



Deposited via The University of Leeds.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/1784/>

Article:

Baines, M.J., Hubbard, M.E., Jimack, P.K. et al. (2006) Scale-invariant moving finite elements for nonlinear partial differential equations in two dimensions. *Applied Numerical Mathematics*, 56 (2). pp. 230-252. ISSN: 0168-9274

<https://doi.org/10.1016/j.apnum.2005.04.002>

Reuse

See Attached

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



White Rose
university consortium
Universities of Leeds, Sheffield & York

White Rose Consortium ePrints Repository

<http://eprints.whiterose.ac.uk/>

This is an author produced version of a paper published in **Applied Numerical Mathematics**.

White Rose Repository URL for this paper:

<http://eprints.whiterose.ac.uk/1784/>

Published paper

Baines, M.J., Hubbard, M.E., Jimack, P.K. and Jones, A.C. (2006) *Scale-invariant moving finite elements for nonlinear partial differential equations in two dimensions*. Applied Numerical Mathematics, 56 (2). pp. 230-252.

Scale-Invariant Moving Finite Elements for Nonlinear Partial Differential Equations in Two Dimensions

M.J. Baines^a M.E. Hubbard^b P.K. Jimack^b A.C. Jones^b

^a*Department of Mathematics, The University of Reading, UK*

^b*School of Computing, University of Leeds, UK*

Abstract

A scale-invariant moving finite element method is proposed for the adaptive solution of nonlinear partial differential equations. The mesh movement is based on a finite element discretisation of a scale-invariant conservation principle incorporating a monitor function, while the time discretisation of the resulting system of ordinary differential equations is carried out using a scale-invariant time-stepping which yields uniform local accuracy in time. The accuracy and reliability of the algorithm are successfully tested against exact self-similar solutions where available, and otherwise against a state-of-the-art h -refinement scheme for solutions of a two-dimensional porous medium equation problem with a moving boundary. The monitor functions used are the dependent variable and a monitor related to the surface area of the solution manifold.

Key words: scale invariance, moving meshes, finite element method, porous medium equation, moving boundaries

1 Introduction

In this paper a moving mesh finite element method is presented for the solution of scale-invariant nonlinear partial differential equations (PDEs) based on conservation of a monitor integral [10] and incorporating scale invariance. A feature of scale invariance is that space and time and the solution are coupled together [9]. If a numerical method is to reflect the scaling properties of the equation then the space discretisation should depend on both the solution and time in a scale-invariant way. The resulting semi-discrete system of ordinary differential equations (ODEs) can then inherit the scale invariance of the original PDE. Similarly the time discretisation of the ODE system should

reflect these scaling properties: scale invariance can then be used to define a time-step strategy which yields uniform local accuracy in time, as in [8].

A natural framework for the space discretisation is a moving mesh (see, for example, [2,5,10,11,15]). In this work the mesh will be coupled with the solution and moved at each time-step in such a way as to seek to conserve in time the scale-invariant integral of a monitor function, as distributed within each local patch of finite elements. It will be shown that this strategy can produce accurate results, even when the integral over the whole domain is not conserved exactly in time, provided that the dependent variable is recovered in an appropriate manner. The conservation principle can be interpreted as a fluid property (cf. the GCL method of [10]) and used in conjunction with the PDE to generate an equation for a velocity field which is a function of the space coordinate and time, as in [7]. The velocity field may then be integrated to obtain new mesh positions. The solution of the PDE can be reconstructed from the new mesh either directly from the conservation principle itself or from moving forms of the PDE. A similar approach has already been applied in [3] to a range of moving boundary problems in one and two space dimensions using the dependent variable as monitor function. This paper extends those results to include scale-invariant monitor functions, scale-invariant time-stepping, a surface area type monitor in 2D and more robust forms of solution recovery. The accuracy of the technique is assessed by comparison with general solutions of the porous medium equation obtained through an alternative numerical method, in addition to the similarity solutions used in [3].

The plan of the paper is as follows. In Section 2 the steps of the new method are summarised, prior to their detailed description in later sections. In Section 3 we discuss the principles of scale invariance as they apply to nonlinear PDEs. Next we recall in Section 4 how a given function can be well represented on an irregular mesh using an equidistribution principle. We then show how, as time evolves, the same principle, extended with the help of scale invariance, becomes a conservation principle governing the evolution of the mesh in time (cf. [11]). Weak forms of these principles are introduced with a view to constructing finite element approximations. In Section 5, in order to generate an associated velocity field, the conservation principle is identified with a Lagrangian statement of conservation of mass for a fluid problem (cf. [10]). By transforming to an Eulerian frame via the Reynolds Transport Theorem and using the PDE we can obtain an equation for the corresponding Eulerian velocity field. Uniqueness is ensured by specifying the vorticity of the field, as in [10], thus introducing a velocity potential which satisfies an elliptic equation. The velocity is integrated in time to give the new position of the moving coordinates. Weak forms of the derivation are also given in preparation for the application of finite elements.

In Section 6 we introduce finite-dimensional forms of the velocity and the

velocity potential equations using standard linear finite elements. These linear elements are moved with a piecewise linear velocity field (which is formally a recovered gradient of a piecewise linear velocity potential) and carry with them a piecewise linear finite element solution. Section 7 is concerned with time-stepping. The finite element discretisation in space generates an ODE system for the nodal positions which is integrated in time using a scaled time variable which provides a scale-invariant local truncation error and hence, as in [8], a relative local truncation error which is uniformly accurate in time.

In Section 8 we discuss how the solution can be recovered from the nodal positions in three possible ways, (i) by inverting the conservation principle directly, (ii) by using an ALE (Arbitrary Lagrangian Eulerian) approach to the differential form of the PDE, or (iii) by using an ALE integral form of the PDE. Monitor functions are considered in Section 9. They are chosen here to be either the dependent variable itself or a surface area type monitor, but the approach is readily generalised to other monitors. In Section 10 the accuracy and reliability of the algorithm is tested for the two-dimensional porous medium equation (PME), both against a range of known analytic similarity solutions and against results from a state-of-the-art h -refinement code. Each of the monitor functions is tested, along with the three different approaches to recovering the nodal solution values at each time-step, for a selection of PME problems. The paper concludes with a short discussion.

2 Overview of the proposed algorithm

In this section we rehearse the main steps of the new method before expanding on them in more detail. The first and pivotal step is the definition of a distributed conservation principle in a moving frame which depends on a time-dependent monitor function. Crucially this monitor function is constructed so that its integral is scale-invariant. The next step is to derive an equation for the velocity associated with this moving frame, carried out by differentiating the distributed conservation principle with respect to time, using the Reynolds Transport Theorem and incorporating the original PDE. This equation, which is actually a weak form of the PDE, is also scale invariant, as are all the equations constructed here. A velocity potential is then introduced, based on the assumption of an irrotational velocity field, which satisfies an associated weak elliptic equation. Finally the velocity is recovered from the potential, using a weak form of their relationship, and integrated in time.

Each of these steps is implemented numerically, using linear finite elements to approximate the solution, the velocity, and the velocity potential. The resulting nodal velocities are integrated forward in time using a special scale-invariant time-stepping method to obtain new positions of the mesh. Finally,

the finite element solution is completed by reconstructing the solution from the new mesh or its nodal velocities.

3 Scale invariance

Scale invariance is a natural property of models of physical systems due to their independence of physical units [4]. For a scale-invariant problem governed by the PDE

$$u_t = Lu \tag{1}$$

(where Lu is a purely spatial operator), together with properties conferred by the boundary conditions, there exist indices β and γ such that the scalings

$$t = \lambda \hat{t}, \quad \mathbf{x} = \lambda^\beta \hat{\mathbf{x}}, \quad u = \lambda^\gamma \hat{u}, \tag{2}$$

leave the problem invariant (λ is simply an arbitrary scaling factor). For example, in the case of the porous medium equation in d dimensions,

$$u_t = \nabla \cdot (u^n \nabla u) \quad \text{in } \Omega, \tag{3}$$

subject to $u|_{\partial\Omega} = 0$ (which enforces conservation of mass) it can be shown [4] that $\beta = 1/(nd + 2)$ and $\gamma = -d/(nd + 2)$. For this problem there exist known self-similar solutions with compact support and a moving boundary, see for example [16], which represent intermediate asymptotic solutions in the sense of [4].

4 Equidistribution, monitor functions and conservation

Given an initial condition $u = u_0$ for equation (1) and a non-negative, solution-dependent monitor function $m(u)$, the basic form of an equidistribution principle is

$$m_{ave}(u_0) \Delta\Omega = c_0, \tag{4}$$

where $\Delta\Omega$ is the size (area in 2D) of a subregion of the total domain Ω , m_{ave} is an average of m over this subregion, and c_0 is a constant, determined by the problem. The principle (4) associates small subregions with large values of m and is often used to position nodes to give high mesh resolution where m is

large. (For exposition purposes the monitor function m is taken here to be a function of u only, although the analysis in this paper is readily extended to monitors which are functions of derivatives of u : indeed, one of the monitor functions that we shall use depends upon the gradient of u .)

In the present work we are interested in maintaining such a distribution in time. It will be assumed here that the initial mesh already has a desired distribution of the monitor, *i.e.*

$$m_{ave}(u_0) \Delta\Omega = c_{\Delta\Omega}, \quad (5)$$

where $c_{\Delta\Omega}$ is a constant associated with the subregion $\Delta\Omega$ and determined by the initial conditions.

