004	125	91	1.25	1.93×10^{-4}
Finite difference					
Three point	0.008	67	181	1.07	4.79×10^{-4}
Five point	0.004	125	91	1.24	0.85×10^{-4}
Seven point	0.004	125	91	1.51	0.73×10^{-4}

comes, in part, from **M**. The role of **M** in increasing the accuracy of the finite element method has been noted elsewhere [7, 28]. It is possible [1] that the higher order spatial schemes would be relatively more efficient when used with higher order temporal schemes. This has not been pursued here.

2.2. *Propagating Sine Wave Problem*

The propagating shock problem, whether governed by Eq. (1) or (6), produces solutions that are dominated by an internal gradient (Fig. 1). In contrast the propagating sine wave problem produces solutions that are characterised by a gradient *adjacent to the right-hand boundary.*

Equation (1) is solved with the following initial and boundary conditions,

$$u(x, 0) = \sin \pi x, \qquad 0 \leqslant x \leqslant 1,$$

and

$$u(0, t) = u(1, t) = 0. \tag{17}$$

The exact solution [29] for $\mathrm{Re} = 48$ is shown in Fig. 6; only the downstream region is indicated. As time increases the top part of the sine wave is convected downstream and the amplitude diminishes under the influence of the dissipative term u_{xx}/Re. At $t = 0.5$ a well-defined *boundary layer* has developed adjacent to the downstream boundary. Eventually (at $t = \infty$) u will be zero everywhere.

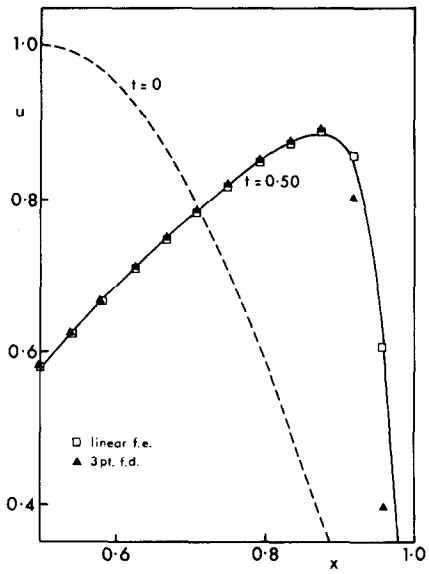

FIG. 6. Exact solutions of the propagating sine wave problem; $\mathrm{Re} = 48$.

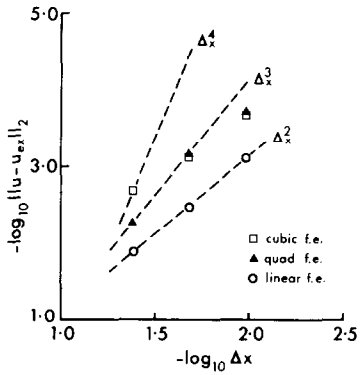

FIG. 7. Spatial convergence properties for group representation finite element solutions of the propagating sine wave; Re = 48; $t = 0.50$.

Typical coarse-mesh solutions, $\Delta x = 1/12$, are shown in Fig. 6. It can be seen that the largest errors in the computational solution occur in the boundary layer region. The linear finite element solutions are clearly more accurate than the three-point finite difference solutions.

2.2.1. *Convergence Properties*

Convergence results for finite element solutions of Eqs. (12) and (17) on a uniform grid at Re = 48 are shown in Fig. 7. It is apparent that the convergence rates for the higher order finite element methods *do not achieve* the theoretical convergence rates. Both quadratic and cubic elements are showing a *nominally second-order* convergence rate. However both schemes are more accurate than the linear finite element scheme.

Corresponding convergence rates for three- and five-point finite difference schemes are shown in Fig. 8. Both schemes are seen to be of nominally second order, although the five-point scheme is considerably more accurate.

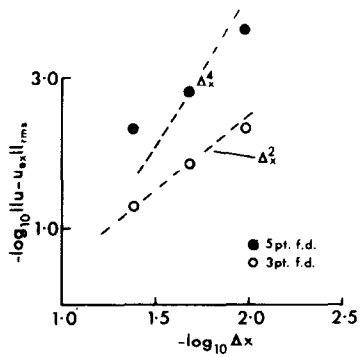

FIG. 8. Spatial convergence properties for finite difference solutions of the propagating sine wave; Re = 48; $t = 0.50$.

A comparison of the results presented in Figs. 7 and 8 to those in Figs. 2–5 indicates that severe boundary gradients reduce the convergence rate more than severe internal gradients.

2.2.2. *Computational Efficiency*

The relative economies of the various schemes shown in Table III also apply to the propagating sine wave problem.

Results are presented in Table VI to compare the computational efficiency of the various schemes. The solutions were obtained on a uniform grid at $\mathrm{Re} = 48$ after a Crank–Nicolson ($\theta = 0.5$) integration from $t = 0$ to $t = 0.50$. The step sizes, Δt and Δx, are chosen to give the smallest execution time for an accuracy comparable to that of the linear finite element scheme.

The three finite element schemes require essentially the same execution time but higher order schemes produce more accurate and hence more efficient solutions. The higher order finite difference schemes are less efficient for this problem because of a relatively large execution time. This follows directly from the *maximum time step* for these categories. This restriction was found necessary to avoid the growth of instabilities in the region of the severe boundary gradient.

All the finite element results shown in Fig. 7 and Table VI have used the group representation. Christie *et al.* [24] have found that the group representation gives *more accurate* solutions than the conventional finite element method for the propagating sine wave problem. They used a Petrov–Galerkin (upwind) formulation with quadratic trial solutions and cubic test functions for $\mathrm{Re} = 100$ and 10,000 and $\Delta x = 1/18$.

