JOURNAL OF COMPUTATIONAL PHYSICS 79, 245–269 (1988)

Moving Finite Element Methods for Evolutionary Problems. I. Theory

M. J. BAINES

Department of Mathematics, University of Reading, Reading, United Kingdom

AND

A. J. WATHEN

School of Mathematics, University of Bristol, Bristol, United Kingdom

Received October 13, 1986; revised September 1, 1987

In this paper, the first of two on the subject, we present a unified approach to moving and fixed finite element methods for evolutionary problems in terms of projections. The central theoretical results are concerned with moving finite elements for one-dimensional scalar problems (particularly hyperbolic equations with shocks), but the viewpoint extends to general systems in any number of dimensions. © 1988 Academic Press, Inc.

1. INTRODUCTION

Finite element methods are well established for equilibrium problems, such as those of stress analysis and have been employed extensively in time-dependent problems, such as those of fluid flow. They have been shown to retain valuable conservation properties and to exhibit high accuracy. However, for problems involving shocks or steep fronts, oscillations present great difficulties.

If the principal feature of interest is a front moving across a domain, fixed mesh methods are found to be inefficient. It was for such problems that the *moving finite element* method [1, 2], in which the grid moves with the feature, was invented. In this paper we discuss finite element methods on fixed and moving grids from a unified point of view, thus giving an insight into the way in which such methods work in time-dependent problems.

Consider evolution equations in any number of dimensions of the form

$$u_t = L(u), \tag{1.1}$$

where all space derivatives are contained in the operator L, and approximations U to u of the form

$$U=\sum_j a_j\alpha_j, \tag{1.2}$$

where a_j is a coefficient and α_j a basis function. Denote by S_α the space spanned by the α_j. We shall assume unless otherwise stated that U belongs to the domain of the operator L. This point is discussed further in Section 5.

To approximate u_t we differentiate (1.2) with respect to time. In order to do so we must specify which parts of a_j and α_j are time dependent.

Suppose first that only the a_j are time dependent, as for example in the fixed finite element (FFE) method. Then

$$U_t=\sum \dot{a}_j\alpha_j, \tag{1.3}$$

the dot denoting differentiation with respect to time. Note that $U_t\in S_\alpha$.

In the unlikely case that $L(U)\in S_\alpha$ we can satisfy (1.1) by solving a system of ordinary differential equations (ODEs) for the a_j by matching coefficients of both sides. If not, we can satisfy (1.1) approximately by first projecting $L(U)$ into S_α, obtaining $P_\alpha L(U)$, say. This is equivalent to minimising the residual of (1.1) over the coefficients $\dot{a}_j$.

Any norm will do for the projection but if we use the L_2 norm, the resulting normal equations lead, by matching coefficients of U_t, to the usual Galerkin equations

$$\langle\alpha_i, U_t-L(U)\rangle=0 \qquad \text{for each } i. \tag{1.4}$$

Together (1.4) and (1.3) give the system of ODEs

$$A\dot{\mathbf{y}}=\mathbf{g}, \tag{1.5}$$

where $A=\{A_{ij}\}$ is the mass matrix,

$$A_{ij}=\langle\alpha_i,\alpha_j\rangle, \quad \mathbf{y}=\{a_j\} \qquad \text{and} \qquad \mathbf{g}=\{\langle\alpha_i, L(u)\rangle\}. \tag{1.6}$$

Here $\langle\cdot,\cdot\rangle$ is the usual L_2 inner product. A is clearly symmetric.

We note one special case. If α_j is the usual linear "hat" basis function in one dimension (see Fig. 4.1a), A is positive definite tri-diagonal and its inverse is full.

The system (1.5) together with a suitable time stepping procedure comprises the Galerkin method for partial differential equations of the form (1.1). Although the method has many useful properties it is, in particular, unsatisfactory for time accurate hyperbolic problems involving shocks. This may be ascribed to the global projection of $L(U)$ which is unphysical in terms of information flow in this case. (A local projection would be more appropriate (see below)).

Take next the moving finite element (MFE) approximation (see [1–3]), in which both the a_j and α_j in (1.2) are time dependent, the latter through its dependence on time-dependent node positions. Then, for *linear* α_j (as in, e.g., [1–5]) we have

$$U_t = \sum (\dot{a}_j \alpha_j + \dot{\mathbf{s}}_j \cdot \boldsymbol{\beta}_j), \tag{1.7}$$

where $\mathbf{s}_j$ is the position vector of node j and

$$\boldsymbol{\beta}_j = -(\nabla U)\,\alpha_j \tag{1.8}$$

is a second type of basis function, which is a different constant multiple of α_j in each element (see [6, 7]). It follows that each component of $\boldsymbol{\beta}_j$ has the same support as α_j but is in general discontinuous across element edges. The functions α_j and a typical component of $\boldsymbol{\beta}_j$ are illustrated in Fig. 1.1 for the two-dimensional case.

Denote by $S_{\alpha\beta}$ the space spanned by the α_j and $\boldsymbol{\beta}_j$. This space is solution dependent and hence time dependent through the appearance of ∇U in (1.8). Note that if the U of (1.2) is to be continuous, U_t must belong to the space $S_{\alpha\beta}$, which is in general a subspace of the space S_ϕ of *all* piecewise linear discontinuous functions on the same finite element grid. In one dimension, however,

$$U_t = \sum (\dot{a}_j \alpha_j + \dot{s}_j \beta_j), \tag{1.9}$$

where

$$\beta_j = -m\alpha_j \tag{1.10}$$

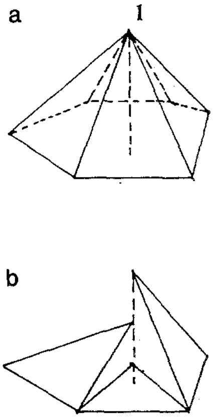

FIG. 1.1. α_j and β_j basis functions in two-dimensions.

and

$$m = dU/dx \tag{1.11}$$

is the (piecewise constant) slope of the solution. In this exceptional case $S_{\alpha\beta} \equiv S_\phi$.

If $L(U) \in S_{\alpha\beta}$ we can satisfy (1.1) by solving a system of ODEs for the a_j and s_j by matching coefficients. This can be done, for example, in one dimension for the equation

$$u_t = L(u) = -uu_x, \tag{1.12}$$

for which $L(U) \in S_{\alpha\beta}$, leading to the equations for characteristics. If $L(U) \notin S_{\alpha\beta}$ we can still satisfy (1.1) approximately by projecting $L(U)$ into $S_{\alpha\beta}$, obtaining $P_{\alpha\beta}L(U)$, say. This is equivalent to minimising the residual of (1.1) over the coefficients $\dot{a}_j$ and $\dot{s}_j$.

Once again, any norm will do for the projection, but if we take the L_2 norm we reproduce the Miller MFE method [1, 2] in the absence of penalty functions. This leads to the double set of Galerkin equations

$$\langle \alpha_i, U_t - L(U) \rangle = 0 \qquad \text{for each } i \tag{1.13}$$

$$\langle \boldsymbol{\beta}_i, U_t - L(U) \rangle = 0 \qquad \text{for each } i \tag{1.14}$$

which, after substituting for U_t from (1.7), gives

$$A(\mathbf{y})\,\dot{\mathbf{y}} = \mathbf{g}, \tag{1.15}$$

where $A(\mathbf{y})$ consists of blocks

$$A_{ij} = \begin{bmatrix} \langle \alpha_i, \alpha_j \rangle & \langle \alpha_i, \boldsymbol{\beta}_j \rangle \\ \langle \boldsymbol{\beta}_i, \alpha_j \rangle & \langle \boldsymbol{\beta}_i, \boldsymbol{\beta}_j \rangle \end{bmatrix}, \tag{1.16}$$

$\mathbf{y}^{\mathrm{T}}$ is in blocks $\mathbf{y}_j^{\mathrm{T}} = [a_j, \mathbf{s}_j^{\mathrm{T}}]$, and $\mathbf{g}$ is in blocks

$$\mathbf{g}_i = \begin{bmatrix} \langle \alpha_i, L(U) \rangle \\ \langle \boldsymbol{\beta}_i, L(U) \rangle \end{bmatrix}. \tag{1.17}$$

The matrix A is symmetric and it has been shown by Wathen and Baines [3] that, for the present case of linear basis functions, A is positive semi-definite with a simple decomposition of the form $M^{\mathrm{T}}CM$, where C is a block diagonal matrix and M is a rectangular matrix, rectangularly block diagonal under a permutation. Moreover, in one dimension, A decomposes into three block diagonal square matrices. We note briefly the reason for this.

