Parabolic Monge–Ampère methods for blow-up problems in several spatial dimensions

2006 J. Phys. A: Math. Gen. 39 5425

(http://iopscience.iop.org/0305-4470/39/19/S06)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **39** (2006) 5425–5444 doi:10.1088/0305-4470/39/19/S06

Parabolic Monge–Ampère methods for blow-up problems in several spatial dimensions

C J Budd[1] **and J F Williams**[2]

[1] Department of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK
[2] Department of Mathematics, Simon Fraser University, Burnaby, BC, V6T 1Z2, Canada

E-mail: cjb@maths.bath.ac.uk and jfw@math.sfu.ca

Received 18 October 2005, in final form 16 December 2005
Published 24 April 2006
Online at stacks.iop.org/JPhysA/39/5425

Abstract
This paper constructs and analyses an adaptive moving mesh scheme for the numerical simulation of singular PDEs in one or more spatial dimensions. The scheme is based on computing a Legendre transformation from a regular to a spatially non-uniform mesh via the solution of a relaxed form of the Monge–Ampère equation. The method is shown to preserve the inherent scaling properties of the PDE and to identify natural computational coordinates. Numerical examples are presented in one and two dimensions which demonstrate the effectiveness of this approach.

PACS numbers: 02.30.Jr, 02.60.Lj

1. Introduction

One of the key geometric features associated with the solution of partial differential equations is the evolution of structures on several different length scales. Indeed, the analysis of *multi-scale* phenomenon is now a major source of research activity. In certain problems, for example porous media flows, the length scales are widely separated and these problems can be studied using, for example, multi-scale finite element methods. However, there are many other problems in which length scales are not widely separated and indeed we see a *continuum* of length scales. These are generally driven by some nonlinearity in the problem and are tightly coupled in an essentially geometric manner. Scaling laws are almost universal in many partial differential equations and often describe asymptotic features of a solution such as the formation of singularities and interfaces, when the effects of boundary and initial conditions are less important. Examples arise in fluid mechanics, magneto hydrodynamics, solid mechanics, nonlinear optics and electrostatics. An excellent survey of the range of such problems is given in [2]. It is clear that numerical methods which can exploit the scaling structures in a partial differential equation are likely to perform well. To construct such methods we need to understand the *scaling symmetries* that lie behind the various scaling

structures. Such symmetries are generally described by Abelian, diagonalizable Lie groups, and thus have a simpler structure than many other problems studied by geometric methods. However, the universality of scaling laws makes it appropriate to develop methods just for them. The sort of problems that we are interested in studying have length scales $L(t)$ driven by nonlinear effects, which are vanishingly small in some limit. Typically there is some time T for which $L \to 0$ as $t \to T$. Such problems pose a challenge for any numerical method and cannot be studied by using a method with a fixed grid of size h when $L(t) \ll h$. In such cases, it is appropriate to use an *adaptive* method, in which the spatial mesh can adapt itself to the natural scale of the solution.

In this paper, we will examine a class of r-adaptive moving mesh methods for approximating those solutions of the parabolic equation

$$u_t = \Delta u + f(u), \qquad \mathbf{x} \in \Omega \subset R^d, \qquad u|_{\partial\Omega} = 0, \tag{1}$$

which become singular (blow-up) in a finite time T. We will concentrate in our calculations on the case of $d = 2$, although the methods can, in principle, work in higher dimensions. Whilst the solution of (1) in one dimension by using (scale invariant) r-adaptive methods is now fairly well understood [8], less progress has been made in extending these methods to higher dimensions where there are significant new difficulties. In this paper, we describe a new method for solving (1) in spatial dimensions greater than 1 which is well adapted to exploiting emergent scaling structures. These methods will be based on the parabolic Monge–Ampère method. In this method, a fixed mesh in *computational coordinates* $\boldsymbol{\xi}$ is mapped to a moving mesh $\mathbf{x}$ in the physical coordinates. The map from computational to physical coordinates is determined in terms of the gradient of a *mesh potential* Q which satisfies a parabolic partial differential equation. This equation is solved in parallel with (1) and is constructed to have the same scaling symmetries. We show that any scaling structure present in the solution u is inherited by the mesh potential function and derive from this a series of properties of the resulting mesh. We demonstrate through examples that (when careful attention is paid to mesh regularity and smoothing) the parabolic Monge–Ampère method is effective for computing the singular solutions of (1) in two spatial dimensions.

The layout of the remainder of this paper is as follows. In section 2, we describe the structure of the typical forms of singularity associated with the solutions of (1). In section 3, we derive the parabolic Monge–Ampère method for calculating a moving mesh and describe some properties of the resulting mesh. In particular, we show that if the natural length scale of the singularity is $L(t) \to 0$ then close to the singularity the mesh naturally inherits this scaling. In section 4, we describe how the mesh can be calculated together with the solution of (1), discussing certain regularization methods. Finally, in section 5 we give some examples of the application of this method.

2. Singularities, scaling structures and the Sundman transform

2.1. Singularities and scaling in PDES

Many nonlinear parabolic partial differential equations, such as blow-up problems of the form (1) with $f(u) = u^p$ [17], the chemotaxis equation [7] and the nonlinear Schrödinger equation [18] have the property that they develop solutions $u(\mathbf{x}, t)$ which become singular at a single point $\mathbf{x}^*$ in a finite time T. Typically such solutions develop a peak of amplitude $U(t) \to \infty$ and width $L(t) \to 0$ as $t \to T$ as illustrated in figure 1.

Asymptotically the solution in the peak, when $|\mathbf{x} - \mathbf{x}^*|$ is small, often has the separated form

$$u(\mathbf{x}, t) = U(t)V(\boldsymbol{\xi}), \qquad \boldsymbol{\xi} = (\mathbf{x} - \mathbf{x}^*)/L(t), \tag{2}$$

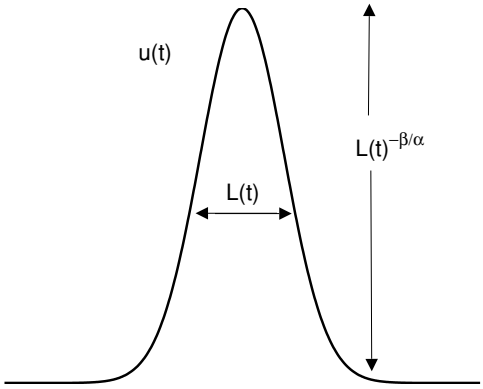

Figure 1. A typical peak, indicating the scaling relationship as the singularity develops for a self-similar solution.

where $U(t) \to \infty$ as $t \to T$ and V is a regular function of the scaled variable $\boldsymbol{\xi}$. This scaled variable represents a *natural coordinate* in which to express the solution and to use for numerical calculations in the computational domain. The three examples considered above all have singular solutions which take the form described by the expression in (2). Our aim is to construct an r-adaptive method in d-spatial dimensions which automatically aligns the mesh along the natural coordinates.

A general class of partial differential equations which admit (singular) solutions of the form described in (2) is that with a scaling structure, so that the partial differential equation is invariant under transformations of the form

$$t \to \lambda t, \qquad \mathbf{x} \to \lambda^{\alpha}\mathbf{x}, \qquad u \to \lambda^{-\beta}u. \tag{3}$$

These differential equations admit (separable) *self-similar* solutions which are themselves invariant under the action of the scaling transformation. These can take the form

$$u(\mathbf{x}, t) = (T - t)^{-\beta} V(\mathbf{x}/(T - t)^{\alpha}), \tag{4}$$

and have a natural length scale

$$L(t) = (T - t)^{\alpha}. \tag{5}$$

Many examples of such problems are found in [2]. For certain problems, however, the function V in (4) may be identically zero (or constant) and the self-similar profile does not adequately describe the asymptotic form of the singularity. There are, however, a class of problems, which includes the blow-up problem, chemotaxis and the two-dimensional nonlinear Schrödinger equation, which have *approximate self-similar* behaviour. In this case, the solution scale and/or the length scale have additional logarithmic terms so that (for example)

$$L(t) = (T - t)^{\alpha}|\log(T - t)|^{\kappa}. \tag{6}$$

Both of these cases will be considered in this paper, and we show that the parabolic Monge–Ampère method will correctly align the mesh in both cases, so that close to the singularity, the map from computational to physical domain is given locally by $\mathbf{x} = \mathbf{x}^* + L(t)S(\boldsymbol{\xi})$ where $S(\boldsymbol{\xi})$ is a (bounded) regular function of the computational coordinates. Computations which are singular in the physical domain then become regular in the computational domain.

