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

$$u_t = \mathcal{L}u \tag{1}$$

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

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

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

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

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 to be the origin of the spatial coordinate x), it follows from (1) and (2) that if $u > 0$ the velocity at the point x is

$$v(x,t) = -\frac{1}{u}\int_0^x \mathcal{L}u(\chi)\,\mathrm{d}\chi \tag{4}$$

In particular, when $\mathcal{L}u \equiv f(u)_x$, the velocity reduces to $v(x,t) = -f(u)/u$ (which may also be deduced from zero net fluxes at the moving nodes).

The rate of change of u following the motion is

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

using (2).

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

$$\int u(\widehat{x}(\chi,t),t)\frac{\partial \widehat{x}(\chi,t)}{\partial \chi}\mathrm{d}\chi,$$

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

$$\widehat{u}(x,t)\frac{\partial \widehat{x}}{\partial x} = \widehat{c}(x), \tag{6}$$

say, is independent of t, where $\widehat{u}(x,t) = u(\widehat{x}(x,t),t)$.

Let the moving coordinate $\widehat{x}(x,t)$ be determined at any fixed time t^0 by the differential equation

$$\frac{\partial \widehat{x}}{\partial t} = v(\widehat{x},t), \qquad \widehat{x}(x,t^0) = x$$

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

$$\dot{x} = v(\widehat{x},t) = -\frac{1}{u(\widehat{x},t)}\int_0^{\widehat{x}(t)} \mathcal{L}u(\chi)\,\mathrm{d}\chi = -\frac{1}{u}\int_0^{\widehat{x}(t)} \mathcal{L}u(\chi)\,\mathrm{d}\chi \tag{7}$$

since $u(\widehat{x},t)$ has the same values as $u(x,t)$.

The rate of change of $\widehat{u}$ following the motion, from (5) and (7), is

$$\dot{u} = -u(\widehat{x},t)v(\widehat{x},t)_x = u(\widehat{x},t)\frac{\partial}{\partial x}\left\{\frac{1}{u(\widehat{x},t)}\int_0^{\widehat{x}(t)} \mathcal{L}u(\chi)\,\mathrm{d}\chi\right\}$$

$$= u(\widehat{x},t)\frac{\partial}{\partial x}\left\{\frac{1}{\widehat{u}(\widehat{x},t)}\int_0^{\widehat{x}(t)} \mathcal{L}u(\chi)\,\mathrm{d}\chi\right\} \tag{8}$$

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

$$\widehat{u}\,\frac{\partial \widehat{x}}{\partial x} = \widehat{c}(x), \qquad \dot{x} = q(u), \qquad \dot{u} = -u\,q(u)_x$$

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

$$\widehat{u}\,\frac{\partial \widehat{x}}{\partial x} = \widehat{c}(x), \qquad \dot{x} = -p_x, \qquad \dot{u} = u\,p_{xx}$$

  or, if p is a function of u only,

$$\widehat{u}\,\frac{\partial \widehat{x}}{\partial x} = \widehat{c}(x), \qquad \dot{x} = -p'(u)u_x, \qquad \dot{u} = u\{p'(u)u_x\}_x \tag{9}$$

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

$$\widehat{u}\,\frac{\partial \widehat{x}}{\partial x} = \widehat{c}(x), \qquad \dot{x} = -\frac{D(u)}{u}u_x, \qquad \dot{u} = u\left(\frac{D(u)}{u}u_x\right)_x \tag{10}$$

In the VSM method equation (8) is solved together with (6) for the two unknown parametric functions $\widehat{x}(x,t)$ and $\widehat{u}(x,t)$.

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

2.1 Numerical schemes for $\widehat{u}$

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

$$v_i = -\frac{1}{u_i}\int_0^{\widehat{x}_i(t)} \mathcal{L}u(\chi)\,\mathrm{d}\chi$$

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

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

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and the superfix 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 $\widehat{u}_i$ remains positive and decreases with n. Moreover, if the spatial approximation $(v_x)_i$ increases with i then $\widehat{u}_i$ 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

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

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)

$$\frac{u_i^n - u_i}{\Delta t} = -\frac{u_i}{\Delta x_i}\left\{v_{i+1/2}\,\frac{(u_{i+1} - u_i)^n}{(u_{i+1} - u_i)} - v_{i-1/2}\frac{(u_i - u_{i-1})^n}{(u_i - u_{i-1})}\right\} \tag{13}$$

($(u_{i\pm1} - u_i) \neq 0$), noting that the quotients introduced into (12) are $O(\Delta t)$ approximations. If either of $(u_{i\pm1} - u_i) = 0$ no such modification is made. (Values of u_i^n at the boundaries are obtained from boundary conditions or local explicit schemes.)

