J. Math. Study
doi: 10.4208/jms.v48n2.15.01

Vol. **48**, No. 2, pp. 93-105
June 2015

Explicit Time-stepping for Moving Meshes

M. J. Baines*

Department of Mathematics and Statistics, P O Box 220, University of Reading, RG6 6AX, UK

Received 31 December, 2014; Accepted (in revised version) 23 May, 2015

Abstract. In order to move the nodes in a moving mesh method a time-stepping scheme is required which is ideally explicit and non-tangling (non-overtaking in one dimension (1-D)). Such a scheme is discussed in this paper, together with its drawbacks, and illustrated in 1-D in the context of a velocity-based Lagrangian conservation method applied to first order and second order examples which exhibit a regime change after node compression. An implementation in multidimensions is also described in some detail.

AMS subject classifications: 65M06, 65M20

Key words: PDEs, moving meshes, time stepping, no tangling.

1 Adaptive moving meshes

Moving mesh methods are an alternative (or addition) to fixed mesh adaptive methods in which a given number of mesh points are relocated at each time step (also known as *r*-adaptivity). Relocation may be based on a velocity generated from geometric or physical principles, as in the GCL method [5] and methods based on conservation [1,2], or on a mapping from a reference space to physical space, as in MMPDEs [6,8,9] and Parabolic Monge-Ampere [7] methods. Thus there is a requirement to advance the mesh in time from a given velocity or map.

In numerical implementations the size of the time step is often governed by stability considerations dependent on the numerical method used. A further challenge in advancing the mesh is the avoidance of node overtaking in 1-D or mesh tangling in 2-D. Thus time steps are sought that are not only stable but also preserve the ordering of the nodal positions in 1-D or the integrity of the mesh in higher dimensions.

For example, in one dimension, given a velocity V_j^n at a node X_j^n, $(j=0,\ldots,J)$, at time level n, the explicit Euler time stepping scheme,

$$X_j^{n+1}=X_j^n+hV_j^n, \tag{1.1}$$

*Corresponding author. *Email addresses:* `m.j.baines@reading.ac.uk` (M. J. Baines) where h is the time step, is often used to update the nodes X_j^n, $(j=0,\ldots,J)$, but there is no guarantee that the ordering of the nodes will be preserved. An obvious sufficient *a priori* condition for preserving the ordering of the nodes is easily obtained from the nodal velocities and the node spacing by restricting the time step h to the shortest time that any node X_j^n takes to cross one half of either of the adjacent node spacings i.e.

$$h<\frac{1}{2}\min_j\left|\frac{\Delta X^m_{j\pm 1/2}}{\Delta V^k_{j\pm 1/2}}\right|$$

for all $j=0,\ldots,J$, where $\Delta X_{j\pm 1/2}$ and $\Delta V_{j\pm 1/2}$ denote the differences in X_j and V_j across the interval $j\pm 1/2$, respectively. However, since nodes often move in concert this condition is highly restrictive and usually far from necessary. At the other exteme, a necessary time step for preserving the ordering of the nodes is obtained pragmatically by taking a speculative time step and reducing it if any node overtaking has taken place, but this is a cumbersome process and not conducive to theoretical analysis.

Implicit schemes fare better, but require more work per time step. For example, in [3] a maximum principle is used in one dimension to ensure ordering of the nodes. However, in this paper we shall only be concerned with *explicit* schemes for moving the nodes.

The layout of the paper is as follows. In the next section we introduce an explicit order-preserving scheme in 1-D and discuss its analytic basis and local truncation error. This is followed by an extension of the scheme using a higher order quadrature. In the next section two evolution problems are described to which the schemes may be applied. Numerical examples are given in Section 4 using the Lagrangian moving mesh finite difference scheme of [11, 12]. Finally, in Section 5 the extension to multidimensions is described in detail, with a summary in Section 6.

2 An explicit order-preserving scheme in 1-D

One way of achieving order-preservation of the nodes in 1-D is to focus on the *differences* $\Delta X_{j+1/2}$ between the nodal positions X_j, X_{j+1}. Applying the explicit Euler scheme (1.1) to $\Delta X_{j+1/2}$

$$\Delta X^{n+1}_{j+1/2}=\Delta X^n_{j+1/2}+h\Delta V^n_{j+1/2}=\Delta X^n_{j+1/2}\left(1+h\frac{\Delta V^n_{n+1/2}}{\Delta X^n_{j+1/2}}\right), \tag{2.1}$$

where the bracket in the final term has the status of an amplification factor. If the amplification factor becomes negative then the interval length $\Delta X_{j+1/2}$ changes sign and tangling occurs.