In one dimension introduce the linear half-basis functions $\varphi_{k,1}$, $\varphi_{k,2}$ in an element k (see Fig. 4.1c). Since the $\varphi_{k,v}$ span the same space as the α_j, β_j in this one-dimensional case, projection of $L(U)$ into the S_ϕ or $S_{\alpha\beta}$ space leads to the same function. Because of the local nature of the $\varphi_{k,v}$ functions, projection into the S_ϕ space leads to a diagonal system to be solved. This gives one of the matrices C of the decomposition. The matrix M arises from transferring the solution on to nodes rather than elements. In other words the MFE method in one dimension decomposes into two local processes. This point will be considered further in Section 4.

The fixed finite element method is clearly equivalent to the MFE method with a constraint, namely that $\dot{s}_j = 0$ for each j. In this case the projection of $L(U)$ into $S_{\alpha\beta}$ must be constrained to lie in S_α, or a double projection will be necessary. The same is true for what one might call the Lagrangian finite element (LFE) method, which constrains $\dot{a}_j = 0$, and requires a projection into S_β.

We wish to unify the above ideas into a single structure. We confine the α_j in the approximation (1.2) to be piecewise linear functions so that U is continuous and piecewise linear. (The generalisation to higher order elements is considered in Section 5.) On the current grid in any number of dimensions introduce element basis functions $\varphi_{k,v}$ in each element k: these are defined to be piecewise linear in each element k, taking the value 1 at vertex v and zero all other vertices (see Fig. 2.1). As above we denote by S_ϕ the space spanned by the $\varphi_{k,v}$, i.e. the space of piecewise linear functions on the grid without continuity across element edges.

Now project $L(U)$ elementwise into the local subspace of S_ϕ. In the case of an L_2 projection this leads to a $(d+1)\times(d+1)$ matrix to be inverted in each element, where d is the number of dimensions (see Section 2). Matching coefficients of the projection with those of U_t gives an "element motion" which may, however, be inconsistent with the continuity requirements of U. In one-dimensional MFE the nodal velocities are easily generated from the "element motions" using a 2×2 matrix system for each node, as will be described in Section 4. In all other cases, however, including higher dimensional MFE and the fixed finite element (FFE) method, the element motions lead to discontinuous U functions and a further projection is needed. This projection is nevertheless easy to characterise and we can regain all the methods discussed so far in a simple way. The viewpoint gives a unified structure to these methods and suggests possible new ones.

The layout of the paper is as follows. In Section 2 details of the local projection of $L(U)$, on which all the methods are constructed, are given. The relationship with both MFE and FFE is brought out in Section 3, in which details of further projections necessary are presented. Section 4 contains an exposition of the simplest form of evolutionary finite element method from this viewpoint, namely MFE in one dimension. In the final Section 5 we show how generalisations to other operators L and higher order basis functions affect the theory and include a discussion of the role of boundary conditions. The important extension to systems of equations is also explored in this section.

2. Local Projection

As suggested in the Introduction, we can regard the approximate solution U of the partial differential equation (1.1) as being driven by the projection of the function $L(U)$ into the space inhabited by U_t.

We again assume in this section that U lies in the domain of L. We also assume that U belongs to the space of piecewise linear continuous functions and therefore that U_t, as in (1.7) (or (1.3)), belongs to the space $S_{\alpha\beta}$ (or S_α) of certain piecewise linear discontinuous functions (or all piecewise linear continuous functions, respectively) on the current grid.

First we discuss the projection of $L(U)$ into S_ϕ, the space of *all* piecewise linear discontinuous functions on the finite element grid. This space is spanned by the basis functions $\varphi_{k,\nu}$ which have been defined in Section 1. A sketch of $\varphi_{k,\nu}$ in the two-dimensional case is given as Fig. 2.1. Define the projection of $L(U)$ into S_ϕ by

$$PL(U) = \sum_k \sum_v \dot{w}_{k,\nu} \phi_{k,\nu}. \tag{2.1}$$

We achieve the projection, in any norm $\|\cdot\|$, by minimising

$$\|L(U) - PL(U)\| \tag{2.2}$$

over the coefficients $\dot{w}_{k,v}$.

If we choose the L_2 norm, then such is the interaction of the $\varphi_{k,\nu}$ that we find that the matrix equations for the $\dot{w}_{k,\nu}$ break into independent elementwise sets (see Ref. [4]). Thus, for the element k we arrive at the normal equations

$$C_k \dot{\mathbf{w}}_k = \mathbf{b}_k, \tag{2.3}$$

where

$$C_k = \{C_{k,\nu,\mu}\}, \qquad \mathbf{b}_k = \{b_{k,\nu}\}, \qquad \text{and} \qquad \dot{\mathbf{w}}_k = \{\dot{w}_{k,\nu}\} \tag{2.4}$$

with

$$C_{k,\nu,\mu} = \langle \varphi_{k,\nu}, \varphi_{k,\mu} \rangle \tag{2.5}$$

and

$$b_{k,\nu} = \langle \varphi_{k,\nu}, L(U) \rangle. \tag{2.6}$$

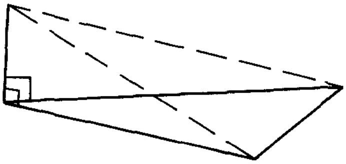

FIG. 2.1. $\varphi_{k,\nu}$ basis function in two-dimensions.

Note that the matrix C_k is a $(d+1)$-square matrix, where d is the number of spatial dimensions, so that normally at most a 4×4 matrix system is involved.

Provided that the support of no $\varphi_{k,v}$ vanishes, i.e., no element becomes vanishingly small, C_k is positive definite and we may invert (2.3) obtaining

$$\dot{\mathbf{w}}_k = C_k^{-1}\mathbf{b}_k \tag{2.7}$$

for each element . This gives an approximate segment motion for the element. We shall return to this point in Section 4.

Having found $PL(U)$ locally in the space S_ϕ of piecewise linear discontinuous functions we ask whether this function can be made to match the function U_t, in the sense of lying in the same space. In general this match cannot be made exactly and hence a global projection step is required. In the special case of one-dimensional MFE with linear elements such a match may be made exactly since $PL(U)$ lies in S_ϕ, U_t lies in $S_{\alpha\beta}$, and in one dimension $S_\phi \equiv S_{\alpha\beta}$. Since no global projection step is required it may be seen that in this special case of linear elements in one dimension, the MFE method is purely local. We follow up this point in our specific discussion of one-dimensional MFE in Section 4.

Generally, however, $PL(U)$ lies in S_ϕ but not in the subspaces $S_{\alpha\beta}$ or S_α. In these cases a further projection is necessary to generate a U_t which is consistent with a continuous function U. This aspect is discussed fully in the next section.

3. Relationship between Fixed and Moving Finite Elements

For clarity of exposition let

$$\boldsymbol{\alpha}_i = (\alpha_i, \boldsymbol{\beta}_i^{\mathrm{T}})^{\mathrm{T}} \tag{3.1}$$

be a vector of nodal basis functions for the moving finite element method corresponding to the node i, and

$$\boldsymbol{\alpha} = (\boldsymbol{\alpha}_i^{\mathrm{T}}, \boldsymbol{\alpha}_2^{\mathrm{T}}, ..., \boldsymbol{\alpha}_n^{\mathrm{T}})^{\mathrm{T}} \tag{3.2}$$

be a vector of all such basis function vectors. For the fixed mesh Galerkin method we will correspondingly have $\boldsymbol{\alpha}_i = (\alpha_i)$, a scalar function, and (3.2). In each case the entries of $\boldsymbol{\alpha}$ form a basis for the space $S_{\alpha\beta}$ or S_α for MFE and FFE, respectively.