Putting $\Delta u_{i+1/2} = u_{i+1} - u_i$ and $\Delta u_{i-1/2} = u_i - u_{i-1}$, the scheme (13) can be written

$$u_i^n - u_i = \kappa_R(u_{i+1}^n - u_i^n) - \kappa_L(u_i^n - u_{i-1}^n) \tag{14}$$

where the coefficients are

$$\kappa_R = -\frac{u_i\Delta t}{\Delta x_i}\left(\frac{v}{\Delta u}\right)_{i+1/2}, \qquad \kappa_L = -\frac{u_i\Delta t}{\Delta x_i}\left(\frac{v}{\Delta u}\right)_{i-1/2} \tag{15}$$

If the ratios $(v/\Delta u)_{i\pm1/2}$ (or products $(v\Delta u)_{i\pm1/2}$) are both negative, so that the κ_L, κ_R are both positive, the u_i^n in (14) satisfy an extremum principle at each time step, admitting no new local extrema and lying between their values on the boundaries. Provided that the boundary values are non-negative the u_i^n remain non-negative.

Similarly, if the ratios $v_{i\pm1/2}/\Delta u_{i\mp1/2}$ (or products $v_{i\pm1/2}\Delta u_{i\mp1/2}$) 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

$$u_i^n - u_i = \lambda_R(u_{i+1}^n - u_i^n) - \lambda_L(u_i^n - u_{i-1}^n) \tag{16}$$

where

$$\lambda_R = -\kappa_L = \frac{u_i\Delta t}{\Delta x_i}\left(\frac{v_{i-1/2}}{\Delta u_{i+1/2}}\right) \qquad \lambda_L = -\kappa_R = \frac{u_i\Delta t}{\Delta x_i}\left(\frac{v_{i+1/2}}{\Delta u_{i-1/2}}\right) \tag{17}$$

2.1.2 Example

For diffusive PDEs of the form $u_t = (up(u)_x)_x$, the velocity from (9) is $v = -p(u)_x$, so from (11) a first-order-in-time explicit scheme for the PDE (5) is

$$\widehat{u}_i^n = \widehat{u}_i \, \exp\left\{\Delta t \, (p(u)_{xx})_i\right\}$$

The velocity can also be written $v = -p'(u)u_x$ which can be approximated as

$$v_{i-1/2} = -\left(p'(u)\right)_{i-1/2}\left(\frac{u_i - u_{i-1}}{x_i - x_{i-1}}\right),$$

so a first-order-in-time semi-implicit scheme of the third of (10) for u_i is

$$\frac{u_i^n - u_i}{\Delta t} = \frac{u_i}{\Delta x_i}\left\{\left(p'(u)\right)_{i+1/2}\frac{(u_{i+1}^n - u_i^n)}{(x_{i+1} - x_i)} - \left(p'(u)\right)_{i-1/2}\frac{(u_i^n - u_{i-1}^n)}{(x_i - x_{i-1})}\right\} \tag{18}$$

which can also be derived immediately from (9). Provided that $p'(u)$ is positive, by the argument of the previous section the extremum principle ensures that no new local extrema in u_i^n are generated and that the u_i^n remain bounded between their maximum and minimum values on the boundaries.

2.1.3 Special points

At (isolated) points where the κ_L, κ_R of (15) or the λ_L, λ_R of (17) are of opposite sign (as for example in fourth order nonlinear diffusion [10]), the argument fails because the necessary strict convexity of the right hand side of (14) or (16) is lost. However, these points, x_I say, may be regarded as internal

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 reflection point x_r, say, where the $v_{r\pm1/2}$ and $\Delta u_{r\pm1/2}$ change sign simultaneously (and hence the κ_L, κ_R or the λ_L, λ_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. Specifically, from (14) the calculated values u_r^n satisfy

$$u_r^n = \frac{u_r + \kappa_L u_{r-1}^n + \kappa_R u_{r+1}^n}{1 + \kappa_L + \kappa_R}$$

where the right hand side is a positive average of adjacent u values and hence lies in their support. When Δt is small (so that the κ's are small) u_r^n is close to u_r and when Δt 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 profile 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 coefficients κ_L, κ_R or λ_L, λ_R are increased by a small positive number ϵ, equivalent to an ϵ change in the diffusion coefficient in (14) together with the addition of an ϵ-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 $\Delta^{\pm} x_i^n$ derived from the Lagrangian conservation law (6) in the approximate form

$$u_i^n \Delta^{\pm} x_i^n = \Delta^{\pm} c_i \tag{19}$$

where $\Delta^{\pm} x_i^n$ is a one-sided difference. The masses $\Delta^{\pm} c_i$ are independent of time and therefore known from initial conditions. Using an anchor point, x_A say, the x_i^n can be constructed from these one-sided intervals through the recursion

$$x_i^n = x_A + \sum_{j=A}^{i-1} \Delta^{\pm} x_j^n = x_A + \sum_{j=A}^{i-1} \frac{\Delta^{\pm} c_j}{u_j^n} \tag{20}$$

Positivity of the u_i^n ensures monotonicity of the x_i^n.