Suppose that the nodes are ordered at time level n so that $\Delta X^n_{j+1/2}$ is positive for all j. Then, if $\Delta V^n_{j+1/2}$ is also positive for all j, the amplification factor in (2.1) is positive and $\Delta X_{j+1/2}$ remains positive after a time step, thus preserving the ordering of the nodes.

Similarly, in the case of the scheme

$$\Delta X^{n+1}_{j+1/2} = \Delta X^n_{j+1/2}\left(1 - h\frac{\Delta V^n_{j+1/2}}{\Delta X^n_{j+1/2}}\right)^{-1} \tag{2.2}$$

it is clear that if $\Delta X^n_{j+1/2}$ is positive and $\Delta V^n_{j+1/2}$ is negative for all j then $\Delta X^{n+1}_{j+1/2}$ is positive and the scheme again preserves the ordering of the nodes.

A time-stepping scheme which incorporates both of these properties (and coincides with the explicit Euler scheme to first order) is

$$\Delta X^{n+1}_{j+1/2} = \Delta X^n_{j+1/2}\exp\left(h\frac{\Delta V^n_{j+1/2}}{\Delta X^n_{j+1/2}}\right), \tag{2.3}$$

where the amplification factor is the exponential. Because the exponential is always positive the sign of $\Delta X_{j+1/2}$ is unchanged in a time step regardless of the sign of $\Delta V^n_{j+1/2}$, thus preserving the ordering of the nodes.

Reconstruction of the nodal positions from the differences $\Delta X^{n+1}_{j+1/2}$ is straightforward. Given X^{n+1}_j at one point, X^{n+1}_0 say,

$$X^{n+1}_j = X^{n+1}_0 + \sum_{k=0}^{j-1}\Delta X^{n+1}_{j+1/2}. \tag{2.4}$$

The analytic basis of the scheme is as follows. Given that the equation being approximated is

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v(x,t), \tag{2.5}$$

by writing the space derivatives of x and v as x_ξ and v_ξ (in terms of a fixed reference coordinate ξ),

$$\frac{\mathrm{d}x_\xi}{\mathrm{d}t} = v_\xi, \quad \text{or} \quad \frac{\mathrm{d}\log x_\xi}{\mathrm{d}t} = \frac{v_\xi}{x_\xi}. \tag{2.6}$$

Integrating the second of (2.6) from t to $t+h$,

$$\log x_\xi(t+h) - \log x_\xi(t) = \log\left(\frac{x_\xi(t+h)}{x_\xi(t)}\right) = \int_t^{t+h}\frac{v_\xi(\tau)}{x_\xi(\tau)}\mathrm{d}\tau, \tag{2.7}$$

so that

$$x_\xi(t+h) = x_\xi(t)\exp\left(\int_t^{t+h}\frac{v_\xi(\tau)}{x_\xi(\tau)}\mathrm{d}\tau\right), \tag{2.8}$$

in which $x_\xi(t+h)$ has the same sign as $x_\xi(t)$. It is this equation that is discretised in (2.3) by a low order quadrature of the integral.

The reconstruction (2.4) of the X_j^{n+1} from the ΔX_j^{n+1} is a discretisation of the integral

$$x(\xi,t)=\int_{\xi_0}^{\xi} x_{\xi'}\,\mathrm{d}\xi', \tag{2.9}$$

at time t, given x at $\xi=\xi^0$ say.