The solutions of the one-dimensional Burgers' equation for the propagating shock and propagating sine wave problems have indicated that quadratic finite element

TABLE VI

Uniform Grid Solution of the Propagating Sine Wave Problem
$\mathrm{Re} = 48$, $t = 0.50$

Method	$\Delta t_{\max}$	No. of Time Steps	No. of Points	Relative Execution Time	$\|u - u_{ex}\|_{\mathrm{rms}}$
Finite element					
Linear	0.016	35	49	1.00	1.46×10^{-3}
Quadratic	0.016	35	41	0.97	1.35×10^{-3}
Cubic	0.016	35	37	0.97	1.00×10^{-3}
Finite difference					
Three point	0.016	35	121	2.76	1.45×10^{-3}
Five point	0.004	127	57	5.47	1.20×10^{-3}
Seven point	0.004	127	37	4.40	1.69×10^{-3}

schemes or five-point finite difference schemes are generally the most efficient. However, there is no indication of overwhelming superiority of the finite element or finite difference methods for all problems.

3. Multidimensional Comparison

In this section we compare the finite difference and finite element methods in more than one dimension. First, we infer the relative economy of the finite difference and finite element methods from a consideration of the number of contributing nodes in the equation that is produced after application of the finite difference or finite element method. This general approach allows the effect of increased order or increased dimension on the relative economy to be clarified.

Second, solutions of the two-dimensional Burgers' equation are obtained to provide a direct measure of both the execution time and accuracy. The solutions have been obtained with linear and quadratic finite elements and three- and five-point finite difference schemes for steady problems with both internal and boundary gradients predominantly in the x direction.

3.1. *Number of Contributing Nodes*

The application of the Galerkin finite element method implies an integration over all the dimensions of the problem. This automatically includes all nodes (unless there is fortuitous cancellation), that are in the same element as the Galerkin node, in the subsequent algebraic expression. This situation may be illustrated by the linear finite element representation for u_{xx} in two dimensions. On a uniform mesh the following expression is obtained:

$$u_{xx} \Rightarrow \frac{1}{\Delta x^2}\left[\frac{1}{6}\{u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1}\} + \frac{2}{3}\{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}\}\right.$$
$$\left. + \frac{1}{6}\{u_{i-1,j-1} - 2u_{i,j-1} + u_{i+1,j-1}\}\right]. \tag{18}$$

Subscripts i and j correspond to the x and y directions, respectively. Part of the formula is recognisable as the one-dimensional formula that would be produced by a three-point finite difference representation. However the one-dimensional formula is distributed in the normal direction by the operator $(\frac{1}{6}, \frac{2}{3}, \frac{1}{6})$. Consequently linear finite elements in two dimensions generate *nine-point* formulae in contrast to the three-point finite difference formulae. In three dimensions linear finite elements generate *twenty-seven-point* formulae.

A consequence of the integral nature of the finite element method is that more algebraic manipulation will be required in multidimensions than with a finite difference method. The greater algebraic manipulation translates into a *considerably*

increased execution time. To achieve a comparable computational efficiency there must be a compensating superiority in the accuracy of the finite element solution.

It is known that the application of the Galerkin finite element formulation to $u_x + v_y = 0$ produces a fourth-order accurate expression on a uniform two-dimensional grid. The corresponding three-point finite difference formula is only second-order accurate. However the higher accuracy comes from *fortuitous cancellation* associated with the coefficients in the perpendicular operator $(\frac{1}{6}, \frac{2}{3}, \frac{1}{6})$, and does not extend to the inclusion of u_{xx}, etc.

An appreciation of the relative algebraic complexity (and hence economy) of the finite element and finite difference methods can be obtained by considering the solution of Laplace's equation,

$$\nabla^2 u = 0. \tag{19}$$

Application of the finite element or finite difference methods results in a system of algebraic equations that can be written

$$\mathbf{Ku} = \mathbf{B}. \tag{20}$$

In Eq. (20) **u** is the solution sought, **B** arises from the known u values on the boundary, and **K** contains the algebraic coefficients arising from the application of the finite element or finite difference method. Each row of **K** contains as many nonzero terms as there are connected nodes (discounting cancellation).

If Eq. (20) is solved iteratively then the execution time per iteration would be proportional to the number of nonzero terms in each row. For the finite difference method applied to internal nodes not connected to the boundary all rows will contain

TABLE VII

Average Number of Nonzero Terms in "Internal" Rows of **K**

	Lagrangian Finite Element		Finite Difference	
Dimension	Order of Shape Function	Av. no. of Terms	n Point Formula	Number of Terms
1	Linear	3	3	3
	Quadratic	4	5	5
	Cubic	5	7	7
2				
	Linear	9	3	5
	Quadratic	17	5	9
	Cubic	29	7	13
3				
	Linear	27	3	7
	Quadratic	70	5	13
	Cubic	140	7	19

the same number of nonzero terms. If Eq. (20) is solved by a direct bandwidth solver, the execution time will be proportional to the square of the number of nonzero terms in each row [30].

For various dimensions and orders of representation the average number of nonzero terms in each row is shown in Table VII. The finite element values have been calculated from Lagrangian shape functions. For quadratic and cubic shape functions an rms average number of nonzero terms per row has been computed. The number for both formulations is appropriate to internal nodes which have no connected boundary nodes. For the higher dimensional finite difference schemes three-, five-, and seven-point formulae have been assumed in each coordinate direction.

In one dimension the finite element method has fewer nonzero terms on average than the finite difference method. Although, as Table II indicates, a cubic finite element formulation is expected to generate solutions of comparable accuracy to a five-point finite difference scheme. If the comparison of the number of nonzero terms is made for comparable convergence rates (Table II), then the one-dimensional finite difference and finite element methods are *equally economical.*

In more than one dimension the finite element method has *significantly more* nonzero terms per row than the finite difference method. The difference becomes substantially greater if *higher order elements* are considered. Thus the use of higher order elements in multidimensions will need to generate solutions of very high accuracy if competitive computational efficiency with the finite difference method is to be achieved.

