A moving-mesh finite difference scheme that preserves scaling symmetry for a class of nonlinear diffusion problems

A moving-mesh finite difference scheme that preserves scaling symmetry for a class of nonlinear diffusion problems

M. J. Baines and **N. Sarahs**

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

Corresponding author: m.j.baines@reading.ac.uk

Abstract

A moving-mesh finite difference scheme based on local conservation is presented for a class of scale-invariant second-order nonlinear diffusion problems with moving boundaries that (a) preserves the scaling properties and (b) is exact at the nodes for initial conditions sampled from similarity solutions. Details are presented for one-dimensional problems, the extension to multidimensions is described, and the exactness property is confirmed for two radially symmetric moving boundary problems, the porous medium equation and a simplistic glacier equation.

In addition, the accuracy of the scheme is also tested for non self-similar initial conditions by computing relative errors in the approximate solution (in the l_∞ norm) and the approximate boundary position, indicating superlinear convergence.

Keywords: Nonlinear diffusion, moving-meshes, scale-invariance, similarity, conservation, finite differences, porous medium equation, glacier equation, radial symmetry.

1 Introduction

Partial differential equations (PDEs) govern many physical processes that occur in branches of applied mathematics. However, due to the complexity of these equations the solution cannot always be determined analytically and numerical approximation becomes fundamental both for extracting quantitative solutions and for achieving a qualitative understanding of the behaviour of the solution.

In this paper we consider one-dimensional second-order scale-invariant nonlinear diffusion equations of the form

$$u_t = (uq)_x \qquad (a(t) < x < b(t)), \qquad (1)\ \boxed{\texttt{PDE}}$$

for a function $u(x,t)$, where q is of the form $\{p(u)_x\}^s$ with s an odd integer, and their radially symmetic counterparts, posed on finite moving domains. Typical boundary conditions for this problem consist of a Dirichlet condition on u and a flux condition on uv, where v is the boundary velocity. We assume here that $u = 0$ on the moving boundaries. In general the position of the boundary depends on the solution.

Many PDE problems that arise in practical applications possess scaling symmetries of the dependent and independent variables which are in some sense more fundamental than the equations themselves. In approximating such problems by numerical schemes it is desirable to construct algorithms that preserve these scaling properties, an objective beyond the reach of conventional numerical schemes based on fixed meshes in which the mesh depends on neither time nor the solution. The geometric integration of scale-invariant ordinary and partial differential equations (PDEs) was reviewed in Budd and Piggott in [11, 12] who considered the effectiveness of numerical methods in preserving the geometric structures of PDE problems, pointing to the need for moving-meshes (see also [13]).

Moving-mesh schemes, referred to as r-adaptive methods, are well suited to problems posed on finite moving domains since they are able to track the movement of the boundaries. Construction of these schemes varies but can be classified into two broad categories; mapping-based and velocity-based methods [19]. The former, which have been extensively studied in [10, 19, 14, 13], control the location of mesh points and are based on equidistribution. Velocity-based methods, on the other hand, rely on determining a velocity for each computational node in the mesh and advancing the nodal positions in time. In this paper we shall be concerned with a particular velocity-based moving-mesh finite difference method that uses local conservation and has been successfully applied to a number of different problems in [7, 18, 1, 34, 2, 3, 25, 4, 24, 23, 6, 16].

The main thrust of this paper is the construction of a scale-invariant moving-mesh scheme for nonlinear diffusion problems of the form (1) that is exact at the nodes for initial conditions sampled from a self-similar scaling solution (thus preserving a scaling symmetry).

The layout of the paper is as follows. In section 2 we recall the scaling properties of a PDE problem of the form (1) and the construction of self-similar scaling solutions. Details are given for two nonlinear diffusion equations of the form (1); a porous medium equation (PME) and a simplistic glacier equation (SGE). A moving-mesh finite difference scheme based on conservation of the local integral of u is then described in section 3 that propagates solutions exactly at the nodes when the initial condition is sampled from a self-similar solution. The algorithm is stated in section 3.2. The extension to radially symmetric PDEs is described in section 3.3. Section 4 shows numerical results using the PME and SGE which confirm that the errors in the solution at the nodes and the approximate boundary position are zero to within rounding error when the initial condition is sampled from a similarity solution. As an addendum, convergence rates from the algorithm are determined for more general initial conditions showing superlinear convergence. The paper ends with a summary including suggestions for further work.

2 Background

⟨sec:2⟩ The work of Budd et al [10, 11, 12, 13] has underlined the importance of preserving the geometric structures of the underlying PDE problem in constructing a moving-mesh method. In this section scale-invariance and similarity solutions are recalled and illustrated in the context of two nonlinear diffusion equations, a porous medium equation and a simplified glacier equation.

2.1 Scale-invariance

?⟨sec:2.1⟩? A PDE problem of the form (1) in one dimension exhibits scale-invariance if the scaling transformation

$$t = \lambda \hat{t}, \quad x = \lambda^{\beta} \hat{x}, \quad u = \lambda^{\alpha} \hat{u}, \quad q = \lambda^{\delta} \hat{q} \tag{2}$$

Transformations

maps the variables (t, x, u, q) to another set ($\hat{t}$, $\hat{x}$, $\hat{u}$, $\hat{q}$) for some arbitrary positive (group) parameter λ such that equation (1) remains the same in the transformed coordinates.

