JOURNAL OF COMPUTATIONAL PHYSICS **40**, 202–249 (1981)

The Moving Finite Element Method: Applications to General Partial Differential Equations with Multiple Large Gradients*

R. J. GELINAS AND S. K. DOSS

Science Applications, Inc., Pleasanton, California 94566

AND

KEITH MILLER

Mathematics Department, University of California, Berkeley, California 94720

Received December 18, 1979; revised May 12, 1980

The moving finite element (MFE) method has been reduced to practice in the automatic solution program DYLA for general systems of transient partial differential equations (PDEs) in 1-D. Several test examples are presented which illustrate the unique node movement and systematic control features which are intrinsic in the MFE method. Computational dilemmas of numerical diffusion, Gibbs overshooting and undershooting, zone tangling, and grid remap (or re-connection) aliasing, which occur frequently in conventional PDE methods, are essentially *eliminated* in the MFE method. Arbitrarily large gradients (or shocks) can be solved with extremely high resolution and accuracy for non-coincident, or even counterdirected, propagating wavefronts. Boundary layers of arbitrarily small dimensions are solved with high accuracy simultaneously with the large-scale structures in reactive and non-reactive fluid calculations. The MFE method requires a small fraction of the grid nodes which are used in conventional PDE solution methods because the nodes migrate continuously and systematically to those positions where they are most needed in order to yield accurate PDE solutions on entire problem domains. Courant–Friedrichs–Lewy time-step limits are exceeded by wide margins (by factors of two to several thousand). Finally, the extension of the MFE method to 2-D is briefly discussed.

Contents. Introduction. I. The Moving Finite Element Method. II. Sample Problems and Results. III. Looking Ahead. Appendix. References.

INTRODUCTION

This article presents some promising advances in the automatic solution of partial differential equation (PDE) systems for general scientific applications. The advances

---

* This research was supported by the Basic Energy Science Office of the Department of Energy under Contract ER-78-C-03-2078.

0021-9991/81/030202–48$02.00/0

which we report result from the reduction-to-practice of a moving finite element (MFE) method which was initially introduced by Miller and Miller [1] and subsequently improved and extended by Miller [2].

In the MFE method, both nodal amplitudes and nodal positions move continuously with time in such a way as to satisfy simultaneous ordinary differential equations (ODEs) which minimize PDE residuals. The computational experience with this approach is that the nodes generally move to those regions where they are most needed in order to resolve PDE solutions with high accuracy. Conventional numerical dilemmas of numerical diffusion and Gibbs overshooting and undershooting have been essentially eliminated by the MFE method in Eulerian calculations, and Lagrangian zone tangling and/or re-map aliasing are not present, either conceptually or practically, in the MFE solution of PDEs. Furthermore, the general integral nature of the MFE method has proven to be very amenable to the practical formulation of automated PDE solvers, such as the DYLA program which is described in this article. We note further that many of the key factors which led to very successful developments of automatic ODE solvers for stiff systems during recent years [3–5] are now becoming visible in our PDE developments as well; so let us review briefly some of these factors and note their logical extensions in the case of PDE systems.

The resolution of dilemmas posed by stiffness was pivotal to the remarkable advances which have occurred in semi-automatic numerical solution methods for ODEs during the past 5–10 years. Recall that, in *ODE* systems, stiffness is encountered when relaxation time constants of greatly disparate magnitudes are present simultaneously, making numerical stability and accuracy more difficult to achieve than usual. In *PDE* systems similar phenomena (which we choose to also term stiffness) will often afflict both the space and the time variables. For example, flame front behavior in coupled hydrodynamic-chemical kinetics calculations is critically determined by:

(1) chemical kinetics processes with disparate time scales; and

(2) simultaneous chemical species and temperature profiles which possess spatial gradients that are disparate in both their magnitudes and their propagation velocities. The scales of these noncoincident gradients extend from microscopic regions of energy release and species mixing to the macroscopic dimensions of large-scale fluid dynamics structures and system geometries.

Detailed physical and chemical understanding of such systems naturally depends greatly upon one's capacity to resolve satisfactorily the now extended stiffness problem in both space and time.

A second key factor which stimulated the recent burst of developments in ODEs was the appearance of C. W. Gear's implicit multi-step method [3]. The fact that Gear's method managed to yield numerical solutions to stiff ODE systems was, however, not the single most important factor in its success; for other methods such as implicit Runge–Kutta methods were also capable of solving systems of stiff ODEs at the time. The most notable aspect of Gear's contribution was the successful reduction to practice of a systematic control method for the numerical integration of stiff ODE systems in an automatic, hands-off manner. Many technical users discovered quickly that Gear's method really worked for very broad classes of ODEs—with high degrees of reliability and with a minimum of dedicated programming effort by the practitioner. In a word, the key to the success of Gear's method was, and is, the effective maintenance of requisite levels of stability *and* accuracy in the time domains where such control is most needed. *The critical advance of the MFE method for PDEs is the extension of this basic feature to adjust automatically both the space* and *the time stepsizes in those critical regions where such control is most needed, and to allow the finely spaced nodes to move* smoothly *with the critical regions which require their presence.*

We wish to point out here certain significant differences between the continuous node movement properties of the MFE method and the properties of well-known adaptive regridding methods. We note that adaptive regridding methods are of two general types:

(1) such methods as Lagrangian calculations move nodes at mean fluid velocities or at some other characteristic velocity in a fluid; and

(2) alternative Eulerian regridding procedures track a fluid property such as a maximum fluid acceleration or a maximum density or temperature gradient, or a specific ignition temperature range, and insert finely spaced grid nodes into the immediate vicinity of the selected fluid property. (These nodes may be finely gridded but still spatially *fixed*, or they may genuinely move in tracking the fluid property.)

In (1) the number of nodes is usually fixed, and their spatial density changes continuously as the problem evolves. In (2) both the number and the density of nodes may be adjusted according to specific criteria selected by the user.[1] Because most adaptive regridding methods are tied to a single fluid property, these methods are applied very successfully in problems where the most critical fluid behavior has its cause and effect origins associated with that fluid property which is tracked by the regridding method. (Of course, in successful adaptive regridding, dependent variable interpolations must be well defined and performed accurately for maximum effectiveness.) In cases where critical fluid behavior propagates at dissimilar velocities (such as multiple travelling waves, which may additionally reflect back and forth in an uncorrelated manner) over entire problem domains, adaptive regridding methods can encounter serious difficulties. To address such difficulties, more systematic adaptive regridding procedures are presently under development. For example, Oliger's recent work [10] uses local truncation error estimates of finite difference solutions in order to determine those regions where fine grid meshes (with spatially fixed nodes) should be inserted at various times. This approach to adaptive regridding

---

[1] There are, of course, numerous variants of these two basic types of adaptive regridding methods. The reader can consult the following references, among others: McCormack and Paullay [6], Oran *et al.* [7] Oran *et al.* [8] and Dwyer *et al.* [9].

is promising because it has the capacity to anticipate multiple regions of critical fluid behavior over entire problem domains on a more systematic basis than is found in many other adaptive regridding methods which are currently in practice.

The continuous node moving properties of the MFE method thus differ from adaptive regridding methods in several major respects: (1) nodes are not tied to individually tracked fluid properties, (2) nodes are not inserted or removed (although the creation and annihilation of extra MFE nodes might be advantageous in certain instances, and could be smoothly accomplished; see Section 14 of Ref. [2] for some discussion of this possibility), and (3) node coordinates and node amplitudes are calculated simultaneously at each time step from a system of ordinary differential equations which are derived so as to minimize the PDE residuals on the entire problem domain. Accordingly, the MFE method tends to be *highly* systematic and anticipatory in locating nodes where they are most needed in order to resolve critically distributed fluid properties with high accuracy over entire problem domains. Lagrangian remapping and other interpolation methods which are required for moving and/or inserting nodes in adaptive regridding methods are eliminated in the MFE method. The practical effects of these MFE node movement properties will be considered in numerous detailed examples in subsequent sections of this article.

Finally, we note that broad families of PDE systems can now be integrated stably and accurately in a semi-automatic format by the MFE method. The mechanics of formulating a user-ready program which automatically reads, compiles, and solves arbitrary sets of PDEs is, of course, considerably more complex than in the case of ODEs because a much broader family of operators appears in PDEs than in ODEs. But these complexities have been largely resolved in the DYLA program, and we shall see by examples of PDE systems in later sections that very effective automatic solution is now available for general scientific use. A notable benefit which is expected from this new capability is the elimination of a great deal of non-essential, dedicated PDE programming which currently prevails in scientific and engineering practice. It is our hope that such an advance in PDE solution methods will increasingly permit valuable human resources to be redirected away from tedious numerical programming chores and into those channels where creative energy can be applied much more profitably to the technological problems which practitioners initally set out to solve.

I. The Moving Finite Element Method

In order for this article to be read independently, we present in this section a brief sketch of the development of the MFE equations. Derivations which are more detailed in their discussion of rigorous mathematical foundations can be developed from relevant literature sources [1, 2, 11, 12].

We consider first the scalar evolution equation,

$$\dot{u} = L(u), \qquad t \geqslant 0, \tag{1}$$

where $u(t)$, for each time t, is an element of the Hilbert space $\mathscr{H} = \mathscr{L}^2[0, 1]$; and $L(u)$ is some nonlinear partial differential operator. In the usual finite element approach, one looks for an approximate solution v to Eq. (1) which is restricted to lie in a linear finite-dimensional subspace $\mathscr{M} \subset \mathscr{H}$ with basis functions α_j, $j = 1,..., n$. For example, one can take $\mathscr{M}$ to be the space of continuous piecewise linear functions with given *fixed* nodal locations s_j and variable nodal amplitudes a_j. The α_j are then the familiar triangular basis functions. We write

$$v(t) = \sum_{j=1}^{n} a_j(t)\alpha_j. \tag{2}$$

The coefficients a_j are determined from the requirement that $\dot{v} - L(v)$ be orthogonal to $\mathscr{M}$, resulting in the following system of ordinary differential equations:

$$\sum_j (\alpha_i, \alpha_j)\dot{a}_j = (\alpha_i, L(v)), \qquad i = 1,..., n. \tag{3}$$

The generalization that the MFE method provides is to allow nonlinear spaces $\mathscr{M}'$ of approximation functions. For example, $\mathscr{M}'$ may now be taken to be the space of continuous piecewise linear functions with both the nodal amplitudes a_j and the nodal positions s_j being variable. (We will see later that, when one forces the MFE nodal positions to remain fixed, the MFE method reduces to the classical finite element method.) Thus in the MFE approach, the approximate solution,

$$v(t) = v(a_1(t),..., a_n(t), s_1(t),..., s_n(t)), \tag{4}$$

is now restricted to lie in the nonlinear manifold $\mathscr{M}' \subset \mathscr{H}$ which is parametrized by the $2n$ parameters $a_1,..., a_n$, $s_1,..., s_n$, with $|a_i| < \infty$ and $s_1 < s_2 < \cdots < s_n$. Application of the chain rule leads to the following expression for $\dot{v}$ (with the argument t suppressed):

$$\dot{v} = \sum_j \dot{a}_j\alpha_j + \dot{s}_j\beta_j, \tag{5}$$

where the functions $\alpha_j(x)$ and $\beta_j(x)$ are defined by

$$\alpha_j = \frac{\partial v}{\partial a_j}, \tag{6a}$$

$$\beta_j = \frac{\partial v}{\partial s_j}. \tag{6b}$$

That is, $\dot{v}$ is restricted to lie in the tangent space $\mathscr{T}_v$ to $\mathscr{M}'$, at v, which is spanned by the basis functions $\alpha_1, \alpha_2,..., \alpha_n, \beta_1, \beta_2,..., \beta_n$ [1, 2].

In a manner similar to the usual finite element method, one now derives a system of ODEs for a_j *and* s_j by requiring that the residual function $R \equiv \dot{v} - L(v)$ be perpendicular to $\mathscr{E}_v$, i.e., be perpendicular to all the basis functions. This is easily seen to be equivalent to the requirement that the $\mathscr{L}^2$ norm of R be minimized with respect to variations of all $\dot{a}_i$ and $\dot{s}_i$. We see that

$$\begin{aligned}(R, R) &= (\dot{v}, \dot{v}) - 2(\dot{v}, L(v)) + (L(v), L(v)) \\ &= \sum_{j,k} ((\alpha_j, \alpha_k)\dot{a}_j\dot{a}_k + 2(\alpha_j, \beta_k)\dot{a}_j\dot{s}_k + (\beta_j, \beta_k)\dot{s}_j\dot{s}_k) \\ &\quad - 2\sum_j ((\alpha_j, L(v))\dot{a}_j + (\beta_j, L(v))\dot{s}_j) + (L(v), L(v)). \end{aligned} \tag{7}$$

By differentiating this expression with respect to $\dot{a}_i$ and $\dot{s}_i$, we get the equations $(\partial/\partial\dot{a}_i)(R, R) = 2(\alpha_i, R) = 0$, $(\partial/\partial\dot{s}_i)(R, R) = 2(\beta_i, R) = 0$, $i = 1,..., n$. In other words, we have the $2n$ ordinary differential equations for $\dot{a}_i(t)$ and $\dot{s}_i(t)$

$$\sum_j (\alpha_i, \alpha_j)\dot{a}_j + \sum_j (\alpha_i, \beta_j)\dot{s}_j - (\alpha_i, L(v)) = 0, \tag{8a}$$

$$\sum_j (\beta_i, \alpha_j)\dot{a}_j + \sum_j (\beta_i, \beta_j)\dot{s}_j - (\beta_i, L(v)) = 0, \qquad i = 1,..., n. \tag{8b}$$

So far, we have made reference to the possibility of rather general nonlinear spaces $\mathscr{M}'$ of approximation functions; however, for the remainder of this article we refer only to the space of continuous piecewise linear approximation functions with variable nodal amplitudes and positions.[2] The choice of approximation space yields the basis functions $\alpha_j(x)$ and $\beta_j(x)$. The function $\alpha_j(x)$ is that continuous piecewise linear function which takes the value 1 at the node s_j and the value 0 at all the other nodes, i.e., α_j is the usual triangular basis function

$$\begin{aligned}\alpha_j(x) &= (x - s_{j-1})/\Delta s_j, && s_{j-1} \leqslant x < s_j, \\ &= 1 - (x - s_j)/\Delta s_{j+1}, && s_j \leqslant x < s_{j+1}, \\ &= 0, && \text{elsewhere,}\end{aligned} \tag{9}$$

where $\Delta s_j \equiv s_j - s_{j-1}$ is the width of the jth cell $[s_{j-1}, s_j]$ to the left of the jth node.

To derive the basis function β_j in Eq. (6b) we notice that, on the jth cell $[s_{j-1}, s_j]$, the piecewise linear function v can be written as

$$v = a_{j-1} + m_j(x - s_{j-1}), \tag{10}$$

where $m_j \equiv (a_j - a_{j-1})/(s_j - s_{j-1})$ is the slope of v in the jth cell. It follows that, for $x \in [s_{j-1}, s_j)$, $(\partial v/\partial s_j)(x) = -m_j\alpha_j(x)$. Likewise for $x \in (s_j, s_{j+1}]$, $(\partial v/\partial s_j)(x) =$

---

[2] The generalization to higher order moving finite element spaces such as quadratic or cubic splines is easily accomplished [2]. (The general MFE method even allows such highly unusual nonlinear moving finite elements as the "moving vorticity blobs" of Doss and Miller [13].) However, we will see in the present article that piecewise linear approximating functions are surprisingly potent in practical applications of the MFE method, in addition to their simplicity for illustrating the basic features of the MFE method.

