

M A T H E M A T I C S   D E P A R T M E N T

A COMPARISON BETWEEN TWO MOVING  
FINITE ELEMENT METHODS

C.P. Reeves

Numerical Analysis Report 17/89

U N I V E R S I T Y   O F   R E A D I N G

# A Comparison Between Two Moving Finite Element Methods

C.P. Reeves

Numerical Analysis Report 17/89

Department of Mathematics  
P.O. Box 220  
University of Reading  
Whiteknights  
Reading RG6 2AX  
U.K.

### Abstract

This report contains a comparison of two numerical methods for the solution of time-dependent partial differential equations in one dimension using moving finite elements. The first method considered is the well-known local MFE method. We compare this with a method which involves a Legendre transformation of the variables in the differential equation, followed by solution in the dual space, again using moving elements before transforming back into the original variables.

# 1 Introduction

The numerical methods we consider in this report are all used to solve one dimensional partial differential equations of the form

$$u_t = \mathcal{L}(u) \quad u = u(x, t)$$

where  $\mathcal{L}$  is an operator involving only spatial derivatives. We will compare two methods, local MFE and a second method also based on finite elements. The second method also uses an adaptive grid, but the equations for the description of the movement of the nodes are calculated in a different solution space. We first apply a Legendre transformation to the equations, then solve in the dual space. The method of solution is based on finite elements and uses the local element basis functions, giving a method similar to local MFE. After solving in this space, the solution is transformed back to the original physical variables.

The aim of the second approach is to produce a method with the advantages of MFE but one which is more robust. In using the Legendre transformation we hope to produce equations which are better to solve, ie. simpler equations, or those which have a more stable behaviour when solved using finite elements.

The contents of the report are as follows. After a brief description of the global MFE equations, we concentrate on the local MFE method. In this section we also rewrite the partial differential equation in the form of an integral equation and solve this by local MFE. Since there are many examples of equations of the type  $u_t = \mathcal{L}(u)$  which produced overturned solutions hence form shocks, we compare two methods which locate this position. The second approach, 'VM method' is then described before we consider the test problems. Finally the results are given followed by a brief comparison of the methods.

## 2 Method 1 : Local MFE

The equations we consider here are of the form

$$u_t = \mathcal{L}(u) \quad (1)$$

where  $\mathcal{L}$  is an operator containing only spatial  $x$  derivatives and

$$u = u(x, t). \quad (2)$$

We begin by giving a brief description of the global and local MFE methods in 1 dimension [1][2].

Let  $U$  be a piecewise linear approximation to  $u$  where  $U$  can be written as

$$U(x, t) = \sum_{j=0}^{n+1} a_j(t) \alpha_j(x, s(t)). \quad (3)$$

The  $\alpha_j$  are linear time-dependent basis functions and are defined by

$$\alpha_j = \begin{cases} \frac{x-s_{j-1}}{s_j-s_{j-1}} & s_{j-1} \leq x \leq s_j \\ \frac{s_{j+1}-x}{s_{j+1}-s_j} & s_j \leq x \leq s_{j+1} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where  $s_j$  are the nodal coordinates and  $a_j$  the nodal heights (see fig. 1). Differentiating  $U$  with respect to  $t$  gives, after some manipulation [3],

$$U_t = \sum_{j=0}^{n+1} (\dot{a}_j \alpha_j + s_j \beta_j) \quad (5)$$

where  $\beta_j$  is given by

$$\beta_j = -U_x \alpha_j. \quad (6)$$

$\beta_j$  is a basis function which has the same support as  $\alpha_j$ , but unlike  $\alpha_j$  it is discontinuous at  $s_j$  since  $U_x$  is piecewise constant (see fig. 2). To find equations for  $\dot{a}_j$  and  $\dot{s}_j$ , the  $L_2$  norm of the residual

$$U_t + \mathcal{L}(U) \quad (7)$$

is minimised over  $\dot{a}_j$  and  $\dot{s}_j$  ie.  $\mathcal{L}(U)$  is projected into the space of  $\alpha, \beta$  functions. This gives the MFE equations

$$\left. \begin{aligned} \langle \alpha_j, U_t + \mathcal{L}(U) \rangle &= 0 \\ \langle \beta_j, U_t + \mathcal{L}(U) \rangle &= 0 \end{aligned} \right\} j = 0, \dots, N+1 \quad (8)$$

where  $\langle \cdot, \cdot \rangle$  denotes the usual  $L_2$  inner product. Substituting (5) into (8) the MFE equations can then be written as a system

$$A(\mathbf{y})\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}) \quad (9)$$

where

$$\mathbf{y} = \begin{pmatrix} a_0 \\ s_0 \\ \vdots \\ a_{N+1} \\ s_{N+1} \end{pmatrix} \quad (10)$$

and  $A$  is a  $2 \times 2$  block tri-diagonal matrix, with blocks

$$A_{ij} = \begin{bmatrix} \langle \alpha_i, \alpha_j \rangle & \langle \alpha_i, \beta_j \rangle \\ \langle \beta_i, \alpha_j \rangle & \langle \beta_i, \beta_j \rangle \end{bmatrix}. \quad (11)$$

The  $\mathbf{g}$  vector is

$$\mathbf{g} = \begin{pmatrix} \langle \alpha_0, \mathcal{L}(U) \rangle \\ \langle \beta_0, \mathcal{L}(U) \rangle \\ \vdots \\ \langle \alpha_{N+1}, \mathcal{L}(U) \rangle \\ \langle \beta_{N+1}, \mathcal{L}(U) \rangle \end{pmatrix}. \quad (12)$$

Before the system can be solved, the appropriate boundary conditions for the problem must be applied, which usually results in some overwriting of entries in the system for the end nodes. This gives the so-called global MFE equation, which may be solved for  $\dot{a}, \dot{s}$  by inverting  $A$ , by for example conjugate gradients. This system can however also be written as a local method which involves only  $2 \times 2$  matrices [2]. In 1 dimension it can be shown that the local and global methods are equivalent [4]. We now describe this local method.

To write the system (9) in a local form, two local basis functions are introduced and defined by (see fig. 3.)

$$\begin{aligned}\phi_{j-\frac{1}{2}}^{(1)} &= \begin{cases} \frac{s_j-x}{s_j-s_{j-1}} & s_{j-1} \leq x \leq s_j \\ 0 & \text{otherwise} \end{cases} \\ \phi_{j-\frac{1}{2}}^{(2)} &= \begin{cases} \frac{x-s_j}{s_j-s_{j-1}} & s_{j-1} \leq x \leq s_j \\ 0 & \text{otherwise} \end{cases}\end{aligned}\quad (13)$$

These can be combined to give the original basis functions

$$\begin{aligned}\alpha_j &= \phi_{j-\frac{1}{2}}^{(2)} + \phi_{j+\frac{1}{2}}^{(1)} \\ \beta_j &= -(m_{j-\frac{1}{2}}\phi_{j-\frac{1}{2}}^{(2)} + m_{j+\frac{1}{2}}\phi_{j+\frac{1}{2}}^{(1)})\end{aligned}\quad (14)$$

where

$$m_{j-\frac{1}{2}} = \frac{a_j - a_{j-1}}{s_j - s_{j-1}},$$

ie.  $m_{j-\frac{1}{2}}$  is the gradient within the  $j - \frac{1}{2}th$  element. Wherever the original basis functions were applied they can be decomposed and the new local functions can be used. This gives in (8)

$$\begin{aligned}\langle \alpha_j, \mathcal{L}(U) \rangle &= \langle \phi_{j-\frac{1}{2}}^{(2)}, \mathcal{L}(U) \rangle + \langle \phi_{j+\frac{1}{2}}^{(1)}, \mathcal{L}(U) \rangle \\ \langle \beta_j, \mathcal{L}(U) \rangle &= -m_{j-\frac{1}{2}} \langle \phi_{j-\frac{1}{2}}^{(2)}, \mathcal{L}(U) \rangle - m_{j+\frac{1}{2}} \langle \phi_{j+\frac{1}{2}}^{(1)}, \mathcal{L}(U) \rangle.\end{aligned}\quad (15)$$

Similarly the inner products  $\langle \alpha_i, \alpha_j \rangle$  etc. in the matrix  $A$  also decompose to allow the system (9) to be written as

$$M^T C M \dot{\mathbf{y}} = M^T \mathbf{b}, \quad (16)$$

where  $M$  and  $C$  are  $2 \times 2$  block diagonal matrices staggered with respect to each other [1][2].  $M$  has entries

$$M_j = \begin{bmatrix} 1 & -m_{j-\frac{1}{2}} \\ 1 & -m_{j+\frac{1}{2}} \end{bmatrix} \quad (17)$$

and  $C$  has entries

$$C_{j-\frac{1}{2}} = \frac{(s_j - s_{j-1})}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (18)$$

with the size and form of the first and last entries being determined by the boundary conditions.  $\mathbf{b}$  is a vector of the inner products of the local basis functions with  $\mathcal{L}(U)$

$$\mathbf{b} = \begin{pmatrix} \langle \phi_{\frac{1}{2}}^{(1)}, \mathcal{L}(U) \rangle \\ \langle \phi_{\frac{1}{2}}^{(2)}, \mathcal{L}(U) \rangle \\ \langle \phi_{\frac{3}{2}}^{(1)}, \mathcal{L}(U) \rangle \\ \vdots \end{pmatrix} \quad (19)$$

and  $\mathbf{y}$  is as before. The advantage of (16) is that it can be solved in two stages, each of which involves decoupled  $2 \times 2$  systems. Firstly write (16) as

$$CM\dot{\mathbf{y}} = \mathbf{b} \quad (20)$$

(where  $M$  is assumed non-singular, ie. parallelism  $m_{j-\frac{1}{2}} = m_{j+\frac{1}{2}}$  is not present). Denoting the element  $[j-1, j]$  as  $k$ , we first solve the  $2 \times 2$  systems

$$C_k \mathbf{w}_k = \mathbf{b}_k \quad (21)$$

for

$$\mathbf{w}_k = \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \end{bmatrix} \quad (22)$$

in each element  $k$ , ie.  $\mathcal{L}(U)$  is projected into the space of  $\phi$  functions locally within each element. The  $\mathbf{w}_k$  blocks are then regrouped into vectors

$$\mathbf{w}_j = \begin{bmatrix} w_k^{(2)} \\ w_{k+1}^{(1)} \end{bmatrix} \quad (23)$$

whose entries are staggered in relation to those of (21) and in the second step we solve

$$M_j \dot{\mathbf{y}}_j = \mathbf{w}_j \quad (24)$$



for  $\dot{y}_j = \begin{bmatrix} \dot{a}_j \\ \dot{s}_j \end{bmatrix}$  at each node  $j$ . The global system has now been reduced to a series of local  $2 \times 2$  systems.

The  $a_j, s_j$  can then be found at the new time-level by using an ODE solver such as Euler's method

$$y^{n+1} = y^n + \Delta t \dot{y} \quad (25)$$

where  $n$  denotes the time-level.

