A positivity- and monotonicity-preserving moving-mesh finite difference scheme based on local conservation

July 31, 2017

M. J. Baines

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

e-mail: m.j.baines@reading.ac.uk

Abstract

A moving-mesh finite difference scheme based on local conservation is constructed that preserves the positivity of the solution and the monotonicity of the mesh. Numerical illustrations are given for several nonlinear diffusion problems.

Keywords : Adaptivity, moving-meshes, nonlinear diffusion, finite differences, conservation, positivity, non-tangling.

1 Introduction

In a moving mesh approach the unknowns are the domain and the solution. The velocity-based local conservation method proposed in [1] uses local conservation. Two key issues need to be addressed, the integrity of the mesh (avoiding tangling) and the positivity of the solution on the mesh (essential for local conservation). A subsidiary issue is smoothness of the solution, avoiding spurious oscillations that might spark off instability.

The local conservation method can be summarised as follows: at each time

1. obtain the Eulerian conservation velocity at each point of the domain,
2. integrate this velocity in time to deform the domain,
3. determine the solution on the new domain from Lagrangian conservation.

We call this method VMS (velocity, then mesh, then solution).

Previous work using this approach can be found in [8, 12, 1, 19, 2, 3, 15, 4, 14, 5, 6, 13, 10, 11, 17, 18]. However, using the above sequence numerically it is difficult to control mesh tangling and retain positivity and smoothness in the solution without requiring small time steps. Here we interchange the second and third steps, solving a PDE for the solution on the moving mesh prior to using Lagrangian conservation to construct the mesh.

The modified approach, called here VSM (velocity, then solution, then mesh), can be stated as follows. At each time,

1. obtain the Eulerian conservation velocity at each point of a domain (as for VMS),
2. integrate the rate of change of the solution following the motion to generate the solution on the new domain (yet to be determined),
3. deduce the domain of the solution from Lagrangian conservation.

The moving PDE of step 2 can be solved numerically by a semi-implicit scheme that satisfies a maximum/minimum principle (*cf.* [6]), admitting no new extrema in the solution and preserving positivity of the solution between extrema in a time step, thereby avoiding oscillations. The mesh is then constructed a posteriori from the Lagrangian integral, preserving the node ordering as a result of the positivity of the solution.

In this paper the VSM moving mesh method is first described for problems that conserve total mass, in section 2. Then, in section 3 the method is generalised to non mass-conserving problems with prescribed boundary fluxes. For the latter the (variable) total mass is inconsistent with local mass conservation but, as in [1, 4, 6, 13, 11], the local mass conservation can be

replaced by a normalised local conservation principle (normalised by the total mass) at the expense of carrying the additional normalising variable. The generalisation parallels the mass-conserving case, using a modified velocity.

Numerical tests are carried out in section 4 which confirm the predictions of the theory.

2 PDEs and local conservation

Consider the generic first-order-in-time-scalar PDE

$$u_t = \mathrm{L}\,u \tag{1}$$

for the function $u(x;t)$, where $\mathrm{L}\,u$ contains spatial derivatives of u, and let the total mass

$$= \int_{a(t)}^{b(t)} u(\,\cdot\,;t)\,\mathrm{d}\,\cdot$$

be constant in time.

A local form of conservation in a fixed frame is

$$u_t + (uv)_x = 0 \tag{2}$$

where v is the Eulerian velocity, whilst an equivalent conservation law (as long as u is positive) is the Lagrangian form

$$\int u(\,\cdot\,;t)\,\mathrm{d}\,\cdot = c; \tag{3}$$

where c is independent of time, for arbitrary limits on the integral. (The equivalence can be shown using Leibnitz' integral rule together with the total mass conservation.)

Assuming an anchor point at which the flux uv vanishes (which we take

The rate of change of u following the motion is

$$\frac{\mathrm{d}u}{\mathrm{d}t} = u_t + vu_x = -(uv)_x + vu_x = -uv_x \tag{5}$$

using (2).

Introducing a moving coordinate $\xi(x;t)$, the Lagrangian conservation law (3) can be written (in the fixed frame) as

$$\int u(\xi(\cdot\,;t);t)\frac{\partial \xi(\cdot\,;t)}{\partial}\mathrm{d}\,;$$

independent of time for arbitrary (fixed) limits of integration. Hence

$$u(x;t)\frac{\partial \xi}{\partial x} = b(x); \tag{6}$$

say, is independent of t, where $u(x;t) = u(\xi(x;t);t)$.

