MOVING MESH GENERATION USING THE PARABOLIC MONGE–AMPÈRE EQUATION[*]

C. J. BUDD[†] AND J. F. WILLIAMS[‡]

Abstract. This article considers a new method for generating a moving mesh which is suitable for the numerical solution of partial differential equations (PDEs) in several spatial dimensions. The mesh is obtained by taking the gradient of a (scalar) mesh potential function which satisfies an appropriate nonlinear parabolic partial differential equation. This method gives a new technique for performing r-adaptivity based on ideas from optimal transportation combined with the equidistribution principle applied to a (time-varying) scalar monitor function (used successfully in moving mesh methods in one-dimension). Detailed analysis of this new method is presented in which the convergence, regularity, and stability of the mesh is studied. Additionally, this new method is shown to be straightforward to program and implement, requiring the solution of only one simple scalar time-dependent equation in arbitrary dimension, with adaptivity along the boundaries handled automatically. We discuss three preexisting methods in the context of this work. Examples are presented in which either the monitor function is prescribed in advance, or it is given by the solution of a partial differential equation.

1. Introduction. The numerical solution of many nonlinear partial differential equations (PDEs) posed in d-dimensions, using a computational mesh, requires some form of mesh adaptivity in order to generate reliable solutions efficiently. This is due to the difficult nature of the underlying physical phenomena with evolving structures on small length scales possibly manifesting as shocks, singularities, localization, or moving interfaces. Regular, as well as even nonregular, but fixed meshes are often insufficient due to the transient character of such behavior. Adaptive meshes allow enhanced resolution in certain regions of the time-space domain, capturing natural time and length scales and often allowing for the computations to be extended where non-adaptive methods breakdown. There are typically three approaches to mesh adaptivity known as p, h, and r [43]. In p-adaptivity higher-order basis functions are used in certain regions as smoothness permits. A posteriori error estimates typically motivate h-adaptivity approaches in which meshes are locally refined or coarsened until error tolerance criteria are met. The third approach, r-adaptivity, moves a (typically) *fixed number* of nodes, with a constant connectivity structure, so that the nodal density is highest where some monitor of the solution is also high. One purpose of such a calculation is to gain as high an accuracy as possible for a given problem and a given number of mesh points. This can be done statically, regridding whenever given conditions are met, or dynamically through the use of a transformation depending continuously on time. While combinations of the different approaches are possible, dynamic r-adaptivity is the focus of this paper, which aims to describe an efficient r-

[†]Department of Mathematical Sciences, University of Bath, Claverton Down, Bath, BA2 7AY, U.K. (mascjb@bath.ac.uk).
[‡]Department of Maths and Computer Science, Simon Fraser University, Burnaby, BC, Canada V5A (jfw@math.sfu.ca).

adaptive method which can be analyzed in some detail. While not as widely used as h- or p-adaptive methods, r-adaptivity has been used with success in many applications including computational fluid mechanics [43], phase-field models and crystal growth [38], and convective heat transfer [18]. It also has a natural application to problems with a close coupling between spatial and temporal length scales, such as in problems with symmetry, scaling invariance, and self-similarity [4], [12], where the mesh points become the *natural coordinates* for an appropriately rescaled problem.

The approach we describe is a natural extension of the moving mesh PDE methods which have been very successfully implemented in one-dimension by Russell and his co-workers [33], [43]. These methods are all based on finding a mesh which evolves to be close to one which *equidistributes* a suitable monitor M of the properties of the underlying solution, computational errors, and/or of the regularity of the grid. The equidistribution principle will be outlined in later sections, and has the difficulty that while it defines a unique mesh in one-dimension, it fails to do so in higher dimensions, and extra constraints have to be applied to the mesh to give uniqueness. Various approaches have been considered in the literature, varying from constructing the gradient flow of a variational principle [36], the condition that the curl (with respect to $\mathbf{X}$) of the mesh velocity should be zero [17], or the requirement that the map should be a harmonic function [43]. In this paper we will instead impose the condition that (in a suitable norm) the mesh should be *as close as possible to a uniform mesh*, consistent with it equidistributing the monitor function. This idea has a long history in the fields of geometry and fluid mechanics, where it is called the *optimal transport condition*. However, it has certain obvious advantages when used as part of a mesh-generating strategy. From a point of view of being effective for discretizing a PDE, meshes close to uniform ones are less likely to have long thin elements than ones which are not. Furthermore, as we shall show, such meshes can be derived by finding the gradient of a scalar *mesh potential* P. Such meshes then have the property that they are *irrotational*. Determining the mesh then becomes equivalent to solving a (fully nonlinear) Monge–Ampère (MA) type elliptic PDE for the scalar P. Thus it is then solved in parallel with solving the underlying PDE. In practice solving the Monge–Ampère equation is hard, and we propose instead a relaxation method to find P which is a generalization to d dimensions of the one-dimensional moving mesh PDE (MMPDE) approach to generating a mesh described in [33]. For our system this reduces to the problem of finding the solution to a parabolic Monge–Ampère (PMA) equation (PMA) which closely resembles the problem of evolving a Gauss mean-curvature system. In this paper we will derive a parabolic Monge–Ampère equation, and will analyze the properties of its solution and the resulting mesh. In particular the strong structure that the optimal transport condition places on the mesh means that we can make much more precise analytic estimates of its behavior than are possible with other schemes. Indeed, one of the most difficult theoretical aspects of r-adaptive methods for $d > 1$ is in establishing that the mapping remains bijective for all time. In this paper we show rigorously that the PMA method not only generates bijective mappings for all time but that the stationary state is exponentially stable in time, and if the solution changes slowly, then the mesh generated by PMA follows its structure faithfully. We shall also show how the skewness of the resulting mesh can be determined in terms of P and apply this to a problem of meshing a front.

The remainder of this paper is as follows. In section 2 we describe the basic theory behind r-adaptive methods based upon equidistribution. In section 3 we develop the theory behind moving mesh PDE method in one-dimension. In section 4 we look at the optimal transport condition and show how this leads to the MA equation for

the mesh. In section 5 we look at PMA which is a relaxed form of MA and prove a number of results on the existence and regularity of the meshes that it generates. In section 6 we compare the PMA method with other mesh-generating algorithms. In section 7 we show how the method can be implemented. Finally, in section 8 we show a number of applications of this method to generating moving meshes, both where the monitor function is prescribed in advance, and also where it depends upon properties of the solution u of the associated PDE. In particular we look at examples with moving fronts, and one with finite time blow-up governed by a scale invariant PDE.