2.2. Scaling in ordinary differential equations and the Sundman transform

In this paper, we will consider taking a semi-discretization of the partial differential equation for u and an associated equation for Q over a fixed computational space. This will lead to a set

of ordinary differential equations. These equations will have singularities in time, with large changes in behaviour as $t \to T$. This can lead to very stiff equations. To reduce the amount of stiffness, we will introduce a computational time coordinate τ so that computations in this coordinate become more regular. A natural way to do this is via the Sundman transform [11] in which we set

$$\frac{\mathrm{d}t}{\mathrm{d}\tau} = g(u)$$

and express all ordinary differential equations in terms of τ. In [11, 4], an analysis is made of the choice of g when solving ordinary differential equations with a similar scaling structure to (3) and it is shown that if g satisfies the scaling law $g(\lambda^{-\beta}u) = \lambda g(u)$, then the numerical method inherits discrete self-similar solutions which uniformly approximate the true self-similar solution. We will show in this paper how such methods can be combined with the parabolic Monge–Ampère method to give good scaling structures in both space and time. Such methods are then very well suited to calculating singular solutions of the form (2).

3. The parabolic Monge–Ampère method

In this section, we will describe the parabolic Monge–Ampère method for generating a moving mesh for parabolic problems in two or more spatial dimensions. We show how scaling invariance can be naturally built into the construction of this method. We then show how the numerical solutions naturally inherit the scaling behaviour of the underlying partial differential equation.

3.1. The Monge–Ampère equation

To solve the partial differential equation (1), we will consider using a semi-discrete r-adaptive moving mesh method to give a spatial mesh. The partial differential equation will then be discretized using this mesh. All discretizations will be made in the so-called *computational domain*. In this method, a fixed set of moving mesh points, also called *physical coordinates*

$$\mathbf{x}_i(t) = (x_i(t), y_i(t)),$$

is moved in accordance with the evolving dynamics of the underlying PDE, exploiting the scaling structure. This set will be considered to be the image of a *fixed* (in time) set of *computational coordinates*

$$\boldsymbol{\xi}_i = (\xi_i, \eta_i).$$

The PDE is then discretized in these variables. We contrast this with other moving mesh methods, for example the moving finite element method [1], in which all discretizations are made in terms of the physical coordinates.

To make things precise, we define a computational domain $\boldsymbol{\xi} \in \Omega_{\mathrm{C}} \subset R^d$ and a physical domain $\mathbf{x} \in \Omega_{\mathrm{P}} \subset R^d$ and consider a time-dependent mapping

$$\mathbf{x}(\boldsymbol{\xi}, t) : \Omega_{\mathrm{C}} \to \Omega_{\mathrm{P}}.$$

To avoid mesh tangling, it is essential that this map should always be locally invertible, so that

$$J \equiv \left| \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right| \neq 0 \qquad \text{for all} \quad \boldsymbol{\xi}, t. \tag{7}$$

A powerful technique for partially determining such a map is the *equidistribution principle* of [5]. To apply this, we define a *monitor function* $M(\mathbf{x}, t) > 0$ and for any (reference) set $A_{\mathrm{C}} \subset \Omega_{\mathrm{C}}$ require that

$$\int_{A_{\mathrm{C}}} \mathrm{d}\boldsymbol{\xi} \int_{\Omega_{\mathrm{P}}} M \,\mathrm{d}\mathbf{x} = \int_{A_{\mathrm{P}}} M \,\mathrm{d}\mathbf{x}, \tag{8}$$

where A_P is the image of A_C under the map $\mathbf{x}$. Differentiating (8), we have

$$MJ = \theta(t) \qquad \text{where} \quad \theta(t) = \int_{\Omega_P} M \, d\mathbf{x}. \tag{9}$$

Condition (9) is sufficient to define a map in one dimension as there is a unique (up to translation) connected interval of a prescribed length. It is used as the basis of a series of moving mesh methods, usually combined with a relaxation term. A summary of such methods can be found in [14], with applications to problems with scaling structure in [8]. However, equidistribution does not define a unique map in more than one dimension, for example many connected sets have the same area. To define such a map, we must augment condition (9). A powerful geometrically based approach to doing this is to insist that the mapping should be a *Legendre transformation*. To construct this, we introduce a *mesh potential* $P(\xi, t)$ and set

$$\mathbf{x} = \nabla P(\boldsymbol{\xi}, t). \tag{10}$$

Here, and in the rest of this paper, we use $\nabla = (\partial\xi_1, \partial\xi_2, \ldots, \partial\xi_d)$ to represent differentiation in the computational space. Derivatives with respect to the physical variables will be indicated, i.e., as ∇_x. Note that P is a scalar function, and the problem of finding a mesh in an arbitrary dimension reduces to the scalar problem of finding a suitable P. Such a mesh is *irrotational* and has the property that in a suitably defined norm, the mesh constructed is the solution of the equidistribution equation (9) which is as close as possible to the identity [6]. The Legendre transformation is always invertible, indeed there is a *dual potential* $R(\mathbf{x})$ so that

$$\boldsymbol{\xi} = \nabla_x R \qquad \text{and} \qquad P + R = \mathbf{x} \cdot \boldsymbol{\xi}. \tag{11}$$

Under this transformation, the Jacobian J becomes the Hessian $H(P)$. For $\boldsymbol{\xi} \in R^2$, this takes the form

$$H(P) = P_{\xi\xi} P_{\eta\eta} - P_{\xi\eta}^2. \tag{12}$$

A useful expression for H for radially symmetric functions P which depend only on $r = \sqrt{\xi^2 + \eta^2}$ is given by

$$H(P) = \frac{P_r P_{rr}}{r}. \tag{13}$$

Under the Legendre transformation, the equidistribution principle (9) becomes the fully nonlinear *Monge–Ampère equation* for P given by

$$MH(P) = \theta(t). \tag{14}$$

This equation must be augmented with suitable boundary conditions. The most natural of these is that the boundary of Ω_C should map to the boundary of Ω_P so that

$$\mathbf{x} : \partial\Omega_C \to \partial\Omega_P. \tag{15}$$

An important domain for computational purposes is given by

$$\Omega_C = \Omega_P \equiv [-1/2, 1/2]^2 \subset R^2,$$

for which both Ω_C and Ω_P have unit area. We can then impose that each side of the square Ω_C maps to the corresponding side of the square Ω_P. This gives the following boundary conditions:

$$P_\xi(-1/2, \eta) = P_\eta(\xi, -1/2) = -1/2, \qquad P_\xi(1/2, \eta) = P_\eta(\xi, 1/2) = 1/2. \tag{16}$$

It is known [6, 12, 16] that, under relatively mild conditions on M, the system (14), (16) has a unique solution P (up to an added constant), leading in turn to a unique mesh. Unfortunately, the system (14), (16) is fully nonlinear and extremely hard to solve directly. We now consider a relaxation approach to finding the solution.

It is convenient for later scaling calculations to rescale the Monge–Ampère equation. In particular, for $x \in R^d$ we set $M = N^d$ and $\theta = \phi^d$ to give

$$NH^{1/d}(P) = \phi(t). \tag{17}$$

It follows immediately from construction that if P solves (17) then

$$\phi = \left(\int_{\Omega_P} N^d \, \mathrm{d}\mathbf{x} \right)^{1/d}.$$

3.2. Time-dependent relaxation methods

Now consider the original partial differential equation (1) for the unknown function $u(x, t)$. To determine a suitable mesh to solve this problem we seek a scaled monitor function N, typically of the form $N(u)$ which is large when the features of the solution that we seek to resolve (typically the height or the gradient) are also large. A possible strategy for finding such a mesh is then to solve the scaled equation (17) in parallel with (1) so that we solve the coupled system

$$u_t = \Delta_x u + f(u), \qquad N(u(\nabla P, t))H^{1/d}(P) = \phi(t). \tag{18}$$

Discretizing (17) leads to a very nasty differential algebraic system to solve for the solution and for the mesh. Moreover, to solve this system we need to determine an accurate value of $\phi(t)$ which requires quadrature. However, (17) does not need to be solved exactly in order to find a good mesh for discretizing (1), nor do we need to find ϕ exactly. We now describe a relaxation method which avoids both difficulties.