It is not generally possible to maintain the same distribution principle on a moving mesh as time progresses. A similar distribution may still exist at later times, but constants different to $c_{\Delta\Omega}$ will arise unless $m_{ave}(u) \Delta\Omega(t)$ remains constant in time. However, if the PDE problem is scale-invariant, with scaling exponents β and γ as in equation (2), we may define a time-dependent scaled monitor function $\tilde{m}(t, u)$ by

$$\tilde{m}(t, u) = t^{-d\beta} m(t^{-\gamma}u). \quad (6)$$

Since $m(t^{-\gamma}u)$ is scale invariant and Ω scales like the d 'th power of a length, $\tilde{m}_{ave}(t, u) \Delta\Omega(t)$ is scale invariant. Moreover, it can be verified that it is independent of time when u is a self-similar solution with functional form

$$\tilde{u} = t^\gamma f(t/x^\beta). \quad (7)$$

With this motivation we define a conservation principle in time based on (5) by

$$\tilde{m}_{ave}(t, u(t, \mathbf{x})) \Delta\Omega(t) = c_{\Delta\Omega} \quad (8)$$

and use this to move the reference frame. We shall refer to equation (8) as the scale-invariant conservation principle, and by differentiating a weak form of this principle with respect to t we will obtain a scale-invariant equation for the corresponding velocity field, using a weak form of the PDE. As already noted, (8) is satisfied by a self-similar solution of equation (1) of the form (7). However it may also be used as the basis for the mesh movement even when u is non-self-similar.

To obtain a weak form of (8), given a suitable test function w (with support $\Delta\Omega$), we first define a weak form of the basic equidistribution principle (4) for

the initial condition u_0 as

$$\int_{\Omega} w(\mathbf{x}) m(u_0(\mathbf{x})) d\Omega = c_0(w), \quad (9)$$

where $c_0(w)$ is a new constant. Again, introducing the scaled monitor function \tilde{m} given by (6), the equation corresponding to the conservation principle (8) is

$$\int_{\Omega(t)} w(t, \mathbf{x}) \tilde{m}(t, u(t, \mathbf{x})) d\Omega = \tilde{c}_0(w), \quad (10)$$

where $\tilde{c}_0(w)$ is a constant for the test function $w(t, \mathbf{x})$. It is assumed that the test function remains invariant under the scaling (2) and has support $\Delta\Omega(t)$ which evolves with the domain $\Omega(t)$. We shall refer to equation (10) as the weak form of the scale-invariant conservation principle and show its equivalence to an equation for a velocity field $\dot{\mathbf{x}}(t, \mathbf{x})$ which is scale-invariant for all $t \geq t_0$ in the resulting moving frame, where t_0 is the initial time.

5 Deriving the velocity field from the conservation principle

The weak scale-invariant conservation principle (10) can be regarded as the Lagrangian conservation law

$$\frac{d}{dt} \int_{\Omega(t)} w(t) \tilde{m}(t, u) d\Omega = 0. \quad (11)$$

for a fluid of density \tilde{m} . In order to extract the velocity field $\dot{\mathbf{x}}(t, \mathbf{x})$ from (11) we transform to an Eulerian frame, using the Reynolds Transport Theorem [18] in the form

$$\frac{d}{dt} \int_{\Omega(t)} w \tilde{m} d\Omega = \int_{\Omega(t)} \frac{\partial(w \tilde{m})}{\partial t} d\Omega + \int_{\Omega(t)} \nabla \cdot (w \tilde{m} \dot{\mathbf{x}}) d\Omega. \quad (12)$$

Note that since w evolves with the domain $\Omega(t)$ it is advected with velocity $\dot{\mathbf{x}}$ and satisfies the equation

$$\frac{\partial w}{\partial t} + \dot{\mathbf{x}} \cdot \nabla w = 0, \quad (13)$$

hence (12) may be expressed as

$$\int_{\Omega(t)} w(t) \left(\frac{\partial \tilde{m}(t, u)}{\partial t} + \nabla \cdot (\tilde{m}(t, u) \dot{\mathbf{x}}) \right) d\Omega = 0. \quad (14)$$

From (6)

$$\begin{aligned} \frac{\partial \tilde{m}}{\partial t} &= \frac{\partial(t^{-d\beta} m(t^{-\gamma} u))}{\partial t} \\ &= -d\beta t^{-d\beta-1} m(t^{-\gamma} u) + t^{-d\beta} \left(-\gamma t^{-\gamma-1} u + t^{-\gamma} \frac{\partial u}{\partial t} \right) m'(t^{-\gamma} u) \end{aligned} \quad (15)$$

in which the partial derivative indicates differentiation with respect to t , but not \mathbf{x} . We may substitute for $\partial u / \partial t$ from the PDE (1) into (15) and thence into (14) to obtain an equation for $\dot{\mathbf{x}}$. Although equation (14) does not determine $\dot{\mathbf{x}}$ uniquely, in the following we employ additional constraints (as in [10]) which do give uniqueness.

By itself, equation (14) is insufficient to determine $\dot{\mathbf{x}}$ uniquely (at least in more than one space dimension). However, as in [10], by the Helmholtz Decomposition Theorem, uniqueness may be obtained by additionally specifying the *curl* of $\dot{\mathbf{x}}$ and a suitable boundary condition. By writing $\mathit{curl} \dot{\mathbf{x}} = \mathit{curl} \mathbf{v}$, where \mathbf{v} is prescribed, it follows that there exists a potential function ϕ such that

$$\dot{\mathbf{x}} = \mathbf{v} + \nabla \phi \quad (16)$$

(although, since we shall not have occasion to use a non-zero \mathbf{v} in what follows it is set to zero, implying an irrotational velocity field $\dot{\mathbf{x}}$ (cf. [3])).

On substituting for $\dot{\mathbf{x}}$ from (16) (with $\mathbf{v} = \mathbf{0}$) equation (14) may be written as the weak form of an elliptic equation for ϕ , namely,

$$- \int_{\Omega(t)} w \nabla \cdot (\tilde{m} \nabla \phi) d\Omega = \int_{\Omega(t)} w \frac{\partial \tilde{m}}{\partial t} d\Omega \quad (17)$$

where $\partial \tilde{m} / \partial t$ is given by (15) and, for the sake of clarity, the dependence of w on t has been dropped. The condition $\phi = 0$ is applied on the boundary to ensure that (17) has a unique solution.

A convenient weak form of (16) (with $\mathbf{v} = \mathbf{0}$) is

$$\int_{\Omega(t)} w (\dot{\mathbf{x}} - \nabla \phi) d\Omega = \mathbf{0}. \quad (18)$$

Equations (10) and (14) are invariant under the scalings (2) by design. Since, in addition to these scalings, ϕ scales as $\phi = \lambda^{2\beta-1}\hat{\phi}$ (in line with equation (16)) with $\mathbf{v} = \mathbf{0}$, then equations (17) and (18) are also scale-invariant.

6 Application of finite elements

Following [3], let $\mathbf{X} \approx \mathbf{x}$, $\dot{\mathbf{X}} \approx \dot{\mathbf{x}}$, $\Phi \approx \phi$, and $U \approx u$ be local piecewise linear finite element functions, and $W_i \approx w_i$ be the usual piecewise linear basis functions moving with the velocity field $\dot{\mathbf{X}}$ (so that their support is restricted to the (moving) patch of elements surrounding node i). These basis functions, being scaled to unity, are already scale-invariant. Then we can use (10), (17) and (18) to define a moving finite element method.

The weak scale-invariant conservation principle (10) becomes

$$\int_{\Omega(t)} W_i \tilde{m}(t, U) d\Omega = \tilde{C}_i, \quad (19)$$

say (for all nodes i in the interior of the mesh). The \tilde{C}_i are assumed to be constants in time for the purposes of deriving suitable mesh velocities. Similarly, the weak form of the potential equation (17) becomes, on integration by parts,

$$\int_{\Omega(t)} \tilde{m} \nabla W_i \cdot \nabla \Phi d\Omega = \int_{\Omega(t)} W_i \frac{\partial \tilde{m}}{\partial t} d\Omega, \quad (20)$$

for each interior node i , with $\partial \tilde{m} / \partial t$ given by (15) with u replaced by U . We use weak forms of the PDE (1) to evaluate the right-hand-side of (20) in order to obtain a finite element form of the equation (17) for the velocity potential Φ . Note that the condition $\Phi = 0$ is applied on the boundary to ensure that (20) has a unique solution: this corresponds to a zero mesh velocity tangential to the boundary in the discretisation.

The velocity equation (18) becomes

$$\int_{\Omega(t)} W_i (\dot{\mathbf{X}} - \nabla \Phi) d\Omega = \mathbf{0}, \quad (21)$$

for all nodes i (including those on the boundary), corresponding to the best approximation $\dot{\mathbf{X}}$ to $\nabla \Phi$ in the space spanned by the W_i .

From the design of \tilde{m} , equation (19) is invariant under the scalings

$$t = \lambda \hat{t}, \quad \mathbf{X} = \lambda^\beta \hat{\mathbf{X}}, \quad U = \lambda^\gamma \hat{U} \quad (22)$$

(the W_i are scale-invariant functions), and thus so are equations (20) and (21).

Using the finite element expansions

$$\mathbf{X} = \sum_j \mathbf{X}_j W_j, \quad \Phi = \sum_j \Phi_j W_j, \quad U = \sum_j U_j W_j, \quad (23)$$

the matrix forms of equations (20) and (21) are, respectively,

$$K(\tilde{m}) \underline{\Phi} = \underline{G} \quad (24)$$

with $\Phi = 0$ on the boundary, and

$$A \underline{\dot{\mathbf{X}}} = \mathbf{B} \underline{\Phi} \quad (25)$$

Specifically,

$$\begin{aligned} K(\tilde{m}) &= \{K_{ij}(\tilde{m})\}, & \underline{\Phi} &= \{\Phi_i\}, & \underline{G} &= \{G_i\}, \\ A &= \{A_{ij}\}, & \underline{\dot{\mathbf{X}}} &= \{\dot{\mathbf{X}}_i\}, & \mathbf{B} &= \{\mathbf{B}_{ij}\} \end{aligned} \quad (26)$$

and

$$\begin{aligned} K_{ij}(\tilde{m}) &= \int_{\Omega(t)} \tilde{m} \nabla W_i \cdot \nabla W_j \, d\Omega, \\ A_{ij} &= \int_{\Omega(t)} W_i W_j \, d\Omega, \\ \mathbf{B}_{ij} &= \int_{\Omega(t)} W_i \nabla W_j \, d\Omega, \\ G_i &= \int_{\Omega(t)} W_i \frac{\partial \tilde{m}}{\partial t} \, d\Omega. \end{aligned} \quad (27)$$

The matrices $K(\tilde{m})$ and A are symmetric weighted stiffness and mass matrices, respectively.