In the moving element methods it is often convenient to rewrite  $U_t$  in a Lagrangian framework. This is done by defining a transformation between  $x, t$  and new independent variables  $\xi, \tau$  by

$$\begin{aligned} x &= \hat{x}(\xi, \tau) \\ u &= \hat{u}(\xi, \tau) \end{aligned} \quad t = \tau \quad (26)$$

where the partial derivatives satisfy

$$\frac{\partial u}{\partial t} = \frac{\partial \hat{u}}{\partial \tau} + \frac{\partial \hat{u}}{\partial \xi} \frac{\partial \xi}{\partial t}, \quad \frac{\partial \hat{u}}{\partial \xi} = \frac{\partial x}{\partial \xi} \frac{\partial u}{\partial x} \quad (27)$$

and hence

$$\frac{\partial u}{\partial t} = \frac{\partial \hat{u}}{\partial \tau} + \frac{\partial u}{\partial x} \frac{\partial \hat{x}}{\partial \tau}. \quad (28)$$

Replacing  $u_t$  using (28) in (1) gives a Lagrangian form

$$\frac{\partial \hat{u}}{\partial \tau} - \frac{\partial u}{\partial x} \frac{\partial \hat{x}}{\partial \tau} = \mathcal{L}(u). \quad (29)$$

Then writing

$$\dot{u} = \frac{\partial \hat{u}}{\partial \tau}, \quad \dot{x} = \frac{\partial \hat{x}}{\partial \tau}, \quad u_x = \frac{\partial u}{\partial x} \quad (30)$$

(29) becomes

$$\dot{u} - u_x \dot{x} = \mathcal{L}(u). \quad (31)$$

In the next section, we shall need the Lagrangian form (31) of (1) to derive the second method used in this report. However the local MFE method may also be obtained using this form of the equation, as follows. From (1) we have the equation  $u_t = \mathcal{L}(u)$  which may be rewritten in Lagrangian form  $\dot{u} - u_x \dot{x} = \mathcal{L}(u)$ . In the MFE approximation space, the left hand side of (31) lies in the space of piecewise linear discontinuous functions, we then in effect

carry out a projection of the right hand side  $\mathcal{L}(u)$  using the basis functions  $\phi^{(1)}, \phi^{(2)}$ , which gives (21). Then using expansions of the type (3) for both  $U$  (in terms of  $a_j$ ) and  $X$  (in terms of  $s_j$ ) we have

$$\sum_j \dot{a}_j \alpha_j - \sum_j m_k \dot{s}_j \alpha_j = \sum_k w_k^{(1)} \phi_k^{(1)} + w_k^{(2)} \phi_k^{(2)} \quad (32)$$

which can be solved for  $\dot{a}, \dot{s}$  to give (24).

Now returning to the solution of the local MFE method the initial node positions must be considered. Although the nodes move they must be placed initially, which will have an important influence on the subsequent solution. The method chosen here is equidistribution using a weight function  $(U_{0xx})^{\frac{2}{5}}$ , where  $U_0$  is the initial data [5]. This means that the nodes are distributed according to

$$\frac{\int_{s_0}^{s_N} (U_{0xx}(x))^{\frac{2}{5}} dx}{N+1} = \int_{s_j}^{s_{j+1}} (U_{0xx}(x))^{\frac{2}{5}} dx \quad (33)$$

where the  $s_j$   $j = 0, \dots, N$  are the node positions. This initially places more nodes in regions of high curvature and fewer elsewhere.

Another important point to consider, is whether the system is singular. There are two possible singularities of the system (16), parallelism and coincided nodes (element folding)[6]. Parallelism occurs when  $m_{j-\frac{1}{2}} = m_{j+\frac{1}{2}}$ , ie. when two or more consecutive elements have the same gradient, whence the  $M$  matrix becomes singular. This is dealt with by firstly finding the nodes at which parallelism occurs: the speed of the problem nodes are then constrained to be a weighted average of the node speeds at the ordinary nodes bounding this region (see fig. 4). Singularities in the  $C$  matrix, due to coincident nodes are avoided by choosing timesteps to prevent this ever happening [7].

For some operators  $\mathcal{L}$ , such as  $\mathcal{L}(u) = uu_x$  in (1), shocks will form in the analytic solution. However the solution generated by the MFE method will overturn, ie. naturally become multivalued. In these cases, the location of the the shock position must be calculated from additional jump conditions. Here we consider two different methods which calculate this position. Both involve conservation of area [8]: the first locates the shock directly while the second involves elimination of a swallowtail in the graph of the integral.

### Shock Location: Method A.

If a solution overturns so that it appears as in fig. 5 then, as is well known, the shock position will be at the point  $x_s$  such that (area A) = (area B), see fig. 6. The shock position  $x_s$ , may then be calculated directly using the bisection method.

### Shock Location: Method B.

This method locates the shock position  $x_s$  by finding the position of the self-intersection of the graph of the integral of  $U$ . (see fig. 7), which uses the same area equality principle. (N.B. The integral curve is made up of piecewise quadratic segments but we shall here approximate it by piecewise linears.) Consider two linear segments of this approximation to the integral of  $U$ ,

$$\begin{aligned} y &= p_i x + q_i \\ y &= p_j x + q_j \end{aligned} \quad i \neq j \quad (34)$$

where  $p_i, p_j, q_i, q_j$  can readily be calculated. The intersection of the segments occurs when

$$\frac{y - q_j}{p_j} = \frac{y - q_i}{p_i} \quad (35)$$

ie.  $y = \frac{q_i p_i - q_j p_i}{p_i - p_j}$  and the shock position is given by deleting the swallowtail which occurs in fig. 7, which corresponds to the overturning in fig. 6.

$$x = \frac{q_j - q_i}{p_i - p_j} \quad (36)$$

There are two main problems which occur numerically with this method:

1. the gradients of the lines can be very close, in which case the solution of (36) is ill-conditioned and  $x$  may be very large.
2. an inappropriate intersection may be found, ie. crossing of extensions (see fig. 8.)

However, these two problems can be easily dealt with. If the gradients are similar, then the intersection will occur outside the region considered for the problem. Therefore the first problem can easily be resolved. The second problem, involving inappropriate intersection of  $x$ , can be avoided by a simple test. If  $(x - s_{i-1})(s_i - x) \geq 0$  and  $(x - s_{j-1})(s_j - x) \geq 0$  the intersection

is valid, otherwise the invalid position is discarded and the next intersection calculated.

### Method 1A: Local MFE applied to integrated equation

For conservation laws, which are a special case of (1), we also consider another method based on the local MFE method. The solution is found by using the local MFE applied to the equation for the integral of  $U$ . First we transform equation (1) into an equation for the integral of  $u$  by the following argument.

$$\begin{aligned}
 u_t + f_x &= 0 \\
 \Rightarrow \int_0^x (u_t + \frac{\partial f(u)}{\partial x}) dx &= 0 \\
 \Rightarrow \frac{\partial}{\partial t} \int_0^x (u_t + \frac{\partial f(u)}{\partial x}) dx &= 0 \\
 \Rightarrow \frac{\partial}{\partial t} \int_0^x u dx + \int_0^x \frac{\partial f(u)}{\partial x} dx &= 0 \\
 \Rightarrow \frac{\partial a}{\partial t} + (f(u(x)) - f(u(0))) &= 0 \\
 \Rightarrow \frac{\partial a}{\partial t} + f(a_x) &= const
 \end{aligned}$$

where  $a = \int_0^x u dx$ .

Once the equation

$$a_t + f(a_x) = const \quad (37)$$

is obtained, we find the solution for  $a$  by using the local MFE method on this equation. When the MFE solution has been found it is then transformed back into the original variables using  $u = a_x$ . N.B. The difference between this and the previous method in this section, is that we are here applying the local MFE method to the variable  $a$  rather than  $u$ , ie.  $a = \sum \alpha_j(x, s(t)) a_j(t)$ . This gives a piecewise constant solution since  $a$  is calculated using piecewise linear approximations. However, for the presentation of the results a piecewise linear solution will be plotted using the following numerical approximations. For each element  $k$  we calculate and plot

$$\begin{aligned}
 x_{j+\frac{1}{2}} &= \frac{s_j + s_{j+1}}{2} \\
 u_{j+\frac{1}{2}} &= \frac{a_j + a_{j+1}}{s_{j+1} - s_j}
 \end{aligned} \quad (38)$$

The results are therefore plotted using different nodes than in the calculation. A piecewise linear solution is plotted so as to be seen in comparison with the solution using the other two methods.

### 3 Method 2 : VM Method

Another approach to these problems is also based on MFE and uses the Lagrangian form (31) of (1). For this method we first define a Legendre Transformation between  $u(x)$  and  $v(m)$  by [9]

$$\begin{aligned} m &= \frac{du}{dx} \\ x &= \frac{dv}{dm} \end{aligned} \tag{39}$$

for which

$$u - mx + v = 0. \tag{40}$$

Note that a straight line in  $u, x$  space is associated with a constant  $v, m$  values and vice versa. This defines the Legendre transform used, where  $u, v$  are dual Legendre functions.

From (40) we have

$$\dot{u} - \dot{m}x - m\dot{x} + \dot{v} = 0 \tag{41}$$

which, substituted into (31) gives

$$-\dot{v} + v_m \dot{m} + \mathcal{L}(mx - v) = 0 \tag{42}$$

where  $v_m = x$ . Let  $M, V$  be the piecewise constant approximations of  $m, v$ . The aim is to project  $\mathcal{L}(u)$  element by element into the same piecewise linear space as before, this time spanned by  $\{1, x\}$ . This is done since, whereas in  $X, U$  space  $X, \dot{X}, U, \dot{U}$  are piecewise linear and  $U_x$  is piecewise constant, in  $M, V$  space  $M, \dot{M}, V, \dot{V}$  are piecewise constant and  $V_m$  is piecewise linear. So following the same ideas as in of the local method, we now obtain a different set of ODE's.

First project  $\mathcal{L}(u)$  into the space spanned by  $\{1, x\}$  so that  $\mathcal{L}(u) = Ax + B$ . We calculate  $A$  and  $B$  by finding the best  $L_2$  fit of  $\mathcal{L}(u)$  by the straight line  $Ax + B$ , ie.

$$\min_{A,B} \|Ax + B - \mathcal{L}(u)\|_2. \tag{43}$$

Now assuming that  $A, B$  have been found for each element in the finite dimensional  $V, M$  space (42) becomes

$$-\dot{V} + V_m \dot{M} + AV_m + B = 0 \tag{44}$$

where  $V_m = X$ . Comparing the coefficient of  $V_m$ , we obtain the equations

$$\begin{aligned}\dot{M} &= -A \\ \dot{V} &= B.\end{aligned}\tag{5}$$

These equations may be solved using an ODE solver such as Euler or Crank-Nicolson.  $M$  and  $V$  are then calculated at each timestep, then transformed back into the original variables  $U, X$ . The method essentially reverses the two steps of the time integration and of projection.

We now review the second method, involving transformation to dual space before solving.

## 4 Description of problems

We now consider three example problems, for which we can test the methods. The problems considered are the inviscid Burgers' equation, the Buckley–Leverett equation, and the linear heat diffusion problem, all in one dimension. Although these problems may all be solved using other methods each provides some interesting feature for the methods considered here. Solutions to both the Burgers' equation and the Buckley–Leverett equation overturn unless we impose jump conditions, so that we can obtain the overturned solution and then find the shock position numerically (shock methods A and B of section 2). For the heat equation there is no overturning so it would seem less interesting; however, knowing the exact solution in this case, we can compare this with the numerical solutions. It also allows us to demonstrate that we can solve a second order equation using this method.