Recall that the purpose of mesh adaptation is to find a mesh good enough to resolve the solution, in particular to respond to changes in the solution in a time scale appropriate to the solution itself. To do this we do not need to solve (17) exactly, indeed, a solution of (17) sufficiently close to the function $P(t)$ will, in general, give a mesh adequate to resolve the solution. A natural procedure for doing this is to *relax* equation (17) to give a *parabolic* partial differential equation, the solution of which evolves towards that of the Monge–Ampère equation. This approach for finding the mesh leads, in one dimension, to the *moving mesh equation* methods described in [14, 8] and has proved successful in this case. Temporal relaxation of the Monge–Ampère equation leads to the *parabolic Monge-Ampère* method. We now describe some of its properties in the context of differential equations with scaling laws. More detailed properties of the method, including a discussion of implementation, mesh regularity, mesh tangling and a comparison with some other methods used for r-adaptivity in two dimensions, are given in the paper [9].

To derive a suitable relaxed method, it is convenient to introduce a new time-dependent variable $Q(\boldsymbol{\xi}, t)$ so that $\mathbf{x}(t) = \nabla Q$. To be a relaxation method, we require that ∇Q should evolve towards (or be close to) ∇P. Note that this allows some freedom as we can have P and Q differing by a function $\Lambda(t)$ of time only, which vanishes on taking gradients. This degree of freedom eliminates the need to find ϕ exactly in advance. The relaxed form of (17) that we now consider for our subsequent calculations takes the form

$$\epsilon(I - \gamma\Delta)Q_t = N(\nabla Q, t)H(Q)^{1/d}, \qquad \mathbf{x} = \nabla Q \tag{19}$$

augmented with the boundary conditions (16) replacing ∇P for ∇Q in these conditions. Observe that we have not included ϕ in this expression. In the numerical calculations, we solve (19) in parallel with the partial differential equation (1). This leads to a combined dynamical system comprising the dynamics of both the solution and of the mesh. Typically, we see a rapid evolution of the mesh towards an equidistributed solution and then slower

evolution together with the solution. In both cases, the equations are discretized in the *computational domain* ξ. In (19), the parameter γ controls the degree of *spatial smoothing* in the method. The use of the smoothing operator $(I - \gamma\Delta)$ is motivated by the work of Huang and Russell [15] and has proved essential in giving smooth and regular meshes. The parameter ϵ controls the relaxation rate, which will be constructed to be a bounded fraction of the natural time scale of the partial differential equation (1). Typically, we start the calculation with a uniform mesh. For a two-dimensional domain this is given by

$$Q(\xi, \eta) = \tfrac{1}{2}(\xi^2 + \eta^2).$$

We then solve the parabolic partial differential equation (19) as cheaply as possible (using an explicit method if possible). Certain details of the implementation will be given in the next section. We now proceed to derive some properties of the solution of (19) for various choices of the function N.

3.3. Constant (in time) monitor functions

For simplicity, we consider firstly the case of a monitor function N depending only upon the *spatial mesh* $\mathbf{x}$ and not explicitly on time, so that it takes the form $N \equiv N(\nabla P) = N(\mathbf{x})$. We then look at the solution of (19) giving an equilibrium mesh and at the evolution of Q towards this solution.

Lemma 3.1. *Suppose that $N \equiv N(\nabla P)$ and P is a solution of the scaled Monge–Ampère equation (17) then the relaxed equation (19) has the particular solution*

$$Q(\boldsymbol{\xi}, t) = \phi t/\epsilon + P(\boldsymbol{\xi}), \qquad \nabla Q = \nabla P. \tag{20}$$

Proof. It is immediate that if Q satisfies (20) then $H(Q) = H(P)$ and $N(\nabla P) = N(\nabla Q)$. Furthermore, $\epsilon(I - \gamma\Delta)Q_t = \epsilon(I - \gamma\Delta)\phi/\epsilon = \phi$. The result follows. □

We now establish local stability of this solution under certain conditions on the function P.

Lemma 3.2. *If for all $\boldsymbol{\xi}$, $H(P) > 0$ and $\Delta P > 0$ then the solution in lemma 3.1 is locally stable.*

Proof. To establish this result, suppose that

$$Q(\boldsymbol{\xi}, t) = \phi t/\epsilon + R(\boldsymbol{\xi}, t) + P(\boldsymbol{\xi}), \tag{21}$$

where R is small. As on the boundary we have $\nabla P = \nabla Q$, it also follows that $\nabla R = 0$. Now, substituting the expression for Q into (19) we get

$$\epsilon(I - \gamma\Delta)R_t + \phi = N(\nabla P + \nabla R)H(P + R)^{1/d}$$

so that, to leading order,

$$\epsilon(I - \gamma\Delta)R_t = \frac{1}{d}N(\nabla P)H(P)^{1/d-1}E(P)R + N'(\nabla P)H(P)^{1/d}\nabla R + \mathcal{O}(R^2), \tag{22}$$

where, if $d = 2$, $E(P)$ is the operator given by

$$E(P)R = P_{\xi\xi}R_{\eta\eta} + P_{\eta\eta}R_{\xi\xi} - 2P_{\eta\xi}R_{\eta\xi},$$

(and if $d = 1$ then $E(P)R = R_{\eta\eta}$). But, as $H(P) > 0$ and $\Delta P > 0$ it follows immediately that $E(P)$ is a uniformly elliptic operator with a negative real spectrum. Furthermore, the operator $G \equiv (I - \gamma\Delta)^{-1}$ is a positive compact operator. It follows immediately that any non-constant function R satisfying (22) must decay to zero. Hence, the solution (and hence the mesh) is locally stable. □

It follows from this calculation that the relaxation time scale T_Q for Q is given by

$$T_Q = \frac{\epsilon}{N \min(\lambda_i)}, \tag{23}$$

where $-\lambda_i$ are the eigenvalues of the operator $GH(P)^{1/d-1}E(P)/d$. A natural geometric condition for generating a mesh for a time-evolving problem is that the evolutionary time scale of the mesh should always be shorter than the underlying time scale T of the physical system. If $T_Q \gg T$ then the mesh cannot evolve rapidly enough and all resolution is lost; on the other hand, if $T_Q \ll T$ then the mesh is evolving too quickly leading ultimately to a very stiff system of evolutionary differential equations. A good example of this phenomenon is described in the paper [7] which looks at meshes to solve the one-dimensional chemotaxis problem. The ideal arrangement is for the mesh to evolve faster than the solution of the PDE but on a comparable time scale, so that there are (time-independent) constants A, B for which

$$AT < T_Q < BT. \tag{24}$$

When posed on a finite domain, the operator $E(P)$ has a simple discrete spectrum with $\min(\lambda_i) = \mathcal{O}(1)$. Condition (24) then becomes

$$AT < \epsilon/N < BT, \tag{25}$$

which will be achieved provided that the monitor function and ϵ are chosen to satisfy the condition

$$\epsilon/N = \mathcal{O}(T). \tag{26}$$

3.4. The evolution of Q when N has a scaling law

We now consider extending the above calculation to a class of time-dependent monitor functions. The case of a general time function N is very hard to analyse, instead we will consider the evolution of a mesh derived from a solution to (19) in which the monitor function is presumed to satisfy a strong scaling law. Suppose that we are considering a function $u(\mathbf{x}, t)$ evolving in the manner described in section 2, so that close to a singularity, which we will assume for the present is located at the origin, i.e. in the centre of Ω_P, the solution $u(\mathbf{x}, t)$ takes the separated form

$$u(\mathbf{x}, t) = U(t)V(\mathbf{x}/L(t)), \tag{27}$$

where $U(t)$ is the solution scale (assumed large) and $L(t)$ the length scale (assumed small). Motivated by this, we will also consider a general scaled monitor function N of the form $N \equiv N(u)$ which takes the separated form

$$N(\mathbf{x}, t) = a(t)b(\mathbf{x}/L(t)) \qquad \text{so that} \quad N(\nabla Q, t) = a(t)b(\nabla Q/L(t)). \tag{28}$$

Whilst this may appear to be a very special form of the monitor, it does describe very well the sort of problems associated with blow-up. Our analysis will immediately dictate the form that the function $a(t)$ must take, so that the mesh can follow solution structures of the form (27).