Let the moving coordinate $\xi(x;t)$ be determined at any fixed time t^0 by the differential equation

$$\frac{\partial \xi}{\partial t} = v(\xi;t); \qquad \xi(x;t^0) = x$$

where v is given by (4). Then, putting $\partial\xi/\partial t = \underline{x}$, from (4)

$$\underline{x} = v(\xi;t) = \frac{1}{u(\xi;t)}\int_0^{\xi(t)} u(\,)\,\mathrm{d}\,\Big) = \frac{1}{u}\int_0^{\xi(t)} u(\,)\,\mathrm{d}\,\Big) \tag{7}$$

since $u(\xi;t)$ has the same values as $u(x;t)$.

The rate of change of u following the motion, from (5) and (7), is

$$\underline{u} = -u(\xi;t)v(\xi;t)_x = -u(\xi;t)\frac{\partial}{\partial x}\left(\frac{1}{u(\xi;t)}\int_0^{\xi(t)} u(\,)\,\mathrm{d}\,\Big)\right)$$

$$= -u(\xi;t)\frac{\partial}{\partial x}\left(\frac{1}{u(\xi;t)}\int_0^{\xi(t)} u(\,)\,\mathrm{d}\,\Big)\right) \tag{8}$$

Examples are

the mass conservation law $u_t + (uq(u))_x = 0$, where $\mathcal{L}u = -(uq(u))_x$, for which equations (6), (7) and (8) can be written

$$\frac{\partial \psi}{\partial x} = \phi(x); \qquad \underline{x} = q(u); \qquad \underline{u} = -uq(u)_x$$

the nonlinear diffusion equation (1) where $\mathcal{L}u = (up_x)_x$, for which equations (6), (7) and (8) can be written

$$\frac{\partial \psi}{\partial x} = \phi(x); \qquad \underline{x} = p_x; \qquad \underline{u} = up_x$$

or, if p is a function of u only,

$$\frac{\partial \psi}{\partial x} = \phi(x); \qquad \underline{x} = p'(u)u_x; \qquad \underline{u} = u\{p'(u)u_x\}_x \tag{9}$$

the more general nonlinear diffusion equation (1) with $\mathcal{L}u = (D(u)u_x)_x$, for which equations (6), (7) and (8) become

$$\frac{\partial \psi}{\partial x} = \phi(x); \qquad \underline{x} = \frac{D(u)}{u}u_x; \qquad \underline{u} = u\left(\frac{D(u)}{u}u_x\right)_x \tag{10}$$

In the VSM method equation (8) is solved together with (6) for the two unknown parametric functions $\psi(x;t)$ and $\phi(x;t)$.

In the next section we discuss numerical schemes for the PDE (5).

2.1 Numerical schemes for ϕ

The domain is discretised using nodes x_i $(i = 1,\ldots,N)$ (not necessarily equally spaced) with function values u_i. Initially, the u_i are sampled from the initial condition at the nodes.

The nodal velocities v_i are obtained from a discretisation

$$v_i = \frac{1}{u_i}\int_0^{\phi_i(t)} \mathcal{L}u(\xi)\,d\xi$$

of (7), where the integral is evaluated using quadrature. A first-order-in-time explicit scheme for the PDE (5) is then

$$\phi_i^n = \phi_i \exp\{-\Delta t(v_x)_i\} \tag{11}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and the superscript n indicates the next time level. If the spatial approximation $(v_x)_i$ is positive $\forall i$ the amplification factor in (11) lies between 0 and 1 so that u_i remains positive and decreases with n. Moreover, if the spatial approximation $(v_x)_i$ increases with i then u_i decreases without oscillations. In general (11) may not avoid oscillations, however.

A semi-implicit scheme that does control oscillations is as follows.

2.1.1 A semi-implicit scheme

On the moving mesh a first-order-in-time explicit scheme for the PDE (5) is

$$\frac{u_i^n - u_i}{\Delta t} = -\frac{u_i}{\Delta x_i}\left(v_{i+1/2} - v_{i-1/2}\right) \tag{12}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and v_i is given by (7). The values at the half points are simple averages.