Substituting the scaling transformation (2) into PDE (1), it is easy to show that the powers α, β and δ satisfy $\alpha - 1 = \alpha + \delta - \beta$ (leading to $\beta - \delta = 1$). A further relation between the scaling powers depends on the particular form of the function $p(u)$ and will be described for two examples in section 2.3.

Moreover, since the total integral (mass)

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

satisfies

$$\begin{aligned}
\frac{\mathrm{d}\theta}{\mathrm{d}t} &= \int_{a(t)}^{b(t)} u_t\,\mathrm{d}\chi + u(b(t),t)\dot{b} - u(a(t),t)\dot{a} \\
&= \int_{a(t)}^{b(t)} (uq)_\chi\,\mathrm{d}\chi + u(b(t),t)\dot{b} - u(a(t),t)\dot{a} \\
&= [uq]_{a(t)}^{b(t)} + u(b(t),t)\dot{b} - u(a(t),t)\dot{a} = 0
\end{aligned}$$

by the $u = 0$ boundary condition, after substitution from (2)

$$\theta = \int_{a(\hat{t})}^{b(\hat{t})} \lambda^\alpha \hat{u}(\hat{\chi},\hat{t})\ \ \mathrm{d}(\lambda^\beta \hat{\chi}) = \lambda^{\alpha+\beta} \int_{a(\hat{t})}^{b(\hat{t})} u(\hat{\chi},\hat{t})\,\mathrm{d}\hat{\chi}$$

where the moving boundaries $a(t)$ and $b(t)$ transform in the same way as x, it follows that the total mass is invariant if and only if $\alpha + \beta = 0$.

2.2 Self-similar solutions

⟨sec:2.2⟩ A systematic approach in which the scaling transformation (2) may be used to construct exact solutions to scale-invariant PDE problems is as follows.

Solutions are sought such that $\lambda^\alpha u(x,t)$ is a function of $\lambda^\beta x$ and λt, which allows the number of independent variables of the differential equation to be reduced by one [8]. These solutions, termed similarity solutions or self-similar solutions, have contributed some of the greatest insights into nonlinear flows [8, 15]. Such a symmetry is useful since the resulting equation may be more easily solved than the original problem.

In order to construct such solutions we introduce a ‘similarity’ transformation which is invariant under the action of (2). Define the so-called similarity variables

$$\eta = \frac{u}{t^\alpha}, \quad \pi = \frac{q}{t^\delta}, \quad \xi = \frac{x}{t^\beta}. \qquad (3)\ \boxed{\texttt{190}}$$

By assuming functional relationships of the form

$$\eta = f(\xi), \quad \pi = g(\xi), \qquad (4)\ \boxed{\texttt{function}}$$

(where f and g are sufficiently differentiable functions) and substituting (2) into equation (1), a time-independent ODE satisfied by $\eta(\xi)$ and $\pi(\xi)$ is obtained. From (3) and (4) the solution is given in terms of x and t by

$$u(x,t) = t^\alpha f\left(\frac{x}{t^\beta}\right). \qquad (5)\ \boxed{\texttt{sss}}$$

For a fixed parameter ξ the solution may be described by the moving coordinate

$$\widehat{x}(\xi,t) = t^\beta \xi \qquad (6)\ \boxed{\texttt{param1}}$$

and the function

$$\widehat{u}(\xi,t) = u(\widehat{x}(\xi,t),t) = u(t^{\beta}\xi,t) = t^{\alpha}f(\xi), \qquad (7)$$?param2?

returning (5) on elimination of ξ. The velocity effecting the movement of $\widehat{x}(\xi,t)$ is given by

$$v(\widehat{x}(\xi,t),t) = \widehat{v}(\xi,t) = \frac{\partial \widehat{x}}{\partial t} = \beta\xi t^{\beta-1} = \frac{\beta\widehat{x}}{t}. \qquad (8)$$ 10.5

As outlined in [8, 9, 30], self-similar solutions often act as attractors to a wide class of other solutions in the sense that solutions of nonlinear diffusion problems of the form (1) with $s = 1$ and arbitrary initial data evolve asymptotically into a self-similar form. The result may be stated as follows: for a self-similar solution (5) of (1) and an arbitrary solution $w(x,t) \geq 0$ of (1) with the same mass and centre of mass, the function u will be a global attractor for w such that

$$\lim_{t\to\infty} t^{\beta}\|u - w\| = 0$$

where β is the scaling power found from scale-invariance and $|| \cdot ||$ is some norm. A proof of this result can be found in [22, 33, 29] which uses either the maximum principle or Lyapunov functions.

