THE MOVING FINITE ELEMENT METHOD FOR THE VISCOUS BURGERS' EQUATION

I.W. JOHNSON

NUMERICAL ANALYSIS REPORT 3/84

ABSTRACT

We discuss the application of the Moving Finite Element (MFE) Method to diffusion problems with solutions which develop steep moving fronts. Attention is focused on one-dimensional examples, in particular the viscous Burger's equation, but we eventually intend to apply the method to two-dimensional thermal conduction problems of interest to A.W.R.E., Aldermaston.

We look at the various approximation procedures for representing the second derivative terms in diffusion equations and show the equivalence between the use of a "recovered" function for this purpose and the method of δ-mollification introduced by Miller. We show also how nodal movement is influenced by the approximation procedure and how node overtaking may be controlled without the introduction of penalty functions.

Comparison of numerical results is made using a number of different time-stepping schemes and the effect of each technique on the problem of node overtaking is considered.

1. INTRODUCTION

The MFE method for the solution of time dependent partial differential equations, introduced by Miller & Miller [2], has been used with considerable success for both parabolic problems (see Miller [3], Miller & Miller [2], Gelinas, Doss & Miller [1], Herbst [4]) and hyperbolic problems (see Wathen [5], Wathen & Baines [6]). A review of the essential details of the method is given in section 2 below.

In section 3 we discuss the problem of the approximation procedure for the second derivative term in parabolic problems, introducing the idea of a recovered function as an alternative to the limiting process used by Miller in [2], and show the equivalence of the two techniques under certain circumstances.

In section 4 we give numerical results for a problem governed by the viscous Burger's equation with a sharp front and show how the problem of node overtaking places a restriction on the time steps which may be taken. This restriction leads us to discuss (in section 5) alternative time stepping techniques to the explicit Euler method used in section 4, and we give a comparison of the effect of each of the techniques on the problem of node overtaking.

Throughout this work we avoid the use of penalty functions to prevent node overtaking or to resolve the problem of parallelism, as introduced by Miller, since that approach has the unsatisfactory feature that several free parameters must be chosen, which may be highly problem dependent. We treat parallelism following Wathen & Baines [6], and as an alternative to the use of penalty functions we analyse in section 6 the effect of the various approximation procedures for second derivative terms on the movement of the nodes and give examples of ways in which nodal movement may be constrained.

In section 7 we briefly investigate the possibility of deleting nodes when overtaking occurs and reintroducing them so as to best preserve the

accuracy of the approximation.

Section 8 gives a conclusion from the numerical results and analysis of the previous sections.

2. MOVING FINITE ELEMENT METHOD

In this section we review the essential details of the MFE method using linear elements in one space dimension.

In general we seek approximate solutions to the evolutionary equation

$$u_t - L(u) = 0 \tag{2.1}$$

where L is some non-linear spacial differential operator.

We take a semi-discrete approximation

$$U(x,t) = \sum_{j=1}^{N} U_j(t)\alpha_j \{x,\underline{s}(t)\} \tag{2.2}$$

where α_j is a standard piecewise linear finite element basis function on the grid defined by the N-dimensional vector $\underline{s}$ of time dependent nodal positions (see Fig. 2.2).

Partial differentiation of (2.2) with respect to time yields

$$U_t(x,t) = \sum_{j=1}^{N} \dot{U}_j\alpha_j(x,\underline{s}(t)) + \dot{s}_j\beta_j(x,\underline{U}(t),\underline{s}(t)) \tag{2.3}$$

where the . denotes time differentiation and where $\beta_j = -\frac{\partial U}{\partial x}\alpha_j$ may be regarded as a second type of basis function (see Fig. 2.3). The basis function α_j is given by

$$\alpha_j = \begin{cases} \dfrac{x - s_{j-1}}{\Delta s_j} & \text{for } s_{j-1} \leq x \leq s_j \\ \dfrac{s_{j+1} - x}{\Delta s_{j+1}} & \text{for } s_j \leq x \leq s_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

where $\Delta s_j = s_j - s_{j-1}$.

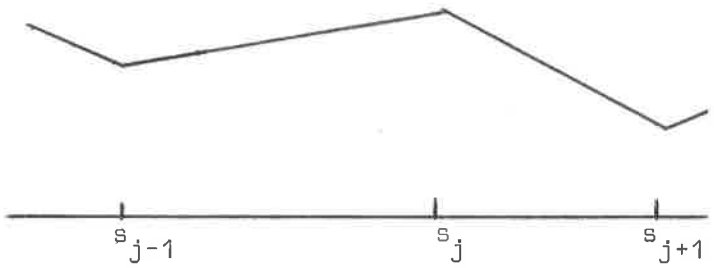

Figure 2.1

Piece wise linear approximation U(x,t)

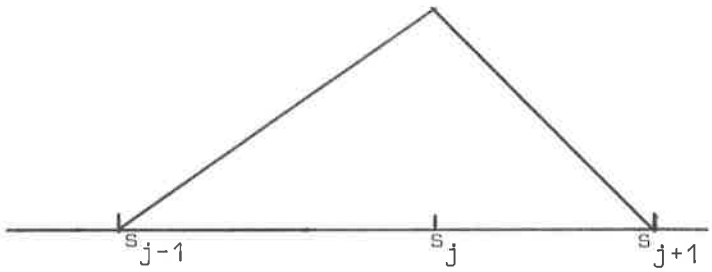

Figure 2.2

Basis function α_j

The basis function β_j is given by

$$\beta_j = \begin{cases} -m_j\alpha_j & \text{for } s_{j-1} \leq x < s_j \\ -m_{j+1}\alpha_{j+1} & \text{for } s_j < x \leq s_{j+1} \end{cases}$$

where $m_j = \dfrac{\Delta U_j}{\Delta s_j}$ is the gradient of the approximation U on the j^{th} element.

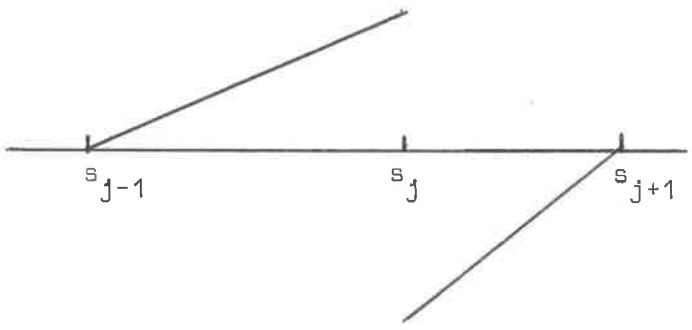

Figure 2.3

Basis function β_j

Minimising the square of the L_2 norm of the residual,

$$\| U_t - L(U) \|^2_{L_2}$$

with respect to the 2N parameters $\dot{U}_j$, $\dot{s}_j$ yields the set of 2N equations.

$$\left.\begin{aligned} \langle U_t - L(U), \alpha_j \rangle &= 0 \\ \langle U_t - L(U), \beta_j \rangle &= 0 \end{aligned}\right\} \quad j = 1,\ldots,N \tag{2.4}$$

where $\langle \cdot,\cdot \rangle$ denotes the L_2 inner product. Substituting U_t given by (2.3) then yields the non-linear system of ordinary differential equations

$$A(\underline{y})\dot{\underline{y}} = \underline{g}(\underline{y}) \tag{2.5}$$

which are referred to as the MFE equations. Here

$$\underline{y} = (U_1, s_1, \ldots, U_N, s_N)^T ,$$

$A(\underline{y})$ is the MFE matrix which is square, symmetric and 2 x 2 block tridiagonal with blocks given by

$$A_{ij} = \begin{bmatrix} \langle \alpha_i, \alpha_j \rangle & \langle \alpha_i, \beta_j \rangle \\ \langle \beta_i, \alpha_j \rangle & \langle \beta_i, \beta_j \rangle \end{bmatrix} \tag{2.6}$$

and the elements of the vector $\underline{g}$ are defined by

$$\left.\begin{aligned} g_{2i-1} &= \langle L(U), \alpha_i \rangle \\ g_{2i} &= \langle L(U), \beta_i \rangle \end{aligned}\right\} \quad (i = 1,\ldots,N) \tag{2.7}$$

The system (2.5) may be solved for the $\dot{U}_i, \dot{s}_i$ using some iterative technique, the generalised conjugate gradient method (see e.g. [7]) being chosen throughout the work in this report.

Details of various time stepping techniques which may then be used to yield U_i, s_i $(i = 1,\ldots,N)$ at the next time level are discussed in section 5, although in the averaging analysis of section 6 it is assumed that explicit Euler time stepping is used.

3. SPECIAL TREATMENT OF SECOND DERIVATIVE TERMS IN DIFFUSION PROBLEMS

In this section we draw attention to the limitations of the approximation procedure described in the previous section when second derivatives are present in $L(u)$, such as in the case of diffusion problems of the form

$$u_t + f_x(u) = \varepsilon u_{xx} \tag{3.1}$$

where in (2.1) $$L(u) = f_x(u) - \varepsilon u_{xx} \quad . \tag{3.2}$$

It is immediately apparent in solving

$$U_t = L(U)$$

with the approximation defined by (2.2) that $L(U)$ does not have a finite L_2 norm since U_{xx} consists of a sum of delta functions. Furthermore the solution of the MFE equations (2.5) requires the evaluation of inner products $\langle \beta_i, U_{xx} \rangle$ where U_{xx} exists only as a sum of delta functions at the nodes and β_i is itself discontinuous at the nodes (see below).

It would seem that one solution is to seek an approximate solution in a space of functions smoother than piece-wise linears, such that $L(U)$ will have a finite L_2 norm, but this will of course lead to an entirely different structure for the resulting MFE matrix. As an alternative, and in order to take advantage of the structure of the MFE matrix analysed by Wathen & Baines [6], we persevere with the approximation defined by (2.2), but introduce a special interpretation of the inner products (2.7) appearing on the right hand side of the MFE equations (2.5).

δ-Mollification

In [2] Miller resolves the difficulty of higher order derivatives by interpreting the limiting equations obtained by applying the minimisation process to the manifold M_δ of "smoothed off" or "δ-mollified" piece-wise linear functions, in the limit as $\delta \to 0$. The resulting inner products tend to their expected limits in the sense of distributions as $\delta \to 0$, but in evaluating

the inner products $\langle U_{xx}, \beta_i \rangle$ on the right hand side of the MFE equations (2.6) the basis function β_i must be assumed to take its mean value $-\frac{1}{2}(m_{i+1} + m_i)$ at the point s_i, where $m_i = U_x$ on element i.