By comparison with (12) a consistent first-order-in-time semi-implicit scheme for the PDE (5) at interior nodes is (dropping the hats)

$$u_i^n$$

Similarly, if the ratios $v_{i\ 1=2}=\ u_{i\ 1=2}$ (or products $v_{i\ 1=2}\ \ u_{i\ 1=2}$) are both positive, the same extremum principle holds at each time step when the terms $(u_{i+1}\ \ u_i)^n$ and $(u_i\ \ u_{i\ 1})^n$ in (14) are interchanged (still maintaining consistency with the PDE (5)). The scheme then becomes

$$u_i^n\ \ u_i = \ _R(u_{i+1}^n\ \ u_i^n)\ \ \ _L(u_i^n\ \ u_{i\ 1}^n$$

) u

boundaries, between which the schemes (14) or (16) hold. A separate scheme is required at the points such as the explicit scheme (11).

At a re ection point x_r, say, where the $v_{r\ 1=2}$ and $u_{r\ 1=2}$ change sign simultaneously (and hence the $_L;\ _R$ or the $_L;\ _R$ do not change sign), u_r^n can be taken from (14) or (16). The values of u_i^n calculated from the system (14) or (16) then remain bounded but relinquish monotonicity at x_r, risking oscillations. Speci cally, from (14) the calculated values u_r^n satisfy

$$u_r^n = \frac{u_r + \ _L u_{r\ 1}^n + \ _R u_{r+1}^n}{1 + \ _L + \ _R}$$

where the right hand side is a positive average of adjacent u values and hence lies in their support. When t is small (so that the 's are small) u_r^n is close to u_r and when t is large it is close to a positive average of $u_{r\ 1}^n$ and u_{r+1}^n, taking only values in between, so there is no oscillation. A similar argument applies to (16). Where the derivative of the pro le is very small, perturbations of the u_i might lead to oscillations that grow: these however can be controlled by a mild regularisation in which the coe cients $_L;\ _R$ or $_L;\ _R$ are increased by a small positive number, equivalent to an change in the di usion coe cient in (14) together with the addition of an -sized Laplacian viscosity term.

2.1.4 Recovering the mesh

Once the u_i^n have been found from (14) or (16), the mesh can be recovered from the interval lengths x_i^n derived from the Lagrangian conservation law (6) in

grre

the initial time, equation (19) carries that approximation forward to the new time.

The full algorithm is as follows.

Algorithm 1

Given x_i and u_i at the initial time, evaluate the mass constants c_i from (19).

Then at each time step, provided that the $_L;\ _R$ or the $_L;\ _R$ are of the same sign,

1. calculate the v_i from a discretisation of (7)
2. determine the solution u_i^n on the new mesh from (11) or from (14)/(16)
3. obtain the new interval lengths x_i^n from (19)
4. construct the new mesh x_i^n using the recurrence (20)

The schemes (14) and (16) are unconditionally stable and admit no new oscillations in u_i^n in a time step. Moreover, provided that the boundary conditions are non-negative, monotonicity of the x_i^n is assured. The overall

be the total mass (varying with time) and introduce a normalised solution $\bar{u}(x;t) = u(x;t)/\mu(t)$. A normalised mass conservation principle (*cf.* (3)) is then

$$\int \bar{u}(\,\cdot\,;t)\,\mathrm{d}\,\cdot = \frac{1}{\mu(t)}\int u(\,\cdot\,;t)\,\mathrm{d}\,\cdot = \bar{c}(x); \tag{22}$$

say, independent of t, which is consistent with the constant total relative mass whose value is unity from (21) and (22). The constants $\bar{c}(x)$ are determined from the initial conditions.