$-m_{j+1}\alpha_j(x)$. Therefore, $\partial v/\partial s_j$ has a simple discontinuity at s_j because $\partial v/\partial s_j$ takes the values $-m_j$ and $-m_{j+1}$ at $s_j - 0$ and $s_j + 0$, respectively. To summarize, $\beta_j(x)$ is given by (also see Fig. 1):

$$\begin{aligned} \beta_j(x) &= -m_j\alpha_j(x), && s_{j-1} \leqslant x < s_j, \\ &= -m_{j+1}\alpha_j(x), && s_j < x \leqslant s_{j+1}, \\ &= 0, && \text{elsewhere.} \end{aligned} \tag{11}$$

Because each basis function has support in only two cells, the contribution to the summations in the canonical Eqs. (8a) and (8b) come only from those terms with indices $j = i - 1$, i, and $i + 1$. To write Eqs. (8) in a more compact vector form, the

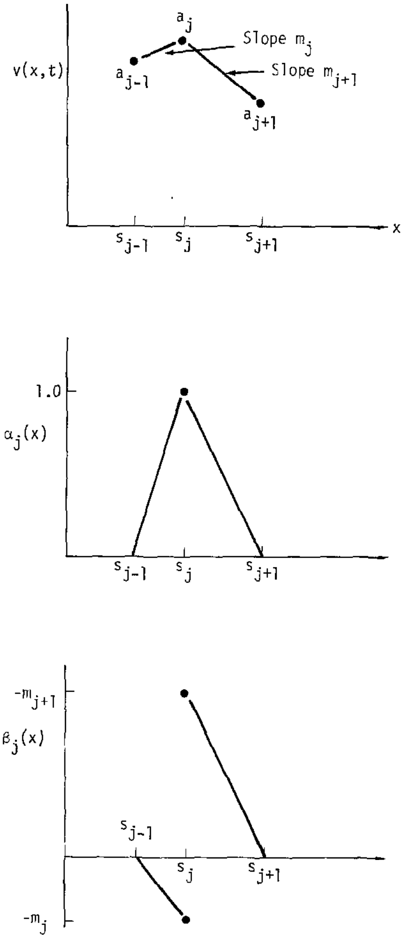


FIG. 1. Graph of $v(x, t)$ and the MFE basis functions $\alpha_j(x)$ and $\beta_j(x)$.

variables $a_1, \ldots, a_n$, $s_1, \ldots, s_n$ can be reordered, and the column vector $y(t)$ of the unknown parameters for the approximant $v(x, t)$ is given by

$$y \equiv (a_1, s_1, \ldots, a_i, s_i, \ldots, a_n, s_n)^T. \tag{12}$$

Equations (8) then takes the form

$$A(y)\,\dot{y} = g(y), \tag{13}$$

where A is an $n \times n$ block-tridiagonal matrix, which we write for $n = 5$ as,

$$A = \begin{pmatrix} A_1 & B_1 & 0 & 0 & 0 \\ C_2 & A_2 & B_2 & 0 & 0 \\ 0 & C_3 & A_3 & B_3 & 0 \\ 0 & 0 & C_4 & A_4 & B_4 \\ 0 & 0 & 0 & C_5 & A_5 \end{pmatrix}. \tag{14}$$

The ith lower block C_i, diagonal block A_i, and upper block B_i are the 2×2 matrices:

$$C_i = \begin{pmatrix} (\alpha_i, \alpha_{i-1}) & (\alpha_i, \beta_{i-1}) \\ (\beta_i, \alpha_{i-1}) & (\beta_i, \beta_{i-1}) \end{pmatrix}, \qquad i = 2, \ldots, n, \tag{15a}$$

$$A_i = \begin{pmatrix} (\alpha_i, \alpha_i) & (\alpha_i, \beta_i) \\ (\beta_i, \alpha_i) & (\beta_i, \beta_i) \end{pmatrix}, \qquad i = 1, \ldots, n, \tag{15b}$$

$$B_i = \begin{pmatrix} (\alpha_i, \alpha_{i+1}) & (\alpha_i, \beta_{i+1}) \\ (\beta_i, \alpha_{i+1}) & (\beta_i, \beta_{i+1}) \end{pmatrix}, \qquad i = 1, \ldots, n-1. \tag{15c}$$

Finally, the $(2i-1)$st and the $(2i)$th elements of the vector valued function $g(y)$ are

$$\begin{aligned} g_{2i-1} &= (\alpha_i, L(v)), \\ g_{2i} &= (\beta_i, L(v)), \qquad i = 1, \ldots, n. \end{aligned} \tag{16}$$

Let us now discuss briefly the necessary extensions of the MFE method to solve the PDE (1) when u and L are vector-valued functions, i.e., when Eq. (1) is a system of PDEs.[3] Suppressing all arguments x and t, we write $u = (u^1, u^2, \ldots, u^p)$, $L = (L^1, L^2, \ldots, L^p)$, and $v = (v^1, v^2, \ldots, v^p)$. For the sake of simplicity in equation structure, we require that all components of the approximating function v share one and the same set of nodes. Each component function v^l, $l = 1, \ldots, p$ is a continuous

---

[3] Extensions of the MFE method to systems of PDEs in 1-D has also been made by Djomehri [14] in the code MFE1DS; there has, in fact, been a mutually beneficial sharing of experience in the developments of DYLA and MFE1DS.

piecewise linear function, with nodal amplitudes a_j^l sharing the common nodal positions s_j. As in the scalar case, we then have

$$v^l = \sum_j a_j^l \alpha_j, \tag{17}$$

$$\dot{v}^l = \sum_j \dot{a}_j^l \alpha_j + \dot{s}_j \beta_j^l, \tag{18}$$

where α_j is defined in Eq. (9), and β_j^l is given by

$$\begin{aligned} \beta_j^l(x) &= -m_j^l \alpha_j(x), && s_j \leqslant x < s_j, \\ &= -m_{j+1}^l \alpha_j(x), && s_j < x \leqslant s_{j+1}, \\ &= 0 && \text{elsewhere,} \end{aligned} \tag{19}$$

where m_j^l is the slope of the function v^l on the jth cell,

$$m_j^l = (a_j^l - a_{j-1}^l)/(s_j - s_{j-1}). \tag{20}$$

As in the scalar case, a system of ordinary differential equations of $\dot{a}_j^l$ and $\dot{s}_j$ is derived from the minimization problem:

$$\begin{aligned} &\text{minimize } |||\dot{v} - L(v)|||^2 \\ &\text{with respect to variations of all } \dot{a}_i^l,\ \dot{s}_i, \\ &1 \leqslant i \leqslant n,\ 1 \leqslant l \leqslant p. \end{aligned} \tag{21}$$

The square of the norm $|||\ |||$ is defined as a weighted sum of the square of the $\mathscr{L}^2$ norms of the lth residual $(\dot{v}^l - L^l(v))$, for $l = 1,..., p$, or

$$|||\dot{v} - L(v)|||^2 = \sum_{l=1}^{p} w_l \, \|\dot{v}^l - L^l(v)\|^2 \tag{22}$$

for some suitable constant weights w_l, $l = 1,..., p$.

To write the minimization problem (21) in a more concise vector form, let $y(t)$ be the $n \times (p+1)$-dimensional column vector of the unknown parameters of the approximating function v, i.e.,

$$y \equiv (a_1^1,..., a_1^p, s_1; a_2^1,..., a_2^p, s_2; \cdots ; a_n^1,..., a_n^p, s_n)^T. \tag{23}$$

One can view y as being composed of n segments (corresponding to the number of nodes), in which the ith segment consists of $p+1$ elements which are simply the amplitudes and nodal position at the ith node.

By using these notations and the basic logic which was discussed for the scalar case, Eqs. (18), (21), and (22) lead to the following system of ODEs (written in the matrix form):

$$\mathscr{A}(y)\,\dot{y} = \mathscr{g}(y), \tag{24}$$

where $\mathscr{A}$ is an $n \times n$ block tridiagonal matrix whose ith lower, diagonal, and upper blocks are denoted by C_i, A_i, and B_i, respectively. Defining the $p+1$ by $p+1$ matrix $D_{i,j}$ by

$$D_{i,j} = \begin{pmatrix} (\alpha_i, \alpha_j) & & & \bigcirc & (\alpha_i, \beta_j^1) \\ & (\alpha_i, \alpha_j) & & & (\alpha_i, \beta_j^2) \\ \bigcirc & & \ddots & (\alpha_i, \alpha_j) & (\alpha_i, \beta_j^p) \\ w_1(\beta_i^1, \alpha_j) & w_2(\beta_i^2, \alpha_j) & \cdots & w_p(\beta_i^p, \alpha_j) & \sum_{l=1}^{p} w_l(\beta_i^l, \beta_j^l) \end{pmatrix}, \tag{25}$$

the blocks C_i, A_i, and B_i are then given by

$$C_i = D_{i,i-1}, \qquad A_i = D_{i,i}, \qquad \text{and} \qquad B_i = D_{i,i+1}. \tag{26}$$

The ith segment of the right-hand side $\mathscr{g}(y)$ consists of the elements

$$\left((\alpha_i, L^1(v)), \ldots, (\alpha_i, L^p(v)), \sum_{l=1}^{p} w_l(\beta_i^l, L^l(v)) \right). \tag{27}$$

Remarks

- From Eqs. (26) and expression (27) (as well as Eq. (31) which appears below) we see that the weights w_l, for $l = 1, \ldots, p$, appear explicitly only in the $\dot{s}$ equations in (24), and they accordingly influence only the node motion. For example, giving the lth component v^l a very large weight, while giving only very small weights to the remaining components, has the effect of forcing the nodes to "follow" only the lth component.
- As was mentioned earlier, all components of the piecewise linear vector valued function v are required to share the same nodal positions. One could have chosen instead to associate with each component function v^l a different set of nodes $s_1^l, \ldots, s_n^l$ which move independently for the different components. On one hand, the resulting approximation space $\mathscr{M}'$ could possibly prove to be useful in solving multiple reactive fluid components which generate fronts of significantly different velocities. On the other hand, the resulting equation structure would be highly complicated even in one dimension and could prove to be practically unmanageable in extensions to higher dimensions. (This, however, could be the subject of future investigation.)
- Our choice of piecewise linear approximations complicates the evaluation of those inner products which involve second-order differential operators. In Eq. (1), if L is simply the second-order operator $\partial^2/\partial x^2$, then $L(v)$ is a sum of delta functions with weights $(m_{i+1} - m_i)$ at the ith node. Because delta functions do not belong to the $\mathscr{L}^2$ space, the minimizing problem does not directly make sense for the piecewise linear approximating functions. Instead, this minimizing problem must be interpreted in the usual sense of "mollification." That is, the manifold $\mathscr{M}'$ is replaced by the manifold $\mathscr{M}'_\delta$ of "smoothed-off" or "δ-mollified" piecewise linear functions with the same nodal

amplitudes a_i and nodal positions s_i. Equation (24) is then interpreted to be the limiting equation as $\delta \to 0$. In this sense, the analysis in Ref. [1] then shows that the inner products (β_i, v_{xx}) should be interpreted as if β_i assumes its mean value of $-(m_i + m_{i+1})/2$ at s_i. Because v_{xx} has a delta function at s_i of weight $m_{i+1} - m_i$, the value of (β_i, v_{xx}) is found to be

$$(\beta_i, v_{xx}) = -(m_{i+1} - m_i)(m_{i+1} + m_i)/2. \tag{28}$$

Similarly, since α_i takes on the value 1 at s_i, the same analysis gives

$$(\alpha_i, v_{xx}) = m_{i+1} - m_i, \tag{29}$$

which is the same result that one finds when the left-hand side is interpreted in the usual sense of distributions; $(\alpha_i, v_{xx}) = -((\alpha_i)_x, v_x)$. The Appendix includes a list of formulas for the inner products which are used in the test examples of this article.

We conclude this section with a brief discussion of the regularization of the ODEs which are used in the MFE method. In certain instances, the mass matrix $\mathscr{A}$ in Eq. (24) may become singular. One can show that this occurs if, and only if, at some one node all of the component functions v_l have a straight portion to their graph (i.e., if for some i we have $m_i^l = m_{i+1}^l$ for all $l = 1, \dots, p$). In order to overcome this potential difficulty, Miller [1, 2] introduced regularization terms into the minimization problem (21). This has the effect of not only keeping the resulting mass matrix positive definite but also penalizing the *relative* motion between nodes. The regularization terms also provide a practical means of maintaining the numerical stiffness properties of the ODE system at very manageable levels—even when *extremely* non-uniform mesh configurations are encountered. The new minimization problem which replaces the previous one now reads

$$\text{minimize} \left\{ \sum_{l=1}^{p} w_l \|\dot{v}^l - L^l(v)\|^2 + \sum\nolimits_{j=2}^{n} (\varepsilon_j \Delta \dot{s}_j - S_j)^2 \right\} \tag{30}$$

with respect to variations of all $\dot{a}_i^l$, $\dot{s}_i$, $1 \leqslant i \leqslant n$, $1 \leqslant l \leqslant p$.

In this new minimization problem ε_j and S_j are internodal control functions on the jth cell. One usually chooses ε_j and S_j to be positive functions of the jth cell width Δs_j which become infinitely large as Δs_j approaches a specified minimum value. (Note that ε_j and S_j can also be made to depend upon quantities other than Δs_j, which lends flexibility to the use of the regularization terms as multi-purpose penalty functions; also see the discussion in Section III.)

The new regularized system of canonical equations is given in explicit form by:

$$\sum_{j=i-1}^{i+1} (\alpha_i, \alpha_j)\dot{a}_j^l + \sum_{j=i-1}^{i+1} (\alpha_i, \beta_j^l)\dot{s}_j = (\alpha_i, L^l(v)), \qquad l = 1, \dots, p, \tag{31a}$$

$$\sum_{j=i-1}^{i+1} \sum_{l=1}^{p} w_l(\beta_i^l, \alpha_j)\dot{a}_j^l + \sum_{j=i-1}^{i+1} \left(\sum_{l=1}^{p} w_l(\beta_i^l, \beta_j^l) \right) \dot{s}_j$$
$$- \varepsilon_i^2 \dot{s}_{i-1} + (\varepsilon_i^2 + \varepsilon_{i+1}^2)\dot{s}_i - \varepsilon_{i+1}^2 \dot{s}_{i+1}$$
$$= \sum_{l=1}^{p} w_l(\beta_i^l, L^l(v)) + \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}, \qquad i = 1,\ldots, n. \tag{31b}$$

A simple relationship can be noted between this regularized system and the unregularized system of Eq. (24). If we write the regularized equations (31) in the vector form

$$\tilde{\mathscr{A}}\dot{y} = \tilde{g}, \tag{32}$$

then $\tilde{\mathscr{A}}$ is obtained from $\mathscr{A}$ by simply adding to the blocks C_i, A_i, and B_i at the locations $(p+1,\ p+1)$ the quantities $-\varepsilon_i^2$, $\varepsilon_i^2 + \varepsilon_{i+1}^2$, and $-\varepsilon_{i+1}^2$, respectively. The ith segment of $\tilde{g}$ consists of the elements (compare with the expression given in (27)) $((\alpha_i, L^1(v)),\ldots, (\alpha_i, L^p(v)), \sum_{l=1}^{p} w_l(\beta_i^l, L^l(v)) + \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1})$.

Further Remarks

- Miller [1] chose not to penalize absolute nodal velocities, which would be accomplished by adding regularization terms such as $\sum_j (\varepsilon \dot{s}_j)^2$ for some small constant ε. Such penalties applied to a degenerate node (that is, a node at which the parameterization of the manifold $\mathscr{M}'$ becomes degenerate at which time the graph of v is straight) would cause very non-smooth movement of that node. Non-smooth node movements would not promote the attainment of large Δt's in the time integration of the MFE equations.