An alternative to the Miller approach is to replace the inner products $\langle U_{xx}, \alpha_i \rangle$, $\langle U_{xx}, \beta_i \rangle$ appearing on the right hand side of (2.5) by $\langle w_{xx}, \alpha_i \rangle$, $\langle w_{xx}, \beta_i \rangle$, respectively, where $w(x)$ is some "recovered" function lying in a smoother space than $U(x)$, such that w_{xx} has a finite L_2 norm. Similarly, these inner products may be replaced by $\langle v_x, \alpha_i \rangle$, $\langle v_x, \beta_i \rangle$, where $v(x)$ is some function which is recovered from the gradient $U_x(x)$ (which for linear elements is piece-wise constant).

We now show that the inner products resulting from the process of δ-mollification are identical to those produced using a Hermite cubic function $w(x)$ recovered from $U(x)$.

We replace the inner products

$$\langle U_{xx}, \alpha_i \rangle \qquad \langle U_{xx}, \beta_i \rangle$$

by

$$\langle w_{xx}, \alpha_i \rangle \qquad \langle w_{xx}, \beta_i \rangle \tag{3.2a}$$

where $w(x)$ is a Hermite cubic function defined on each element i by

$$w(x) = \begin{cases} U_{i-1} & \text{at } x = s_{i-1} \\ U_i & \text{at } x = s_i \end{cases} \tag{3.3}$$

$$w_x(x) = \begin{cases} \frac{1}{2}(m_i + m_{i-1}) & \text{at } x = s_{i-1} \\ \frac{1}{2}(m_i + m_{i+1}) & \text{at } x = s_i \end{cases}$$

Using the notation

$$\alpha_i^- = \begin{cases} \dfrac{x - s_{i-1}}{\Delta s_i} & s_{i-1} \leq x \leq s_i \\ 0 & \text{elsewhere} \end{cases}$$

$$\alpha_i^+ = \begin{cases} \dfrac{s_{i+1} - x}{\Delta s_{i+1}} & s_i \leq x \leq s_{i+1} \\ 0 & \text{elsewhere} \end{cases} \tag{3.4}$$

we have,

$$\langle w_{xx},\alpha_i\rangle = \langle w_{xx},\alpha_i^-\rangle + \langle w_{xx},\alpha_i^+\rangle$$

$$= [w_x\,\alpha_i^-]_{s_{i-1}}^{s_i} - \frac{1}{\Delta s_i}\int_{s_{i-1}}^{s_i} w_x dx + [w_x\,\alpha_i^+]_{s_i}^{s_{i+1}} + \frac{1}{\Delta s_{i+1}}\int_{s_i}^{s_{i+1}} w_x dx$$

$$= w_x(s_i) - \frac{1}{\Delta s_i}(w(s_i) - w(s_{i-1})) - w_x(s_i) + \frac{1}{\Delta s_{i+1}}(w(s_{i+1}) - w(s_i)) \qquad (3.5)$$

and using (3.3) we have

$$\langle w_{xx},\alpha_i\rangle = m_{i+1} - m_i\,. \qquad (3.6)$$

Noting that we may write

$$\beta_i = -m_i\alpha_i^- - m_{i+1}\alpha_i^+$$

then

$$\langle w_{xx},\beta_i\rangle = -m_i\langle w_{xx},\alpha_i^-\rangle - m_{i+1}\langle w_{xx},\alpha_i^+\rangle$$

$$= -m_i(\tfrac{1}{2}(m_i + m_{i+1}) - m_i) - m_{i+1}(-\tfrac{1}{2}(m_i + m_{i+1}) + m_{i+1}).$$

Using (3.5) and (3.3)

$$\langle w_{xx},\beta_i\rangle = -\tfrac{1}{2}m_i(m_{i+1} - m_i) - \tfrac{1}{2}m_{i+1}(m_{i+1} - m_i)$$

$$= -\tfrac{1}{2}(m_{i+1}^2 - m_i^2)\,. \qquad (3.7)$$

The results (3.6), (3.7) are identical to those achieved by Miller's δ-mollification process referenced above.

Although this result is of no particular practical advantage in the one-dimensional problem, the use of a recovered function for the interpretation of the inner products on the right hand side of (2.5), rather than the δ-mollification process, generalises directly to two-dimensional problems, as one may readily define two-dimensional Hermite cubic functions and avoid interpretations of the delta functions in two-dimensions.

By using an element-wise formulation of the MFE equations it is also possible to show that if we recover U_{xx} in the form $(DU)_x$ where $DU(x)$ is some function recovered from the gradients $(U_x)_i$, lying in a smoother space than U_x, then in order to prevent node overtaking $DU(x)$ must be at least piece-wise quadratic.

If we begin by looking at the inviscid Burgers' equation

$$u_t + f_x = 0 \tag{3.8}$$

then following Herbst [4] we may replace the equations (2.4), which are given here by

$$\begin{aligned} &\langle U_t + f_x, \alpha_i \rangle = 0 \\ &\langle U_t + f_x, \beta_i \rangle = 0 \qquad i = 1,\ldots,N \end{aligned}$$

by the equations

$$\begin{aligned} &\langle U_t + f_x, \alpha_i \rangle = 0 \\ &\langle U_t + f_x, \hat{\beta}_i \rangle = 0 \qquad i = 1,\ldots,N \end{aligned} \tag{3.9}$$

where $\hat{\beta}_i$ is defined by

$$\beta_i = \tfrac{1}{2}(m_i + m_{i+1})\alpha_i + \tfrac{1}{2}(m_{i+1} - m_i)\hat{\beta}_i$$

(see Fig. 3.1).

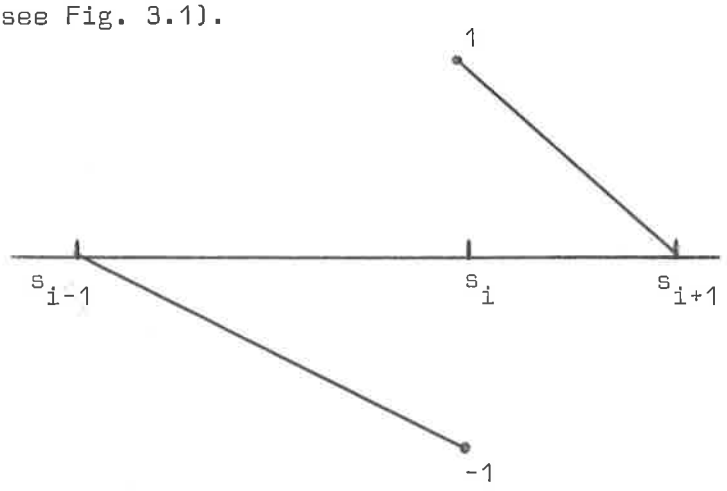

Figure 3.1
Basis function $\hat{\beta}_i$

Equations (3.9) may in turn be replaced by

$$\langle U_t + f_x, \phi_i^{\pm} \rangle = 0 \qquad i = 1,\ldots,N \tag{3.10}$$

where the element basis functions $\phi_i^{\pm}$ are given by

$$\phi_i^- = \begin{cases} \dfrac{x - s_{i-1}}{\Delta s_i} & s_{i-1} \leq x \leq s_i \\ 0 & \text{elsewhere} \end{cases} \qquad (3.11)$$

$$\phi_i^+ = \begin{cases} \dfrac{s_{i+1} - x}{\Delta s_i} & s_i \leq x \leq s_{i+1} \\ 0 & \text{elsewhere} \end{cases}$$

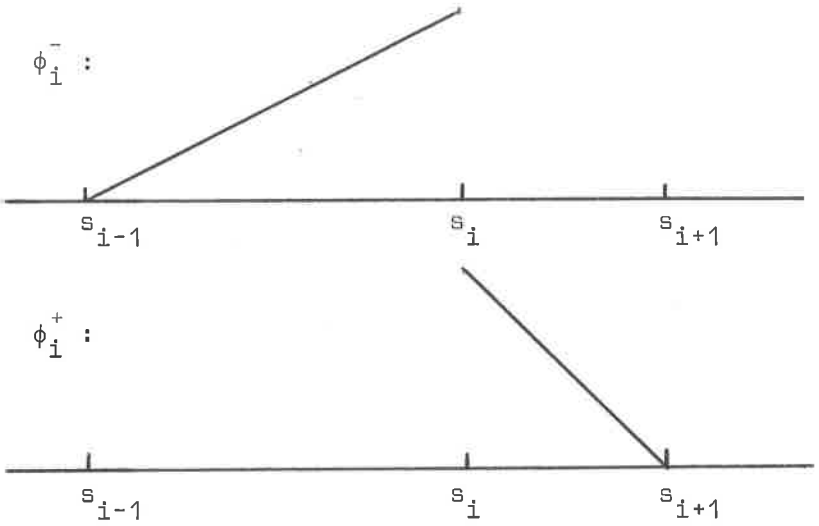

Figure 3.2

Evaluating the inner products in (3.10) on element i we have

$$2\langle U_t + f_x, \phi_{i-1}^+ \rangle = \frac{2}{3}\dot{U}_{i-1} + \frac{1}{3}\dot{U}_i - \frac{\Delta_- U_i}{\Delta_- s_i}\left(\frac{2}{3}\dot{s}_{i-1} + \frac{1}{3}\dot{s}_i\right)$$

$$+ \frac{2}{\Delta_- s_i}\left(\langle f\rangle_{i-1,i} - f_{i-1}\right) = 0 \qquad (3.12)$$

and

$$2\langle U_t + f_x, \phi_i^- \rangle = \frac{1}{3}\dot{U}_{i-1} + \frac{2}{3}\dot{U}_i - \frac{\Delta_- U_i}{\Delta_- s_i}\left(\frac{1}{3}\dot{s}_{i-1} + \frac{2}{3}\dot{s}_i\right)$$

$$+ \frac{2}{\Delta_- s_i}\left(f_i - \langle f\rangle_{i-1,i}\right) = 0 \qquad (3.13)$$

where

$$\langle f\rangle_{i-1,i} = \int_0^1 f \, dt$$

and

$$t = \frac{s_i - x}{\Delta_- s_i}$$

Eliminating $(\dot{U}_{i-1},\dot{s}_{i-1})$ or $(\dot{U}_i,\dot{s}_i)$ between (3.13) and (3.12) yields

$$\left.\begin{aligned} &\dot{U}_{i-1} - \frac{\Delta_- U_i}{\Delta_- s_i}\,\dot{s}_{i-1} + \frac{2}{\Delta_- s_i}\left[3\langle f\rangle_{i-1,i} - 2f_{i-1} - f_i\right] = 0 \\ \text{and}\quad &\dot{U}_i - \frac{\Delta_- U_i}{\Delta_- s_i}\,\dot{s}_i + \frac{2}{\Delta_- s_i}\left[f_{i-1} + 2f_i - 3\langle f\rangle_{i-1,i}\right] = 0 \end{aligned}\right\} \qquad (3.14)$$

Now to prevent node overtaking (using explicit Euler time stepping) we require

$$\Delta t\,\Delta_-\dot{s}_i \geq -\Delta_- s_i \qquad (3.15)$$

Using (3.15) and equations (3.14), and assuming that for a steady shock we have $\dot{U}_j = 0$ $(j = 1,\ldots,N)$, then in order to prevent node overtaking in time Δt we require

$$\Delta t\,\frac{\Delta_- s_i}{\Delta_- U_i} + \frac{2}{\Delta_- s_i}\left[3(f_{i-1}+f_i) - 6\langle f\rangle_{i-1,i}\right] \geq -\Delta_- s_i$$

and assuming $\Delta_- U_i < 0$ then

$$\frac{\Delta_- U_i}{\Delta t} + \frac{2}{\Delta s_i}\left[3(f_{i-1} + f_i) - 6\langle f\rangle_{i-1,i}\right] \leq 0\,. \qquad (3.16)$$

If we now consider the viscous form of equation (3.8)

$$u_t + f_x = \varepsilon u_{xx}$$

and recover U_{xx} in the form $(DU)_x$, then we are solving

$$u_t + \tilde{f}_x = 0 \qquad (3.17)$$

where $\tilde{f}(x) = f(x) - \varepsilon(DU)_x$.