We note that equation (19) reproduces the same approximation at time t as at the initial time. In particular, if the u_i are sampled at the nodes at

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 κ_L, κ_R or the λ_L, λ_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 Δx_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 scheme is illustrated in the numerical results section 4 below.

The VSM approach in this paper improves on the VMS schemes described in [13] or [6] in that the algebraic equation (19) linking the mesh and the solution determines the Δx_i^n from the u_i^n, rather than the other way round, avoiding the possible oscillations in u_i^n generated by those in Δx_i^n.

3 Non mass-conserving problems

The argument in section 2 applies only to problems with constant total mass, allowing consistent local mass conservation. In this section we present a generalisation for problems with varying total mass.

For problems that do not conserve total mass but have prescribed fluxes ϕ_a, ϕ_b at the boundaries, we may use a normalised form of local conservation. Let

$$\theta(t) = \int_0^{b(t)} u(\chi, t)\, \mathrm{d}\chi \tag{21}$$

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

$$\int \overline{u}(\chi,t)\,\mathrm{d}\chi = \frac{1}{\theta(t)}\int u(\chi,t)\,\mathrm{d}\chi = \overline{c}(x), \tag{22}$$

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

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

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

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

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

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

$$(\overline{u})_t = \left(\frac{u}{\theta}\right)_t = \frac{1}{\theta}\left(u_t - \frac{\dot\theta}{\theta}u\right) = \frac{1}{\theta}\left(\mathcal{L}u - \frac{\dot\theta}{\theta}u\right) = \frac{1}{\theta}\mathcal{L}u - \frac{\dot\theta}{\theta}\overline{u},$$

the velocity (24) can be written

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

using (22).

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

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

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

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

using (23).

Introducing a moving coordinate $\widehat{x}(x,t)$, the normalised Lagrangian conservation law (22) can be written as

$$\int \overline{u}(\widehat{x}(\chi,t),t)\frac{\partial \widehat{x}(\chi,t)}{\partial \chi}\mathrm{d}\chi$$

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

$$\widehat{\overline{u}}(x,t)\frac{\partial \widehat{x}}{\partial x} = \widehat{\overline{c}}(x), \tag{28}$$

say, independent of time, where $\widehat{\overline{u}}(x,t) = \overline{u}(\widehat{x}(x,t),t)$. The values of $\widehat{\overline{c}}(x)$ are determined from initial data.

Suppose that the moving coordinate $\widehat{x}(x,t)$ is defined at any fixed time t^0 by

$$\frac{\partial \widehat{x}}{\partial t} = \overline{v}(\widehat{x},t), \qquad \widehat{x}(x,t^0) = x,$$

Then, putting $\partial\widehat{x}/\partial t = \dot{x}$, from (25)

$$\dot{x} = \frac{\partial \widehat{x}}{\partial t} = -\frac{1}{\widehat{u}}\int_0^{\widehat{x}(t)} \mathcal{L}u(\chi)\,\mathrm{d}\chi + \frac{\overline{c}(\widehat{x})}{\widehat{u}}\dot{\theta} \tag{29}$$

The rate of change of $\overline{u}$ following the motion, from (27) and (29), is

$$\dot{u} = \frac{\partial \widehat{u}}{\partial t} = -\overline{u}\dot{x} = \widehat{\overline{u}}\frac{\partial}{\partial x}\left\{\frac{1}{\widehat{u}}\int_0^{\widehat{x}(t)} \mathcal{L}u(\chi)\,\mathrm{d}\chi - \frac{\overline{c}(x)}{u}\dot{\theta}\right\} \tag{30}$$

In the VSM method equation (30) is solved, together with (28) and (26), for the unknown parametric functions $\widehat{\overline{u}}(x,t)$, $\widehat{x}(x,t)$, and $\theta(t)$, where $\overline{c}(x)$ is determined by (22) at the initial time. The un-normalised solution u is obtained from $u = \overline{u}/\theta(t)$, where $\theta(t)$ is found by integration of (26).