The comparison made in Table VII is appropriate to linear terms like u_{xx} and u_{yy} in Eq. (2). However the conventional finite element treatment of the convective terms, e.g., uu_x or vu_y in Eq. (2), introduces a *further increase* in algebraic complexity in that each nodal contribution to u can be linked to each nodal contribution to u_x in the same element. In viscous compressible flow [23] triple products, like ρvu_y, occur which produces an even greater increase in algebraic complexity. Fortunately the *group representation* [18], when applicable, restores the level of algebraic complexity shown in Table VII.

3.2. *Two-Dimensional Burgers' Equation*

In this section the two-dimensional Burgers' equation (2) will be used to see if the finite element method can provide sufficiently accurate solutions to compensate for the reduced economy discussed in Section 3.1. In view of the previous remarks concerning nonlinear terms, the linear conventional, linear group, and quadratic group finite element representations will be compared to the three-point and five-point finite difference methods.

As with the one-dimensional problems, the two-dimensional Burgers' equations will be solved for problems that are dominated by either internal or boundary gradients. This will make them more representative of "difficult" fluid-dynamic problems.

In order to make precise determinations of the error in the computational solution it is necessary to have exact solutions of Burgers' equations. An exact solution of the

two-dimensional Burgers' equations can be generated efficiently [15] by making use of the Cole–Hopf transformation [13],

$$u = -\left(\frac{2}{\mathrm{Re}}\right)\frac{\phi_x}{\phi} \qquad \text{and} \qquad v = -\left(\frac{2}{\mathrm{Re}}\right)\frac{\phi_y}{\phi}, \tag{21}$$

where ϕ is the solution of

$$\phi_t = \phi_{xx} + \phi_{yy}. \tag{22}$$

In the present investigation only steady-state solutions of Eq. (2) will be considered. Consequently if a general steady-state solution of Eq. (22) is postulated [15], evaluation of Eq. (21) gives the following expressions for u and v,

$$u = -\frac{2}{\mathrm{Re}}\left[\frac{a_1 + a_3 y + ka_4\{\exp(x - x_0)) - \exp(-k(x - x_0))\}\cos ky}{a_0 + a_1 x + a_2 y + a_3 xy + a_4\{\exp(k(x - x_0)) + \exp(-k(x - x_0))\}\sin ky}\right]. \tag{23}$$

and

$$v = -\frac{2}{\mathrm{Re}}\left[\frac{a_2 + a_3 x - ka_4\{\exp(k(x - x_0)) + \exp(-k(x - x_0))\}\sin ky}{a_0 + a_1 x + a_2 y + a_3 xy + a_4\{\exp(k(x - x_0)) + \exp(-k(x - x_0))\}\cos ky}\right]. \tag{24}$$

Equations (23) and (24) are solutions of the two-dimensional Burgers' equations (2). By varying the various coefficients, a_0, a_1, etc., in Eqs. (23) and (24) different related exact solutions can be constructed. The solutions for u and v shown in Fig. 9 contain a moderate gradient predominantly in the x direction. The following coefficient values were used to generate that solution:

$$a_0 = a_1 = 110.13, \qquad a_2 = a_3 = 0, \qquad a_4 = 1.0,$$
$$k = 5, \qquad x_0 = 1, \qquad \mathrm{Re} = 10.$$

The form of the Burgers' equations represented by Eq. (2) does not permit use of the group representation for the convective terms discussed earlier. However, a group representation is possible for the *inhomogeneous* form of the Burgers' equations, in which a source term is included to eliminate the additional terms associated with the "divergence" form for the convective terms

$$u_t + (u^2)_x + (uv)_y - \frac{1}{\mathrm{Re}}(u_{xx} + u_{yy}) = SX, \tag{25}$$

and

$$v_t + (uv)_x + (v^2)_y - \frac{1}{\mathrm{Re}}(v_{xx} + v_{yy}) = SY, \tag{26}$$