To proceed, we consider the nature of the corresponding solutions of the parabolic Monge–Ampère equation (19) in this case. If we substitute expression (28) into (19), we obtain

$$\epsilon(I - \gamma\Delta)Q_t = a(t)b(\nabla Q/L(t))H(Q)^{1/d}. \tag{29}$$

Now $\mathbf{x} = \nabla Q$ and close to the singularity $\mathbf{x}$ scales like $L(t)$. We require computational variable $\boldsymbol{\xi}$ to be *independent of scale* so that the numerical calculations are not affected by the scale of the solution. It follows that Q must scale in the same manner as $\mathbf{x}$ and hence to have a length scale of $L(t)$. Thus (locally)

$$Q \sim LS(\boldsymbol{\xi}, t),$$

where the function S is slowly evolving in time and in space. It follows immediately that

$$b(\nabla Q/L(t)) \sim b(\nabla S) \qquad \text{and} \qquad H(Q)^{1/d} \sim LH(S)^{1/d}.$$

The simplicity of the latter expression is indeed the motivation for using the function $H^{1/d}$ in (19) rather than H. Following this calculation, we see that the left-hand side of (29) scales in time as L_t and the right-hand side scales in time as $a(t)L(t)$. The two sides have the same time scaling (and thus the mesh will be expected to evolve at the same rate as the solution) provided that

$$a(t) \sim L_t/L. \tag{30}$$

Suppose now that a singularity forms as $t \to T$ for some fixed T. As discussed in section 2, the two most commonly observed forms of $L(T)$ in blow-up-type problems are

$$L(t) = (T-t)^{\alpha} \qquad \text{(polynomial scaling)} \tag{31}$$

and

$$L(t) = (T-t)^{\alpha}|\log(T-t)|^{\kappa} \qquad \text{(approximate polynomial scaling).} \tag{32}$$

In both cases, we have asymptotically

$$a(t) \sim 1/(T-t), \tag{33}$$

so that $1/a(t)$ is evolving at the 'natural' time scale of the partial differential equation. For all of our subsequent calculations, we will assume that the monitor function N has been chosen so that $a(t)$ satisfies (30) and (33).

We now show that the solutions of (19) with N as in (28), (30) and (33) have the property that close to the singularity they can potentially inherit the scaling dynamics of the underlying solution. This is a useful result, as it means that the resulting mesh will follow the natural coordinates of the solution.

Lemma 3.3. *If N takes the form (28) and L either of the forms (31), (32), then the parabolic Monge–Ampère equation (19) admits solutions of the form*

$$Q(\boldsymbol{\xi}, t) = L(t)S(\boldsymbol{\xi}), \qquad \textit{for polynomial scaling} \tag{34}$$

or

$$Q(\boldsymbol{\xi}, t) = L(t)[S(\boldsymbol{\xi}) + \mathcal{O}(1/|\log(T-t)|)], \qquad \textit{for approximate polynomial scaling} \tag{35}$$

where the function $S(\xi)$ satisfies the (time-independent) differential equation

$$-\delta(I - \gamma\Delta)S = b(\nabla S)H^{1/d}(S), \qquad |\nabla S| \to \infty \quad \textit{as} \quad \boldsymbol{\xi} \to \partial\Omega_{\mathrm{C}}, \tag{36}$$

for a positive constant δ, which is small if ϵ is small.

Proof. Suppose that we pose a solution of the form

$$Q = L(t)S(\boldsymbol{\xi}, t).$$

Substituting into equation (19) and using the scaling properties of H, we have

$$\epsilon(I - \gamma\Delta)[L_t S + LS_t] = a(t)b(\nabla S)LH^{1/d}(S).$$

Dividing by $a(t)L(t)$, we have

$$\epsilon(I - \gamma\Delta)[SL_t/aL + S_t/a] = b(\nabla S)H^{1/d}(S).$$

For the case of a pure power law scaling for L, we have $L_t/L = -\alpha/(T-t)$ and if we assume that $a(t) = C/(T-t)$ for some constant $C > 0$, it follows that there exists a constant $\delta > 0$ for which

$$(I - \gamma\Delta)[-\delta S + \epsilon(T-t)S_t/C] = b(\nabla S)H^{1/d}(S),$$

where $\delta = \epsilon\alpha/C$ can be made small by taking ϵ to be small.

Evidently, this equation admits a constant solution $S(\xi)$ satisfying the differential equation (36).

The case of approximate polynomial scaling laws is a little more complicated. In this case, we have

$$L_t = -\alpha(T-t)^{\alpha-1}|\log(T-t)|^{\kappa} + \kappa(T-t)^{\alpha-1}|\log(T-t)|^{\kappa-1}$$

so that

$$(I-\gamma\Delta)[-\delta S + \epsilon\kappa S/|\log(T-t)| + \epsilon(T-t)S_t/C] = b(\nabla S)H^{1/d}(S),$$

where again $\delta = \epsilon\alpha/C$. If $|\log(T-t)|$ is large, we can thus develop an asymptotic series for $S(\boldsymbol{\xi}, t)$ of the form $S(\boldsymbol{\xi}) + \mathcal{O}(1/|\log(T-t)|)$ where $S(\boldsymbol{\xi})$ satisfies (36).

Finally, we consider the boundary conditions for $S(\boldsymbol{\xi})$. The condition $Q = L(t)S$ implies that $\nabla S = \nabla Q/L(t)$. Hence, if ∇Q is of order unity we must have ∇S unbounded in the limit $L(t) \to 0$. Whilst looking rather strange, this condition is very common when meshing around a singularity. It simply states that on the length scale of the singularity, the boundary of the domain looks very far away. □

We now compare the solutions of equation (19) with those of the underlying scaled Monge–Ampère equation. In this case, we have

$$\phi(t) = N(\nabla P)H(P)^{1/d}.$$

Now,

$$\phi^d = \int_{\Omega_P} N^d \,\mathrm{d}\mathbf{x} = a(t)^d \int_{\Omega_P} b^d(\mathbf{x}/L)\,\mathrm{d}\mathbf{x} = D^d a(t)^d L(t)^d,$$

where D is an appropriate constant. It follows that

$$\phi = Da(t)L(t) \sim L/(T-t). \tag{37}$$

Hence, P satisfies the equation

$$DL(t) = b(\nabla P/L)H(P)^{1/d}.$$

This equation has a solution $P = L\bar{S}$ where $\bar{S}$ satisfies

$$D = b(\nabla\bar{S})H(\bar{S})^{1/d}.$$

Now we compare this to the equation satisfied by $Q = LS$ for which

$$-\delta(I-\gamma\Delta)S - b(\nabla S)H^{1/d}(S) = 0.$$

Setting $\hat{S} = \bar{S} - D/\delta$, we have

$$\begin{aligned}-\delta(I-\gamma\Delta)\hat{S} - b(\nabla\hat{S})H^{1/d}(\hat{S}) &= D - \delta(I-\gamma\Delta)\bar{S} - b(\nabla\bar{S})H^{1/d}(\bar{S}) \\ &= -\delta(I-\gamma\Delta)\bar{S} = \mathcal{O}(\delta).\end{aligned}$$

Provided that ϵ is small, and hence $\delta\Delta_\xi\bar{S}$, $\delta\bar{S}$ are small, we see immediately that S is closely approximated by the function $\hat{S}$ so that ∇S and $\nabla\bar{S}$ differ by $\mathcal{O}(\delta)$. Hence, the mesh derived from the potential satisfying the parabolic Monge–Ampère equation is δ-close to that given by the solution of the Monge–Ampère equation.