The fixed-domain conservation law for $\bar{u}(x;t)$ is

$$(\bar{u})_t + (\bar{u}\bar{v})_x = 0 \tag{23}$$

where $\bar{v}$ is the Eulerian velocity (*cf.* (2)). As in section 2, assuming an anchor point at which the flux $\bar{u}\bar{v}$ vanishes (which we may take as the origin of the spatial coordinate x), it follows from (23) that the induced velocity $\bar{v}$ is

$$\bar{v}(x;t) = -\frac{1}{\bar{u}}\int_0^x (\bar{u})_t\,\mathrm{d}\cdot \tag{24}$$

(*cf.* (4)). Since for the PDE $u_t = \mathcal{L}u$,

$$(\bar{u})_t = \left(\frac{u}{\mu}\right)_t = \frac{1}{\mu}\,u_t - \frac{\dot{\mu}}{\mu^2}\,u = \frac{1}{\mu}\,\mathcal{L}u - \frac{\dot{\mu}}{\mu}\,\bar{u};$$

the velocity (24) can be written

$$\bar{v}(x;t) = -\frac{1}{\bar{u}}\int_0^x \left(\frac{1}{\mu}\mathcal{L}u(\,\cdot\,) - \frac{\dot{\mu}}{\mu}\,\bar{u}\right)\mathrm{d}\cdot = -\frac{1}{\bar{u}}\int_0^x \frac{1}{\mu}\,\mathcal{L}u(\,\cdot\,)\,\mathrm{d}\cdot + \frac{\bar{c}(x)}{\bar{u}}\,\frac{\dot{\mu}}{\mu} \tag{25}$$

using (22).

The rate of change $\dot{\mu}$ of the total mass is given from u and the prescribed boundary fluxes by Leibnitz' integral rule in the form

$$\dot{\mu} = \frac{\mathrm{d}}{\mathrm{d}t}\int_0^{b(t)} u(\,\cdot\,;t)\,\mathrm{d}\cdot = \int_0^{b(t)} u_t\,\mathrm{d}\cdot + [uv]_0^{b(t)} = \int_0^{b(t)} \mathcal{L}u(\,\cdot\,)\,\mathrm{d}\cdot + [\,\cdot\,]_0^{b(t)} \tag{26}$$

The rate of change of $\bar{u}$ following the motion is

$$\frac{\mathrm{d}\bar{u}}{\mathrm{d}t} = \bar{u}_t + \bar{v}\bar{u}_x = -(\bar{u}\bar{v})_x + \bar{v}\bar{u}_x = -\bar{u}\bar{v}_x \tag{27}$$

using (23).

Introducing a moving coordinate x'

where, from (26),

$$\_ = [\quad f(u) + \quad]_0^b + \int_0^b s\,\mathrm{d} \tag{32}$$

Mass is not conserved but we are assuming that the boundary fluxes are known, so $\_$ depends only on u and s.

We now consider numerical schemes for (30).

3.1 Numerical schemes for $\bar{u}$

Approximate nodal velocities $\bar{v}_i$ are first obtained from a discretisation

$$\bar{v}_i = \quad \frac{1}{u_i} \int_0^{x_i} \mathrm{L}\,u(\ )\,\mathrm{d} \quad + \frac{\bar{c}_i}{u_i}\_ \tag{33}$$

of (25), where $\bar{c}_i$ are the mass constants determined from initial data using (22), and where $\_$ is given from (32) by

$$\_ = \int_0^{x_N} \mathrm{L}\,u(\ )\mathrm{d} \quad + [\ ]_0^{x_N}; \tag{34}$$

the integrals being evaluated by quadrature.

Numerical schemes for (27) and (28) are essentially the same as in sections 2.1 and 2.1.4, with u replaced by $\bar{u}$ and v replaced by $\bar{v}$. Thus a first-order-in-time explicit scheme for the PDE (27) that maintains the sign of $\bar{u}$ is (dropping the hats)

$$\bar{u}_i^n = \bar{u}_i \exp\{\quad t\,(\bar{v}_x)_i\} \tag{35}$$

while a similar first-order-in-time explicit scheme for the ODE (26) is

$$^n = \quad \exp \quad t\,\_ \tag{36}$$

If the spatial approximation $(\bar{v}_x)_i$ is positive $\forall i$ the amplification factor in (35) lies between 0 and 1, so $\bar{u}_i$ remains positive and decreases with n. Moreover, if the spatial approximation to $(\bar{v}_x)_i$ increases with i then $\bar{u}_i$ remains monotonic, although (35) will not avoid oscillations in general.