2.3 Examples

ffusionproblems⟩

2.3.1 A porous medium equation

In one dimension the porous medium equation (PME) is given by

$$u_t = (u^m u_x)_x \qquad (a(t) < x < b(t)) \qquad (9)$$ PME

where $m > 0$, which is of the form (1) with $s = 1$ and $p(u) = u^m u_x = (u^m)_x/m$. Typical boundary conditions are $u = 0$ at $x = a(t), b(t)$, implying zero net flux and hence constant global mass.

The porous medium equation has stimulated considerable interest from mathematicians, applied and pure, as well as in many fields; biology, heat radiation in plasmas, ground-water hydrology and more. A well-known application is to the flow of an isentropic gas through a porous medium. Other applications include biological modelling, where for example bone cartilage and muscle are modelled as porous media, assisting understanding of pathological conditions.

A simple derivation of equation (9) is as follows. The flow through a porous medium in 1D is governed by three model equations:

$$\begin{array}{lcll} \rho_t & = & -(\rho v)_x & \text{(continuity equation)} \\ v & = & -\kappa p_x/\mu & \text{(Darcy's law)} \\ p & = & \rho^{\gamma} & \text{(equation of state)} \end{array}$$

where ρ is the density, v is the velocity (given by Darcy's law), μ is the viscosity, κ is the permeability of the medium (taken to be a constant), p is the pressure and $\gamma > 0$ is the ratio of specific heats.

By substituting the equation of state into Darcy's law we obtain

$$v = -\kappa(\rho^{\gamma})_x/\mu = -\gamma\kappa\rho^{\gamma-1}\rho_x/\mu.$$

The continuity equation then becomes $\rho_t = \gamma\kappa(\rho^{\gamma}\rho_x)_x/\mu$. By scaling the constant $\gamma\kappa/\mu$ to unity and setting $\rho = u$ and $\gamma = m$ we obtain (9) where $u = u(x,t)$ is the density and $m = \gamma - 1$.

Due to the form of $p(u)$ and the boundary conditions equation (9) is scale invariant with $\delta = m\alpha - \beta$ and conserves global mass, so $\alpha + \beta = 0$ and $\beta - \delta = 1$, leading to

$$\alpha = -\frac{1}{m+2}, \qquad \beta = \frac{1}{m+2} \qquad \text{and} \qquad \delta = -\frac{m+1}{m+2}.$$

A solution of the ODE obtained from the substitution of (5) into (1) is $f(\xi) = t^{\alpha}(1-\xi^2)^{1/m}$, leading to the self-similar scaling solution

$$u(x,t) = \frac{1}{t^{1/(m+2)}} \left(\frac{m}{2(m+2)}\right)^{1/m} \left(1 - \left(\frac{x}{t^{1/(m+2)}}\right)^2\right)_+^{1/m}, \qquad (10)\ \boxed{\texttt{PMEsss}}$$

discovered independently by Barenblatt [8] and Pattle [27]. The notation + in equation (10) indicates restriction to the positive part of u, thus determining the support of the solution. The boundary at $\pm t^{1/m+2}$ moves with velocity $v = \pm(1/(m+2))t^{1/(m+2)-1}$, in accordance with (8).

2.3.2 A simplified glacier equation

For a glacier to form, snow must accumulate in one area over each year. This snow compresses into ice over years (or centuries). The weight of the accumulated snow and ice causes the glacier to move, and a simplistic one-dimensional glacier equation (SGE) for the ice thickness $u = u(x,t)$ is given by

$$u_t = (u^5 u_x^3)_x \qquad (0 < x < b(t)), \qquad (11)\ \boxed{\texttt{SGE}}$$

omitting any ongoing ice-accumulation and/or ice-removal processes. Equation (11) is a nonlinear evolution equation that contains the essential singularities inherent in the flow of ice in an ice sheet. The boundary conditions are $u_x = 0$ at $x = 0$ (the ice divide) and $u = 0$ at $x = b$ (the ice margin), implying zero net fluxes and hence constant total mass. Equation (11) is of the form (1) with $s = 3$, $q = u^4 u_x^3$ and $p(u) = (3/7)u^{7/3}$.

We give a brief derivation of equation (11). Under the assumption that there is no accumulation of snow or basal melting affecting the glacier, the continuity equation for the ice thickness in 1D is

$$u_t = -(uv)_x \qquad (12)\ \boxed{\texttt{continuity}}$$

where $v(x,t)$ represents the vertically-averaged ice velocity. Under the shallow ice approximation, see [21], and Glen's flow law (an established law for steady state ice deformation) the velocity is modelled as $v = -cu^4 u_x^3$ where c is a constant, assuming constant bed elevation (flat bed). In accordance with Van Der Veen in [32], $c = 2A\rho^3 g^3/5 > 0$, where ρ is the ice density and g represents gravity. By a choice of units, $c = 1$ and so the velocity can be written as

$$v = -\,u^4 u_x^3 = -(3/7)^3\{(u^{7/3})_x\}^3.$$

Hence we obtain equation (11), found by substituting v into equation (12), posed on the finite moving domain $0 \leq x \leq b(t)$.