Now since each α_i and each entry of $\boldsymbol{\beta}_i$ is piecewise linear on the same grid as the $\varphi_{k,v}$, it follows that $S_\alpha \subset S_{\alpha\beta} \subseteq S_\phi$ and that for any $\boldsymbol{\alpha}$ there exists a matrix M, say, such that

$$\boldsymbol{\alpha} = M^{\mathrm{T}}\boldsymbol{\varphi}, \tag{3.3}$$

where

$$\boldsymbol{\varphi} = (\varphi_{1,1}, ..., \varphi_{1,d+1}; \varphi_{2,1}, ..., \varphi_{2,d+1}; ...; \varphi_{n,1}, ..., \varphi_{n,d+1})^{\mathrm{T}} \tag{3.4}$$

is a vector containing all the element basis functions $\varphi_{k,v}$, which form a basis for S_ϕ. The matrix M^T thus represents the transformation between the natural basis for S_ϕ and $S_{\alpha\beta}$ (or S_α).

Now we wish to find $U_t \in S_{\alpha\beta}$ such that the point function

$$U_t = \mathbf{a}^T\dot{\mathbf{y}} = \boldsymbol{\varphi}^T M\dot{\mathbf{y}} \tag{3.5}$$

can be matched with the projection (Fig. 3.1), given by

$$PL(U) = \sum \dot{w}_{k,v}\varphi_{k,v} = \boldsymbol{\varphi}^T\dot{\mathbf{w}} \in S_\phi, \tag{3.6}$$

of $L(U)$ into the subspace $S_{\alpha\beta}$, where

$$\dot{\mathbf{w}} = (\dot{w}_{1,1}, ..., \dot{w}_{1,d+1}; \dot{w}_{2,1}, ..., \dot{w}_{2,d+1}; ...; \dot{w}_{n,1}, ..., \dot{w}_{n,d+1})^T .$$

A natural approach is to minimise

$$\|U_t - PL(U)\|_{L_2} \tag{3.7}$$

over $\dot{\mathbf{y}}$, i.e., to seek

$$\min_{\dot{\mathbf{y}}} \|\boldsymbol{\varphi}^T M\dot{\mathbf{y}} - \boldsymbol{\varphi}^T\dot{\mathbf{w}}\|_{L_2}. \tag{3.8}$$

For M of full rank, this is equivalent to finding the unique element in $S_{\alpha\beta}$ given by $\dot{\mathbf{y}}$ which satisfies

$$M^T\langle\boldsymbol{\varphi}, \boldsymbol{\varphi}^T\rangle M\dot{\mathbf{y}} = M^T\langle\boldsymbol{\varphi}, \boldsymbol{\varphi}^T\rangle\dot{\mathbf{w}}, \tag{3.9}$$

i.e.,

$$M^T CM\dot{\mathbf{y}} = M^T C\dot{\mathbf{w}} = M^T\mathbf{b} = \mathbf{g}, \tag{3.10}$$

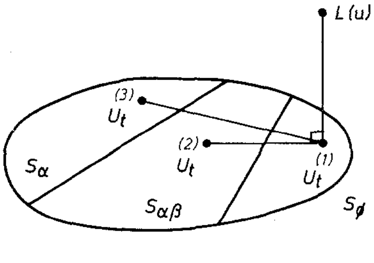

FIG. 3.1. Projections $PL(U)$ of $L(U)$ into S_ϕ, $S_{\alpha\beta}$, S_α, denoted by $U_t^{(1)}$, $U_t^{(2)}$, and $U_t^{(3)}$, respectively.

since $C = \langle \boldsymbol{\varphi}, \boldsymbol{\varphi}^{\mathrm{T}} \rangle$ ((2.5)). This is the MFE method described by Miller and Miller [1, 2], Wathen and Baines [3]. In purely algebraic terms (3.8) is equivalent to finding

$$\min_{\dot{\mathbf{y}}} \| C^{1/2}(M\dot{\mathbf{y}} - \dot{\mathbf{w}}) \|_{l_2}. \tag{3.11}$$

The MFE method for the partial differential equation

$$u_t - L(u) = 0 \tag{3.12}$$

thus comprises the two projection steps, namely, to find

$$\begin{aligned} &(1) \quad P_1 L(U) = \boldsymbol{\varphi}^{\mathrm{T}}\dot{\mathbf{w}} \in S_\phi \text{ by minimising } \|\boldsymbol{\varphi}^{\mathrm{T}}\dot{\mathbf{w}} - L(U)\|_{L_2} \text{ over } \dot{\mathbf{w}} \\ &(2) \quad P_2 P_1 L(U) = P_2(\boldsymbol{\varphi}^{\mathrm{T}}\dot{\mathbf{w}}) = \boldsymbol{\alpha}^{\mathrm{T}}\dot{\mathbf{y}} \in S_{\alpha\beta} \text{ by minimising } \|C^{1/2}(M\dot{\mathbf{y}} - \dot{\mathbf{w}})\|_{l_2} \\ &\qquad \text{over } \dot{\mathbf{y}}. \end{aligned} \tag{3.13}$$

Step (1) is local elementwise, but the weighted projection (2) is global unless $\boldsymbol{\varphi}^{\mathrm{T}}\dot{\mathbf{w}} \in S_{\alpha\beta}$. (In one-dimensional MFE $S_{\alpha\beta} \equiv S_\phi$, so step (2) is equivalent to the equality

$$M\dot{\mathbf{y}} = \dot{\mathbf{w}} \tag{3.14}$$

which is local nodewise in the sense that M is block diagonal nodewise (see [3] or the description given in the next section).)

For the FFE method the same analysis applies, the only difference being in the matrix M, which now represents the mapping of the basis of S_ϕ to the basis of S_α (rather than to the basis of the larger subspace $S_{\alpha\beta}$ as in MFE (see [10])). The Galerkin FFE method for the evolutionary partial differential equation (1.1) is thus seen to comprise the local projection step, i.e., to find

$$\min_{\dot{\mathbf{w}}} \|\boldsymbol{\varphi}^{\mathrm{T}}\dot{\mathbf{w}} - L(U)\|_{L_2}, \tag{3.15}$$

followed by the least squares step, find

$$\min_{\dot{\mathbf{y}}} \|C^{1/2}(M\dot{\mathbf{y}} - \dot{\mathbf{w}})\|_{l_2},$$

where M is simply a boolean matrix which represents the assembly of element matrices C_k into the global matrix, as described in [10]. Note that $S_\alpha \subset S_\phi$, the inclusion always being strict, so that for FFE the map M is always rectangular whatever the dimension; thus there is no local Galerkin FFE method.

A weaker local property of both FFE and MFE has, however, been proved by Wathen [9, 10]. This is the observation that algebraically the eigenvalues of the "mass matrix" $A = M^{\mathrm{T}}CM$, when preconditioned by the inverse of the matrix $D = M^{\mathrm{T}}$ (diag(C)) M are tightly clustered in a precise manner about unity.

4. Moving Finite Elements in One Dimension

4.1. *General*

We now give details of the MFE method in one dimension. The main property of the method in this case is its local character (see [5]), which we shall now emphasise.

The function U is again (1.2) where the α_j are the usual one-dimensional linear "hat" functions as shown in Fig. 4.1a. The time derivative U_t of U is given by

$$U_t = \sum_j \dot{a}_j \alpha_j + \dot{s}_j \beta_j \tag{4.1}$$

(c.f. (1.7)), where the $\dot{s}_j$ and β_j are scalars and, as in (1.10),

$$\beta_j = -m\alpha_j, \tag{4.2}$$

m being the local slope of U (varying from element to element). The function β_j is shown in Fig. 4.1b. We now follow the line of argument begun in Section 1 and introduce the linear half basis functions $\varphi_{k,1}$, $\varphi_{k,2}$, shown in Fig. 4.1c. Note that the space $S_{\alpha\beta}$ spanned by the α_j, β_j coincides with the space $S_{\alpha\beta}$ spanned by the $\varphi_{k,1}$, $\varphi_{k,2}$.