The scheme (2.3) also arises from numerically integrating (2.7), giving

$$\log\left(\frac{\Delta X_{j+1/2}^{n+1}}{\Delta X_{j+1/2}^{n}}\right)=h\frac{\Delta V_{j+1/2}^{n}}{\Delta X_{j+1/2}^{n}}. \tag{2.10}$$

2.1 Truncation and quadrature errors

The local truncation error (LTE) of the scheme (2.3) applied to the first of (2.6) is

$$T_1=\frac{1}{h}\left\{x_\xi(t+h)-x_\xi(t)\exp\left(\frac{hv_\xi(t)}{x_\xi(t)}\right)\right\}.$$

Expanding in powers of h,

$$T_1=\frac{1}{h}\left\{x_\xi(t)+hx'_\xi(t)-x_\xi(t)\left(1+\frac{hx'_\xi(t)}{x_\xi(t)}\right)\right\}+O(h), \tag{2.11}$$

which is of order $O(h)$, the same as for the explicit Euler scheme. Similarly, the LTE of (2.10) is

$$T'_j=\frac{1}{h}\left\{\log\left(\frac{x_\xi^{n+1}}{x_\xi^n}\right)-h\frac{v_\xi^n}{x_\xi^n}\right\}=\frac{1}{h}\left\{\log\left(\frac{x_\xi^n+hv_\xi^{n+1}}{x_\xi^n}\right)-h\frac{v_\xi^n}{x_\xi^n}\right\} \tag{2.12}$$

which is also of order $O(h)$.

2.1.1 Higher order quadrature

A higher order explicit order-preserving scheme in 1-D can be constructed using a Runge Kutta (RK) approach to the second of (2.6), giving instead of (2.10)

$$\log\left(\frac{\Delta X_{j+1/2}^{n+1}}{\Delta X_{j+1/2}^{n}}\right)=\frac{h}{2}(K_1+K_2)_{j+1/2}^{n} \tag{2.13}$$

where $K_1=\Delta V/\Delta X$ and K_2 is $\Delta V/\Delta X$ evaluated at $\Delta X+hK_1$. The scheme corresponding to (2.3) is then

$$\Delta X_{j+1/2}^{n+1}=\Delta X_{j+1/2}^{n}\exp\left(\frac{h}{2}\frac{(K_1+K_2)}{\Delta X}\right)_{j+1/2}^{n} \tag{2.14}$$

which has the same monotonicity property as (2.3). However, the LTE of the scheme (2.14) applied to the first of (2.6) is

$$T_2 = \frac{1}{h}\left\{ x_\xi(t+h) - x_\xi(t)\exp\left(\frac{h}{2}\frac{(k_1(t)+k_2(t))}{x_\xi(t)}\right)\right\} \tag{2.15}$$

where $k_1(t) = x'_\xi(t)/x_\xi(t)$ and $k^*_2(t)$ is $x'_\xi(t)/x_\xi(t)$ evaluated at $x_\xi(t)+hx'_\xi(t)$, leading to

$$\frac{1}{h}\left\{ x_\xi + hx'_\xi + \frac{1}{2!}h^2 x''_\xi - x_\xi\left(1+\frac{h}{2x_\xi}\{2x'_\xi + hx''_\xi\} + \frac{1}{2!}h^2(x'_\xi)^2\right)\right\} + O(h^2) \tag{2.16}$$

which is only of $O(h)$ (first order) even though the integral in (2.8) has been approximated by a higher order integration scheme. Neverthless, the scheme (2.14) is a second order scheme for the second of (2.6).

2.2 Effect of rounding error

In principle, any positive value of the time step h is allowed in the schemes (2.3) and (2.14) when preserving the node ordering. However, when used with inexact arithmetic this is no longer the case.

If $\Delta X^n_{j+1/2}$ falls below the level of rounding error (as is likely when $\Delta V^n_{j+1/2} < 0$ so that adjacent nodes approach one another), the positivity of $\Delta X_{j+1/2}$ may not be maintained in a time step due to the random nature of rounding error, ϵ say. To avoid this difficulty a simple regularisation is to raise $\Delta X^n_{j+1/2}$ in (2.3) or (2.14) above the level of rounding error to $(\Delta X^n_{j+1/2} + |\epsilon|)$, thus converting (2.3) for example to

$$\Delta X^{n+1}_{j+1/2} = (\Delta X^n_{j+1/2} + |\epsilon|)\exp\left(h\frac{\Delta V^n_{j+1/2}}{(\Delta X^n_{j+1/2} + |\epsilon|)}\right). \tag{2.17}$$

In practice this regularisation has an insignificant effect on the computations.

2.3 Use in conjunction with another equation

When used in conjunction with another equation, such as the PDE whose solution is required, large values of h can generate unwanted numerical features. Although node overtaking is avoided there is no guarantee that the nodal spacings remain smooth, thus it is possible to generate oscillatory behaviour in the related equation as a knock-on effect.