For Burgers' equation and the Buckley–Leverett equation a third method is also considered. This method is based on solving an integral form of each equation by the first method and allows a direct approach to the shock positioning algorithm.

We shall first consider the different methods for Burgers' equation.

### Problem 1 : Inviscid Burgers' Equation

The problem considered here is

$$u_t + uu_x = 0 \quad 0 \leq x \leq 1 \quad (46)$$

with initial conditions

$$u = \tanh(5 - 10x) \quad 0 \leq x \leq 1$$

and Dirichlet boundary conditions given by the initial data.

### Local MFE

This is the local MFE method where two systems of linear equations are solved,

$$C_k \mathbf{w}_k = \mathbf{b}_k \quad (47)$$

and

$$M_j \dot{\mathbf{y}}_j = \mathbf{w}_j.$$



We have already calculated  $C_k$  to be

$$C_k = \frac{s_j - s_{j-1}}{6} \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \quad (48)$$

and  $\mathbf{b}_k$  is given by

$$\mathbf{b}_k = \begin{pmatrix} \langle UU_x, \phi_k^{(1)} \rangle \\ \langle UU_x, \phi_k^{(2)} \rangle \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}. \quad (49)$$

From (49)  $b_1$  becomes

$$b_1 = \int_{s_{j-1}}^{s_j} \frac{1}{2} (U^2)_x \phi_k^{(1)} dx. \quad (50)$$

Using integration by parts, we obtain

$$b_1 = \frac{-U^2}{2} + \frac{1}{s_j - s_{j-1}} \int_{s_{j-1}}^{s_j} \frac{U^2}{2} dx. \quad (51)$$

The remaining integral may be calculated exactly or by Simpson's rule. (The projection is not actually needed here since  $UU_x$  is already piecewise linear but this stage is included in order to calculate  $w_1, w_2$  easily.)  $b_2$  is found using a similar method, hence the system of  $2 \times 2$  matrices  $C_k \mathbf{w}_k = \mathbf{b}_k$  can be solved. Once  $\mathbf{w}_k$  has been found for each element  $k$ , the second system  $M_j \dot{\mathbf{y}}_j = \mathbf{w}_j$  can then be obtained. This then allows us to update the solution of  $\mathbf{y}_j$  at the new time-level using Euler time-stepping.

#### Local MFE applied to integrated equation.

This is the same method as we have described in section 2. However in this case Burgers' equation (46) has been written in the form (37)

$$a_t + f(a_x) = \text{const} \quad (52)$$

where  $f(u) = \frac{u^2}{2}$ ,  $a = \int_0^x u dx$  and applying the boundary condition gives  $\text{const} = \frac{1}{2}$ . We again have to solve the system  $C_k \mathbf{w}_k = \mathbf{b}_k$  where

$$\mathbf{b}_k = \begin{pmatrix} \langle f(a_x) - \frac{1}{2}, \phi_k^{(1)} \rangle \\ \langle f(a_x) - \frac{1}{2}, \phi_k^{(2)} - k \rangle \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}. \quad (53)$$

To calculate  $b_1$  we need to evaluate the integral

$$b_1 = \int_{s_{j-1}}^{s_j} (f(a_x) - \frac{1}{2}) \phi_{j-\frac{1}{2}}^{(1)} dx. \quad (54)$$

However, since  $a$  is piecewise linear, hence  $a_x$  is piecewise constant, then  $f(a_x)$  is also piecewise constant, therefore

$$b_1 = (f(a_x) - \frac{1}{2}) \int_{s_{j-1}}^{s_j} \phi_{j-\frac{1}{2}}^{(1)} dx$$

hence

$$b_1 = \frac{(f(a_x) - \frac{1}{2})(s_j - s_{j-1})}{2} = b_2.$$

The resulting equations are then solved as in the previous method.

### VM Method

In this method the initial data in  $a_j, s_j$  space is transformed into  $v_k, m_k$  space using the equations (39)

$$\begin{aligned} m_k &= \frac{a_j - a_{j-1}}{s_j - s_{j-1}} & j = 1, \dots, N & \quad k = j - \frac{1}{2}. \\ v_k &= m_k s_j - a_j \end{aligned} \quad (55)$$

From (31) and (46) we obtain the equation

$$-\dot{v} + x\dot{m} - m(mx - v) = 0. \quad (56)$$

Note that Burgers' equation does not follow the general method as stated, since we do not need to project  $M(MX - V)$  into a piecewise linear space because it is already of the correct form. We can compare the coefficients of  $1, X$  directly to give the two equations

$$\begin{aligned} \dot{M} &= -M^2 \\ \dot{V} &= -MV. \end{aligned} \quad (57)$$

These can be solved exactly, since choosing  $t_0, v_0$  as initial conditions we obtain

$$M = \frac{1}{t - t_0} \quad \text{and} \quad V = \frac{v_0}{t - v_0}. \quad (58)$$

This means that  $M^{-1}$  and  $V^{-1}$  are linear in  $t$ , so using Euler's rule to integrate for  $M^{-1}, V^{-1}$  will give exact results.

The final update of the solution is obtained by using Euler's rule, then the original solution in  $U, X$  space is found by transforming back from  $V, M$ .

### Problem 2: Buckley-Leverett Equation.

The second problem we considered is the Buckley-Leverett equation given by

$$u_t + \left( \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2} \right)_x = 0 \quad 0 \leq x \leq 2 \quad (59)$$

with initial conditions

$$u = \frac{1}{1 + 10x} \quad 0 \leq x \leq 2$$

and Dirichlet boundary conditions given by initial data.

#### Local MFE

Using the first method, we again solve the system  $C_k \mathbf{w}_k = \mathbf{b}_k$  for each element  $k$ .  $C_k$  and  $\mathbf{w}_k$  are as before, so here we only need to calculate

$$\mathbf{b}_k = \begin{pmatrix} \left\langle \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2}, \phi_k^{(1)} \right\rangle \\ \left\langle \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2}, \phi_k^{(2)} \right\rangle \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

Hence

$$b_1 = \int_{s_{j-1}}^{s_j} \left( \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2} \right)_x \phi_{j-\frac{1}{2}}^{(1)} dx,$$

and similarly

$$b_2 = \int_{s_{j-1}}^{s_j} \left( \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2} \right)_x \phi_{j-\frac{1}{2}}^{(2)} dx,$$

which are calculated using either Simpson's rule or Gaussian quadrature. The second system  $M_j \dot{\mathbf{y}}_j = \mathbf{w}_j$  is then solved for each node before updating  $\mathbf{y}$  using forward Euler time-stepping.

### Local MFE applied to integrated equation

Rewriting the Buckley-Leverett Equation (59) in the integral form (37) with  $f(u) = \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2}$  and the constant=1, the equation then becomes

$$a_t + f(a_x) = 1 \quad (60)$$

where

$$a = \int_0^x u dx.$$

First solving the system  $C_k w_k = b_k$  for each element  $k$  where

$$b_k = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \frac{(s_j - s_{j-1})(f(a_x) - 1)}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

the system  $M_j \dot{y}_j = w_j$  is then solved and  $y$  is updated.

### VM Method

Writing (59) in Lagrangian form gives

$$\dot{u} - u_x \dot{x} + \left( \frac{u^2}{u^2 + \frac{1}{2}(1-u)^2} \right)_x = 0 \quad (61)$$

This is then projected onto the basis  $\{1, x\}$  so we get

$$\dot{u} - u_x \dot{x} + Ax + B = 0 \quad (62)$$

where  $A, B$  are found from

$$\min_{A, B} \|Ax + B - f_x\|_2 \quad (63)$$

$$\Rightarrow \int_{s_{j-1}}^{s_j} (Ax + b - f_x) dx = 0$$

$$\text{and } \int_{s_{j-1}}^{s_j} (Ax + b - f_x) x dx = 0.$$

These give

$$\frac{A}{3} [x^3]_{s_{j-1}}^{s_j} + \frac{B}{2} [x^2]_{s_{j-1}}^{s_j} = [xf]_{s_{j-1}}^{s_j} - \int_{s_{j-1}}^{s_j} f dx$$

and

$$\frac{A}{2}[x^2]_{s_{j-1}}^{s_j} + B[x]_{s_{j-1}}^{s_j} = [f]_{s_{j-1}}^{s_j}.$$

The integrals can be obtained using Gaussian quadrature or Simpson's rule, and (63) leads to a  $2 \times 2$  system to solve for  $A, B$ . Now using (3) in  $VM$  space and collecting coefficients of 1 and  $V_m$  gives

$$\begin{aligned} \dot{M} &= -A \\ \dot{V} &= B \end{aligned} \quad (64)$$

These equations may then be solved using either forward Euler or Crank-Nicolson time-stepping.

### Problem 3: Linear Heat Equation

The third problem considered is the linear heat equation.

$$u_t = u_{xx} \quad (65)$$

where  $u = u(x, t)$ , with initial data

$$u = \sin^2(\pi x) \quad 0 \leq x \leq 1$$

and zero Neumann boundary conditions. For this problem we are only considering the two methods, local MFE and the  $VM$  method.

#### Local MFE

For the local MFE method the system  $C_k w_k = b_k$  is solved first. This is the system

$$\frac{s_j - s_{j-1}}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} \int_{s_{j-1}}^{s_j} \phi_{j-\frac{1}{2}}^{(1)} u_{xx} dx \\ \int_{s_{j-1}}^{s_j} \phi_{j-\frac{1}{2}}^{(2)} u_{xx} dx \end{pmatrix}. \quad (66)$$

Using integration by parts

$$\begin{aligned} \int_{s_{j-1}}^{s_j} \phi_{j-\frac{1}{2}}^{(1)} u_{xx} dx &= [\phi_{j-\frac{1}{2}}^{(1)} u_x]_{s_{j-1}}^{s_j} - \int_{s_{j-1}}^{s_j} \frac{\partial \phi^{(1)}}{\partial x} u_x dx \\ \int_{s_{j-1}}^{s_j} \phi_{j-\frac{1}{2}}^{(1)} u_{xx} dx &= -u_x|_{s_{j-1}} + \frac{[u]_{s_{j-1}}^{s_j}}{(s_j - s_{j-1})}. \end{aligned}$$

The last term is an approximation to  $u_x$  over the interval  $[s_{j-1}, s_j]$ , so

$$\int_{s_{j-1}}^{s_j} \phi_{j-\frac{1}{2}}^{(1)} u_{xx} dx = -u_x|_{s_{j-1}} + u_x|_{\text{in } [s_{j-1}, s_j]}.$$

Now since  $u_x$  is the gradient (see fig. 9), and there are no values of  $u_x$  at the nodes. We will take  $u_x|_{s_{j-1}}$  to be the average of the values on each side. Using the notation that  $u_x = m_k$  in element  $k$ , the right hand side becomes

$$\frac{1}{2} \begin{pmatrix} m_k - m_{k-1} \\ m_{k+1} - m_k \end{pmatrix}.$$

This gives the system

$$\frac{(s_j - s_{j-1})}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} m_k - m_{k-1} \\ m_{k+1} - m_k \end{pmatrix} \quad (67)$$

which is solved for  $w_1, w_2$  for each element  $k$ . Hence, as long as  $s_j \neq s_{j-1}$ ,

$$w_1 = \frac{1}{(s_j - s_{j-1})} (-2m_{k-1} + 3m_k - m_{k+1}) \quad (68)$$

$$w_2 = \frac{1}{(s_j - s_{j-1})} (m_{k-1} - 3m_k + m_{k+1}). \quad (69)$$

We now need to solve the system  $M_j \dot{y}_j = w_j$  for  $\dot{y}$ , where  $M_j$  and  $w_j$  are known.  $y$  is updated using forward Euler time-stepping.