The consequence of these results is that the mesh obtained from solving the parabolic Monge–Ampère equation will be expected to follow closely the natural coordinates of the underlying equation. Discretizing the solution in these coordinates, we expect to see very similar solution profiles in the computational variables independent of the scale of the physical solution. We show that this occurs in the numerical experiments reported in section 5.

3.5. Time regularization and the determination of suitable monitor functions

It is very convenient for all numerical calculations to introduce a time regularization of both the partial differential equation (1) and the parabolic Monge–Ampère equation (19). As constructed, for a problem developing a singularity at time T, equation (19) takes the form

$$\epsilon(I-\gamma\Delta)Q_t = \frac{C}{(T-t)}b(\nabla Q/L)H(Q)^{1/d}$$

and the partial differential equation takes the form

$$u_t = \Delta_x u + f(u).$$

For problems of the separable form considered in section 2, we can assume that as $t \to T$ the function $f(u)$ scales as $u/(T-t)$ and that the natural time scale of the evolution is of the order of $T-t$. It follows by construction that both the parabolic Monge–Ampère equation and the underlying partial differential equation evolve at the same rate. This allows us to time regularize the equations by introducing a *fictive time* τ through the Sundman transformation. For this particular calculation, it is natural to set

$$\frac{\mathrm{d}t}{\mathrm{d}\tau} = (T-t) \qquad \text{so that} \quad \tau = -\log(T-t) \to \infty \quad \text{as} \quad t \to T. \tag{38}$$

The parabolic Monge–Ampère equation and the underlying partial differential equation then both regularize to

$$\epsilon(I-\gamma\Delta)Q_\tau = b(\nabla Q/L)H(Q)^{1/d}, \qquad u_\tau = (T-t)(\Delta_x u + f(u)). \tag{39}$$

The purpose of this transformation is to allow events which happen over a short time scale in t to be computed over a regular time scale in τ. Observe that $\tau \to \infty$ as $t \to T$.

More generally, suppose that we have a general monitor function $N(u)$ (to scale the spatial variables) and a scaling function $\mathrm{d}t/\mathrm{d}\tau = g(u)$ in the Sundman transformation to scale the temporal variables. We now seek to solve a partial differential equation invariant under the scaling

$$t \to \lambda t, \qquad x \to \lambda^\alpha x, \qquad u \to \lambda^{-\beta} u.$$

Under this transformation, the parabolic Monge–Ampère equation transforms to

$$\epsilon(I-\gamma\Delta)Q_t/\lambda = N(\lambda^{-\beta}u)H(Q)^{1/d}.$$

The right- and left-hand sides balance, and as a consequence the parabolic Monge–Ampère equation is invariant under the same scaling transformations as the original partial differential equation, provided that the function N satisfies the functional equation

$$N(\lambda^{-\beta}u) = N(u)/\lambda. \tag{40}$$

If this holds, then we expect the same *spatial* computational variables to describe the solution, independent of scale. Observe that this condition is satisfied if $N(u) = u^{1/\beta}$. The usual solution form with the scaling law described has $u = (T-t)^{-\beta}V(\mathbf{x}/L)$ so that N takes the form

$$N(u) = \frac{1}{(T-t)}b(\mathbf{x}/L),$$

which is precisely the expression considered in the previous subsection. Similarly, we aim to regularize the whole system in time by using the Sundman transformation $\mathrm{d}t/\mathrm{d}\tau = g(u)$. This transforms to $\lambda\mathrm{d}t/\mathrm{d}\tau = g(\lambda^\beta u)$. The two sides balance, so that in this case the same *temporal* computational variable can be used independent of scale, provided that

$$g(\lambda^{-\beta}u) = \lambda g(u). \tag{41}$$

In general, we will assume that N and g are initially chosen to satisfy (40) and (41) (although some smoothing will be introduced presently). The differential equations that we solve to determine the solution and the mesh are

$$\begin{cases} \epsilon(I-\gamma\Delta)Q_\tau = g(u)N(u)H(Q)^{1/d}, & \mathbf{x}=\nabla Q \\ u_\tau = g(u)(\Delta_x u + f(u)), & \\ \dfrac{\mathrm{d}t}{\mathrm{d}\tau} = g(u). & \end{cases} \tag{42}$$

Note that in this formulation neither the blow-up time nor the blow-up location are assumed known.

4. Regularization, implementation and discretization

We now describe how the parabolic Monge–Ampère method in the form (42) is implemented and discretized.

4.1. Regularization and smoothing

To implement the parabolic Monge–Ampère method (19) for blow-up problems, care must be taken to prevent stiffness of the resulting system and to allow for a degree of mesh uniformity. A problem which occurs frequently in the calculation of singularities using r-adaptive mesh methods (with a fixed number of mesh points) is that too many mesh points are placed close to the singularity. This can lead to a loss of resolution of the solution away from the singularity where the monitor function does not indicate they are required. This is certainly true of the sort of problems we are looking at and occurs when the integral ϕ^d of the (unscaled) monitor function M is due almost entirely to the contribution of M in the singularity [10]. As a result, all of the mesh points will be clustered at the singularity and the resulting mesh is non-uniform. This leads to large truncation (and interpolation) errors when a function satisfying the differential equation (1) is discretized on such a mesh (see, for example, [8]). To avoid this, we introduce a regularization due to McKenzie [3] which has the two features that it does not affect the scaling properties of the mesh close to the singularity described in section 3 and it also ensures that a substantial fraction of the mesh points is placed away from the singularity. Suppose that, as before $\int_{\Omega_P} M\,\mathrm{d}\mathbf{x} = \phi^d$, we replace M by the regularized monitor function $\bar{M}$ and the scaled function N by the regularized function $\bar{N}$ given simply by

$$\bar{M} = M + \phi^d \qquad \text{so that} \quad \bar{N} = (N^d + \phi^d)^{1/d}. \tag{43}$$

Note that

$$\int_{\Omega_P} \bar{M}\,\mathrm{d}\mathbf{x} = 2\phi^d. \tag{44}$$

For the type of problem we are considering, we will assume that $M \gg \phi^d$ inside the singularity and $M \ll \phi^d$ away from the singularity.

For example, consider the functions of the previous section in which

$$N(\mathbf{x}) = b(\mathbf{x}/L)/(T-t),$$

so that from (37)

$$\phi = \mathcal{O}(L/(T-t)).$$

For a typical problem with a singularity, $b(\mathbf{x}/L)$ is of order 1 close to the singularity and then rapidly decays away from the singularity. An example which leads to exactly this type of

behaviour will be given in the next section. It follows that if $L \ll 1$ and $b(\mathbf{x}/L) = \mathcal{O}(1)$ then $N \gg \phi$. Similarly, if b decays sufficiently rapidly for $\mathbf{x}/L$ large then away from the singularity $\phi \gg N$.

As close to the singularity $N \gg \phi$ it follows that in this range $\bar{N}$ is asymptotically close to N, so if we replace N by $\bar{N}$ the analysis of the previous section still applies and the mesh will scale with $L(t)$ for points close to the singularity.

Furthermore, suppose $A_S \subset \Omega_C$ is the subset of the computational space mapped to the (very small) subset $S \subset \Omega_P$ close to the singularity and $A_O \subset \Omega_C$ is the subset of the computational space mapped to the (much larger) subset $O \subset \Omega_P$ outside the singularity. Then, replacing M by $\bar{M}$ in expression (8) we have

$$2\phi^d \int_{A_S} \mathrm{d}\boldsymbol{\xi} = \int_S \bar{M}\,\mathrm{d}\mathbf{x} \approx \int_S M\,\mathrm{d}\mathbf{x} = \phi^d$$

and

$$2\phi^d \int_{A_O} \mathrm{d}\boldsymbol{\xi} = \int_O \bar{M}\,\mathrm{d}\mathbf{x} \approx \int_O \phi^d\,\mathrm{d}\mathbf{x} = \phi^d.$$

It follows that both A_S and A_O have area approximately half of that of the computational space. Accordingly, half of the computational mesh points are mapped into the region around the singularity and half are mapped elsewhere. This allows for good resolution of the solution of (1) both inside and outside the singularity. For all subsequent calculations, we will assume that the monitor function has been normalized in this manner.