Provided that ΔX_j and ΔV_j are smooth at time level n, the source of oscillatory behaviour in ΔX_j at time level $(n+1)$ is the spatial variation of the factor e^Q where Q is the quadrature error in the approximation of the integral in (2.8). Although this quadrature error is small it is not necessarily smooth and will be exaggerated by large time steps. The resulting oscillations can be reduced by a post-processing of the ΔX_j's by a a Laplacian smoother which filter out the oscillations without invalidating the monotonicity property. For example, the filter

$$\Delta X_j^{n+1} = \frac{1}{4}\left(\Delta X_{j-1}^{n+1} + 2\Delta X_j^{n+1} + \Delta X_{j+1}^{n+1}\right) \tag{2.18}$$

supresses the sawtooth component of ΔX_j^{n+1}, giving a smoother nodal spacing. In effect this smoother is a filter on the quadrature error.

3 Lagrangian conservation

A velocity-based method to which these schemes may be applied is the Lagrangian conservation method of [2,12] for a flux-driven mass-conserving PDE problem. The velocity $v(x,t)$ is provided in terms of a flux function, $f(u,u_x)$ say, of the solution $u(x,t)$ by the flux balance equation

$$[-f(u,u_x) + uv] = 0 \tag{3.1}$$

where $[\cdot]$ denotes the jump in the argument across any two moving points in the domain. The function $f(u,u_x)$ depends on t through its dependence on $u(x,t)$ and $u_x(x,t)$. The evolution of a general moving coordinate $\widehat{x}(t)$ is determined by integrating

$$\frac{\mathrm{d}\widehat{x}}{\mathrm{d}t} = v(\widehat{x},t), \tag{3.2}$$

and the solution $u(\widehat{x},t)$ (required to be positive) found from the Lagrangian form of the conservation law,

$$\int_{\widehat{x}_1(t)}^{\widehat{x}_2(t)} u(x,t)\,\mathrm{d}x \text{ is constant in time,} \tag{3.3}$$

for any two arbitrary moving points $\widehat{x}_1(t), \widehat{x}_2(t)$.

Given a zero net mass flux boundary condition on v (such that $(-f(u,u_x)+uv=0)$, then from (3.1) the velocity at a general moving point $\widehat{x}(t)$ is

$$v(\widehat{x},t) = \left.\frac{f(u,u_x)}{u}\right|_{x=\widehat{x}(t)} \tag{3.4}$$

provided that $u(\widehat{x},t) \neq 0$.

Two instances of the flux function $f(u,u_x)$ occur in the following examples:

1. the well-known inviscid Burgers equation, for which the flux function is

$$f(u,u_x) = \frac{1}{2}u^2 \tag{3.5}$$

and the Lagrangian velocity, from (3.4), is

$$v(x,t) = \frac{1}{2}u(x,t). \tag{3.6}$$

(Note that the velocity differs from the standard *characteristics* velocity $v(x,t)=u(x,t)$, for which $u(x,t)$ remains constant in time). A significant feature of solutions is the formation of a shock in finite time.

2. a porous medium equation [13], for which the flux function is

$$f(u,u_x)=-u^2u_x \tag{3.7}$$

and the velocity is $v=-uu_x$. Important features of the solution of this problem are the existence of a waiting time before the boundary moves, and the infinite slope at the boundary when movement takes place.

4 Numerical experiments

We now show the results of numerical experiments carried out on the inviscid Burgers problem and porous medium problem described in the previous section using a finite difference form of the Lagrangian conservation method [4,11,12] to move the nodes.