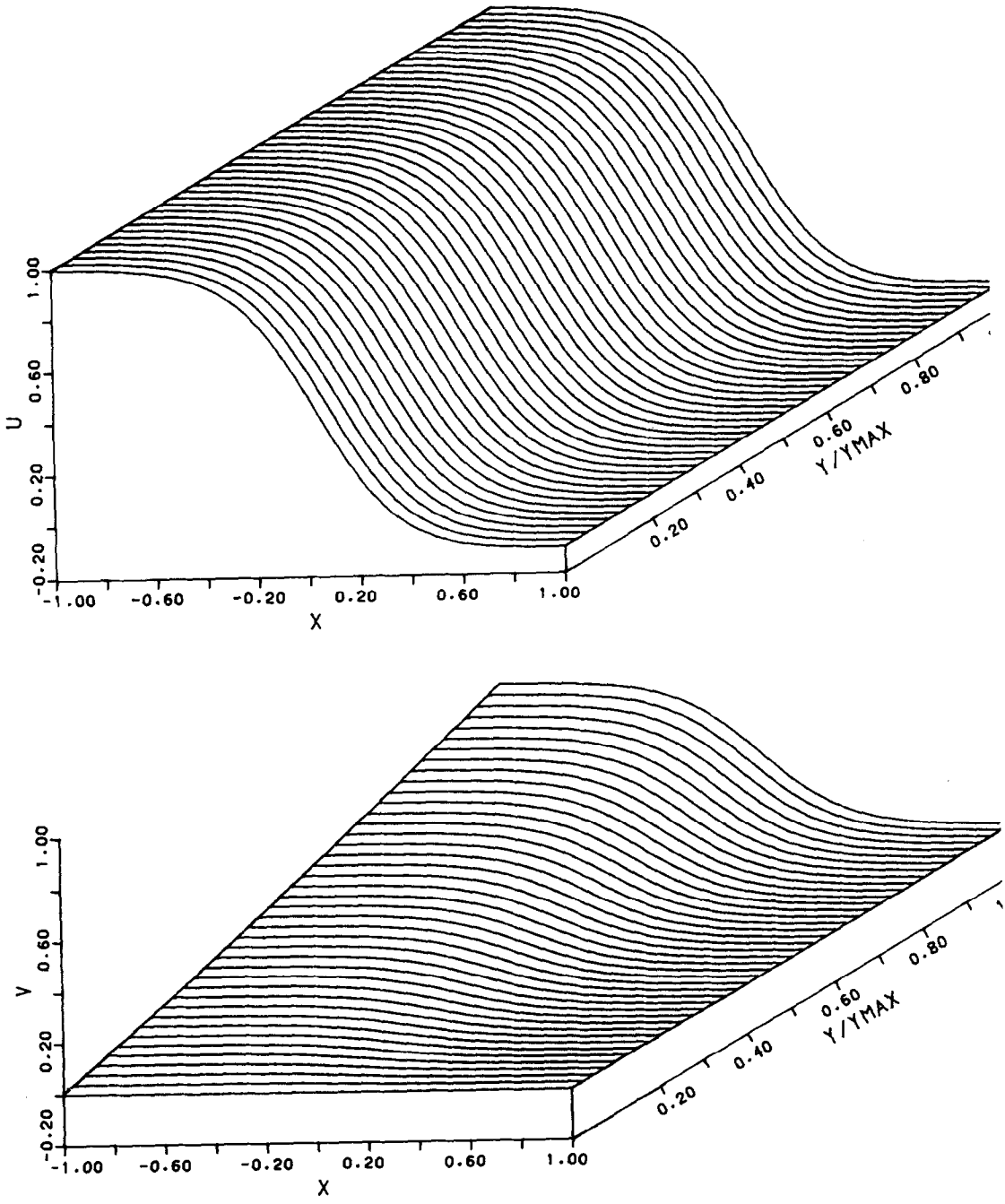

FIG. 9. Exact steady-state solutions of the two-dimensional Burgers' equations with a moderate internal gradient.

where the source terms, SX and SY, are given by

$$SX = 0.5 \operatorname{Re} u(u^2 + v^2), \qquad SY = 0.5 \operatorname{Re} v(u^2 + v^2). \tag{27}$$

The exact steady-state solutions of Eqs. (25) and (26) are given by Eqs. (23) and (24). Equations (25) and (26), with Dirichlet boundary conditions given by Eqs. (23) and (24), are solved numerically to compare the accuracy, economy, and computational efficiency of the computational schemes shown in Table VIII.

All the finite element schemes described in Table VIII have used Lagrangian shape functions on a rectangular mesh. For all the finite element schemes a trial solution has been introduced for the source terms directly, e.g.,

$$SX = \sum_j N_j(x, y)\, \overline{SX}_j, \tag{28}$$

where N_j are linear or quadratic shape functions. It is well established that this procedure is consistent [31] with the use of the same order of trial functions for u and v and introduces an important economy [3]. Clearly the treatment of the source terms is equivalent to the group representation of the convective terms.

To obtain steady-state solutions of Eqs. (25) and (26) a *time-split algorithm* [17] has been constructed. This entails the repeated factorisation and solution, using a generalised Thomas algorithm, of a linear system of equations along each grid line in the x and y directions. Conceptually the time-split algorithm can be interpreted as a generalised ADI method [32].

The exact solutions, Eqs. (23) and (24), have provided the starting solution so that relatively few time steps are required to obtain the exact solutions of the algebraic system of equations. Convergence was assumed if the rms steady-state residual was less than 1×10^{-9}. Typically this was achieved in 20 to 30 time steps.

3.2.1. *Relative Execution Times*

The execution time per time step was taken as a measure of the economy of the various schemes. The execution time per time step is dominated by the evaluation of the *steady-state residual*; this is required once per time step. Consequently whether the steady-state equations are solved iteratively or with a pseudotransient procedure,

TABLE VIII

Schemes for Solving the Two-Dimensional Burgers' Equations

Scheme	Description
3–FD	Three-point finite difference formulae
LFE(C)	Conventional linear finite element representation
LFE(G)	Linear group f.e. representation for u^2, uv, etc.
QFE(G)	Quadratic group f.e. representation for u^2, uv, etc.
5–FD	Five-point finite difference formulae

as in the present situation, the execution time per time step will give a reasonable indication of the relative economy of the various schemes shown in Table VIII.

Solutions to Eqs. (25) and (26) have been obtained on a rectangular mesh, $-1 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant \pi/6k$, with a uniform grid in each direction. The relative execution times for the various schemes with increasing mesh refinement is shown in Table IX. The results were obtained on a CYBER–172.

The results shown in Table IX indicate that the finite element schemes are less economical than comparable finite difference schemes. The conventional linear finite element formulation is about *seven times* less economical than the three-point finite difference scheme and the linear group finite element representation is about *two and a half times* less economical. The quadratic finite element scheme and the five-point finite difference scheme are typically *two and a half to three times* less economical than the linear finite element or three-point finite difference schemes respectively.

The poor economy of the conventional finite element method comes from the inefficient treatment of the convective terms. Disregarding this method the relative execution times of the other methods correspond approximately to the relative number of nonzero terms in each row shown in Table VII.

3.2.2. *Convergence Properties and Accuracy*

To permit a direct comparison of the finite difference and finite element solutions, rms errors in the nodal solutions for u are shown in Figs. 10, 12, and 14. The rms errors in the solutions for v are typically 25% of those for the u solutions and otherwise follow the same trend with mesh size and method.

For a moderate internal gradient the variation in the rms error with mesh size is shown in Fig. 10. The corresponding exact solution for a moderate internal gradient is shown in Fig. 9.

The convergence properties of the various schemes conform to Table II, at least on a sufficiently refined mesh. However, the two-dimensional results do not demonstrate the same level of agreement as the one-dimensional results.