- Since the added regularization terms in Eq. (30) are chosen not to depend explicitly on $\dot{a}_j^l$, $1 \leqslant l \leqslant p$, $1 \leqslant j \leqslant n$, only the $\dot{s}_i$ equations (31b) are altered in the regularized system of canonical equations. The equations (31a) are untouched by the regularization process; hence, for each component function v^l, we have that $\dot{v}^l - L^l(v)$ is orthogonal to all linear combinations of the α_i; i.e., to all continuous piecewise linear test functions.

- The regularizing functions ε and S which were used in most of this work are given by

$$\varepsilon(\Delta s) = \frac{k_2}{\Delta s - k_1} + k_3, \tag{33a}$$

$$S(\Delta s) = \frac{k_4}{\Delta s - k_1}. \tag{33b}$$

Since $\varepsilon(\Delta s)$ and $S(\Delta s)$ become infinite as the cell width Δs approaches k_1, one sees that k_1 gives a specified lower limit on the allowed distance of closest approach for neighboring nodes. The parameters k_2, k_3, and k_4 are usually small constants. (It is interesting to note further that, by simply assigning large values for k_3 or k_2, the MFE nodes become fixed, thereby yielding the conventional fixed node finite element method.)

- The parameters k_1, k_2, k_3, and k_4 provide a practical means of maintaining the numerical stiffness of the ODE system at levels which are readily integrated with implicit, stiffly stable methods. In any technical application which requires that small scales (e.g., those of viscous or other microscale transport processes, shocks, or boundary layer effects) be resolved simultaneously with much larger scales in the physical system, k_1 is simply assigned a value which would resolve accurately the critical physical processes. The values of k_2, k_3, and k_4, which prevent the system from becoming impractically stiff and which yield efficient numerical solutions, are then readily determined. Results in Section II will demonstrate the effectiveness of this regularization feature in the solution of sample problems with unprecedented levels of mesh non-uniformities. We are also finding (in current research which will be reported at a later time) that new generations of regularization terms can be even more effective than our first-generation regularization terms in the solution of difficult PDE problems.
- The system of ODEs in Eqs. (31) is solved by the implicit, stiffly stable method of Gear. Further details of this implementation in the DYLA program are discussed in the Appendix.

II. SAMPLE PROBLEMS AND RESULTS

The automatic PDE solution program DYLA has been used to solve five sample problems with the MFE method. Sample problems have been selected in order to illustrate the ability of the MFE method to resolve the most troublesome numerical dilemmas which arise in the application of PDE solution methods; these dilemmas include the following effects:

(1) Distortion of large gradients in regions of propagating wave-fronts (or shocks) by numerical diffusion and/or Gibbs phenomena in Eulerian calculations;

(2) Zone tangling in colliding wavefronts or aliasing of fluid components when repeated remapping interpolations are applied to multi-fluid or multiple species distributions in Lagrangian calculations;

(3) Excessive numbers of spatial zones and severely constrained time steps when stiff chemical kinetics and transport mechanisms are strongly coupled in reactive flow systems; and

(4) The need for simultaneous and precise resolution of non-coincident gradients. (For example, flame fronts are commonly generated in which multiple species and energy may propagate and diffuse at individually dissimilar velocities relative to the mean fluid flow. These same dilemmas occur in many other physical

systems such as fully and partially ionized plasmas, chemical lasers, and the general atmospheric circulation of Earth and other planetary and astrophysical systems.)

Our purpose in this presentation of results is to show the reader problem runs which represent our early attempts with regard to the selection of numbers of nodes, error constraints, minimum node separations, and other problem parameter values in the use of this general purpose PDE solution code. Accordingly, these results were obtained without laborious fine-tuning. Obviously, a great deal of optimization can yet be accomplished, and in some examples we indicate alternative selections of problem parameters which are more nearly optimal than the originally cited test case values. In view of these factors (in addition to the fact that the DYLA program itself has not yet been optimized to a production code version) computational timing comparisons between the MFE and other PDE solution methods are probably somewhat premature. Another difficulty also impedes meaningful timing comparisons at the present time: many of the examples which are considered in this section have simply not been solved by other PDE methods with comparable levels of resolution and accuracy; and meaningful comparisons can be made, of course, only when identical problem requirements have been fulfilled. In other examples, where comparable solutions have been achieved by alternative methods, the literature has often not included sufficient information to determine the total CPU (central processing unit) time on an equivalent CDC 7600 basis. (Alternatively, operations counts are often not given in sufficient detail, with due allowances made for individual program and computer structures, to provide quantitative timing comparisons.) In the following discussions of MFE results we indicate the total CPU times on the Lawrence Berkeley Laboratory CDC 7600 computer in order to provide readers with some idea of easily attainable MFE computing times. However, we can see already that, in many cases, optimal MFE computing times will be significantly smaller than the presently cited values.

Results of the five sample problems are now discussed on an individual problem basis.

Problem 1. *Propagation of a single square wave.*

Consider the test problem which has been previously discussed by Boris and Book [15]:

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho}{\partial x}, \qquad x \in (0, 100). \tag{34}$$

The gradients at the front and rear edges of the square wave are exceedingly steep (10^3 in our example). The exact solution of Eq. (34) requires that this wave propagates (from left to right) at a velocity of 1.0, with no distortion of the initial profile. Initial conditions are

$$\begin{aligned} \rho(x,0) &= 0.5, & 0 \leqslant x \leqslant 0.002 \\ &= \text{linear}, & 0.002 \leqslant x \leqslant 0.003 \\ &= 2.0, & 0.003 \leqslant x \leqslant 20.003 \\ &= \text{linear}, & 20.003 \leqslant x \leqslant 20.004 \\ &= 0.5, & 20.004 \leqslant x \leqslant 100, \end{aligned} \tag{35}$$

and 11 nodes are used (see Fig. 2). Boundary conditions are

$$\rho(0,t) = \rho(100,t) = 0.5, \qquad t \geqslant 0. \tag{36}$$

The inconsistent boundary condition at $x = 100$ and $t > 79.996$, is applied (purposely) in order to demonstrate the sensitivity of the MFE method to the presence of an ill-posed boundary condition.

Results. A propagating static wave form is, in a sense, trivial for the MFE method. The initial node positions are obviously selected to resolve the square wave profile exactly. With no physical forces acting to alter the wave form at subsequent times, the true solution remains exactly representable as a piecewise linear function, and we would expect the MFE solution to move its nodes in a manner which continues to resolve the initial waveform accurately as it propagates across the interval [0, 100].

Figure 2 indicates that this expectation is fulfilled. A relative error (as defined in Ref. [15]) of approximately 10^{-6} was maintained by the use of only eleven nodes throughout the course of this single square wave problem. Wave distortions such as overshooting, undershooting, or numerical diffusion do not develop because the nodes continually adjust their positions to resolve stably and accurately the largest physical gradients in the problem. Note further that, because nodes are not allowed to cross each other, zone tangling does not occur. Comparisons of these MFE solutions to Lax–Wendroff, leapfrog, donor cell, and flux corrected transport (FCT) solutions with 101 nodes are shown in Fig. 3.

This trial run used node controls which would have permitted a minimum node separation of $k_1 = 10^{-6}$, with $k_2 = 10^{-5}$.[4] Using an initial time step of 10^{-6}, this problem ran to completion in approximately 105 time-step cycles; and approximately 0.4 sec CPU time was required. Note that this problem can also be solved with high accuracy using only the minimal number of nodes which are required to resolve the square wave profile. That is, using six nodes (a total of two at the boundaries and a [illegible] time were required. In these runs, the local truncation error tolerance in the ODE solver was 10^{-3}. Artificial viscosity is not used because this sample problem is well-posed for $0 \leqslant t \leqslant 79.996$ in terms of finite, but large, gradients; and accurate

---

[4] In this and in several subsequent examples, $k_3 = 0$ and $k_4 \leqslant 10^{-6}$. In order to avoid repetition in our discussion, these parameters will be mentioned only when they have magnitudes which are different from the presently cited values.

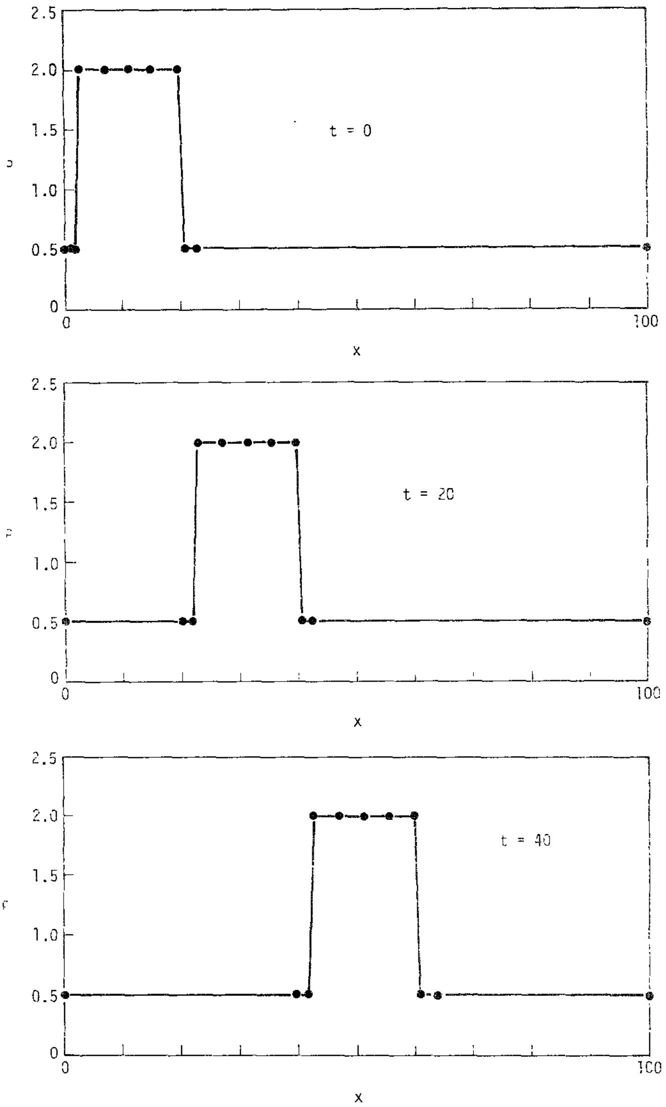


FIG. 2. MFE solution of single square wave propagation with 11 nodes.

solutions are readily achieved by the MFE method in this interval. But for $t > 79.996$, the boundary condition is not consistent with the solutions to the hyperbolic equation (34). That is, at $x = 100$ and for $t > 79.996$, ρ cannot assume the boundary value of 0.5 simultaneously with other characteristic solutions for the on-coming square wave. We find that, at times exceeding 79.996, the square wave has approached the right-hand boundary so closely that: (i) the nodes are compressed to their minimum separation; (ii) numerical integration time steps for the MFE

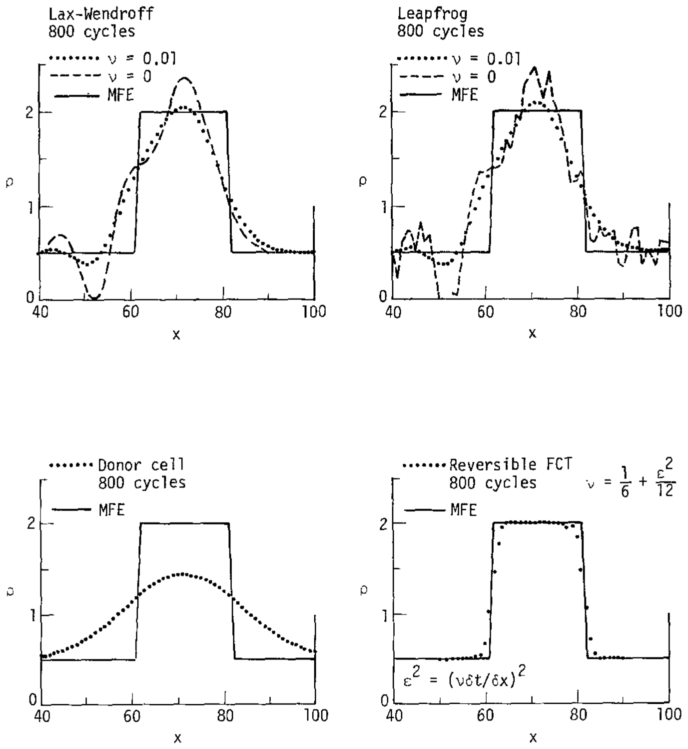


FIG. 3. Comparison of single square wave results from alternative PDE solution methods with 101 nodes.

equations reduce to values of $\leqslant 10^{-15}$; and (iii) integration halts because the ODE solver cannot generate acceptable solutions to the ill-posed problem at the right-hand boundary.

In a word, the MFE method has the capacity to move nodes automatically so as to resolve arbitrarily large gradients, without zone tangling and without wasting nodes in regions of weak gradients. The MFE method also demonstrates a significant sensitivity to ill-posed boundary conditions.

Problem 2. Burger's equation.

This classic test problem is given by the equation

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) + \frac{1}{R}\left(\frac{\partial^2 u}{\partial x^2}\right), \qquad x \in (0, 1). \tag{37}$$

Solutions of Burger's equation develop shocks of width $0(1/R)$, travelling at velocities of $(u_+ + u_-)/2$, where u_+ and u_- are the values of u just ahead of, and just behind, the shock front. To test the applicability of various PDE solution methods, arbitrarily steep fronts can be generated by arbitrarily large values of R. Three test cases are considered:

Case A. $R = 10^3$, *and* 21 *nodes are used for an initial step-like wave.*

Initial conditions are given by

$$\begin{aligned} u(x,0) &= 10.0, && 0 \leqslant x \leqslant 0.100 \\ &= \text{linear}, && 0.100 \leqslant x \leqslant 0.101 \\ &= 0. && 0.101 \leqslant x \leqslant 1 \end{aligned} \tag{38}$$

Boundary conditions are

$$\begin{aligned} u(0,t) &= 10, && t \geqslant 0, \\ u(1,t) &= 0, && t \geqslant 0. \end{aligned} \tag{39}$$

Case B. $R = 10^4$, *and* 21 *nodes are used for an initial sinusoidal wave form.*

Initial and boundary conditions are given by

$$u(x,0) = \sin(2\pi x) + 0.5\sin(\pi x), \qquad x \in [0, 1], \tag{40}$$

and

$$u(0,t) = u(1,t) = 0, \qquad t \geqslant 0. \tag{41}$$

Case C. $R = 10^4$, *and* 31 *nodes are used for an initial unit impulse function.*

Initial conditions are given by

$$\begin{aligned} u(x,0) &= 0, && 0 \leqslant x \leqslant 0.48 \\ &= \text{linear}, && 0.48 \leqslant x \leqslant 0.52 \\ &= 1, && 0.52 \leqslant x \leqslant 1.48 \\ &= \text{linear}, && 1.48 \leqslant x \leqslant 1.52 \\ &= 0, && 1.52 \leqslant x \leqslant 2 \end{aligned} \tag{42}$$

Boundary conditions are

$$u(0,t) = u(2,t) = 0, \qquad t \geqslant 0. \tag{43}$$

Results. Burger's equation describes the propagation of dynamic waveforms, in contrast to the static waveform in Problem 1. In the present examples, infinitely steep gradients are generated as $R \to \infty$, and we expect that the MFE nodes will migrate to those regions where steep shocks are generated in the course of a waveform's evolution. This expectation is verified in the test cases which are discussed immediately below.