It can be shown that equation (11) with the given boundary conditions is scale-invariant under the scaling transformations in (2) if the scaling powers α and β satisfy $\alpha - 1 = 8\alpha - 4\beta$ and $\alpha + \beta = 0$, implying that

$$-\alpha = \beta = 1/11.$$

A self-similar scaling solution, given in [20, 17], is therefore

$$u(x,t) = \frac{1}{t^{1/11}} \left(\frac{7}{4\sqrt[3]{11}}\right)^{3/7} \left(1 - \left\{\frac{x}{t^{1/11}}\right\}^{4/3}\right)_+^{3/7} \qquad (13)\ \boxed{\texttt{GEsss}}$$

where the notation + denotes the positive part of the solution, thus determining the extent $b(t)$ of the domain. The position $b(t)$ of the boundary is given by $\widehat{x}(t) = t^{1/11}$ and its velocity $v = (1/11)t^{-10/11}$, in accordance with (8).

3 A moving-mesh approach

⟨sec:3⟩ In general the extent of the domain of the solution of (1) depends on the solution itself, and so the approach taken to solve for u is crucial. A standard approach is to solve for u on a fixed domain and then adjust the boundary according to the boundary conditions. Another approach is to solve for u and the boundary position simultaneously. A useful device is to stretch the domain in proportion to the (unknown) boundary position and solve a modified PDE, although this procedure may affect the essential structure of the PDE [5]. A more consistent way of deforming the domain is based on a local conservation of mass, which determines a nodal velocity v (in terms of the solution u) and has the advantage that the subsequent recovery of u is algebraic [1, 23]. This approach is summarised below.

The Eulerian equation of conservation (continuity) for a conserved quantity u is

$$u_t + (uv)_x = 0 \qquad (14)\ \boxed{\texttt{334}}$$

where v is the Eulerian velocity, which is scale-invariant when v scales as $\lambda^{\beta-1}$. From (14) and the scale-invariant PDE (1) it follows that

$$(uq)_x + (uv)_x = 0,$$

yielding (given a boundary or anchor condition on v) the velocity

$$v(x,t) = -q \qquad (15)\ \boxed{\texttt{5}}$$

at all points of the domain (provided that $u \neq 0$). For the nonlinear diffusion equations (1) the function $q = \{p(u)_x\}^s$ where s is an odd integer. Hence the velocity (15) is

$$v(x,t) = -\{p(u)_x\}^s. \qquad (16)\ \boxed{\texttt{=}}$$

Note that if u is constant at the boundary $x = b(t)$, say, for all t then

$$\frac{Du}{Dt} = 0 = u_t + v_b u_x = (uq)_x + v_b u_x$$

where v_b is the boundary velocity, from which

$$v_b = -\{(uq)_x / u_x\} \qquad (17)\ \{\textbf{?}\}$$

if $u_x \neq 0$. Thus the boundary velocity depends on the solution, which is often the case with moving boundary problems. In particular, if $u \to 0$ at the boundary,

$$v_b = -\lim_{u\to 0}\{(uq)_x/u_x\} = -\lim_{u\to 0}\{q\}_{u=0} = -\lim_{u\to 0}\{p(u_x)\}^s \qquad (18)\ \boxed{\texttt{2.5}}$$

by l'Hopital's Rule. The velocity (18) is identical to the velocity (16) derived from the local mass principle (14) at the boundary.

A deformation of the domain is defined by integrating the ODE

$$\frac{\mathrm{d}\widehat{x}}{\mathrm{d}t} = v(\widehat{x}(t), t) \qquad (19)\ \boxed{\texttt{341}}$$

for a moving coordinate $\widehat{x}(t)$ with initial condition $\widehat{x} = x$. Subsequently the solution $u(\widehat{x}(t), t)$ is determined from the Lagrangian form of conservation,

$$\int_{\widehat{x}_1(t)}^{\widehat{x}_2(t)} u(\chi, t)\mathrm{d}\chi = \text{constant} \qquad (20)\ \boxed{\texttt{343}}$$

for any $a(t) \leq \widehat{x}_1(t) < \widehat{x}_2(t) \leq b(t)$.

The steps (16), (19), and (20) are scale-invariant under the transformation (2).

We describe a finite difference scheme based on this approach.

3.1 A moving-mesh finite difference scheme

?⟨sec:3.1⟩? Consider a one-dimensional mesh with time-dependent mesh points

$$a(t) = x_0(t) < x_i(t) \ldots, < x_N(t) = b(t)$$

where $a(t)$ and $b(t)$ are the (moving) boundaries.

3.1.1 Generating the mesh velocities

?⟨sec:3.1.1⟩? The velocity is taken to be a finite difference approximation of (16). In the case where $s = 1$ a convenient second-order centred accurate approximation for v_j at any time t^n consists of a barycentric average of the two first-order approximations to $p(u)_x$ in adjacent cells (see e.g. [25]). Thus the mesh velocity v_j at any point x_j is calculated as

$$v_j = -\frac{\frac{p(u_{j+1})-p(u_j)}{(x_{j+1}-x_j)^2} + \frac{p(u_j)-p(u_{j-1})}{(x_j-x_{j-1})^2}}{\frac{1}{x_{j+1}-x_j} + \frac{1}{x_j-x_{j-1}}} \qquad (21)\ \boxed{\texttt{413}}$$

with truncation error

$$T_j = \frac{1}{6}(x_j - x_{j-1})(x_{j+1} - x_j)\, p(u)_{xxx}\big|_{x=\vartheta_i} \qquad (22)\ \boxed{\texttt{third}}$$

where ϑ_i is an intermediate value. It is simple to confirm that the formula (21) is scale-invariant under the transformation (2).