2. r-adaptivity and the equidistribution principle.

2.1. r-adaptivity. Suppose that we have a PDE with solution $u(\mathbf{X}, t)$ which is posed in a *physical* domain $\Omega_P \subset R^d$ with independent spatial variable $\mathbf{X} \in R^d$. (The theory developed in this paper applies for any $d \geqslant 1$ but for convenience we will generally consider the case of $d = 2$.) Conceptually, an r-adaptive approach to generating a moving mesh continuously maps a suitable *computational* space Ω_C into Ω_P. To do this we presume that a *computational coordinate* $\boldsymbol{\xi} \in R^d$ is continuously mapped to the physical coordinate so that

$$\mathbf{X} \equiv \mathbf{X}(\boldsymbol{\xi}, t).$$

Thus the r-adaptive method is implementing a coordinate change prior to making any discretization. The basis of the r-adaptive approach is that a *fixed* set of mesh points (with fixed connectivity) in Ω_C is moved by this map to a moving set of points in Ω_P where the solution is developing interesting structure. For simplicity we will often assume that Ω_C is a cuboid with unit volume so that

$$(1) \qquad \int_{\Omega_C} d\boldsymbol{\xi} = 1.$$

Throughout this paper we will define coordinates in the computational domain using *greek* letters $\boldsymbol{\xi} = (\xi, \eta, \dots)$ and in the physical domain by roman letters $\mathbf{X} = (X, Y, \dots)$. We will usually assume that the computational domain is meshed by a uniform mesh with N^d cells; see Figure 1. However, there is always complete freedom in choosing an appropriate mesh for the computational domain, and in certain cases (such as when important behavior occurs close to boundaries) it may be useful to consider a nonuniform but fixed mesh in Ω_C which is then mapped to a moving mesh in Ω_P.

Now consider a time-dependent map $\mathbf{F}$ from Ω_C to Ω_P so that

$$\mathbf{X} = \mathbf{F}(\boldsymbol{\xi}, t).$$

We assume further that, for all t, $\mathbf{F} : \partial\Omega_C \to \partial\Omega_P$. For the r-adaptive procedure to be effective, the function $\mathbf{F}$ should be easy to compute and should evolve at least as rapidly as the underlying solution of the PDE being solved. This map takes the uniform mesh in Ω_C to a nonuniform mesh in Ω_P. To avoid *mesh tangling* and a failure of the solution procedure, the map $\mathbf{F}$ must also be 1-1, so that if

$$(2) \qquad J(\boldsymbol{\xi}, t) = \det\left(\frac{\partial \mathbf{X}(\boldsymbol{\xi}, t)}{\partial \boldsymbol{\xi}}\right),$$

then $J > 0$. In general mesh tangling can be a significant problem with the implementation of r-adaptive methods. However, we will show that the condition (2) is
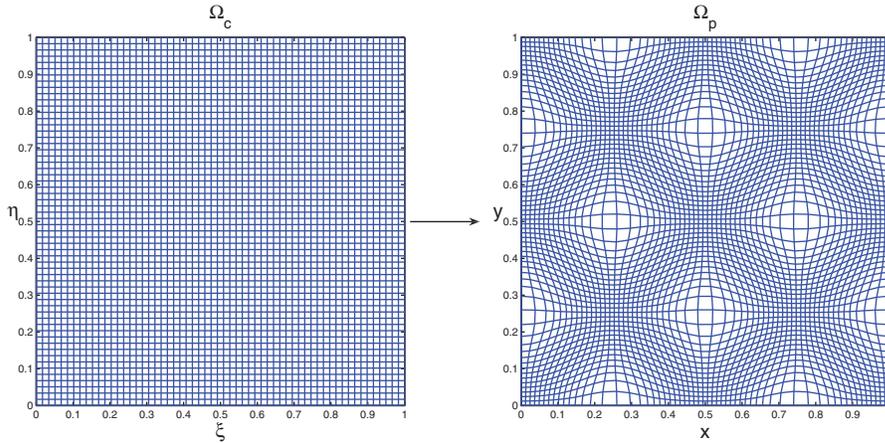
FIG. 1. *A generic mapping from the computational to physical space.*

automatically satisfied by the maps constructed using the parabolic Monge–Ampère algorithm, and will show a series of computations of meshes in which no tangling is observed.

2.2. Maps which equidistribute a monitor function. We will construct all of the maps for the r-adaptive procedure by using the equidistribution principle. The idea of equidistribution is a powerful one for identifying a possible map and goes back to de Boor [25]. To implement it we introduce a (time-dependent) scalar Stieltjes measure $M(\mathbf{X}, t)\, d\mathbf{X}$ into the physical domain. Here the scalar function $M(\mathbf{X}, t) > 0$ is called the *monitor function*, which is presumed to be large in those regions of Ω_P where the mesh points need to be clustered, so that the *density* of the mesh points is large in these regions. This may occur, for example, in the neighborhood of a solution singularity or a sharp front. This monitor function is often defined indirectly via the solution so that we might have $M(\mathbf{X}, t) \equiv M(\mathbf{X}, u(\mathbf{X}, t), \nabla u(\mathbf{X}, t), \ldots, t)$. To implement the equidistribution strategy we introduce an arbitrary nonempty set $\mathbf{A} \subset \Omega_C$ in the computational domain, with a corresponding image set $\mathbf{F}(\mathbf{A}, t) \subset \Omega_P$. The map $\mathbf{F}$ *equidistributes the respective monitor function M* if the Stieltjes measure of $\mathbf{A}$ and $\mathbf{F}(\mathbf{A}, t)$ normalized over the measure of their respective domains are the same. This implies that

$$(3) \qquad \frac{\int_{\mathbf{A}} d\boldsymbol{\xi}}{\int_{\Omega_C} d\boldsymbol{\xi}} = \frac{\int_{\mathbf{F}(\mathbf{A}, t)} M(\mathbf{X}, t)\, d\mathbf{X}}{\int_{\Omega_P} M(\mathbf{X}, t)\, d\mathbf{X}}.$$