Case A. $R = 10^3$, *and* 21 *nodes are used for an initial step-like wave.*

The slope of the initial gradient is 10^3 and the initial node locations are selected so that they cluster about the steep gradient. The speed of propagation is expected to be 5 (the mean value of 10 and zero). Figure 4 shows the following results:

- A majority of the nodes migrate into the shock front as time progresses.
- The shock propagates at the correct speed ($u = 5$) until the effects of the boundary at $x = 1$ are felt.
- At $t = 0.05$, the shock width is 0.002.
- At $t = 0.05$, the shock gradient is $v_x \approx 12{,}000$ at the center of the shock. (The slope varies slightly from point to point within the highly resolved shock front.)
- No wave distortion appears in the MFE solutions. Here we have used a minimum nodal separation of $k_1 = 10^{-4}$, with $k_2 = 10^{-4}$, and a local truncation error tolerance of 10^{-3} in the ODE solver.
- As the shock approaches the right-hand boundary at $t = 0.20$, a steady state boundary layer is formed with a thickness of approximately $1/R$, as would be expected theoretically. The magnified view of this boundary layer in Fig. 4 demonstrates conclusively the powerful resolution and smooth time parametrization of the nodes as the steady state solution is approached.

For the solution conditions which were indicated above and an initial time step of 10^{-3}, we find that approximately 240 time-step cycles and 1.8 sec CPU time are required for the shock to run freely to the right-hand boundary at $x = 1$. An additional 130 time-step cycles were required to run the solution to $t = 50$, which is very much beyond the time at which steady state is established (at $t \approx 0.2$).

The approach to steady state is particularly interesting. The shock front "feels" the right-hand boundary when the several nodes to the right of the shock front are compressed to their minimum allowable separations of 10^{-4} by the wall at $x = 1$. At this time, the shock front is located at $x \approx 0.98$, and the time step begins to decrease from its large free-running values of $\Delta t \approx 0.04$. We note also that, at $t \cong 0.17$, just prior to "feeling" the right-hand boundary, the MFE time step exceeded the Courant–Friederich–Lewy condition by a factor of 2000.

This problem can be solved just as well with any reasonable value for R. For example, with $R = 10^6$, a final stationary boundary layer is established with a width of approximately 10^{-6} at $x \cong 1$. Of course, in this case one would permit a minimum node separation of less than 10^{-6} in order to resolve accurately, without Gibbs' phenomena, the shock width of approximately 10^{-6} which would develop during the transient solution, as well as at the asymptotic boundary layer.

Case B. $R = 10^4$ *and* 21 *nodes are used for an initial sinusoidal waveform.*

This example, shown in Fig. 5, demonstrates vividly the growth of a very steep gradient from an initial gentle gradient in the MFE solution of Burger's equation. The initial node positions were distributed uniformly over the spatial interval [0, 1].

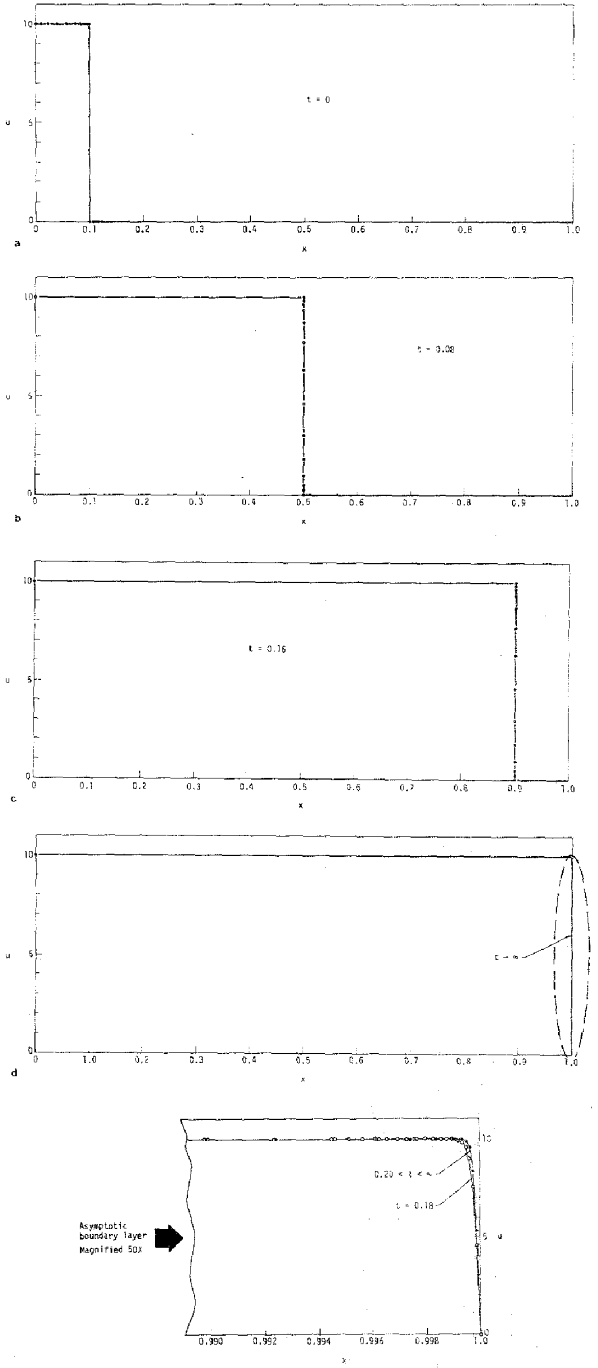


FIG. 4. MFE solution of Burger's equation—step-like wave at $t = 0$, 21 nodes.

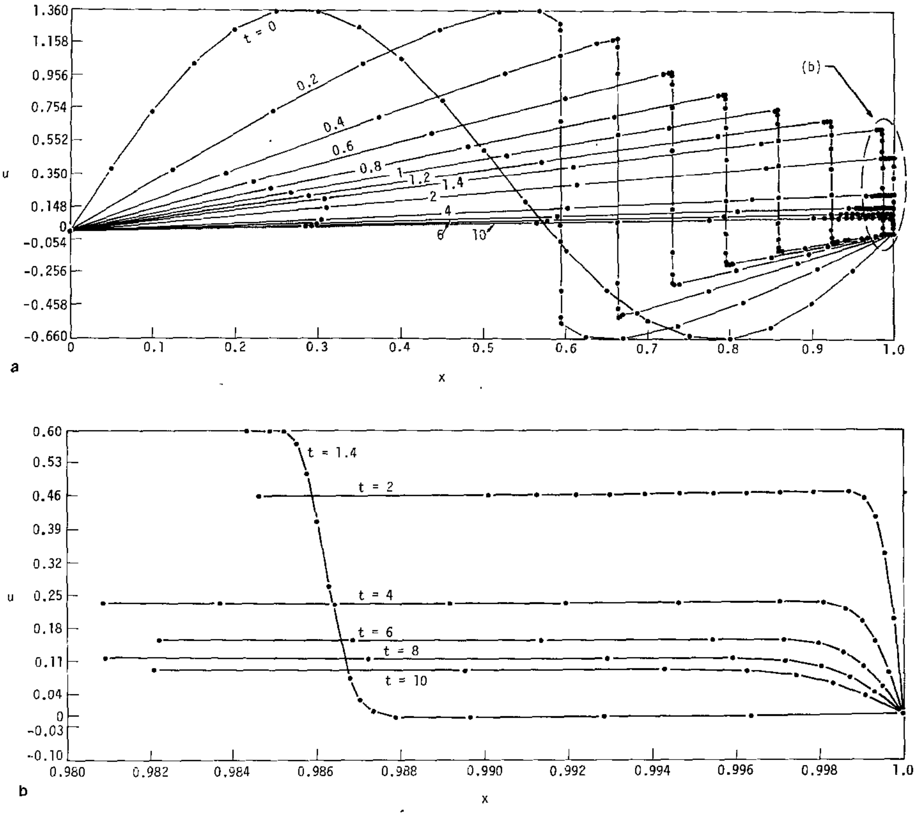


FIG. 5. MFE solution of Burger's equation—sinusoidal waveform at $t = 0$. (a) Transient development of the MFE solution. (b) Detailed view (50X magnification) of formation of asymptotic boundary layer.

Shortly after start-up the nodes migrated very effectively to the regions of sharp curvature in neighborhoods of steepening gradients. Because the Dirichlet boundary values are zero at both boundaries, the wave amplitudes damp toward the asymptotic value of zero at late times. These solutions were obtained with a minimum allowable node separation of $k_1 = 10^{-4}$, with $k_2 = 10^{-3}$, and a local truncation error constraint of 10^{-3} in the ODE integrations. The lower graph in Fig. 5 presents a magnified view (50X horizontal magnification) of the boundary layer evolution at the right-hand boundary.

These results demonstrate once again the capacity of the MFE method to resolve, smoothly and accurately, small scale and large scale physical processes simultaneously. This capacity results directly from dynamic node positioning in those regions where the nodes are most needed in order to minimize the PDE residuals over the entire space-time domain. With an initial time of 10^{-5}, this example is solved completely (from $t = 0$ to asymptotic times) in approximately 100 time-step cycles and 2.4 sec CPU time.

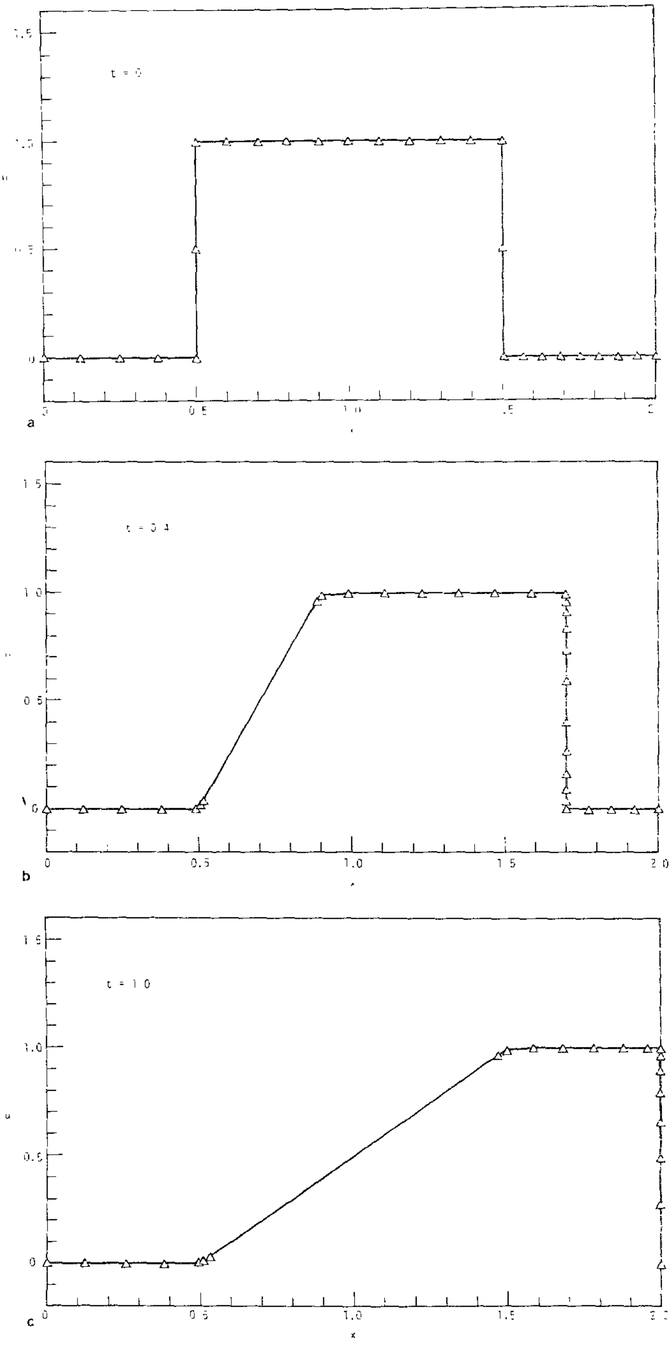


FIG. 6. MFE solution of Burger's equation for an initial unit impulse function. This solution was obtained with 31 MFE nodes, and $1/R = 10^{-4}$.

Case C. *$R = 10^4$ and* 31 *nodes are used for an initial unit impulse function.*

This example, which was abstracted from recent work by Byrne [16] provides an illustration of fixed node, versus moving node, finite element solutions of Burger's equation. The MFE solutions in Fig. 6 propagate smoothly and very accurately (results are reproducible to several significant figures). At times prior to $t = 1$, the slope of the leading wavefront becomes very steep; gradient magnitudes are on the order of 5000. At $t = 1$, the leading wavefront encounters the right boundary. At this time, the nodes compress smoothly to nearly their minimum allowable separations ($k_1 = 5 \times 10^{-5}$, $k_2 = 10^{-3}$) and a boundary layer is formed with a thickness of approximately 10^{-4} at the right boundary (in a manner which is very similar to Case B immediately above). Approximately 5.4 sec CPU time and 392 cycles were

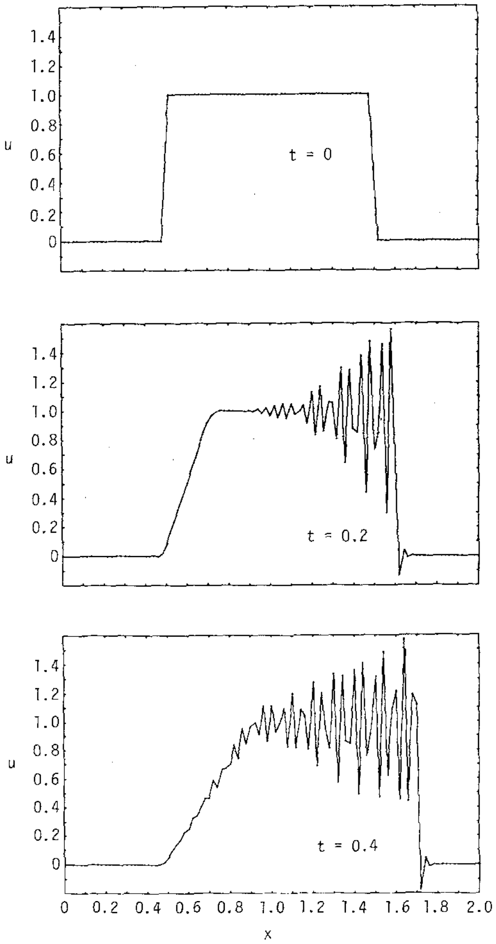


FIG. 7. Fixed node finite element solution of Burger's equation for an initial unit impulse function. The solution was obtained with 101 fixed FE nodes and $1/R = 10^{-4}$.

required to reach $t = 10$ in our initial solution efforts. (This particular example, whose exact solution remains nearly piecewise linear, can also be solved effectively by MFEs with many fewer nodes and many fewer time steps.)