Equations (24) and (25), being special forms of (20) and (21), respectively, are scale-invariant under the scalings (22). Since both the exact solution and the

finite element solution at the nodes scale in the same way the semi-discrete error is scale-invariant and does not scale up with λ . Equations (24) and (25) form the basis of an algorithm for a moving mesh finite element method whereby, given U on a mesh determined by the vector $\underline{\mathbf{X}}$ of nodal positions, a vector of nodal mesh velocities $\dot{\underline{\mathbf{X}}}$ can be derived from (25), once Φ has been found from (24). The scale-invariant ordinary differential equation system for $\underline{\mathbf{X}}$ (25) may be written as

$$\dot{\underline{\mathbf{X}}} = \underline{\mathbf{F}}(\underline{\mathbf{X}}) \quad (28)$$

where $\underline{\mathbf{F}}$ is a known function of $\underline{\mathbf{X}}$, and can be integrated to update the nodal mesh positions via a scale-invariant time-stepping scheme, which we now discuss.

7 Scale-invariant time-stepping

In the Method of Lines approach employed here the ODE system (28) is, as usual, time-stepped by a finite difference method. In what follows we consider only the forward Euler discretisation of (28) but the argument extends to linear multistep and Runge-Kutta methods generally (cf. [8]).

Equation (28) is scale-invariant in the sense that it is invariant under the mapping (22). Specifically, the power of λ which occurs on the left-hand side of (28) also occurs on the right-hand side and the two cancel out. This is also true of the forward Euler discretisation of (28),

$$\frac{\underline{\mathbf{X}}_{N+1} - \underline{\mathbf{X}}_N}{t_{N+1} - t_N} = \underline{\mathbf{F}}(\underline{\mathbf{X}}_N), \quad (29)$$

for the same scalings as in (22). However, the local truncation error (LTE) of (29),

$$LTE = \frac{\underline{\mathbf{X}}(t_{N+1}) - \underline{\mathbf{X}}(t_N)}{t_{N+1} - t_N} - \underline{\mathbf{F}}(\underline{\mathbf{X}}(t_N)) = \frac{1}{2}(t_{N+1} - t_N) \left. \frac{d^2 \underline{\mathbf{X}}}{dt^2} \right|_{t=\theta_N}, \quad (30)$$

where $t_N < \theta_N < t_{N+1}$, is not scale-invariant under the scalings (22) in general because it contains a power of λ as a factor.

To obtain a time-stepping method with a scale-invariant LTE we introduce the time-like variable

$$\sigma = t^\beta, \quad (31)$$

under which the ODE (28) transforms to

$$\frac{d\mathbf{X}}{d\sigma} = \beta^{-1}\sigma^{\beta-1-1}\mathbf{F}(\mathbf{X}) = \mathbf{G}(\sigma, \mathbf{X}), \quad (32)$$

say. Applying the forward Euler discretisation to equation (32) gives

$$\frac{\mathbf{X}_{N+1} - \mathbf{X}_N}{\sigma_{N+1} - \sigma_N} = \mathbf{G}(\sigma_N, \mathbf{X}_N) \quad (33)$$

with LTE

$$\frac{\mathbf{X}(\sigma_{N+1}) - \mathbf{X}(\sigma_N)}{\sigma_{N+1} - \sigma_N} - \mathbf{G}(\sigma_N, \mathbf{X}(\sigma_N)) = \frac{1}{2}(\sigma_{N+1} - \sigma_N) \left. \frac{d^2\mathbf{X}(\sigma)}{d\sigma^2} \right|_{\sigma=\theta'_N}, \quad (34)$$

where $\sigma_N < \theta'_N < \sigma_{N+1}$. Since \mathbf{X} and σ scale in the same way (see (22) and (31)) it follows that the LTE (34) is scale-invariant. Hence, provided that $\Delta\sigma = \sigma_{N+1} - \sigma_N$ is constant (unscaled) this LTE does not scale up with λ and remains uniform with respect to σ (cf. [8] where, following a more general rescaling, a similar result is obtained for the relative LTE, and this is extended to prove existence of a discrete self-similar solution to the discretised ODE system which is uniformly approximated for all time). This may lead to attractive results in the future for this work.

The scaled scheme, from (32) and (33), is

$$\mathbf{X}_{N+1} = \mathbf{X}_N + \beta^{-1} \left((t_{N+1})^\beta - (t_N)^\beta \right) t_N^{1-\beta} \mathbf{F}(\mathbf{X}_N) \quad (35)$$

where

$$t_{N+1} - t_N = \Delta t \approx \beta^{-1}\sigma^{\beta-1-1} \Delta\sigma \quad (36)$$

and $\Delta\sigma = \sigma_{N+1} - \sigma_N$ is constant. A similar argument may be applied to any linear multistep or Runge-Kutta scheme (cf. [8]).

8 Recovering the nodal values of the solution

Once \mathbf{X} has been updated it is necessary to recover the solution U on the resulting mesh. The most obvious way to do this is to invert the conservation principle (19) for U . However, the inversion will require the solution of a nonlinear system of equations in general and the resulting approach is therefore

not always robust. Furthermore this approach is not always appropriate since the true solution of (1) need not always satisfy (19) exactly.

Alternatively, once $\dot{\mathbf{X}}$ is known, U may be updated separately by time-stepping the weak form of the PDE (1) in arbitrary Lagrangian-Eulerian (ALE) form. At least two such forms are possible. First the weak differential form of (1),

$$\int_{\Omega(t)} W_i \dot{U} d\Omega = \int_{\Omega(t)} W_i (\nabla U \cdot \dot{\mathbf{X}} + LU) d\Omega \quad (37)$$

(for all interior nodes i), with \dot{U} given on the outer boundary, can be used to obtain \dot{U} everywhere. This can then be integrated in time using the time-stepping strategy derived in Section 7. In the finite element implementation this leads to the matrix equation

$$A \underline{\dot{U}} = \underline{\Psi} \quad (38)$$

where, for the PME (3), the components of $\underline{\Psi}$ are given by

$$\Psi_i = \int_{\Omega(t)} (W_i \nabla U \cdot \dot{\mathbf{X}} - U^n \nabla W_i \cdot \nabla U) d\Omega \quad (39)$$

after integration by parts and the imposition of the boundary condition $U|_{\partial\Omega} = 0$. Imposition of $\dot{U} = 0$ on the boundary maintains this condition exactly, but a major drawback of this approach is that it is not possible to guarantee conservation of mass by this route.

An alternative is to consider the weak integral form of (1),

$$\frac{d}{dt} \int_{\Omega(t)} W_i U d\Omega = \int_{\Omega(t)} W_i (LU + \nabla \cdot (U \dot{\mathbf{X}})) d\Omega \quad (40)$$

for all nodes i (including those on the boundary), cf. (11), (14) with $m = u$. For the PME, and after integration by parts, (40) yields

$$\dot{\Theta}_i = \int_{\Omega(t)} (W_i \nabla U \cdot \dot{\mathbf{X}} - U^n \nabla W_i \cdot \nabla U) d\Omega \quad (41)$$

in which

$$\Theta_i(t) = \int_{\Omega(t)} W_i U d\Omega. \quad (42)$$

Equation (41) can be integrated to obtain new values for $\underline{\Theta} = \{\Theta_i\}$. The solution is then recovered from (42) which, using the expansions in (23), yields the matrix form

$$A \underline{U} = \underline{\Theta} \quad (43)$$

where A is the same mass matrix as in (25). We refer to (43) as conservative ALE recovery. Note that summing (40) over i gives zero, so exact recovery of U from (42) will guarantee conservation of mass. This acknowledges that the velocity field may induce a redistribution of mass between the nodes but that, overall, mass is conserved. The boundary conditions on U are only imposed weakly in this description, but it is also possible to apply them strongly and still maintain exact conservation (although we do not discuss this point further here).

9 Monitor functions

The use of the dependent variable u as the monitor function has been introduced in [3], where the corresponding moving mesh is shown to behave well for a variety of test problems when using a constant time step. In this paper, as well as using the scale-invariant time-stepping introduced above, we also consider a “surface area” monitor given by

$$m(\nabla u) = \sqrt{1 + (\nabla u)^2}. \quad (44)$$

This m is a function of ∇u and the previous theory has only been given for functions of u . However, the modifications are straightforward. Equation (6) becomes

$$\tilde{m}(t, \nabla u) = t^{-d\beta} m(t^{-\gamma+\beta} \nabla u) \quad (45)$$

with the consequent alterations to (15). The modifications are illustrated below in the specific case of the monitor function (44).

For this monitor, whose integral represents a surface element area on the u manifold, the appropriate scaled monitor is (cf. (6))

$$\tilde{m}(t, u) = t^{-d\beta} \sqrt{1 + t^{2(-\gamma+\beta)} (\nabla u)^2} = \sqrt{t^{-2d\beta} + t^{-2\Gamma} (\nabla u)^2}, \quad (46)$$

where $\Gamma = \gamma - (1 - d)\beta$ and

$$\frac{\partial \tilde{m}}{\partial t} = \frac{-d\beta t^{-2d\beta-1} - \Gamma t^{-2\Gamma-1}(\nabla u)^2 + t^{-2\Gamma}\nabla u \cdot \nabla q}{\sqrt{t^{-2d\beta} + t^{-2\Gamma}(\nabla u)^2}} \quad (47)$$

where from the PDE (1)

$$q = \frac{\partial u}{\partial t} = Lu. \quad (48)$$

In applying linear finite elements in this case we shall also use the approximation $Q \approx q$ by piecewise linear functions, where Q may be recovered from the weak form

$$\int_{\Omega(t)} W_i Q \, d\Omega = - \int_{\Omega(t)} U^n \nabla W_i \cdot \nabla U \, d\Omega, \quad (49)$$

of (48), using $Q = \sum_j Q_j W_j$.