Hence, using (3.16) with $f(x)$ replaced by $\tilde{f}(x)$, to prevent node overtaking in (3.17) we require that

$$\frac{\Delta_- U_i}{\Delta t} + \frac{2}{\Delta_- s_i}\left[3(\tilde{f}_{i-1}+\tilde{f}_i) - 6\langle\tilde{f}\rangle_{i-1,i}\right] \leq 0\,. \qquad (3.18)$$

Hence in order to prevent node overtaking in (3.17) we need to reduce

$$\tfrac{1}{2}\left(f_{i-1} + f_i\right) - \langle f\rangle_{i-1,i}$$

in the change from f to $\tilde{f}$: clearly this cannot be achieved using a piece-wise linear $DU(x)$ and hence we must choose at least a piece-wise quadratic $DU(x)$.

In the numerical results given in the next section we have chosen $DU(x)$ as follows. On element i $DU(x)$ is the quadratic defined by

$$DU(x) = \begin{cases} \frac{1}{2}(m_{i-1} + m_i) & \text{at } x = s_{i-1} \\ m_i & \text{at } x = \frac{1}{2}(s_{i-1} + s_i) \\ \frac{1}{2}(m_i + m_{i+1}) & \text{at } x = s_i \end{cases} \tag{3.19}$$

Note that $(DU)x$ is in general discontinuous at the nodes.

On element i $DU(x)$ is the quadratic

$$DU(x) = \frac{(x-s_{i-1})(x-\frac{1}{2}(s_i+s_{i-1}))}{\Delta_- s_i \ \frac{1}{2}\Delta_- s_i} \cdot \tfrac{1}{2}(m_i+m_{i+1})$$

$$+\frac{(x-s_{i-1})(x-s_i)}{\frac{1}{2}\Delta_- s_i(-\frac{1}{2}\Delta_- s_i)} \cdot m_i + \frac{(x-\frac{1}{2}(s_i+s_{i-1}))(x-s_i)}{(-\frac{1}{2}\Delta_- s_i)(-\Delta_- s_i)} \cdot \tfrac{1}{2}(m_i + m_{i-1})$$

$$= \frac{2}{(\Delta_- s_i)^2} \Big\{ x^2[\tfrac{1}{2}m_{i-1} - m_i + \tfrac{1}{2}m_{i+1}]$$

$$+ x[s_i m_i + s_{i-1}m_i - \frac{1}{4}s_i m_{i+1} - \frac{3}{4}s_{i-1}m_{i+1} - \frac{3}{4}s_i m_{i-1} - \frac{1}{4}s_{i-1}m_{i-1}]$$

$$+ [-\frac{3}{2}s_i s_{i-1}m_i + \frac{1}{4}s_{i-1}^2 m_i + \frac{1}{4}s_i^2 m_i + \frac{1}{4}s_i s_{i-1}m_{i+1}$$

$$+ \frac{1}{4}s_{i-1}^2 m_{i+1} + \frac{1}{4}s_i s_{i-1}m_{i-1} + \frac{1}{4}s_i^2 m_{i-1}] \Big\} . \tag{3.20}$$

If we are solving the equation

$$u_t + f_x = \varepsilon u_{xx} \ ,$$

then on the right hand side of the MFE equations we require the inner products

$$\langle (DU)_x, \alpha_i \rangle \langle (DU)_x, \beta_i \rangle \qquad (i = 1,\ldots,N)$$

Now

$$\langle (DU(x))_x, \alpha^i \rangle = I1 + I2 \tag{3.21}$$

where

$$I1 = \int_{s_{i-1}}^{s_i} (DU)x \; \frac{x-s_{i-1}}{\Delta_- s_i} \; dx$$

$$I2 = \int_{s_i}^{s_{i+1}} (DU)x \; \frac{s_{i+1}-x}{\Delta_- s_{i+1}} \; dx \qquad (3.22)$$

Integrating by parts in (3.22) gives

$$I1 = \left[DU(x) \cdot \frac{x-s_{i-1}}{\Delta_- s_i}\right]_{s_{i-1}}^{s_i} - \frac{1}{\Delta_- s_i} \int_{s_{i-1}}^{s_i} DU(x)dx$$

$$I2 = \left[DU(x) \cdot \frac{s_{i+1}-x}{\Delta_- s_{i+1}}\right]_{s_i}^{s_{i+1}} + \frac{1}{\Delta_- s_{i+1}} \int_{s_i}^{s_{i+1}} DU(x)dx \qquad (3.23)$$

Using (3.20) and omitting the details

$$\int_{s_{i-1}}^{s_i} DU(x)dx = \frac{\Delta_- s_i}{12} (m_{i-1} + 10m_i + m_{i+1})$$

Hence in (3.23)

$$I1 = \tfrac{1}{2}(m_i + m_{i+1}) - \frac{1}{12}(m_{i-1} + 10m_i + m_{i+1}) \qquad (3.24)$$

and

$$I2 = -\tfrac{1}{2}(m_i + m_{i+1}) + \frac{1}{12}(m_i + 10m_{i+1} + m_{i+2})$$

Hence in (3.21)

$$\langle (DU)x, \alpha_i \rangle = \frac{1}{12} \{(m_i + 10m_{i+1} + m_{i+2}) - (m_{i-1} + 10m_i + m_{i+1})\} \qquad (3.25)$$

Also

$$\begin{aligned} \langle (DU)x, \beta^i \rangle &= - m_i I1 - m_{i+1} I2 \\ &= \frac{-1}{12} \{m_{i+1}(m_{i+1} 10m_{i+1} + m_{i+2}) \\ &\qquad - m_i (m_{i-1} + 10m_i + m_{i+1})\} \end{aligned} \qquad (3.26)$$

The inner products (3.25), (3.26) are used to produce the numerical results in the next section.

4. NUMERICAL RESULTS

The numerical results in this section refer to the test problem given in [4] which is described below. A particular solution of the viscous Burgers' equation

$$u_t + uu_x = \varepsilon u_{xx} \tag{4.1}$$

may be obtained using the Cole-Hopf transformation, giving

$$u(x,t) = f(\xi) \qquad \xi = x - \mu t - \beta \ . \tag{4.2}$$

Here

$$f(\xi) = \left[\mu + \alpha + (\mu-\alpha)\exp\left(\frac{\alpha\xi}{\varepsilon}\right)\right] / \left[1 + \exp\left(\frac{\alpha\xi}{\varepsilon}\right)\right] \tag{4.3}$$

where α, β and μ are arbitrary constants. This solution represents a steep wave front initially at $x = \beta$ travelling with speed μ.

Initial and boundary conditions are obtained from (4.2) as

$$u(x,0) = f(x - \beta) \tag{4.4}$$

$$\begin{aligned} u(0,t) &= f(-\mu t - \beta) \\ u(1,t) &= f(1 - \mu t - \beta) \end{aligned} \tag{4.5}$$

The values of the arbitrary constants were chosen as $\mu = 0.6$, $\alpha = 0.4$, $\beta = 0.125$ and $\varepsilon = 0.01$ throughout the results given below.

The time-dependent boundary conditions may be approximated by Dirichlet boundary conditions

$$U_0 = 1.0 \qquad U_{N+1} = 0.2$$

This approximation is accurate for small values of ε, but in cases where the approximation is not sufficiently accurate we may proceed as follows.

Consider the general problem

$$u_t - L(u) = 0 \qquad\qquad x \in [a,b]$$

with time dependent boundary conditions

$$\begin{aligned} u(a,t) &= f_1(t) \\ u(b,t) &= f_2(t) \ . \end{aligned} \tag{4.6}$$

We seek a piecewise linear approximation

$$U(x,t) = \sum_{j=0}^{N+1} U_j \alpha_j \tag{4.7}$$

where the nodes s_0, s_{N+1} are fixed at the end points $x = a$, $x = b$.

Partial differentiation with respect to time in (4.7) yields

$$\begin{aligned} U_t &= \sum_{j=0}^{N+1} \dot{U}_j \alpha_j + \dot{s}_j \beta_j \\ &= \dot{U}_0 \alpha_0 + \dot{U}_{N+1} \alpha_{N+1} + \sum_{j=1}^{N} \dot{U}_j \alpha_j + \dot{s}_j \beta_j \\ &= f_1(t)\alpha_0 + f_2(t)\alpha_{N+1} + \sum_{j=1}^{N} \dot{U}_j \alpha_j + \dot{s}_j \beta_j \, . \end{aligned} \tag{4.8}$$

Minimising $\| U_t - L(U) \|_{L_2}^2$ with respect to $\dot{U}_j$, $\dot{s}_j$ $(j = 1,\ldots,N)$ then yields the MFE equations

$$A(\underline{y})\dot{\underline{y}} = \tilde{\underline{g}}(\underline{y}) \tag{4.9}$$

where $\tilde{\underline{g}}(\underline{y})$ differs from $\underline{g}(\underline{y})$ in (2.7) for fixed boundary conditions as follows:

$$\begin{aligned} \tilde{g}_1 &= g_1 - f_1(t) \langle \alpha_0, \alpha_1 \rangle \\ \tilde{g}_2 &= g_2 - f_1(t) \langle \alpha_0, \beta_1 \rangle \\ \tilde{g}_i &= g_i \qquad (i = 3,\ldots, 2N-2) \\ \tilde{g}_{2N-1} &= g_{2N-1} - f_2(t) \langle \alpha_{N+1}, \alpha_N \rangle \\ \tilde{g}_{2N} &= g_{2N} - f_2(t) \langle \alpha_{N+1}, \beta_N \rangle \, . \end{aligned} \tag{4.10}$$

Hence the extension of the method to a problem with time dependent boundary conditions has resulted in a simple change to the vector on the right hand side of the MFE equations, with no change in the MFE matrix itself.