Example

In the case of the often-used scalar time-dependent PDE

$$u_t + f_x(u) = s$$

where $f(u)$ is a nonlinear flux function (depending only on u and its spatial derivatives) and s is a specified source term (for which $\mathcal{L}u = -f(u)_x + s$), the velocity (29) becomes

$$\overline{v}(x,t) = \frac{f(u) - \int_0^x s\,\mathrm{d}\chi + \overline{c}(x)\dot{\theta}}{u} \tag{31}$$

where, from (26),

$$\dot{\theta} = [-f(u) + \phi]_0^b + \int_0^b s \, \mathrm{d}\chi \tag{32}$$

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

We now consider numerical schemes for (30).

3.1 Numerical schemes for $\widehat{\overline{u}}$

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

$$\overline{v}_i = -\frac{1}{u_i} \int_0^{x_i} \mathcal{L}u(\chi) \, \mathrm{d}\chi + \frac{\overline{c}_i}{u_i} \dot{\theta} \tag{33}$$

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

$$\dot{\theta} = \int_0^{x_N} \mathcal{L}u(\chi) \mathrm{d}\chi + [\phi]_0^{x_N}, \tag{34}$$

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 $\overline{u}$ and v replaced by $\overline{v}$. Thus a first-order-in-time explicit scheme for the PDE (27) that maintains the sign of u is (dropping the hats)

$$\overline{u}_i^n = \overline{u}_i \, \exp\left\{-\Delta t \, (\overline{v}_x)_i\right\} \tag{35}$$

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

$$\theta^n = \theta \, \exp\left(\Delta t \, \dot{\theta}/\theta\right) \tag{36}$$

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

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

$$\frac{\overline{u}_i^n - \overline{u}_i}{\Delta\tau} = -\frac{\overline{u}_i}{\Delta x_i} \left\{ \overline{v}_{i+1/2} \frac{(\overline{u}_{i+1} - \overline{u}_i)^n}{(\overline{u}_{i+1} - \overline{u}_i)} - \overline{v}_{i-1/2} \frac{(\overline{u}_i - \overline{u}_{i-1})^n}{(\overline{u}_i - \overline{u}_{i-1})} \right\}$$

$((\overline{u}_{i\pm1} - \overline{u}_i \neq 0))$, which can be written

$$\overline{u}_i^n - \overline{u}_i = \overline{\kappa}_R(\overline{u}_{i+1}^n - \overline{u}_i^n) - \overline{\kappa}_L(\overline{u}_i^n - \overline{u}_{i-1}^n) \tag{37}$$

where the coefficients are

$$\overline{\kappa}_R = -\frac{\overline{u}_i \Delta t}{\Delta x_i}\left(\frac{\overline{v}}{\Delta \overline{u}}\right)_{i+1/2}, \qquad \overline{\kappa}_L = -\frac{\overline{u}_i \Delta t}{\Delta x_i}\left(\frac{\overline{v}}{\Delta \overline{u}}\right)_{i-1/2} \tag{38}$$

Once $\overline{u}_i^n$ and θ^n have been determined, the approximate solution in the moving frame $u_i^n = \theta^n \overline{u}_i^n$ can be obtained and the interval lengths $\Delta^{\pm} x_i^n$ at the forward time step can be found from the Lagrangian form of normalised conservation in the approximate form

$$\overline{u}_i^n \Delta^{\pm} x_i^n = \Delta^{\pm} \overline{c}_i \tag{39}$$

(see (19)) with the $\Delta^{\pm}\overline{c}_i$ obtained from initial data (omitting the hats). Using the anchor point, the interval lengths $\Delta^{\pm} x_i^n$ from (19) can be used to reconstruct the x_i^n through the recursion (20) where the $\Delta^{\pm} c_i \, (= \theta \Delta^{\pm} \overline{c}_i)$ are their initial values.

We note that positivity of the $\overline{u}_i$ ensures monotonicity of the x_i.

The algorithm is as follows.

Algorithm 2

Given initial values u_i and x_i evaluate the initial total mass θ from (21) and the normalised mass constants $\overline{C}_i$ from (22). Then, at each time step, provided that the $\overline{\kappa}_L, \overline{\kappa}_R$ or the $\overline{\lambda}_L, \overline{\lambda}_R$ are of the same sign,

1. calculate the $\overline{v}_i$ from (25) and $\dot{\theta}$ from (26)
2. determine the $\overline{u}_i^n$ from (37)
3. calculate θ from (36)
4. derive the u_i^n from the relation $\overline{u} = u/\theta$ and obtain the interval lengths Δx_i^n from (39) with mass constants $c_i^{\pm}$ calculated from initial data
5. construct the x_i^n using the recurrence (20).