For a given mesh size the 3–FD scheme is least accurate and the 5–FD scheme is most accurate. The LFE(C) scheme is slightly more accurate that the LFE(G)

TABLE IX

Comparison of Relative Execution Times per Time Step

Mesh	Δx	3 FD	LFE(C)	LFE(G)	QFE(G)	5–FD
6×6	0.4	1.0	4.9	2.2		2.9
7×7	0.3333				7.9	
11×11	0.2	4.0	25.3	10.2	25.9	13.0
21×21	0.1	16.1	111.2	45.3	110.7	53.9
41×41	0.05	67.8	478.1	189.8	464.1	224.7

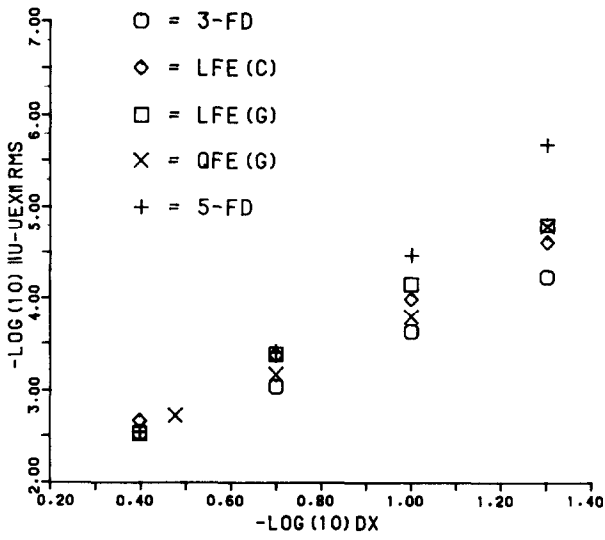

FIG. 10. Spatial convergence properties for a moderate internal gradient; Re = 10.

scheme on a coarse mesh and slightly less accurate on a refined mesh. A surprising result is that QFE(G) is less accurate than the LFE(G) scheme. However, it may be more accurate on a sufficiently refined mesh. On a relatively coarse mesh the 3–FD, LFE(G), and 5–FD schemes are all demonstrating comparable computational efficiency. However, on a very refined mesh the 5–FD scheme is more efficient than the other schemes.

An exact solution, demonstrating a severe internal gradient, is shown in Fig. 11. To generate the results the following parameter values were used in Eqs. (23) and (24),

$$a_0 = a_1 = 1.2962 \times 10^{13}, \qquad a_2 = a_3 = 0, \qquad a_4 = 1.0,$$
$$k = 25, \qquad x_0 = 1, \qquad \mathrm{Re} = 50. \tag{29}$$

The corresponding rms errors in the computational solutions for u are shown in Fig. 12.

The results indicate that all schemes produce *nominally second-order convergence* and that higher order schemes are not generating more accurate solutions even on a refined mesh. On a coarse mesh the 3–FD scheme is the most efficient scheme and on a refined mesh the 3–FD and LFE(G) are the most efficient schemes. In terms of the number of mesh points to represent the gradient, $\Delta x = 0.05$ for the severe internal gradient is approximately equivalent to $\Delta x = 0.20$ for a moderate internal gradient.

The exact solution shown in Fig. 13 includes a moderate gradient for u in the x

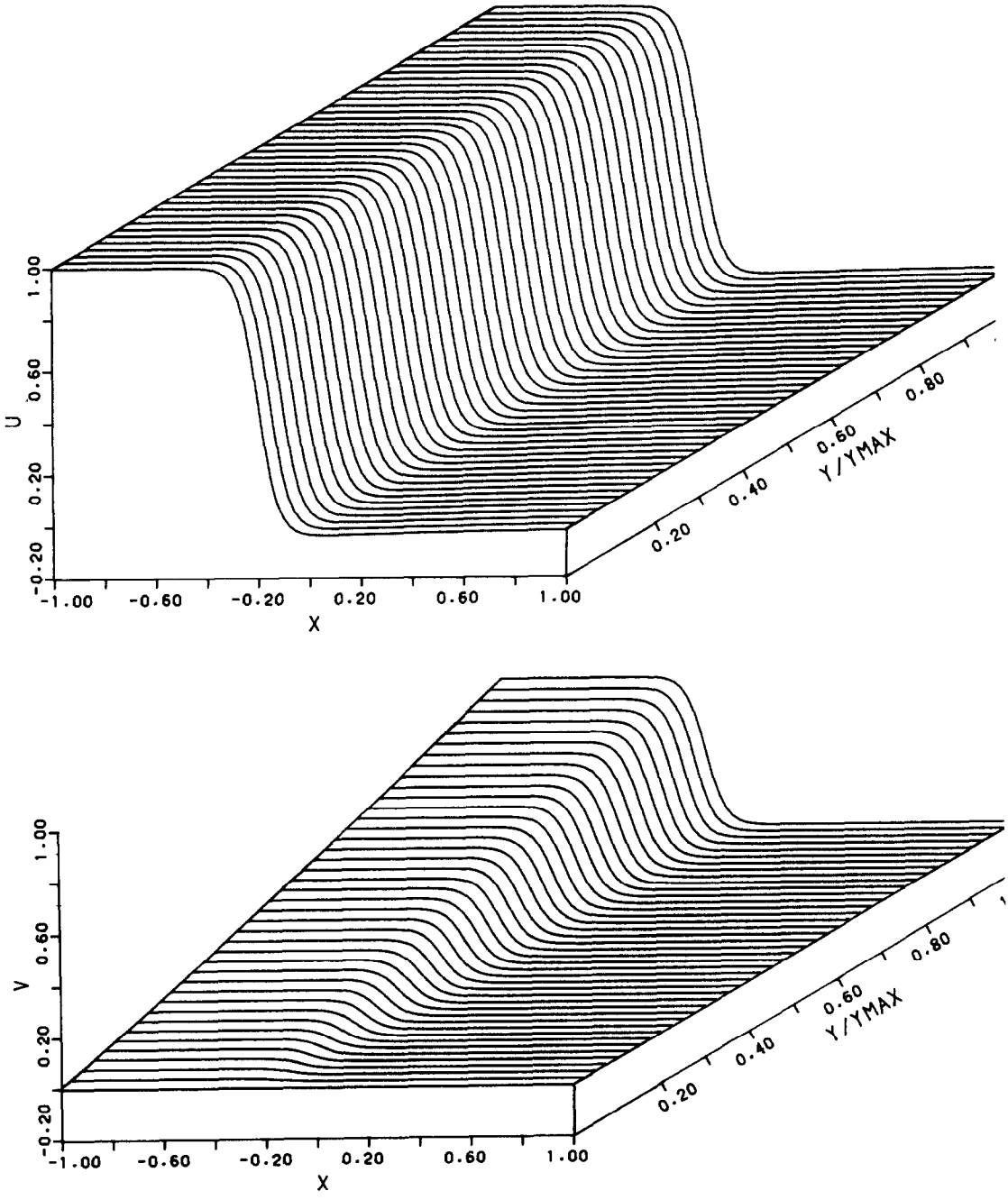

FIG. 11. Exact steady-state solutions of the two-dimensional Burgers' equations with a severe internal gradient.