The initial nodal positions are calculated using an equidistributing principle, following Herbst [4]. An approximate equidistribution of $\int |u''|^{\frac{1}{2}} dx$ over each element is achieved as follows.

Let $g(x)$ denote a piece-wise constant approximation to the second derivative of the initial condition $f(x)$, i.e.

$$g(x) \cong |f''(x)| \qquad x \in [0,1] \ . \tag{4.11}$$

The approximation is taken over a large number of equal sub-intervals of [0,1].

Define

$$I(x) = \int_0^x (g(p))^{\frac{1}{2}} dp \ . \tag{4.12}$$

The initial nodal positions s_i are given by

$$\int_0^{s_i} (g(p))^{\frac{1}{2}} dp = \frac{i}{N+1} \int_0^1 (g(p))^{\frac{1}{2}} dp \tag{4.13}$$

giving

$$s_i = I^{-1}(iI(1)/N+1) \qquad (i = 1...,N) \tag{4.14}$$

where N is the number of moving nodes, and hence $N + 1$ the number of elements.

In the results given below explicit Euler time-stepping is used throughout, and where results using Miller's method are given it must be emphasised that no penalty functions have been used, parallelism being treated as in Wathen & Baines [5].

Figures (4.1) and (4.2) show typical solutions for the test problem described above at intervals of 0.1 time units, using 10 and 20 moving nodes respectively, with a fixed time step $\Delta t = 0.0005$. The broken lines give the analytic solution at the corresponding times.

Accuracy of Approximation

We look at how the accuracy of the approximation is affected by the time step and by the number of moving nodes used. The error in the approximation is measured in terms of

$$\|U - u\|_{L_2}$$

and

$$\|U_x - u_x\|_{L_2}$$

which are calculated over each element using eight point Gaussian quadrature.

Table (4.1) shows the effect of reducing the time step on the accuracy of the approximation, comparing the solution obtained using Miller's method and by quadratic recovery (QR) from U_x as described in section 3. The results

in Table (4.1) are taken at time $t = 1.0$ and eight moving nodes are used.

Initially $\|U - u\|_{L_2} = 0.006199$

and $\|U_x - u_x\|_{L_2} = 0.25776$.

Δt	$\|U - u\|_{L_2}$		$\|U_x - u_x\|_{L_2}$	
	Q.R.	Miller	Q.R.	Miller
0.008	0.01934	*	0.31407	*
0.004	0.01935	0.02069	0.31360	0.30261
0.002	0.01935	0.02069	0.31335	0.30246
0.001	0.01935	0.02069	0.31332	0.30249
0.0005	0.01935	0.02069	0.31311	0.30226

TABLE 4.1

Results for Miller's method with $\Delta t = 0.008$ are not given as the nodes overtake in this case.

It may be seen from Table 4.1 that decreasing the size of the time step has virtually no effect on the accuracy, and one may take as large a time step as is possible before the nodes overtake with very little loss of accuracy.

Table 4.2 shows the effect of the number of moving nodes used on the accuracy of the approximation. A fixed time step $\Delta t = 0.0005$ is used and the results are given at time $t = 1.0$.

No. of Nodes	T = 0.0		T = 1.0			
	$\|\|U-u\|\|_{L_2}$	$\|\|U_x-u_x\|\|_{L_2}$	$\|\|U-u\|\|_{L_2}$		$\|\|U_x-u_x\|\|_{L_2}$	
			Q.R.	Miller	Q.R.	Miller
20	0.00111	0.10882	0.00795	0.00897	0.14843	0.14676
10	0.00414	0.21224	0.01542	0.01672	0.26343	0.25551
9	0.00498	0.22578	0.01786	0.01918	0.28538	0.27621
8	0.00619	0.25776	0.01935	0.02069	0.31309	0.30229
7	0.00784	0.28412	0.02351	0.02468	0.35126	0.33585
6	0.01030	0.32799	0.02609	0.02733	0.39049	0.37538
5	0.01411	0.38627	0.03457	0.03492	0.47807	0.44394
4	0.02041	0.45195	0.04349	0.04099	0.54259	0.52039

TABLE 4.2

From Table 4.2 it may be seen that, unlike the size of time step, the number of moving nodes used has a significant effect on the accuracy of the approximation.

From Tables 4.1 and 4.2 it may be seen that there is very little difference in the accuracy of the two methods (Q.R. and Miller) used.

Node Overtaking

Table 4.3 gives the maximum explicit Euler time step which may be taken before node overtaking occurs, using both quadratic recovery and Miller's method without penalty functions for various numbers of nodes.

Number of nodes	Max. Δt(Q.R.)	Max. Δt (Miller)
20	0.0008	0.0006
10	0.005	0.00275
9	0.0055	0.003
8	0.0098	0.0045
7	0.0105	0.0055
6	0.0175	0.011
5	0.019	0.011
4	0.0395	0.025
3	0.045	0.024
2	No restriction	No restriction

TABLE 4.3

It was hoped that by using larger numbers of moving nodes to model the second derivative term more accurately the nodes would be less likely to overtake, but it may be seen from Table 4.3 that, for both, the restriction on the time step before the nodes overtake becomes increasingly severe as we seek a more accurate approximation by using more nodes. Comparing the two methods it may also be seen that using the method of quadratic recovery a time step of up to two times larger than that which may be taken using Miller's method is possible before the nodes overtake. Further analysis of the problem of node overtaking is given in the next two sections.

5. ALTERNATIVE TIME STEPPING TECHNIQUES

As shown in the previous section the problem of node overtaking places a restriction on the maximum time step which may be taken, this restriction becoming increasingly severe as we seek a more accurate approximation by using a larger number of elements. As an attempt to overcome this problem we have considered a number of different time stepping schemes as alternatives to the explicit Euler method used in the previous section, these being

(i) Mid-point rule

(ii) Explicit 4th order Runge-Kutta

(iii) Predictor-Corrector schemes.

The same test problem described in the previous section was used to test these alternatives.

The mid-point rule, with starting value given by explicit 4th order Runge-Kutta, gave no improvement over the explicit Euler method.

Results for the 4th order Runge-Kutta method are given in Table 5.1 below.

Number of nodes	Max. Δt(R-K)	Max. Δt (Euler)
20	0.0015	0.008
10	0.012	0.005
9	0.010	0.0055
8	0.015	0.0098
7	0.0175	0.0105
6	0.0275	0.0175
5	0.03	0.019
4	0.0675	0.0395
3	0.075	0.045
2	No restriction	No restriction

TABLE 5.1

Comparing the results using Runge-Kutta with those for the explicit Euler method we see that the maximum time step is increased by at most a factor of about two. At each time step the 4th order Runge-Kutta scheme requires the evaluation of four coefficients, each requiring inversion of the MFE matrix, hence each Runge-Kutta time step requires about four times as much work as an explicit Euler step.

We have also used two different predictor-corrector type schemes as follows. At each time level we solve the MFE equations

$$A(\underline{y})\dot{\underline{y}} = \underline{g}(\underline{y}) \quad .$$

Assuming $\underline{y}^n$ at time level t_n is known we then generate $\underline{y}^{n+1}$ at time level t_{n+1} by one of the two methods given below

a) Predict : $A(\underline{y}^n)(\underline{y}_0^{n+1} - \underline{y}^n) = \Delta t\ \underline{g}(\underline{y}^n)$

Correct : $A(\underline{y}_k^{n+1})(\underline{y}_{k+1}^{n+1} - \underline{y}^n) = \Delta t\ \underline{g}(\underline{y}_k^{n+1})$

$k = 0,1,2,....$ (5.1)

b) Writing the right hand side of the MFE equations in the form $\underline{g}(\underline{y}^n) = c(\underline{y}^n) + \varepsilon d(\underline{y}^n)$, where the contribution to the vector $d(\underline{y}^n)$ is from inner products of the form $\langle U_{xx}, \beta_i^{\alpha_i} \rangle$, then we have:

Predict : $A(\underline{y}^n)(\underline{y}_0^{n+1} - \underline{y}^n) = \Delta t\ (c(\underline{y}^n) + \varepsilon d(\underline{y}^n))$

Correct : $A(\underline{y}^n)(\underline{y}_{k+1}^{n+1} - \underline{y}^n) = \Delta t(c(\underline{y}^n) + \varepsilon\theta d(\underline{y}^n) + \varepsilon(1-\theta)\underline{d}(\underline{y}_k^{n+1}))$

$k = 0,1,2,...$ (5.2)

where $0 \leq \theta \leq 1$.

For (5.1) each iteration of the corrector requires inversion of the MFE matrix, but for (5.2) in which we only correct on the diffusion term $A^{-1}(\underline{y}^n)$ may be stored and used for each correction.

It was hoped that these predictor-corrector schemes would, by adding some degree of implicitness, allow larger time steps to be taken before node overtaking occurred, but in fact no noticeable improvement was achieved, and indeed in some cases the corrector iterations actually produced node overtaking from a predicted solution in which no overtaking was present.

Hence it would appear that, unlike the problem of classical stability in finite difference schemes for diffusion problems, the problem of node overtaking in the MFE method is not reduced by increasing implicitness in the time stepping scheme.

This failure to improve on the maximum time step which may be taken before node overtaking occurs leads us to look more closely in the next section at the effect of the approximation procedure itself on the movement of the nodes.

6. FURTHER ANALYSIS OF NODAL MOVEMENT

The restriction on the size of the time step which may be taken before node overtaking occurs, even when using higher order explicit or predictor-corrector type time-stepping schemes, has led us to analyse further the movement of the nodes, and in particular the effect of our choice of approximation procedure for the second derivative term on this movement. We aim to constrain the movement of the nodes to prevent them overtaking, without having to choose ad hoc free parameters, as is the case when using penalty functions.

Consider again solving the viscous Burger's equation

$$u_t + uu_x = \varepsilon u_{xx} \tag{6.1}$$

and seek as before an approximate solution of the form

$$u(x,t) = \sum_{j=1}^{N} U_j \alpha_j(x,\underline{s}(t)) \ . \tag{6.2}$$

Then

$$U_t = \sum_{j=1}^{N} \dot{U}_j \alpha_j + \dot{s}_j \beta_j$$

$$UU_x = \sum_{j=1}^{N} U_j \frac{\partial U_j}{\partial x} = - \sum_{j=1}^{N} U_j \beta_j \ ,$$

see e.g. [5]. Suppose also we recover U_{xx} with some function w_{xx} where

$$w_{xx} = \sum_{j=1}^{N} c_j \alpha_j + b_j \beta_j \ . \tag{6.3}$$

The approximation (6.2) lies on a finite dimensional non-linear manifold

M in a Hilbert space, and both U_t and w_{xx} are restricted to lie in the tangent space T_u of M at the point U, which is spanned by the basis functions $\{\alpha_j\}$ and $\{\beta_j\}$ (see Fig. 6.1 and Miller [2]). Note that w_{xx} is piecewise linear and possibly discontinuous at the nodes.