A semi-implicit scheme for $\bar{u}_i$ that does control oscillations is

$$\frac{\bar{u}_i^n \quad \bar{u}_i}{} = \quad \frac{\bar{u}_i}{x_i}\Big(\bar{v}_{i+1=2}\frac{(\bar{u}_{i+1} \quad \bar{u}_i)^n}{(\bar{u}_{i+1} \quad \bar{u}_i)} \quad \bar{v}_{i\ 1=2}\frac{(\bar{u}_i \quad \bar{u}_{i\ 1})^n}{(\bar{u}_i \quad \bar{u}_{i\ 1})}\Big)$$

((u_{i-1} − u_i ≤ 0)), which can be written

$$u_i^n - u_i = \bar{}_R(u_{i+1}^n - u_i^n) - \bar{}_L(u_i^n - u_{i-1}^n) \tag{37}$$

where the coe cients are

$$\bar{}_R = \frac{u_i \, t}{x_i} \frac{v}{u}_{i+1=2}; \qquad \bar{}_L = \frac{u_i \, t}{x_i} \frac{v}{u}_{i-1=2} \tag{38}$$

Once u_i^n and ^{n} have been determined, the approximate solution in the moving frame $u_i^n = {}^n u_i^n$ can be obtained and the interval lengths *98 w 0 0 m 6.088 0 l*

The overall scheme is stable and consistent. If the initial and boundary conditions are such that the u_i^n determined from (35) or (37) remain positive, monotonicity of the x_i is assured.

The scheme (37) is unconditionally stable and admits no new oscillations in u_i^n in a time step.

4 Numerical tests

We illustrate the properties of Algorithms 1 and 2 applied to a standard nonlinear di usion problem (the porous medium equation.

4.1 A mass conserving problem

The nonlinear porous medium equation (PME) with a quadratic di usion coe cient,

$$u_t = (u^2 u_x)_x \qquad (a(t) < x < b(t) \tag{40}$$

where $u = 0$ on the boundaries is mass-conserving and has the exact self-similar solution [9, 16]

$$u(x;t) = \frac{1}{2t^{1=4}} \left(1 \quad \frac{x}{t^{1=4}} \right)_+^{2)\ 1=2} \tag{41}$$

in the expanding region $t^{1=4} < x < t^{1=4}$, where the su x + indicates the positive part of the argument. Note that the derivative of the solution is unbounded at $x = \ t^{1=4}$, so the problem is numerically challenging. We use this problem to demonstrate the stability and accuracy properties of Algorithm 1.

At the initial time, $t = 1$ say, the initial condition is taken from the function (41) as

$$u(x;1) = \frac{1}{2}(1 \quad x^2)^{1=2}$$

Due to the re ective symmetry $x = 0$ is the obvious anchor point. The mass constants are therefore

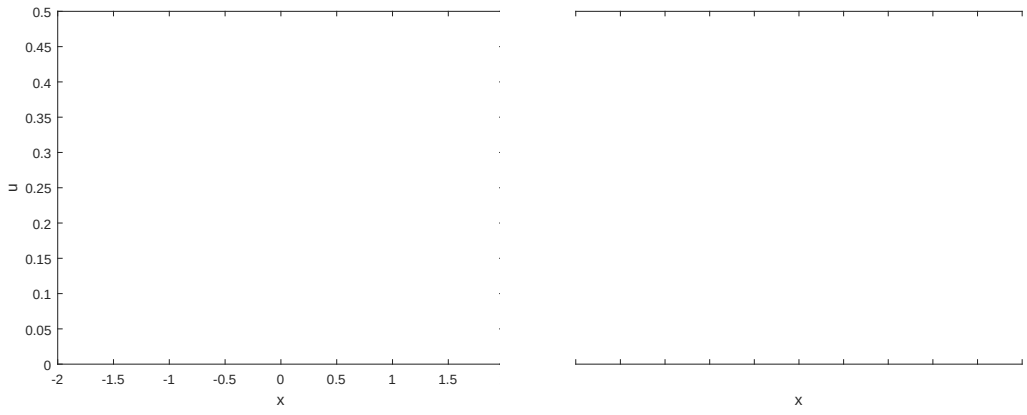
$$c_i = \frac{1}{2} \int_0^{x_i} (1 \qquad ^2)^{1=2} \mathrm{d}$$

The problem is of the form $u_t = (p(u)u_x)_x$ with $p(u) = \frac{1}{2}u^2$, so from (9) the velocity is $v = \frac{1}{2}(u^2)_x$. The velocity is approximated as

$$v_i = \frac{1}{2}\frac{(u_{i+1}^2 - u_{i-1}^2)}{(x_{i+1} - x_{i-1})}$$

The explicit scheme for u_i for this problem is (11) and the semi-implicit scheme is (18). Although the u_i are not monotonic at the origin, the $_L$; $_R$ of (38) are always of the same sign, so the internal boundary condition can be evaluated from the semi-implicit scheme (losing monotonicity at the maximum).