The weak conservation principle (19) sought is now

$$\int_{\Omega(t)} W_i \sqrt{t^{-2d\beta} + t^{-2\Gamma}(\nabla U)^2} \, d\Omega = \tilde{C}_i \quad (50)$$

which leads to the following form of (20),

$$\int_{\Omega(t)} \sqrt{t^{-2d\beta} + t^{-2\Gamma}(\nabla U)^2} \nabla W_i \cdot \nabla \Phi \, d\Omega = \int_{\Omega(t)} W_i \frac{\partial \tilde{m}}{\partial t} \, d\Omega, \quad (51)$$

where $\partial \tilde{m}/\partial t$ is given by (47) with u replaced by U and q replaced by Q .

10 Computational results for the porous medium equation

The proposed method is tested using a range of problems involving the two-dimensional porous medium equation (3) referred to in Section 1 (where the corresponding values for the scaling parameters, β and γ , are given). This equation possesses self-similar solutions with compact support and a moving boundary which act as intermediate asymptotic solutions as well as attractors [4] and are used here to verify the present approach. In addition to comparison with known analytic solutions, more complex cases are also considered, for

which there is no exact solution. The accuracy of these results is assessed by comparison with those obtained using a more conventional adaptive approach based on h -refinement. It should be noted however that the goal of this work is *not* to test the relative efficiency of the two methods but rather to use the h -refinement solution as a mechanism for confirming the validity and accuracy of the moving mesh results.

It is worth briefly summarising here the moving mesh method for which tests will be presented. The vectors $\underline{\mathbf{X}}$ and \underline{U} of mesh node positions and solution values at these nodes are updated from one time level to the next as follows:

- (1) Calculate the velocity potentials Φ , using (20), for the given choice of monitor: $\tilde{m} = u$ or $\tilde{m} = t^{-d\beta} \sqrt{1 + t^{2(\beta-\gamma)} (\nabla u)^2}$ in this paper. Use of the latter requires $Q = LU$ to be recovered via (49) prior to this step.
- (2) Calculate the nodal velocities $\dot{\mathbf{X}}$ via (21).
- (3) Choose one of the following options for updating U (*cf.* section 8).
 - (a) Recover U directly from the scale-invariant conservation principle (19). (This is straightforward for $\tilde{m} = u$ but leads to an inappropriate nonlinear system when $\tilde{m} = t^{-d\beta} \sqrt{1 + t^{2(\beta-\gamma)} (\nabla u)^2}$.)
 - (b) Calculate \dot{U} from the *non-conservative* ALE formulation (37) and use scale-invariant, forward Euler time-stepping to update U .
 - (c) Calculate $\dot{\Theta}$ from the *conservative* ALE formulation (40) and use scale-invariant forward Euler time-stepping to update Θ (which can either be carried within the code or calculated at any stage from U), finally recovering U from (42). This is equivalent to option (a) in the case when $\tilde{m} = u$.
- (4) Update \mathbf{X} from (28) using the scale-invariant forward Euler discretisation of the time derivative, as specified in (35).
- (5) If the end time of the experiment has not been reached then repeat from step (1).

It is worth noting here that in step (3) options (b) and (c) require a similar amount of computational effort whereas option (a) is generally considerably more expensive when $\tilde{m} \neq u$. Furthermore, the results obtained from option (a) are, strictly speaking, only valid when $\int_{\Omega(t)} \tilde{m} d\Omega$ is constant in time (which is true in the case of the mass-conserving PME for $\tilde{m} = u$ but not generally when $\tilde{m} = t^{-d\beta} \sqrt{1 + t^{2(\beta-\gamma)} (\nabla u)^2}$). In fact it is demonstrated in the examples below that option (c) is clearly preferable both in terms of accuracy and robustness.

10.1 Numerical results for self-similar solutions

A family of radially symmetric self-similar solutions to the porous medium equation (3) is given by [16]

$$u(r, t) = \frac{1}{\mu^d} \left[\max \left(1 - \left(\frac{r}{r_0 \mu} \right)^2, 0 \right) \right]^{\frac{1}{n}} \quad (52)$$

in which d is the number of space dimensions, r is the radial coordinate, and

$$\mu = \left(\frac{t}{t_0} \right)^{\frac{1}{2+dn}}, \quad t_0 = \frac{r_0^2 n}{2(2+dn)}. \quad (53)$$

The specification of the problem is completed by choosing the initial radius r_0 . Only two-dimensional ($d = 2$) solutions will be considered here, though the theory is equally applicable to one- and three-dimensional problems. All test cases considered use either $n = 1$ or $n = 2$. For the latter, the self-similar solution has an infinite gradient normal to the moving boundary.

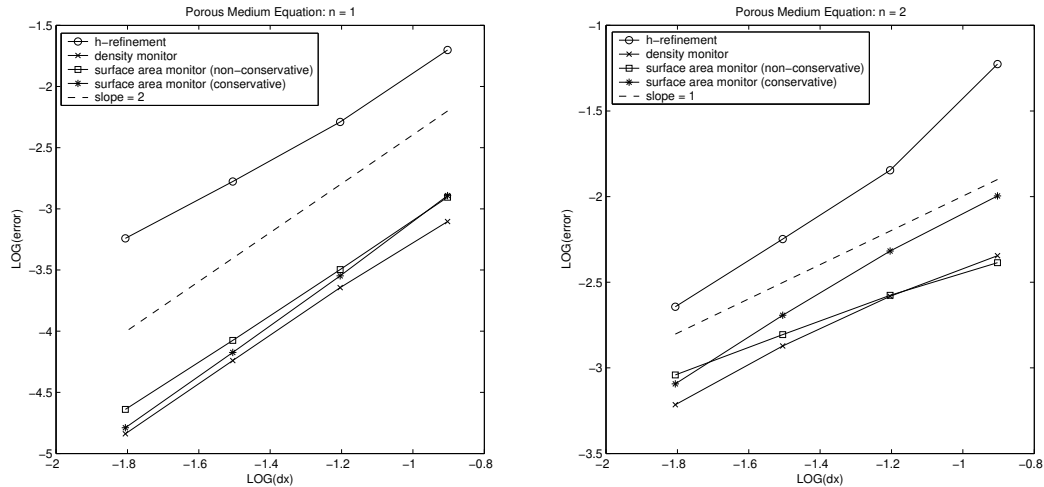


Fig. 1. Comparison of L^1 errors for the different approximation schemes.

Figures 1 and 2 show how the L^1 and L^2 errors vary with the size of the mesh for the different approaches. They compare results obtained on moving meshes for the two monitors used and, in the case of the surface area monitor, the two valid methods (b) and (c) for recovering U . Comparison is also made with results from an h -refinement scheme that is outlined in Section 10.2 below. For the moving mesh method a series of uniform triangular meshes have been used for the *initial* data and dx refers to the initial mesh size. Scale-invariant time-stepping has been used throughout, although the results obtained using constant dt rather than constant $d\sigma$ are very similar (except that they require

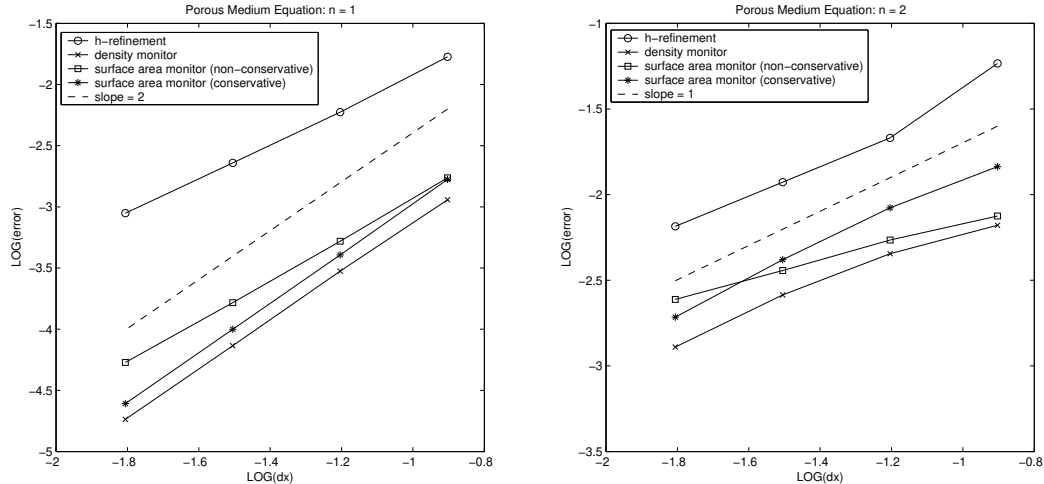


Fig. 2. Comparison of L^2 errors for the different approximation schemes.

more time steps). The h -refinement method has been used in a manner which covers the support of the solution with a uniform quadrilateral mesh of size dx . Both sets of results were initialised using (52) with $r_0 = 1.0$ and $t = t_0$, where t_0 is the initial time calculated using (53), and the comparisons are made at $T = 0.04$ where $T = t - t_0$.

Typically the density monitor $m = u$ gives the most accurate approximations at these mesh resolutions, but the surface area monitor with conservative ALE recovery for u appears to exhibit a slightly higher *order* of accuracy, particularly when $n = 2$. In general the moving mesh method loses an order of accuracy when $n = 2$, where the gradient of the exact solution is infinite at the boundary. The apparent lower accuracy of the h -refinement approach in this case is not surprising because the solutions it is approximating have infinite second derivative at the moving boundary which is, in this case, inside the domain. It is also worth noting that for both $n = 1$ and $n = 2$ mass conservation is important for maintaining accuracy. In [6] the problem of reduced accuracy at the front is successfully treated in one dimension using h refinement within the moving frame.