It follows from a change of variables that

$$(4) \qquad \frac{\int_{\mathbf{A}} d\boldsymbol{\xi}}{\int_{\Omega_C} d\boldsymbol{\xi}} = \frac{\int_{\mathbf{A}} M(\mathbf{X}(\boldsymbol{\xi}, t), t)|J(\boldsymbol{\xi}, t)|\, d\boldsymbol{\xi}}{\int_{\Omega_P} M(\mathbf{X}(\boldsymbol{\xi}, t), t)\, d\mathbf{X}}.$$

As the set $\mathbf{A}$ is arbitrary, the map $\mathbf{F}(\boldsymbol{\xi}, t)$ must (for all $(\boldsymbol{\xi}, t)$) obey the identity

$$(5) \qquad M(\mathbf{X}(\boldsymbol{\xi}, t), t)|J(\boldsymbol{\xi}, t)| = \theta(t) \quad \text{where} \quad \theta(t) = \frac{\int_{\Omega_P} M(\mathbf{X}(\boldsymbol{\xi}, t), t)\, d\mathbf{X}}{\int_{\Omega_C} d\boldsymbol{\xi}}.$$

We shall refer to (5) as the *equidistribution equation* and it must always be satisfied by the map $\mathbf{F}(\boldsymbol{\xi}, t)$. The equidistribution principle has different consequences in one-dimension from higher dimensions. In one-dimension it (together with boundary conditions) uniquely defines the map $\mathbf{F}(\boldsymbol{\xi}, t)$. However, uniqueness is lost in higher dimensions. Informally this is because there is a unique interval (up to translation) of prescribed length in one-dimension, but there are an uncountable number of sets of prescribed area in higher dimensions. Thus the equidistribution principle needs to be augmented with additional conditions if it is to be applied in dimension $d > 1$.

2.3. Choice of the monitor function. The choice of an appropriate monitor function M is difficult, problem dependent, and the subject of much research. We do not consider this in detail here but give a brief review of various choices used for certain problem classes. The function M can be determined by a priori considerations of the geometry or of the physics of the solution. An example is the generalized solution arc-length given by

$$(6) \qquad M = \sqrt{1 + c^2 |\nabla_x u(x)|^2} \quad \text{or alternatively} \quad M = \sqrt{1 + c^2 |\nabla_\xi u(x(\xi))|^2}.$$

The first of these is often used to construct meshes which can follow moving fronts with locally high gradients [48], [31]. In [18] the second (with u being the temperature) was used sucessfully to resolve small scale singular structures in Boussinesq convection. It is also common to use monitor functions based on the (potential) vorticity, or curvature, of the solution [5], and these have been used in computations of weather front formation [12], [13], [47]. In certain problems, moving fronts are associated with changes in the physics of the solution. An example is problems with phase changes, where the phase front occurs at those points $(x_m)_i$ at a temperature $T = T_m$. In such cases it is possible to construct meshes which resolve behavior close to the phase boundary by using the monitor function $M = a/\sqrt{b|x - x_m| + c}$ where $|x - x_m| = \min |x - (x_m)_i|$ [38]. Alternatively, M can be linked to a posteriori estimates of the solution error. An example of such, in the context of a piecewise linear finite element approximation u_h to a function u, is $M = \sqrt{1 + \alpha \zeta^2}$, where

$$(7) \qquad |u - u_h|^2_{1, \Omega_P} \sim \zeta^2(u_h) \equiv \sum_{\text{l: interior edge}} \int_l [\nabla u_h . n_l]^2_l \, dl$$

and $[.]_l$ is the jump in the computed solution along the element edges. This monitor function is used by Tang [43] to adaptively compute solutions to the Navier–Stokes equations with thin shear layers and/or high Mach numbers. Similarly, in a series of papers [27], [28], [29], [30], [31], Huang explicitly considers monitor functions which are designed to control the regularity, alignment, and quality of the mesh. These include monitor functions which are based as estimates of the interpolation error of the computed solution. Finally, it is sometimes possible in the case of PDEs with *strong scaling structures* (such as problems related to combustion and gas dynamics) to find suitable monitor functions which give meshes that reflect the natural scales of the problem [8]. We give an example of such in section 7, looking at a PDE which has solutions which blow up in a finite time. In this case we need a fine mesh when the solution is large, and take $M(u) = \sqrt{a^2 + b^2 u^{2p}}, \quad p > 0$.

3. Equidistribution and r-adaptivity in one-dimension.

3.1. Existence and uniqueness of the mesh. In one-dimension, the equidistribution condition (5) uniquely defines the map $\mathbf{F}$. To see this, we set $\Omega_C = \Omega_P =$

$[0, 1]$ and pose (5), enforcing the condition of mapping boundaries to boundaries. This gives

$$(8) \qquad M(X(\xi, t), t) \, X_\xi = \theta(t), \quad X(0) = 0, \quad X(1) = 1.$$

Integrating both sides with respect to ξ we have

$$(9)$$
$$N(X, t) = \theta(t) \, \xi \quad \text{where} \quad N(X, t) = \int_0^\xi M(X(\xi, t), t) |X_\xi| \, d\xi = \int_0^X M(X', t) \, dX'.$$

As $M > 0$ it follows that the function $N(X, t)$ is monotone increasing in X; thus each value of ξ is associated with a unique X. For a *given* function $M(X, t)$ (9) is a nonlinear function of X which must be solved at the same time as solving an underlying time-dependent PDE. It is in general hard to do this, especially when also solving the underlying PDE, and various strategies have been devised to make the problem easier.

3.2. Relaxation and MMPDES. One approach which will motivate the calculation of meshes in higher dimensions, and couples naturally to the solution of time-dependent PDEs, is to use relaxation. This is based on the idea that it is not *necessary* to solve (9) *exactly* in order to generate a mesh which is effective for the computation of the solution of the underlying PDE. Instead we can solve a relaxed version of the first equation of (8). As calculating the integral of M over Ω_P is inconvenient, we differentiate (8) with respect to ξ and perform a parabolic relaxation. (Typically this relaxation is in *pseudo-time* when initially generating the mesh and proceeds in *real time* thereafter.) Two possible relaxations are

$$(10) \qquad \varepsilon X_t = (M X_\xi)_\xi \quad \text{and} \quad - \varepsilon X_{t\xi\xi} = (M X_\xi)_\xi \, .$$

Both forms of relaxation are employed in the code MOVCOL where they are referred to as moving mesh PDE MMPDE5 and MMPDE6, respectively. The advantage of this approach is that it can be intialized with an arbitrary mesh which need not satisfy the equidistribution equation (9). Typically we then see a period of rapid evolution over a time-scale $\mathcal{O}(\varepsilon)$ during which the mesh X evolves toward an equidistributed mesh. As the monitor function M evolves with time, it then remains ε-close to the equidistributed mesh for subsequent times. In this evolution the mesh evolves sufficiently fast to track the solution of the underlying PDE provided that *the time-scale for the mesh to evolve under* (10) *is shorter than the evolutionary time-scale of the underlying PDE*. The construction of meshes with this property for the solution of PDEs with a scaling structure is described in [8], [2].