### VM Method

For the second method, we write equation (65) in Lagrangian form,

$$\dot{u} - u\dot{x} = u_{xx} \quad (70)$$

then it can be projected into the space spanned by either  $\{1, x\}$  or  $\{\phi^{(1)}, \phi^{(2)}\}$  hence (70) becomes

$$\dot{u} - u_x \dot{x} = Ax + B \quad (71)$$

or

$$\dot{u} - u_x \dot{x} = w_1 \phi^{(1)} + w_2 \phi^{(2)}. \quad (72)$$

We used the local basis functions as this seemed to give more stable numerical results. This means that we have to find

$$\min_{w_1, w_2} \|w_1 \phi^{(1)} + w_2 \phi^{(2)} - u_{xx}\|_2. \quad (73)$$

Solving this gives the same system as we obtained for the local method so that  $w_1, w_2$  are the same as in equations (68,69). Now rewriting (72) in  $MV$  notation we obtain the equation

$$-\dot{v} + v_m \dot{m} = w_1 \phi^{(1)} + w_2 \phi^{(2)} \quad (74)$$

which we can rewrite as

$$-\dot{v} + v_m \dot{m} = Av_m + B \quad (75)$$

where

$$A = \frac{-w_1 + w_2}{s_j - s_{j-1}} \quad (76)$$

and

$$B = \frac{w_1 s_j - w_2 s_{j-1}}{s_j - s_{j-1}}. \quad (77)$$

Hence comparing the coefficients of 1 and  $v_m$  we get the equations

$$-\dot{V} = B \quad (78)$$

$$\dot{M} = A. \quad (79)$$

So for equation (79) we get

$$\dot{M}_k = \frac{3M_{k-1} - 6M_k + 3M_{k+1}}{(s_j - s_{j-1})^2}. \quad (80)$$

This may then be solved using either an explicit or an implicit method. Writing this equation so that is solved using Euler time-stepping, gives

$$\frac{M_k^{n+1} - M_k^n}{\Delta t} = \frac{3M_{k-1} - 6M_k + 3M_{k+1}}{(s_j - s_{j-1})^2} \quad (81)$$

$\dot{V}$  is solved similarly. The problem given has Neumann boundary conditions so this gives  $M_0 = -M_1$  and  $M_{N+2} = -M_{N+1}$  for the end elements. These are applied to the equations above at the end equations.

## 5 Results and Discussion

In this section we will consider the results of both methods applied to the problems described in the previous section. Note that method 1 (local MFE) is applied twice to problems 1 and 2. It is first applied to the original problem and then applied to the same problem this time written in an integrated form. For problems 1 and 2 we also consider the two shock methods which are only applied when the equation is being solved using Method 1. Since the equation no longer being solved in conserved variables. The results are given in figs. 10-21.

Comparing figs. (10,12),(15,17) and figs. (20,21) respectively we can see little difference between the two sets of results. For problems 1 and 2 it is not possible to obtain a closed form solution, this means that we cannot easily calculate the error for the the two methods. However problem 3. the heat equation can be solved analytically so that we can calculate the  $L_2$  error. The  $L_2$  error between the exact and approximate solutions is given below for both methods. The timestep is 0.0001 for both methods and the error is printed every 100 time-steps.

The equation  $u_t = u_{xx}$  with Neumann boundary conditions and initial data  $u = \sin^2(\pi x)$  has solution

$$u(x, t) = \frac{1}{2}(1 - e^{-4\pi^2 t} \cos(2\pi x))$$

hence the  $L_2$  error is

$$E_2 = \sqrt{\int_0^1 (U - u)^2 dx}$$

where  $U$  is the approximate solution in each element.



Number of timesteps	Method 1	Method 2
0	$0.147 \times 10^{-3}$	$0.197 \times 10^{-3}$
100	$0.261 \times 10^{-3}$	$0.255 \times 10^{-3}$
200	$0.120 \times 10^{-3}$	$0.118 \times 10^{-3}$
300	$0.547 \times 10^{-4}$	$0.539 \times 10^{-4}$
400	$0.256 \times 10^{-4}$	$0.252 \times 10^{-4}$
500	$0.127 \times 10^{-4}$	$0.124 \times 10^{-4}$
600	$0.676 \times 10^{-5}$	$0.663 \times 10^{-5}$
700	$0.418 \times 10^{-5}$	$0.409 \times 10^{-5}$
800	$0.306 \times 10^{-5}$	$0.299 \times 10^{-5}$

$L_2$  errors for the heat equation.

From these results we can see that the  $L_2$  errors for the two methods are very similar. This implies that in terms of accuracy there is little to choose between the two methods.

The initial results for all the problems are calculated using 21 nodes. The timesteps used for each problem are the same for both methods. We will instead compare the cpu time (given in seconds) for each of the methods. We first compare the cpu time for the two methods, and in each case for similar problems the cpu times are approximately the same. We now consider the cpu time for the two shock location methods.

	Problem 1		Problem 2	
	t=0.01	t=0.0001	t=0.01	t=0.001
Method A	1.60	15.60	0.43	3.10
Method B	0.24	8.90	0.24	1.10

Cpu time for the shock location methods.

The results and cpu timings show that for a given timestep there is little to choose between the two methods, although between the shock recovery procedures, Method A clearly has the advantage.

Another consideration apart from accuracy and cost is the robustness of the method. To compare the robustness of the two methods (1 and 2), both the timestep and the number of nodes in the calculations have been varied. The results are given (\*-method stable,●-method unstable).

#### Burgers Equation

timestep	number of nodes								
	Method 1			Method 1A			Method 2		
	21	40	80	21	40	80	21	40	80
0.8	*	*	*	*	*	*	*	*	*
0.1	*	*	*	*	*	*	*	*	*
0.01	*	*	*	*	*	*	*	*	*

#### Buckley-Leverett Equation

timestep	number of nodes								
	Method 1			Method 1A			Method 2		
	21	40	80	21	40	80	21	40	80
0.1	*	*	*	*	*	*	●	●	●
0.01	*	*	*	*	*	*	●	●	●
0.001	*	*	*	*	*	*	*	●	●

#### Heat Equation

timestep	number of nodes					
	Method 1			Method 2		
	21	40	80	21	40	80
0.001	●	●	●	●	●	●
0.0001	*	●	●	*	●	●
0.00001	*	*	*	*	*	*
0.000001	*	*	*	*	*	*

The results on stability show that for the problems discussed here there is little to choose between the two methods. However for problem 2, there is a clear difference between the sets of results. This could imply that method

2 is more problem dependent than method 1, but we would have to consider more problems before reaching any conclusion.

From the results given it is clear that we have not found any real advantage to either method. The main advantages we have found are problem dependent in the sense that it may be easier to implement one of the methods than the other for a given set of equations.

## 6 References

- [1] Baines, M.J. and Wathen, A.J. (1985)  
Moving Finite Element Modelling of Compressible Flow. Numerical Analysis Report. 4/85. University of Reading.
- [2] Sweby, P.K. (1987)  
Some observations on the moving finite element method and its implementation. Numerical Analysis Report. 13/87, University of Reading.
- [3] Baines, M.J. (1985)  
Local Moving Finite Elements. Numerical Analysis Report. 9/85. University of Reading.
- [4] Baines, M.J. and Wathen, A.J. (1988)  
Moving Finite Element Methods for Evolutionary Problems. I. Theory. Journal of Computational Physics. Vol 79. No. 2.
- [5] Sweby, P.K. (1987)  
Data-dependent grids. Numerical Analysis Report. 7/87, University of Reading.
- [6] Wathen, A.J. and Baines, M.J. (1985)  
On the Structure of the Moving Finite Element Equations. I.M.A. Journal of Numerical Analysis (1985)5,161-182
- [7] Johnson, I.W., Wathen, A.J. and Baines, M.J. (1988)  
Moving Finite Methods for Evolutionary Problems. II. Applications. Journal of Computational Physics. Vol. 79. No. 2.
- [8] Brenner, Y. (1984)  
Averaged Multivalued Solutions for Scalar Conservation Laws. SIAM J. Numer. Anal., Vol. 21, No. 6, December 1984.
- [9] Baines, M.J. (1989)  
Moving Finite Elements and Approximate Legendre Transformations. Numerical Analysis Report. 5/89, University of Reading.

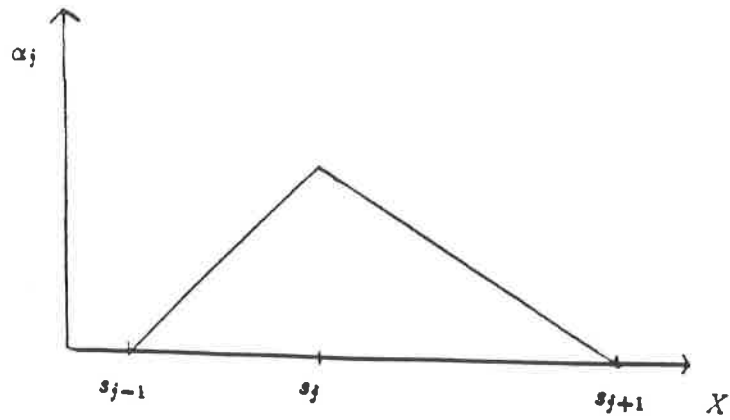


Figure 1:  $\alpha_j$ -basis function

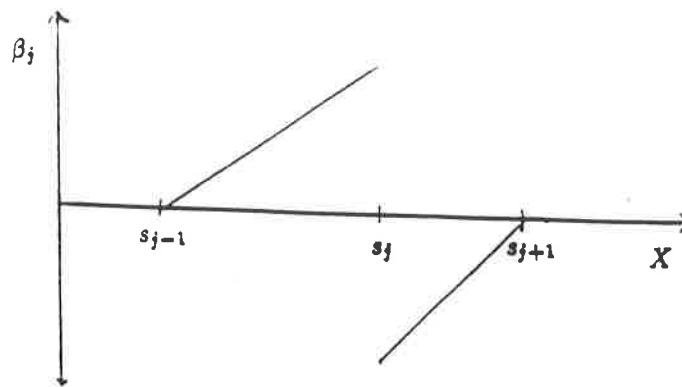


Figure 2:  $\beta_j$ -basis function.