Consider now $L(U)$. (We assume again that U lies in the domain of the operator L). If $L(U) \in S_\phi$ as it does when, for example, $L(u) = uu_x$ we obtain immediately ODEs for the $\dot{a}_j$ and $\dot{s}_j$ which, in this case, give precisely the solution by characteristics. If $L(U) \notin S_\phi$ then as in (2.1) we first project $L(U)$ into the space S_ϕ spanned by $\varphi_{k,1}$ and $\varphi_{k,2}$. This gives

$$PL(U) = \sum_k \dot{w}_{k,1}\varphi_{k,1} + \dot{w}_{k,2}\varphi_{k,2}, \tag{4.3}$$

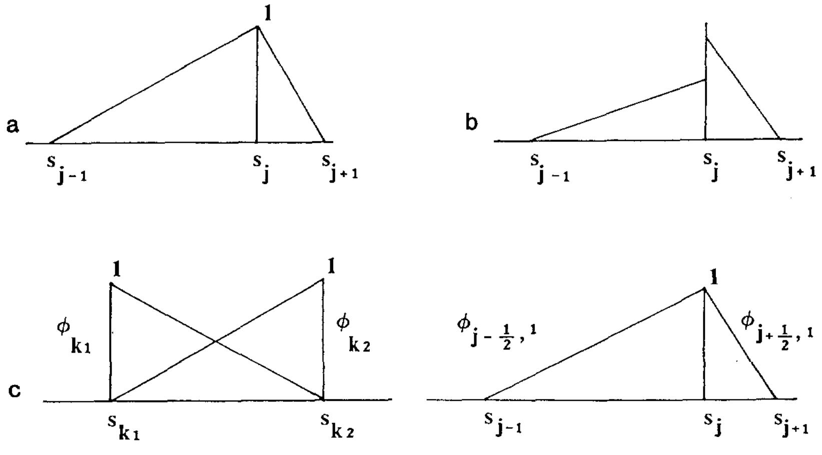

FIG. 4.1. α_j, β_j, and φ_k basis functions in one dimension.

where in the case of L_2 projection, as in (2.3),

$$C_k \dot{\mathbf{w}}_k = \mathbf{b}_k. \tag{4.4}$$

In this case,

$$C_k = \begin{bmatrix} C_{k,1,1} & C_{k,1,2} \\ C_{k,2,1} & C_{k,2,2} \end{bmatrix}, \qquad \mathbf{b}_k = \begin{bmatrix} b_{k,1} \\ b_{k,2} \end{bmatrix}, \tag{4.5}$$

the elements being given by (2.5) and (2.6). In one dimension we may evaluate the inner products in (2.5) to obtain

$$C_k = \Delta s_k \begin{bmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{bmatrix}, \tag{4.6}$$

$$b_{k,1} = \langle \varphi_{k,1}, L(U) \rangle, \qquad b_{k,2} = \langle \varphi_{k,2}, L(U) \rangle, \tag{4.7}$$

where Δs_k is the length of the element k.

We are now in a position to match the left- and right-hand sides of the approximation

$$U_t = PL(U) \tag{4.8}$$

to Eq. (1.1). Let the node j connect elements k and $k+1$. Then (see Fig. 4.1)

$$\begin{aligned} \alpha_j &= \varphi_{k,2} + \varphi_{k+1,1} \\ \beta_j &= -m_k \varphi_{k,2} - m_{k+1} \varphi_{k+1,1} \end{aligned} \tag{4.9}$$

(i.e., $\boldsymbol{\alpha} = M^{\mathrm{T}} \boldsymbol{\varphi}$ as in (3.3)). From (4.8), (4.1), (4.3), and (4.9) we obtain

$$\begin{aligned} \dot{a}_j - m_k \dot{s}_j &= \dot{w}_{k,2} \\ \dot{a}_j - m_{k+1} \dot{s}_j &= \dot{w}_{k+1,1} \end{aligned} \tag{4.10}$$

at least for interior nodes. We may write (4.10) in the form of (3.14) as

$$M_j \dot{\mathbf{y}}_j = \dot{\mathbf{w}}_j, \tag{4.11}$$

where

$$M_j = \begin{bmatrix} 1 & -m_k \\ 1 & -m_{k+1} \end{bmatrix}, \qquad \dot{\mathbf{y}}_j = \begin{bmatrix} \dot{a}_j \\ \dot{s}_j \end{bmatrix}, \qquad \dot{\mathbf{w}}_j = \begin{bmatrix} \dot{w}_{k,2} \\ \dot{w}_{k+1,1} \end{bmatrix}. \tag{4.12}$$

Provided that $m_k \neq m_{k+1}$ Eqs. (4.10) or (4.11) may be solved to give

$$\begin{aligned} \dot{a}_j &= \frac{m_{k+1} \dot{w}_{k,2} + m_k \dot{w}_{k+1,1}}{m_{k+1} - m_k} \qquad \text{for each } j \\ \dot{s}_j &= \frac{\dot{w}_{k,2} - \dot{w}_{k+1,1}}{m_{k+1} - m_k} \qquad \text{for each } j, \end{aligned} \tag{4.13}$$

a set of ODEs for the evolution of the nodal point (s_j, a_j). To integrate (4.13) for a_j, s_j we need only to add a time stepping algorithm.

It is convenient first to summarise the method. The steps are

1. Project $L(U)$ into S_ϕ locally within an element.
2. Use this information to calculate $\dot{a}_j$, $\dot{s}_j$ locally at a node.

In the case of the L_2 norm, step 1 gives rise to Eqs. (4.4) with C_k given by (4.6) and $\mathbf{b}_k$ given by (4.7). This is the main step. Step 2 merely transfers the information gained in step 1 from elementwise information to nodewise information, a purely algebraic step.

Note that both steps involve only 2×2 matrices. This is, of course, a result of the local nature of the method. It is clear that the method is particularly well adapted to hyperbolic equations with their local domains of dependence.

In the case of a scalar conservation law, it has been shown in [8] that, for asymptotically small times, the MFE equations in one dimension carry the best least squares fit to the exact solution. Since in this particular case the local elementwise method is simply an alternative way of the implementing the MFE equations derived by Miller (without penalty functions), the properties derived from these two approaches to moving finite elements are identical.

Thus we have shown that the MFE method with linear elements consists of finding a straight line best fit to $L(U)$ in each element, together with a local mapping from the elementwise description to the nodal velocities.

4.2. *Elementwise Motion*

To emphasise the local nature of the method and the fact that the segment motion arises purely out of step 1, we now show that the motion of a local segment of the approximating picewise linear function, the "element motion" may be obtained entirely from the local projection step (4.4).

Let V_k be the velocity of the midpoint of the segment in the direction perpendicular to the segment and let θ_k be the angle between the segment and the x-axis, as shown in Fig. 4.2. Then the pairs of Eqs. (4.10) can be written in the form

$$\begin{aligned} \dot{a}_{j-1} \cos \theta_k - \dot{s}_{j-1} \sin \theta_k &= \dot{w}_{k,1} \cos \theta_k \\ \dot{a}_j \cos \theta_k - \dot{s}_j \sin \theta_k &= \dot{w}_{k,2} \cos \theta_k \end{aligned} \tag{4.14}$$

for each k, where nodes $j-1, j$ are the ends of element k, as in Fig. 4.1c. The left-hand sides of (4.14) are the velocities V_{j-1}, V_j (due to the motion of the single element k only) of the ends of the element k at right angles to the element (see Fig. 4.2). (The full velocity of a node will be a combination of two such element end velocities from adjacent segments.) Since

$$m_k = \frac{a_j - a_{j-1}}{s_j - s_{j-1}} = \tan \theta_k \tag{4.15}$$

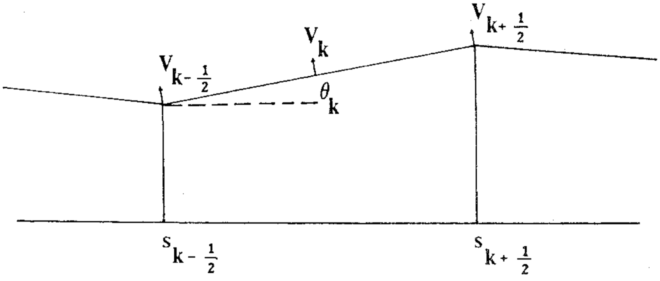

FIG. 4.2. Elementwise motions.

then these end velocities may be written as a vector in the form

$$\begin{bmatrix} V_{j-1} \\ V_j \end{bmatrix} = \begin{bmatrix} \dot{w}_{k,1} \\ \dot{w}_{k,2} \end{bmatrix} \cos\theta_k = \frac{1}{\sqrt{(1+m_k^2)}} C_k^{-1} \begin{bmatrix} b_{k,1} \\ b_{k,2} \end{bmatrix} = \frac{1}{\sqrt{(1+m_k^2)}} C_k^{-1} \begin{bmatrix} \langle \varphi_{k,1}, L(U) \rangle \\ \langle \varphi_{k,2}, L(U) \rangle \end{bmatrix} \tag{4.16}$$

using (4.6), (4.5), and (4.7). Since $[1, 1]^{\mathrm{T}}$ is an eigenvector of the symmetric matrix C_k with eigenvalue $\frac{1}{2}\Delta s_k$, we obtain

$$[1, 1] \begin{bmatrix} V_{j-1} \\ V_j \end{bmatrix} = \frac{1}{\sqrt{(1+m_k^2)}} \frac{2}{\Delta s_k} [1, 1] \begin{bmatrix} \langle \varphi_{k,1}, L(U) \rangle \\ \langle \varphi_{k,2}, L(U) \rangle \end{bmatrix}. \tag{4.17}$$