For practical calculations it has also been found necessary to introduce a degree of *smoothing* into the scaled monitor function $\bar{N}$ (see, for example, [14]). If we assume that $\bar{N}$ is given at a series of discrete points to give values $N_{i,j}$, the smoothing is done by taking a weighted average of $N_{i,j}$ and several of its nearest neighbours.

4.2. Spatial discretization in two dimensions

For all calculations we will assume that the computational domain Ω_C together with its boundary is divided up into a uniform constant mesh with spacing $\mathrm{d}\xi = 1/(N-1)$ giving N^2 mesh points (ξ_i, η_j) with the mesh potential $Q_{i,j}(t)$ given at each such point. These are in turn mapped to spatial mesh points $x_{i,j}(t) = (x_{i,j}(t), y_{i,j}(t))$ at which there is a discrete solution value $U_{i,j}(t)$ so that

$$U_{i,j}(t) \approx u(x_{i,j}(t), y_{i,j}(t), t).$$

As the mesh points are moving, we must introduce an additional Lagrangian term into equation (1) so that we solve the equation

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \Delta_x u + f(u) + \nabla_x u \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}. \tag{45}$$

This equation is solved in parallel with the parabolic Monge–Ampère equation

$$\epsilon(I - \gamma\Delta)Q_t = \bar{N}(u(\mathbf{x}))H^{1/2}(Q), \qquad \mathbf{x} = \nabla Q. \tag{46}$$

Equations (45), (46) are then discretized in the *computational space*, by recasting the Laplacian in (45) into computational variables. In particular, the Hessian $H(Q)$ is calculated by using a standard uniform 9-point stencil. The Neumann boundary conditions on Q are included by introducing ghost variables on the boundary and including these in the calculation of the stencil. In this calculation, the values of $(Q_{i,j}(t))_t$ are first calculated from (46) by discretizing the right-hand side and then inverting the smoothing operator $\epsilon(I - \gamma\Delta)$. To do this latter calculation, we exploit the fact that all calculations are made on a uniform rectangular mesh.

This allows us to invert the smoothing operator cheaply by using a fast cosine transformation. Having found Q_t we then determine $\mathbf{x}_t = \nabla Q_t$ and from this calculate $\mathrm{d}u/\mathrm{d}t$ from (45). More details of this calculation are given in [9].

4.3. Temporal discretization

The above calculation leads to a set of N^2 ordinary differential equations (in t) for $Q_{i,j}(t)$ and a similar set of N^2 equations for $U_{i,j}(t)$. These are then regularized and expressed as equations in the computational variable τ by using the Sundman transformation. Multiplying each equation by the function $g(u)$ and augmenting the system with the additional equation $\mathrm{d}t/\mathrm{d}\tau = g(u)$ leads to a set of $2N^2 + 1$ equations simultaneously describing the solution, the mesh and the evolution of t in terms of τ. In practice, the Sundman transformation automatically identifies the natural time scale for the evolution of both the physical PDE and the grid. This provides adaptivity based directly on the solution components rather than approximations of time derivatives which are often unreliable as the singularity is approached. This makes the coupled system relatively easy to solve using the standard Matlab solvers `ode15i` or `ode45`. A more careful discussion of this phenomenon is presented in [10].

5. Results

5.1. Overview

In this section, we consider applying this method to the problem

$$u_t = \Delta_x u + u^3, \qquad u|_{\partial\Omega} = 0 \quad \Omega \subset \mathbf{R}^2. \tag{47}$$

We consider both Ω_P to be the unit disc, look for radially symmetric solutions, and also for Ω_P to be the unit square $[-1/2, 1/2]^2$. In the case of the unit disc, we assume that the parabolic Monge–Ampère equation has a radially symmetric solution (in terms of the computational coordinates) with the Hessian given by formula (13). No such restrictions are applied to the unit square.

It is known [17] that for sufficiently large initial data, the solutions of this parabolic partial differential equation blow up in a finite time T at a unique point $\mathbf{x}^*$. It is further known that as $t \to T$ we have

$$\|u\|_\infty = \mathcal{O}((T-t)^{-1/2}), \qquad L(t) = (T-t)^{1/2}|\log(T-t)|^{1/2}. \tag{48}$$

Observe that this equation has an approximately polynomial scaling law for the length scale. The equation has the scaling invariance

$$t \to \lambda t, \qquad \mathbf{x} \to \lambda^{1/2}\mathbf{x}, \qquad u \to \lambda^{-1/2}u, \tag{49}$$

so that $\alpha = \beta = 1/2$. By considering (40) and (41), suitable monitor functions that will lead to scale-invariant discretizations are given by

$$N(u) = u^2, \qquad g(u) = \|N\|_\infty^{-1}. \tag{50}$$

Observe from (48) that, as required, both $N(u)$ and $1/g(u)$ are of order $1/(T-t)$. The asymptotic structure of u is well understood close to the blow-up peak, for both domains. In particular,

$$u(x,t) \sim \frac{1}{\sqrt{2(T-t)}} \frac{1}{\sqrt{1 + |\mathbf{x} - \mathbf{x}^*|^2/6L^2}}, \tag{51}$$

where the asymptotic form of u is essentially independent of the initial conditions. It follows immediately that as $t \to T$

$$\frac{\mathrm{d}t}{\mathrm{d}\tau} = g(u) \sim 2(T-t) \qquad \text{so that} \quad \tau = -\log(T-t)/2, \quad \|u\|_\infty \sim \mathrm{e}^\tau/\sqrt{2}. \tag{52}$$

The natural length-scale $L(t) = (T-t)^{1/2}|\log(T-t)|^{1/2}$ satisfies the simple identity

$$L^2\|u\|_\infty^2 \sim \tau, \tag{53}$$

which we will use as a test for the method.

From (50) and (51) it follows that close to the singularity

$$N \sim \frac{1}{2(T-t)} \frac{1}{(1+|\mathbf{x}-\mathbf{x}^*|^2/(6(T-t)|\log(T-t)|))}.$$

Observe that N varies from $\mathcal{O}(1/(T-t))$ close to the singularity to $\mathcal{O}(\log(T-t))$ when $|\mathbf{x}-\mathbf{x}^*| = \mathcal{O}(1)$. The value of $\phi = \sqrt{\int_{\omega_C} N^2\,\mathrm{d}\mathbf{x}}$ is given by

$$\phi \sim L/(T-t) = \frac{\sqrt{|\log(T-t)|}}{\sqrt{(T-t)}}.$$

As discussed in the last section, $N \gg \phi$ inside the singularity and $\phi \gg N$ away from it. Indeed all of the contributions to ϕ coming from the integration of the values of N^2 in the peak. This emphasizes the point made in section 4 that the monitor function N will concentrate points too much close to the singularity and not place enough points close to the boundary of Ω_C. Accordingly, we regularize N by applying (43) to give the regularized monitor function

$$\bar{N} = (u^4 + \phi^2)^{1/2} \tag{54}$$

so that close to the singularity $\bar{N} = \mathcal{O}(1/(T-t))$ and distant from the singularity $\bar{N} = \mathcal{O}(\sqrt{|\log(T-t)|}/\sqrt{(T-t)})$.