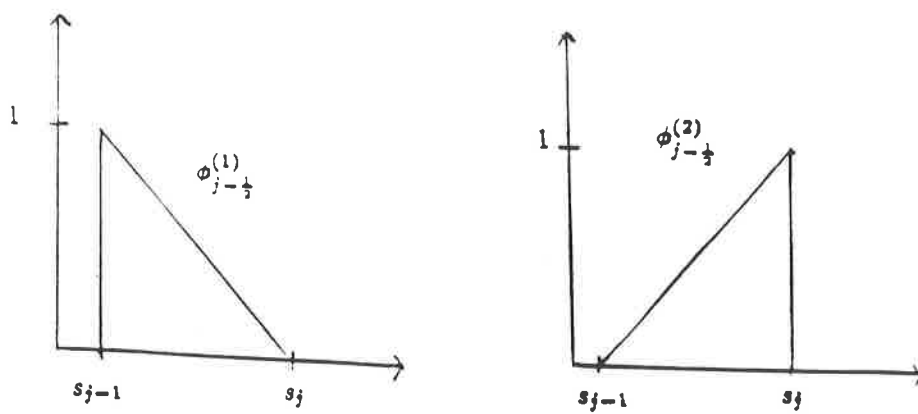


Figure 3: local basis functions

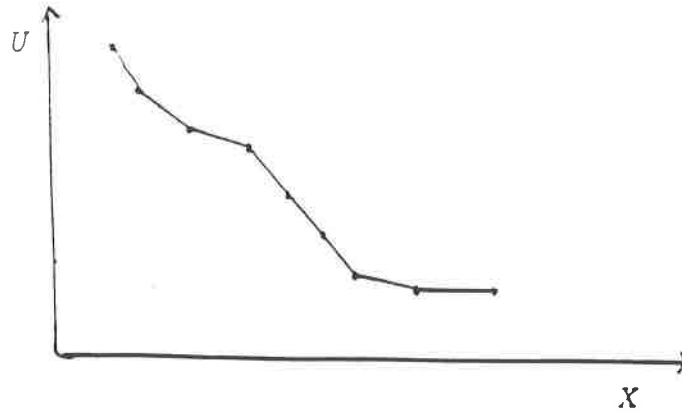


Figure 4: parallelism

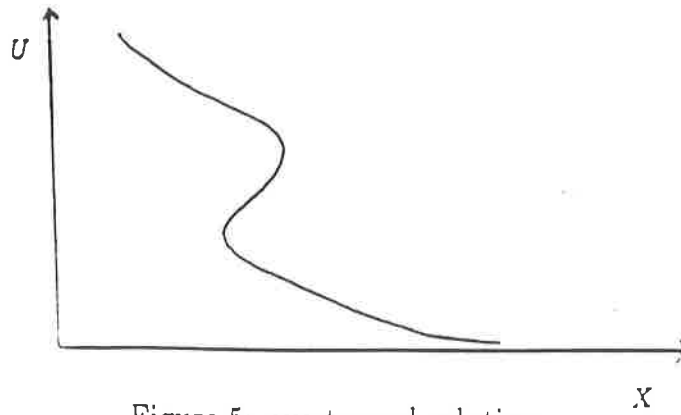


Figure 5: overturned solution

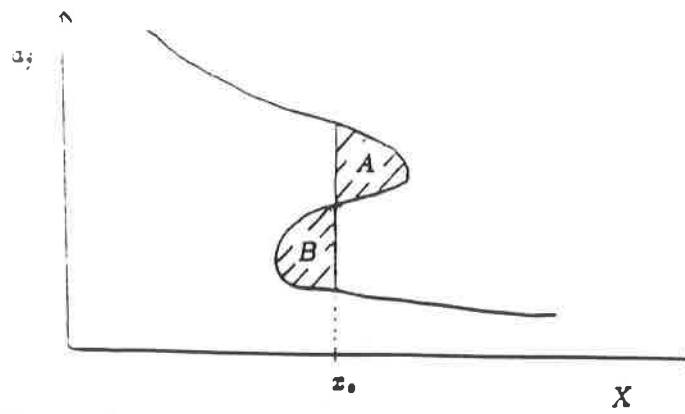


Figure 6: overturned solution with shock position

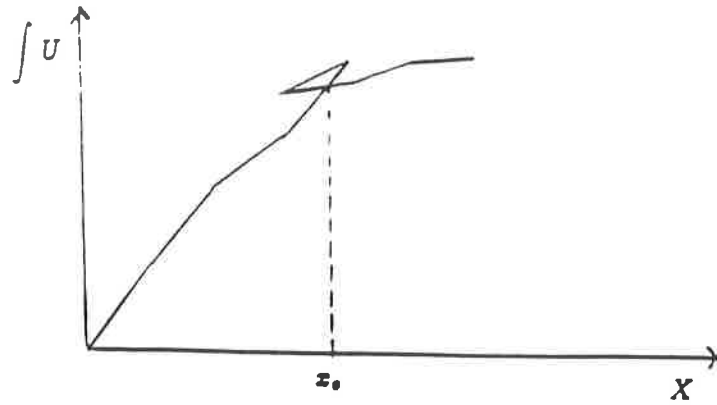


Figure 7: integral of  $U$  against  $x$  gives swallowtail

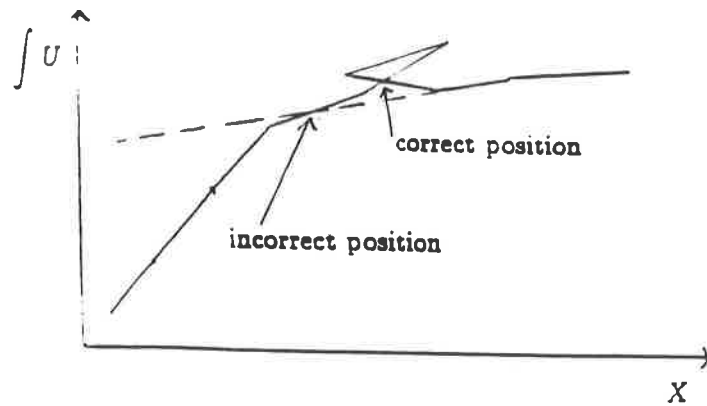


Figure 8: integrated curve with swallowtail

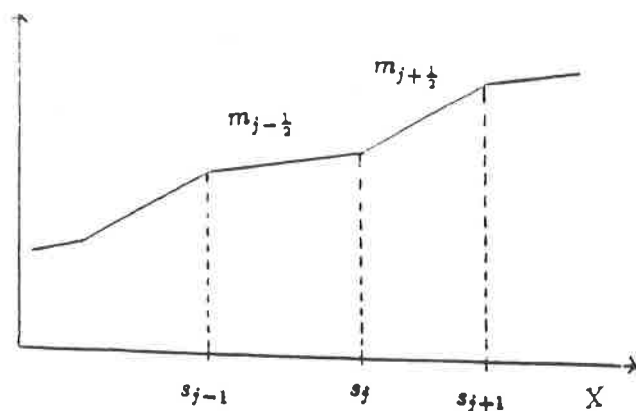
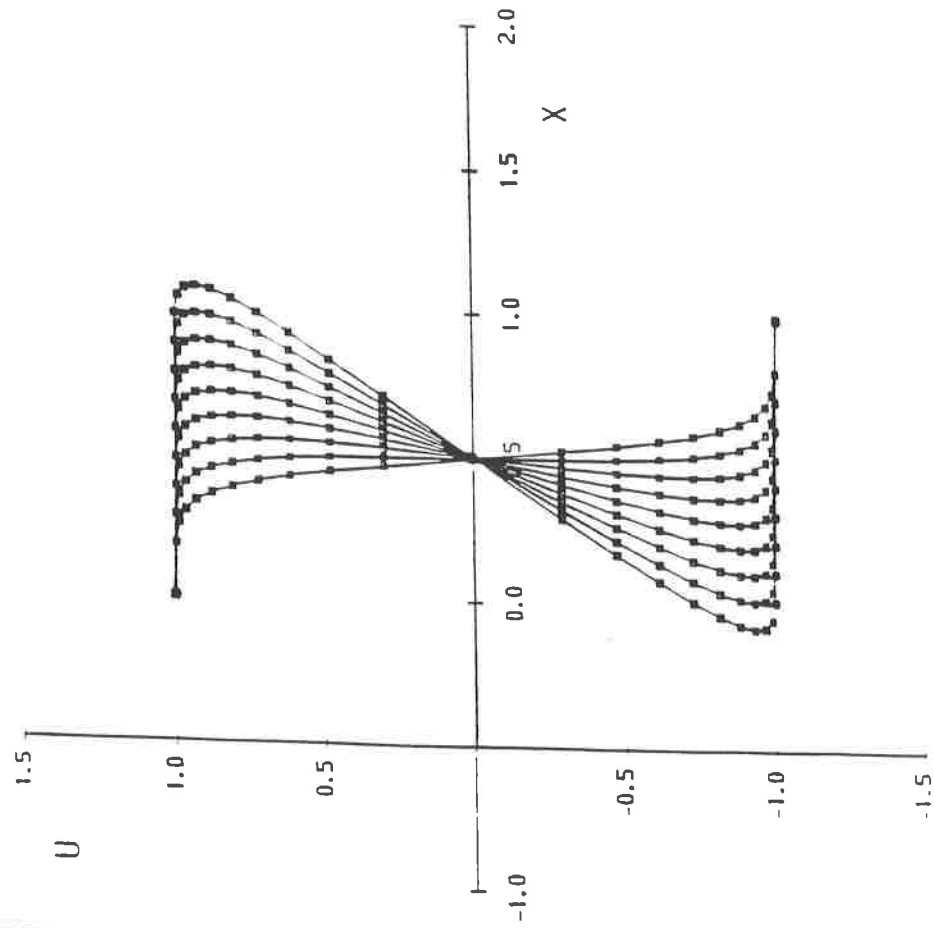


Figure 9: piecewise linear curve

SOLUTION OF BURGERS EQUATION



METHOD OF SOLUTION  
MFE IN A,S SPACE

INITIAL DATA FUNCTION IS

$$\text{TANH}(5-10X) \quad 0 \leq X \leq 1$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 80

RESULTS EVERY 10 TIMESTEPS

figure 10



SOLUTION OF BURGERS EQUATION

METHOD OF SOLUTION  
VIA INTEGRAL USING MFE  
IN A, S SPACE

INITIAL DATA FUNCTION IS

$$\tanh(5-10x) \quad 0 \leq x \leq 1$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 80