As indicated in Section I, the MFE method reduces to the classic fixed node finite element method whenever the control parameter k_3 is assigned a very large value. Figure 7 presents such a solution for $k_3 = 10^{10}$ and 101 (fixed) nodes. As one expects, Gibbs overshooting and undershooting occurs because the very small scales which are associated with a value of $1/R = 10^{-4}$ in the dispersion term cannot be resolved adequately with 101 fixed nodes. Here we note that, depending upon specific problem requirements, users may or may not require extremely accurate resolution of physical shock front structures. In cases (also see Problem 5 below) where one does wish to resolve shock structures to the same degree as the MFE solutions in Fig. 6, as many

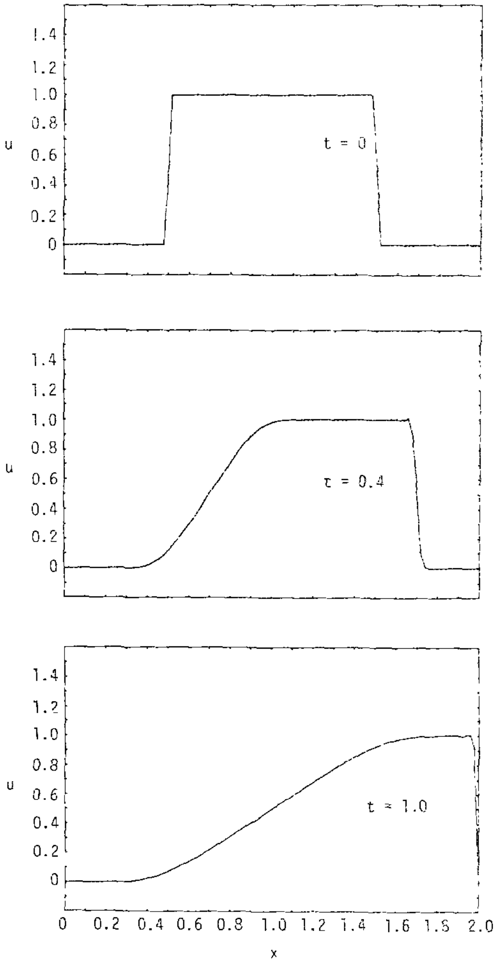


FIG. 8. Fixed node finite element solution of Burger's equation for an initial unit impulse function. This solution was obtained with 101 fixed FE nodes and $1/R = 0.005$.

as 10,000 nodes would be required with several standard fixed node finite element and finite difference solution methods. As an alternative to applying impractically large numbers of nodes, one can increase the value of $1/R$ to 0.005 in order to prevent Gibbs overshooting phenomena in this example with 101 *fixed* nodes. Figure 8 presents fixed node finite element results which were generated by DYLA and by Byrne [16] with 101 nodes and $1/R = 0.005$. We observe in Fig. 8 that the Gibbs oscillations are no longer present; but the wavefront has been smeared to a significant extent.

Problem 3. *The Buckley–Leverett equation* [17].

The equation,

$$\frac{\partial u}{\partial t} + \frac{Q}{\phi}\frac{\partial}{\partial x} f(u) = 0, \tag{44}$$

is used to describe simultaneous one-dimensional flow of two immiscible fluids through a porous medium, neglecting capillary pressure and gravitational forces. In the specific case of oil and water flowing through sand, the dependent variable, $u(x, t)$, describes the water saturation of the sand, and $f(u)$ is a flux function of the flowing stream. The total flow is denoted by the constant Q, and ϕ is the porosity. The ratio Q/ϕ is assigned a valued of unity. We will solve the example which was considered previously by Concus and Proskurowski [17] in which

$$f(u) = \frac{u^2}{u^2 + \alpha(1-u)^2}. \tag{45}$$

The quantity α is the ratio of viscosities, which is assigned a value of 0.5. The solution of Eq. (44) is expected to develop a sharp front for the following initial and boundary conditions:

$$u(x, 0) = \frac{0.1}{0.1 + x}, \quad x \in [0, 1], \quad (\text{or } x \in [0, 2] \text{ in Figs. 10 and 13}) \tag{46}$$

and

$$u(0, t) = 1, \quad u_x(1, t) = 0, \quad t > 0 \ (\text{or } u_x(2, t) = 0 \text{ in Figs. 10 and 13}).$$

The number of nodes used in our MFE solutions ranges from 11 to 31.

Results. As the Buckley–Leverett equation is written in Eq. (44), some fundamental physical transport processes which are described by terms proportional to u_{xx} have been eliminated. As a result, the true solutions of the inviscid hyperbolic Eq. (44) must develop truly infinite gradients in u, and thereafter continue to exist only as weak (nonclassical) solutions, whereas the actual (viscous) physical system would develop only finite, albeit exceedingly large, gradients in u. The weak solutions desired of the inviscid equation are the limits of the viscous solutions as the viscous

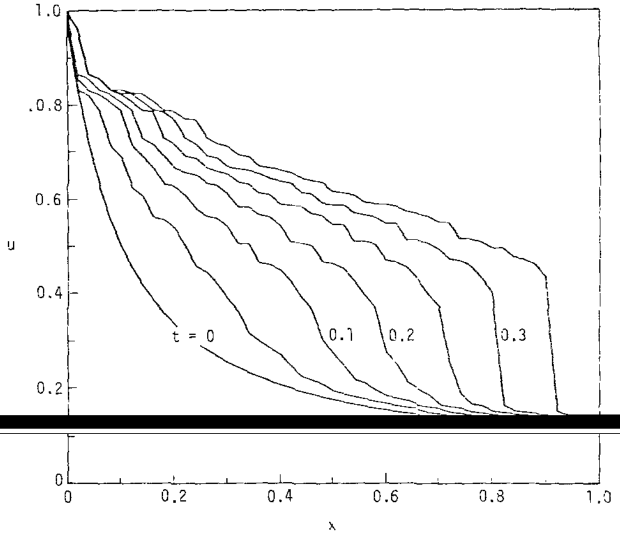


FIG. 9. Solutions of the Buckley–Leverett equation by Glimm's method with 50 fixed nodes.

term in the PDE tends to zero. The inviscid Buckley–Leverett Eq. (44) has been solved previously by Glimm's random choice method [17], and results appear in Fig. 9.[5] In solving the viscous Buckley–Leverett equation, we have used the zero Neumann condition at the right-hand boundary (at $x = 2$ or at $x = 1$) as an approximation to an out-flow boundary condition. Once again we find that the MFE nodes migrate in their usual fashion from initial distributions to those subsequent locations which resolve accurately the steepening saturation fronts. The non-convex nature of $f(u)$ in Eq. (45) gives great difficulties for some finite difference methods such as Lax–Wendroff, but it presents no difficulties in the MFE solution. Likewise, Gibbs phenomena are not encountered in our MFE solutions because in all cases the minimum allowable node separation k_1 was maintained at a value which resolves adequately the smallest scales in the PDE system. Approximately 300 time steps and 9 CPU seconds were required in initial solutions of this sample problem. (These figures can readily be reduced by a factor of three by using fewer Gauss quadrature points and alternative control parameters.)

Figure 10 presents the basic MFE solution of the Buckley–Leverett equation with a value of $1/R = 10^{-4}$ and 31 nodes on the interval $[0, 2]$. The minimum allowable node separation is $k_1 = 5 \times 10^{-5}$; the short- and long-range force parameters, k_2 and k_3, have respective values of 10^{-6} and 0.1 in this test case. At $t = 0.3$, the maximum slope of the saturation front is approximately 200. At $t = 0.5$, the maximum slope is approximately 1000.

---

[5] The "jitter" which appears in Fig. 9 is due to statistical fluctuations in the sampling method and not to the segmental plotting jitter which appears in some of our computer graphics (e.g., Figs. 10, 11, 13, etc.).

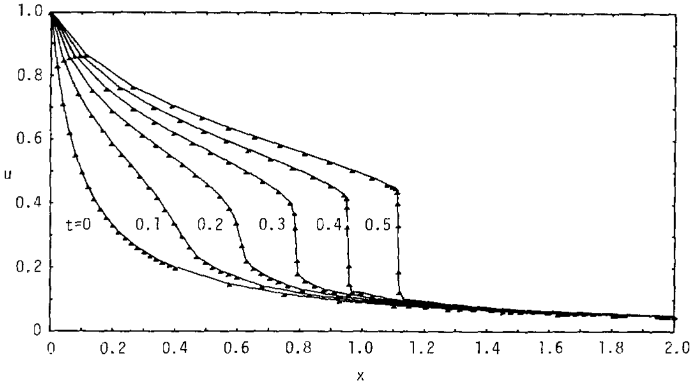


FIG. 10. MFE Solution of the Buckley–Leverett equation with $1/R = 10^{-4}$ and 31 nodes.

At this stage we ask, "Has the inviscid limit been approached with the present solution?" To provide an answer by direct testing, we also obtained (but do not include the graphs here) MFE solutions with 31 nodes for $1/R = 10^{-5}$, $1/R = 10^{-7}$, and $1/R = 10^{-12}$, with the following results:

(1) The MFE results for $1/R = 10^{-5}$ and $k_1 = 5 \times 10^{-6}$ are identical to those which appear in Fig. 10 to within the width of the plotted lines. At $t = 0.3$ the maximum slope of the saturation front is approximately 2000, and at $t = 0.5$ the maximum slope of the front is approximately 10^4.

(2) For $1/R = 10^{-7}$ and $k_1 = 10^{-8}$, the MFE results again overlay exactly the plotted solutions in Fig. 10. At $t = 0.3$, the maximum slope of the saturation front is approximately 10^5, and at $t = 0.5$ the maximum slope of the front is nearly 10^6.

(3) The value of $1/R = 10^{-12}$ is a computational zero for the algorithms which were used in our MFE solution methods on the CDC 7600 computer. For minimum allowable node separations of $k_1 = 10^{-13}$ we find that the numerical integration has become very stiff, and integration terminates at approximately $t = 0.23$ because matrix solutions in the MFE calculations have become ill-conditioned. This corresponds to the theoretical time at which an infinitely steep shock begins to form in the purely hyperbolic equation.

We have thus demonstrated by explicit computations the approach of the viscid Buckley–Leverett equation to the inviscid limit—to the full limits of computational significance of the numerical algorithms and the CDC 7600. This exercise has been repeated with as many as 31 MFE nodes and with as few as 11 MFE nodes with the same results.

A second pertinent question which was addressed is, "At what point does increasing $1/R$ broaden the saturation front artificially (to more than the width of an ink line)?" Figure 11 presents the results for 21 MFE nodes on the interval [0, 1] with $1/R = 10^{-3}$. These results are identical (to within graphing accuracies) to the results for $x \in [0, 1]$ with $1/R = 10^{-4}$ in Fig. 10. We observe that the Neumann

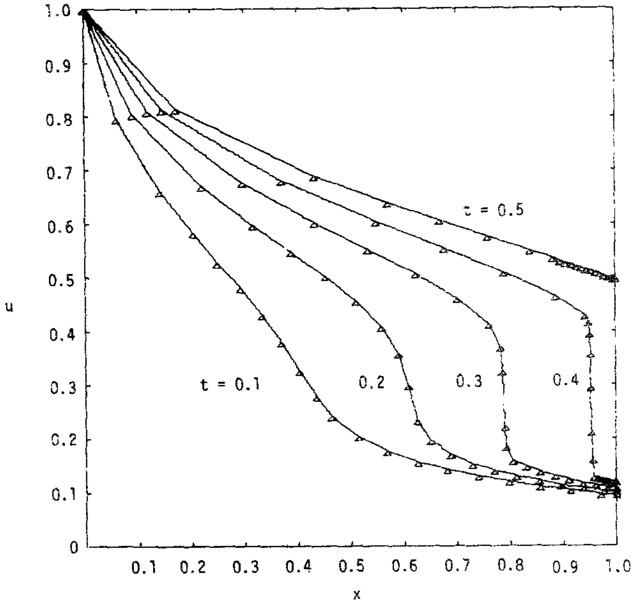


FIG. 11. MFE solution of the Buckley–Leverett equation with $1/R = 10^{-3}$ and 21 nodes.

boundary condition at $x = 1$ does not perturb the solution in Fig. 11 from the corresponding solution at $x = 1$ in Fig. 10. Figure 12 presents similar MFE solutions for $1/R = 0.01$ on the interval $[0, 1]$; here we finally see considerable wavefront broadening. We note that the wavefront broadening which appears in the random choice results (Fig. 9) due to positional uncertainty is intermediate between the wavefront slopes which result for values of $1/R = 0.01$ and $1/R = 0.001$ in the MFE solutions (Figs. 12 and 11, respectively).

Finally, we examine the effects of node density on the quality of MFE solutions. Figure 13 presents results for $1/R = 10^{-3}$ with only 21 nodes on the interval $[0, 2]$. Comparison of these results with corresponding MFE solutions in Figs. 10 and 11, which have greater node densities, indicate the potency of MFE method's approximation properties with limited node densities. The locations of wavefronts are identical, and variances in the approximate solutions at the corners of wavefronts lie equally above and below the true solutions. We expect that the solutions in Figs. 12 and 13 are nearly optimal for such a limited number of nodes.

Problem 4. *Two counter-streaming square waves.*

This example extends Problem 1 so that [square wave]$_1$ propagates from left to right, and [square wave]$_2$ propagates from right to left. The relative velocity of the two waves is 1. We consider two basic cases: in Case A, the two waves are nonreacting and are expected to pass through each other undistorted; in Case B, the two waves act like two distinct fluids which react with and consume each other when the waves intersect.

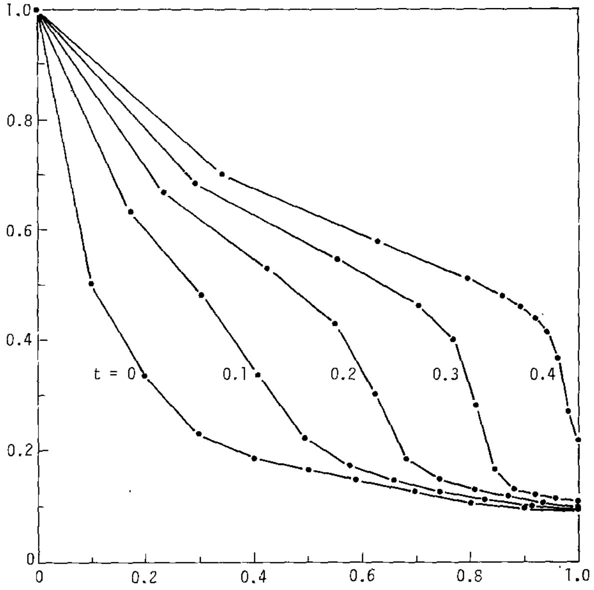


FIG. 12. MFE solution of the Buckley–Leverett equation with $1/R = 10^{-2}$ and 11 nodes.

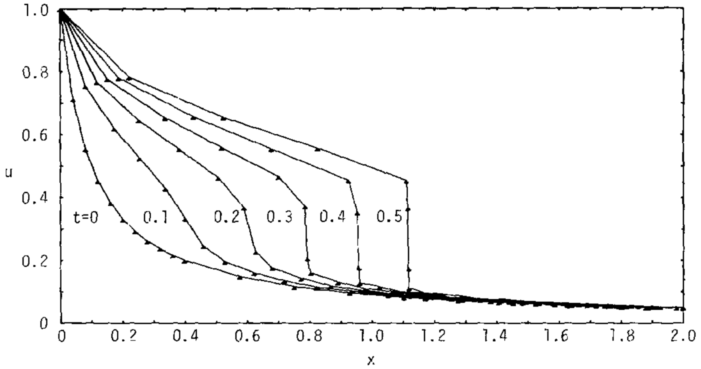


FIG. 13. MFE solution of the Buckley–Leverett equation with $1/R = 10^{-3}$ and 21 nodes.

Case A. *Non-reactive square waves.*

The system of PDEs for the two non-reacting square waves is

$$\begin{aligned} \frac{\partial \rho_1}{\partial t} &= -1/2 \frac{\partial \rho_1}{\partial x}, \\ \frac{\partial \rho_2}{\partial t} &= 1/2 \frac{\partial \rho_2}{\partial x}, \qquad x \in (0, 100). \end{aligned} \tag{47}$$

Initial conditions are

$$\begin{aligned} \rho_1(x,0) &= 0, & 0 \leqslant x \leqslant 0.002 \\ &= \text{linear}, & 0.002 \leqslant x \leqslant 0.003 \\ &= 1, & 0.003 \leqslant x \leqslant 20.003 \\ &= \text{linear}, & 20.003 \leqslant x \leqslant 20.004 \\ &= 0, & 20.004 \leqslant x \leqslant 100, \\ \rho_2(x,0) &= \rho_1((100-x),0), & 0 \leqslant x \leqslant 100, \end{aligned} \tag{48}$$

and 31 nodes are used.