5.2. *Radially symmetric solutions*

It is convenient to test the scaling laws by first looking at radially symmetric solutions which blow up at the origin. Accordingly, we set the scalars x and ξ to be the radial variables in physical and computational space, respectively, and look for solutions of the combined equations

Example 1:
$$\begin{cases} \epsilon(I-\gamma\Delta)Q_t = g(u)\bar{N}(u(\nabla Q))(Q_\xi Q_{\xi\xi}/\xi)^{1/2}, \\ u_t = u_{xx} + \dfrac{1}{x}u_x + u^3, \\ \dfrac{\mathrm{d}t}{\mathrm{d}\tau} = g(u), \\ Q_\xi(0) = 0, \qquad Q_\xi(1) = 1, \\ u_x(0,t) = u(1,t) = 0, \\ t(0) = 0, \end{cases} \tag{55}$$

where $\bar{N}$ is the regularized monitor function given in (54) with ϕ estimated by quadrature. In this problem, blow-up occurs at the origin $x^* = 0$, at which point $u_x = 0$. This equation was discretized in computational space by using a 51-point uniform mesh with $\epsilon = 1e-3$ and $\gamma = 1$. An initial function of $u(x,0) = 4\cos(\pi x/2)$ was chosen. Using this, solutions of magnitude up to 10^{18} could be readily computed in about a minute on a G4 laptop. The solution computed in the computational coordinates is presented in figure 2. In this figure, we see that (owing to the use of the McKenzie regularization) approximately half of the mesh points (the points corresponding to $\xi < 1/2$) are placed inside the peak and half are placed

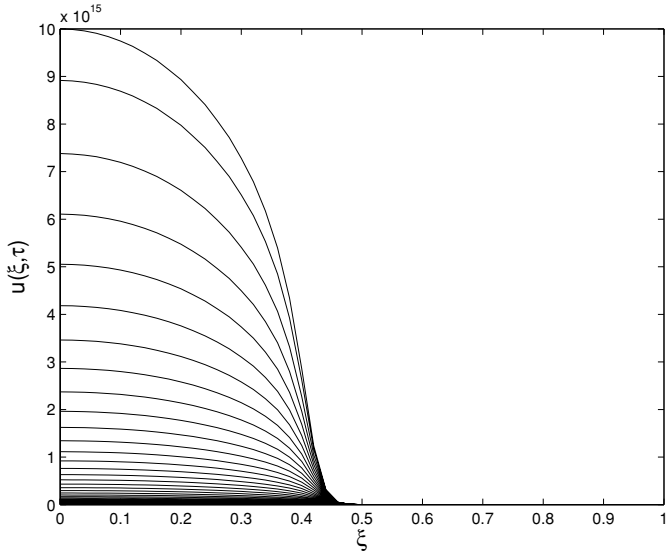

Figure 2. Blow-up profiles in the computational coordinate.

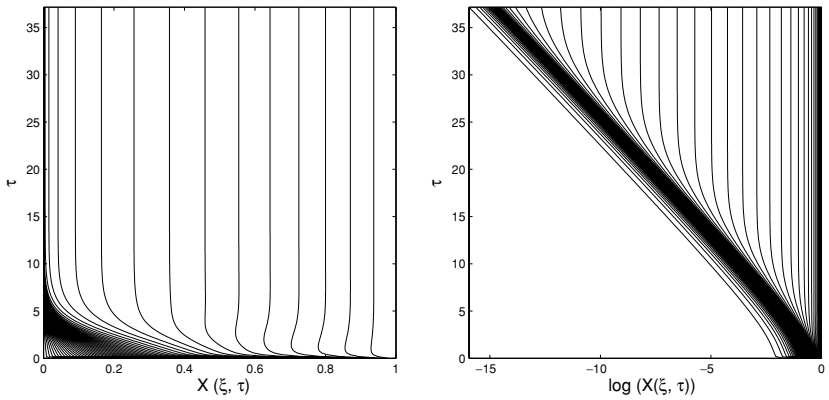

Figure 3. Blow-up mesh. Left: grid in natural coordinates. Near the blow-up time the outer region is static. Right: plotting τ as a function of $\log(x_i)$ at equally spaced values of ξ we see the contraction rate near the origin. In this inner region, the grid cells move at the same rate.

outside the peak. Within the peak, the solution profile in these figures changes little (in shape) during the computation, indicating that the numerical method has identified the correct natural coordinates.

The movement of the (logarithm of the) mesh points at equal values of the computational coordinate is presented in figure 3, in which τ is shown plotted against x_i. Observe again that approximately half of the mesh points move towards the origin. The effects of the McKenzie regularization (using the function $\bar{N}$ rather than N) can be seen clearly. In particular not all of the points move towards the origin, but a substantial proportion remains bounded away, giving good resolution in the bulk of the region away from the singularity.

Observe further that a substantial number of points also resolve the singularity and move towards the origin. Indeed, all of the points closest to the origin move at similar rates. We expect from the analysis of the previous sections that if close to the origin we have **x**

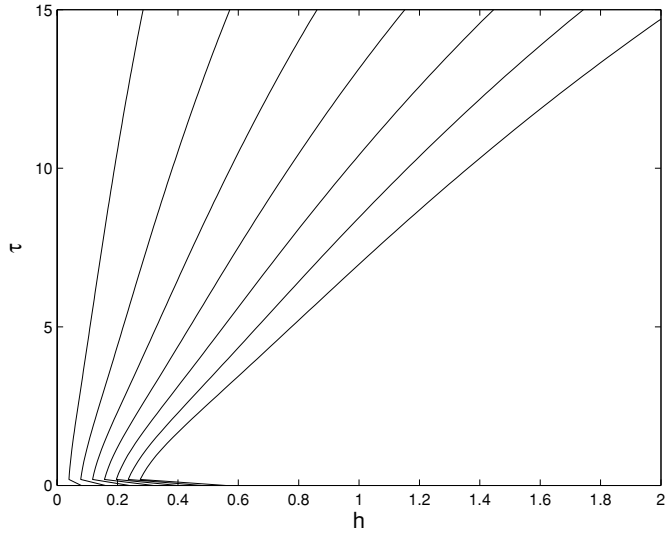

Figure 4. A plot of τ against h taken at equally spaced values of ξ showing the linear dependence.

proportional to $L(t)$, then $x \sim (T-t)^{1/2}|\log(T-t)|^{1/2}$ implies that $\log(x) \sim \log(T-t)/2 = -\tau$. The implication of this calculation is that the asymptotic slope of the graph of $\log(x)$ against τ should be equal to unity, which is indeed observed. As a final test, we calculate the value of the function $h(\xi,\tau) = u(0,t)^2 x(\xi,t)^2$ for different values of ξ. As, close to the origin we expect, from the analysis of section 3 to have $x(\xi,t) = L(t)S(\xi)$ it follows from (51) that $h(\xi,\tau) = S^2(\xi)\tau$. A plot of the form of h is shown in figure 4 showing the correct asymptotic behaviour.

Away from the origin at $\xi \approx 0.45$ there is a sharp transition in the solution where the inner (dynamic) blow-up region matches to the far-field solution which is an essentially static decaying profile. Because the solution is quasi-static here the sharp gradient in the grid poses no problems in solving the system of approximating equations. Indeed, apart from this small transition layer between the inner (blow-up) solution and the outer (quasi-static) solution we have excellent resolution of a solution changing through 15 orders of magnitude. It is possible to show that the relative local interpolation error of the solution is equidistributed in both the inner and outer regions, with a localized larger error in the transition region. A more detailed analysis of this and the associated truncation error is clearly required but beyond the scope of this current paper.

5.3. Solution in the square

We now look at solutions of (47) on the unit square so that we solve the combined equations

Example 2:
$$\begin{cases} \epsilon(I-\gamma\Delta)Q_t = g(u)\bar{N}(u(\nabla Q))(Q_{\xi\xi}Q_{\eta\eta} - Q_{\eta\xi}^2)^{1/2}, \\ u_t = u_{xx} + u_{yy} + u^3, \\ \dfrac{\mathrm{d}t}{\mathrm{d}\tau} = g(u), \\ Q_\xi(-1/2,\eta) = Q_\eta(\xi,-1/2) = -1/2, \quad Q_\xi(1/2,\eta) = Q_\eta(\xi,1/2) = 1/2, \\ u(-1/2,y,t) = u(1/2,y,t) = u(x,1/2,t) = u(x,-1/2,t) = 0, \\ t(0) = 0. \end{cases} \tag{56}$$

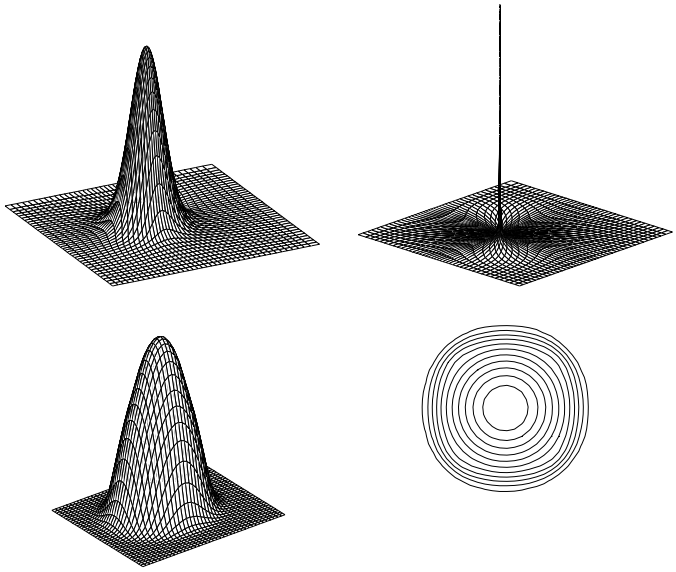

Figure 5. Blow-up in example 2. Top left: initial data in physical variables, $\|u\|_\infty = 2$. Top right: final profile in the physical variables when $\|u\|_\infty = 10^8$. Bottom left : final profile in computational variables. Bottom right: contour plot of final profile in the computational variables.