RESULTS EVERY 10 TIMESTEPS

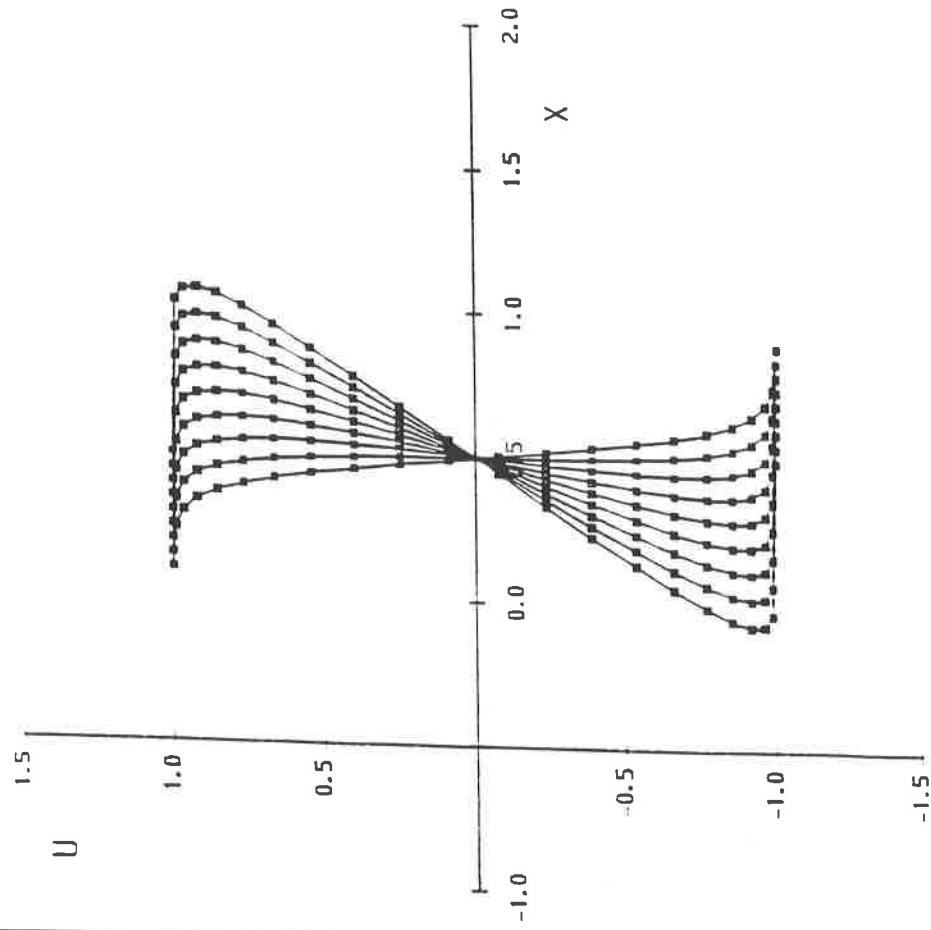
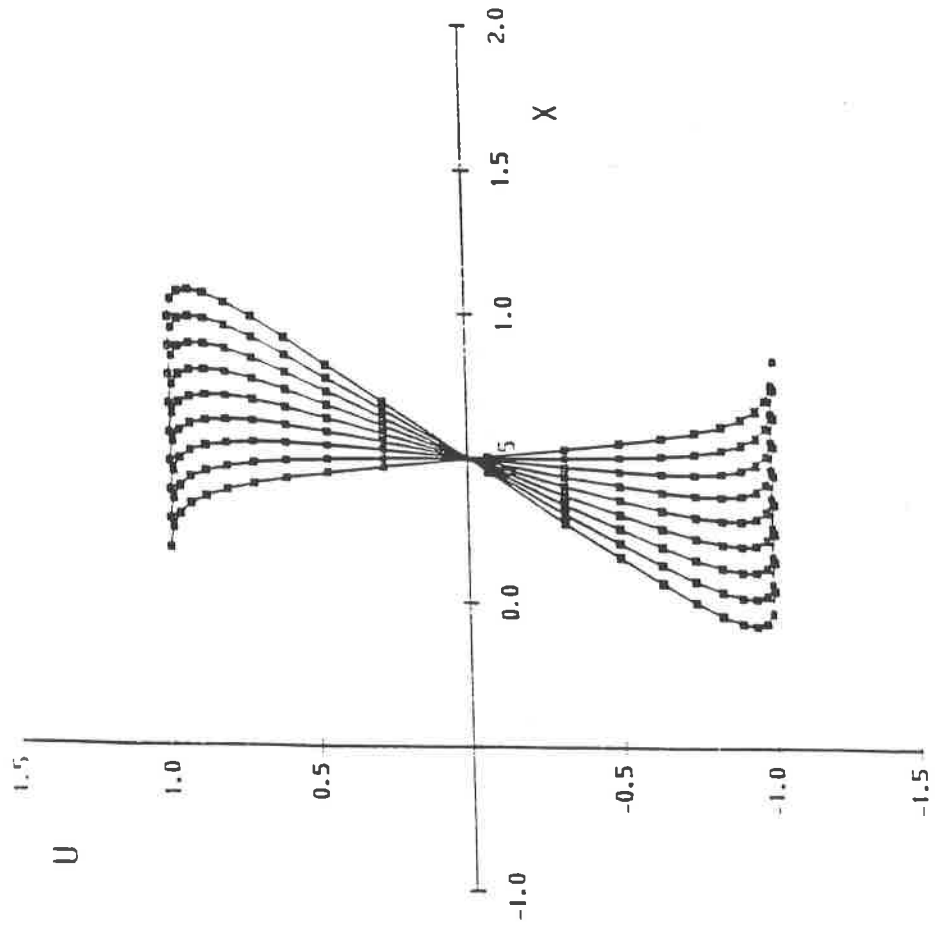


figure 11

### SOLUTION OF BURGERS EQUATION



### METHOD OF SOLUTION IN M, V SPACE

INITIAL DATA FUNCTION IS

$$\tanh(5-10X) \quad 0 \leq X \leq 1$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

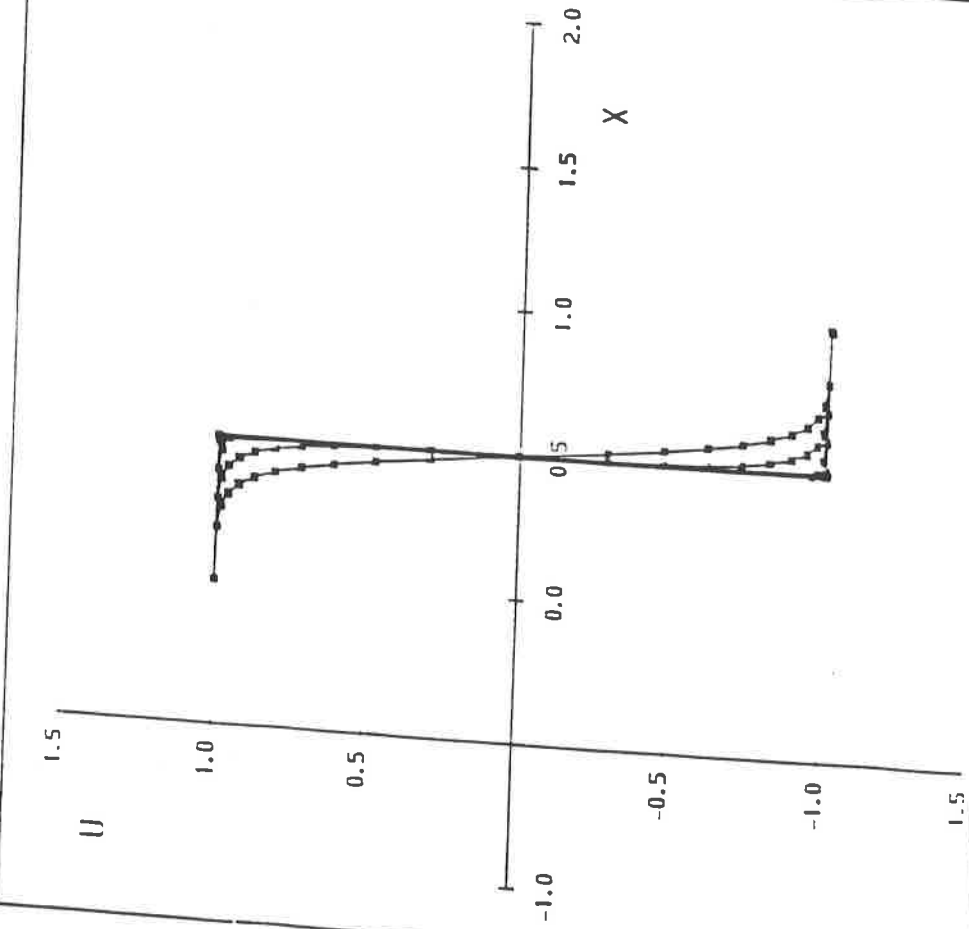
TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 80

PRINT IS EVERY 10 TIMESTEPS

figure 12

### SOLUTION OF BURGERS EQUATION



METHOD OF SOLUTION  
FIVE IN A, S SPACE  
SHOCK CALCULATED USING  
DIRECT METHOD

INITIAL DATA FUNCTION IS

$$\tanh(5-10X) \quad 0 \leq X \leq 1$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

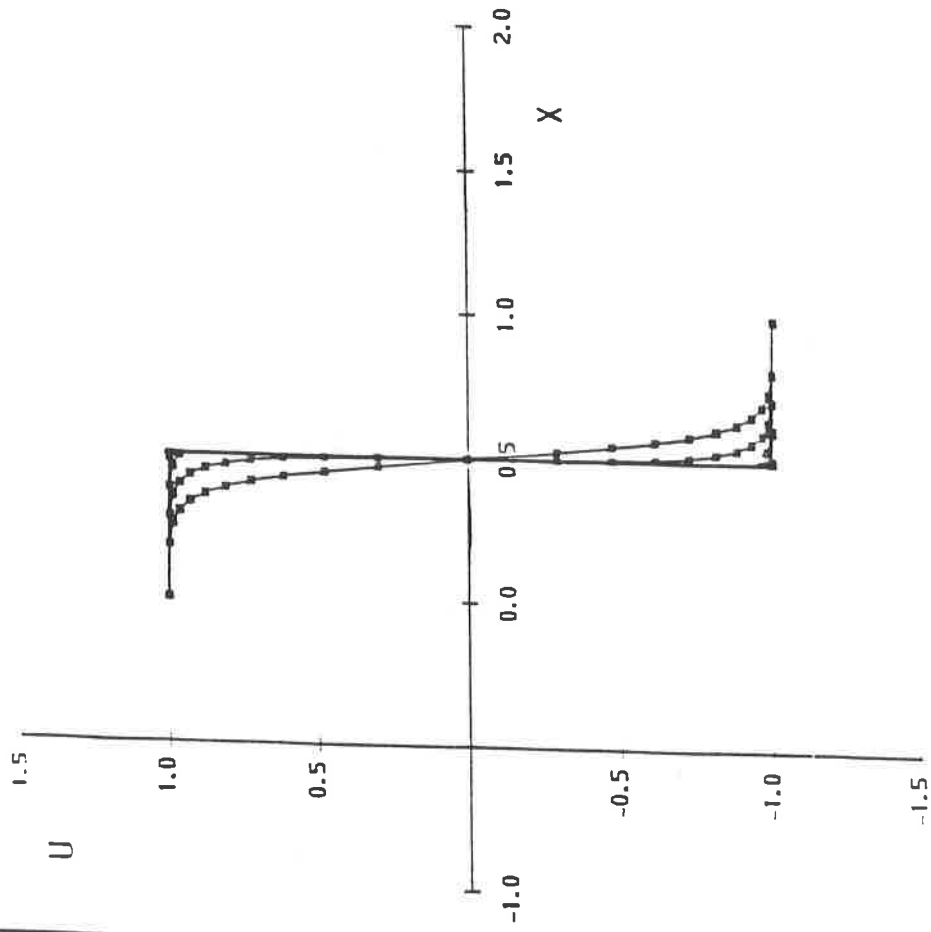
TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 60

RESULT IS EVERY 10 TIMESTEPS

figure 13

SOLUTION OF BURGERS EQUATION



METHOD OF SOLUTION  
MFE IN A, S SPACE  
SHOCK CALCULATED USING  
LINE INTERSECTIONS

INITIAL DATA FUNCTION IS  
 $\tanh(15-10X) \quad 0 \leq X \leq 1$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