4. Optimal transport and Monge–Ampère equation in R^d.

4.1. Optimal transport. We now consider the question of generating a mesh in dimension $d > 1$ for which equidistribution does *not* uniquely define a mesh. Thus, in addition to the equidistribution condition, additional constraints need to be imposed in order to uniquely define a mesh. Various of these have been suggested; see the review article [43]. From a perspective of solving PDEs using, say, the finite element method, we might seek meshes which avoid long thin triangles and the possibility of mesh tangling arising. Observing that the mesh with the least possibility of either of these occurring is a *uniform* mesh, then we propose that a good equidistributed mesh is one which is as close as possible to a uniform mesh. There are various ways to

define this, but a simple one is to use the L_p measure of distance, so that we define an *optimal* equidistributed mesh to be one closest in the L_p norm to a uniform mesh, i.e., one which minimizes the distance that the uniformly distributed mesh points in the computational domain have to move to the associated points in the physical domain. We make this precise as follows.

DEFINITION. *An optimally equidistributed mapping* $\mathbf{F}$ *minimizes* I_p, *where*

$$(11) \qquad I_p = \int_{\Omega_C} |\mathbf{F}(\boldsymbol{\xi}, t) - \boldsymbol{\xi}|^p \, d\boldsymbol{\xi},$$

over all invertible mappings $\mathbf{X}(\boldsymbol{\xi}, t) = \mathbf{F}(\boldsymbol{\xi}, t)$ *for which the equidistribution condition* (5) *also holds.*

Here I_p is the *cost of rearranging* the initial uniform mesh into the equidistributed mesh. The scientific study of determining such optimal mappings dates back to Monge in 1781, who considered optimally rearranging a pile of material from one configuration to another. This problem has, in various forms, seen a wide array of interest in the past ten years with applications to such diverse fields as meteorology [23], probability, differential geometry, astrophysics, and the study of porous media equations; see the references in [6] and [26] and the book [46]. Of particular relevance to the present study are the works by Brenier [7], Caffarelli [14], [15], and Benamou and Brenier [6]. It also lies at the heart of the problems of image registration. In the original problem Monge took $p = 1$; the case $p = 2$ corresponds to the Kantorovich (or Wasserstein) distance. Consequently the minimizer $\mathbf{F}$ with $p = 2$ is said to optimally transport the unit measure to M, or, solve the Monge–Kantorovich problem. We will consider the case of $p = 2$ for the remainder of this paper. In this case the class of optimal transported maps can be completely classified by the remarkable results in the papers [7] and [15]. These can be summarized in the following theorem which makes the numerical study of optimally transported systems possible.

THEOREM 4.1 (Brenier, Cafferelli [7, 15, 46]). *There* exists *a* unique *optimal mapping* $\mathbf{F}(\boldsymbol{\xi}, t)$ *satisfying* (5). *This map has the same regularity as* M. *Furthermore* $\mathbf{F}(\boldsymbol{\xi}, t)$ *is the unique mapping from this class which can be written as the gradient (with respect to* $\boldsymbol{\xi}$) *of a* convex *(mesh) potential* $P(\boldsymbol{\xi}, t)$, *so that*

$$(12) \qquad \mathbf{F}(\boldsymbol{\xi}, t) = \nabla_{\boldsymbol{\xi}} P(\boldsymbol{\xi}, t), \quad \Delta_\xi P(\boldsymbol{\xi}, t) > 0.$$

COROLLARY 4.2. *The map* $\mathbf{F}$ *is* irrotational *so that* $\nabla_\xi \times \mathbf{F} = \mathbf{0}$, *and the Jacobian of* $\mathbf{F}$ *is* symmetric.

4.2. Legendre transformations. The class of mappings which are the gradient of a scalar function $P(\boldsymbol{\xi}, t)$ (or equivalently have symmetric Jacobians) are termed *Legendre transformations* [41]. They have the property that they are *invertible* and that their inverses are also Legendre transformations with a dual potential $R(\mathbf{X}, t)$. Indeed if $\mathbf{F} = \nabla_\xi P$, then $\mathbf{F}^{-1} = \nabla_X R$, $P + R = \mathbf{X} \cdot \boldsymbol{\xi}$. This fact is key to many applications of optimal transport, particularly in the formation of fronts in meteorology [22], [19], [20], and [23], and has been the motivation for some adaptive schemes described in the papers [13], [3]; see also [21], [24]. Many natural maps are examples of Legendre transformations. For example: (a) A translation by $\mathbf{b}$ is given by $P = \mathbf{b} \cdot \boldsymbol{\xi}$. (b) A uniform scaling by a scalar factor Λ is given by $P = \frac{\Lambda}{2} \boldsymbol{\xi}^T \boldsymbol{\xi}$. (c) A linear map described by the action of a symmetric positive definite matrix A is given by $P = \boldsymbol{\xi}^T A \boldsymbol{\xi}$. (d) A tensor product map in two-dimensions $\boldsymbol{\xi} = (\xi, \eta)$ is given by $P(\xi, \eta) = A(\xi) + B(\eta)$ so that $X = A_\xi(\xi), Y = B_\eta(\eta)$.