Boundary conditions are

$$\begin{aligned} \rho_1(0,t) &= \rho_1(100,t) = 0, \\ \rho_2(0,t) &= \rho_2(100,t) = 0, \qquad t \geqslant 0. \end{aligned} \tag{49}$$

Case B. *Reactive square waves.*

This test problem has been designed to pose the basic numerical difficulties which are encountered in colliding reactive jets and/or simultaneous mixing and kinetics in reacting eddies. Fluid 1 and fluid 2 interact to consume each other at a rate given by $k_{12}\rho_1\rho_2$. The system of PDEs is

$$\frac{\partial \rho_1}{\partial t} = -1/2\,\frac{\partial \rho_1}{\partial x} - k_{12}\rho_1\rho_2, \tag{50}$$

$$\frac{\partial \rho_2}{\partial t} = 1/2\,\frac{\partial \rho_2}{\partial x} - k_{12}\rho_1\rho_2, \qquad x \in (0, 100), \tag{51}$$

with $k_{12} = 1$ in our calculations. Initial and boundary conditions for ρ_1 and ρ_2 are the same as in the non-reacting case above.

Results. In this problem and in the remaining sample problems our attention focuses on systems of PDEs. The challenges posed by PDE systems are, of course, more meaningful tests of numerical solution methods for ultimate applications to a broad spectrum of practical scientific problems.

Case A. *Non-reactive square waves.*

The most important fundamental features of the MFE method are contained in the results of this sample problem, as seen in Fig. 14. The situation is the following: Initially, the nodes move with the respective freely propagating square waves, very much in a Lagrangian fashion, which maintains exceedingly high resolution of the exact individual waveforms. As the front edges of the waves are about to intersect, the node behavior must change dramatically because the nodes are not permitted to

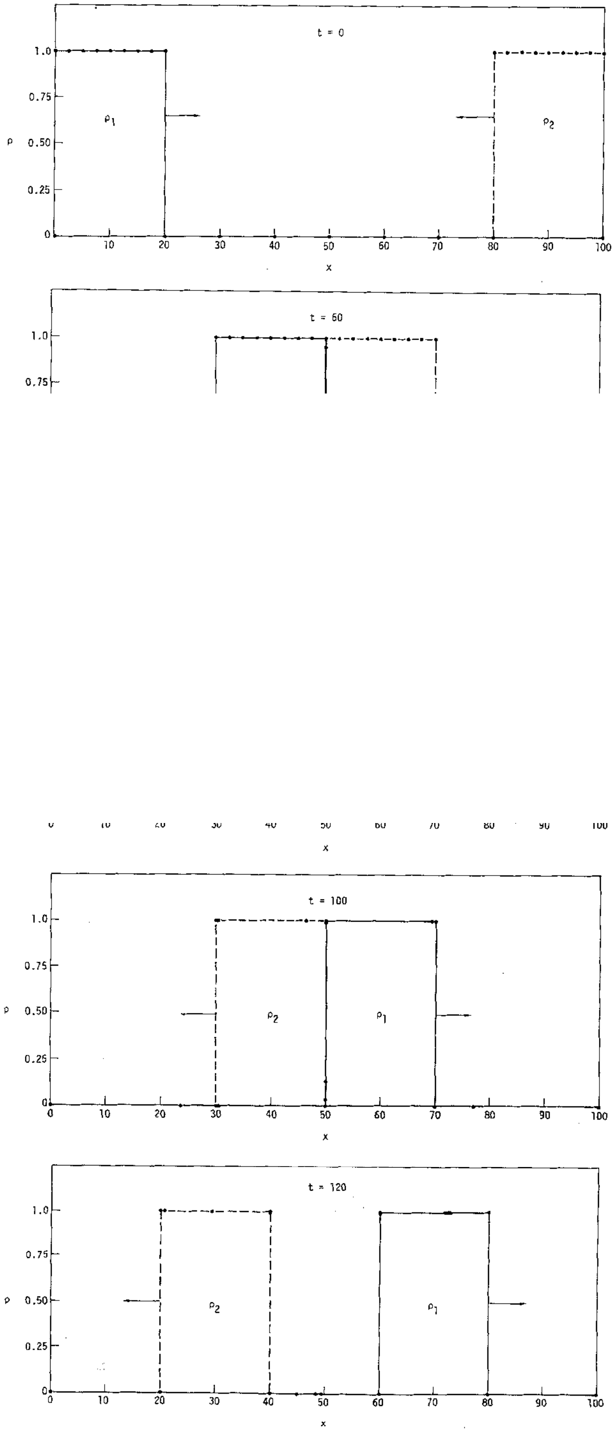


FIG. 14. Two counter-streaming square waves—non reactive case.

interchange relative positions with any neighbors. (The crossing of nodes would be tantamount to zone tangling in conventional methods, which we wish to avoid as we proceed to multi-dimensional MFE solutions in the future.) Instead of crossing each other, the MFE nodes merely approach each other more and more closely—to within the minimum allowable separation k_1, which is specified by the user—so as to resolve simultaneously both of the steep wavefronts. During this stage of wave intersection, the MFE solution method—in the localized region where nodes are colliding—operates very much in an Eulerian mode, in which the wave forms propagate in opposite directions through the nearly stationary (and nearly colliding) grid points. The reader should note that, at time $t = 60$, the two wave fronts (of the exact solution) have intersected by an increment in Δx of only 0.006. At this time the nodes which appear in Fig. 14 at $x \approx 50$ are clustered to within spatial separations of approximately 2×10^{-4}. Therefore, the apparent single node at $x \approx 50$ and $\rho \approx 1$ actually represents six nodes which are clustered to separations which are much less than the thickness of the ink line. Similarly, the apparent single node at $x \approx 50$ and $\rho = 0$ actually represents two MFE nodes which are separated by $\Delta x = 0.006$ in Fig. 14.

At later times, after the waves have separated and are running freely again, the nodes also separate again and resume their travel with the respective waves *at their two distinctly different characteristic velocities*, $v_{1,2} = \pm 1/2$. This MFE solution demonstrates the capacity to resolve smoothly and very accurately—and without specialized programming or specialized coordinate transformations—the simultaneous propagation of multiple fluids at multiple (vector) velocities. Further examples suggest that this distinctive feature of the MFE method will apply effectively to a host of disparate scale processes in multi-dimensional, multicomponent, and perhaps multi-phase, reactive fluid dynamics applications.

The present example of two intersecting non-reactive square waves was solved using 31 nodes and a minimum node separation of $k_1 = 10^{-5}$. In actual practice, the closest distance of approach for adjacent nodes was approximately 5×10^{-5}; and this problem was run in approximately 1000 time-step cycles and 15 CPU seconds. This problem can be run equally well with: (i) 20 nodes in 400 time-step cycles and less than 5 CPU second, or (ii) 10 nodes in less than 200 time-step cycles and 0.7 second CPU time. Since MFE nodes are not allowed to cross, a significant fraction of the computation time is spent in the region of the problem where the nodes are slowing down and reversing direction, as necessary. This is a significant computational cost which would probably be unacceptable if one wished to solve only such linear problems as this non-reactive test case. (Accordingly, the MFE method is applied to this problem for illustrative purposes only; in general practice, linear problems would be solved by more economical methods whenever possible.) However, the MFE method applies equally well to non-linear reactive problems, as will be seen in the next example.

In the present example we observed that, during the early stages when the waves were running freely toward each other, the MFE time-steps exceeded the Courant–Friedrichs–Lewy (CFL) condition by a factor of 3000. While the waves were intersecting at the middle stages of the problem's evolution, the MFE time step exceeded the CFL limit by a factor of 5; and after separation of the waves the MFE time step exceeded the CFL limit by factors of approximately 200. Similarly, in all of the previously discussed sample problems (1–3) the MFE time step exceeded the CFL limit by very large factors (of approximately 10^3). Our numerical results agree with exact solutions to four significant figures, using a local truncation error constraint of 10^{-3} for the ODE solver.

Case B. *Reactive Square Waves.*

We wish to consider now a visually tractable example of two chemically reactive species which contains many of the essential computational features of more general multi-component reactive fluid systems. In this example two chemical constituents are simulated in the collision of two initially square wave parcels of species 1 and 2 which are streaming towards each other (as in Case A above) and reacting when the initial species parcels overlap. As these fluid parcels pass through each other, we expect to see a depletion of the reactive species due to chemical reaction.

Figure 15 indicates that our expectations are again fulfilled. The MFE solutions agree with exact analytic solutions to three significant figures. The performance characteristics of the MFE solution for this reactive multi-component fluid example parallels very closely the non-reactive example in Case A above. By noting in Fig. 15 the very sharp features of the (essentially exact) species distributions, one can readily appreciate the serious consequence that even slight degrees of numerical diffusion or Gibbs phenomena would have inflicted upon the proper treatment of the nonlinear chemical reaction process.

Problem 5. *Dwyer–Sanders Model Flame.*

Two coupled equations for mass density and temperature have been proposed by Dwyer and Sanders [18] to simulate basic features of flame propagation (for example, flame propagation in a solid without gas generation, or flame propagation in a gas without large heat release). The model equations are

$$\frac{\partial \rho}{\partial t} = \frac{\partial^2 \rho}{\partial x^2} - N_{DA}\rho, \tag{52}$$

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + N_{DA}\rho, \qquad x \in (0, 1), \tag{53}$$

where

$$N_{DA} = 3.52 \times 10^6 \exp(-4/T). \tag{54}$$

Initial conditions are

$$\begin{aligned} &\rho(x, 0) = 1, \\ &T(x, 0) = 0.2, \qquad 0 \leqslant x \leqslant 1. \end{aligned} \tag{55}$$

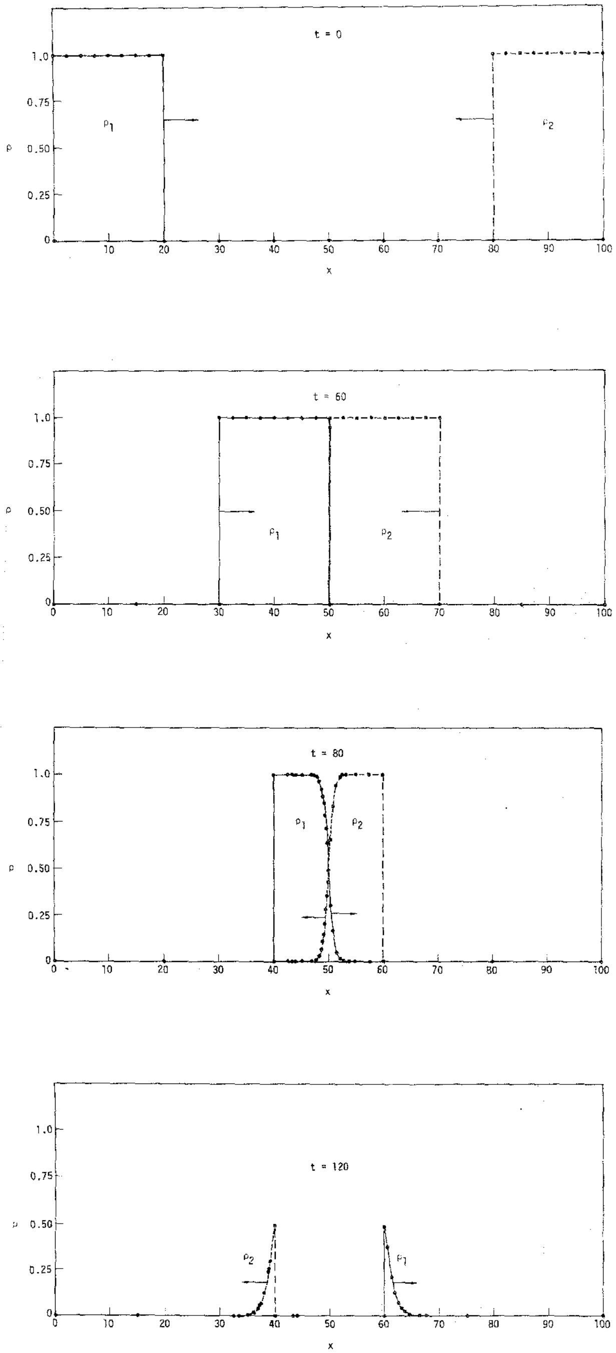


FIG. 15. Two counter-streaming square waves—reactive case.

Boundary conditions are

$$\begin{aligned} &\rho_x(0,t) = T_x(0,t) = 0, \\ \rho_x(1,t) = 0, \qquad &T(1,t) = f(t), \qquad t \geqslant 0; \end{aligned} \tag{56}$$

and the time-dependent forcing function is given by

$$\begin{aligned} f(t) &= 0.2 + t/(2 \times 10^{-4}), \qquad && t \leqslant 2 \times 10^{-4} \\ &= 1.2, && t \geqslant 2 \times 10^{-4}. \end{aligned} \tag{57}$$

The heat source at the wall ($x = 1$) simply generates a flame front which propagates from right to left at a relatively high speed. The flame front is not nearly as steep, however, as the gradients which have been encountered in sample problems discussed above.

Results. This model problem serves as a very useful test example for the computation of certain essential mechanisms which occur in more general physical flame systems. The MFE solutions for this sample problem were obtained with 11, 21, 31, and 51 nodes; essentially identical results were obtained in all runs. Figures 16 and 17 present representative results from the 21 node and the 51 node solutions. Minimum allowable node separations of $k_1 = 10^{-5}$, with $k_2 = 10^{-3}$ to 10^{-2}, $k_4 = 10^{-4}$, an initial time step of 10^{-7}, and a local truncation error of 10^{-3} were used in these runs. By resolving accurately the flame front structures, we note that the fame propagates at nearly (but not precisely) constant speeds in the interval from $t = 0.003$ sec to $t = 0.006$ sec. In this interval, the maximum slope magnitudes are 57 ± 0.5, and the flame front was propagating freely. Flame speeds are derived from the precise positions of the amplitude $T = 1.0$ at many sample points between

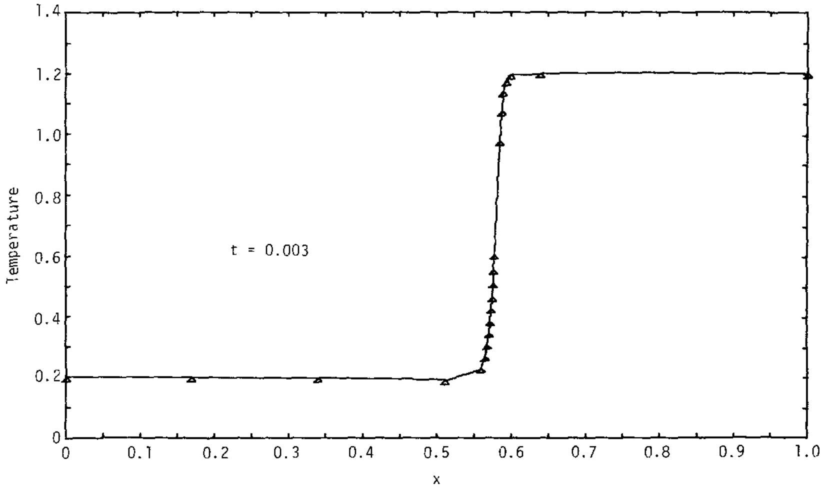
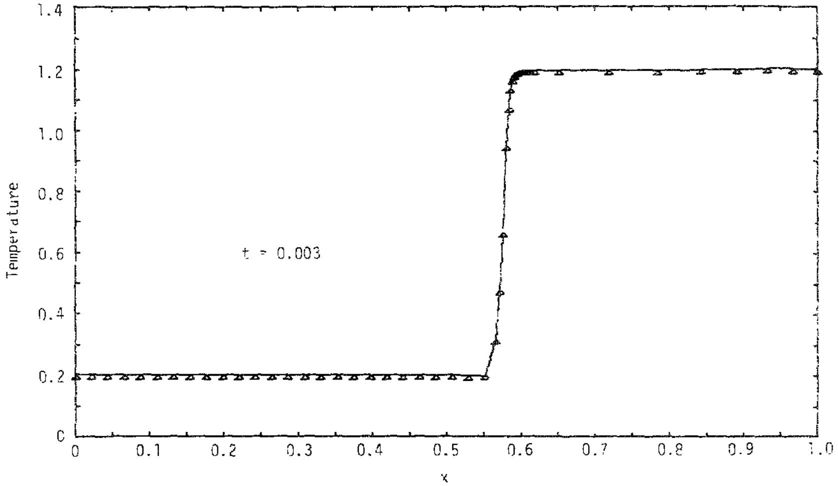


FIG. 16. MFE solution of sample problem 5 with 21 nodes.