The overall scheme is stable and consistent. If the initial and boundary conditions are such that the $\overline{u}_i^n$ 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 $\overline{u}_i^n$ in a time step.

4 Numerical tests

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

4.1 A mass conserving problem

The nonlinear porous medium equation (PME) with a quadratic diffusion coefficient,

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

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

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

in the expanding region $-t^{1/4} < x < t^{1/4}$, where the suffix + indicates the positive part of the argument. Note that the derivative of the solution is unbounded at $x = \pm t^{1/4}$, 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

$$u(x,1) = \frac{1}{2}(1 - x^2)^{1/2}$$

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

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

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 κ_L, κ_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 has been obtained, the mesh is obtained from (20) using one-sided differences.

Algorithm 1 of section 2.1 is run for 40 interior points. Five increasing Δt's are used, $\Delta t = 0.5$, 0.25, 5, 25, 50, in reaching the fixed time $t = 151$, progressively forfeiting accuracy as Δt increases, as monitored by the relative error in the l^2 norm of the solution and the l^∞ norm of the free boundary, (see Table 1).

The solution is always positive and the mesh monotonic (no tangling).

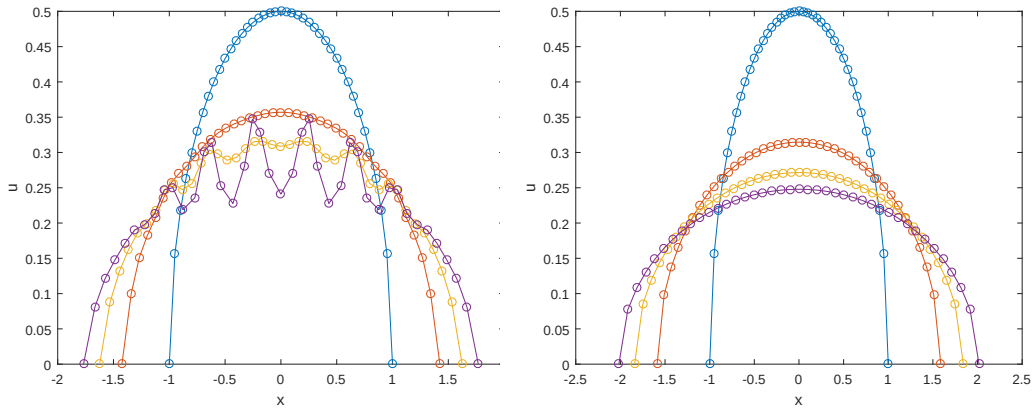
Δt	Relative error in u	Relative error in x_N
0.5	0.0016	0.0016
2.5	0.01057	01060
5	0.353	0.336
25	0.5218	1.0113
50	0.7405	2.8540

Table 1: Relative errors in u and x_N at $t = 151$ from the solution with 40 nodes for the PME (40).

Figure 1 compares the VSM solutions for the PDE (40) at $t = 16$ using a time step $\Delta t = 0.5$ with the corresponding VMS solutions (see section 1 or [13, 11]). It is apparent that the oscillations in the VMS profile are completely avoided in the VSM profile.

4.2 Non mass-conserving problems

In order to test the algorithm for non mass-conserving problems with prescribed fluxes we use the simplest PME with either a positive or negative source term.



(a) (b)

Figure 1: Solutions using (a) the VMS and (b) the VSM methods for the problem (40) at $t = 16$ taking time steps $\Delta t = 0.5$ 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

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

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

$$\theta(1) = \frac{1}{6}\int_{-1}^{1}(1-\chi^2)\,\mathrm{d}\chi = \frac{2}{9}$$

and hence $\overline{u}(x,1) = u(x,1)/\theta(1) = (3/4)\,(1-x^2)$.

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

$$\overline{c}_i = \frac{3}{4}\int_0^{x_i(1)} \left(1-\chi^2\right)\,\mathrm{d}\chi = \frac{3}{4}x_i(1) - \frac{1}{2}x_i(1)^3$$

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

$$\dot{\theta} = \int_0^{x_N} \frac{2}{9}\mathrm{d}\chi = \frac{2}{9}x_N \tag{45}$$

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

$$\overline{v}_i = \frac{-u_i(u_x)_i - \int_0^{x_i}(2/9)\mathrm{d}\chi + \overline{c}_i\,\dot{\theta}}{u_i} = -(u_x)_i - \frac{2}{9}\frac{x_i}{u_i} + \frac{2}{9}\frac{\overline{c}_i\,x_N}{u_i} \tag{46}$$

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

The semi-implicit scheme for $\overline{u}_i$ is (37) with boundary conditions $\overline{u} = 0$ at the free boundaries. Although the values of $\overline{u}_i$ are not monotonic at the origin the $\overline{\kappa}_L, \overline{\kappa}_R$ 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 θ is advanced in time by the explicit scheme (36) with $\dot{\theta}$ given by (45). The mesh is then calculated from (20) with one-sided differences.