10.2 Numerical results for non-self-similar solutions

In this section the accuracy of the moving mesh method will be assessed for a wider range of PME problems, for which analytical solutions are not available. In order to achieve this we make use of a standard adaptive finite element approach based upon the use of h -refinement [12], which has also been checked against known similarity solutions in order to confirm its accuracy, as demonstrated in Section 10.1 above.

The benchmark adaptive h -refinement scheme that is used is based upon a standard quad-tree data structure for the organisation of a hierarchical mesh of bilinear quadrilateral elements. This mesh may be refined to different levels in different geometric regions with the restriction that no neighbouring elements may be more than one level apart in the hierarchy. This ensures that no elements in the mesh have any edges with more than one “hanging node”. The solution procedure on this mesh is based upon a standard finite element discretisation of the PME on a spatial domain that far exceeds the support of the initial data. This leads to an ordinary differential equation (ODE) system for the nodal solution values which is solved using an implicit, second order scheme (the trapezoidal rule). At each time-step this requires the solution of a system of nonlinear algebraic equations which is found using a full approximation scheme (FAS) multigrid algorithm, as described in [12]. This particular solver deals with the hanging nodes quite naturally by prescribing the solution values at such points to be the average of the values at the ends of the edge upon which the hanging node lies, [14]. Furthermore, because the implicit time-stepping scheme is a one-step method it is straightforward to adapt the mesh at the end of each step and interpolate the latest solution onto the new mesh before continuing.

For the purposes of the benchmark solutions obtained here a particularly cautious adaptivity strategy is used (relative efficiency not being an issue considered here). This essentially undertakes uniform refinement up to a maximum prescribed level wherever the solution is non-zero (to be more precise, $|u| \leq \varepsilon$ for some small choice of ε). In addition, numerous “safety layers” of refinement are imposed around the edge of this refined region and further safety layers of elements are imposed at each level of the mesh hierarchy down to the coarsest mesh. A typical mesh, with a maximum refinement level of 8, is shown in Figure 3 (right). It should be noted that, in all of the examples shown, the spatial error dominates the temporal error (reducing the length of the time-steps has little effect on the overall error) and the solutions are not affected significantly by altering either the value of ε or the number of safety layers in the adaptive algorithm. In the comparisons that follow the level 9 mesh is used (the finest of the ones for which results are presented in Section 10.1).

The first set of problems studied in this section are radially symmetric and use the initial conditions (when $t = t_0$) of the self-similar solutions (52),

$$u(r, t_0) = \left[\max \left(1 - \left(\frac{r}{r_0} \right)^2, 0 \right) \right]^{\frac{1}{n}} \quad (54)$$

for $n = 1$ and $n = 2$ with $r_0 = 0.5$. In order to deviate from the known similarity solutions, when the initial conditions are prescribed by the self-

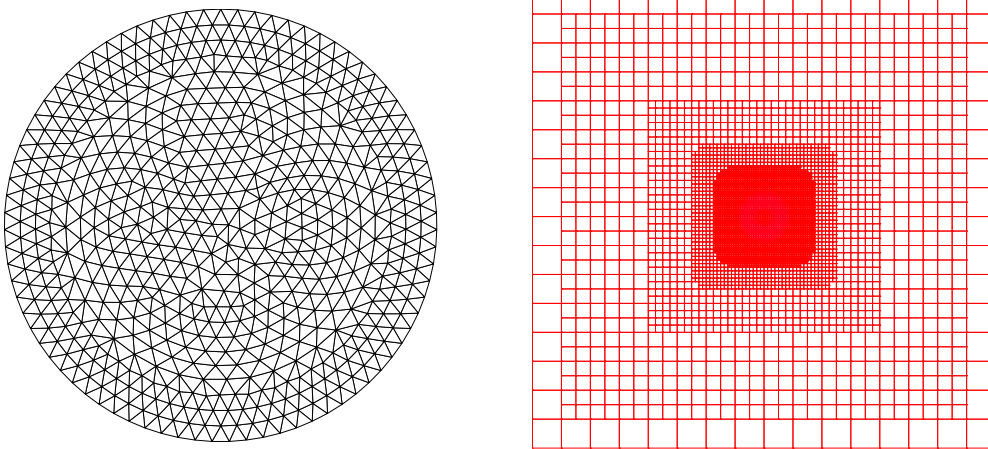


Fig. 3. The initial, 615 node, 1149 cell, unstructured triangular mesh used for all of the radially symmetric, non-self-similar, moving mesh results with $r_0 = 0.5$ (left) and a typical initial adaptive quadrilateral mesh (shown to level 8 meshes) as constructed by the h -refinement approach on the much larger domain $[-4, 4] \times [-4, 4]$ (right).

similar solution for $n = 1$ the solution is evolved according to the PME with $n = 2$, and vice versa. In all cases the moving mesh method uses the scale-invariant forward Euler time-stepping with constant $d\sigma = 0.0001$. The initial mesh used to produce all the moving mesh results is shown in Figure 3 (left), along with a typical initial mesh used by the h -refinement approach (right, and shown on a different scale since this initial mesh covers a much larger region).

Figure 4 shows slices through the solution surface along $y = 0$ for three moving mesh approaches, all compared with the solution obtained from the h -refinement approach on the level 9 mesh. Note that the circles are plotted where the unstructured mesh edges intersect $y = 0$, giving an uneven looking distribution. The moving mesh methods with conservative recovery clearly agree closely with the h -refinement method, although it is worth noting that at the moving boundary small oscillations appear in the fixed mesh solutions (while the moving mesh solutions do not match the condition $u = 0$ exactly on the boundary). The non-conservative recovery is clearly inferior even though the boundary condition is satisfied exactly. In fact, with the $n = 2$ initial conditions, almost 10% of the “mass” has been lost by the end of the experiment (whereas it is conserved exactly by the conservative ALE scheme). Figures 5 and 6 show the initial and final ($T = 2.0$) solution surfaces obtained with the moving mesh method using the surface area monitor and conservative ALE recovery. Figure 7 shows the final meshes for this case and illustrates how the nodes move towards the boundary as the solution steepens there in the case when $n = 2$ is used to evolve the solution. It should be noted that the large aspect ratio of the elements near the boundary in this case is advantageous since the elements are appropriately aligned with the solution (i.e. with steep

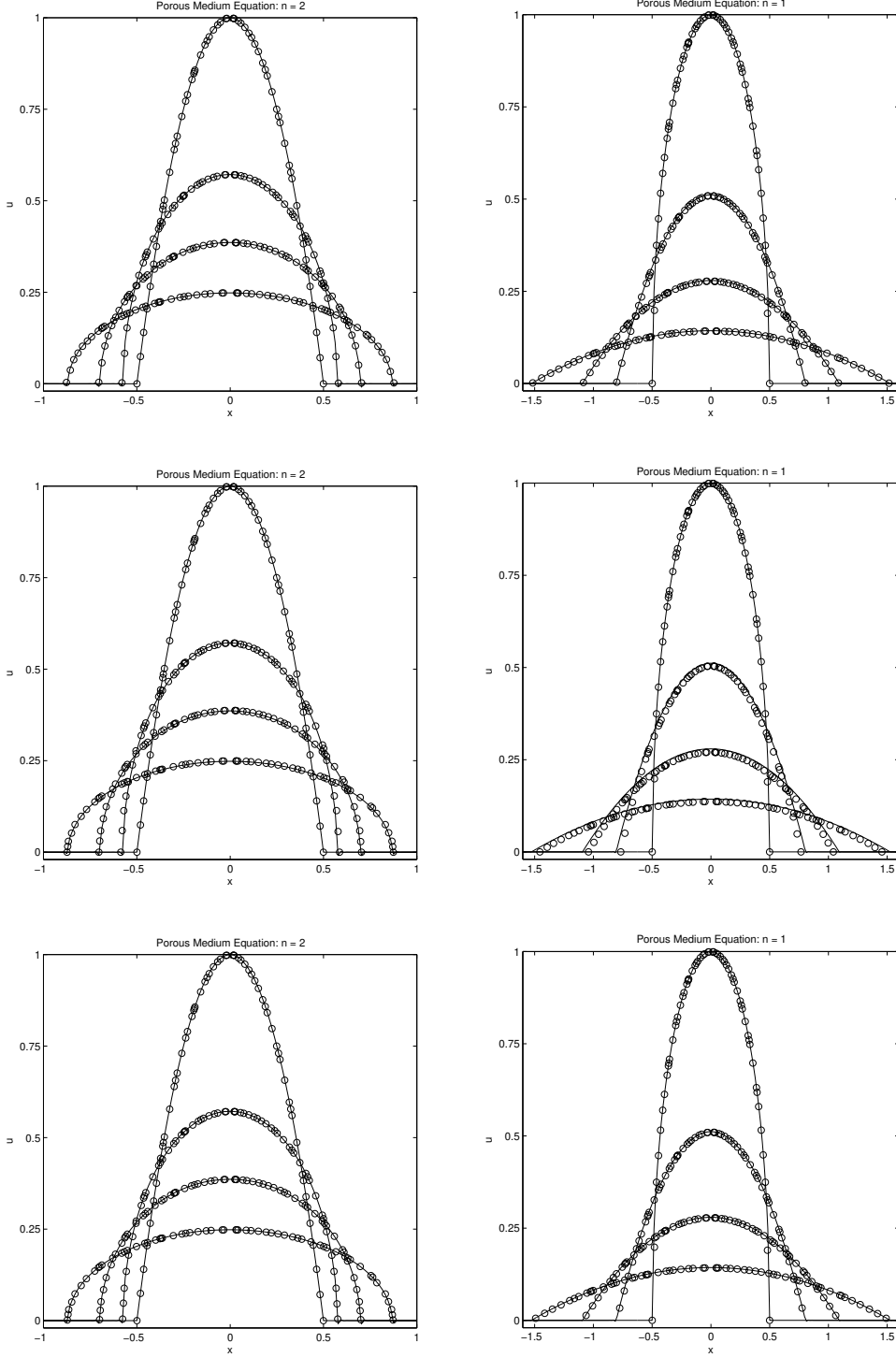


Fig. 4. Slices through the solution surface along $y = 0$ comparing the h -refinement method (solid lines) with three moving mesh methods (circles): $\tilde{m} = u$ with direct recovery of the solution (top) and $\tilde{m} = t^{-d\beta} \sqrt{1 + t^{2(\beta-\gamma)} (\nabla u)^2}$ with non-conservative (middle) and conservative (bottom) ALE formulations for recovering the solution. Initial conditions are given by the self-similar solution with $n = 1$ while the evolution is governed by the PME with $n = 2$ (left) or vice versa (right). The four snapshots are taken at $T = 0.0, 0.125, 0.5, 2.0$.

gradients in the direction where the element is shortest) [1,13].