Hence

$$V_k = \frac{1}{2}(V_{j-1} + V_j) = \frac{1}{\sqrt{(1+m_k^2)}} \frac{1}{\Delta s_k} \langle 1, L(U) \rangle = \frac{1}{\Delta s_k \sqrt{(1+m_k^2)}} \int_{s_{j-1}}^{s_j} L(U)\, dx. \tag{4.18}$$

Subtracting pairs of Eqs. (4.10) we obtain another important result, namely,

$$\dot{a}_j - \dot{a}_{j-1} - m_k(\dot{s}_j - \dot{s}_{j-1}) = \dot{w}_{k,2} - \dot{w}_{,1} \tag{4.19}$$

or, using (4.15),

$$\frac{dm_k}{dt} = \frac{1}{\Delta s_k} [-1, 1] \begin{bmatrix} \dot{w}_{k,1} \\ \dot{w}_{k,2} \end{bmatrix}. \tag{4.20}$$

Since $[-1, 1]$ is also an eigenvector of C_k with eigenvalue $\frac{1}{6}\Delta s_k$, we find

$$\frac{dm_k}{dt} = \frac{6}{[\Delta s_k]^2} [-1, 1] \begin{bmatrix} \langle \varphi_{k,1}, L(U) \rangle \\ \langle \varphi_{k,2}, L(U) \rangle \end{bmatrix} = \frac{6}{[\Delta s_k]^2} \int_{s_{j-1}}^{s_j} (\varphi_{k,2} - \varphi_{k,1})\, L(U)\, dx \tag{4.21}$$

or

$$\frac{d\theta_k}{dt}=\frac{6}{[\Delta s_k]^2(1+m_k^2)}\int_{s_{j-1}}^{s_j}(\varphi_{k,2}-\varphi_{k,1})\,L(U)\,dx. \tag{4.22}$$

Thus both the velocity of the midpoint of the straight line segment of the solution between the nodes $j-1$ and j and the angular speed of the segment are found from the projection step 1 above. Each segment can be tracked in this way although, since s_{j-1} and s_j appear in both (4.18) and (4.22), the nodal positions must be computed alongside. As the segments move, the intersections (nodes) also move, giving the nodal velocities. The segments have lengths that vary with time. The movement of a node is thus the locus of the intersection of adjacent elements (see Fig. 4.3).

4.3. *Conservation Laws*

In the special case of a scalar conservation law

$$u_t+f(u)_x=0$$

we can deduce some particularly useful results. Now $L(U)=-f(U)_x$ so that (4.18) becomes

$$\begin{aligned}V_k&=-\frac{1}{\Delta s_k\sqrt{(1+m_k^2)}}\int_{s_{j-1}}^{s_j}f(U)_x\,dx\\&=-\frac{\Delta f_k}{\Delta s_k}\cos\theta_k=-\frac{\Delta f_k}{\Delta U_k}\sin\theta_k,\end{aligned} \tag{4.23}$$

where

and

$$\begin{aligned}\Delta f_k&=f(a_j)-f(a_{j-1})\\ \Delta U_k&=a_j-a_{j-1}.\end{aligned} \tag{4.24}$$

Thus we have the important result that the speed of the mid-point of the segment in Fig. 4.2 in the direction normal to the segment is consistent with the local average wave speed $\Delta f_k/\Delta U_k$ in the element.

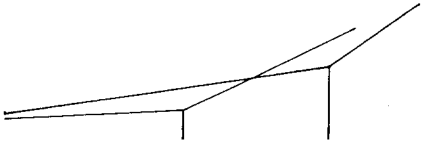

FIG. 4.3. Nodal motions as intersections of element segments.

Further, (4.21) becomes

$$\frac{dm_k}{dt}=\frac{-6}{[\Delta s_k]^2}\int_{s_{j-1}}^{s_j}(\varphi_{k,2}-\varphi_{k,1})f(U)_x\,dx$$

$$=\frac{12}{[\Delta s_k]^2}(\hat{f}-\bar{f}) \tag{4.25}$$

$$=f_{xx}(\eta_k)=m_k^2 f''(\eta_k), \tag{4.26}$$

using integration by parts and the mean value theorem, where

$$\begin{aligned}\hat{f}&=\frac{1}{\Delta s_k}\int_{s_{j-1}}^{s_j} f(U)\,dx\\ \bar{f}&=\tfrac{1}{2}((f(U))_j+(f(U))_{j-1})\end{aligned} \tag{4.27}$$

and $\eta_k \in (s_{j-1}, s_j)$. Hence we have the result that the rate of change of the slope of the solution in an element is equal to the second space derivative of the flux function. In other words, the solution segment rotates in response to the local convexity of f. Another form of this result, using (4.22), is

$$\frac{d\theta_k}{dt}=\frac{m_k^2}{1+m_k^2}f''(\eta_k). \tag{4.28}$$

The results (4.23) and (4.26) show clearly the element behaviour in terms of the flux function.

4.4. *Parallelism*

As noted elsewhere, the method may break down if either of the matrices C_k of (4.6) or M_j of (4.12) are singular. In the next section we deal with singularity of C_k. Singularity of M_j corresponds to collinearity of nodes, known by the term parallelism. In one dimension this occurs when

$$m_k=m_{k+1} \tag{4.29}$$

in (4.12).

In the present approach we can give explicit details of the algorithm used in dealing with parallelism. In such cases we obtain inconsistent solutions of the pairs of Eqs. (4.10). As a result we can no longer solve purely locally. The remedy that we have used, as in Refs. [3, 4], is to temporarily fix any parallel nodes, solve over a patch consisting of these nodes and their neighbours, and relocate the parallel nodes in some averaged way. An alternative approach [5] is to impose a local average node velocity. We illustrate here the implications of our approach for a single parallel node.

It is convenient to return to the α, β basis by combining Eqs. (4.4) in staggered pairs. Then, in the event of parallelism at a single node, we retain the combination of Eqs. (4.10) corresponding to the test function α_j and replace the second (corresponding to β_j) by

$$\dot{s}_j^* = 0, \tag{4.30}$$

where the * refers to the special solution with the node (temporarily) fixed. This gives

$$\begin{aligned}\tfrac{1}{6}(\Delta s_k)\, w_{k,1} + \tfrac{1}{3}(\Delta s_k)\, w_{k,2} + \tfrac{1}{3}(\Delta s_{k+1})\, w_{k+1,1} + \tfrac{1}{6}(\Delta s_{k+1})\, w_{k+1,2}& \\ = b_{k,2} + b_{k+1,1}.&\end{aligned} \tag{4.31}$$

Since now $\dot{s}_j^* = 0$ we have, from (3.9), that $\dot{a}_j = w_{k,2} = w_{k+1,1}$ in the reduced problem which yields

$$\begin{aligned}\dot{a}_j^* &= \frac{b_{k,2} + b_{k+1,1} - (1/6)\{(\Delta s_k)\, w_{k,1} + (\Delta s_{k+1})\, w_{k+1,2}\}}{(1/3)(\Delta s_k + \Delta s_{k+1})} \\ \dot{s}_j^* &= 0\end{aligned} \tag{4.32}$$

as the solution for the modified system with the parallel node fixed.

The null space of the singular matrix M_j is spanned by the vector $[m, 1]^{\mathrm{T}}$ (where $m_k = m_{k+1} = m$) and an appropriate multiple of this vector may be added to satisfy an externally imposed averaged velocity or position as in Refs. [3, 4]. If adjacent nodes become parallel simultaneously, a number of equations of type (4.31) will occur and it will be necessary to solve a tri-diagonal system.

In a truly transient problem, it is our experience that parallelism is a rather rare occurrence and is usually associated with the curvature of the solution changing sign. Even in regions which appear parallel to the eye, for example, in a very steep propagating front, the parallelism algorithm is rarely called upon in practice. When the initial data contains regions of linear or constant values of the variables, our technique will essentially reduce to fixed finite elements in these regions until there is sufficient curvature to justify the use of MFE. If regions of "flat" steady state solution are reached, the need for moving nodes disappears and the algorithm reverts to FFE in these regions.

