SIMILARITY, MASS CONSERVATION, AND THE NUMERICAL SIMULATION OF A SIMPLIFIED GLACIER EQUATION

NICOLE SARAHS*

Abstract. A one-dimensional nonlinear scale-invariant PDE problem with moving boundaries (a model glacier equation) is solved numerically using a moving-mesh scheme based on local conservation and compared with a self-similar scaling solution, showing numerical convergence under mesh refinement. It is also shown, analytically and numerically, that the waiting-time exhibited by this problem for general data ends when the local profile close to the boundary becomes self-similar.

*Department of Mathematics and Statistics, University of Reading, PO Box 220, Reading, RG6 6AX, UK (m.j.baines@reading.ac.uk)

1. Introduction.

In order to simulate the flow of large ice masses such as those in the Antarctic and Greenland [15, 25] it is necessary to make credible predictions about the evolution of glaciers, which is essential for accurately estimating future contribution to sea-levels.

The evolution of a glacier (or ice flow) is governed by Stokes equations of fluid flow, but due to the complexity of these equations, and the computational cost of numerical simulations, simplifications are often made. The Shallow Ice Approximation (SIA), essentially designed for ice sheets, is the most widely used method, capturing the nonlinear dynamics and considerably reducing the complexity of the model equations and boundary conditions. The SIA is a zero-order model based upon three assumptions; the ice flow is primarily horizontal, the ice width is large in comparison with the ice depth, and the change in stress (and consequently strain) is negligible in the horizontal directions in comparison to the vertical [18, 23]. In this paper, a solution of a nonlinear diffusion equation is considered which is the isothermal, non-sliding SIA in the case of a flat bed, focusing on the concept of similarity [14, 15, 16]. It is called here the glacier equation for simplicity, although more complex dynamical models for glaciers are widespread in the glaciological literature [24, 28].

In order to solve the SIA numerical methods are used to approximate the solution. Results were first attained for grounded ice-sheets with fixed-grid models but were found to perform poorly without high resolution and so adaptive and moving techniques were developed to overcome this problem. There are many approaches but in this paper we consider a velocity-based moving-mesh method based on local conservation in which each node is assigned a time-dependent velocity which is used to advance the solution in time [1, 2, 3, 4, 12]. Self-similar solutions, originally derived in [9, 14, 17], allow us to verify numerical features of the model.

A goal of this paper is to construct a moving-mesh method that follows the symmetries of the glacier equation in which the grid points move at rates guided by the similarity analysis. Building on the scale invariance and conservation properties of this problem in [9, 14, 15, 16] this paper generates nodal velocities based on a local scale-invariant conservation principle and implements a similarity-based time-stepping scheme. It is shown that a moving-mesh constructed in this way can offer a relative error of less than 0.1% with as few as ten nodes. A comparison is provided between the similarity-based time-stepping scheme and the standard forward Euler time-stepping scheme. The approximation of the waiting-time property of the glacier margin is also discussed: analysis and numerical results show that for initial data that vanishes at the ice margin the waiting-time ends when the profile of the solution close to the boundary is proportional to that of the similarity solution.

A significant advantage of the moving-mesh method is that it follows the natural scaling symmetries of the glacier equation: however, there are some limitations. The moving-mesh method uses a fixed number of mesh points and may be unable to resolve fine structures arising during the evolution. Moreover, the method inherits a form of instability known as mesh tangling in which nodes overtake one another, avoided here by restricting the time step.

1.1. Outline of paper.

In section 2 we illustrate similarity solutions in the context of the glacier equation (GE) with symmetric initial data, using concepts presented in section 1.2, below.

Section 3 describes a local mass-conserving method, using finite differences, as a means of approximation. From a local mass conservation principle an equation is derived for the velocities of the mesh points, which are used to move the grid. Importance is placed on preserving the symmetry of the scheme in time, so a special similarity-based time-stepping strategy is used.

Section 4 contains an implementation of the numerical method of section 3 to obtain numerical results for the GE, where the initial data is taken to be the self-similar scaling solution. An assessment is made of the accuracy of the numerical method by computing the relative error for varying numbers of mesh points $N\psi$and time step Δt such that $\Delta t = O(1/N^2)$ (Compare with [9] where the stability restriction $\Delta t = O(1/N^2)$ is used in the case of an explicit fixed mesh method (see e.g. [22])).

Section 5 focuses on the waiting-time phenomenon exhibited by the GE, in which the existing theory is discussed. Then, by analysing the behaviour of the initial velocity at the boundary, the onset of the waiting-time is identified. By looking at when the waiting-time ends, i.e. when the boundary first moves, it is shown that the critical shape of the solution close to the boundary is proportional to the self-similar solution described in section 2. As in section 4, the moving-mesh method outlined in section 3 is implemented in Matlab using the similarity-based time-stepping scheme and is shown to closely approximate the self-similar solution in the l^2 norm.

1.2. Background.

1.2.1. Sophie Lie, Symmetry Methods and Similarity.

Most physical processes that arise in fields of engineering and science are governed by partial differential equations (PDEs). Due to the scale invariance of many of these problems it is often possible to find a change of variables such that the governing PDE is reduced to an ordinary differential equation (ODE) [5, 6]. These symmetry methods are highly algorithmic and amenable to symbolic computation [10]. In general, by symmetry here we mean that when carrying out a mapping from an object to itself or another object the structural properties of the original object will be preserved.

Symmetry methods of differential equations were recognised in the late nineteenth century by the Norwegian mathematician Sophus Lie. He consolidated widely-used methods of solving differential equations including reduction of order, integrating factor, and the method of undetermined coefficients till then restricted to special types of linear PDEs. Then, in extending these methods to more specialised cases he introduced continuous transformation groups, now called *Lie groups* [10]. These transformations are what we call symmetries. A powerful concept introduced by Lie's work is that, given that a differential equation is invariant under a one parameter Lie group of point transformations, the number of independent variables of a differential equation can be reduced by one [5, 6].

DEFINITION 1.1. *A symmetry of a differential equation is an invertible transformation that maps solutions to solutions.* Examples of transformations that map solutions to other solutions include translation, scaling and rotation.

An important class of symmetries is termed scale invariance.

DEFINITION 1.2. *(Scale Invariance): Consider a PDE of the general form:*

$$u_t = f(x, u, u_x, u_{xx}, ...). \tag{1.1}$$

If a scaling transformation can be determined which maps the coordinate system (u, x, t) to another system, ($\hat{u}$, $\hat{x}$, $\hat{t}$), such that

$$t = \lambda\hat{t}, \quad x = \lambda^{\beta}\hat{x} \quad \textit{and} \quad u = \lambda^{\gamma}\hat{u} \tag{1.2}$$

for some arbitrary positive scale parameter λ, with scaling powers β and γ, and if PDE (1.1) is identical in both the original and transformed coordinates, then it is said to be scale-invariant.

A useful invariant in certain cases is the global mass.

DEFINITION 1.3. *(Conservation of Mass Principle): Define global mass by*

$$\mathcal{M} = \int_0^{b(t)} u(x,t)\,\mathrm{d}x \tag{1.3}$$

where $b(t)$ is the moving boundary of the domain. By transforming $\mathcal{M}$ to the coordinate system $(\hat{u}, \hat{x}, \hat{t})$ by substituting for x and u, defined in (1.2), we obtain

$$\mathcal{M} = \int_0^{b(t)} (\lambda^{\gamma}\hat{u}) \ \mathrm{d}(\lambda^{\beta}\hat{x}).$$

By grouping the λ terms on the right-hand side and equation (1.3) we have

$$\lambda^{\gamma+\beta} \int_0^{b(t)} \hat{u}\,\mathrm{d}\hat{x} = \lambda^{\gamma+\beta}\mathcal{M} \tag{1.4}$$

so that the global mass is conserved if and only if $\gamma + \beta = 0$.