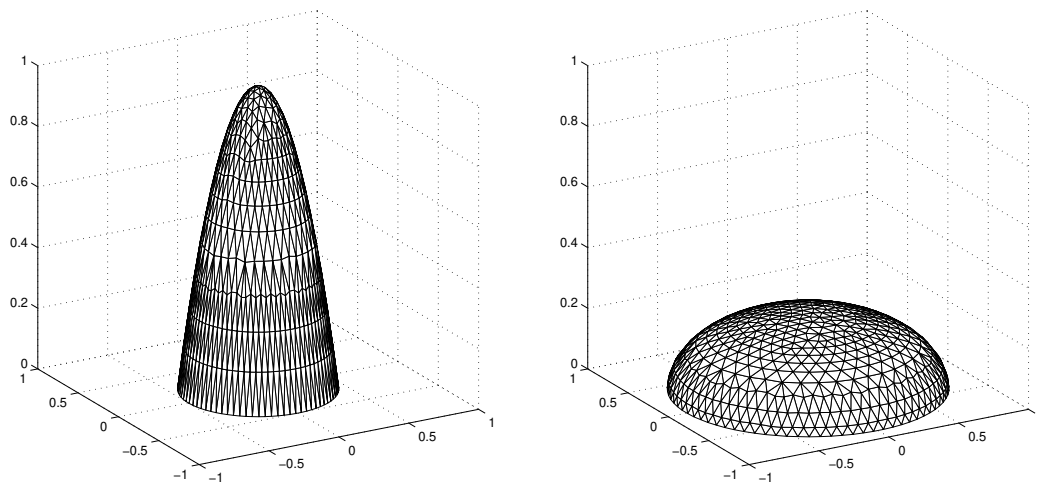


Fig. 5. The initial (left) and final (right) solution surfaces obtained using the moving mesh method with the surface area type monitor and conservative ALE recovery, with initial conditions calculated from the self-similar solution using $n = 1$ and the evolution governed by the PME with $n = 2$.

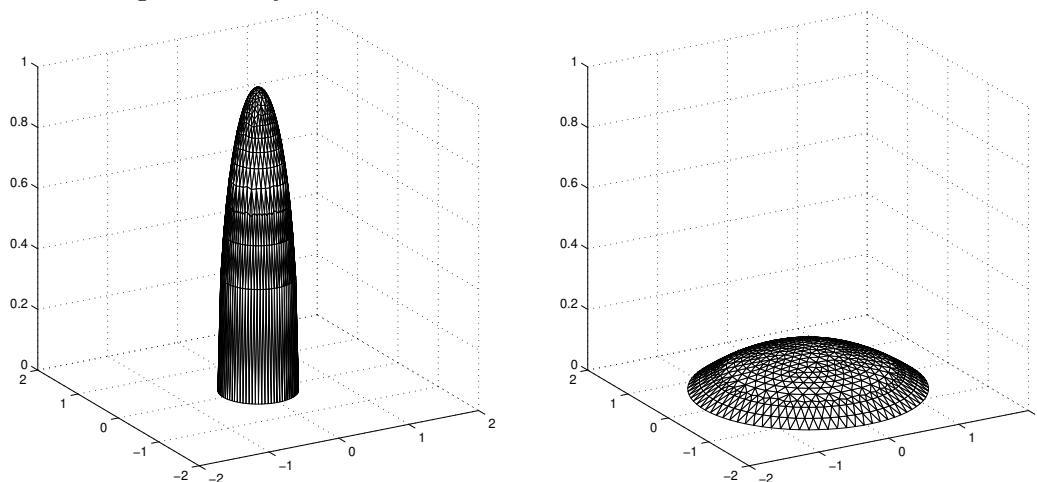


Fig. 6. The initial (left) and final (right) solution surfaces obtained using the moving mesh method with the surface area type monitor and conservative ALE recovery, with initial conditions calculated from the self-similar solution with $n = 2$ and the evolution governed by the PME with $n = 1$.

It is informative here to consider the evolution of the integrals on the left-hand side of (19) after U has been recovered. Figure 8 contains plots of the value of this quantity associated with 15 representative mesh nodes (those closest to a ray emanating from the centre of the domain along the negative y -axis) against time: one self-similar solution and one non-self-similar solution are shown. In both cases there is a brief “settling down” period, after which these integrals become roughly constant. In the self-similar case the dependent variable is recovered via the conservative ALE approach, not directly from (19), so exact preservation is not expected: it will depend on the accuracy

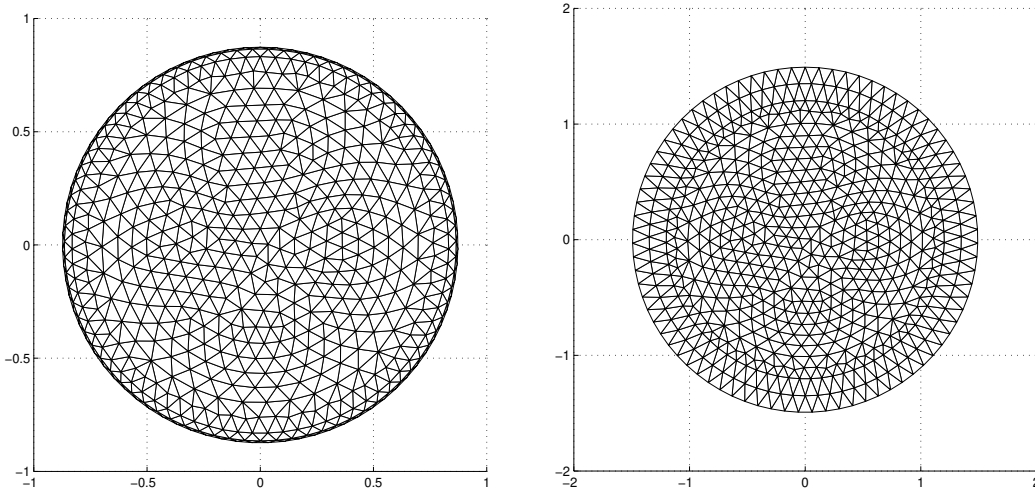


Fig. 7. The final meshes obtained using the moving mesh method with the surface area type monitor and conservative ALE recovery, with initial conditions calculated from the self-similar solution with $n = 1$ and the evolution governed by the PME with $n = 2$ (left) and vice versa (right).

with which the mesh velocity is calculated and constrained by total mass conservation. The initial perturbations are far more pronounced in the non-self-similar case, where these integrals are not expected to remain exactly constant after recovery, particularly at the nodes near to the boundary where the additional recovery stages also have some effect on the accuracy of the approximation. Subsequently, the local integrals in both cases become almost constant. The integral of the monitor over the whole domain exhibits similar behaviour: in the non-self-similar case it drops by about 10% in the early stages and then settles down: in the self-similar case it varies by less than 1%.

Finally, the fixed and moving mesh approaches are compared for problems which do not possess any radial symmetry. The initial conditions are now given by

$$u(r, t_0) = \left[\max \left(1 - \left(\frac{r}{r'_0} \right)^2, 0 \right) \right]^{\frac{1}{n}} \quad (55)$$

where

$$r'_0 = r_0(1 + \epsilon \cos(N \tan^{-1}(y/x))).$$

The cases considered here use $r_0 = 0.5$, $\epsilon = 0.2$ and $N = 3$, along with $n = 1$ and $n = 2$ in the PME. Once more, if the initial conditions are prescribed by one value of n in (55) the solution is evolved using the PME with the other. To construct the initial grid for the moving mesh method each node in the

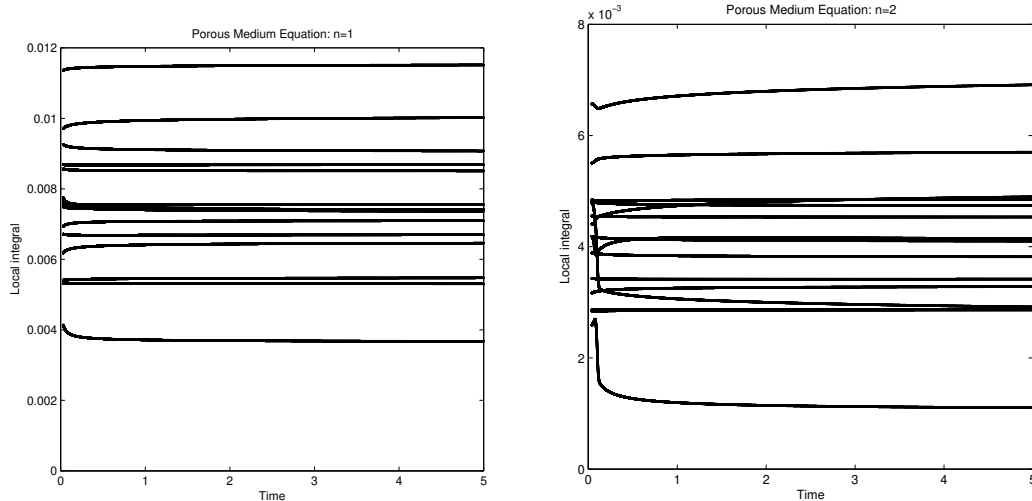


Fig. 8. The evolution of the values of the local integral of the scale-invariant monitor associated with 15 representative mesh nodes, using the surface area monitor and conservative ALE recovery, with initial conditions given by the self-similar solution with $n = 1$ and the evolution governed by the PME with $n = 1$ (left) and $n = 2$ (right).

mesh shown in Figure 3 (left) undergoes the transformation

$$\mathbf{x}_i = \mathbf{x}_i (1 + \epsilon \cos(N \tan^{-1}(y/x))).$$

In this case only the surface area monitor (which moves points towards regions where the solution gradient increases) is used, combined with conservative ALE recovery (which has shown itself to be superior to the non-conservative approach), for the comparison.