4.1 The finite difference Lagrangian conservation method

Within a timestep the algorithm is as follows:

1. Approximate the velocity $v(x,t)$ using (3.4) by $V_j^n=f(U_j^n,(U_x)_j^n)/U_j^n$ and an upwind discretisation of $(U_x)_j^n$.
2. Advance the nodes X_j^n to the next time level $n+1$ by the timestepping scheme.
3. Determine the approximate solution U_j^{n+1} at the next time level using an approximation to (3.3) as follows:

   In the case of the inviscid Burgers equations approximate the integral (3.3) using the first order upwind quadrature

$$(X_j^{n+1}-X_{j-1}^{n+1})U_j^{n+1}=C_j \tag{4.1}$$

   where the C_j are constants (in time) prescribed by the initial data, at t^0 say, in the form

$$C_j=(X_j^0-X_{j-1}^0)U_j^0.$$

   In the case of the porous medium equation approximate the integral (3.3) by the midpoint quadrature

$$(X_{j+1}^{n+1}-X_{j-1}^{n+1})U_j^{n+1}=C_j', \tag{4.2}$$

   where the C_j' are constants (in time) prescribed from initial data by

$$C_j'=(X_{j+1}^0-X_{j-1}^0)U_j^0.$$

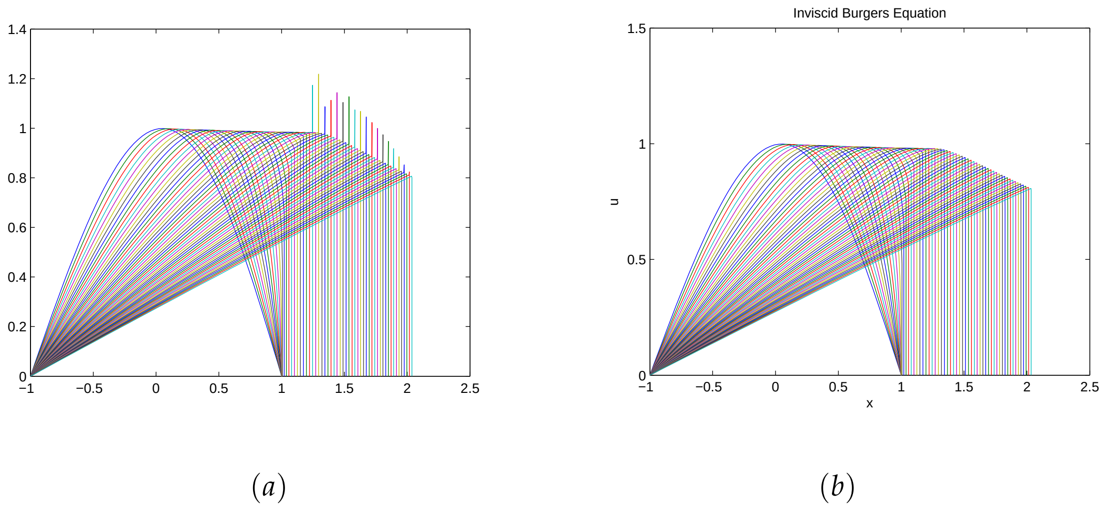

Figure 1: Time series of the evolution of the solution to the Inviscid Burgers Equation (flux fuunction (3.5)) with 61 nodes and timesteps $h=0.01$ at time intervals 0.05 from $t=0$ to $t=3$ using the Lagrangian conservation method with (a) the time-stepping scheme (2.3) and (b) the time-stepping scheme (2.14).

4.1.1 The inviscid Burgers equation

In this example $f(u)=\frac{1}{2}u^2$ and $v=\frac{1}{2}u$, from (3.4).

We take initial data $u(x,0)=\cos(\pi x/2)$ in $|x|<1$ and boundary conditions $u=v=0$ at $x=-1$ with $u=0$ at the downwind boundary. Conservation holds both globally and locally in this method.

Interest lies in the capacity of the method to follow the solution through the compression occurring in $(0<x<1)$ (in which the intervals between the nodes become extremely small) through to the formation of the shock (after which the method relies on local conservation) with the explicit order-preserving time-stepping schemes.

The scheme (2.17) takes the form

$$\Delta X^{n+1}_{j+1/2}=(\Delta X^n_{j+1/2}+|\epsilon|)\exp\left(\frac{h}{2}\frac{\Delta U^n}{\Delta X+|\epsilon|}\right)^n_{j+1/2}, \tag{4.3}$$

while in the scheme (2.14) $K_1=\frac{1}{2}\Delta U/\Delta X$ and K_2 is $\frac{1}{2}\Delta U/\Delta X$ evaluated at $\Delta X+hK_1$. The regularising parameter ϵ is at the level of rounding error.

The initial domain $|x|\le 1$ is discretised by 61 equispaced nodes X_j, $(j=0,\ldots,J)$ and the initial values of U_j at X_j are sampled directly from the initial condition at the nodes. The time step is $h=0.01$.