By the use of the transformation group (1.2) in Definition 1.2 a systematic approach allows the construction of self-similar solutions in which these invariant properties are preserved, leaving the differential equation invariant.

We first specify a time-dependent 'similarity' transformation by defining new variables.

DEFINITION 1.4. *(Scale-invariant similarity variables): Suppose PDE (1.1) has scale invariance (1.2). Then the variables defined by*

$$\eta = \frac{u}{t^\gamma} = \frac{\hat{u}}{\hat{t}^\gamma} \quad \text{and} \quad \xi = \frac{x}{t^\beta} = \frac{\hat{x}}{\hat{t}^\beta}$$

are called the similarity variables.

We can now define what it means for a solution to be a self-similar solution.

DEFINITION 1.5. *(Self-similar solution): By substituting scale-invariant similarity variables η and ξ from Definition 1.4 into PDE (1.1), we obtain a time-independent reduced-order differential equation for η in terms of ξ. Solutions of the form*

$$\eta = f(\xi), \tag{1.5}$$

where f is a sufficiently differentiable function, exhibit scaling symmetry because the number of independent and dependent variables have been reduced by one. To express self-similar solutions in terms of x and t, by substituting the scale-invariant similarity variables η and ξ into (1.5) we obtain the self-similar solution

$$u(x,t) = t^\gamma f\left(\frac{x}{t^\beta}\right),$$

for scaling powers β and γ.

REMARK 1. *The velocity effecting the time-dependent similarity transformation (with fixed ξ) is given by*

$$v = \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\xi t^\beta\right) = \beta \xi t^{\beta-1} = \frac{\beta x}{t}$$

where we have substituted $x = \xi t^\beta$.

Self-similar solutions are useful because they give exact solutions to many physical processes, as outlined in [5, 6]. In addition, they often act as attractors for more general solutions of the PDE problem with arbitrary initial data. In many problems, if we take the self-similar solution u and another arbitrary solution v to a problem of the form (1.1), both of which have the same mass and centre of mass, then the self-similar solution u will often be a global attractor for v such that

$$t^\beta \, ||u - v|| \to \ 0 \quad \text{as} \quad t \to \infty$$

where β is found from scale invariance and $|| \cdot ||$ is some norm. This corresponds to the self-similar solution presenting the correct asymptotic behaviour of the system over a long time period [11].

1.2.2. Waiting-times. Under certain conditions the boundaries of a PDE problem which depends on its solutions may remain fixed for a finite time before they start to move (and don't stop). This period of time is called the *waiting-time period*, during which the initial density redistributes itself before any movement occurs at the boundaries [21].

The existence of a waiting-time period was conjectured by Knerr in [20], whose motivation in seeking such solutions stemmed from observations he made into physical applications. For example, when observing water vapour in dry powder he found that, before spreading out, sections of the water vapour remained localised for periods of time. Also, by observing a blob of viscous liquid placed on a horizontal plane he found that it had a stationary perimeter over a non-zero time interval, before spreading out under gravity. His work stimulated further investigations into waiting-time behaviour, in particular by Lacey and Taylor, see [21], who described the behaviour mathematically.

2. A One-Dimensional Glacier Equation.

A nonlinear diffusion problem, the 1D glacier equation (GE), is studied in this paper, namely

$$u_t = \left(u^5 u_x^3\right)_x \tag{2.1}$$

where $u(x,t)$ is the ice thickness. This is a nonlinear evolution equation of parabolic type, which contains the essential non-singularities in the flow of ice in a glacier.

For a glacier to form, snow must accumulate in one area over each year. This snow compresses into ice over years (or centuries in cold regions). The weight of the accumulated snow and ice causes the glacier to move, and equation (2.1) is a model for this motion, though it does not include any ongoing ice-accumulation and/or ice-removal processes.

The study of the GE is important since the effects of glacier movement are of great environmental relevance. For instance, predictions have been made for the next one hundred years by climatologists: in accordance with the IPCC, see [26], it is anticipated that, with expectations of an increase in temperature, even though mountain glaciers make up a low percentage of the world's land-ice area they will contribute significantly to an increase in sea levels.

We now give a brief derivation of equation (2.1). 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 given by

$$u_t = -(uv)_x \tag{2.2}$$

where $v(x,t)$ represents the vertically-averaged ice velocity.

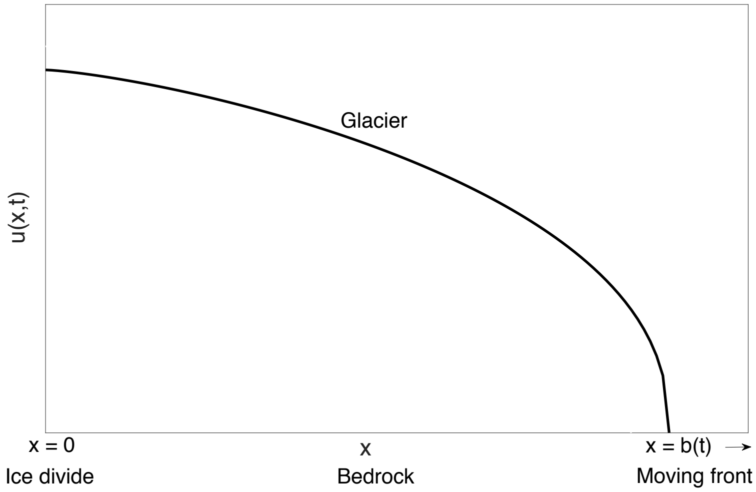

FIG. 1. *A one-dimensional glacier profile.*

Under the shallow ice approximation, see [18], the velocity is modelled as $v = -cu^4u_x^3$ assuming constant bed elevation (flat bed). Under Glen's flow law (an established law for steady state ice deformation), and in accordance with Van Der Veen in [28], $c = 2A\rho^3g^3/5 > 0$, where ρ is the ice density and g represents gravity. Without loss of generality (by a choice of units) we take $c = 1$ and so the velocity can be written as

$$v = -\,u^4u_x^3.$$

Hence, we have the 1D GE (2.1), found by substituting v into equation (2.2), on a finite moving domain $0 \leq x \leq b(t)$, say. As shown in Figure 1, the left-hand boundary $x = 0$ represents the top of the glacier, where $u_x = 0$ is assumed, and the right-hand boundary $x = b(t)$ (the moving front) has the condition $u = 0$ as the ice descends to meet the bedrock. Thus, the boundary conditions for this problem are given by $u(b(t),t) = 0$ and $u_x(0,t) = 0$.

The GE is scale-invariant and conserves global mass under the boundary conditions. From equation (1.4), the scaling powers β and γ satisfy $\gamma + \beta = 0$. From this, and under the transformations (2.1), β and γ are given by

$$\beta = \frac{1}{11} \quad \text{and} \quad \gamma = -\frac{1}{11}.$$

The self-similar solution is given by [17]

$$u(x,t) = \frac{1}{t^{1/11}} \left[\frac{7}{4\sqrt[3]{11}} \left(1 - \left(\frac{x}{t^{1/11}} \right)^{4/3} \right) \right]_{+}^{3/7} \tag{2.3}$$

where the notation $(.)_{+}^{3/7}$ denotes the positive part of the solution. The moving boundary occurs at $x = b(t) = t^{\beta} = t^{1/11}$.