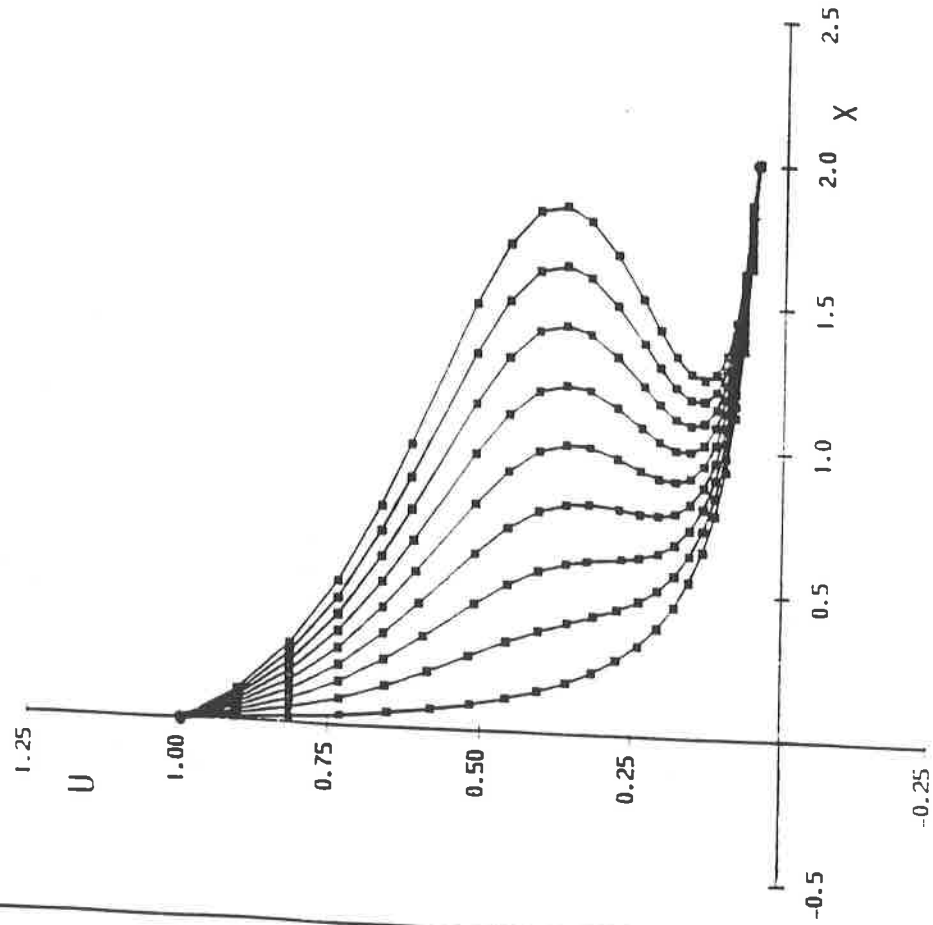
TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 80

RESULTS EVERY 10 TIMESTEPS

figure 14

### SOLUTION OF BUCKLEY-LEVERETT EQUATION



METHOD OF SOLUTION  
MFE IN A, S SPACE

INITIAL DATA FUNCTION IS

$$1/(1+10X) \quad 0 \leq X \leq 2$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 80

RESIN IS EVERY 10 TIMESTEPS

figure 16

SOLUTION OF BUCKLEY-LEVERETT EQUATION

METHOD OF SOLUTION  
VIA INTEGRAL USING  
MFE IN A, S SPACE

INITIAL DATA FUNCTION IS

$$1/(1+10X) \quad 0 \rightarrow X \rightarrow 2$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 80

RESULT IS EVERY 10 TIMESTEPS

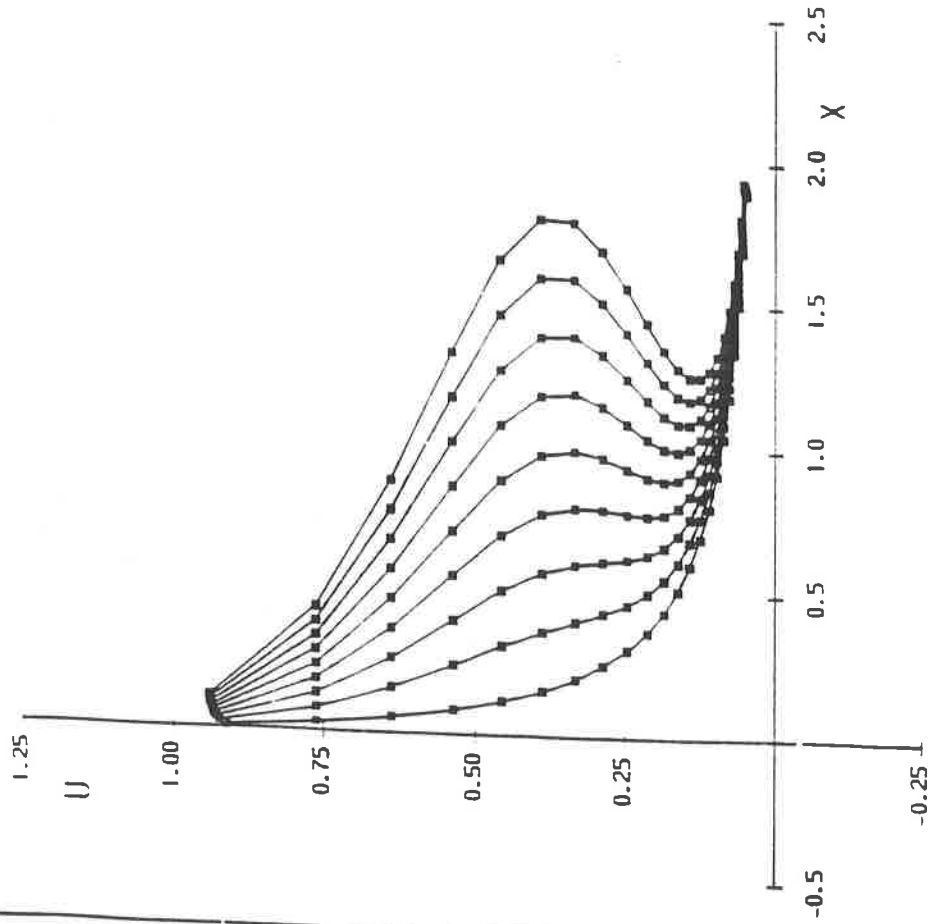
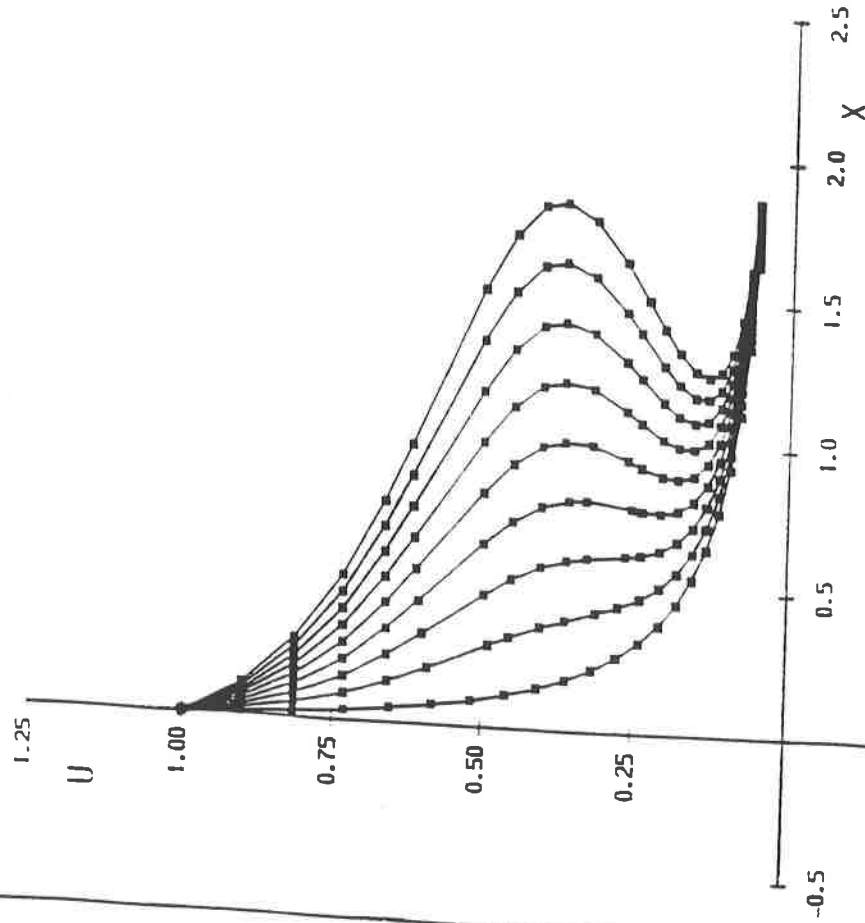


figure 16

SOLUTION OF BUCKLEY-LEVERETT EQUATION



METHOD OF SOLUTION  
IN M, V SPACE

INITIAL DATA FUNCTION IS

$$1/(1+10X) \quad 0 \leq X \leq 2$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

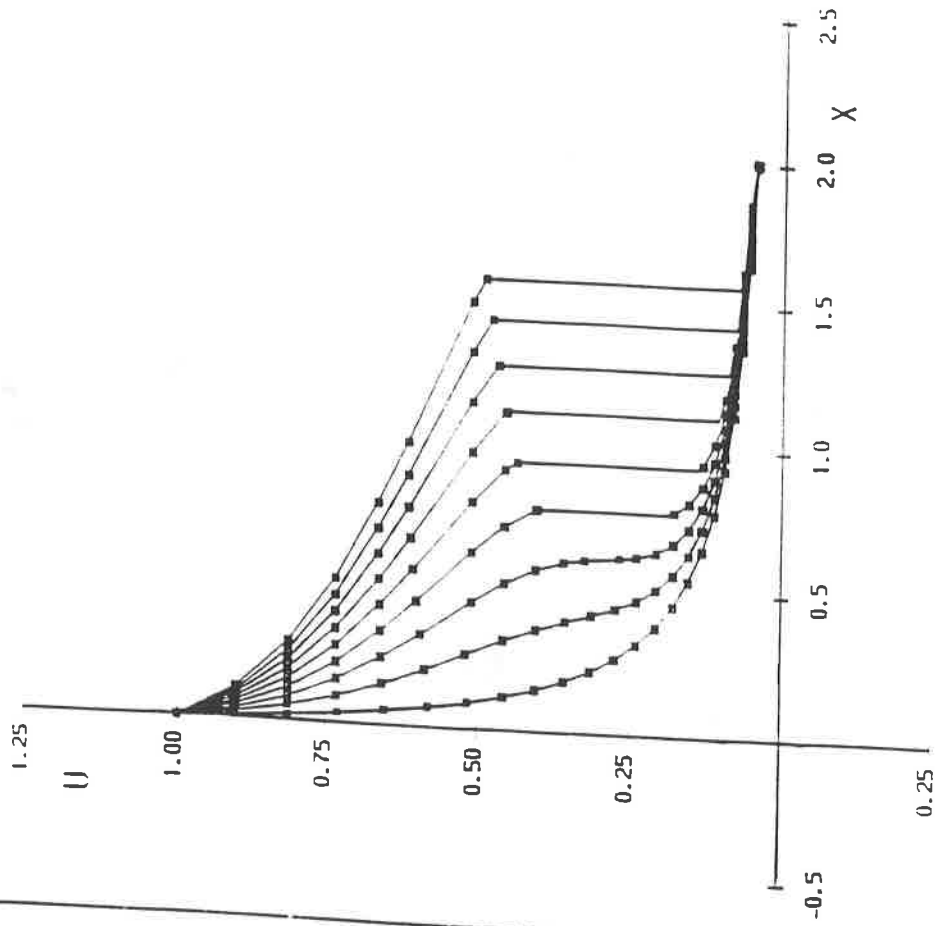
TIMESTEP 0.001

TOTAL NUMBER OF TIMESTEPS 800

RESID IS EVERY 100 TIMESTEPS

figure 17

SOLUTION OF BUCKLEY LEVERETT EQUATION



METHOD OF SOLUTION  
MFE IN A, S SPACE  
SHOCK CALCULATED USING  
DIRECT METHOD

INITIAL DATA FUNCTION IS  
 $1/(1+10X) \quad 0 \rightarrow X \rightarrow 2$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