In the case of similarity the velocity is proportional to $\widehat{x}$ by (8) and equal to $-p(u)_x$ when $s = 1$ by (16). Thus $p(u)_x$ is proportional to $\widehat{x}$, the truncation error (22) vanishes, and the general second-order formula (21) is exact in this case.

Remark 1. *The same result is obtained by evaluating the derivative of the quadratic interpolating polynomial through $p(u_{j-1})$, $p(u_j)$ and $p(u_{j+1})$ at $x = x_j$, as we now show.*

For general values of the odd integer s (including $s = 1$) the velocity is $v = -\{p(u)_x\}^s$ by (16). Because the velocity is proportional to x in the case of similarity by (8), it follows that $p(u)_x$ is proportional to $x^{1/s}$. Then by integration (taking the origin of x at a point where $p(u)$ vanishes) the function $p(u)$ is proportional to $x^{1+1/s}$ and hence $\{p(u)\}^s$ is a monomial $Q(x)$ of degree $1 + s$. The velocity in terms of $Q(x)$ is then

$$v = -\{p(u)_x\}^s = -\Big(\{Q(x)^{1/s}\}_x\Big)^s = -\Big\{(1/s)\Big(Q(x)^{1/s-1}Q_x\Big)\Big\}^s = -(1/s)^s Q(x)^{1-s}(Q_x)^s. \qquad (23)\ \boxed{\texttt{498}}$$

The evaluation of $Q(x_j) = \{p(u_j)\}^s$ at $x = x_j$ is straightforward. Moreover, since $Q(x)$ is a monomial of degree $1 + s$ the evaluation of Q_x at $x = x_j$ is exact if it is calculated by differentiating the interpolating polynomial of degree $1 + s$ through three adjacent values of $Q(x_j)$.

PME

For the PME we have $s = 1$ and $p(u) = (u^m)_x/m$ with $v = -(u^m)_x/m$. The velocity can therefore be calculated either from (21) or from (23) with $Q(x_j) = (u_j)^m/m$ and the derivative Q_x can be found by differentiating the quadratic interpolating polynomial through adjacent values of u_j^m/m.

SGE

For the SGE $s = 3$ and $p(u) = (3/7)u^{7/3}$ with $v = -\{p(u)_x\}^3$. The velocity can therefore be calculated from (23) with $Q(x_j) = (3/7)^3(u_j)^7$ and the derivative Q_x can be found by differentiating the quadratic interpolating polynomial through adjacent values of $(3/7)^3(u_j)^7$.

3.1.2 Advancing x(t)

?⟨sec:3.1.2⟩? The mesh point locations $x_j(t)$ can now be obtained via time integration of the ODE system

$$\frac{\mathrm{d}x_j}{\mathrm{d}t} = v(x_j, t), \qquad (j = 1, ..., N-1)$$

(dropping the hats). Let t^n be the time at the nth time step and x_j^n be the computed mesh point at the nth time step, i.e. x_j^n is $x_j(t)$ at $t = t^n$. Also, let $\Delta t = t^{n+1} - t^n$ be the time step from t^n to t^{n+1}, where Δt is constant, and v_j^n be the velocity at x_j^n at the nth time step.

We now seek a time-stepping scheme which is stable and avoids the overtaking of nodes (or mesh tangling). Often used is the explicit Euler time-stepping scheme,

$$x_j^{n+1} = x_j^n + \Delta t\, v_j^n, \qquad j = 1, ..., N-1 \tag{24}$$

?Euler?

which is only first order accurate and not exact for self-similar solutions.

Observe from (6) that the function $y = x^{1/\beta}$ is linear in t in the case of similarity. Hence the formula

$$y^{n+1} = y^n + \Delta t \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^n$$

generates y^{n+1} from y^n exactly. Since

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\mathrm{d}y}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}t} = \beta^{-1} x^{(1/\beta)-1}\, v,$$

the formula

$$y^{n+1} = y^n + \Delta t \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^n = y^n + \beta^{-1}\Delta t\, (y^n)^{(1-\beta)}\, v^n \tag{25}$$

720

is exact in the case of similarity. Since $y = x^{1/\beta}$ the formula (25) can be written as

$$x^{n+1} = x^n \left(1 + \beta^{-1}\Delta t \frac{v^n}{x^n}\right)^\beta .$$

We therefore choose the discrete time-stepping scheme

$$x_j^{n+1} = x_j^n \left(1 + \beta^{-1}\Delta t \frac{v_j^n}{x_j^n}\right)^\beta , \qquad j = 1, ..., N-1, \tag{26}$$

scaleinvarianttimeste

which is exact when v_j is the similarity velocity (8). It is simple to confirm that the formula (26) is scale-invariant under the transformation (2).