4.3. The Monge–Ampère equation and the properties of its solution.

4.3.1. The existence of an optimal, regular, equidistributed mesh. It is immediate that if $\mathbf{X} = \nabla_\xi P$, then

$$\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} = H(P),$$

where $H(P)$ is the *Hessian* of P. Additionally, if the measure $M \in W^2(\Omega_P)$ is strictly positive on its supports (assumed to be convex), then the potential $P \in W^2_{loc}(\Omega_C)$ and satisfies, in the classical sense, the Monge–Ampère equation

$$(13) \qquad\qquad M(\nabla_\xi P, t)|H(P)| = \theta(t).$$

Here $|H(P)|$ denotes the determinant of the Hessian matrix of P. This is a famous equation in differential geometry. Solving it defines the map $\mathbf{F}$ *uniquely*. In order to solve it we must specify boundary conditions for the solution. In general applications of mesh generation, we consider bounded domains mapping to bounded domains. We can then prescribe a boundary condition where (13) is supplemented with the condition that the boundary of the computational domain Ω_C must map to the boundary of the physical domain Ω_P. Suppose that the boundary of Ω_C is given by those points $\boldsymbol{\xi}$ which satisfy implicit equation $G_C(\boldsymbol{\xi}) = 0$, and that of Ω_P by those points $\mathbf{X}$ which satisfy $G_P(\mathbf{X}) = 0$. We then have the following (nonlinear) Neumann boundary condition for the solutions of (13):

$$(14) \qquad\qquad G_P(\nabla_\xi P) = 0 \quad \text{if} \quad G_C(\boldsymbol{\xi}) = 0.$$

The existence, uniqueness, and regularity of the solutions of (13, 14) has been well studied [7], [14], [15]. From the point of view of grid generation this puts us into the nice situation of being able to infer properties of the mesh from those of a function P with known regularity. We have the following very important result.

THEOREM 4.3. *If both Ω_C and Ω_P are smooth, convex domains, then (13, 14) has a unique solution (up to an additive constant) which is as regular as the monitor function M. This in turn leads to a unique regular mesh.*

Proof. This follows immediately from the results of Brenier [7] and Caffarelli [14], [15]. ☐

If Ω_C and/or Ω_P are not smooth, then there is a possible reduction of mesh regularity. In the case of solutions which are logical rectangles, this is not severe and applies only at the corners, where it can be shown [24] that $P \in C^3$ and $F \in C^2$. This is sufficient regularity for most applications.

Note. The result as stated applies to convex (and thus simply connected domains). Clearly many problems in adaptivity also arise in nonconvex domains (such as those with reentrant corners). The present study, however, will concentrate on the problems associated with resolving solution features, such as singularities and fronts, which occur interior to the domain.

4.3.2. Boundary behavior. An immediate consequence of Theorem 4.3 is that the solution of the Monge–Ampère equation not only defines a map from Ω_C to Ω_P but also from the boundaries of the respective domains. This is a desirable property from the perspective of mesh generation as we do not have to consider separate equations for the mesh on the boundary. This is in contrast to other methods for mesh generation such as those described in [35] which require separate equations to

describe the mesh on the boundaries. For much of this paper, and in many numerical analysis applications, we will limit ourselves to mappings between fixed logical cubes, taking

$$(15) \qquad\qquad \Omega_C = [0, 1]^d = \Omega_P.$$

The advantage of doing this is that the underlying mesh in the computational domain can be as simple as possible. In particular, if we are considering a map in two-dimensions from a unit square domain to itself (as in the examples discussed in section 7), the appropriate Neumann boundary conditions are

$$(16) \qquad\quad P_\xi = 0, 1 \quad \text{if} \quad \xi = 0, 1 \quad \text{and} \quad P_\eta = 0, 1 \quad \text{if} \quad \eta = 0, 1.$$

4.3.3. Mesh symmetries. Some very desirable properties of the mesh follow immediately from the properties of the Monge–Ampère equation. It is trivial to see that the equation is invariant under translations in (ξ, η). It is also easy to see that (in a similar manner to the Laplacian operator) the Monge–Ampère equation is also invariant under any *orthogonal* map such as a rotation or a reflexion. This is because the Monge–Ampère equation is the determinant of the Hessian which transforms covariantly under such maps. This simple observation implies that (away from boundaries) the meshes generated by solving the Monge–Ampère equation should have no difficulty aligning themselves to structures, such as shocks, which may occur anywhere in a domain and at any orientation. We will see evidence for this in the examples at the end of this paper. (In this sense Monge–Ampère generated meshes should perform better than meshes constrained by the underlying coordinate axes.) It is also immediate that Monge–Ampère equation is also invariant under *scaling transformations* of the form $\xi \to \lambda\xi, \quad \eta \to \mu\eta, \quad P \to \nu P$.

4.3.4. Mesh skewness. The regularity of the mesh generated by this approach gives a useful feature in seeing control in the variation of the element size across the domain. However this does not immediately indicate how appropriate the mesh will be for calculations, such as in the finite element method, where we might seek to avoid long, thin regions. However, it is possible to make some estimates for the resulting skewness of the mesh in terms of the properties of the function P. Consider a two-dimensional problem for which the map from Ω_C to Ω_P has the Jacobian J. A measure for the skewness s of the mesh in Ω_P is given by

$$s = \frac{\lambda_1}{\lambda_2} + \frac{\lambda_2}{\lambda_1} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} - 2 = \frac{\text{trace}(J)^2}{\det(J)} - 2,$$

where λ_1 and λ_2 are the (real and positive) eigenvalues of J. As J is the Hessian of P, it follows immediately that

$$(17) \qquad\qquad s = \frac{\Delta(P)^2}{H(P)} - 2.$$

The value of s could, in principle, be controlled directly by including (17) directly into the monitor function. This is similar to the approach used by Huang [29], [30] in which he took a monitor function which controlled the mesh alignment. Alternatively, the skewness can be estimated in certain cases. One example of this arises in the scale-invariant meshes for the local singularities blow-up problems studied in [8], in

which a sequence of meshes are calculated which close to the singularity take the form $P(\xi, t) = \Lambda(t)\hat{P}(\xi)$ for an appropriate scaling function $\Lambda(t)$. It is immediate that

$$\frac{\Delta(P)^2}{H(P)} - 2 = \frac{\Delta(\hat{P})^2}{H(\hat{P})} - 2$$

so that the skewness of the rescaled mesh is the same as the original. Hence, if an initially uniform mesh is used, then the mesh close to the singularity will retain local uniformity. We will see an example of this in section 8. A further calculation of s in the case of a mesh close to a narrow front is also given in section 8.