4.5. *Time Stepping and Node Overtaking*

The MFE method is semi-discrete and gives rise to ordinary differential equations in time which require integration to obtain the full solution.

It has been found that the simple Euler explicit forward difference method is sufficient in the examples tried so far in [3, 4, 13]. In no case do implicit methods give any advantage. In one-dimension this method gives

$$\begin{pmatrix} a_j^{n+1} \\ s_j^{n+1} \end{pmatrix} = \mathbf{y}_j^{n+1} = \mathbf{y}_j^n + \Delta t(M_j^{-1}\mathbf{w}_j)^n = \begin{pmatrix} a_j^n \\ s_j^n \end{pmatrix} + \Delta t(M_j^{-1})^n \begin{pmatrix} w_{k,2} \\ w_{k+1,1} \end{pmatrix}^n \tag{4.33}$$

(see (4.10)).

As far as the choice of Δt is concerned, we still require an accuracy criterion. Algorithms for accuracy are not well developed but in view of the simplicity of the method we can afford to be generous in taking a trial and error approach. A possible algorithm compares the result of one MFE step with that of two half steps and continues halving the step until the difference between the two results is acceptable (see [4]).

A major difficulty with time stepping is that the nodes may overtake one another if Δt is not small enough, and this gives an absolute restriction on the time step. In one dimension this restriction is in effect that Δt should not be greater than the smallest time $(\Delta t)_0$ taken for any node to catch up with its neighbour. This is easily calculated if Euler time stepping is used. However, accuracy is also being lost in the time stepping and so Δt should not be too large.

In problems whose solution is expected to be smooth we may expect nodes to merge when they overtake if Δt is small enough. Because of time inaccuracy this may not occur in practice and we have found that a practical time step is obtained by taking a fixed fraction of the time step $(\Delta t)_0$. For hyperbolic problems which admit shocks, however, we anticipate the formation of discontinuities and can take advantage of node overtaking to model shocks in an effective way, a technique which we discuss in the next subsection.

If it is not known in advance whether a shock is forming or not, we can test the slope in the element whose length is tending to zero. If this remains small then there should be no shock formation.

4.6. *Shock Formation*

As the separation of nodes tends to zero, the element segment becomes vertical (parallel to the U-axis) and from (4.6) with $\theta_k \to \pi/2$ the normal velocity of the mid-point of the segment tends smoothly to the shock speed, at least in the semi-discrete case. The shock speed may then be "frozen," i.e., imposed on both nodes of the shocked element. This acts as an internal boundary condition and the solution on the adjacent elements to left and right may proceed separately. For more details see Ref. [3]. The procedure is also feasible when nodes run into shocks or when shocks overtake shocks, in which cases a node can be deleted.

From Eq. (4.28) we see that the angular speed of the segment is non-zero when the shock forms $(m_k \to \infty)$ so that there is a change of state at this instant. It follows from (4.28) and the way in which the shock speed is calculated in the MFE method that this part of the method satisfies the well-known geometric entropy inequality of Oleinik ([11]). Also, from (4.28), this is seen to be true for expansions.

Boundary conditions may be imposed locally on the elements adjacent to the boundaries, as in Ref. [4]. In the case of a Neumann-type condition at the end $j=0$ we can use

$$\dot{s}_0 = 0 \tag{4.34}$$

which, in conjunction with the second of (4.14), gives

$$\dot{a}_0 = \dot{w}_{1,1} \tag{4.35}$$

and hence the motion of the boundary node.

If the boundary condition at the end $j=0$ is Dirichlet then because we cannot impose both

$$\dot{s}_0 = 0, \qquad \dot{a}_0 = 0 \tag{4.36}$$

simultaneously and preserve the local projection, a special constrained projection must be carried out in the end element. The result is that

$$\dot{w}_{1,1} = 0, \qquad \dot{w}_{1,2} = 3b_{1,2}/(s_1 - s_0), \tag{4.37}$$

where $b_{1,2}$ is the second of (4.7) for the end element. This is consistent with (4.36).

We have seen that in the case of the iniviscid Burgers' equation the nodes move along characteristics, while for the general scalar hyperbolic law the least squares best fit to the exact solution is carried asymptotically for small time steps. For larger time steps the path pursued by the MFE method is not precisely that described by the characteristics nor by an L_2 best fit to the solution, but remains close to both of these paths.

5. Discussion

In this section we discuss generalisations of the foregoing and also consider some specific issues which have not so far been fully developed.

5.1. *Higher Order Operators*

The analysis of Section 2 applies strictly only when the piecewise linear approximation U lies in the domain space of the spatial operator L. Thus even for second-order operators (such as $L(u) = u_{xx}$), $L(U)$ exists only in the sense of distributions. (For third and higher order operators we require higher order approximation spaces S_α, $S_{\alpha\beta}$ as in the FFE method.) However, the unifying description of Section 2 can be applied if we consider U to be "smoothed" arbitrarily by some smoothing operator S. Thus we shall interpret the projection of $L(U)$ into S_ϕ as

$$\lim_{S\to I} PL(S(U)), \tag{5.1}$$

where I is the identity operator. Miller (see [2]) has proved that this limit is independent of δ in the case where S denotes δ-mollification. Mueller [12] has also implicitly used such an arbitrary smoothing in the evaluation of inner products involving second derivaties—see also [13].

An alternative approach suggested by Morton [14] is to use a "recovery" procedure where U is replaced by a smoother function locally (see [12]). This is also consistent with the formalism of (5.1).

We leave until Part II detailed descriptions of particular smoothing and recovery procedures which have been used for second-order operators.

5.2. *Higher Order Basis Functions*

Assume an approximate solution of the form (1.2) where the finite element basis functions α_j are of arbitrary order. For each j, and each element k, consider the set of functions $\bar{\alpha}_{j,k}$ which is the restriction of the α_j to the single element k; that is, a restricted function $\bar{\alpha}_{j,k}$ coincides with α_j on the element k, but is zero elsewhere. Each such restricted function is an element basis function, denoted by $\varphi_{k,j}$, say, and the space $\bar{S}_\phi$ is the span of all such element basis functions.

For the Galerkin FFE method

$$U_t = \sum_j \dot{a}_j \alpha_j \in S_\alpha \subset \bar{S}_\phi, \tag{5.2}$$

by construction. For the MFE method we correspondingly have

$$U_t = \sum \dot{a}_j \alpha_j + \dot{\mathbf{s}}_j \cdot \boldsymbol{\beta}_j, \tag{5.3}$$

where for isoparametric elements (as in the linear case)

$$\boldsymbol{\beta}_j = -(\nabla U)\,\alpha_j, \tag{5.4}$$

using the results of [7]. We write $S_{\alpha\beta}$ to be the span of all α_j, $\boldsymbol{\beta}_j$ and note that, for α_j of higher order than piecewise linear, $S_{\alpha\beta} \subseteq \bar{S}_\phi$ in general, since ∇U is not piecewise constant and $\boldsymbol{\beta}_j \notin \bar{S}_\phi$, in general. However, as in the piecewise linear case, there exists a rectangular matrix M (not now necessarily constant) satisfying

$$\boldsymbol{\alpha} = M^{\mathrm{T}} \boldsymbol{\varphi}, \tag{5.5}$$

where $\boldsymbol{\alpha}$ is the α_j, $\boldsymbol{\beta}_j$ basis for $S_{\alpha\beta}$ (or the basis of α_j's for S_α), and $\boldsymbol{\varphi}$ the $\bar{\varphi}_{k,j}$ basis for $\bar{S}_\phi$.

For Galerkin FFE, the matrix M is constant and is in fact the boolean assembly matrix (see [10]). However, for MFE with higher order α_j, the matrix M depends on the spatial variable. While for both methods we may project $L(U)$ into $\bar{S}_\phi$ to obtain

$$PL(U) = \boldsymbol{\varphi}^{\mathrm{T}}\dot{\mathbf{w}}, \tag{5.6}$$

as in (3.6), the second projection (3.8) differs qualitatively in the case of MFE.