We show results for the two schemes (2.3) and (2.14) in Figs. 1(a) and 1(b), respectively. Each figure shows the evolution of the initial data from time $t=0$ to time $t=5$ at intervals 0.5 (five times the time step h for clarity).

The main difference between the figures is the suppression of the oscillations that occur in scheme (4.3) by the scheme (2.14) at the formation of the shock. The scheme not only copes well with the compression of the nodes as they approach the shock but also provides a smooth transition to the post-shock behaviour.

We note the slight decay in amplitude of the wave at early times due to the low order upwind approximation used for the integral (3.3).

The most striking thing about Figure 1 is the ease with which the scheme handles the compression of the nodes as they approach the infinite slope and the smooth transition from the waiting time regime to the moving boundary regime. A specimen grid history is shown in Fig. 2.

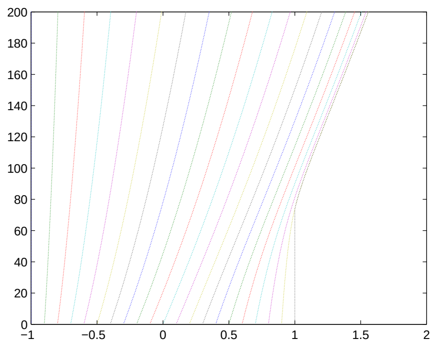

Figure 2: Grid history of the evolution of the solution to the Burgers' equation problem with 21 nodes and timesteps $h=0.01$ at all time intervals from $t=0$ to $t=2$ using the scheme (4.4).

4.1.2 The porous medium equation

In this example $f(u)=-u^2u_x$ and $v=-uu_x=-\frac{1}{2}(u^2)_x$ from (3.4).

We take initial data $u(x,0)=\cos(\pi/2)x$ in $|x|<1$ and zero net flux (and therefore mass conserving) boundary conditions at $x=\pm1$. Conservation holds both globally and locally in this method.

Interest lies in the capacity of the method to follow the solution during the waiting time [10]) (in which the intervals between the nodes become extremely small) through to the spontaneous movement of the boundary (which occurs when the slope becomes infinite).

Again, the initial domain $|x|\le 1$ is discretised by 61 equispaced nodes X_j, $(j=0,\dots,J)$ and the initial values of U_j at X_j are sampled directly from the initial condition at the nodes. The time step is again $h=0.01$.

The scheme (2.17) takes the form

$$\Delta X^{n+1}_{j+1/2}=(\Delta X^n_{j+1/2}+|\epsilon|)\exp\left(-\frac{h}{2}\frac{\Delta(U^2)_x}{\Delta X+|\epsilon|}\right)^n_{j+1/2} \tag{4.4}$$

while in the scheme (2.14) $K_1=-\frac{1}{2}\Delta(U^2)_x/\Delta X$ and K_2 is $-\frac{1}{2}\Delta(U^2)_x/\Delta X$ evaluated at $\Delta X+hK_1$. Here $\Delta(U^2)_x$ is discretised as the barycentric average of the slopes $\Delta(U^2)/\Delta X$ in the adjacent intervals.

We show results for the scheme (2.3) in Fig. 3. The figure shows the evolution of the initial data from time $t=0$ to time $t=3$ at intervals 0.2. The results for the scheme (2.14) are very similar.

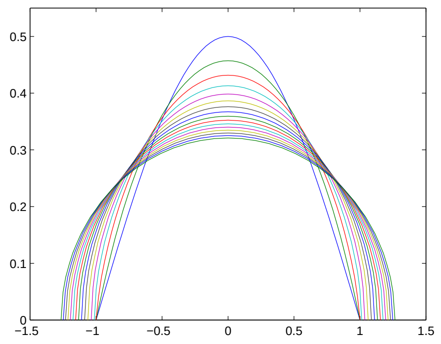

Figure 3: Time series of the evolution of the solution to the Porous Medium Equation (flux function (3.7)) with 61 nodes and timesteps $h=0.01$ at time intervals 0.2 from $t=0$ to $t=3$ using the scheme (4.4).

A specimen grid history is shown in Fig. 4 (on the right hand half of the domain), in which the transition of the boundary point from waiting to movement is clearly seen.