The solutions can still be compared adequately using slices through the data along $y = 0$ and these are shown in Figure 9. It is clear that the two approaches again agree closely for these cases. The evolution of the solution surfaces for the two methods are compared in Figures 10 and 11, which illustrate that no anomalies were hidden by only taking slices through the data. The initial and final meshes for the moving mesh approach are shown for both cases in Figures 12 and 13 and exhibit similar qualitative features to the earlier results.

11 Discussion

We have presented a scale-invariant moving mesh finite element method which extends the work of [3] to enforce the scale invariance which the numerical scheme should, ideally, inherit from the PDE that it is modelling. These extensions include the construction of scale-invariant monitor functions (one

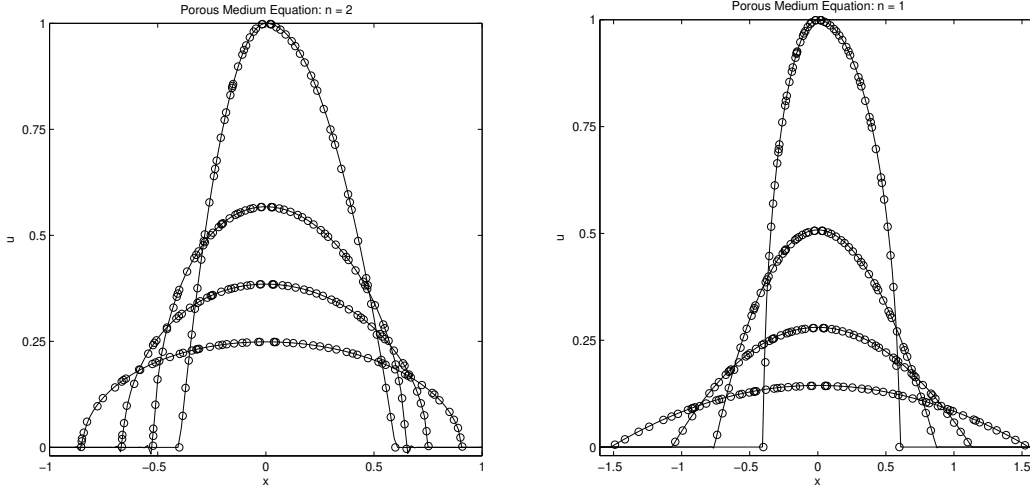


Fig. 9. Slices through the solution along $y = 0$ comparing the moving mesh method (circles) with the h -refinement approach (solid lines). Initial conditions are derived from (55) with $n = 1$ and the evolution governed by the PME with $n = 2$ (left) or vice versa (right). The four snapshots are taken at $T = 0.0, 0.125, 0.5, 2.0$.

of which, based on surface area, is illustrated in this paper) used to drive the mesh movement through a conservation principle, and the use of scale-invariant time-stepping which yields a scheme with uniform local truncation error in time.

Results have been presented for a variety of test cases using the porous medium equation in two space dimensions and have been validated against self-similar analytical solutions and other solutions. In the latter case they have been compared with results from a state-of-the-art h -refinement finite element approach on uniform quadrilateral meshes. It has been shown that the conservative ALE approach provides the best means of recovering solution values at the nodes once the mesh velocity has been obtained and that the resulting scale-invariant moving mesh method is highly accurate. This remains true even in cases when the conservation principle (19) is only approximately constant in time (*i.e.* non-self-similar solutions simulated using the surface area monitor), so long as the monitor is only used in the construction of the mesh velocities, not the recovery of the dependent variable, and so long as the scale-invariant version of the monitor is used. (Note that results obtained using the unscaled monitor are extremely poor.)

It is emphasised that the comparison with the h -refinement approach is not intended to illustrate the efficiency of either method. The moving mesh approach is explicit, while the h -refinement method employs an implicit multigrid scheme. In addition the latter approach was used with a very cautious adaptive strategy (with no de-refinement) to give solutions for comparison which could be deemed accurate. The only safe conclusion which can be drawn is that the moving mesh approach is able to provide similar accuracy to the h -refinement

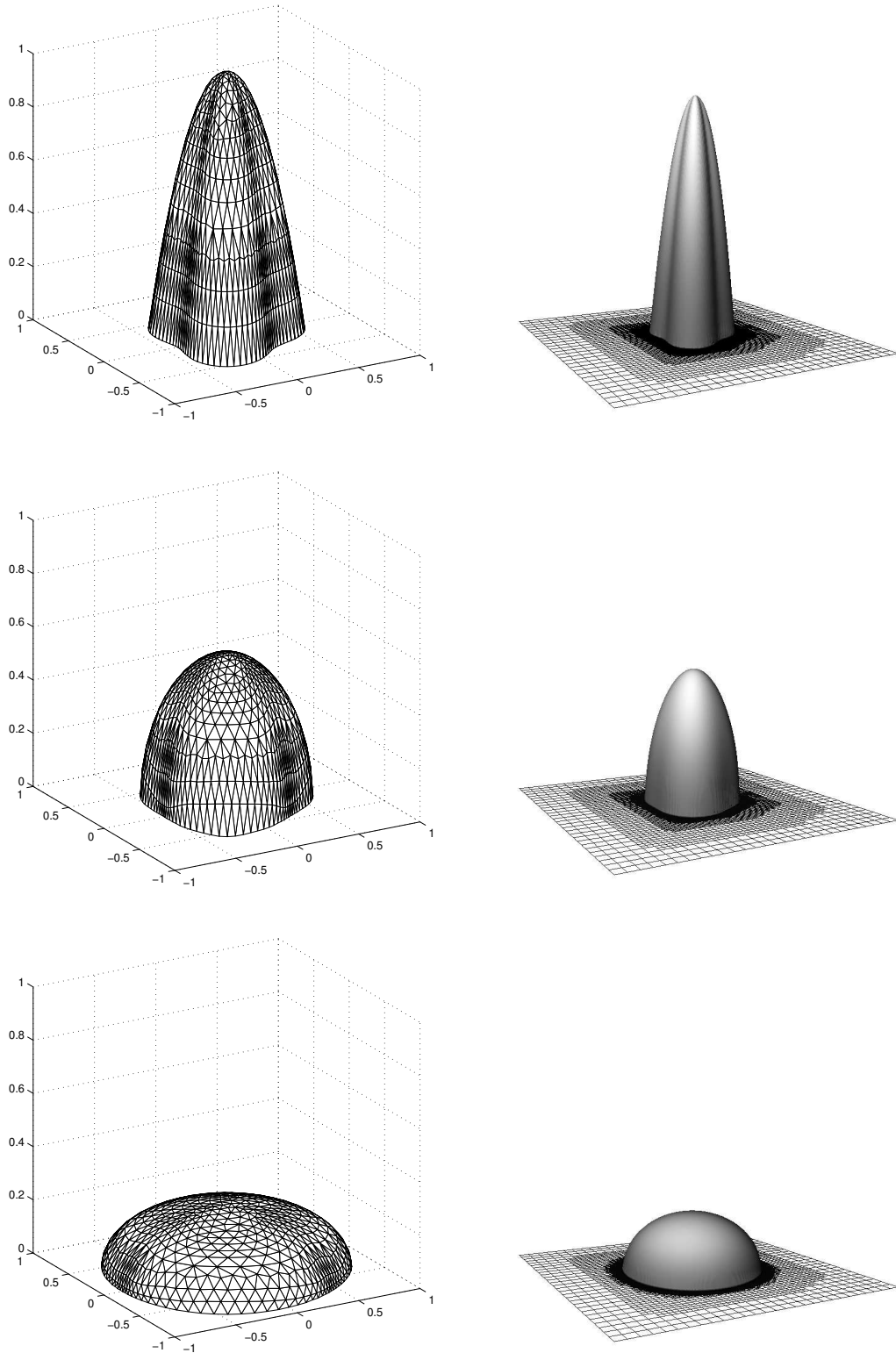


Fig. 10. The solution surfaces obtained from the moving mesh method (left) and the h -refinement method (right) at times $T = 0.0$ (top), $T = 0.125$ (middle) and $T = 2.0$ (bottom), for the non-radially-symmetric case with initial conditions calculated using (55) with $n = 1$ and the evolution governed by the PME with $n = 2$.