Once the approximate u_i



(a) (b)

Figure 1: Solutions using (a) the VMS and (b) the VSM methods for the problem (40) at $t = 16$ taking time steps $t = 0{:}5$ with initial conditions sampled from (41).

4.2.1 An accumulating non mass-conserving problem

The non mass-conserving problem

$$u_t = (uu_x)_x + \frac{2}{9} \qquad (t < x < t) \tag{42}$$

with $u = 0$ and zero fluxes on the boundary of the expanding interval has the self-similar solution (see Appendix)

$$u(x; t) = \frac{1}{6t}\left(1 - \frac{x^2}{t^2}\right) \tag{43}$$

We use this problem to demonstrate the stability and accuracy properties of Algorithm 2. Initial conditions are taken from the self-similar solution (43).

At the initial time, $t = 1$ say, the initial condition is

$$u(x; 1) = \frac{1}{6}\left(1 - x^2\right); \tag{44}$$

taken from (43). The initial total mass, by (21), is therefore

$$(1) = \frac{1}{6}\int_{-1}^{1}(1 - x^2)\,dx = \frac{2}{9}$$

and hence $\bar{u}(x;1) = u(x;1) = (1) = (3=4)(1 \quad x^2)$.

Due to the symmetry about the origin, $x = 0$ is taken as the anchor point. The normalised mass constants, from (22), are thus

$$\bar{c}_i = \frac{3}{4}\int_0^{x_i(1)} 1 \quad {}^2 \, d \; = \frac{3}{4}x_i(1) \quad \frac{1}{2}x_i(1)^3$$

The x_i are initially equally spaced (although this is not essential).

The rate of change of the total mass, from (34) and the boundary conditions, is

$$\dot{-} = \int_0^{x_N} \frac{2}{9} d \; = \frac{2}{9}x_N \tag{45}$$

and hence from (33) the velocity (relative to the anchor point) is

$$\bar{v}_i = \frac{u_i(u_x)_i \quad \int_0^{x_i} (2=9) d \; + \bar{c}_i \dot{-}}{u_i} = \quad (u_x)_i \quad \frac{2}{9}\frac{x_i}{u_i} + \frac{2}{9}\frac{\bar{c}_i x_N}{u_i} \tag{46}$$

The explicit scheme for $\bar{u}_i$ is (35) with $\bar{v}_i$ given by (46).

The semi-implicit scheme for $\bar{u}_i$ is (37) with boundary conditions $\bar{u} = 0$ at the free boundaries. Although the values of $\bar{u}_i$ are not monotonic at the origin the $\bar{}_L; \bar{}_R$ of (38) are always of the same sign, so the internal boundary condition can be obtained from the semi-implicit scheme (losing monotonicity at the maximum).

The total mass is advanced in time by the explicit scheme (36) with $\dot{-}$ given by (45). The mesh is then calculated from (20) with one-sided di erences.

Algorithm 2 of section 3 is run for 40 interior points. Four increasing t's are used, $t = 0{:}1; 0{:}5; 1; 5$, in reaching the xed time $t = 51$, progressively forfeiting accuracy as t increases, as monitored by the relative error in the l^2 norm of the solution and the l^1 norm of the free boundary. (see Table 2).

The totalr 11.9552 Tf 281.653 420.877 Td [(u)]TJ/F29 7t9701 Tf 6.088 -1.793 TF28 i

t	Relative error in u	Relative error in x_N
0:1	0:01675	0:0000053

$\bar{u}(x;1)$, and mass constants $\bar{C}_i$, are the same as in section 4.2.1.

The x_i are initially equally spaced (although this is not a requirement).