Substituting in (6.1) we have

$$\sum_{j=1}^{N} (\dot{U}_j\alpha_j + \dot{s}_j\beta_j - U_j\beta_j) = \varepsilon\sum_{j=1}^{N} (c_j\alpha_j + b_j\beta_j)$$

and hence

$$\sum_{j=1}^{N} (\dot{U}_j - \varepsilon c_j)\alpha_j + \sum_{j=1}^{N} (\dot{s}_j - U_j - \varepsilon b_j)\beta_j = 0 . \tag{6.4}$$

Since the β_j are discontinuous at the nodes s_j the only solution of (6.4) is

$$\dot{s}_j - U_j - \varepsilon b_j = 0$$

and

$$\dot{U}_j - \varepsilon c_j = 0 ,$$

i.e.

$$\begin{aligned} \dot{s}_j &= U_j + \varepsilon b_j \\ \dot{U}_j &= \varepsilon c_j . \end{aligned} \tag{6.5}$$

Now consider the case in which $w(x)$ is recovered from $U(x)$ by fitting a cubic spline. In this case w_{xx} is piece-wise linear and continuous at the nodes, i.e. we have $b_j = 0$ $(j = 1,\ldots,N)$ in (6.3).

Setting $b_j = 0$ in (6.5) yields

$$\begin{aligned} \dot{s}_j &= U_j \\ \dot{U}_j &= \varepsilon c_j . \end{aligned} \tag{6.6}$$

Hence in this type of recovery the $\dot{s}_j$ are the same as for the hyperbolic equation $(\varepsilon = 0)$, and we may expect node overtaking to occur exactly as in the hyperbolic case. It is therefore not possible in this case to control the movement

of the nodes by the choice of coefficients c_j in (6.3).

Next consider the situation when in general $b_j \neq 0$, i.e. when w_{xx} is discontinuous at the nodes. We have

$$w_{xx} = \sum_{j=1}^{N} c_j \alpha_j + b_j \beta_j$$

and hence

$$(w_{xx})_j^- = c_j - b_j m_j$$

$$(w_{xx})_j^+ = c_j - b_j m_{j+1} ,$$

where $$m_j = \frac{\Delta_- U_j}{\Delta_- s_j} .$$

If we define the jump in w_{xx} at node j to be d_j then

$$d_j = (w_{xx})_j^+ - (w_{xx})_j^- = - b_j(m_{j+1} - m_j) = -b_j \Delta_+ m_j .$$

Hence $$b_j = - \frac{d_j}{\Delta_+ m_j}$$

and from (6.5) we have

$$\dot{s}_j = U_j - \frac{\varepsilon d_j}{\Delta_+ m_j} \tag{6.7}$$

$$\dot{U}_j = \varepsilon c_j .$$

Now in order for node j not to overtake node $j+1$ in time Δt we require that

$$\Delta t \, \Delta_+ \dot{s}_j > - \Delta_+ s_j . \tag{6.8}$$

Hence using (6.7) in (6.8) we need

$$\Delta t \, \Delta_+ \left(U_j - \frac{\varepsilon d_j}{\Delta_+ m_j} \right) > -\Delta_+ s_j$$

i.e. $$\varepsilon \Delta_+ \left(\frac{d_j}{\Delta_+ m_j} \right) < \Delta_+ U_j + \frac{\Delta_+ s_j}{\Delta t} . \tag{6.9}$$

It follows that if we choose a fixed time step Δt then we may choose the coefficients d_j so that (6.9) holds, and there will be no node overtaking.

A simple algorithm used to implement this procedure is as follows.

(i) Evaluate d_j $(j = 1,\ldots,N)$ from $U(x,t^n)$. For the quadratic $(DU)(x)$ as defined in section 3 the jumps d_j are given by

$$d_j = \frac{1}{2\Delta_- s_{j+1}} (-3m_j + 4m_{j+1} - m_{j+2})$$

$$- \frac{1}{2\Delta_- s_j} (m_{j-1} - 4m_j + 3m_{j+1}) .$$

For Miller's method, i.e using a Hermite cubic $w(x)$ to recover $w(x)$ from $U(x)$,

$$d_j = \frac{1}{\Delta_- s_{j+1}} (-2m_j + 3m_{j+1} - m_{j+2})$$

$$- \frac{1}{\Delta_- s_j} (m_{j-1} - 3m_j + 2m_{j+1}) .$$

(ii) Modify d_j so as to ensure no overtaking : starting with $j = N-1$ check that

$$\frac{d_j}{\Delta_+ m_j} \geq \frac{d_{j+1}}{\Delta_+ m_{j+1}} - \frac{1}{\varepsilon} \left[\frac{\Delta_+ s_j}{\Delta t} + \Delta_+ U_j \right] + \text{tolerance} \qquad (6.10)$$

with d_N unmodified. If (6.10) does not hold then replace d_j by the value given by equality in (6.10). (If $\Delta_+ m_j = 0$, i.e. in the event of parallelism, set $d_j = 0$ and define $\dot{s}_j$ as for the case $\varepsilon = 0$).

(iii) Update $\dot{s}_j$, $\dot{U}_j$ using

$$\dot{s}_j = U_j - \varepsilon b_j$$

$$\dot{U}_j = \varepsilon c_j$$

where

$$b_j = \frac{-d_j}{\Delta_+ m_j}$$

and $$c_j = \tfrac{1}{2}\left[(w_{xx})_j^+ + (w_{xx})_j^- - \frac{d_j}{\Delta_+ m_j}(m_j + m_{j+1})\right] .$$

Using the notation $(w_{xx})_j^+ = r_j^+$

$$(w_{xx})_j^- = r_j^-$$

then $$c_j = \frac{1}{\Delta_+ m_j}(m_{j+1} r_j^- - m_j r_j^+) .$$

(iv) Set $j = j-1$ and return to (ii).

Although this procedure guarantees that the nodes will not overtake for a given time step Δt, it does not necessarily guarantee preservation of monotonicity in the solution.

It is possible to derive a condition similar to (6.10) such that monotonicity in the solution will be preserved for the fixed time step Δt and we attempted to choose the coefficients d_j such that both conditions were satisfied. In practice it was found that for time steps larger than that used in section 4 monotonicity in the solution was not preserved when the nodes were constrained not to overtake and the choice of a suitable tolerance to satisfy condition (6.10) and that for monotonicity preservation was difficult to achieve.

It is possible to control the movement of the nodes by using a constrained minimisation technique such that the recovered function w_{xx} is restricted to lie in a part of the tangent space spanned by $\{\alpha_j\}$, $\{\beta_j\}$.

If we write

$$w_{xx} = \sum_{j=1}^{N} c_j \alpha_j + b_j \beta_j$$

then w_{xx} lies in the tangent space and we may solve

$$u_t + uu_x = \varepsilon w_{xx} \qquad (6.11)$$

exactly.

The error in solving

$$u_t + uu_x = \varepsilon u_{xx}$$

rather than (6.11) is $\varepsilon(u_{xx} - w_{xx})$, and this may be minimised by choosing c_j, b_j

such that

$$\|U_{xx} - w_{xx}\|^2_{L_2} \qquad \text{is least.}$$

If we control the movement of the nodes by choosing the b_j to satisfy equality constraints, e.g. imposing one of the conditions

$b_j = 0$: nodes move as for the hyperbolic problem

$b_j = -\varepsilon^{-1} U_j$: nodes do not move, $\dot{s}_j = 0$

$$\Delta(U_j + \varepsilon b_j) = -\theta \frac{\Delta s_j}{\Delta t} \qquad 0 < \theta < 1\text{: no overtaking in time } \Delta t,$$

then we minimise $\|U_{xx} - w_{xx}\|^2_{L_2}$ over the c_j only ($j = 1,\ldots,N$), yielding

$$\langle \alpha_j, U_{xx} - w_{xx} \rangle = 0 \qquad (j = 1,\ldots,N) . \tag{6.12}$$

We may write (6.12) in the form

$$A\underline{c} = \underline{g} \tag{6.13}$$

where $\underline{c} = (c_1,\ldots,c_N)^T$.

The matrix A is the standard piece-wise linear finite element mass matrix, with

$$A_{ij} = \langle \alpha_i, \alpha_j \rangle .$$

The right hand side vector $\underline{g}$ is defined by

$$g_i = -\langle \alpha_i, \sum_{j=1}^{N} b_j \beta_j \rangle + \langle \alpha_i, U_{xx} \rangle$$

$$= \frac{1}{6} \Delta U_i b_{i-1} + \frac{1}{3}(\Delta U_i + \Delta U_{i+1}) b_i + \frac{1}{6} \Delta U_{i+1} b_{i+1}$$

$$+ (m_{i+1} - m_i)$$

where the b_i are given by the equality constraints. By using equality constraints we have avoided interpretation of the inner products $\langle \beta_i, U_{xx} \rangle$.

If we minimise $\|U_{xx} - w_{xx}\|^2_{L_2}$ with respect to b_j <u>and</u> c_j ($j = 1,\ldots,N$)

subject this time to inequality constraints of the form

$$\Delta(U_j + \varepsilon b_j) \geq -\frac{\Delta s_j}{\Delta t} \qquad \text{(i.e. no overtaking in time } \Delta t\text{)}$$

then we are minimising a quadratic functional subject to a system of linear inequality constraints; hence we have a quadratic programming problem. Note that minimising over both the b_j and c_j also means that we must interpret the inner products $\langle U_{xx}, \beta_i \rangle$.

We hope to pursue this technique of constrained minimisation at a later date and obtain numerical results.

7. MERGING OF NODES

In this section we introduce a method which treats the problem of overtaking nodes in a different way. This consists of merging nodes when they overtake and, to avoid depletion of nodes, introducing them elsewhere so that the accuracy of the approximation is maintained.

At each time level t^n one may compute, for each node, a time step which will cause the node to overtake : in fact the node $j-1$ will overtake node j in time Δt_j given by

$$\Delta t_j = -\frac{\Delta s_j}{\Delta \dot{s}_j}$$

(if $\Delta \dot{s}_j \geq 0$ then the nodes will not overtake and we set $\Delta t_j = 0$).