Algorithm 2 of section 3 is run for 40 interior points. Four increasing Δt's are used, $\Delta t = 0.1,\ 0.5,\ 1,\ 5$, in reaching the fixed time $t = 51$, progressively forfeiting accuracy as Δt increases, as monitored by the relative error in the l^2 norm of the solution and the l^∞ norm of the free boundary. (see Table 2).

Figure 2 shows the approximate solutions (circles) from $t = 1$ at time intervals 10 up to $t = 50$ using $\Delta t = 1$, together with the exact solution (crosses).

The solution is always positive and the mesh monotonic.

4.2.2 A non mass-conserving problem with evaporation

The non mass-conserving problem

$$u_t = (uu_x)_x - \frac{4/9}{t^4} \qquad (-t^{-1} < x < t^{-1}) \tag{47}$$

Δt	Relative error in u	Relative error in x_N
0.1	0.01675	0.0000053
0.5	0.07785	0.01829
1	0.07915	0.02864
5	0.5004	37.198

Table 2: Relative errors in u^n and x_N^n at $t = 51$ for the VSM method applied to the PDE (42) with initial condition (44)), using 40 nodes and taking time steps $\Delta t = 0.1, 0.5, 1, 5$.

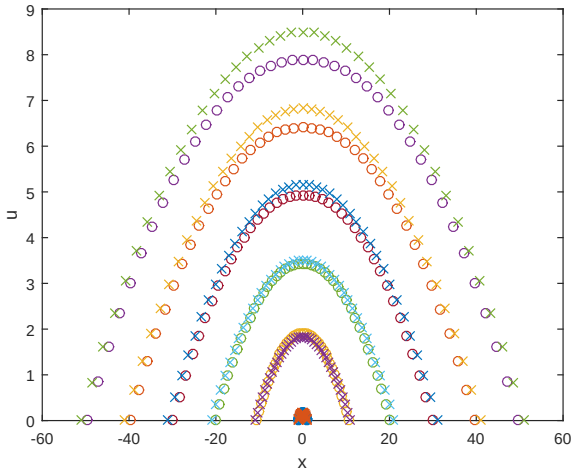


Figure 2: VSM solutions (circles) and exact solutions (crosses) of the problem (42) with initial condition sampled from (43) at time intervals 10 (bottom to top) from $t = 1$ to $t = 51$ using $\Delta t = 1$.

with $u = 0$ on the boundary of the contracting interval has the self-similar solution

$$u(x,t) = \frac{1}{6}\left(t^{-3} - \frac{x^2}{t}\right) \tag{48}$$

(see Appendix). We use this equation to demonstrate the stability and accuracy properties of Algorithm 2 for a negative source term.

Initial conditions are taken from (48) at $t = 1$. Due to the symmetry about the origin, $x = 0$ is taken as the anchor point. At the initial time, $t = 1$ say, the initial condition $u(x, 1)$, total mass $\theta(1)$, normalised solution

$\overline{u}(x,1)$, and mass constants $\overline{c}_i$, 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

$$\dot{\theta} = -\int_0^{x_N} \frac{4}{9} t^{-4} \mathrm{d}\chi = -\frac{4}{9} t^{-4} x_N \tag{49}$$

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

$$\overline{v}_i = \frac{-u_i (u_x)_i + \int_0^{x_i} (4/9) t^{-4} \mathrm{d}\chi - \overline{C}_i \dot{\theta}}{u_i} = -(u_x)_i + \frac{4/9}{t^4} \frac{x_i}{u_i} - \frac{4/9}{t^4} \frac{\overline{C}_i \, x_N}{u_i} \tag{50}$$

The explicit scheme for $\overline{u}_i$ is (35) with $\overline{v}_i$ given by (50). The semi-implicit scheme for this problem is (37). The $\overline{\lambda}_L, \overline{\lambda}_R$ are not of opposite sign for the parameters studied here so the semi-implicit scheme can be used to compute u_0 at the origin.

The total mass θ is found from (36) with $\dot{\theta}$ given by (49). The mesh is then obtained from (20) with one-sided differences.