From (34) and the boundary conditions, the rate of change of the total mass is

$$\dot{M} = \int_0^{x_N} \frac{4}{9} t^{-4} \, dx = \frac{4}{9} t^{-4} x_N \qquad (49)$$

and hence the velocity, from (31) and (32), is

$$\bar{v}_i = \frac{u_i(u_x)_i + \int_0^{x_i} (4/9) t^{-4} \, dx}{\bar{C}_i \dot{M}}$$

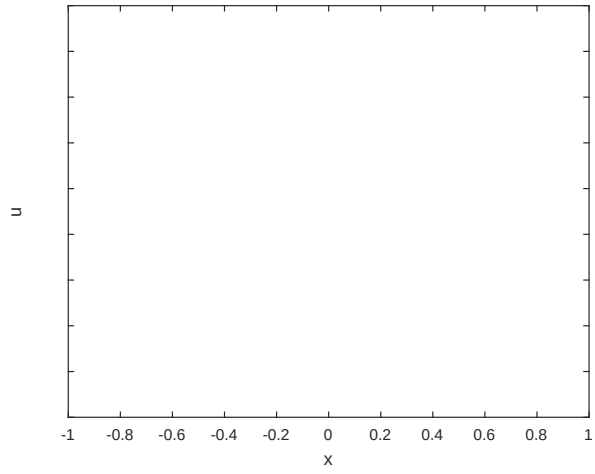


Figure 3: VSM solutions (circles) and exact solutions (crosses) of the PME problem (47) with a negative source term, with initial condition sampled from (48), at time intervals $t = 0.1$ (top to bottom) from $t = 1$ to $t = 1.5$.

5 Conclusions

In this paper we have studied a velocity-based moving mesh scheme based on local conservation for scalar one-dimensional time-dependent PDEs with moving boundaries. We showed first that for mass-conserving problems there exists a semi-implicit moving mesh scheme (VSM) based on conservation that preserves positivity and monotonicity of the solution and avoids mesh tangling for arbitrarily large tme steps. The method was then generalised to problems that do not conserve total mass but for which boundary fluxes are prescribed.

Analytically, a velocity was derived from local conservation and used to obtain a PDE for the solution on the moving domain following the motion. The deformation of the domain was then determined from this solution using the Lagrangian form of local conservation.

Numerically, given the mesh and solution at an initial time, the velocity was approximated and the PDE following the motion solved by a semi-implicit scheme possessing an extremum principle. The mesh was then obtained (algebraically) from a simple quadrature of the Lagrangian conservation principle.

The sequence (VSM) of calculating the (V)-elocity, solving for the (S)-olution, and then recovering the (M)-esh di ers from the conservation method published in the literature (see [13] and references therein), giving improved stability through the positivity and monotonicity-preservation properties for arbitrarily large time steps.

In section 2, devoted to problems conserving total mass, a local conservation principle was used to obtain a generalised Eulerian velocity (4) which was then used to derive the time-dependent PDE (5) for the solution on the moving domain. The semi-implicit schemes (14) and (16) were then constructed which preserved positivity and monotonicity of the solution and, when substituted into an (approximate) form of conservation law (19), preserved monotonicity of the nodes for any time step (see Algorithm 1).

In section 3 the method was generalised to non mass-conserving problems using a normalised local conservation principle. A generalised Eulerian velocity (31) was used to derive the time-dependent PDE (30) following the motion for the normalised solution on the moving domain, while the total mass was computed through its time rate of change (26). Semi-implicit schemes (38) and (37) were derived for the normalised solution which preserved positivity and monotonicity of the normalised solution for any time step. (The normalised solution di uses as in the mass-conserving case but the solution itself is capable of additional variation.) When used with the approximate normalised Lagrangian conservation law, monotonicity of the nodes was preserved (see Algorithm 2.)

Numerical tests on the two algorithms were carried out in section 4 on simple nonlinear di usion problems with prescribed uxes having analytic solutions, rst for a non-trivial mass-conserving nonlinear di usion problem (a porous medium equation with a quadratic di usion coe cient), and then for two non mass-conserving nonlinear di usion problem with growing or shrinking solutions. The solution was always positivity preserving and the mesh remained untangled.

The velocity-based moving mesh VSM algorithms in this paper represent an advance on the VMS methods used previously [8, 14, 5, 6, 13, 10, 7, 11, 17] in that they preserve positivity of the solution and monotonicity of the mesh for arbitrary time steps. However, they are only rst-order accurate in tine, so care is required in their use.