[illegible]

the times 0.003 sec and 0.006 sec. (We have verified that essentially identical results are obtained if one uses the amplitude $T = 0.5$ to derive flame speeds in this same interval of time.) The average flame speed in this time interval ($t = 0.003$ sec to 0.006 sec) is 143.7 ± 0.8 cm/sec, with the fluctuations being associated with transient variations in the flame front structure. At times less than 0.003 sec and at times greater than 0.006 sec the propagation of the flame front is affected by the presence of the boundaries at $x = 0$ and $x = 1$, respectively.

For solving this sample problem from $t = 0$ to $t = 0.006$ sec, the 11 node solution required approximately 400 time steps and 4 CPU seconds; the 21 node solution required 600 steps and 13 CPU seconds; the 31 node solution required 540 steps and 16 CPU seconds; and the 51 node solution required 570 steps and 30 CPU seconds. One finds here (as in previous examples) that, with the benefit of additional experience, this sample problem can be solved much more efficiently than was indicated in the previous discussion. Also, one finds in this example—in analogy with the results which appeared for sample problem 3—that the 11 node solution locates the wavefront accurately and has small variances which lie equally above and below the corners of the wavefront in the more precise results from the 21, 31, and 51 node solutions.

This example was also solved by the conventional fixed node finite element method (using piecewise linear elements) with 101 fixed nodes; the flame speeds are 151.2 ± 3.5 cm/sec in the interval $t = 0.003$ sec to $t = 0.006$ sec. Also in this interval the maximum flame front slopes have values of 54.1 ± 2.1. The disparity between the fixed and moving node finite element solutions arises from the less accurate resolution of the flame front (only four cells define the flame front) in the fixed node method, *vis a vis* the more accurate flame front resolution, using many more cells in the flame front, in he MFE method.

In undertaking a comparison between the MFE results and alternative FD methods, we note that Dwyer and Sanders [18] initially reported "best flame speed values" of 136–141 for this problem. In more recent work which uses more nodes (as many as 400) for several finite difference methods and a collocation method, Otey and Dwyer [19] report "best values" of approximately 141. Agreement between the FD and collocation methods and the MFE results are thus converging progressively toward the MFE results. We suspect that any remaining discrepancies could be related to the fact that the non-linear forcing functions in the PDEs are treated much more approximately in the pointwise evaluations of the FD and collocation methods than in the MFE method which puts many nodes in the steep front and which integrates all such terms exactly on entire spatial elements throughout the problem domain.

The Dwyer–Sanders equations (52) and (53) have also been solved for the alternative value of $N_{DA} = 4 \times 10^7 \exp(-4/T)$, using 11, 31 and 51 nodes. In this test case the maximum slopes of the freely propagating flames are 198 ± 6 and the flame speed is approximately 484 cm/sec. The MFE nodes were observed to compress to actual separations of 6×10^{-4} during the solution of this problem, which implies that perhaps 2000, or more, nodes would be required in order to obtain fixed node solutions with comparable resolution. The MFE solutions for this case were run to the time $t = 0.19$ in 350 steps, requiring 4 sec CPU time with 11 nodes; and correspondingly greater times were required with 31 and 51 nodes, as was indicated in the discussion above. An attempt to solve this test case with 101 fixed finite elements verified that satisfactory solutions could not be obtained with such a limited number of fixed nodes. Otey and Dwyer [19] report flame speeds of 474–477, using up to 600 fixed nodes in FD and collocation solutions. We again suspect that the alternative fixed node solutions will converge toward the MFE solutions as the number of fixed nodes is increased beyond current levels (of 600).

II. LOOKING AHEAD

A key question in all PDE work is whether or not a solution method which is effective in 1-D can be extended with similar success into 2-D applications. In the case of classical finite element methods, their forte historically has been in the effective resolution of multidimensional applications; and we expect that the MFE extension will further enhance the ultimate performance of finite element solution methods in 2-D. But beyond such a general statement of expectation, we consider explicitly in this section some of the specific properties that appeared in our 1-D MFE results which should carry over effectively into 2-D applications, as well. These properties include: (i) node economies, which will magnify computer savings in higher-dimensional applications; (ii) systemetic control of PDE numerical integrations, using an intrinsically stable integration method; (iii) high resolution of multiple, large gradients which may propagate at dissimilar velocities; (iv) elimination of zone tangling and/or remap aliasing in complex flow fields; and (v) non-dedicated programming.[6]

Without question, effective node control is the key factor which underlies many of these properties. The effectiveness of every adaptive regridding or moving node method is determined ultimately by the capacity to locate nodes "in the right place at the right time." By now we have seen that optimally-located nodes may have to satisfy several criteria simultaneously: their movement should be mobile—but in some cases not too mobile; the nodes should resolve accurately all significant curvatures in simultaneous dependent variables over entire problem domains—without wasting too many nodes either in areas of large gradients or in regions with small gradients; some nodes should remain sufficiently dispersed to resolve any new features (such as shocks or rarefactions) which may develop as a problem progresses; and nodes must be able to approach each other very closely (in order to resolve shocks and boundary layers simultaneously with larger scale effects in a flow field)—but the nodes must not cross in 1-D or tangle the mesh in higher dimensions. In 1-D applications we have found that: (i) such numerous and specific criteria can be expressed and executed effectively in terms of penalty functions in the MFE minimizing equations; and (ii) general users can exercise such node controls on a problem-dependent basis by the use of externally-specified control parameters. Future research will undoubtedly continue to improve the universality of node controlling penalty functions.

One of our examples in 1-D serves to illustrate MFE node control properties which are essential in extensions to 2-D. This example is the intersecting reactive wave problem from Section II, which illustrates the non-tangling, high-resolution properties of the MFE nodes. By recalling the discussion (and Fig. 15) in Section II we note that, at early times ($t < 60$), the MFE nodes move with the respective freely propagating square waves, very much in a Lagrangian fashion. But it must be emphasized that the MFE nodes have not been "attached" to either wave by any sort of a coordinate transformation, such as is done in conventional Lagragian calculations. A moment of truth occurs just prior to $t = 60$ when the square waves are about to intersect. At these times the nodes approach each other very closely without crossing. In this central region the MFE solution is now very much Eulerian, with the wave forms clearly propagating through the mesh of essentially stationary, closely spaced nodes about $x = 50$. Concurrently, the large gradients at the rear portions of the waves for fluids 1 and 2 are being resolved to high accuracy by the moving nodes in that region. This example shows that the finite elements contract to very small scales in regions of intersecting (and non-intersecting) large gradients and then elongate again when fine resolution is no longer required in those regions of the problem domain. Although the ODE system is a stiff system, extremely non-uniform grid meshes pose no integration difficulties for implicit, stiffly stable ODE solution methods, as is clear from the results which appeared in Section II.

---

[6] Existing practices of writing new dedicated PDE programs for each new multi-dimensional application can be unacceptably costly in many practical circumstances.

In 2-D, conceptually similar node behavior must occur simultaneously in the x and y coordinates. In contrast to the expanding and contracting interval elements which we have observed in 1-D, we will see in 2-D a mesh of expanding and contracting triangular elements which are calculated at each time-step—in complete analogy to the 1-D case—to minimize PDE residuals. In a flexible triangular mesh in 2-D, triangles are not permitted to become any "thinner" than a certain user-specified minimum separation. With the aid of a few hand-drawn sketches one can readily demonstrate that the same principles which yielded the non-tangling, high-resolution nodal properties in 1-D should apply with similar effectiveness in 2-D. A crucial question is whether or not steep profiles with dynamic contours can be resolved accurately in 2-D with a practical number of nodes.

The recent work by Alexander *et al.* [20, 21] provides a first answer to this question. For the Stefan problem (of ice–slush–water transitions [2, 20]) their MFE solutions in 2-D resolved very efficiently the evolution of simultaneous, steep wavefronts which, in fact, pinched together in mid-region as the problem evolved so that the ice region (initially dumbbell shaped) eventually severed—thereby changing its topological type completely.

Finally, one is interested in the capacity of a PDE solution method to solve practical systems of equations for physically meaningful variables. Again, a first indication of this capacity is available for the MFE method. We have solved the gas dynamics examples of Sod [22] and have obtained essentially exact results for all fluid variables, using fewer than 30 MFE nodes. Although a detailed discussion of these (and other) gas dynamics results is deferred to a separate article, we note the following points which are pertinent to the present discussion: In addition to significant node economies, the MFE time step was found to exceed the CFL condition by large factors (of 20 to 100). Very large gradients were resolved with high accuracy; in fact, viscous and thermal conductivity terms were incorporated with their *physical* magnitudes which, in turn, generated shock gradients on the order of 10^3. (Highly accurate resolution of these physical scales proves to be very beneficial to accurate partioning of internal energy and kinetic energy densities.) Lastly, new regularization terms were developed in order to emphasize the migration of nodes to all solution regions which contain high curvatures. This feature tends to prevent the buildup of nodes in those regions of steep fronts which are essentially straight lines; and the growth of such new structures as the shock front and rarefaction fan are sensed more effectively by these new penalty functions. This development further suggests that alternative penalty functions can be devised to satisfy many different types of problem requirements which may be faced by scientific practitioners in numerous disciplines.

APPENDIX

The DYLA Program

DYLA is an automated general computer program structure which can accommodate numerous PDE solution methods in the same overall package. The solution methods are pre-programmed and users can—by the use of simple input commands—select not only the partial differential equation system which they wish to compile and solve but also the PDE solution method which is to be used. At the present time, the DYLA code emphasizes the MFE method; however, in the past we have experimented with fixed node bilinear finite element methods (that is, the approximate solution is piecewise linear in both the space and the time variables) as well as centered finite-difference discretization methods [25] in the DYLA format. This general approach to PDE solution is intended to relieve practical users of the need to: (i) write a new computer program for the solution of new partial differential equation systems; and (ii) alter existing computer programs in order to incorporate new finite difference or finite element methods. In the first instance, we note that very large programming efforts are often expended in programming new equation systems which are in large part identical to those which existed in previous program versions. In the second instance of implementing new or revised solution methods, we note that existing computer programs are frequently victimized by a cancerous mode of growth in which reliability, maintainability, and efficiency decline drastically. The guiding philosophy in the construction of DYLA is to automate, as fully as practical judgments warrant, every potentially tedious or error-prone action which is required in the implementation of a physical modeling code.

We are, of course, constantly aware of the tenet that computational generality may destroy computational efficiency. But in this era of declining computing costs, it is now the human practitioner whose efficiency should be enhanced. The DYLA program structure permits problem extension, expansion, modification, and, finally, optimization to be made quickly, accurately, and safely. At this time, we have not observed any irretrievable loss of efficiency in the DYLA solution program, using the MFE method.

In undertaking the construction of an automatic partial differential equation solver which is to be used for modeling physical systems, one can identify the following major components:

(1) A system of time evolving differential equations together with a set of initial and boundary conditions. The differential equations generally contain partial differential operators of a few standard forms with a set of parameters that describes the physical system of interest. Such PDE systems are usually classified as parabolic or hyperbolic.

(2) Some types of spatial discretization of differential operators for the PDE systems of interest. Roughly speaking, in finite element methods one postulates the functional space of the approximate solution and thereafter obtains approximations of these differential operators acting upon the approximate solution. Alternatively, finite difference methods apply local approximations directly to the differential operators, *per se*. (Naturally, the "semi-discrete" continuous-in-time equations which result from these two different approaches are generally dissimilar, particularly for nonlinear PDEs.)

(3) Time integration of the resulting system of ordinary differential equations. Many of the most difficult physical problems require implicit, stiffly stable time integration methods. Such methods involve the solution of a large system of algebraic equations. We remark that in the MFE method, this system of algebraic equations is always nonlinear—even if the physical problem, *per se*, is linear.

We now describe more specifically the implementation of these main components in DYLA for the MFE method. Later we will note some modifications which would be required if one wished to use an alternative spatial discretization scheme in the general framework provided by the DYLA concept.

1. *The Differential Equations and Boundary Conditions*

Many automatic PDE solvers are constructed to solve a single specific type of differential equation system; e.g., a system of parabolic equations, a system of hyperbolic equations, etc. As may have been apparent in previous sections of this article, the DYLA program structure takes a different and more general form. A set of various operators which arise in scientific applications are programmed into the DYLA structure so that one can construct sets of differential equations by providing simple input commands. (Specific examples will be given below.) Therefore, a novel feature of DYLA—which became possible to implement via the robust and versatile nature of the MFE method—is that parabolic or hyperbolic systems of equations seem to be handled with equal facility. (Of course, an elliptic problem might be handled as the steady state case of a corresponding parabolic problem, but in many cases this may not be as efficient as other alternatives in terms of CPU time.)

[redacted]

Neumann and mixed boundary conditions will be tested in future work.

2. *Spatial Approximation Methods*

In implementing the piecewise linear MFE method, a first step which must be taken is the evaluation of some inner products which involve only basis functions. These inner products enter only in the mass matrix $\tilde{\mathscr{A}}$ in the equation (See Eq. (31) of Section I)

$$\tilde{\mathscr{A}}\dot{y} = \tilde{g}. \tag{A1}$$

The following results are readily obtained:

$$\begin{aligned} (\alpha_i, \alpha_j) &= \tfrac{1}{6}\,\Delta s_i, && j = i - 1 \\ &= \tfrac{1}{3}(\Delta s_i + \Delta s_{i+1}), && j = i \\ &= \tfrac{1}{6}\,\Delta s_{i+1}, && j = i + 1, \end{aligned} \tag{A2}$$

$$\begin{aligned}(\alpha_i, \beta_j) &= -\tfrac{1}{6}\,\Delta a_i, && j = i-1\\ &= -\tfrac{1}{3}(\Delta a_i + \Delta a_{i+1}), && j = i\\ &= -\tfrac{1}{6}\,\Delta a_{i+1}, && j = i+1,\end{aligned} \tag{A3}$$

$$\begin{aligned}(\beta_i, \beta_j) &= \tfrac{1}{6} m_i\, \Delta a_i, && j = i-1\\ &= \tfrac{1}{3}(m_i\, \Delta a_i + m_{i+1}\, \Delta a_{i+1}), && j = i\\ &= \tfrac{1}{6} m_{i+1}\, \Delta a_{i+1}, && j = i+1.\end{aligned} \tag{A4}$$

The remaining inner products which appear on the right-hand side of Eq. (A1) are problem-dependent. We give below the values of numerous inner products which were encountered in the examples in this article:[7]

$$\text{(i)} \qquad L(v) = \frac{\partial}{\partial x} v^p, \tag{A5}$$

$$(\alpha_i, L(v)) = \tfrac{1}{2}(\Delta a_i^p + \Delta a_{i+1}^p), \tag{A6}$$

$$(\beta_i^l, L(v)) = -\tfrac{1}{2}(m_i^l\, \Delta a_i^p + m_{i+1}^l\, \Delta a_{i+1}^p). \tag{A7}$$

$$\text{(ii)} \qquad L(v) = v^p \frac{\partial v^q}{\partial x}, \tag{A8}$$

$$(\alpha_i, L(v)) = Q_1 + Q_2, \tag{A9}$$

$$(\beta_i^l, L(v)) = -(m_i^l Q_1 + m_{i+1}^l Q_2), \tag{A10}$$

where

$$Q_1 = \Delta a_i^q(\tfrac{1}{2} a_i^p - \tfrac{1}{6}\Delta a_i^p), \tag{A11}$$

$$Q_2 = \Delta a_{i+1}^q(\tfrac{1}{2} a_i^p + \tfrac{1}{6}\, \Delta a_{i+1}^p). \tag{A12}$$