A similar device was described in [2, 31] which used a scaled (variable) time step. In particular, in [31] a similarity-based time-stepping scheme is implemented which is exact in the case of the similarity velocity for the simplistic glacier equation, obtained by rescaling the time variable rather than the spatial variable. The scheme chosen here depends on a preference for equal time steps.

3.1.3 Recovering the solution

?⟨sec:3.1.3⟩? The final step of the conservation-based finite difference method is to obtain the updated approximate solution u_j at the next time step. From the Lagrangian form (20) of the Eulerian conservation principle (14) we have

$$\int_{x_{j-1}(t^{n+1})}^{x_{j+1}(t^{n+1})} u(\chi, t^{n+1})\, \mathrm{d}\chi = c_j, \tag{27}$$

26.5

independent of t. The value of c_j is determined by the left hand side of (27) at the initial time.

A consistent approximation of (27) is the algebraic formula

$$(x_{j+1}^{n+1} - x_{j-1}^{n+1})\, u_j^{n+1} = \bar{c}_j \qquad (28)\ \boxed{\texttt{int}}$$

say, where the $\bar{c}_j$ are the initial values of the left hand side of (28), which yields

$$u_j^{n+1} = \frac{\bar{c}_j}{(x_{j+1}^{n+1} - x_{j-1}^{n+1})}. \qquad (29)\ \boxed{\texttt{updateu}}$$

The formula (28) is scale-invariant under the transformation (2).

Remark 2. *Provided that the x_j^{n+1} values are exact when v_j is the similarity velocity (8), the u_j^{n+1} values calculated from (29) are also exact at the nodes when the initial $\bar{c}_j$ values are calculated from the initial solution in a consistent way. Any linear quadrature of (27) can be used: (28) is the most convenient since it gives u_j^{n+1} explicitly.*

3.2 The numerical algorithm

⟨sec:3.2⟩ In summary, a scale-invariant moving-mesh algorithm for the approximate solution of nonlinear diffusion equations of the form (1) is as follows:

Given initial data with nodal points x_j^0 and values u_j^0, evaluate the $\bar{c}_j$'s from (28) at the initial time. Then for each time step:

(1) Compute the mesh velocities v_j using (21) (when $s = 1$) or (23).

(2) Move the mesh from time t^n to t^{n+1} to obtain x_j^{n+1} using the time-stepping scheme (26).

(3) Update the values u_j^{n+1} values at the next time step from equation (29).

Remark 3. *The solution is propagated exactly when the initial condition is sampled from a self-similar solution initially. Thus any vector of nodal values sampled from a self-similar solution is a fixed point of the scheme.*

3.3 Extension to multidimensions

⟨sec:3.3⟩

Due to the isotropy of the diffusion coefficient in the multidimensional form

$$u_t = \nabla \cdot (u\mathbf{q}) \qquad (30)\ \boxed{\texttt{PDE2}}$$

of PDE (1), scale-invariance demands that all coordinates scale in the same way, so PDE (30) must exhibit radial symmetry, i.e. it is sufficient to show that the numerical method of section 3.2 generalises to radially symmetric PDEs of the form

$$u_t = \frac{1}{r^{d-1}} \frac{\partial}{\partial r}(r^{d-1} q) \qquad (0 < r < R(t)), \qquad (31)\ \boxed{\texttt{rPDE}}$$

for a function $u(r, t)$, where r is the radial coordinate, q is of the form $\{p(u)_r\}^s$ with s odd, and d is the number of dimensions. Boundary conditions are $u = 0$ at the radial boundary and a reflection condition at $r = 0$, ensuring constant total mass.

The scaling powers β anad γ of section 2.2 are now altered but the functional form of equation (5) is the same [4]. Exact self-similar solutions are available for the PME [1] and for the SGE [17].

The application of the theory to multidimensions is therefore a straightforward generalisation of the one-dimensional algorithm of section 3.2 [26, 16]. The algorithm for the approximate solution of PDEs of the form (31) is as follows:

Given initial data with nodal points r_j^0 and values u_j^0, evaluate the $\bar{c}_j$'s from

$$\bar{c}_j = u_j^0\{(r_{j+1}^0)^d - (r_{j-1}^0)^d\} \tag{32}$$?radial?

at the initial time ($cf.$ (28)). Then at each time step:

(1) Compute the mesh velocities v_j using (21) (when $s = 1$) or (23) (with x replaced by r).

(2) Move the mesh from time t^n to t^{n+1} to obtain r_j^{n+1} using the time-stepping scheme (26) (with x replaced by r and the relevant value of β).

(3) Update the values u_j^{n+1} values at the next time step from

$$u_j^{n+1} = \frac{\bar{c}_j}{(r_{j+1}^{n+1})^d - (r_{j-1}^{n+1})^d}, \tag{33}$$?rupdateu?

($cf.$ (29)). The algorithm possesses the property of propagating exact values at the nodes for radially symmetric problems.