4.3.5. Mesh orthogonality. The boundary condition, (16), applied in two-dimensions implies the orthogonality of the grid lines at the boundaries of the square. To see this, consider the bottom boundary of the square in the physical domain, for which $Y = 0$. A grid line Γ which intersects this side is given by

$$\Gamma = \{(X, Y) : 0 \leqslant \eta \leqslant 1, \quad \xi \quad \text{fixed}\}.$$

The tangent to this line in the physical domain is given by $\boldsymbol{\tau} = (X_\eta, Y_\eta)^T$, and, trivially, the tangent to the bottom boundary of the square by the vector $\mathbf{t} = (1, 0)^T$. However, on the bottom boundary, the function P satisfies the identity $P_\eta(\xi, 0) = 0$ and by definition $X = P_\xi$. It follows immediately that when $\eta = 0$,

$$X_\eta(\xi, 0) = P_{\eta\xi}(\xi, 0) = 0,$$

so that on the lower boundary $\boldsymbol{\tau}.\mathbf{t} = 0$. Thus the mesh is orthogonal to the lower boundary. Similar results apply to the other sides as well. The condition of mesh orthogonality at the boundary of the square is often desirable in certain circumstances [45]. However it may cause problems when resolving features, such as fronts, which intersect boundaries at an angle. In such cases it may be useful to refine the mesh close to the boundary, either by introducing a finer computational mesh there, or by locally increasing the value of the monitor function M at mesh points adjacent to the boundary.

Note. This result is special to mappings from the square to itself and does not necessarily follow for other more general boundaries.

5. Parabolic regularization of the Monge–Ampère equation.

5.1. Overview. The Monge–Ampère equation (13) for the mesh potential P, coupled to the boundary conditions (14), is a fully nonlinear elliptic equation whose solution determines a grid for a given $M(\mathbf{X}, t)$. In general $M(\mathbf{X}, t)$ will depend on the solution $u(\mathbf{X}, t)$ of the underlying PDE which is to be solved on the evolving mesh. Suppose that at a time t the solution $u(\mathbf{X}, t)$ is known, we can then determine the mesh $\mathbf{X}(\boldsymbol{\xi}, t)$ by solving the Monge–Ampère equation (13) exactly. However, this is a formidable task as it involves repeated solutions of a fully nonlinear time-dependent PDE. It is also, possibly, an unnecessary task as an accurate solution of the underlying PDE may not necessarily require an exact solution of the mesh equation. An alternative approach, motivated by the discussion of the MMPDEs in section 3, is to introduce a parabolic regularization to (13) so that the gradient of solutions of this evolves toward the gradient of the solutions of (13) over a (relatively) short timescale. This method also couples naturally to the solution of a time-dependent PDE. Accordingly we consider using *relaxation* to generate an approximate solution

of (13) which evolves together with the solution of the underlying PDE. Accordingly we consider a time-evolving function $Q(\boldsymbol{\xi}, t)$ with associated mesh $\mathbf{X}(\boldsymbol{\xi}, t) = \nabla_\xi Q(\boldsymbol{\xi}, t)$ with the property that this mesh should be close to that determined by the solution of the Monge–Ampère equation. To do this we consider a relaxed form of (13) taking the form

$$(18) \qquad\qquad \varepsilon(I - \gamma\Delta)Q_t = (|H(Q)|M(\nabla Q))^{1/d}.$$

(In this expression, and for the remainder of the paper, we shall assume, unless stated otherwise, that all spatial derivatives are taken with respect to the *computational variable* $\boldsymbol{\xi}$.) To find a moving mesh, we start with an initially uniform mesh for which

$$Q(\boldsymbol{\xi}, 0) = \frac{1}{2}|\boldsymbol{\xi}|^2.$$

The function Q then evolves according to (18). In (18) the scaling power $1/d$ is necessary for global existence of the solution. This is because if Q is scaled by a factor $L(t)$, then the term $|H(Q)|$ scales as $L(t)^d$. If M is constant, then (18) without the power law scaling admits a variables separable solution for which $L_t = CL^d$. If $d > 1$, then this equation has solutions which blow up in a finite time. The rescaling prevents this possibility. The operator on the left of this system is a smoothing operator, similar to the operator used by Ceniceros and Hou in [18], which reduces the stiffness of this system when it is discretized. Observe that the term $\theta(t)$ has not been included in (18). It will transpire that this term arises naturally as a constant of integration. Indeed, we note that if $P(\boldsymbol{\xi}, t)$ is a solution of the Monge–Ampère equation with Neumann boundary conditions, then so is $P(\boldsymbol{\xi}, t) + \Lambda(t)$ where the function $\Lambda(t)$ is arbitrary. As we are only interested in the resulting map $\mathbf{X} = \nabla P$ this difference is unimportant. Similarly, we generically expect Q and P to differ by functions of t only, together with perturbations of order ε.

As in the case of the Monge–Ampère equation, the parabolic Monge–Ampère equation is invariant under translations and rotations. It is also invariant under scalings provided that the function M must be chosen with care. This is discussed in [8], [2] and allows the PMA method to perform effectively when solving PDEs which themselves have a scaling invariance.

In one-dimension, differentiating (18) with respect to ξ gives

$$\varepsilon(Q_{\xi t} - \gamma Q_{\xi\xi t}) = \varepsilon(X_t - \gamma X_{\xi\xi t}) = (M(X)X_\xi)_\xi,$$

which is a linear combination of the mesh equations MMPDE5 and MMPDE6 (10).

5.2. Analysis of the solutions of the parabolic Monge–Ampère equation. The advantage of using the parabolic Monge–Ampère equation to generate a mesh is that it does so by solving a single scalar equation for any dimension d. However, for the method to be usable we must establish that (18) has regular solutions, which avoid mesh tangling and which converge to a stable solution of the Monge–Ampère equation.

5.2.1. Regularity and convexity. Given some initial mesh potential P, it follows from results of Cafferelli [14], [15] that (18) has, local in time, smooth solutions. For these to correspond to well-defined transformations (and to avoid mesh crossing) we must show that the map determined from the solution of (18) is bijective. This is equivalent to showing that the Jacobian of the map is bounded away from zero.

This Jacobian is precisely given by $|H(Q)|$. The bijective nature of the map in two-dimensions follows from the lemma below.