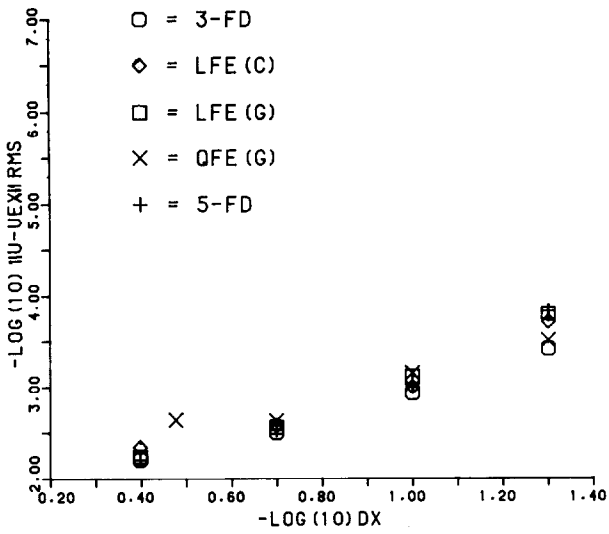

FIG. 12. Spatial convergence properties for a severe internal gradient; Re = 50.

direction adjacent to the right-hand boundary. To generate this solution the following parameter values were used in Eqs. (23) and (24),

$$a_0 = a_1 = 0.011013, \qquad a_2 = a_3 = 0, \qquad a_4 = 1.0,$$
$$k = 5, \qquad x_0 = 1.0, \qquad \text{and} \qquad \text{Re} = 10. \tag{30}$$

The errors in the computational solutions for u are shown in Fig. 14.

The relative accuracies and computational efficiencies correspond approximately to those for the moderate internal gradient shown in Fig. 10. However, for the moderate boundary gradient the LFE(C) scheme generates more accurate results than the LFE(G) scheme on a refined mesh, although less efficiently.

We conclude this section with the observation that on a coarse mesh, or *where sharp gradients are expected*, the 3–FD or LFE(G) schemes are the most efficient. On a refined mesh the 5–FD scheme is most efficient.

4. DISCUSSION

Because of their similarity to the momentum equations that govern incompressible, viscous fluid flow, Burgers' equations have been used to compare the various finite element and finite difference methods. However, Burgers' equations have the advantage of possessing easily computed *exact solutions*.

Viscous flow past an obstruction or adjacent to a wall is characterised by two different length scales. Consequently flow fields demonstrate rapid changes in the velocity components over short distances, i.e., severe gradients.