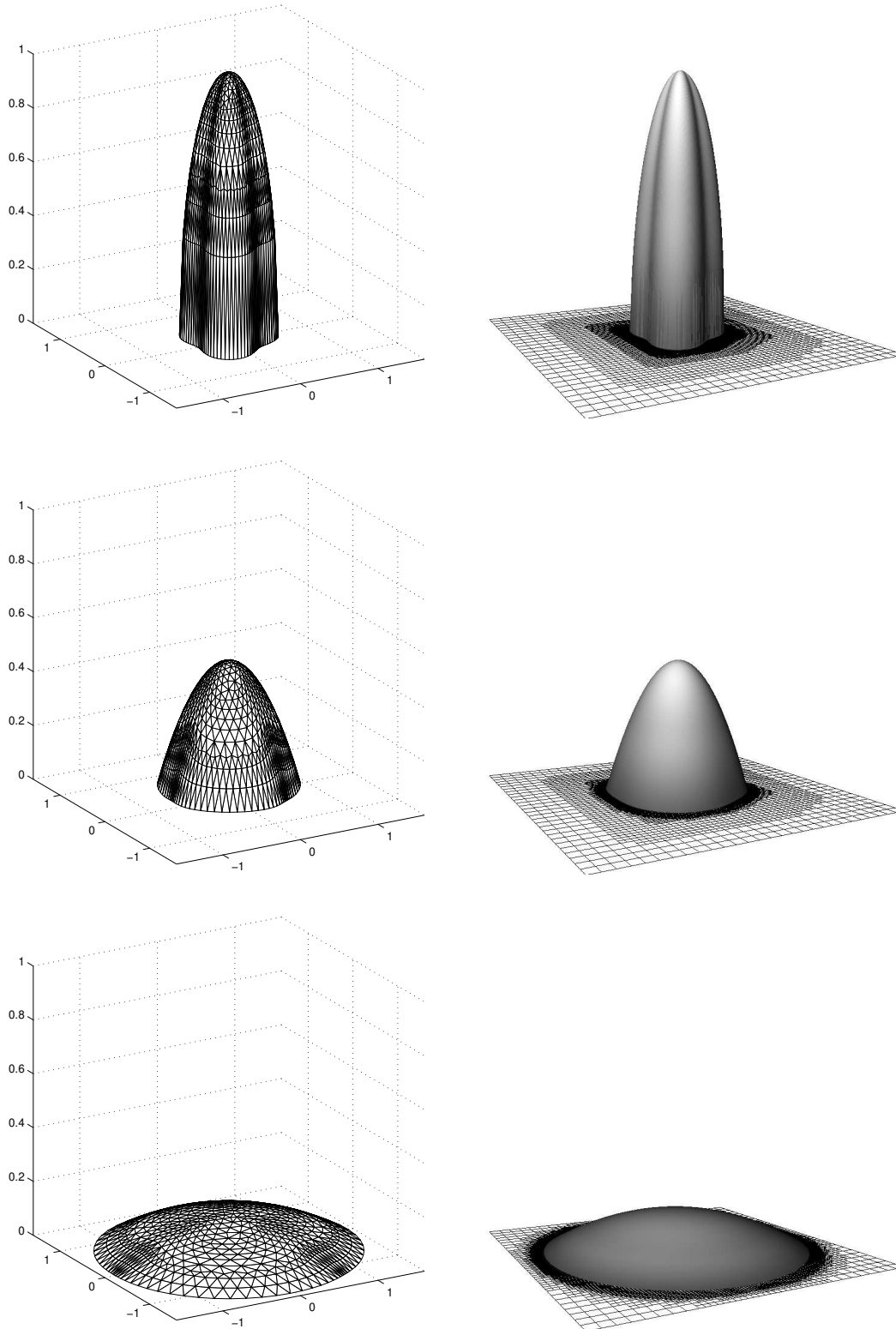


Fig. 11. The solution surfaces obtained from the moving mesh method (left) and the h -refinement method (right) at times $T = 0.0$ (top), $T = 0.125$ (middle) and $T = 2.0$ (bottom), for the non-radially-symmetric case with initial conditions calculated using (55) with $n = 2$ and the evolution governed by the PME with $n = 1$.

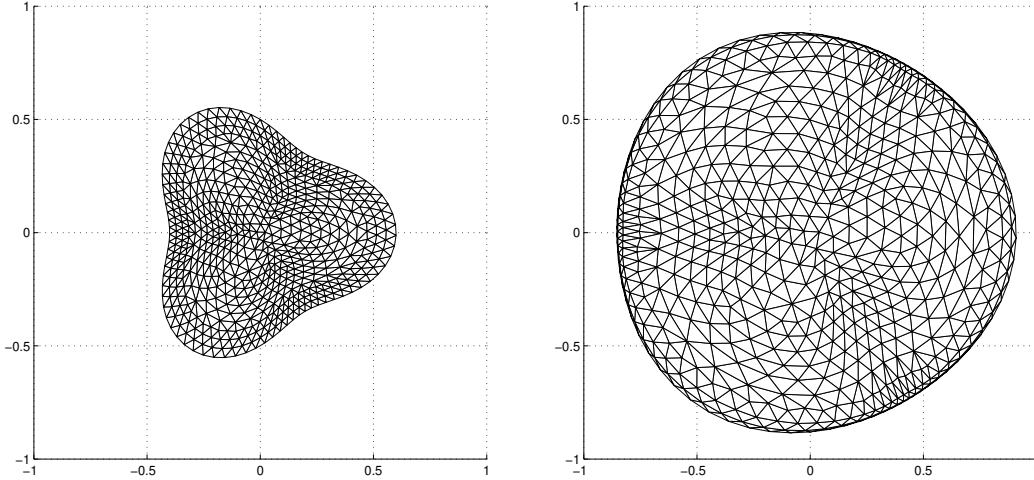


Fig. 12. The initial (left) and final (right) meshes obtained using the moving mesh method with non-radially-symmetric initial conditions calculated using (55) with $n = 1$ and the evolution governed by the PME with $n = 2$.

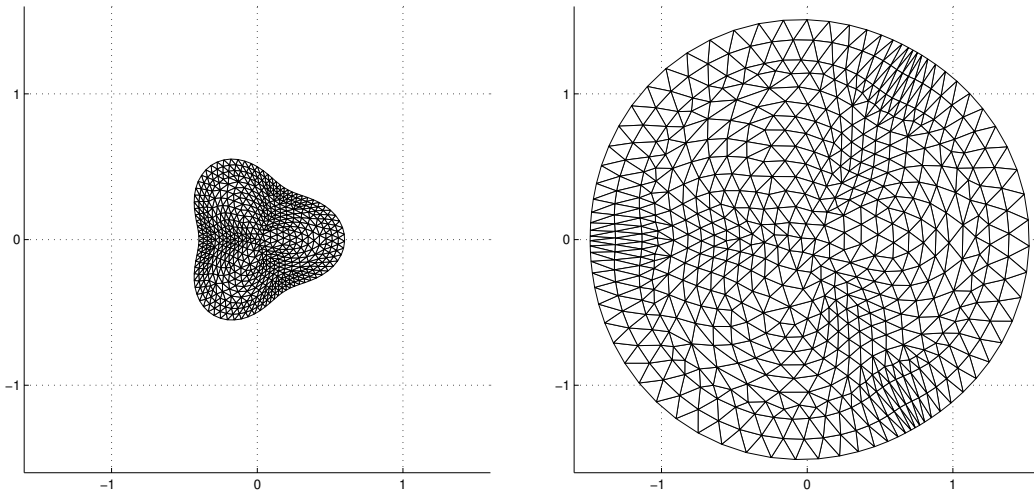


Fig. 13. The initial (left) and final (right) meshes obtained using the moving mesh method with non-radially-symmetric initial conditions calculated using (55) with $n = 2$ and the evolution governed by the PME with $n = 1$.

approach with fewer degrees of freedom.

References

- [1] Apel T, Grosman S, Jimack PK, Meyer A. A new methodology for anisotropic mesh refinement based upon error gradients. *Appl. Numer. Math.* 2004; **50**: 329-341.
- [2] Baines, MJ. *Moving Finite Elements*, OUP 1994.

- [3] Baines MJ, Hubbard ME, Jimack PK. A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries. *Appl. Numer. Math.* 2004; to appear.
- [4] Barenblatt GI. *Scale Invariance, Self-similarity and Intermediate Asymptotics*. CUP 1996.
- [5] Beckett G, Mackenzie JA, Robertson ML. A moving mesh finite element method for the solution of two-dimensional Stefan problems. *J. Comput. Phys.* 2001; **168**:500–518.
- [6] Blake KW, Baines MJ. A moving mesh method for nonlinear parabolic problems. Numerical Analysis Report 2/2002, Department of Mathematics, University of Reading, 2002.
- [7] Blake KW. *Moving Mesh Methods for Nonlinear Partial Differential Equations*. PhD Thesis, University of Reading, 2001. (See also Blake KW, Baines MJ, Numerical Analysis Report 2/02, Department of Mathematics, University of Reading, UK.)
- [8] Budd CJ, Leimkuhler B, Piggott M. Scaling invariance and adaptivity. *Appl. Numer. Math.* 2001; **39**: 261-288.
- [9] Budd CJ, Piggott M. Geometric integration and its applications. *Handbook of Numerical Analysis, XI* 2003; **128**:35–139.
- [10] Cao W, Huang W, Russell RD. A moving mesh method based on the geometric conservation law. *SIAM J. Sci. Comput.* 2002; **24**:118–142.
- [11] Huang W, Ren Y, Russell RD. Moving mesh partial differential equations (MMPDEs) based on the equidistribution principle. *SIAM J. Numer. Anal.* 1994; **31**:709–730.
- [12] Jones AC, Jimack PK. An adaptive multigrid tool for elliptic and parabolic systems *Int. J. Numer. Meth. Fluids* 2004; to appear.
- [13] Kunert, G. Robust local problem error estimation for a singularly perturbed problem on anisotropic finite element meshes. *Math. Model. Numer. Anal.* 2001; **35**:1079–1109.
- [14] Meyer A. Projection techniques embedded in the PCGM for handling hanging nodes and boundary restrictions. In *Engineering Computational Technology*, Topping B.H.V, Bittnar Z (eds). Saxe-Coburg Publications, Stirling, Scotland, 2002; 147–165.
- [15] Miller K, Miller RN. Moving finite elements, part I. *SIAM J. Numer. Anal.* 1982; **18**:1019–1057.
- [16] Murray JD. *Mathematical Biology: An Introduction* (3rd edition). Springer, 2002.
- [17] Thomas PD, Lombard CK. The geometric conservation law and its application to flow computations on moving grids, *AIAA Journal* 1979; **17**:1030–1037.

- [18] Wesseling P. *Principles of Fluid Dynamics*. Springer-Verlag: Berlin, Heidelberg, 2001.