Algorithm 2 of section 3 is run for 40 interior points. Four increasing Δt's are used, $\Delta t =$ 0.005, 0.01, 0.05, 0.1, in reaching the fixed time $t = 6$, forfeiting accuracy as Δt increases monitored by the relative error in the l^2 norm of the solution and the l^∞ norm of the free boundary (see 3).

Δt	Relative error in u	Relative error in x_N
0.005	0.01462	0.01462
0.01	0.05061	0.01721
0.05	0.04672	0.01489
0.1	0.07789	0.06969

Table 3: Relative errors in u^n and x_N^n at $t = 6$ from the VSM method applied to the PDE (47) with initial condition (44), using 40 nodes and taking time steps $\Delta t =$ 0.005, 0.01, 0.05, 0.1.

Figure 3 shows the approximate solutions (circles) together with the exact solution (crosses) using $\Delta t = 0.1$ from $t = 1$ to $t = 1.5$.

The solution is always positive and the mesh monotonic.

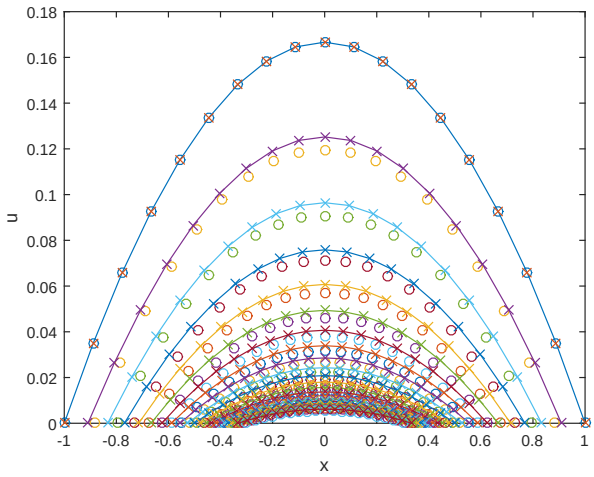


Figure 3: VSM solutions (circles) and exact solutions (crosses) of the PME problem (47) with a negative source term, with initial condition sampled from (48), at time intervals $\Delta t = 0.1$ (top to bottom) from $t = 1$ to $t = 1.5$.

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 *a posteriori* 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 differs 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 diffuses 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 diffusion problems with prescribed fluxes having analytic solutions, first for a non-trivial mass-conserving nonlinear diffusion problem (a porous medium equation with a quadratic diffusion coefficient), and then for two non mass-conserving nonlinear diffusion 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 first-order accurate in tine, so care is required in their use.

The extension to multidimensions is planned. The calculation of the velocity $\mathbf{v}$ in multidimensions has been considered elsewhere (see e.g. [1, 4]) in both finite diffeernce and finite element contexts. The PDE following the

motion (5) generalises to

$$u_t = -u\nabla \cdot \mathbf{v}$$

In a finite difference approach the $\nabla \cdot \mathbf{v}$ term may be approximated at any point in a mesh of triangles by a linear sum of values of a velocity potential ϕ at adjacent nodes. By introducing quotients of differences 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 define 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)

$$u(x,t) = \frac{1}{6}t^{\gamma}(1-\xi^2), \qquad \xi = \frac{x}{t^{\beta}} \tag{51}$$

Differentiation gives

$$u_t = \frac{1}{6}\gamma t^{\gamma-1}(1-\xi^2) + \frac{1}{6}t^{\gamma}(-2\xi\xi_t) = \gamma\frac{1}{6}t^{\gamma-1}(1-\xi^2) + \frac{1}{3}t^{\gamma-1}\beta\xi^2,$$

$$u_x = \frac{1}{6}t^{\gamma}(-2\xi\xi_x) = -\frac{1}{3}t^{\gamma-\beta}\xi,$$

$$uu_x = -\frac{1}{18}t^{2\gamma-\beta}\xi(1-\xi^2) = -\frac{1}{18}t^{2\gamma-\beta}\xi + \frac{1}{18}t^{2\gamma-\beta}\xi^3,$$

so that

$$(uu_x)_x = -\frac{1}{18}t^{2\gamma-2\beta} + \frac{1}{6}t^{2\gamma-2\beta}\xi^2,$$

Mutual scale-invariance of $\partial_t u$ and $\partial_x(u\partial_x u)$ holds if $\beta = \frac{1}{2}(1+\gamma)$.