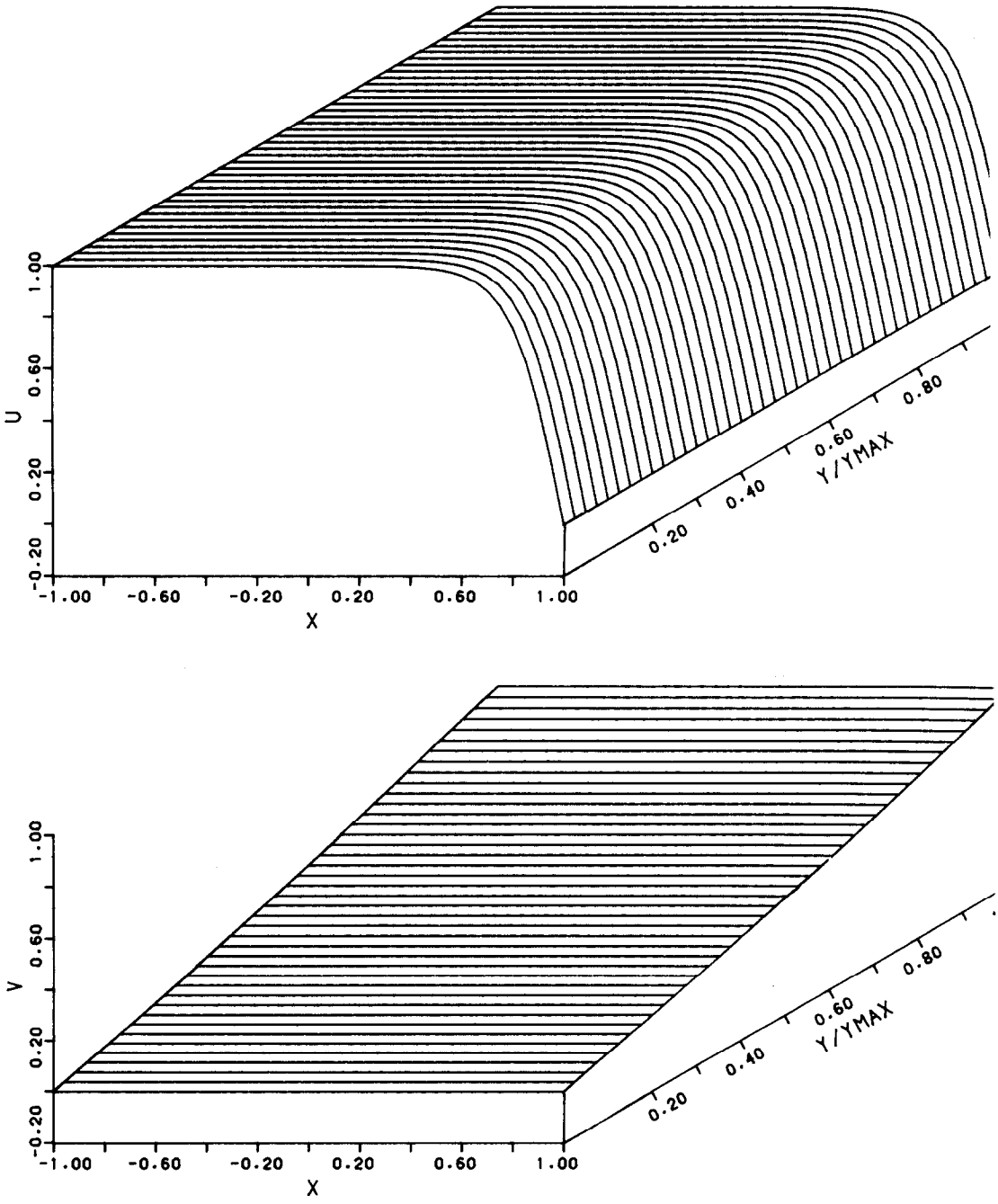

FIG. 13. Exact steady-state solutions of the two-dimensional Burgers' equations with a moderate boundary gradient.

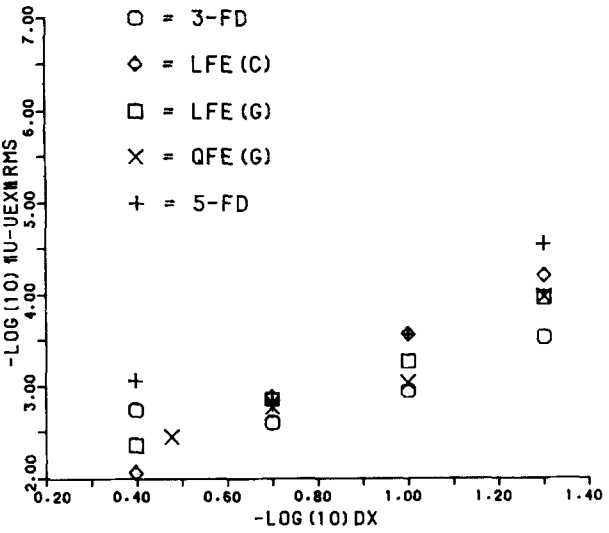

FIG. 14. Spatial convergence properties for a moderate boundary gradient; Re = 10.

The errors in a computational solution are largest, typically, where significant gradients in the dependent variables occur. Thus solutions to Burgers' equation with severe gradients in the interior or adjacent to a boundary can be expected to provide guidance for the behaviour of the various computational schemes applied to real fluid flows.

The solutions to the one-dimensional Burgers' equation (Section 2) indicate that quadratic finite element and five-point finite difference schemes are generally computationally more efficient than linear finite element or three-point finite difference schemes.

The efficiency has been compared on sufficiently coarse meshes *to be appropriate to practical applications*. The convergence data (e.g., Fig. 3 or 7) suggests that as the mesh is refined the computationally most efficient order increases. This is because the accuracy increases more rapidly for higher order schemes as the mesh is refined whereas the reduction in economy is relatively small.

As indicated by Table III the quadratic finite element scheme is only 11% less economical than the linear finite element scheme and the five-point finite difference scheme is only 17% less economical than the three-point scheme.

The relatively small penalty in reduced economy in using a higher order formulation, that occurs in one dimension, is not a feature of two-dimensional calculations. In two dimensions the quadratic finite element scheme is *about* 150% *less economical* than the linear finite element scheme and the five-point finite difference scheme is *about* 250% *less economical* than the three-point finite difference scheme.

In one dimension the relative computational efficiency of finite difference and finite element schemes depends on whether the problem is steady or unsteady. For steady problems the mass matrix **M** in Eq. (8) has no influence on the accuracy. Not surprisingly, steady solutions of the modified Burgers' equation were obtained most efficiently by the five-point finite difference scheme. However, the transient solutions of Burgers' equation were obtained most efficiently by the quadratic finite element scheme.

For multidimensional problems the nature of the finite element method gives rise to connected nodes in the total region (all dimensions) surrounding the node at which the equation is formed. Where nonlinearities arise, as in the convective terms, a product of connected nodes occurs within each element, unless the group representation is intrduced. In contrast the finite difference method only involves nodes appropriate to the direction of the derivative. For nonderivative terms only a single nodal value is required.

We expect the execution time to depend on the number of nonzero terms in the steady-state residual, whether an unsteady or a steady problem is being considered. Consequently we find that finite difference methods are typically more economical in two dimensions than finite element methods (Table VIII), even after the group representation [18] is introduced. The linear group finite element representation is *about* 150% *less economical* than the three-point finite difference scheme in two dimensions. In one dimension it was 7% more economical.

From a consideration of the number of contributing nodes (Table VII) it is clear that the finite element method will be progressively less economical in comparison with the finite difference method *as the order or the number of dimensions is increased.*

For the two-dimensional steady solutions of Burgers' equations the five-point finite difference scheme is considerably more accurate than the three-point finite difference scheme. However, the linear and quadratic finite element schemes are of comparable accuracy even on a refined mesh.

Because of the relatively poor economy finite element schemes are typically computationally less efficient than finite difference schemes. However, the linear group finite element representation is competitive on a coarse mesh or if a severe gradient is present. The conventional finite element method is never competitive due to the *uneconomical treatment* of the convective terms.

In attempting to extrapolate the present results for Burgers' equaions to the Navier–Stokes equations a certain amount of caution is necessary. First the inhomogeneous two-dimensional Burgers' equations do not include pressure gradient terms but do include source terms. Also no account is taken of the continuity equation.

Second fluid flow around bodies of nonsimple shape introduces the complication of an irregular mesh. Third the boundary conditions for a flow problem will include, typically, Neumann boundary conditions on downstream boundaries.