4 Numerical results

⟨sec:4⟩

4.1 Self-similar initial conditions

?⟨sec:4.1⟩? The moving-mesh algorithm of section 3.3 is implemented in Matlab for the PME and the SGE. The PME example chosen is the three-dimensional contaminant spreading problem ($cf.$ [24]) governed by the PDE

$$u_t = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 u^2 \frac{\partial u}{\partial r}\right)$$

with $u = 0$ on the moving boundary, for which $\beta = 1/8$. The SGE example is the two-dimensional ice cap problem [26, 16]

$$u_t = \frac{1}{r}\frac{\partial}{\partial r}\left(r u^5 \left(\frac{\partial u}{\partial r}\right)^3\right)$$

with $u = 0$ on the moving boundary, for which $\beta = 1/18$. In each case the algorithm propagates initial self-similar solutions exactly at the nodes (to within rounding error), as expected.

By contrast, if the time-stepping scheme (step 2 of the algorithm) is replaced by the more common forward Euler scheme, corresponding to $\beta = 1$ in (26), the algorithm reverts to finite difference algorithms already described in the literature, in [23] for the PME and in [26, 16] for the SGE, giving only approximate values at the nodes.

4.2 Non self-similar initial conditions

?⟨sec:4.2⟩? It is also of interest to investigate the accuracy of the algorithm of section 3.3 for more general initial conditions, based on the convergence rate obtained from a sequence of solutions in which the number of points N is progressively doubled (see e.g. [23, 26]). The algorithm is implemented using Matlab for the PME (with $m = 2$) and the SGE in one dimension using an initially equally spaced mesh. Convergence of the solutions are investigated at time $t = 2$ with $N = 10 \times 2^k$ where $k = 0, 1, 2, 3$ and $\Delta t = O(1/N^2)$, chosen on stability grounds (see e.g. [28]), using a relative l_∞ error calculated from

$$e_N(u) = \frac{\|u_N - u_{160}\|_\infty}{\|u_{160}\|_\infty}$$

where u_N is the approximate value found from the algorithm and u_{160} is regarded as a highly accurate solution. Similarly, the error in the boundary position is calculated from

$$e_N(X) = \frac{|X_N - X_{160}|}{X_{160}}$$

where X_N is the approximate value found from the algorithm and X_{160} is regarded as a highly accurate solution.

PME

The initial condition for the PME (9) with $m = 2$ is taken as

$$u(x, t^0) = \frac{1}{4}(1 - x^2)^{1/2} + \frac{1}{2}(1 - x^2) \qquad (-1 < x < 1) \tag{34}$$ icPME

at $t^0 = 1$, differing from the self-similar solution (10) but avoiding the complication of waiting times [23].

Computed values of the relative errors $e_N(u)$ and $e_N(X)$ for $N = 10, 20, 40, 80$ against those for $N = 160$ (a highly accurate solution) are shown in Table 1. A relative error e_N of 1.2% is obtained with as few as 20 nodes. As N increases the relative errors for both the solution and the moving boundary suggest superlinear convergence.

N	Δt	Relative error $e_N(u)$	Relative error $e_N(X)$
10	0.01	1.2×10^{-2}	2.6×10^{-3}
20	0.0025	5.5×10^{-3}	9.0×10^{-4}
40	0.000625	2.4×10^{-3}	3.0×10^{-4}
80	0.00015625	8.7×10^{-4}	7.3×10^{-5}

Table 1: Relative errors $e_N(u)$ and $e_N(X)$ at $t = 2$ for the PME with $m = 2$ when the initial condition is (34).

⟨table3⟩

SGE

The initial condition for the SGE is taken to be

$$u(x, t^0) = c(1 - x^{4/3})^{3/7} - \frac{1}{2}(1 - x^2) \qquad (0 < x < 1) \tag{35}$$ icSGE

at $t^0 = 1$ where $c = (7/4\sqrt[3]{11})^{3/7}$, again differing from the self-similar solution (13) and avoiding the complication of waiting times [26, 16].

N	Δt	Relative error $e_N(u)$	Relative error $e_N(X)$
10	0.01	9.2×10^{-3}	5.4×10^{-4}
20	0.0025	2.7×10^{-3}	5.7×10^{-4}
40	0.000625	9.0×10^{-4}	1.5×10^{-4}
80	0.00015625	3.0×10^{-4}	3.9×10^{-5}

Table 2: Relative errors $e_N(u)$ and $e_N(X)$ at $t = 2$ for the SGE when the initial condition is (35).

⟨table4⟩

Computed values of relative error $e_N(u)$ and $e_N(X)$ for $N = 10, 20, 40, 80$ against those for $N = 160$ (taken to be a highly accurate solution) are shown in Table 2. For the smallest number of nodes ($N = 10$) the boundary position is computed very accurately (better than a 0.1% relative error). As in the case of the PME, as N increases the results for both the relative error of the solution and the relative position of the moving boundary suggest superlinear convergence.

5 Summary

?⟨sec:5⟩? In this paper we have shown that, for a class of second-order scale-invariant nonlinear diffusion equations with constant global mass in one dimension, the scale-invariant moving-mesh finite difference algorithm of section 3.2 propagates self-similar solutions exactly in the l_∞ norm. The property extends to radially symmetric problems in multidimensions.