We then set

$$\Delta t^* = \min_{j=2,\ldots,N} \Delta t_j \quad . \tag{7.1}$$

In practice it is found that it is not always advantageous to take the time step Δt^* given by (7.1), as too large a time step results in the subsequent Δt^*'s given by (7.1) being very small. To overcome this problem we set a maximum time step Δt_{max} so that if

$$\Delta t^* > \Delta t_{max}$$

then we take a time step Δt_{max}, and the nodes do not overtake, but if

$$\Delta t^* < \Delta t_{max}$$

then we take a time step Δt^* and one of the nodes just overtakes the next, say s_{i-1} and s_i in Fig. 7.1.

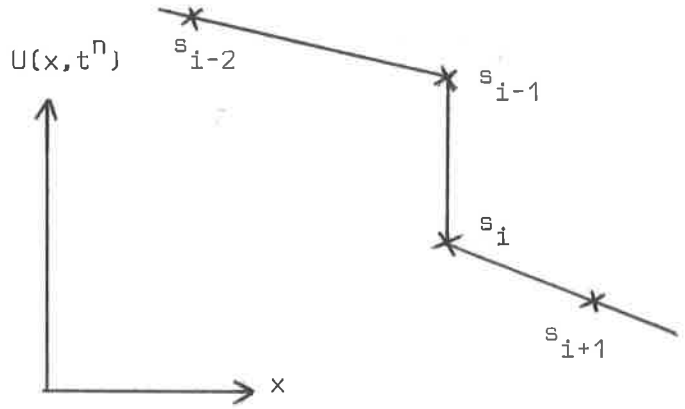

Figure 7.1

We now replace the nodes at s_{i-1} and s_i by a single node at $s_1^* = s_{i-1} = s_i$. In order to choose the nodal amplitude U_i^* corresponding to s_i^* so as to best preserve accuracy in the approximation under consideration we look at the effect of overtaking and merging on the basis functions α_{i-1}, α_i and the consequent new basis function α_i^*. When the nodes overtake we have the situation shown in Fig. 7.2a, and on merging s_{i-1} and s_i we obtain that shown in Fig. 7.2b.

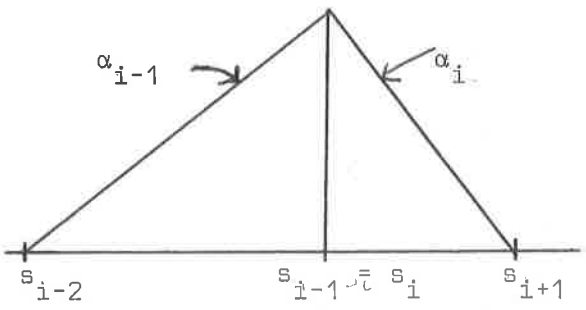

Figure 7.2a

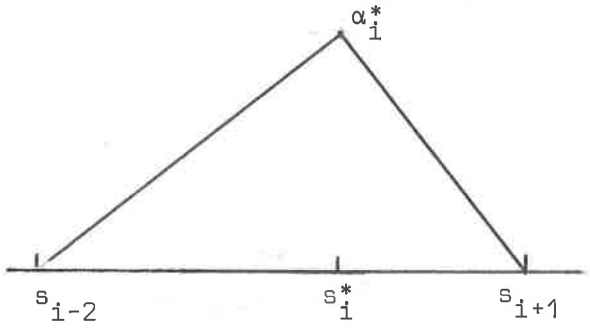

Figure 7.2b

Hence the two basis functions α_{i-1} and α_i have been merged into one basis function α_i^*. We now choose U_i^* to minimise

$$\| U_{i-1}\alpha_{i-1} + U_i\alpha_i - U_i^*\alpha_i^* \|_{L_2}^2 \qquad (7.2)$$

with respect to U_i^*. This gives

$$\langle \alpha_i^*, U_{i-1}\alpha_{i-1} + U_i\alpha_i - U_i^*\alpha_i^* \rangle = 0 .$$

Hence

$$U_{i-1}\langle \alpha_i^*, \alpha_{i-1} \rangle + U_i \langle \alpha_i^*, \alpha_i \rangle - U_i^* \langle \alpha_i^*, \alpha_i^* \rangle = 0$$

and, evaluating the inner products, we obtain

$$\frac{1}{3}\Delta s_i U_{i-1} + \frac{1}{3}\Delta s_{i+1} U_i - \frac{1}{3}(\Delta s_i + \Delta s_{i+1}) U_i^* = 0$$

and

$$U_i^* = \frac{\Delta s_i U_{i-1} + \Delta s_{i+1} U_i}{\Delta s_i + \Delta s_{i+1}} . \qquad (7.3)$$

This is the amplitude set for the merged node.

It is not clear where one should insert a node to replace the node which is deleted in order to best preserve accuracy, so at present we have used the following ad hoc procedure.

Suppose node i has been merged with node $i-1$, and that we have N moving nodes, then

a) if $i > \frac{N}{2}$ we introduce a new node s_N^* such that $s_N^* = s_N + (1-\theta)(s_{N+1} - s_N)$ where $0 < \theta < 1$ and choose U_N^* to lie on the straight line joining U_N, U_{N+1}, as shown in Fig. 7.3.

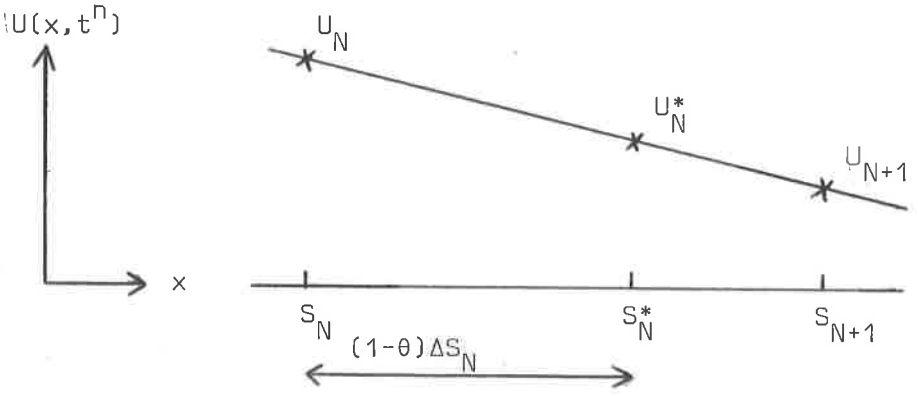

Figure 7.3

b) if $i \leq \frac{N}{2}$ we introduce a new node s_1^* such that $s_1^* = s_1 + \theta(s_1 - s_0)$ where $0 < \theta < 1$ and choose U_1^* to lie on the straight line joining U_0, U_1, as shown in Fig. 7.4.

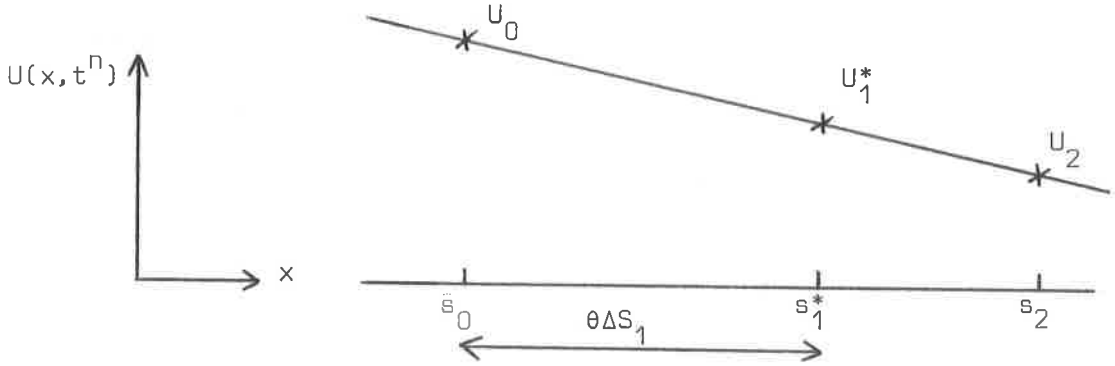

Figure 7.4

This method was used on the test problem described in section 4 using 10 moving nodes, and results are given in Table 7.1 below are taken at time $t = 1.0$. We use the method of quadratic recovery for the U_{xx} term as described previously, and the results show how the accuracy of the approximation is affected by our choice of the parameter θ which determines where new nodes are introduced. As stated above it was found to be beneficial to impose a maximum time step but as the time step varies from step to step the average time step up until $t = 1.0$ is also given.

The accuracy of the approximation and the size of the time step may be comapred with the numerical results in section 4.

Δt max	θ	Average Δt	$\|U - u\|_{L_2}$	$\|U_x - u_x\|_{L_2}$
0.01	0.3	0.00944	0.02145	0.45168
0.01	0.4	0.00926	0.01502	0.40530
0.01	0.5	0.00926	0.01986	0.37183
0.01	0.6	0.00909	0.02206	0.45863
0.01	0.7	0.00885	0.01443	0.29944
0.02	0.3	0.01587	0.02245	0.62161
0.02	0.4	0.00934	0.01087	0.30401
0.02	0.5	0.01408	0.04015	0.63267
0.02	0.6	0.01538	0.00505	0.32620
0.02	0.7	0.01408	0.02539	0.56814

TABLE 7.1

For the sake of comparison with the previous results, using 10 nodes with a fixed time step $\Delta t = 0.0005$, and using the quadratic recovery method (see Table 4.2)

$$\|U - u\|_{L_2} = 0.01542$$

$$\|U_x - u_x\|_{L_2} = 0.26343 \ .$$

The accuracy of the approximation is seen to be highly dependent on the value of the parameter θ , although in some cases the method works very successfully. It is hoped that a more sophisticated means of introducing nodes, combined with this merging technique, may improve the results.

Figure 7.5 gives the solution at $t = 0.0$ and $t = 1.0$ using the method described above, with $\theta = 0.6$ and $\Delta t_{max} = 0.2$.

Figure 7.6 gives a comparison with Figure 7.5 using a fixed time step $\Delta t = 0.005$.

8. CONCLUSION

The introduction of a recovered function to model the second derivative term in diffusion problems, as an alternative to the method of δ-mollification used by Miller, has been successfully introduced in the one-dimensional problem, and may be extended readily to two dimensional problems.

Experiments with different time stepping schemes suggest that we should persevere with explicit Euler time stepping and either control the movement of the nodes by limiting the size of the time step, or use the idea of merging and introducing new nodes in order to allow larger time steps to be taken.

It is hoped that combining the methods of controlling nodal movement and the merging of nodes, together with some more sophisticated method of introducing new nodes may with further investigation provide an effective alternative to the use of penalty functions, which as well as introducing ad hoc parameters also require a stiff solver to cope with the resulting stiffness introduced into the O.D.E.'s.