Figure 2 illustrates the evolution of the self-similar solution of the GE, plotted at times $t = 1, 2, 3$.

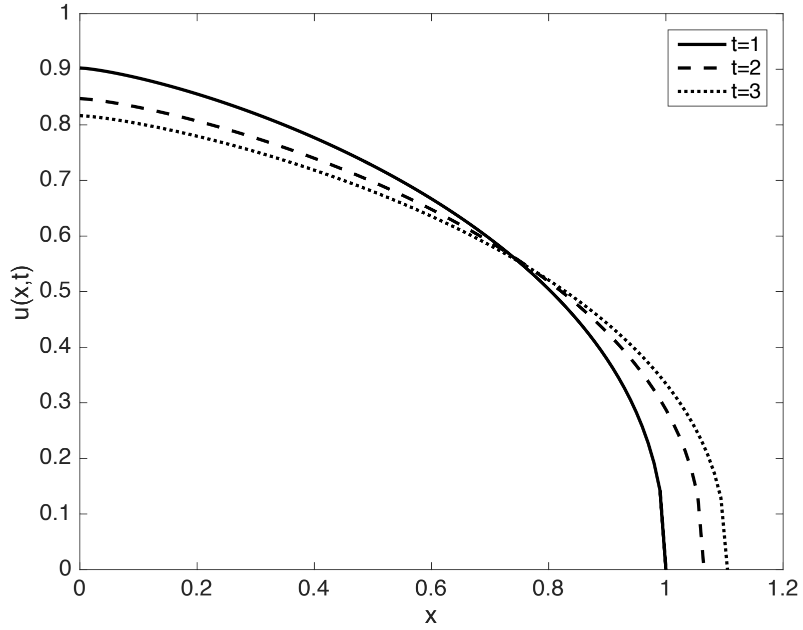

FIG. 2. *The self-similar solution of the GE (2.3) at times* $t = 1, 2, 3$.

3. Numerical Approximation.

This section describes a moving-mesh numerical method, chosen to display scale-invariant properties identical to those in equation (2.1). In subsection 3.1.1 mesh point velocities derived from a local conservation principle are used to move the grid points. Two time-stepping strategies are then stated in subsection 3.1.2, forward Euler and a similarity-based time-stepping (which has zero local truncation error under similarity by making a change of variable). The final step of the numerical method, where the updated approximate solutions are recovered at the next time step, is described in subsection 3.1.3.

3.1. A Moving-Mesh Method.

In order to obtain approximate solutions of optimal accuracy without excessive increase in computational effort, adaptive procedures make modifications to the existing numerical mesh in response to changes in the dependent variable. Modifications to the mesh include refinement, coarsening, relocating and/or adjusting the approximation basis. These procedures can generally be categorised in one of three main classes, defined in Table 1.

As we are considering time-dependent problems on moving domains which depend on the solutions of the PDE, our interest lies in *r*-refinement adaptivity, also known as *moving-mesh* methods, since, in accordance with Budd, Huang and Russell in [12], only by varying the solution and mesh simultaneously can solutions display scale-invariant properties identical to those in the underlying differential equation. A mesh point located at the moving boundary moves with the boundary, so no interpolation is required for the boundary.

Relocation of mesh points is achieved when each mesh point x is assigned a velocity v, by which it moves, subsequently advancing the solution in time [1, 12]. The mesh velocities may be generated from an appropriate choice of monitor

Procedure	Definition
h-refinement	The most prevalent form of mesh adaptivity involving refinement and/or coarsening of mesh points, usually carried out by a repeated subdivision process.
p-refinement	Polynomials of higher degree represent the solution locally, to improve accuracy, as well as to approximate difficult derivatives.
r-refinement	A mesh adaptivity which maintains the number of existing mesh points, relocating them at each time step, strategically across the domain.

TABLE 1
Adaptive Procedures.

function, defined as $M(x, u, u_x, u_{xx}, ...)$, which may be a function of the solution u and/or its derivatives. In order for the *r*-refinement numerical scheme to display self-similar properties of the analytic solution the monitor function should be chosen in such a way that it exhibits the invariant property (1.2) given in Definition 1.2. With this in mind, a sensible choice is $M(u) = u$ which determines a local conservation principle, consistent with global mass conservation. This monitor displays the appropriate invariant properties under the scalings in Definition 1.2, allowing both the discrete solution and the moving mesh to exhibit the same scale invariance.

3.1.1. Generating Mesh-Velocities. Consider a one-dimensional mesh with time-dependent mesh points $x_j(t)$, $j = 0, 1, 2, ...N$, where $x_0(t) = 0$ and $x_N(t)$ are the (moving) boundaries. We move the mesh such that the local integrals

$$\int_{x_{j-1}(t)}^{x_j(t)} u \, \mathrm{d}x = c_j \qquad j = 1, ..., N-1 \tag{3.1}$$

where the c_j are constant in time. By differentiating with respect to time using Leibnitz' Integral Rule,

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{x_{j-1}(t)}^{x_j(t)} u \, \mathrm{d}x = \int_{x_{j-1}(t)}^{x_j(t)} u_t \, \mathrm{d}x \, + \, [uv]_{x_{j-1}(t)}^{x_j(t)} = 0. \tag{3.2}$$

Substituting for u_t from the GE (2.1) a mesh-velocity v_j is generated.

Equivalently, by taking the range of integration to be $(0, x_j(t))$ and substituting for u_t from equation (2.1), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_0^{x_j(t)} u \, \mathrm{d}x = \int_0^{x_j(t)} u_t \, \mathrm{d}x \, + \, [uv]_0^{x_j(t)} = \left[u^5 u_x^3 \, + \, uv\right]_0^{x_j(t)} .$$

Then since $v = 0$ at $x = 0$ we have

$$u^5 u_x^3 \, + \, uv = 0$$

at the point x_j. Provided that $u \neq 0$ we obtain the mesh-velocity v for the GE at any point $x_j(t)$ as

$$v = -u^4 u_x^3 = -\left[\left(\frac{3}{7} u^{7/3}\right)_x\right]^3 .$$

Assuming continuity of v as $x \to x_N$ this equation also holds at the boundary.

Applying upwind finite differences at the mesh points an approximation to the velocity is

$$v_j = -\left[\frac{3}{7}\left(\frac{u_j^{7/3} - u_{j-1}^{7/3}}{x_j - x_{j-1}}\right)\right]^3 \qquad j = 1, ..., N. \tag{3.3}$$

New positions of the mesh can be obtained by time integration, specifically as shown in the following section.

3.1.2. Advancing $x(t)$. Now that an approximation to the velocity v_j has been constructed, the mesh point locations $x_j(t)$ can be obtained via time integration of the ODE system

$$\frac{\mathrm{d}x_j}{\mathrm{d}t} = v(x_j, t) = v_j \qquad j = 1, ..., N-1. \tag{3.4}$$

In this section we consider the forward Euler time-stepping scheme; a consistent and (conditionally) stable method (see [27]). To estimate the accuracy of the scheme we focus on the local truncation error (LTE), known to be $O(\Delta t)$ [27]. Although the scheme is scale-invariant, the x_j location are not exact when the time step is fixed due to the LTE. However, we may construct a related time-stepping procedure scaled in such a way that under the scale-invariant transformations (1.2) the LTE vanishes. When the time-stepping is implemented numerically, as in subsection 4.2, comparisons can be made between the two time-stepping schemes.