LEMMA 5.1. *Suppose that Q satisfies the equation*

$$(19) \qquad Q_t = (M(\nabla Q)|H(Q)|)^{1/2}, \qquad |H(Q)| = Q_{\eta\eta}Q_{\xi\xi} - Q_{\eta\xi}^2$$

for all $t < T$, with $\boldsymbol{\xi} = (\eta, \xi) \in \Omega_C$ and if $|H(Q)| > 0$ and $\Delta Q > 0$ at $t = 0$, and with $M(\boldsymbol{\xi}, t) > 0$ for all $t < T, \boldsymbol{\xi} \in \Omega_C$. Then $|H(Q)| > 0$, and $\Delta Q > 0$ for all $t < T, \boldsymbol{\xi} \in \Omega_C$.

Proof. As $|H(Q)| > 0$ at $t = 0$, there must be a first time t^* at which $|H(Q)|$ takes its minimum value of $|H(Q)| = 0$. We initially consider the case where the minimum point does not lie on the boundary and is interior to the domain so that $\nabla|H(Q)| = 0$ at this point. We shall assume that this is a regular minimum and without loss of generatlity taking the minimum point to be the origin. We can, by taking a suitable rotation and using the fact that the Monge–Ampère equation is invariant under rotations, say that close to this minimum point we have

$$(20) \qquad |H(Q)|(\xi, \eta) = a\xi^2 + b\eta^2, \quad \text{for some} \quad a, b > 0.$$

Now, from the definition it follows immediately that

$$|H(Q)|_t = Q_{\eta\eta}Q_{\xi\xi t} + Q_{\xi\xi}Q_{\eta\eta t} - 2Q_{\eta\xi}Q_{\eta\xi t} \quad \text{and} \quad Q_t = (M(\nabla Q)|H(Q)|)^{1/2}.$$

Let $M^{1/2} = N$ and $|H(Q)|^{1/2} = G$, and then $Q_{\eta\eta t} = N_{\eta\eta}G + 2N_\eta G_\eta + NG_{\eta\eta}$, etc. Now $G = \sqrt{a\xi^2 + b\eta^2}$ so that

$$G_{\eta\eta} = \frac{ab\xi^2}{(a\xi^2 + b\eta^2)^{3/2}}, \quad G_{\xi\xi} = \frac{ab\eta^2}{(a\xi^2 + b\eta^2)^{3/2}}, \quad G_{\eta\xi} = \frac{-ab\eta\xi}{(a\xi^2 + b\eta^2)^{3/2}}.$$

If $r = \sqrt{\eta^2 + \xi^2}$, then, for small r the second derivatives of G are $\mathcal{O}(1/r)$ and the first derivatives of G are $\mathcal{O}(1)$ and $G = \mathcal{O}(r)$. Thus the terms in $Q_{\eta\eta t}$ (and similar) and hence the terms in the expression for $|H(Q)|_t$ are dominated by the second derivatives of G. Hence, to leading order, close to the origin we have

$$|H(Q)|_t = \frac{abN}{(a\xi^2 + b\eta^2)^{3/2}} \left(\eta^2 Q_{\eta\eta} + \xi^2 Q_{\xi\xi} + 2\eta\xi Q_{\eta\xi} \right).$$

But

$$\eta^2 Q_{\eta\eta} + \xi^2 Q_{\xi\xi} + 2\eta\xi Q_{\eta\xi} = (\xi \ \eta) \begin{pmatrix} Q_{\xi\xi} & Q_{\eta\xi} \\ Q_{\eta\xi} & Q_{\eta\eta} \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} = \boldsymbol{\xi}^T H(Q)\boldsymbol{\xi}.$$

Now, suppose that the (real) eigenvalues of $H(Q)$ are λ_1, λ_2. It follows that $|H(Q)| = \lambda_1\lambda_2$ and that $\Delta Q = \text{trace}(H(Q)) = \lambda_1 + \lambda_2$. It follows that initially λ_1 and λ_2 are everywhere positive and that at the time t^* they are positive everywhere apart from at the origin where one or both vanish. Thus $|H(Q)|$ is positive definite at all points apart from at the origin, where it is positive semi-definite. Accordingly at $t = t^*$, at all points close to the origin we have $|H(Q)|_t > 0$ and that $|H(Q)|_t \geqslant 0$ at the origin. (In fact, a careful analysis indicates that $|H(Q)|_t > 0$ at the origin as it is well provided that λ_1 and λ_2 both vanish simultaneously). Thus, from the usual application of the maximum principle it follows that $|H(Q)| > 0$ for all t. As an immediate consequence λ_1 and λ_2 are both positive and hence $\Delta|H(Q)|$ is also positive for all t.

In the case when $|H(Q)|$ vanishes on the boundary of the domain, for $|H(Q)|$ to be positive within the domain it is necessary that $\nabla H \cdot \boldsymbol{\tau} = 0$, where $\boldsymbol{\tau}$ is the tangent vector to the boundary. Furthermore, the boundary is given by the set of points for which $F(\nabla Q) = 0$ for appropriate F. Hence, on differentiating with respect to t we have $\mathbf{n} \cdot \nabla Q_t = 0$, where $\mathbf{n} = \nabla F$ is the normal vector to the boundary. But, as $Q_t = (M|H(Q)|)^{1/2}$ we have $\nabla Q_t = \nabla (M|H(Q)|)^{1/2}$. It follows immediately that if at some point and at $t = t^*$ we have $|H(Q)| = 0$ and $M > 0$, then also at this point we have that ∇Q_t and $\nabla|H(Q)|$ are parallel so that $n.\nabla|H(Q)| = 0$. It follows immediately that $\nabla|H(Q)| = 0$. The result then follows as above. $\qquad\square$

Note. In the case of the more general equation $\varepsilon(I - \gamma\Delta_\xi)Q_t = (M|H(Q)|)^{1/2}$ we have $\varepsilon Q_t = K(M|H(Q)|)^{1/2}$, where K is a positive compact operator. The positivity of K ensures that the various estimates of $|H(Q)|_t \geqslant 0$ do not change in this more general case.

5.2.2. Mesh existence and local stability for a constant in time monitor function. Having shown that (18) has a solution which leads to a regular transformation, we now look at the form of the corresponding map. First we show that if the monitor function M is *constant in time*, then PMA admits a stable solution which generates the same mesh as the solution of the Monge–Ampère equation.