The additional complexities of a real fluid flow will not influence the relative economy of the finite element and finite difference methods. The relative economy is

a property of the method and *not problem dependent.* However, the additional complexities may alter the relative accuracies and convergence rates of the two methods that have been reported here for Burgers' equations, with predominantly uniform meshes.

5. Conclusions

Solutions have been obtained to the one- and two-dimensional Burgers' equations with boundary conditions that generate moderate to severe gradients predominantly in the flow direction. Linear, quadratic, and cubic finite element schemes have been compared with three-, five-, and seven-point finite difference schemes for both steady and unsteady problems on the basis of convergence rate, economy, accuracy, and computational efficiency.

The following specific conclusions have been reached:

(i) Theoretical convergence rates are achieved unless very severe gradients or discontinuities occur.

(ii) In one dimension higher order schemes are almost as economical as three-point schemes. In multidimensions higher order schemes (particularly finite element schemes) are much less economical than three-point schemes.

(iii) In multidimensions finite element schemes are less economical than finite difference schemes, due to the integral construction of the algebraic formulae.

(iv) The conventional finite element treatment of nonlinearities, like the convective terms, adds a further economic penalty. This penalty can be avoided without loss of accuracy by adopting the group finite element representation [18].

(v) Steady solutions of the modified one-dimensional Burgers' equation are obtained most efficiently by the five-point finite difference scheme. Transient solutions of the one-dimensional Burgers' equation are obtained most efficiently by the quadratic group finite element representation.

(vi) For two-dimensional steady solutions of the inhomogeneous Burgers' equation the three-point finite difference and the linear group finite element representation are most efficient on a coarse mesh or if a severe gradient is present. For a refined mesh the five-point finite difference scheme is most efficient.

References

1. B. Swartz, *in* "Mathematical Aspects of Finite Elements in Partial Differential Equations" (C. de Boor, Ed.), pp. 297–312, Academic Press, New York, 1974.
2. B. Swartz and B. Wendroff, *SIAM J. Numer. Anal.* **11** (1974), 979.
3. W. E. Culham and R. S. Varga, *Soc. Pet. Eng. J.* **11** (1971), 374.
4. T. R. Hopkins and R. Wait, *Int. J. Numer. Methods Eng.* **12** (1978), 1081.
5. D. B. Haidvogel, A. R. Robinson, and E. E. Schulman, *J. Comput. Phys.* **34** (1980), 1.

6. S. A. ORSZAG, *J. Fluid Mech.* **49** (1971), 75.
7. P. M. GRESHO, R. L. LEE, AND R. L. SANI, "Finite Elements in Fluids," Vol. 3, pp. 335–350, Wiley, London, 1978.
8. M. O. SOLIMAN AND A. J. BAKER, *Comput. Fluids* **9** (1981), 43.
9. M. O. SOLIMAN AND A. J. BAKER, *Comput. Methods. Appl. Mech. Eng.* **28** (1981), 81.
10. C. A. J. FLETCHER AND R. W. FLEET, *in* "Eighth Int. Conf. on Num. Meth. in Fluid Dynamics," Aachen, June 1982.
11. R. W. FLEET AND C. A. J. FLETCHER, *in* "Fourth Int. Conf. in Australia on Finite Element Methods," (P. Hoadley, Ed.), pp. 59–63, Melbourne, Aug. 1982.
12. S. Y. TUANN AND M. D. OLSON, *J. Comput. Phys.* **29** (1978), 1.
13. C. A. J. FLETCHER, *in* "Numerical Solution of Partial Differential Equations" (J. Noye, Ed.), pp. 139–225, North-Holland, Amsterdam, 1982.
14. E. R. BENTON AND G. W. PLATZMAN, *Q. Appl. Math.* **30** (1972), 195.
15. C. A. J. FLETCHER, *Int. J. Numer. Methods Fluids*, (1983), to appear.
16. J. T. ODEN AND J. N. REDDY, "An Introduction to the Mathematical Theory of Finite Elements," Wiley, New York, 1976.
17. C. A. J. FLETCHER, "Computational Galerkin Methods," Springer-Verlag, Heidelberg, 1983.
18. C. A. J. FLETCHER, *Comput. Methods Appl. Mech. Eng.* (1983), to appear.
19. B. SWARTZ AND B. WENDROFF, *Math. Comput.* **23** (1969), 37.
20. C. A. J. FLETCHER AND M. HOLT, *J. Fluid Mech.* **74** (1976), 561.
21. R. C. Y. CHIN, G. W. HEDSTROM, AND K. E. KARLSSON, *Math. Comput.* **33** (1979), 647.
22. C. A. J. FLETCHER, *J. Comput. Phys.* **33** (1979), 301.
23. C. A. J. FLETCHER, *Comput. Methods. Appl. Mech. Eng.* **30** (1982), 307.
24. I. CHRISTIE, D. F. GRIFFITHS, A. R. MITCHELL, AND J. M. SANZ-SERNA, *Inst. Math. Appl. J. Numer. Anal.* **1** (1981), 253.
25. K. W. MORTON, *in* "The State of the Art in Numerical Analysis" (D. Jacob, Ed.), pp. 699–756, Academic Press, New York/London, 1977.
26. C. A. J. FLETCHER, *in* "Finite Element Flow Analysis" (T. Kawai, Ed.), pp. 1003–1010, Univ. of Tokyo Press, 1982.
27. A. MAJDA AND S. OSHER, *Commun. Pure Appl. Math.* **30** (1977), 671.
28. A. J. BAKER AND M. O. SOLIMAN, *J. Comput. Phys.* **32** (1979), 289.
29. J. D. COLE, *Q. Appl. Math.* **9** (1951), 225.
30. A. JENNINGS, "Matrix Computation for Engineers and Scientists," p. 152, Wiley, London, 1977.
31. G. STRANG AND G. J. FIX, "An Analysis of the Finite Element Method," p. 30, Prentice-Hall, Englewood Cliffs, N. J., 1973.
32. A. R. GOURLAY, *in* "The State of the Art in Numerical Analysis" (D. Jacob, Ed.), pp. 757–796, Academic Press, New York/London, 1977.