Forward Euler time-stepping. 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 the nth time step.

Forward Euler time-stepping is given by

$$x_j^{n+1} = x_j^n + \Delta t\, v_j^n \qquad j = 1, ..., N-1. \tag{3.5}$$

By substituting the velocity v_j given by equation (3.3) we obtain new mesh positions

$$x_j^{n+1} = x_j^n - \Delta t \left[\frac{3}{7}\left(\frac{(u_j^n)^{7/3} - (u_{j-1}^n)^{7/3}}{x_j^n - x_{j-1}^n}\right)\right]^3 \qquad j = 1, ..., N-1. \tag{3.6}$$

Scale Invariance. Equation (3.4) and the forward Euler time-stepping scheme, given by equation (3.5), are scale-invariant under the transformations (1.2) given in Definition 1.2. To see this, by substituting $t = \lambda\hat{t}$ and $x = \lambda^\beta \hat{x}$, we have from (3.4) and (3.5)

$$\begin{aligned} v(x_j, t) &= \frac{\mathrm{d}x_j}{\mathrm{d}t} = \frac{\mathrm{d}(\lambda^\beta)\hat{x_j}}{\mathrm{d}(\lambda\hat{t})} = (\lambda^{\beta-1})\frac{\mathrm{d}\hat{x_j}}{\mathrm{d}\hat{t}} \\ v(x_j^n, t) &= \frac{x_j^{n+1} - x_j^n}{\Delta t} = \frac{(\lambda^\beta \hat{x_j})^{n+1} - (\lambda^\beta \hat{x_j})^n}{\Delta(\lambda\hat{t})} = (\lambda^{\beta-1})\frac{\hat{x}_j^{n+1} - \hat{x}_j^n}{\Delta\hat{t}}. \end{aligned}$$

Since the power of λ is identical in both cases, they cancel and thence they are scale-invariant under the transformations in Definition 1.2.

Similarity-based time-stepping. As in [3], by re-scaling the time variable by the substitution $s = t^\beta$, we obtain a scheme with a vanishing LTE in the case of similarity.

First, let $s = t^\beta$ which implies $t = s^{1/\beta}$. It follows that $\mathrm{d}t = (1/\beta)s^{(1/\beta)-1}\mathrm{d}s$, so that

$$\frac{\mathrm{d}x}{\mathrm{d}s} = \frac{\mathrm{d}t}{\mathrm{d}s}\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{1}{\beta}s^{(1/\beta)-1}\, v(x, s^{1/\beta}). \tag{3.7}$$

Then, to obtain the invariant scheme, we discretise equation (3.7), where after some rearrangement and since, to leading order, $\Delta t \approx \mathrm{d}t = (1/\beta)s^{(1/\beta)-1}\mathrm{d}s \approx \Delta s\,(1/\beta)s^{(1/\beta)-1}$ the explicit Euler scheme (3.5) can be written

$$x_j^{n+1} = x_j^n + \Delta t\, v_j^n = x_j^n + \Delta s\,\frac{1}{\beta}(t^n)^{1-\beta}\, v_j^n = x_j^n + \Delta s\,\frac{1}{\beta}s^{(1/\beta)-1}\, v_j^n. \tag{3.8}$$

To obtain the LTE of (3.8) we expand $x(s)$ in the neighbourhood of s using a truncated Taylor series expansion to obtain

$$x(s + \Delta s) = x(s) + \Delta s\frac{\mathrm{d}x}{\mathrm{d}s} + \frac{1}{2}(\Delta s)^2\left.\frac{\mathrm{d}^2x}{\mathrm{d}s^2}\right|_\sigma$$

such that $s < \sigma < s + \Delta s$, so that the LTE at time $s = s^n$ for the scheme (3.8) is

$$\mathrm{LTE} = \frac{1}{2}\Delta s \left.\frac{\mathrm{d}^2 x}{\mathrm{d}s^2}\right|_{\sigma^n} \quad : \quad s^n < \sigma^n < s^{n+1}.$$

Now, since $x = \lambda^\beta \hat{x}$ and $s = t^\beta = (\lambda \hat{t})^\beta$, under similarity it follows that x is linear in s and so the LTE vanishes. The change in s, given by $\Delta s = s^{n+1} - s^n$ can be taken to be constant. From $t = s^{1/\beta}$,

$$t^{n+1} = (s^{n+1})^{1/\beta.}$$

Now that we have advanced x_j, we are left to recover the solution u_j at the mesh points.

3.1.3. Recovering the Solution u. The final step of the numerical method is to obtain the updated approximate solution u_j at the next time step. For any interval, the value of the integral in equation (3.1) is constant in time. Therefore, it follows that the integral is the same at the initial time t^0, i.e.

$$\int_{x_{j-1}(t^{n+1})}^{x_j(t^{n+1})} u \,\mathrm{d}x = \int_{x_{j-1}(t^0)}^{x_j(t^0)} u \,\mathrm{d}x = c_j.$$

We can find the value of c_j at time t^0 from the initial condition.

Approximating the equivalent condition

$$\int_{x_{j-1}(t^{n+1})}^{x_j(t^{n+1})} u \,\mathrm{d}x = \bar{c}_j$$

where $\bar{c}_j = c_j + c_{j+1}$ by the midpoint rule the approximate solution u_j satisfies

$$(3.9) \qquad (x_{j+1}^{n+1} - x_{j-1}^{n+1})\, u_j^{n+1} = \int_{x_{j-1}(1)}^{x_{j+1}(1)} u \,\mathrm{d}x = \bar{c}_j$$

which implies

$$(3.10) \qquad u_j^{n+1} = \frac{\bar{c}_j}{(x_{j+1}^{n+1} - x_{j-1}^{n+1})},$$

where $\bar{c}_j$ is approximated from the integral in equation (3.9) by $\bar{c}_j = (x_{j+1} - x_{j-1})\, u_j$ at the initial time. This formula is implemented in the numerical method to find the updated values of u_j at the next time step.

3.1.4. The Numerical Algorithm. In summary, the overall scheme is:

Given the initial mesh with points x_j^0 and values u_j^0 such that $j = [1, N]$, evaluate the c_j's at the initial time $t^0 = 1$. Then for each time step:

- Compute the interior mesh velocities v_j using equation (3.3).
- Time step the mesh, from t^n to t^{n+1}, to obtain x_j^{n+1} using either the forward Euler time-stepping scheme given by equation (3.5) or the similarity-based time-stepping scheme (3.8).
- Update the values u_j^n at the next time step by recovering the solution u_j^{n+1} at interior points from equation (3.10), using a reflection principle to evaluate u_0^{n+1} and the boundary condition for u_N^{n+1}.
- Repeat for each time step until the desired final time is reached.

4. Numerical Results. This section concerns an implementation of the numerical method outlined in section 3, obtaining results for equation (2.1). The initial condition is taken to coincide with the self-similar solution (2.3) described in section 2, and in subsection 4.1 by computing the relative error for varying number of mesh points and time steps the accuracy of the numerical method is assessed. Subsection 4.2 concerns a comparison of the two time-stepping strategies that were outlined in section 3.1.2.

4.1. Convergence of the Approximations. We implement the moving-mesh method outlined in subsection 3.1.4 for the GE, where we generate approximate solutions using a Matlab program.

The one-dimensional GE of section 2 is given by equation (2.1) in the domain $t > 1,\ x \in [0, b(t)]$. Here we take the origin at the left-hand boundary and the right-hand boundary (the moving front) to be $b(t) = 1$. Boundary conditions for this problem are given by $u_x(0,t) = 0$ and $u(1,t) = 0$.