The extension to multidimensions is planned. The calculation of the velocity v in multidimensions has been considered elsewhere (see e.g. [1, 4])

motion (5) generalises to

$$u_t = \quad u\mathrm{r} \quad \mathrm{v}$$

In a nite di erence approach the r v term may be approximated at any point in a mesh of triangles by a linear sum of values of a velocity potential at adjacent nodes. By introducing quotients of di erences in u into this equation a second order parabolic PDE for u can be created which admits a semi-implicit system with a solution that possesses a positive averaging property. Positive triangle areas can be then be determined from the discrete Lagrangian form of the conservation principle. Although these areas do not de ne the mesh uniquely (as they do in the one-dimensional case) an approximate mesh can be constructed that avoids mesh tangling by maintaining the signs of the triangle areas.

Appendix: Similarity solutions for the PME with source terms

Let the function $u(x;t)$ have the form (ansatz)

$$u(x;t) = \frac{1}{6}t\ (1 \quad {}^2); \qquad = \frac{x}{t} \tag{51}$$

Di erentiation gives

$$u_t = \frac{1}{6}\ t \quad {}^1(1 \quad {}^2) + \frac{1}{6}t\ (\ 2 \quad {}_t) = \quad 1$$

and hence (51) is a self-similar solution of the inhomogeneous scale-invariant partial di erential equation

$$u_t \quad (uu_x)_x = \frac{1}{6}t^{\ 1} \quad + \frac{1}{3}$$

for general , where $= \frac{1}{2}(1 + \ )$.

If $= 1$ then $= 1$, so $= x/t$ and

$$u(x;t) = \frac{1}{6} \ t \quad \frac{x^2}{t}^{!}$$

is a self-similar solution of the partial di erential equation

$$u_t = (uu_x)_x + \frac{2}{9}$$

in the expanding interval ($t < x < t$).

If $= 3$ then $= 1$, so $= x/t^{\ 1}$ and

$$u(x;t) = \frac{1}{6} \ t^{\ 3} \quad \frac{x^2}{t}^{!}$$

is a self-similar solution of the partial di erential equation

$$u_t = (uu_x)_x \quad \frac{4/9}{t^4}$$

in the contracting interval ($t^{\ 1} < x < t^{\ 1}$).

References

[1] M.J. Baines, M.E. Hubbard, and P.K. Jimack , A
, Appl. Numer. Math., 54, pp. 450-469, 2005.

[2] M.J. Baines, M.E. Hubbard, P.K. Jimack and A.C. Jones , , Appl. Numer. Math., 56, pp. 230-252, 2006.

[3] M.J. Baines, M.E. Hubbard, P.K. Jimack and R. Mahmood , A
, Commun. Comput. Phys., 6, pp. 595-624, 2009.

[4] M.J. Baines, M.E. Hubbard, and P.K. Jimack , , Com-
mun. Comput. Phys., 10, pp. 509-576, 2011.

[5] M.J. Baines, T.E. Lee, S. Langdon and M.J. Tindall , An
, Appl. Numer. Math., 72, pp. 99-114 (2013).

[6] M. J. Baines and T. E. Lee , Adaptive
, Num. Soln. PDEs, 30, pp.321-338 (2014).

[7] M.J. Baines , Error
, J. Math Study, 48, pp. 93-105 (2015).

[8] K.W. Blake , Variable
, PhD thesis, University of Reading, UK, 2001.

[9] G.I. Barenblatt , On some
, Prik. Mat. Mekh, 16 (1952), pp. 67-68.

[10] N. Bird , hp-DG
, PhD thesis, Department of Mathematics and Statistics, University of Reading, UK, 2015.

[11] B. Bonan, M.J. Baines, N.K. Nichols and D. Partridge , A
, The Cryosphere, 10, pp. 1-14, doi: 10.5194/tc-10-1-2016 (2016).

[12] W. Cao, W. Huang, and R.D. Russell , An
, SIAM J. Sci. Comput., 24 (2002), pp. 118-142.

[13] T.E. Lee, M.J. Baines, and S. Langdon , A
,
J. Comp. Appl. Math, 288, pp.1-17 (2015).

[14] A.V. Lukyanov, M.M. Sushchikh, M.J. Baines, and T.G. Theofanous , , Phys. Rev. Letters, 109, 214501 (2012).