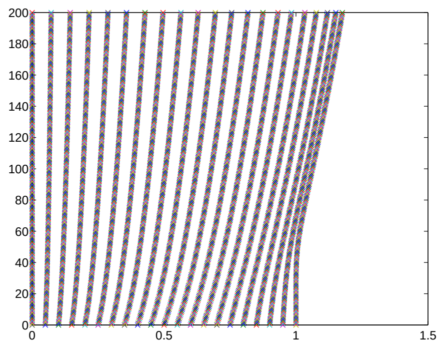

Figure 4: Grid history of the evolution of the solution to the porous medium equation problem with 21 nodes (for the right hand half of the domain) for timesteps $h=0.01$ at all time intervals from $t=0$ to $t=2$ using the scheme (4.4).

5 Multidimensions

We now discuss the extension of the procedure to multidimensions. As in the one-dimensional case there are two stages to the approach. First, given the velocities of the nodes of a multidimensional mesh, we can compute relative velocities along mesh edges which can be used to compute new positive edge lengths using (2.3) or (2.14).

Secondly, given the complete set of positive edge lengths, we would like to use them to compute the nodal positions in such a way as to avoid mesh tangling. This aspect is the multidimensional equivalent of (2.9). To prepare for the generalisation write equation (2.9) as

$$\frac{X_{j+1}-X_j}{\Delta X_{j+1/2}}=\frac{X_j-X_{j-1}}{\Delta X_{j+1/2}} \tag{5.1}$$

where the interval lengths $\Delta X_{j\pm1/2}$ are known but the nodal positions X_j are not. Note that (5.1) is equivalent to equating the sum of the signed unit vectors in the two adjacent intervals to zero. The system of equations (5.1) for all j is implicit but can be written as a matrix equation where one value of X_j is prescribed for a unique solution. Equation (5.1) can also be written in the form

$$X_j=\frac{(\Delta X_{j-1/2})^{-1}X_{j-1}+(\Delta X_{j+1/2})^{-1}X_{j+1}}{(\Delta X_{j-1/2})^{-1}+(\Delta X_{j+1/2})^{-1}} \tag{5.2}$$

showing that X_j is a barycentric average of its neighbours.

An iterative form of (5.1) (using an iteration index p) is

$$X_j^{p+1}=X_j^p+\phi\left\{(\Delta X_{j+1/2})^{-1}(X_{j+1}-X_j)^p-(\Delta X_{j+1/2})^{-1}(X_j-X_{j-1})^p\right\} \tag{5.3}$$

where the relaxation factor $\phi<\frac{1}{2}$. Since the weights $(\Delta X_{j-1/2})^{-1}$ are positive the node at X_j^{p+1} always lies between the midpoints of the intervals adjacent to X_j^p.

Straightforward generalisations of (5.1) and (5.2) to multidimensions are

$$\sum_{jk}\frac{\mathbf{X}_{jk}-\mathbf{X}_j}{\Delta X_{jk}}=0 \tag{5.4}$$

and

$$\mathbf{X}_j=\frac{\sum_{jk}(\Delta X_{jk})^{-1}\mathbf{X}_{jk}}{\sum_{jk}(\Delta X_{jk})^{-1}} \tag{5.5}$$

where jk is the index of nodes $\mathbf{X}_{jk}$ neighbouring node $\mathbf{X}_j$, the ΔX_{jk} being the known positive edge lengths joining $\mathbf{X}_j$ to $\mathbf{X}_{jk}$. Note that equation (5.4) is equivalent to equating the resultant of the unit vectors measured outwards from $\mathbf{X}_j$ along these edges to zero (which is the crux of the procedure). Unlike the one-dimensional case there is no unique solution to the system (5.4). Instead, we minimise a norm of the residual of (5.4). If the norm is the l_2 norm the minimiser is given by equation (5.5).

The implicit system of equations (5.4) for all j can be written as a matrix equation, where at least one value of $\mathbf{X}_j$ is prescribed for a unique solution.