The self-similar solution (2.3) described in section 2 is taken as the initial data, so at the initial time $t = 1$,

$$u(x,1) = \left[\frac{7}{4\sqrt[3]{11}}\left(1 - x^{4/3}\right)\right]^{3/7}, \quad x \in [0,1]. \tag{4.1}$$

Results showing the monotonicity of the mesh are shown in Figure 3, where even-numbered node positions are plotted every 10 time steps with $N = 40$. By trial and error we find that $\Delta t = 0.01$ gives monotonicity of the mesh.

For the explicit Euler scheme, Δt is chosen as discussed in section 3.1.4 on scaling grounds such that $\Delta t = O(1/N^2)$ to avoid instability. We examine the convergence of the moving-mesh method as the number of mesh points (nodes) N increases and as Δt decreases. We approximate the solution at $t = 3$ and compute results for $N = 10 \cdot 2^k,\ k = 0, 1, 2, 3$. We use a relative error given by

$$Re = \frac{\|u_{\text{exact}} - u_{\text{approx}}\|_2}{\|u_{\text{exact}}\|_2}, \qquad \|u_{\text{exact}}\|_2 \neq 0.$$

We investigate the hypothesis that the relative error computed for N mesh points satisfies

$$(Re)_N \propto \frac{1}{N^p} \tag{4.2}$$

where p for N large is the estimated order (of convergence) of the scheme. Computations are carried out for $N = 10 \cdot 2^k,\ k = 0, 1, 2, 3$, and p is recovered from hypothesis (4.2) by calculating

$$\frac{(Re)_N}{(Re)_{2N}} \sim \frac{1/N^p}{1/(2N)^p} = 2^p.$$

Results are computed using the moving-mesh method given in section 3.1 with explicit Euler time-stepping (3.5).

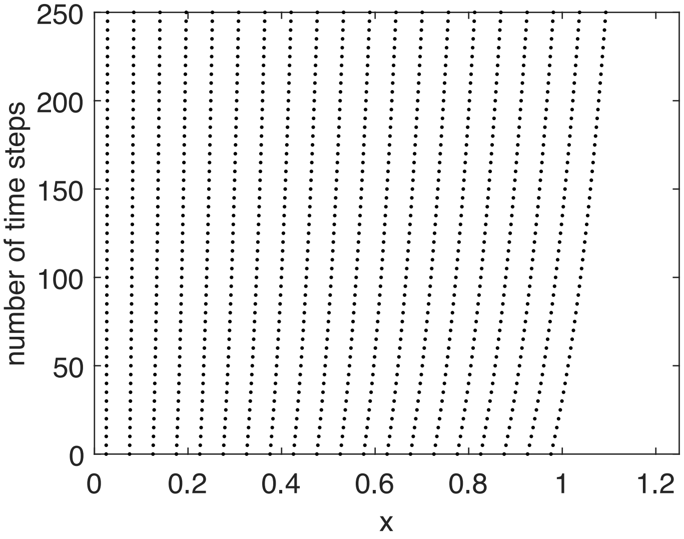

FIG. 3. *The evolution of even-numbered nodal trajectories plotted at* 10 *step intervals against their initial position x for $N = 40$, showing monotonicity of the mesh.*

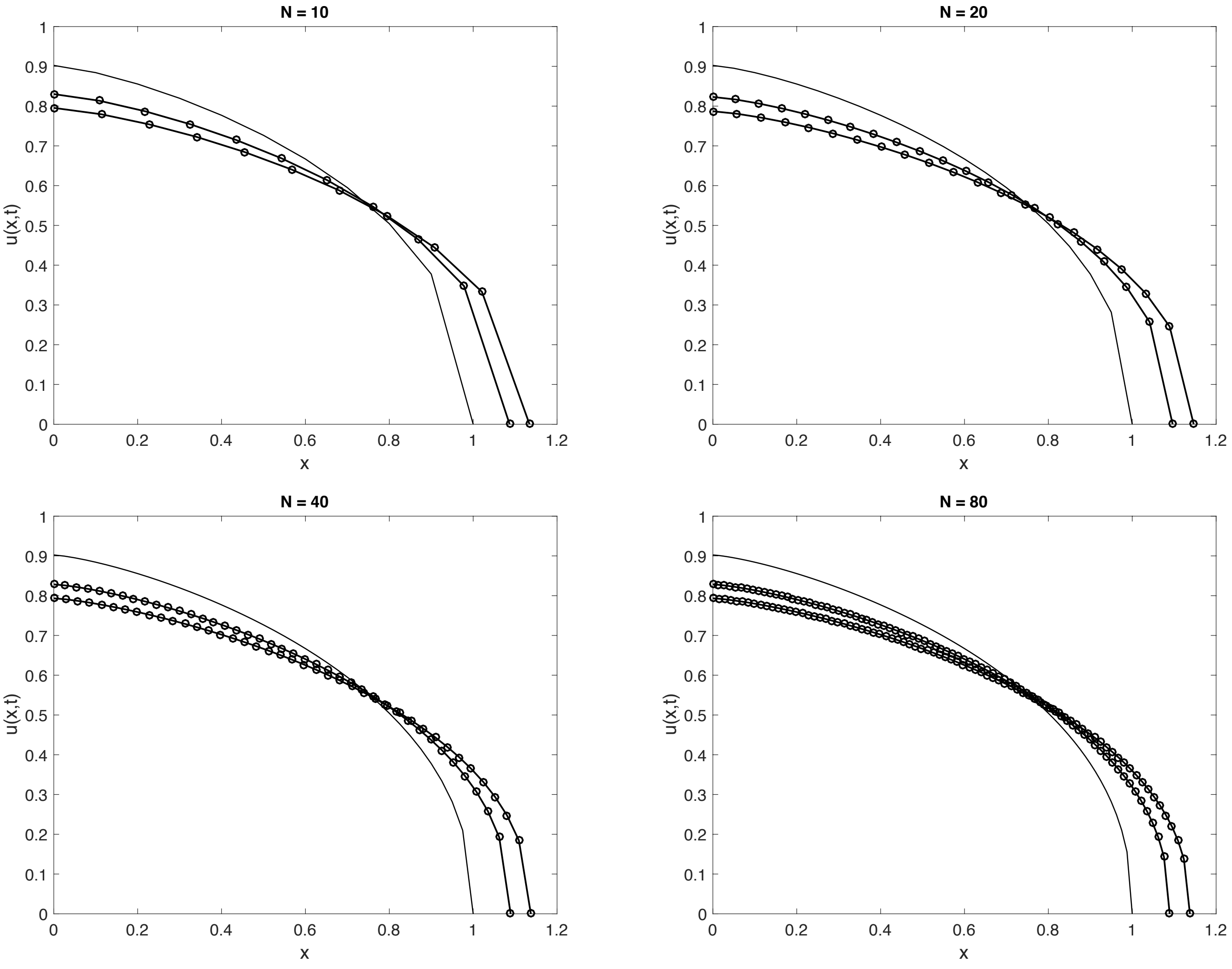

FIG. 4. *Numerical solution of the GE (using forward Euler time-stepping (3.5)), represented by 'o', converging to the exact solution, represented by '$-$', with increasing $N = 10, 20, 40, 80$ with self-similar initial data (4.1).*

We thus implement the numerical method outlined in 3.1.4, where we solve for $t \in [1, 5]$ and compute results for $N = 10 \cdot 2^k$, $k = 0, 1, 2, 3$ with the chosen $\Delta t = O(1/N^2)$. As before, the self-similar solution at $t = 1$, given in equation (4.1), is taken as initial data.