It is proposed to apply the method to more complicated one-dimensional examples before we extend it into two dimensions.

ACKNOWLEDGEMENTS

I would firstly like to acknowledge the encouragement and assistance given to me throughout this work, and in the writing of this report by my supervisor, Dr. M.J. Baines. I am also indebted to Professor K.W. Morton, who suggested the use of a recovered function, A.J. Wathen and Dr. B.M. Herbst for their help and ideas during numerous discussions.

I would also like to acknowledge the support of an SERC CASE Studentship with A.W.R.E., Aldermaston, and would like to thank Dr. D.P. Rowse of A.W.R.E. for his personal help and interest shown during the work.

REFERENCES

[1] GELINAS, R., DOSS, S. & MILLER, K. The Moving Finite Element Method: Applications to General Partial Differential Equations with Multiple Large Gradients. J. Comp. Phys., 40, (1981), 202-249.

[2] MILLER, K. & MILLER, R., Moving Finite Elements, Part I. SIAM J.N.A., 18, No. 6, 1019-1032, (1981).

[3] MILLER, K., Moving Finite Elements, Part II. SIAM J.N.A., 18, No. 6, 1033-1057, (1981).

[4] HERBST, B. Moving Finite Element Methods for the Solution of Evolution Equations. Ph.D. Thesis, University of The Orange Free State, (1982).

[5] WATHEN, A.J. Moving Finite Elements and Applications to some problems in Oil Reservoir Modelling. University of Reading, Numerical Analysis Report 4/82.

[6] WATHEN, A.J. & BAINES, M.J. On the structure of the Moving Finite Element Equations. University of Reading, Numerical Analysis Report 5/83.

[7] CONCUS, P., GOLUB, G.H. & O'LEARY, D.P. A Generalised Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations. Sparse Matrix Computations Ed. J.R. Bunch & D.J. Rose. Academic Press, (1975).