Because $\beta_i(x) = -m_i \alpha_i(x)$ on the ith cell and $\beta_i(x) = -m_{i+1} \alpha_i(x)$ on the $(i+1)$st cell, the relationships (A9) and (A10) between the inner products $(\alpha_i, L(v))$ and $(\beta_i, L(v))$ carry over for many operators $L(v)$. Henceforth, the values of the inner products $(\beta_i, L(v))$ will not be written when the relationship in Eq. (A9) and (A10) holds for some Q_1 and Q_2.

$$\text{(iii)} \quad L(v) = \frac{\partial^2}{\partial x^2} v^p, \tag{A13}$$

$$(\alpha_i, L(v)) = m_{i+1}^p - m_i^p,$$

$$(\beta_i^l, L(v)) = -\tfrac{1}{2}(m_i^l + m_{i+1}^l)(m_{i+1}^p - m_i^p). \tag{A15}$$

$$\text{(iv)} \qquad L(v) = v^p v^q, \tag{A16}$$

$$(\alpha_i, L(v)) = Q_1 + Q_2, \tag{A17}$$

---

[7] Except for inner products of basis functions with second-order derivatives, the derivations of these quantities are straightforward.

where

$$Q_1 = \frac{\Delta s_i}{2}\left(a_i^p a_i^q + \tfrac{1}{6}\Delta a_i^p\, \Delta a_i^q - \tfrac{1}{3}(a_i^p \Delta a_i^q + a_i^q\, \Delta a_i^p)\right), \tag{A18}$$

$$Q_2 = \frac{\Delta s_{i+1}}{2}\left(a_i^p a_i^q + \tfrac{1}{6}\Delta a_{i+1}^p\, \Delta a_{i+1}^q + \tfrac{1}{3}(a_i^p\, \Delta a_{i+1}^q + a_i^q\, \Delta a_{i+1}^p)\right). \tag{A19}$$

(v) $L(v) = \dfrac{\partial}{\partial x}\phi(v^p)$, where ϕ is some decent function, (A20)

$$(\alpha_i, L(v)) = Q_1 + Q_2, \tag{A21}$$

where

$$Q_1 = \phi(a_i^p) - \int_0^1 \phi(a_i^p - \Delta a_i^p \tau)\, d\tau, \tag{A22}$$

$$Q_2 = -\phi(a_i^p) + \int_0^1 \phi(a_i^p + \Delta a_{i+1}^p \tau)\, d\tau. \tag{A23}$$

In general, the integrals in Eqs. (A22) and (A23) are evaluated by numerical quadratures for arbitrary functions ϕ.

For each of the "primitive" operators in (i) through (v), there is a subroutine in DYLA which calculates the corresponding inner products and loads these values appropriately in the array $\tilde{g}$. In order to give an example showing how this is done, we first refer the reader to Table I, which contains a list of all operators in (i) through (v), as well as other operators which are currently implemented in DYLA. Note that, in Table I, the "problem variables" of the approximate solution vector v (i.e., the component functions v^l) are denoted by f, g, h, etc. The indices NF, NG, and NH denote that $l = NF$, NG, and NH, respectively, for the component functions f, g, and h. The generic coefficient K is a constant, independent of x, whereas Q is either an explicit or an implicit function of all other problem variables. The parameter LPDE is a pointer to the PDE containing the differential operator which is being loaded into $\tilde{g}$.

To illustrate the operation of DYLA, we can consider the previous two-reactive wave example, written in slightly different notation (than in the text) as:

$$\frac{\partial f}{\partial t} = -\frac{1}{2}\frac{\partial f}{\partial x} - fg, \tag{A24a}$$

$$\frac{\partial g}{\partial t} = \frac{1}{2}\frac{\partial g}{\partial x} - fg. \tag{A24b}$$

We start by ordering our solution vector as (f, g) and thus the indices of f and g are given by 1 and 2, respectively. In order to load the right-hand side of Eq. (A24a), we note first that LPDE = 1; so to load the operator $-(1/2)(\partial/\partial x)$ acting on the problem variable f, whose index NF = 1, we simply dial subroutine DFDX by the call statement (see Table I):

CALL DFDX(NF, LPDE, −0.5).

TABLE I

Library of Some Differential Operators for the MFE Method in DYLA

$L(v)$	Subroutine name and argument	Comments
(1) $K\,\partial f/\partial x$	DFDX(NF, LPDE, K)	● It is assumed that the problem variables g (or h) appearing in any denominator are bounded away from zero.
(2) $K(\partial/\partial x)(f^2/g)$	DF2OGDX(NF, NG, LPDE, K)	
(3) $K(\partial/\partial x)(f^3/g^2)$	DF3OG2(NF, NG, LPDE, K)	
(4) $K(\partial/\partial x)(f \times g/h)$	DFGOHDX(NF, NG, NH, L-PDE, K)	● Note that the operator in (1) through (5) can be used to generate the gas dynamics equations in conservation form.
(5) $K\,\partial^2 f/\partial x^2$	D2FDX2(NF, LPDE, K)	● Using (5) and (6) one can dial Burger's equation.
(6) $K f\,\partial g/\partial x$	FDGDX(NF, NG, LPDE, K)	
(7) Kf	KF(NF, LPDE, K)	● Using (1) and (8) one can dial the two-reactive wave problem.
(8) $Kf \times g$	KFG(NF, NG, LPDE, K)	
(9) $Kf \times g \times h$	KFGH(NF, NG, NH, LPDE, K)	● Note that all inner products in (1) through (9) and all inner products in the associated Jacobians are evaluated analytically. For the operators in (10)–(18) below, all such inner products are evaluated by numerical quadrature.

Specific operators for fluid dynamics applications

(10) $K\,\partial Q/\partial x$	● Q is some external function in one problem variable.
(11) Qf	● In (11), (12) and (13), Q is expected to be an Arrhenius-type rate coefficient, which is a function of temperature, which in turn is a function of all problem variables.
(12) $Qf \times g$	
(13) $Qf \times g \times h$	
(14) $K\,\partial p/\partial x$	$p \equiv$ pressure
(15) $K(\partial/\partial x)(V_i y_i)$	$y_i \equiv$ number density of ith chemical species
(16) $K\,\partial/\partial x \sum_i h_i V_i y_i$	$V_i \equiv$ peculiar velocity of ith chemical species (i.e., the statistically fluctuating component)
(17) $K(\partial/\partial x)(p \times m/\rho)$	$h_i \equiv$ enthalpy of ith chemical species
(18) $\partial/\partial x\,(\lambda(\partial T/\partial x))$	$m \equiv$ fluid momentum, $\rho \equiv$ fluid mass density
	$\lambda \equiv$ thermal conductivity, or diffusion coefficient
	$T \equiv$ temperature

In order to load the second term, $-fg$, we note that NG = 2; we thus need to dial subroutine KFG by the call statement:

CALL KFG(NF, NG, LPDE, –0.1).

This completes the loading of Eq. (A24a).

Analogously, we load the second equation, LPDE = 2. To summarize, the right-hand side of Eqs. (A24) is loaded into $\tilde{\mathscr{g}}$ as follows:[8]

Load the first equation:

CALL DFDX(1, 1, –0.5)

CALL KFG(1, 2, 1, –1.).

Load the second equation:

CALL DFDX(2, 2, 0.5)

CALL KFG(1, 2, 2, –1.).

As a second example, let us consider the one-dimensional gas dynamics equations in conservation form:

$$\rho_t = -m_x, \tag{A25a}$$

$$m_t = -\left(\frac{m^2}{\rho} + p\right)_x + \nu m_{xx}, \tag{A25b}$$

$$E_t = -\left(\left(\frac{m}{\rho}\right)(E+p)\right)_x + \nu' E_{xx}, \tag{A25c}$$

where ρ, m, and E are, respectively, mass density, momentum, and total energy per unit volume. The parameters ν and ν' (assumed to be equal henceforth) are viscosity and thermal conduction coefficients which are assigned *physical* magnitudes. The pressure p for an ideal gas is given by the equation of state

$$p = (\gamma - 1)\left(E - \frac{1}{2}\left(\frac{m^2}{\rho}\right)\right), \tag{A26}$$

where γ is the ratio of specific heats. Assuming that γ is constant, the substitution of p into Eq. (A25) gives

$$\rho_t = -m_x, \tag{A27a}$$

$$m_t = -((3-\gamma)/2)(m^2/\rho)_x - (\gamma - 1)E_x + \nu m_{xx}, \tag{A27b}$$

$$E_t = -\gamma((m/\rho)E)_x + ((\gamma-1)/2)(m^3/\rho^2)_x + \nu E_{xx}. \tag{A27c}$$

---

[8] These call statements are contained in a user-written (problem-dependent) subroutine; and, except for analytic Jacobian options or input-output requirements, the DYLA program operates with essentially no further user interfacing.

By assigning the indices 1, 2, and 3 to the variables ρ, m, and E, respectively, the right-hand side of Eq. (A27) is loaded into $\tilde{g}$ through the following lines of coding (See Table I):

Load Eq. (A27a):

CALL DFDX(1, 1, $-1.$).

Load Eq. (A27b):

CALL DF2OGDX(2, 1, 2, $-(3-\gamma)/2$),

CALL DFDX(3, 2, $-(\gamma-1)$),

CALL D2FDX2(2, 2, ν).

Load Eq. (A27c):

CALL DFGOHDX(2, 3, 1, 3, $-\gamma$)

CALL DF3OG2(2, 1, 3, $(\gamma-1)/2$),

CALL D2FDX2(3, 3, ν).

This concept of automated dialing and subsequent loading of partial differential equations can be implemented for a variety of different spatial approximation methods. Accordingly, DYLA could be extended to contain several different PDE solution methods; and one could then select a particular solution method via a simple method flag designator—much as is done in such automatic ODE solution packages as EPISODE [5].

One example would be to implement a fixed node piecewise linear finite element method. The ODEs for such a method can be obtained by merely eliminating the $s_j(t)$ from the set of unknowns in the solution vector $y(t)$ and all inner products involving the basis function β_i^l from the mass matrix $\tilde{\mathscr{A}}(y)$ and from the vector valued function $\tilde{g}(y)$.

A second example might be to implement a central finite difference approximation on a uniform grid with node spacing h. The basic block tridiagonal structure of the resulting ODEs would not be changed; and the same implicit stiff ODE part of the program, as well as the same "dial an operator" input logic, can all be retained.

3. *Time Integration of a System of ODEs*

The ODE system which is solved in Eq. (A1) is frequently stiff because of the extreme non-uniformity in the distribution of nodal spacings. We therefore must use an implicit stiffly stable ODE solver in order to obtain accurate numerical solutions with economical time-steps. Currently DYLA uses Gear's implicit stiffly stable [illegible] package integrated readily all of the examples which appeared in this article. Two specific modifications which were essential to solving the system of Eq. (A1) are (i) the derivative (with respect to y) of the mass matrix $\tilde{\mathscr{A}}$ must be evaluated in calculating the Jacobian in DYLA, whereas these derivatives were discarded in GEARIB, and (ii) additional error control norms on the nodal spacings, Δs_j, were included in the ODE solutions in DYLA. Finally, the reader may note that we have also tested the A-stable, second-order diagonally implicit Runge–Kutta method DIRK2 [1] for solving the ODE systems in DYLA. We found that, due to a lack of code optimization of the DIRK2 method, the GEARIB package currently yields a somewhat greater computational economy.

REFERENCES

1. K. MILLER AND R. MILLER, *SIAM J. Numer. Anal.*, in press.
2. K. MILLER, *SIAM J. Numer. Anal.*, in press.
3. C. W. GEAR, *Comm. Assoc. Comput. Mach.* **14** (1971), 176–179.
4. R. J. GELINAS, *J. Comput. Phys.* **9**, No. 2 (1972), 222.
5. A. C. HINDMARSH AND G. D. BYRNE, "EPISODE: An Effective Package for the Integration of Systems of Ordinary Differential Equations," Lawrence Livermore Laboratory Report UCID-30112, Rev. 1, Computer Documentation, April 1977.
6. R. MCCORMACK AND A. J. PAULLAY, *Computers and Fluids* **2** (1974), 339–361.
7. E. S. ORAN, T. YOUNG, AND J. BORIS, *in* "Seventeenth Symposium (International) on Combustion," p. 43, The Combustion Institute, 1979.
8. E. S. ORAN, J. P. BORIS, T. YOUNG, M. FLANIGAN, T. BURKS, AND M. PICONE, "Detonations in Hydrogen-Air and Methane–Air Mixtures," Naval Research Laboratory preprint. 1979.
9. H. A. DWYER, R. J. KEE, AND B. R. SANDERS, *in* "Proceedings, 7th International Colloquium on Gas Dynamics of Explosions and Reactive Systems, AIAA, 1979.
10. J. OLIGER, Invited talk at SIAM Meeting in Denver, Colorado, November 1979, and subsequent private communications.
11. J. T. ODEN, "Finite Elements of Nonlinear Continua," McGraw–Hill, New York, 1972.
12. G. STRANG AND G. J. FIX, "An Analysis of the Finite Element Method," Prentice–Hall, Englewood Cliffs, N.J., 1973.
13. S. K. DOSS AND K. MILLER, "Moving Finite Element Solution of the Vorticity Equations," presently unpublished results (1974). [See also References 1 and 2 for a brief summary of these results.]
14. M. J. DJOMEHRI, "Moving Finite Element Solution of Systems of Partial Differential Equations in 1-Dimension," Ph.D. Thesis, University of California, Berkeley.
15. J. P. BORIS AND D. L. BOOK, *J. Comput. Phys.* **11** (1973), 38. [See also Boris, *et al.*, *J. Comput. Phys.* **18** (1975), 248 and *J. Comput. Phys.* **20** (1976), 397.]
16. G. C. BYRNE, *in* "Proceedings, American Nuclear Society Topical Meeting on Computational Methods in Nuclear Engineering, Williamsburg, VA, April 1979."
17. P. CONCUS AND W. PROSKUROWSKI, "Numerical Solution of a Nonlinear Hyperbolic Equation by the Random Choice Method," Lawrence Berkeley Laboratory Report LBL-6487, Rev., December 1977.
18. H. A. DWYER AND B. R. SANDERS, "Numerical Modeling of Unsteady Flame Propagation," Sandia Livermore Laboratories Report SAND 77-8275, February 1978.
19. G. R. OTEY AND H. A. DWYER, "A Numerical Study of the Interaction of Fast Chemistry and Diffusion," Sandia Livermore Laboratories Report SAND 78-8686, April 1979.
20. R. ALEXANDER, P. MANSELLI, AND K. MILLER, to appear.
21. R. ALEXANDER, P. MANSELLI, AND K. MILLER, "Moving Finite Elements for the Stefan Problem in Two Dimensions," presented at SIAM 1979 Fall Meeting, Denver, Colorado, November 12–14, 1979.

22. G. SOD, *J. Comput. Phys.* **27** (1978), 1–31.
23. A. C. HINDMARSH, "Preliminary Documentation of GEARIB: Solution of Implicit Systems of ODE's with Banded Jacobian," Lawrence Livermore Laboratory Report UCID-30130, Computer Documentation, February 1976.
24. A. C. HINDMARSH, "Solution of Block-Tridiagonal Systems of Linear Algebraic Equations," Lawrence Livermore Laboratory Report UCID-30150, Computer Documentation, February 1977.
25. R. J. GELINAS, D. K. HALL, AND R. G. NELSON. *Nature* **266**, No. 559 (1977), 229.