Convergence to the exact self-similar solution can be seen with the increase of N in Figure 4 where the numerical solution, represented by 'o', approximates the self-similar solution represented by '$-$'. An indication of how the mesh moves when applying the moving-mesh method, outlined in 3.1.4, can also be seen. We observe that mesh points remain evenly spread for all time t with no overtaking of nodes.

In Table 2 relative errors are computed and found to be very small within the interval $1 < t < 3$. Convergence of the results can be seen since the relative error decreases as the resolution increases, suggesting that as the number of mesh points N increases the approximations are converging to the exact solution.

The results suggest super-linear convergence of the numerical solution as N increases. In Table 2 we compute $p = \log((Re)_N/(Re)_{2N})$ for each $N = 10, 20, 40$ and 80, obtaining $p \approx 1.75$ on the finer grids.

N	Δt	Relative error $(Re)_N$	Order p
10	0.01	3.8×10^{-4}	–
20	0.0025	1.3×10^{-4}	1.64
40	0.000625	4.1×10^{-5}	1.71
80	0.00015625	1.2×10^{-4}	1.75
160	0.0000390625	3.6×10^{-6}	1.77

TABLE 2

Relative errors (forward Euler time-stepping (3.5)) and estimated order of accuracy p.

4.2. Comparison of Time-Stepping Results. In section 3.1.2 an alternative similarity-based time-stepping procedure was introduced. The prediction was made that since the time-stepping procedure is scaled in such a way that the LTE vanishes (and so the error arising from the time step is eliminated) the results may give better approximations. Results are shown in Table 3 where we see that in fact the two time-stepping procedures give very similar errors (dominated by the error in the computation of the velocity, equation (3.3)), the main difference being that the similarity-based time-stepping reaches a time $t = 3$ in about $1/20$ th of the time that standard Euler time-stepping takes.

N	Δt	Relative Error $(Re)_N$	
		Euler time-stepping	Similarity-based time-stepping
10	0.01	9.5×10^{-4}	1.9×10^{-4}
20	0.0025	4.5×10^{-4}	5.1×10^{-5}
40	0.000625	2.2×10^{-4}	1.3×10^{-5}
80	0.000015625	1.1×10^{-4}	4.5×10^{-6}

TABLE 3

Relative error results comparing forward Euler time-stepping (3.5) and the similarity-based time-stepping (3.8).

5. Waiting-times. A specific characteristic feature of the GE is that under certain conditions, the right-hand boundary of the glacier may remain fixed for a finite interval of time, say t_w. This period of time is called the *waiting-time period*, during which the initial density redistributes itself before any movement occurs at the boundary [21]. In this section we investigate waiting-times for the GE including a discussion on the profile of u, observations on the initial behaviour of the velocity boundary and discuss the link with similarity.

5.1. Profile of u. Consider the profile of the solution u to (2.1) when the boundary is fixed at $x = 1$. We assume that $u > 0$ for $x < 1$ and $u = 0$ at $x = 1$ for all t. It follows that for these conditions to hold the initial data $u(x, t)$ must contain some positive power of the factor $(1 - x(t))$ at all times t. We choose the power at time t to be the positive real number $\alpha(t)$, such that

$$u(x,t) = (1-x)^{\alpha(t)}\, g(x,t) \tag{5.1}$$

where $g(x, t)$ is a strictly positive finite function with a finite derivative at the boundary $x = 1$ [19]. Close to the boundary, to leading order we may take $g(x, t)$ to be a constant g, and so equation (5.1) becomes

$$u(x,t) = (1-x)^{\alpha(t)} g \tag{5.2}$$

to leading order.

Suppose $x(t)$ is a near-boundary location which moves according to the conservation of mass principle, Definition 1.3. Then, by the following equation to leading order:

$$\int_{x(t)}^{1} u(x,t)\,\mathrm{d}x = \int_{x(t)}^{1} (1-x)^{\alpha(t)} g\,\mathrm{d}x = \frac{(1-x(t))^{\alpha(t)+1}}{\alpha(t)+1} g = C, \tag{5.3}$$

independent of time. Since $(1 - x(t))$ is strictly decreasing for points sufficiently close to the boundary it follows that $\alpha(t)$ decreases as $x(t) \to 1$. Over time, as $\alpha(t)$ decreases, a steep front develops with an infinite slope as $\alpha(t)$ falls below 1. Then, at some point in time, the value of $\alpha(t)$ becomes critical, which is when the waiting-time ends and the boundary is forced to move.

5.2. Initial Behaviour of the Boundary Velocity. To analyse the conditions under which the waiting-time period occurs, we consider the behaviour of the velocity at the boundary. The initial data is given by equation (5.2).

To see the behaviour at the boundary for the GE we recall the diffusive velocity, equation (3.3) of section 3.1.1. From this equation the value of the velocity at the boundary $x_N = 1$ (assuming continuity of v at the boundary) is obtained by taking the limit as u tends to zero, or equivalently the limit $x \to 1$. Denoting the velocity at the boundary by $v_B = v(1,t)$, specifically for the GE we obtain

$$v_B = -\lim_{x\to 1}\left[\left(\frac{3}{7}(1-x)^{7\alpha(t)/3}\right)_x\right]^3.$$

By substituting the initial data, equation (5.2),

$$v_B = \lim_{x\to 1}(\alpha(t))^3(1-x)^{7\alpha(t)-3} \tag{5.4}$$

for small $(1-x)$. As x tends to 1 we obtain different possible behaviours for the velocity v_B. In fact there is a critical value α_c such that

i) if $\alpha(t) > \alpha_c$ then $v_B \to 0$ as $x \to 1$
ii) if $\alpha(t) = \alpha_c$ then v_B remains finite (and positive) as $x \to 1$.

We see from equation (5.4) that the critical value is $\alpha_c = 3/7$. In case i), no movement occurs at the boundary and so the boundary is waiting. As time increases the value of $\alpha(t)$ decreases to α_c in case *ii*), at which point there is a finite (non-zero) velocity v_B which induces movement of the boundary, ending the waiting-time. A numerical investigation into this behaviour is presented in section 5.4.

In the following section we focus on the time when $\alpha(t)$ reaches the critical value α_c; the point at which the boundary begins to move ending the waiting-time.

5.3. Profile of u when Waiting ends. When $\alpha(t) = \alpha_c = 3/7$, the solution $u(x,t)$ close to the boundary at this instant from (5.2) is

$$u(x,t) = (1-x)^{3/7}g. \tag{5.5}$$

We show that equation (5.5) coincides with that of the self-similar solution (2.3) close to the boundary.