TIMESTEP 0.01

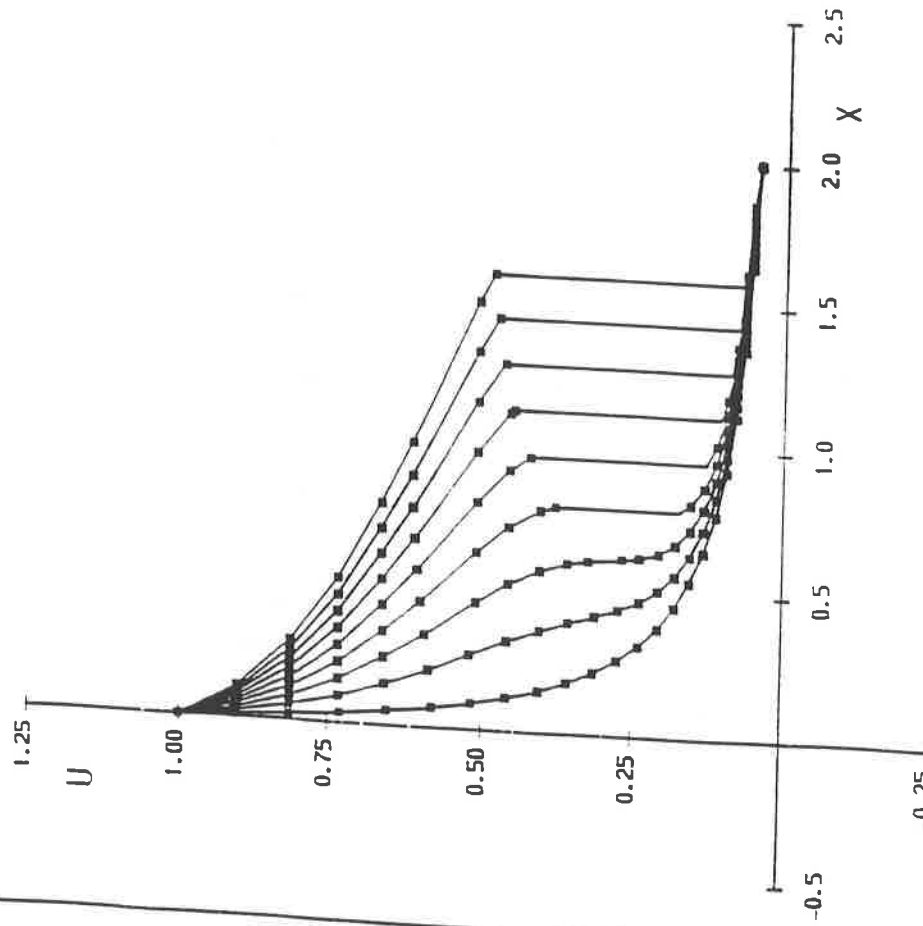
TOTAL NUMBER OF TIMESTEPS 60

RESULTS EVERY 10 TIMESTEPS

figure 18



### SOLUTION OF BUCKLEY-LEVERETT EQUATION



METHOD OF SOLUTION  
MFE IN A, S SPACE  
SHOCK CALCULATED USING  
LINE INTERSECTIONS

INITIAL DATA FUNCTION IS

$$1/(1+10X) \quad 0 \leq X \leq 2$$

BOUNDARY CONDITIONS DIRICHLET

NUMBER OF NODES 21

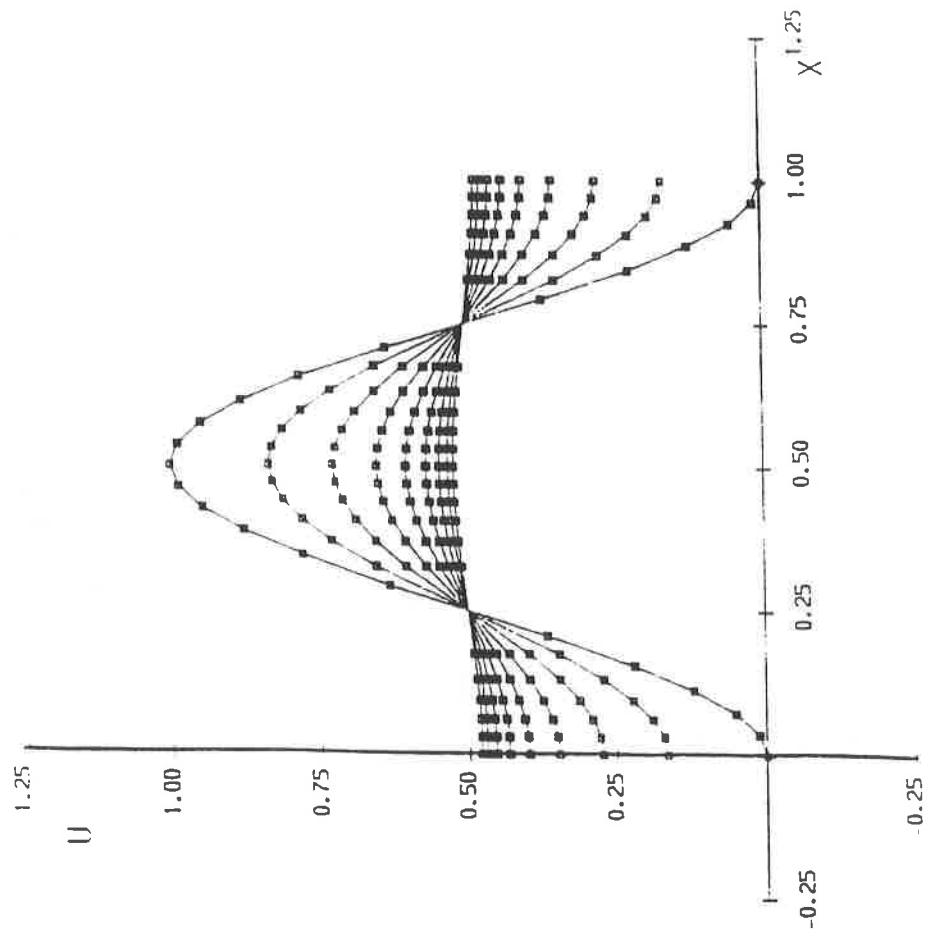
TIMESTEP 0.01

TOTAL NUMBER OF TIMESTEPS 80

RESULTS EVERY 10 TIMESTEPS

figure 19

SOLUTION OF HEAT EQUATION



METHOD OF SOLUTION  
MFE IN A, S SPACE

INITIAL DATA FUNCTION IS

$$\sin^2 \pi x \quad 0 \leq x \leq 1$$

BOUNDARY CONDITIONS NEUMANN

NUMBER OF NODES 21

TIMESTEP 0.0001000

TOTAL NUMBER OF TIMESTEPS 800

RESULTS EVERY 100 TIMESTEPS

figure 20

### SOLUTION OF HEAT EQUATION

METHOD OF SOLUTION  
V, M IN SPACE

INITIAL DATA FUNCTION IS  
 $\sin^2(x)$

BOUNDARY CONDITIONS NEUMANN

NUMBER OF NODES 21  
TIMESTEP 0.0001

TOTAL NUMBER OF TIMESTEPS 800  
RESULTS EVERY 100 TIMESTEPS

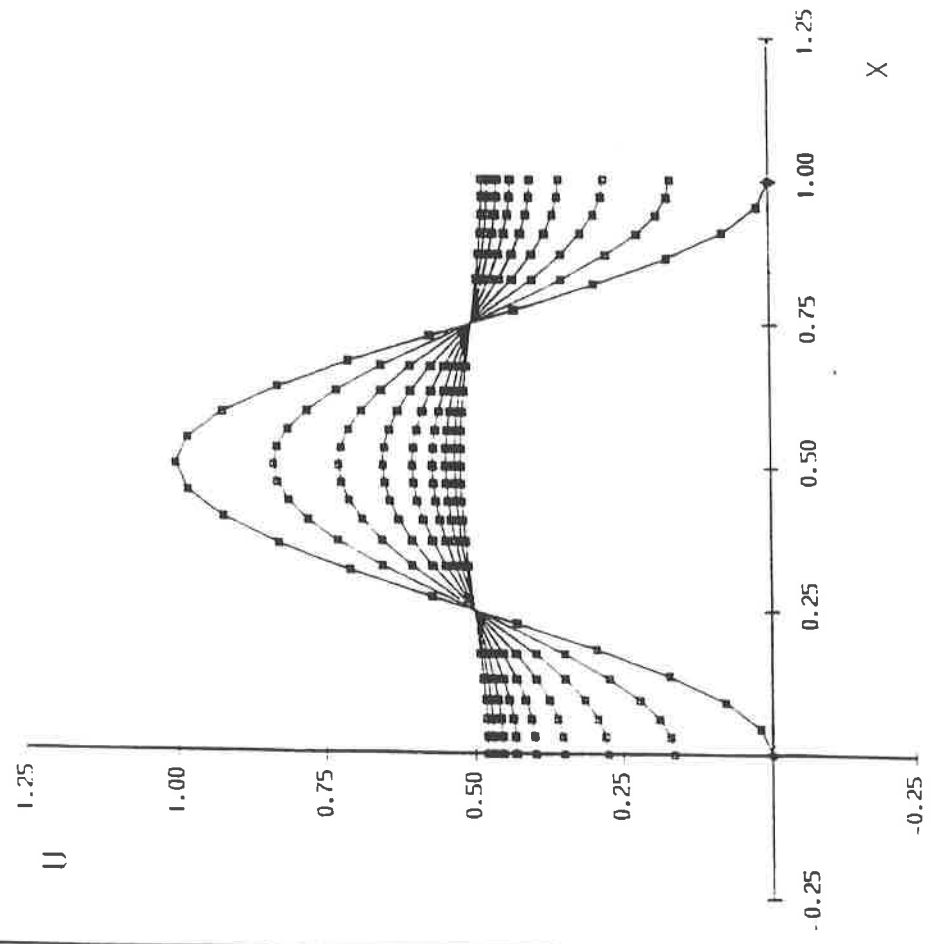


figure 21