For the Galerkin FFE we have

$$M^{\mathrm{T}}\langle\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathrm{T}}\rangle M\dot{\mathbf{y}} = M^{\mathrm{T}}CM\dot{\mathbf{y}} = \mathbf{g} \tag{5.7}$$

as in the linear case ((3.9), (3.10)), but in the MFE case with higher order isoparametric elements we obtain

$$\langle M^{\mathrm{T}}\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathrm{T}}M\rangle\dot{\mathbf{y}} = \mathbf{g}, \tag{5.8}$$

where M^{T} is not now constant and cannot be taken out of the inner product. Hence for MFE with higher order isoparametric elements the simple decomposition properties in Ref. [3] are lost. For certain subparametric elements (in particular, the quadratic element in 1D) our analysis remains valid.

The use of high order subparametric and isoparametric elements for MFE is the subject of current research and will be reported on at a later date.

5.3. *Higher Dimensions*

The extension of the theory in Section 4 to higher dimensions is essentially covered in Sections 1–3 but there are two additional points worth making here.

In higher dimensions the projection of $L(U)$ into the space S_ϕ leads to Eq. (2.3) as before, with, for example, in the two-dimensional case,

$$C_k = \tfrac{1}{12}\Delta A_k \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \tag{5.9}$$

where ΔA_k is the area of the element k. However, $U_t \in S_{\alpha\beta}$ and, as pointed out in Section 2, $S_\phi \not\equiv S_{\alpha\beta}$ in higher dimensions, in general. Therefore, to obtain the MFE equations a further projection is necessary, which takes the form of a least squares minimisation with weight $C^{1/2}$ (see (3.11)). (In one dimension the $C^{1/2}$ weight is of no consequence since all the matrices are square and the $C^{1/2}$ factors out.)

A significant effect of the $C^{1/2}$ weight is to link the nodal velocities together, turning the method from a local one into a global one. This is a crucial difference between the MFE method in one and in higher dimensions.

The local form of the method is however regained if the weight $C^{1/2}$ is replaced by the diagonal weighting matrix $\Delta^{1/2}$, where, for example, in two dimensions,

$$\Delta = \begin{bmatrix} \ddots & & 0 \\ & \Delta A_k & \\ 0 & & \ddots \end{bmatrix} \tag{5.10}$$

(c.f. 5.9). The resulting local MFE equations can be written in the form

$$M^{\mathrm{T}} \Delta M \dot{\mathbf{y}} = M^{\mathrm{T}} \Delta \dot{w} = M^{\mathrm{T}} \Delta C^{-1} b \tag{5.11}$$

(c.f. (1.15) *et. seq.* and (3.10)). This is a local method, since no coupling of nodal velocities is involved (that is, $M^{\mathrm{T}} \Delta M$ is a block diagonal matrix). It is therefore well adapted to hyperbolic problems and has been used successfully on scalar problems in two dimensions. It is also a Petrov–Galerkin method. If we write

$$C = C_0 \Delta, \tag{5.12}$$

where C_0 is the numerical part of (5.9), we see that the left-hand side of (5.11) can be written

$$M^{\mathrm{T}} C_0^{-1} C M = (C_0^{-\mathrm{T}} M)^{\mathrm{T}} C M. \tag{5.13}$$

As in Section 3, Eq. (3.9), this can be written

$$(C_0^{-\mathrm{T}} M)^{\mathrm{T}} \langle \boldsymbol{\varphi}, \boldsymbol{\varphi}^{\mathrm{T}} \rangle M = \langle \alpha', \alpha^{\mathrm{T}} \rangle, \tag{5.14}$$

where $\boldsymbol{\alpha} = M^{\mathrm{T}} \boldsymbol{\varphi}$, $\boldsymbol{\alpha}' = (C_0^{-\mathrm{T}} M)^{\mathrm{T}} \boldsymbol{\varphi} = M^{\mathrm{T}} C_0^{-1} \boldsymbol{\varphi}$. The matrix $M^{\mathrm{T}} \Delta M$ of (5.11) can therefore be obtained by a Petrov–Galerkin method with test function $\boldsymbol{\alpha}'$. In two dimensions $\boldsymbol{\alpha}'$ takes the form $M^{\mathrm{T}} \boldsymbol{\varphi}'$, where

$$\boldsymbol{\varphi}' = \begin{bmatrix} 9 & -3 & -3 \\ -3 & 9 & -3 \\ -3 & -3 & 9 \end{bmatrix} \boldsymbol{\varphi}.. \tag{5.15}$$

The second point concerns shock modelling. For the two-dimensional equation

$$u_t + f'(u)\, u_x + g'(u)\, u_y = 0 \tag{5.16}$$

with $f'(u) = au$, $g'(u) = bu$, the operator $L(u) = -auu_x - buu_y$ belongs (exceptionally) to $S_{\alpha\beta}$. For this operator, therefore the "element motion" properties of Section 4 can be derived. In summary, these are that the velocity component of the corners of each triangular segment of the solution, at right angles to the segment, are determined completely from the local projection step, and that the corresponding component of the velocity of the centroid of the segment is consistent with the wave velocity. This is of great help in modelling shocks in two dimensions. There is also a three-dimensional analog.

For more general $f'(u)$ and $g'(u)$ in (5.16) the projection of $L(u)$ into S_ϕ will not lie in $S_{\alpha\beta}$ and the above results will no longer hold. However, the extent to which these results are degraded is related to the extent to which $f'(u)$ and $g'(u)$ *cannot* be approximated by a linear function of u in each local segment, which is a small quantity of order $(\Delta u)^2$. If we go ahead and define the local element motion velocities as before, there is an error of this order in that at each node these velocities do not derive from a single nodal velocity. However, to a good approximation the ideas can still be used.

5.4. *Systems of Partial Differential Equations*

Consider now a system of evolutionary partial differential equations

$$u_t^{(l)} - L^{(l)}(u^{(1)}, ..., u^{(\mathscr{L})}) = 0, \qquad l = 1, ..., \mathscr{L}. \tag{5.17}$$

Approximate each $u^{(l)}$ by

$$U^{(l)} = \sum a_j^{(l)} a_j^{(l)}, \tag{5.18}$$

where the sum is over piecewise linear finite element trial (basis) functions $\alpha_j^{(l)}$ which form a basis for the space $S_\alpha^{(l)}$. As before, let the restrictions of these trial functions $\alpha_j^{(l)}$ to have support on a single element be the element basis functions, denoted by $\varphi_{k,v}^{(l)}$. Let $S_\phi^{(l)}$ denote the span of all such element basis functions. In applications of the Galerkin method to systems of PDEs, it is usual to take $S_\alpha^{(l)} = S_\alpha^{(m)}$ for all l, m, that is, to take the same grid and same approximation space for each component $u^{(l)}$.

The direct extension of Sections 2 and 3 is to project $L^{(l)}$ into $S_\phi^{(l)}$ giving

$$PL^{(l)}(U^{(1)}, U^{(2)}, ..., U^{(\mathscr{L})}) = \boldsymbol{\varphi}^{(l)\mathrm{T}} \dot{\mathbf{w}}^{(l)} \tag{5.19}$$

for each l, and then to further project $PL^{(l)}$ into $S_\alpha^{(l)}$. The double projection is then equivalent to the standard Galerkin method

$$\langle U_t^{(l)} - L^{(l)}(U^{(1)}, ..., U^{(\mathscr{L})}), \alpha_i^{(l)} \rangle = 0 \qquad \text{for} \quad l = 1, ..., \mathscr{L}, \tag{5.20}$$

exactly as described above for a scalar equation. The choice of a common S_α space, of course, makes the projection of each $L^{(l)}$ into the common S_ϕ, represented by the inner products

$$\langle L^{(l)}(U^{(1)}, U^{(2)}, ..., U^{(\mathscr{L})}), \varphi_{k,v} \rangle, \tag{5.21}$$

easier to evaluate, since all quantities are defined on the same computational grid. (We write $\varphi_{k,v}$ for $\varphi_{k,v}^{(1)} = \varphi_{k,v}^{(2)} = \cdots = \varphi_{k,v}^{(\mathscr{L})}$ and S_ϕ for $S_\phi^{(1)} \equiv S_\phi^{(2)} \equiv \cdots \equiv S_\phi^{(\mathscr{L})}$.)

In extending MFE to systems of PDEs it is neither necessary nor obvious that a common approximation space S_α should be chosen for all components, though this can, of course, be done (see [15, 16]). The general extension of MFE to systems of PDEs allowing different grids is as follows.