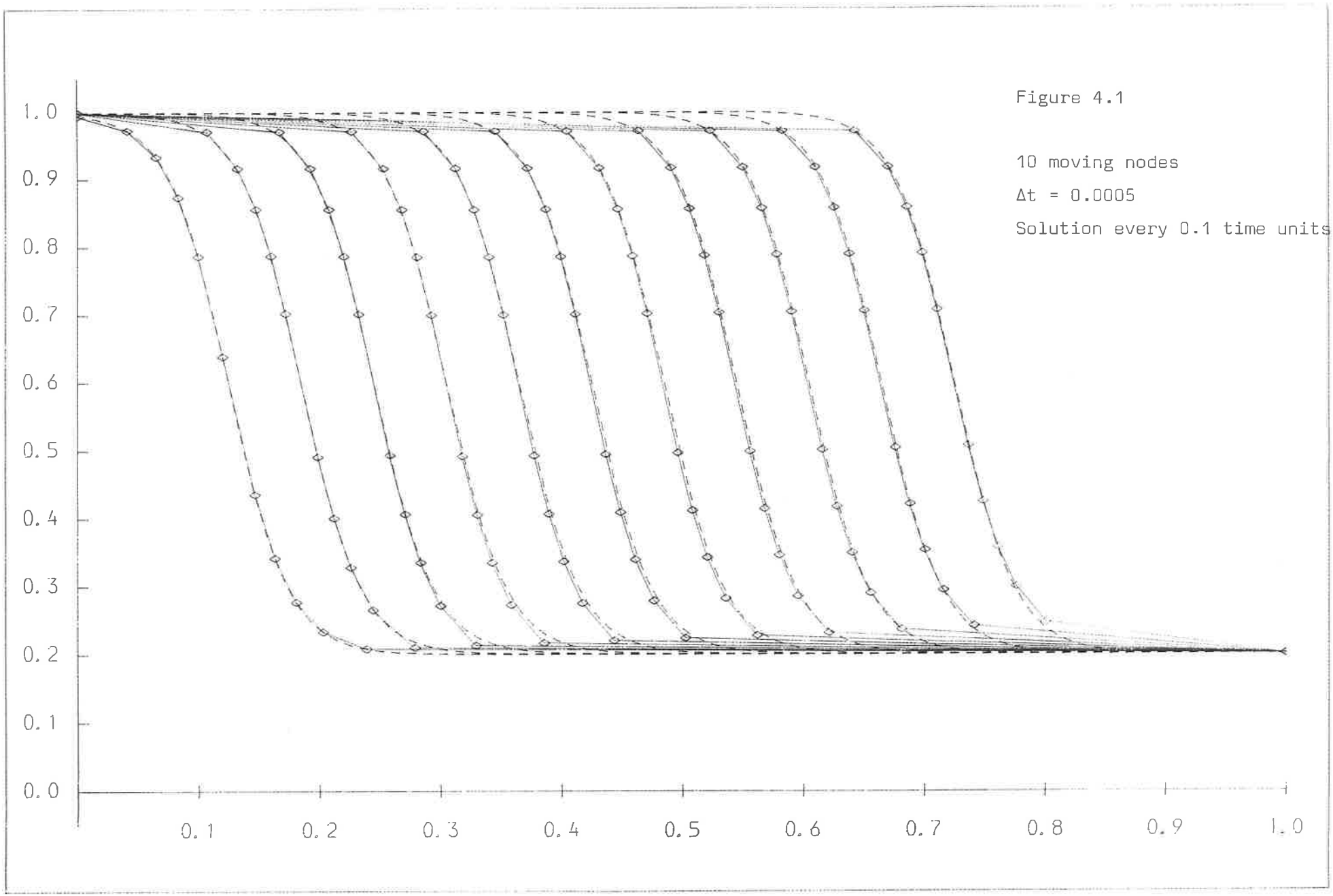

Figure 4.1

10 moving nodes

Δt = 0.0005

Solution every 0.1 time units

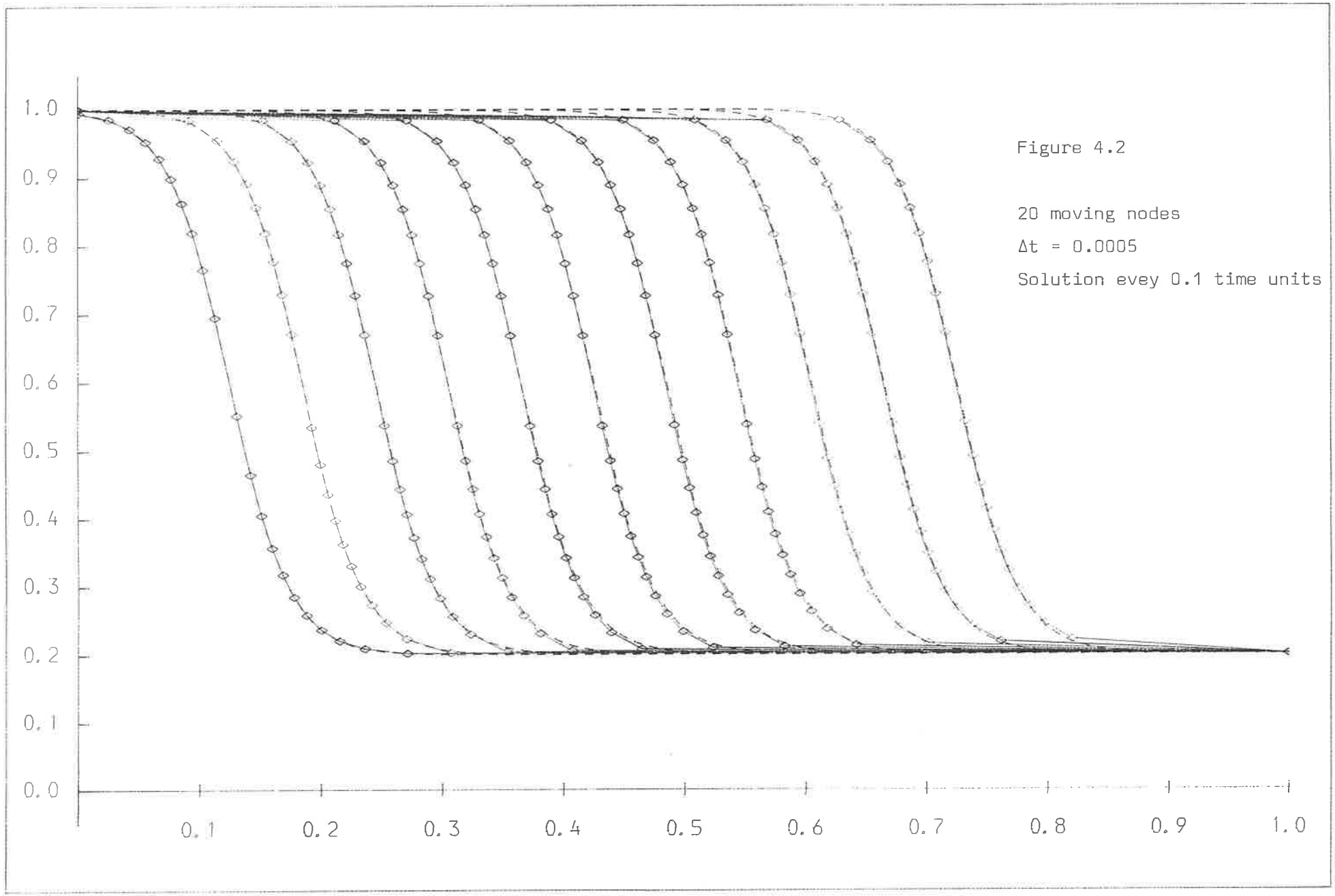

Figure 4.2

20 moving nodes

$\Delta t = 0.0005$

Solution evey 0.1 time units

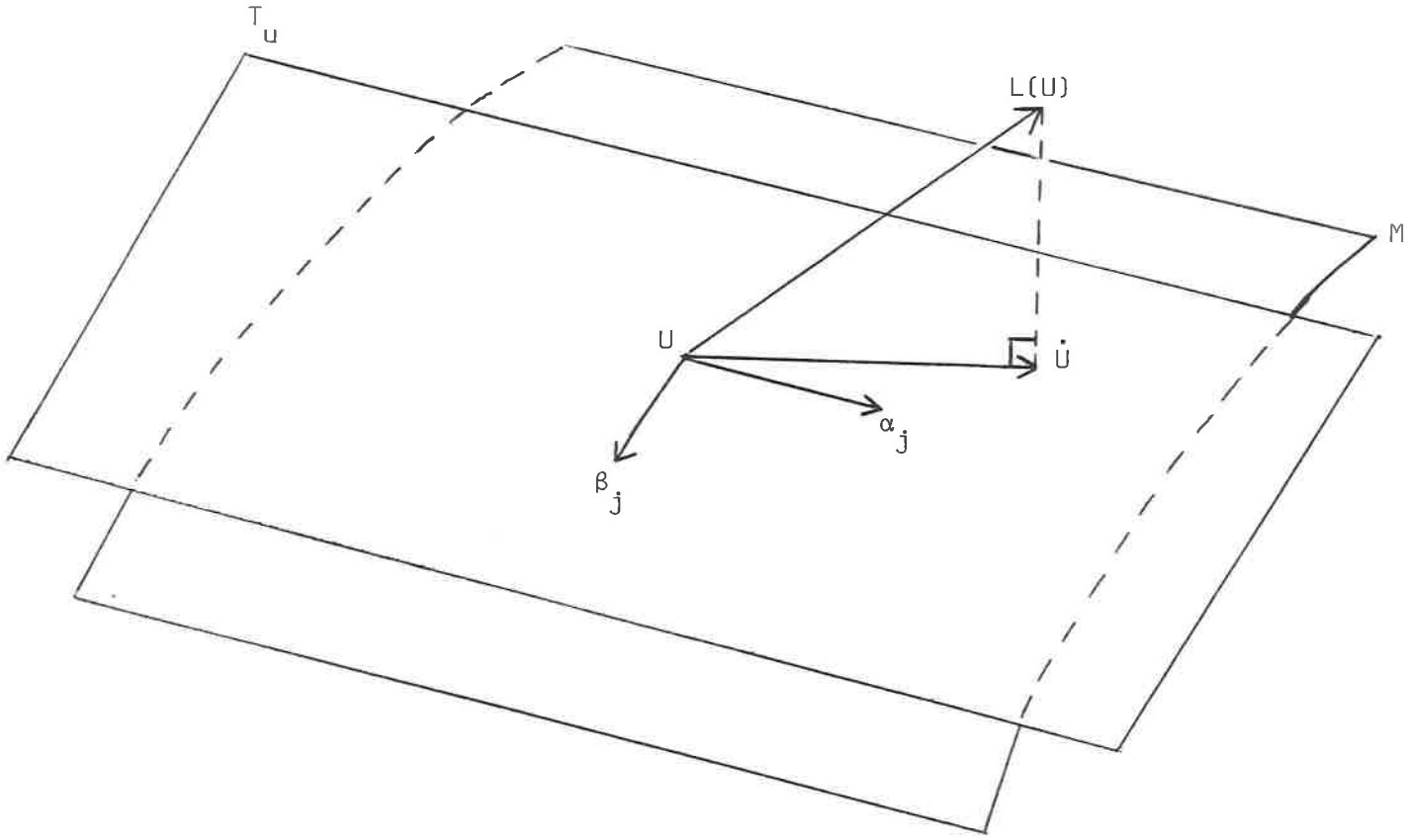

Figure 6.1 : Non-linear manifold M with tangent space T_u

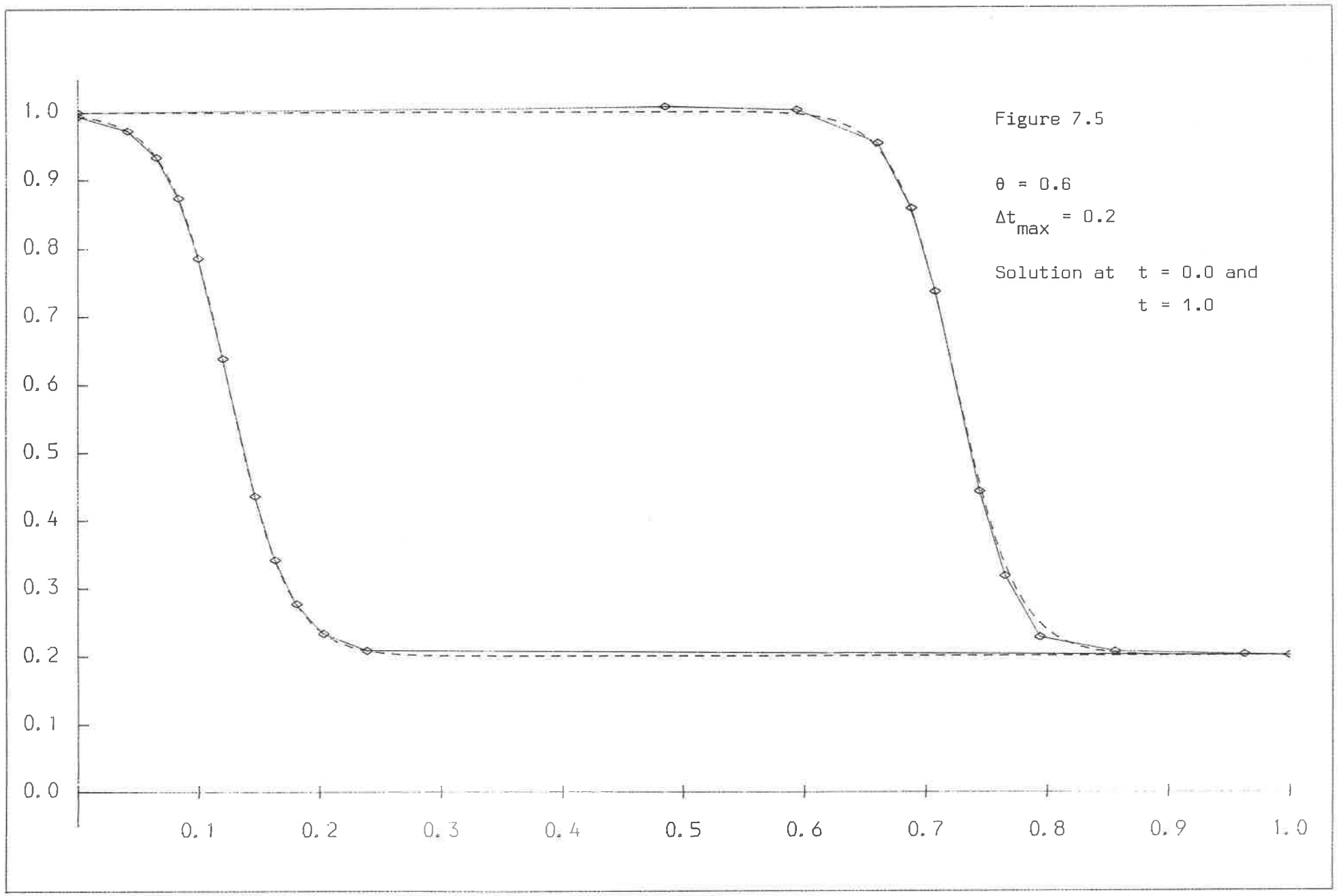

Figure 7.5

$\theta = 0.6$

$\Delta t_{max} = 0.2$

Solution at t = 0.0 and
t = 1.0

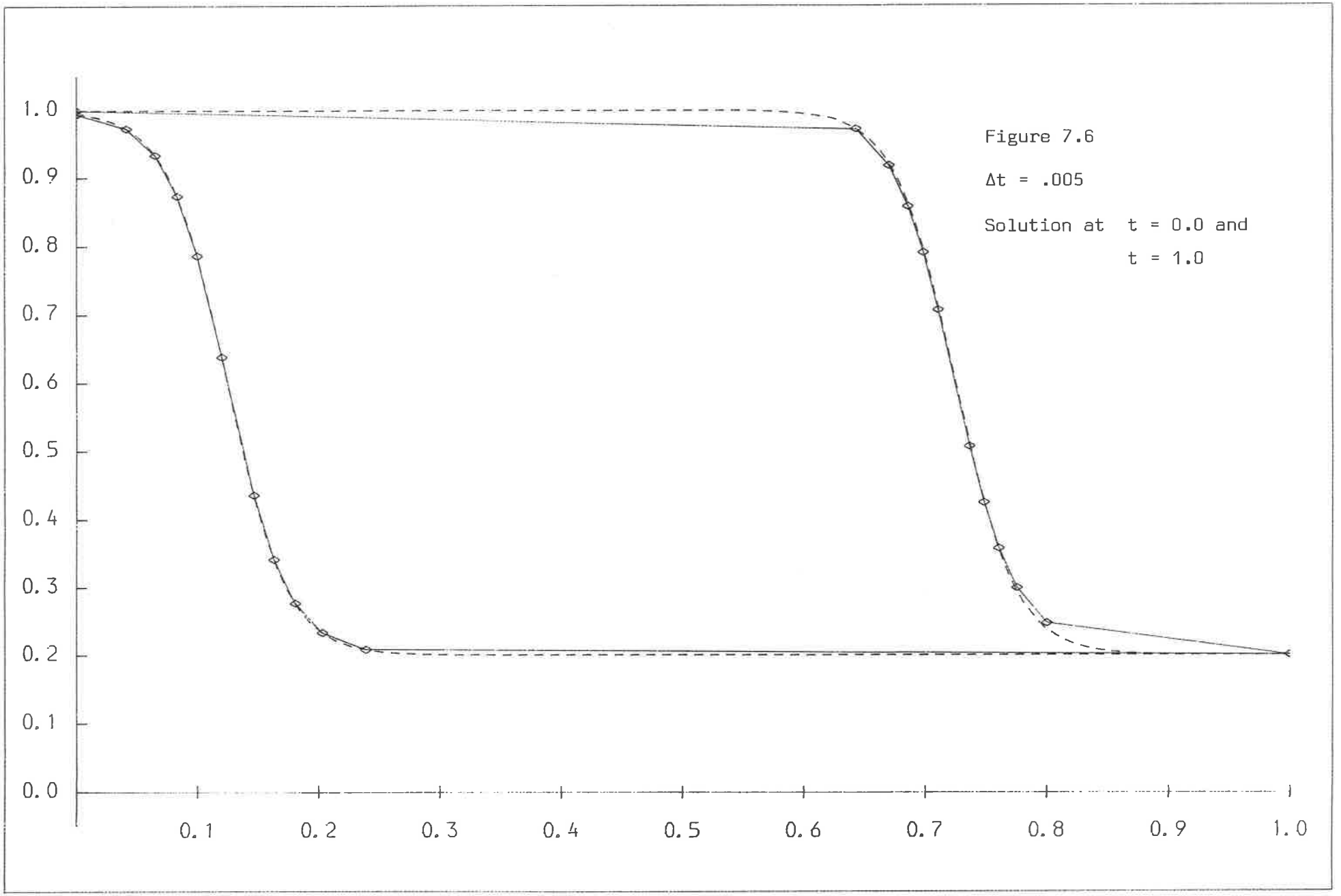

Figure 7.6

$\Delta t = .005$

Solution at $t = 0.0$ and $t = 1.0$