The symmetry preserving property was illustrated for two radially symmetric examples, a porous medium problem (in three dimensions) and a simplified glacier equation (in two dimensions), confirming that the method carries a self-similar scaling solution exactly at the nodes (to within rounding error).

The accuracy of the method was also investigated for non self-similar initial conditions, the results indicating superlinear convergence in each case.

Further work in this area would be the extension to non mass-conserving problems, more general boundary conditions, and anisotropic diffusion.

References

BHJ05 [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.

BHJJ06 [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.

BHJM09 [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.

BHJ11 [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.

BLLT12 [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).

B15 [6] M.J. Baines, *Explicit time stepping for moving meshes*, J. Math Study, 48, pp. 93-105 (2015).

Bla01 [7] K.W. Blake, *Moving mesh methods for nonlinear partial differential equations*, PhD thesis, University of Reading, UK, 2001.

B [8] G.I. Barenblatt, *On some unsteady motions of fluids and gases in a porous medium*, Prik. Mat. Mekh, 16 (1952), pp. 67-68.

Barenblatt [9] G.I. Barenblatt, *Scaling, Self-similarity, and Intermediate Asymptotics*, Cambridge Univ. Press, 1996; *Scaling*, Cambridge Univ. Press, 2003.

BCHR99 [10] C.J. Budd, G.J. Collins, W.Z. Huang and R.D. Russell, *Self-similar numerical solutions of the porous medium equation using moving mesh methods*, Phil. Trans. Roy. Soc. A, 357, 1754 (1999).

BP01 [11] C.J. Budd and M.D. Piggott, *The Geometric Integration of Scale Invariant Ordinary and Partial Differential Equations*, J. Comp. Appl. Math., 128 (2001), pp. 399-422.

BP03 [12] C.J. Budd and M.D. Piggott, *Geometric integration and its applications*, Found. Comput. Math., Handbook of Numerical Analysis XI, ed. P.G. Ciarlet and F. Cucker, Elsevier, pp. 35-139, 2003.

BHR09 [13] C.J. Budd, W. Huang and R.D. Russell, *Adaptivity with moving grids*, Acta Numerica, 18, pp. 111-241 (2009).

BW09 [14] C.J. Budd and J.F. Williams, *Moving mesh generation using the parabolic MongeAmpre equation*, SIAM J. Sci. Comput., 31 (5), pp. 3438-3465 (2009).

Birkhoff [15] G. Birkhoff, *Hydrodynamics*, Princeton University Press, NJ (1950).

BBNP16 [16] 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).

Bueler [17] E. Bueler at al: *Exact solutions to the thermomechanically coupled shallow-ice approximation: effective tools for verification*, J. of Glaciology, 53, pp. 499-516 (2007).

CHR02 [18] 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.

CHR03 [19] W. Cao, W. Huang, and R. Russell, *Approaches for Generating Moving Adaptive Meshes: Location versus Velocity*, Appl. Numer. Math., 47 (2003), pp. 121-138.

Halfar [20] P. Halfar, *On the Dynamics of Ice Sheets*, J. Geophys. Res. Oceans, 86, pp. 11065-11072 (1981).

H [21] K. Hutter, *Theoretical Glaciology: Material Science of Ice and the Mechanics of Glaciers and Ice Sheets*, Springer, 1983.

K [22] S. Kamenomostoskaya, *The asymptotic behaviour of the solutions of the filtration equation*, Israel J. Math., 14, pp. 76-78, (1973).

LBL15 [23] 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).

Ly12 [24] 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).

P10 [25] J. Parker, *An invariant approach to moving mesh methods for PDEs*, MSc thesis, Math. Model. Comput., University of Oxford, UK, 2010.

Pa [26] D. Partridge, *Numerical Modelling of Glaciers: Moving Meshes and Data Assimilation*, PhD thesis, University of Reading, UK, (2013).

pattle [27] R.E. Pattle, *Diffusion from an instantaneous point source with a concentration-dependent coefficient*, Quart. J. Mech. Appl. Math, 12, pp. 407-409, (1959).

MM [28] K.W. Morton, D.F. Mayers, *Numerical Solution of Partial Differential Equations*, 2nd edition, Cambridge Univ. Press, 2005.

R [29] J. Ralston, *A Lyapunov functional for the evolution of solutions to the porous medium equation to self-similarity. II*, J. Math. Phys., (1984) 25, pp. 3124-3127.

S [30] P.L. Sachdev, *Self-symmetry and beyond: exact solutions of nonlinear problems*, Chapman & Hall/CRC Monogr. Surv. Pure Appl. Math., 113, (2000).

NS16 [31] 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).

Van2 [32] C.J. Van Der Veen, *Fundamentals of Glacier Dynamics*, 2nd Edition, Taylor and Francis, CRC Press, 2013.

Vaz [33] J.L. Vazquez, *An introduction to the mathematical theory of the porous medium equation*, In: Shape optimisation and free boundaries, M. Delfour ed. (1992), pp. 247-389.

Wel05 [34] 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.