Thus

$$u_t - (uu_x)_x = \frac{1}{6}\gamma t^{\gamma-1}(1-\xi^2) + \frac{1}{3}t^{\gamma-1}\beta\xi^2 + \frac{1}{18}t^{\gamma-1} - \frac{1}{6}t^{\gamma-1}\xi^2$$

$$= \frac{1}{6}t^{\gamma-1}\{\gamma - \gamma\xi^2 + 2(\gamma+1)\xi^2 + \frac{1}{3} - \xi^2\}$$

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

$$u_t - (uu_x)_x = \frac{1}{6} t^{\gamma-1} \left(\gamma + \frac{1}{3} \right)$$

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

If $\gamma = 1$ then $\beta = 1$, so $\xi = x/t$ and

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

is a self-similar solution of the partial differential equation

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

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

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

$$u(x,t) = \frac{1}{6} \left(t^{-3} - \frac{x^2}{t} \right)$$

is a self-similar solution of the partial differential equation

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

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

References

[1] M.J. Baines, M.E. Hubbard, and P.K. Jimack, *A Moving Mesh Finite Element Algorithm for the Adaptive Solution of Time-Dependent Partial Differential Equations with Moving Boundaries*, Appl. Numer. Math., 54, pp. 450-469, 2005.

[2] M.J. Baines, M.E. Hubbard, P.K. Jimack and A.C. Jones, *Scale-invariant Moving Finite Elements for Nonlinear Partial Differential Equations in Two Dimensions*, Appl. Numer. Math., 56, pp. 230-252, 2006.

[3] M.J. Baines, M.E. Hubbard, P.K. Jimack and R. Mahmood, *A moving-mesh finite element method and its application to the numerical solution of phase-change problems*, Commun. Comput. Phys., 6, pp. 595-624, 2009.

[4] M.J. Baines, M.E. Hubbard, and P.K. Jimack, *Velocity-based moving mesh methods for nonlinear partial differential equations*, Commun. Comput. Phys., 10, pp. 509-576, 2011.

[5] M.J. Baines, T.E. Lee, S. Langdon and M.J. Tindall, *A moving mesh approach for modelling avascular tumour growth*, Appl. Numer. Math., 72, pp. 99-114 (2013).

[6] M. J. Baines and T. E. Lee, *A large time-step implicit moving mesh scheme for moving boundary problems*, Num. Soln. PDEs, 30, pp.321-338 (2014).

[7] M.J. Baines, *Explicit time stepping for moving meshes*, J. Math Study, 48, pp. 93-105 (2015).

[8] K.W. Blake, *Moving mesh methods for nonlinear partial differential equations*, PhD thesis, University of Reading, UK, 2001.

[9] G.I. Barenblatt, *On some unsteady motions of fluids and gases in a porous medium*, Prik. Mat. Mekh, 16 (1952), pp. 67-68.

[10] N. Bird, *High Order Nonlinear Diffusion*, 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 moving-point approach to model shallow ice sheets: a study case with radially symmetrical ice sheets*, The Cryosphere, 10, pp. 1-14, doi: 10.5194/tc-10-1-2016 (2016).

[12] W. Cao, W. Huang, and R.D. Russell, *A Moving Mesh Method Based on the Geometric Conservation Law*, SIAM J. Sci. Comput., 24 (2002), pp. 118-142.

[13] T.E. Lee, M.J. Baines, and S. Langdon, *A finite difference moving mesh method based on conservation for moving boundary problems*, J. Comp. Appl. Math, 288, pp.1-17 (2015).

[14] A.V. Lukyanov, M.M. Sushchikh, M.J. Baines, and T.G. Theofanous, *Superfast Nonlinear Diffusion: Capillary Transport in Particulate Porous Media*, Phys. Rev. Letters, 109, 214501 (2012).

[15] D. Partridge, *Numerical Modelling of Glaciers: Moving Meshes and Data Assimilation*, PhD thesis, University of Reading, UK, (2013).

[16] R.E. Pattle, *Diffusion from an instantaneous point source with a concentration-dependent coefficient*, Quart. J. Mech. Appl. Math, 12, pp. 407-409, (1959).

[17] N. Sarahs, *Similarity, Mass Conservation, and the Numerical Simulation of a Simplified Glacier Equation*, SIURO, 9, http://dx.doi.org/10.1137/siuro.2016v9 (2016).

[18] A. Watkins, *A moving mesh finite element method and its application to population dynamics*, PhD thesis, Department of Mathematics and Statistics, University of Reading, UK, 2017.

[19] B.V. Wells, *A moving mesh finite element method for the numerical solution of partial differential equations and systems*, PhD thesis, Department of Mathematics, University of Reading, UK, 2005.