Recall the similarity variables, $\eta = u/t^{\gamma}$ and $\xi = x/t^{\beta}$, from Definition 1.4 where $\eta = f(\xi)$. By substituting $u(x,t)$ into the GE (2.1) and transforming the variables, we have

$$\eta\gamma - \beta\xi\eta_\xi = (\eta^5(\eta_\xi)^3)_\xi$$

leading to

$$-\beta(\xi\eta)_\xi = (3/7)^3(\eta(\eta^{7/3})_3)_3.$$

Thus, since $\eta_\xi = 0$ when $\xi = 0$ (equivalent to $u_x = 0$ when $x = 0$) and $\eta \neq 0$,

$$-\beta\xi = (3/7)^3\,(\eta_\xi^{7/3})^3$$

and hence

$$-\beta(3/7)^3\int \xi^{1/3}\,\mathrm{d}\xi = \eta^{7/3} \quad\Longrightarrow\quad -\frac{3\beta(3/7)^3\xi^{4/3}}{4} + a = \eta^{7/3} \tag{5.6}$$

where a is a constant of integration. Since $\eta = 0$ at the boundary $\xi = 1$ (equivalent to the boundary condition $u = 0$ at $x = 1$), we obtain

$$a = \frac{3\beta\,(3/7)^3}{4}.$$

Substituting this into equation (5.6), after some rearrangement we have

$$\eta = (4/3)^{3/7}\left(1 - \xi^{4/3}\right)^{3/7}$$

leading to

$$u(x,t) \propto \left(1 - x^{4/3}\right)^{3/7}. \tag{5.7}$$

Hence, the critical shape of the solution (5.7) is proportional to that of the self-similar solution (2.3), close to the boundary at $t = 1$.

Also, since the velocity $v > 0$, due to equation (5.3), as time increases $x \to 1$ so that $\alpha(t)$ decreases, eventually reaching $a_c = 3/7$ at which point the boundary moves.

5.4. Numerical Results. In this section, we implement the moving-mesh method outlined in section 3.1 to investigate waiting-time behaviour of the GE equation (2.1). We choose the initial data to be

$$u(x,t^0) = (1-x)^{\alpha(t^0)} \tag{5.8}$$

which is chosen such that it coincides with (5.2) with $g = 1$ close to the boundary. Numerical solutions will be presented where the two properties of $\alpha(t)$ in section 5.2 will be verified and observations will be made on the effect of the $\alpha(t)$ value on the behaviour of the solution.

5.4.1. Waiting Results for the GE. When implementing the numerical method, we take initial time to be $t^0 = 1$ and $N = 80$. From (5.8), the appropriate initial data obtained in section 5.1 for this problem is given by

$$u(x,1) = (1-x)^{\alpha(1)}, \quad x \in [0,1]. \tag{5.9}$$

Recall that the critical value of $\alpha(t)$ for this problem is given by $\alpha_c = 3/7$. In section 5.1 we showed that under the moving-mesh strategy since u at the boundary $x_N = 1$ is fixed for all time, $\alpha(t)$ decreases. Thus, when carrying out the numerical method we should see, as time increases, the appearance of a front which moves when $\alpha(t) = 3/7$. By our findings in section 5.3, where we looked at the steep initial movement of the boundary, the profile of u should be proportional to the self-similar solution, close to the boundary.

Results are shown in Figure 5. In each case the initial data is represented by a dotted line showing the initial profile of u. Observe Figure 5 (top left), a close up of the moving boundary, where results are shown for when $\alpha(1) = 3/7$; the critical value. The program is run for $6,000$ steps and $\Delta t = 0.001$. As expected, since the velocity at the boundary is finite, no waiting-time occurs and the boundary moves instantly.

Results for the case $\alpha(1) = 1$ are shown in Figure 5 (bottom left) where the program is run for $27,495$ steps and $\Delta t = 0.00002$. Since $\alpha(1) > \alpha_c$ we expect a waiting period to occur until $\alpha(1)$ decreases to the critical value $3/7$, where the waiting-time ends and the boundary should begin to move. We see the expected waiting period occurring in which the boundary, $x = 1$, remains fixed. The dashed line is when $\alpha(t) = 3/7$ where we observe that the boundary is no longer $x = 1$ and has moved to the right. Over time the solution will continue to move as seen by the bold lines. By trial and error, we find that the point at which the boundary first moves is when the program is run for $18,500$ steps. When the program is run for $18,500$ steps results give a final time $t^f = 1.37$, resulting in a waiting period $t_w \in [1, 1.37]$.

In Figure 5 (top right) we have taken $\alpha(1) = 0.55$; a value very close to the critical value. We run the program for $27,495$ steps and choose $\Delta t = 0.00002$. By comparison with the results given in Figure 5 (bottom left) a very short waiting-time can be seen, which is due to α decreasing by a very small value $0.55 - 3/7 \approx 0.1214$. By trial and error,

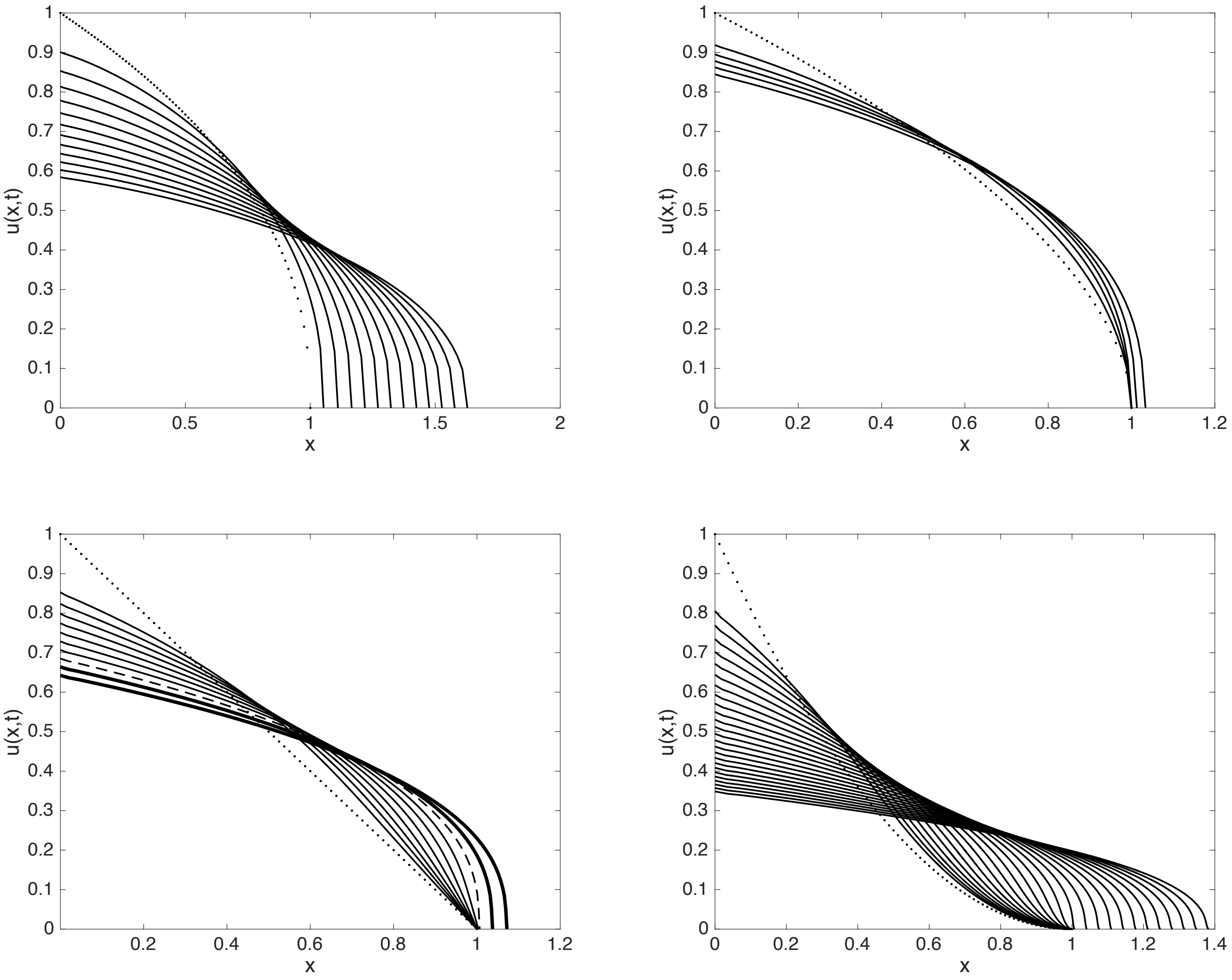

FIG. 5. *Numerical solution for various values of* $\alpha(1)$ *for the GE, see equation (5.9). Top left,* $\alpha(1) = 3/7$ *for* $t \in [1, 7]$*, top right* $\alpha(1) = 0.55$ *for* $t \in [1, 1.55]$*, bottom left* $\alpha(1) = 1$ *for* $t \in [1, 1.55]$ *and bottom right* $\alpha(1) = 2$ *for* $t \in [1, 3]$.

we find that the first instant at which the boundary does not move occurs when the program is run for $7,500$ steps with final time $t^f = 1.15$, resulting in a waiting period $t_w \in [1, 1.15]$.