Approximate each $u^{(l)}$ by

$$U^{(l)} = \sum a_j^{(l)} \alpha_j^{(l)} \in S_\alpha^{(l)} \tag{5.22}$$

giving

$$U_t^{(l)} = \sum \{\dot{a}_j^{(l)}\alpha_j^{(l)} + \dot{\mathbf{s}}_j^{(l)} \cdot \boldsymbol{\beta}_j^{(l)}\} \in S_{\alpha\beta}^{(l)}. \tag{5.23}$$

Project each $L^{(l)}$ into $S_\phi^{(l)}$ (which corresponds to $S_\alpha^{(l)}$, $S_{\alpha\beta}^{(l)}$ as given in the scalar case above) giving

$$PL^{(l)} = \boldsymbol{\varphi}^{(l)\mathrm{T}}\dot{\mathbf{w}}^{(l)}. \tag{5.24}$$

The quadrature in the first projection represents in practice the most difficult part of the method using different grids, but it is a problem which arises in other areas and for which simple procedures have been proposed (see, for example, [17, 3]). The second projection is different in the cases when

(a) there are no common grids and

(b) some components share the same grid.

In case (a), for each l, $PL^{(l)}$ may be independently projected into $S_{\alpha\beta}^{(l)}$ exactly as in the scalar case. For (b), if $S_\alpha^{(l)} = S_\alpha^{(m)}$, say, for some $l \neq m$, then $S_{\alpha\beta}^{(l)}$ is the span of $\{\alpha_i, \boldsymbol{\beta}_i^{(l)}\}$ and $S_{\alpha\beta}^{(m)}$ is the span of $\{\alpha_i, \boldsymbol{\beta}_i^{(m)}\}$, where $\alpha_i = \alpha_i^{(l)} = \alpha_i^{(m)}$ but, because of the dependence of the $\boldsymbol{\beta}$ basis function on $U^{(l)}$, $\boldsymbol{\beta}_i^{(l)} \neq \boldsymbol{\beta}_i^{(m)}$. Thus in this case

$$S_{\alpha\beta}^{(l)} \cap S_{\alpha\beta}^{(m)} = \text{span of } \{\alpha_i\} \tag{5.25}$$

but

$$S_{\alpha\beta}^{(l)} \neq S_{\alpha\beta}^{(m)}. \tag{5.26}$$

The important difference in case (b) is that the projections of $PL^{(l)}$ into $S_{\alpha\beta}^{(l)}$ and $PL^{(m)}$ into $S_{\alpha\beta}^{(m)}$ are *not* independent, since a common grid $\dot{s}$ is required. For this reason, a constraint is required on the projection into $S_{\alpha\beta}^{(l)}$ and $S_{\alpha\beta}^{(m)}$ (see [19]), and thus the method with common grids is not so simply admitted into the present framework. Nevertheless, the formulation here is capable of providing a projection with the necessary properties, namely, by minimising

$$\|C^{1/2}(M^{lm}\dot{\mathbf{y}} - \dot{\mathbf{w}})\|_{l_2} \tag{5.27}$$

(c.f. 3.15) over $\dot{\mathbf{y}}$, where, for example, for a one-dimensional system of two equations,

$$\dot{\mathbf{y}}^{\mathrm{T}} = [\ldots; \dot{a}_j^{(1)}, \dot{a}_j^{(2)}, \dot{s}_j; \ldots] \tag{5.28}$$

and

$$M^{12} = \begin{bmatrix} 1 & 0 & -m_{\mathrm{L}}^{(1)} \\ 1 & 0 & -m_{\mathrm{R}}^{(1)} \\ 0 & 1 & -m_{\mathrm{L}}^{(2)} \\ 0 & 1 & -m_{\mathrm{R}}^{(2)} \end{bmatrix}. \tag{5.29}$$

Here m_{L} and m_{R} are the slopes in elements on either side of the node.

Generalisation to higher dimensions and conversion to the alternative local method (in which $C^{1/2}$ in (5.27) is replaced by $\Delta^{1/2}$) are straightforward.

5.5. *Boundary Conditions*

We comment on the important cases of Dirichlet and homogeneous Neumann boundary conditions: however, general boundary conditions are the subject of current research.

In terms of the Galerkin FFE method, if the value of the solution is specified in some or all of the boundary, then in effect some subset of the nodal coefficients a_j of the sum (1.2) is specified. Thus we seek U_t ((1.3)) only in a subspace S'_α of S_α whose basis is the same as that of S_α except that the basis functions α_i corresponding to specified a_i are removed. Similarly, we can model the boundary shape and boundary conditions in the MFE method by restricting U_t to lie in a subspace $S'_{\alpha\beta}$ of $S_{\alpha\beta}$. Note the additional feature here that the boundary position may move if $S'_{\alpha\beta}$ contains basis functions $\boldsymbol{\beta}_i$ which correspond to boundary nodes. This feature is of use in the solution of moving boundary problems (see [18]).

For a homogeneous Neumann condition we apply no restriction to boundary coefficients a_i. For straight sided boundaries this is easily justified using a reflection principle.

Thus the entire analysis of this paper applies verbatim, with boundary conditions having the sole effect of reducing the dimension of S_α or $S_{\alpha\beta}$.

6. Conclusion

In this paper we have emphasised projections in the theory of both moving and fixed finite element methods for evolutionary problems. This allows a unified view of the methods and, in particular, focuses attention on moving finite elements for scalar problems in one dimension as the simplest implementation. Out of this approach comes the local nature of moving finite elements and the "element motions," which explain why the method is so good at following the wave and entropy properties of scalar conservation laws, and why the special treatment of shocks described in Section 4.6 works so well.

From this core method generalisations can be made to higher dimensions, higher order operators, and systems of equations. The role of boundary conditions is also clear.

The main result of the paper is that finite element methods for evolutionary problems may be regarded as the result of two projections, one a local projection within an element, and the other a global (in general) projection which transfers element information on to the nodes. Further aspects and a number of practical applications are given in the subsequent paper in this volume.

REFERENCES

1. K. MILLER AND R. N. MILLER, *SIAM J. Numer. Anal.* **18**, 1019 (1981).
2. K. MILLER, *SIAM J. Numer. Anal.* **18**, 1033 (1981).
3. A. J. WATHEN AND M. J. BAINES, *IMA J. Numer. Anal.* **5**, 161 (1985).
4. M. J. BAINES, in *Numerical Methods for Fluid Dynamics II*, edited by K. W. Morton and M. J. Baines (Oxford Univ. Press, London, 1985), p. 1.
5. M. J. BAINES, Numerical Analysis Report 1/85, Department of Mathematics, University of Reading, 1985 (unpublished).
6. R. ALEXANDER, P. MANSELLI, AND K. MILLER, *Accad. Naz. Lincei Ser. 8*, **67**, 57 (1979).
7. D. R. LYNCH, *J. Comput. Phys.* **47**, 387 (1982).
8. K. W. MORTON, Department of Mathematics, University of Reading, private communication (1982).
9. A. J. WATHEN, *SIAM J. Numer. Anal.* **23**, No. 797 (1986).
10. A. J. WATHEN, *IMA J. Numer. Anal.* **7**, 449 (1987).
11. O. A. OLEINIK, Amer. Math. Soc. Trans. Ser. 2, **26**, 95 (1955).
12. A. MUELLER, Ph.D. thesis, University of Texas in Austin, 1983 (unpublished).
13. I. W. JOHNSON, Numerical Analysis Report 12/85, Department of Mathematics, University of Reading, 1985 (unpublished).
14. K. W. MORTON, Numerical Analysis Report 7/83, Department of Mathematics, University of Reading, 1983 (unpublished).
15. J. DJOMEHRI AND K. MILLER, Center for Pure and Applied Mathematics, University of California, Berkely, Report PAM-57, 1981 (unpublished).
16. J. DJOMEHRI, Ph.D. thesis, University of California, Berkely, 1983 (unpublished).
17. R. LOEHNER AND K. MORGAN, in *Proceedings, Conf. on Numerical Methods in Thermal Problems*, edited by K. Board (Pineridge, Swansea, 1985).
18. R. O. MOODY, Numerical Analysis Report 17/85, Department of Mathematics, University of Reading, 1985 (unpublished).
19. M. J. BAINES, Numerical Analysis Report 15/85, Department of Mathematics, University of Reading, 1985 (unpublished).