To highlight the fact that symmetry is not required, the initial solution is chosen to be $u(x, y, 0) = 2\exp(-100((x + 0.05)^2 + (y - 0.05)^2))$ for which we expect the blow-up point $\mathbf{x}^*$ to be close to $(-0.05, 0.05)$. The same monitor functions $\bar{N}$ and g were chosen as in the previous section. In this calculation, the square in the computational domain was discretized using a uniform 31×31 mesh. The McKenzie regularization was again employed together with a 9-point spatial smoother for the resulting monitor function. We took $\epsilon = 0.1$ and $\gamma = 0.1$. With these values no problems were encountered in computing solutions to an amplitude of $\|u\|_\infty = 10^8$ using Matlab on a laptop. Two results of the computations are presented in figure 5, the first close to the initial time and the second much closer to the blow-up time, where we compare the solution in the physical coordinates to that in the computational coordinates.

A striking feature of these figures is that the solution near the blow-up time is very well resolved in the computational coordinates, despite having a very narrow peak in the physical coordinates. This gives a strong indication that the numerical method is correctly identifying the natural spatial coordinates. The computed mesh corresponding to these two solutions is shown in figure 6. Note the high degree of mesh refinement close to the singularity near to the point $(-0.05, 0.05)$. We can also see from this figure that the mesh is regular both close to the singularity and distant from it. Indeed, we can see from the contour plot of the solution in the computational coordinates that the contours close to the peak are very nearly circular and display excellent regularity, again indicating that the correct natural coordinates have been identified.

As a test that the mesh is scaling correctly, we repeat the calculations (51)–(53) from the previous subsection. In particular, we look at the area A of each element. For elements close to the boundary we expect $A = \mathcal{O}(1)$ and for elements close to the singularity $A = \mathcal{O}(L^2)$. Accordingly, defining $h(\tau)$ by

$$h(\tau) = \max(u^2)\min(A)$$

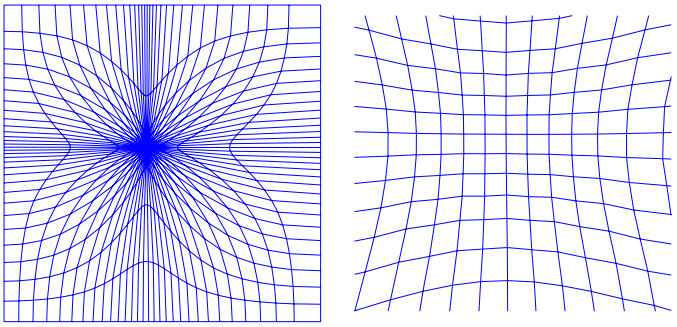

Figure 6. Final grid for example 2. Left: entire grid. Right: detail near the blow-up point. The grid is quite regular in the vicinity of the singularity.

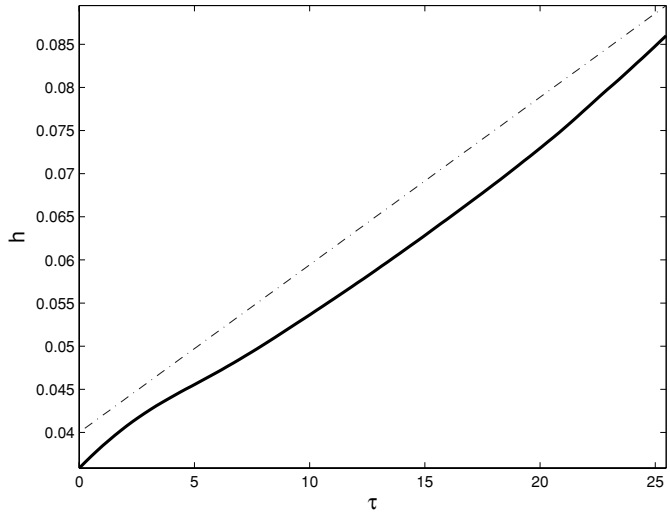

Figure 7. Mesh compression over time showing that $h(\tau) = \max(u^2)\min(A) \sim C\tau$. The dashed curve is linear in τ and is shown for comparison.

we expect to observe that

$$h(\tau) \sim C\tau$$

for some $C > 0$. A plot of h as a function of τ is displayed in figure 7. This shows precisely the linear growth expected, indicating that the mesh is indeed inheriting the scaling properties of the underlying solution.

References

[1] Baines M J 1994 *Moving Finite Elements* (Oxford: Oxford University Press)
[2] Barenblatt G I 1996 *Scaling, Self-similarity, and Intermediate Asymptotics* (Cambridge: Cambridge University Press)
[3] Beckett G and Mackenzie J A 2000 Convergence analysis of finite difference approximations on equidistributed grids to a singularly perturbed boundary value problem *Appl. Numer. Math.* **35** 87–109
[4] Blanes S and Budd C J 2005 Adaptive geometric integrators for Hamiltonian problems with approximate scale invariance *SIAM J. Sci. Comput.* **26** 1089–113

[5] de Boor C 1973 *Good Approximation by Splines with Variable Knots: II (Springer Lecture Note Series* vol 363*)* (Berlin: Springer)
[6] Brenier Y 1991 Polar factorization and monotone rearrangement of vector-valued functions *Commun. Pure Appl. Math.* **44** 375–417
[7] Budd C J, Carretero-Gonzalez R and Russell R D 2005 Precise computations of chemotactic collapse using moving mesh methods *J. Comput. Phys.* **202** 462–87
[8] Budd C J, Huang W and Russell R D 1996 Moving mesh methods for problems with blow-up *SIAM J. Sci. Comput.* **17** 305–27
[9] Budd C J and Williams J F 2006 The parabolic Monge-Ampére equation and adaptivity in arbitrary dimension (in preparation)
[10] Budd C J, Russell R D and Williams J F 2006 Optimal r-adaptive grids (in preparation)
[11] Budd C J, Leimkuhler B and Piggott M D 2001 Scaling invariance and adaptivity *Appl. Numer. Math.* **39** 261–88
[12] Caffarelli L A 1992 The regularity of mappings with a convex potential *J. Am. Math. Soc.* **5** 99–104
[13] Ceniceros H D and Hou T Y 2001 An efficient dynamically adaptive mesh for potentially singular solutions *J. Comput. Phys.* **172** 609–39
[14] Huang W, Ren Y and Russell R D 1994 Moving mesh methods based on moving mesh partial differential equations *J. Comput. Phys.* **112** 279–790
[15] Huang W and Russell R D 1997 Analysis of moving mesh partial differential equations with spatial smoothing *SIAM J. Numer. Anal.* **34** 1106–26
[16] Leiberman G 1996 *Second Order Parabolic Equations* (NJ: World Scientific)
[17] Samarskii A A, Galaktionov V A, Kurdyumov S P and Mikhailov A P 1995 *Blow-Up in Quasilinear Parabolic Equations* (Berlin: de Gruyter & Co)
[18] Sulem C and Sulem P L 1999 *The Nonlinear Schrödinger Equation (Appl. Math. Sci.* vol 139*)* (Berlin: Springer)