In each case, in agreement with section 5.3, the profile of u shows behaviour proportional to that of the self-similar solution, becoming less steep as time t increases. In Figure 5 (bottom right), results for when $\alpha(1) = 2$ is run for $200,000$ steps and $\Delta t = 0.00001$. Again, supporting results in section 5.2, since $\alpha(1) > \alpha_c$ a waiting-time occurs and with a sufficient increase in steps we can see that the critical shape of the solution coincides with that of the self-similar solution close to the boundary.

6. Conclusion. In this paper a velocity-based moving-mesh method based on local conservation is presented that preserves both the scale invariance and symmetry properties of a simplistic nonlinear glacier equation [9] in one dimension with a moving boundary. Our main interest has been to demonstrate that these properties are reflected in the method. This has been achieved by generating nodal velocities based on a local conservation principle using a scale-invariant monitor function, and by implementing a similarity-based time-stepping scheme for the nodes for which the local truncation error vanishes in the case of similarity. The results obtained when implementing the method show strong super-linear convergence when validated against known self-similar solutions.

Furthermore we have investigated a specific aspect of the glacier equation called the waiting-time period where the conditions required for the right-hand boundary to move were analysed. For an initial condition of the form (5.2) with $\alpha(t) > 3/7$ the waiting-time was simulated using the moving-mesh method and in Figure 5 the waiting behaviour can be seen as the power of $\alpha(t)$ decreases towards $3/7$. The waiting-time was determined numerically for each discrete $\alpha(t)$, but an analytic relation between the waiting-time period and $\alpha(t)$ was not found, which is an avenue for future work. The main result observed was that when the waiting-time ends the critical profile of the solution close to the boundary is proportional to that of the similarity solution. This was shown analytically and supported numerically for various $\alpha(t)$'s.

It should be noted that considerable further research is yet to be done that analyses more realistic representations of a glacier. The model in this paper was a particularly simple one-dimensional flow-line model (albeit with an awkward non-linearity) and a natural continuation would be to extend the analysis and computation to multi-dimensions. Although the moving-mesh method has been implemented in two dimensions (using linear finite elements) on a very simple glacier test problem in [23], it has been tested on several other nonlinear diffusion problems including the second-order porous medium equation and the fourth-order thin film equation [1, 2, 7]. As in this paper, key properties of these problems, including mass conservation, scale invariance and self-similarity, are preserved in the numerical method when they are present [7]. Further work extends the simple glacier model to include accumulation of snow and ablation. This involves a source term, say $s(x) = s_a(x) + s_b(x)$ in the continuity equation (2.2) where $s_a(x)$ represents the accumulation rate of snow and $s_b(x)$ is the basal melting rate, see [8, 23]. A fundamental issue that still remains is that the moving-mesh method is yet to be implemented for realistic two-dimensional glacier models, essential in real applications. This does not exist in the current literature and should be an important topic of future research.

Acknowledgement
I should like to thank Professor Mike Baines of the University of Reading, UK, for his help and advice.

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, 2005, pp. 450-469.

[2] M.J. Baines, M.E. Hubbard. and P.K. Jimack, *Velocity-Based Moving Mesh Methods for Nonlinear Partial Differential Equations*, Commun. Comput. Phys., 10 (2011), pp. 509-576.

[3] 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 (2006), pp. 230-252.

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

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

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

[7] N. Bird, *A Moving-Mesh Method for High Order Nonlinear Diffusion*, PhD Thesis: University of Reading, UK, 2014.

[8] B. Bonan, M.J. Baines, N.K. Nichols, D. Partridge, *A moving point approach to model shallow ice sheets: a study case with radially-symmetrical ice sheets, The Crysophere*, doi: 10.5194/tc-10-1-2016, 10 (2016), pp. 1-14.

[9] E. Bueler, C. S. Lingle, J. A. Kallen-Brown, D. N. Covey, and L. N. Bowman, *Exact solutions and numerical verification for isothermal ice sheets*, J. Glaciol., 51(173) (2005), pp. 291-306.

[10] G.W Blueman and S.C. Anco, *Symmetry and Integration Methods for Differential Equations*, Springer-Verlag, New York, 2002.

[11] C.J Budd and G.J. Collins, *Symmetry-based Numerical Methods for Partial Differential Equations*, in Numerical Analysis 1997, D.F. Griffiths, D Higham and G.A Watson, eds., Addison Wesley Longman Limited,

1998, pp. 16-36.

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

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

[14] N. Calvo, J. Durany, and C. Vazquez, *Numerical computation of ice sheet profiles with free boundary models*, Appl. Numer. Math., 35 (2000), pp. 111-128.

[15] N. Calvo, J. D az, J. Durany, E. Schiavi, and C. Vazquez, *On a doubly nonlinear parabolic obstacle problem modelling ice sheet dynamics*, SIAM J. Appl. Math., 63(2) (2002), pp. 683-707.

[16] J.I. Díaz and R.J. Wiltshire, *Similarity solutions of an equation describing ice sheet dynamics*, Physica D: Nonlinear Phenomena, 216(2) (2006), pp. 319-326.

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

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

[19] R.A. Klassen, *Glacier History and Ice Flow Dynamics to Drift Prospecting and Geochemical Exploration*, in Proceedings of Exploration 97: Fourth Decennial International Conference on Mineral Exploration, 1997, pp. 221-231.

[20] B.J. Knerr, *The Porous Medium Equation in One Dimension*, Trans. Amer. Math. Soc., 234 (1977), pp. 381-415.

[21] A.A. Lacey, J.R. Ockendon and A.B. Taylor, *Waiting-time Solutions of a Nonlinear Diffusion Equation*, SIAM J. Appl. Math., 42 (1982), pp. 1252-1264.

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

[23] D. Partridge, *Numerical Modelling of Glaciers: Moving Meshes and Data Assimilation*, PhD Thesis: University of Reading, UK, 2013.

[24] W.S.B. Paterson, *The Physics of Glaciers*, 3rd edition, Butterworth Heinemann, 1994.

[25] S.F. Price, A.J. Payne, I.M. Howat and B.E. Smith, Committed sea-level rise for the next century from Greenland ice sheet dynamics during the past decade, Proc. Natl. Acad. Sci. USA, 108(22) (2011), pp. 8978-8983.

[26] The Intergovernmental Panel on Climate Change (IPCC): Working Group 1, *Climate Change 2001: The Scientific Basis*, Cambridge Univ. Press, New York, 2001.

[27] E. Suli and D. Mayers, *An An Introduction to Numerical Analysis*, Cambridge Univ. Press, 2003.

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