An iterative form of (5.5) (using an iteration index p) is

$$\mathbf{X}_j^{p+1}=\mathbf{X}_j^p+\phi\frac{\sum_{jk}(\Delta X_{jk})^{-1}(\mathbf{X}_{jk}-\mathbf{X}_j)^p}{\sum_{jk}(\Delta X_{jk})^{-1}} \tag{5.6}$$

(cf. (5.3)), where ϕ is a relaxation factor. It also has (5.4) as its limit (if it converges) and in addition has the property that at each iteration the node at $\mathbf{X}_j^{p+1}$ lies in the convex hull of the surrounding nodes scaled by the relaxation factor ϕ, thus enforcing non-tangling if ϕ is sufficiently small. Taking $\phi \leq 1/2$ so that $\mathbf{X}_j^{p+1}$ lies in the convex hull of the midpoints of the edges through $\mathbf{X}_j^p$ is sufficient in most cases for non-tangling, although smaller values of ϕ may be used in extreme cases.

6 Summary

In this paper we have discussed some explicit time-stepping strategies for moving mesh methods.

The strategy suggested in this paper is a two-step approach which relies on interval lengths in one dimension and edge lengths in multidimensions. The basic building brick is a scheme which evolves lengths numerically in such a way as to keep them positive. This is achieved by a scheme which uses an amplification factor in the form of an exponential, which has the same order of accuracy as the explicit Euler scheme. The accuracy of the scheme is increased by a higher order quadrature. The second step is to construct a mesh from these positive lengths. In one dimension this is straightforward when the position of one node is known, but in higher dimensions a special principle is required. Generalising from one dimension we propose that the resultant of the unit vectors along the edges emanating from a node be minimised in order to locate the node from the edge lengths.

There are some drawbacks in the implementation of the scheme. Rounding errors can upset the positivity of the amplification factor in the first step and, more seriously, a lack of smoothness of the intervals or edge lengths can engender oscillations when used in conjunction with the numerical approximation of another equation which may invalidate schemes relying on a positive monitor function.

Numerical illustrations are shown for a first and second order problem, each of which exhibits a compression followed by a regime change.

7 Acknowlegement

I should like to thank Niall Arthurs for useful discussions.

References

[1] M. J. Baines, M. E. Hubbard and P. K. Jimack. A Moving Mesh Finite Element Algorithm for the Adaptive Solution of Time-Dependent Partial Differential Equations with Moving Boundaries. Applied Numerical Mathematics, 54: 450–469, 2005.

[2] M. J. Baines, M. E. Hubbard and P. K. Jimack. Velocity-based Moving Mesh Methods for Nonlinear Partial Differential Equations with Moving Boundaries. Commun. Comput. Phys., 10: 509–576, 2011.

[3] M. J. Baines and T. E. Lee. A large time step implicit moving mesh scheme for moving boundary problems. Numerical Methods for Partial Differential Equations, 30: 321–338, 2014.

[4] K. W. Blake and M. J. Baines. Moving mesh methods for nonlinear partial differential equations. Numerical Analysis Report 7/01, Dept of Mathematics, University of Reading, UK, 2001.

[5] W. Cao, W. Huang and R. D. Russell. A Moving Mesh Method Based on the Geometric Conservation Law. SIAM Journal of Scientific Computing, 24: 118–142, 2002.

[6] C. Budd, W. Huang and R.Russell. Adaptivity with Moving Grids. Acta Numerica, 18: 111–241, 2009.

[7] C. Budd and J. Williams. Moving Mesh Generation using the Parabolic Mong-Ampere equation. SIAM Journal on Scientific Computing, 31: 3438–3465, 2009.

[8] W. Huang, Y. Ren and R. D. Russell. Moving Mesh Method Partial Differential Equations Based on the Equidistribution Principle. SIAM Journal of Numerical Analysis, 31: 709–730, 1994.

[9] W. Huang and R. D. Russell. Adaptive Moving Mesh Methods. Springer, 2011.

[10] W. L. Kath and D. S. Cohen. Waiting-time behaviour in a nonlinear diffusion equation. Stud. Appl. Math., 67: 79–106, 1982.

[11] T. E. Lee. Modelling time dependent partial differential equations. PhD thesis, Department of Mathematics and Statistics, University of Reading, UK, 2011.

[12] T. E. Lee, M. J. Baines and S. Langdon. A finite difference moving mesh method based on conservation for moving boundary problems. J. Comput. Appl. Math., 288: 1–17, 2015.

[13] J. Vazquez. The Porous Medium Equation. Oxford University Press, 2007.