LEMMA 5.2. *Suppose that $M_t = 0$ and the Monge–Ampère equation (13) admits a (steady) convex solution $P(\boldsymbol{\xi})$ with associated map $\mathbf{X}(\boldsymbol{\xi})$ for which $H(P) > 0$, so that P satisfies the equation*

$$M(\nabla P)|H(P)| = \theta,$$

then

(i) *the PMA equation (18) admits a time-dependent solution*

$$(21) \qquad Q(\boldsymbol{\xi}, t) = \frac{\theta^{1/d} t}{\varepsilon} + P(\boldsymbol{\xi})$$

for which $\nabla Q = \nabla P = \mathbf{X}(\boldsymbol{\xi})$.

(ii) *The resulting mesh is locally stable.*

Proof. (i) Suppose that for an initially arbitrary scalar φ and function $P(\boldsymbol{\xi})$ we consider the time-dependent function $Q(\boldsymbol{\xi}, t) = \varphi t/\varepsilon + P(\boldsymbol{\xi})$. Then $\varepsilon(I - \gamma\Delta)Q_t = \varphi$ and $\nabla Q = \nabla P$, $|H(Q)| = |H(P)|$. Thus

$$\varepsilon(I - \gamma\Delta)Q_t - (M(\nabla Q)|H(Q)|)^{1/d} = \varphi - (M(\nabla P)|H(P)|)^{1/d}.$$

Setting $\varphi = \theta^{1/d}$ and P to be a solution of (13) we have that Q satisfies (18) identically.

(ii) To show stability, we set $Q(\boldsymbol{\xi}, t) = \varphi t/\varepsilon + P(\boldsymbol{\xi}) + \delta R(\boldsymbol{\xi}, t)$ with $\delta \ll 1$ and $N = M^{1/d}$. Then

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

so that, to leading order in δ,

$$(22) \qquad \varepsilon(I - \gamma\Delta)R_t = \frac{1}{d}N(\nabla P)H(P)^{1/d-1}E(P)R + N'(\nabla P)H(P)^{1/d}\nabla R.$$

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 (and hence can be locally transformed into the Laplacian). 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. Indeed, convergence onto the steady-state solution is locally exponential. □

Note. The case of constant monitor function M is especially important in the first stage of a moving mesh algorithm. Typically in this stage an initially uniform mesh is evolved in an artificial time to equidistribute a fixed monitor function. Having done this, the mesh and monitor function evolve together in real time.

5.3. Slowly varying monitor functions. More generally, the monitor function $M(\mathbf{X}, t) = M(\nabla P, t)$ will evolve with time. We suppose next that this evolution is relatively slow, so that εM_t is small. Let $P(\boldsymbol{\xi}, t)$ be the solution of (13), with corresponding map $\mathbf{X}(\boldsymbol{\xi}, t)$. We now prove the following.

LEMMA 5.3. *If $\varepsilon|M_t| \ll 1$ and $d\Phi/dt = \theta(t)^{1/d}$, then the parabolic Monge–Ampère equation (PMA) (18) admits a solution $Q(\boldsymbol{\xi}, t)$ such that*

$$(23) \qquad Q(\boldsymbol{\xi}, t) = \frac{\Phi(t)}{\varepsilon} + P(\boldsymbol{\xi}, t) + \mathcal{O}(\varepsilon) \quad and \quad |\nabla Q(\boldsymbol{\xi}, t) - \mathbf{X}(\boldsymbol{\xi}, t)| = \mathcal{O}(\varepsilon).$$

Proof. We consider a solution to the PMA of the form

$$Q(\boldsymbol{\xi}, t) = \frac{\Phi(t)}{\varepsilon} + P(\boldsymbol{\xi}, t) + \varepsilon R(\boldsymbol{\xi}, t).$$

Substituting into (18), using the identity $\Phi_t = \theta^{1/d}$ and assuming that R varies slowly (so that R_t is of the same order as R) we have

$$\theta(t)^{1/d} + \varepsilon(I - \gamma\Delta)P_t + \mathcal{O}\left(\varepsilon^2\right) = M(\nabla P, t)^{1/d}|H(P)|^{1/d}$$
$$+ \varepsilon\left(E(P)R + N'(\nabla P, t)|H(P)|^{1/d}\nabla R\right) + \mathcal{O}\left(\varepsilon^2\right).$$

As P satisfies (13) it follows (on subtracting off the solution of (13) and dividing by ε) that R satisfies the equation

$$E(P)R + N'(\nabla P, t)|H(P)|^{1/d}\nabla R = (I - \gamma\Delta)P_t + \mathcal{O}(\varepsilon).$$

It follows from the uniform ellipticity of E and the assumed slow variation of the function P_t that R is bounded and has bounded derivatives. The results in (23) then follow. □

Hence, provided εP_t (i.e., provided that εM_t) is small, then certain solutions of the PMA equation will remain ε close to the solutions of the Monge–Ampère equation. In the evolution of the solutions of (18) from an initial arbitrary mesh, we expect to see rapid evolution over a short time as the solutions tend toward those of the Monge–Ampère equation, followed by slow evolution thereafter as the solutions stay ε-close to the manifold of equidistributed solutions satisfying (13). This is similar in many respects to the behavior of the solutions of the MMPDES described in section 3. The cases when M_t evolves rapidly are much more subtle and are not considered in this paper. The special case where M has a scaling law behavior are described in the paper [8].

6. Connections of PMA to other moving grid methods. As stated previously, there are many methods which generate meshes in higher dimension. These fall into two categories, methods which determine the location of the mesh points (such as PMA), and those which determine the velocity of the mesh points (such as the moving finite element method). We now present examples of some of these algorithms with direct connections to the method (18).

6.1. The gradient flow equidistribution method. Location-based methods often construct meshes by means of a gradient flow descent of a given functional. The method of Huang and Russell et al. [37], [36] uses a functional defined in the physical coordinates. In two-dimensions this takes the form

$$(24) \qquad I[\boldsymbol{\xi}] = \frac{1}{2} \int_{\Omega_p} \left(\nabla_X \xi^T G_1^{-1} \nabla_X \xi + \nabla_X \eta^T G_2^{-1} \nabla_X \eta \right) \, d\mathbf{X},$$

where G_i are given symmetric positive definite matrices closely related to the monitor function M in (18) (for example, we can take $G_i = M I_d$). Defining a relaxed gradient flow by

$$(25) \qquad \frac{\partial \xi}{\partial t} = -\frac{1}{\tau \sqrt{g_1}} \frac{\delta I}{\delta \xi}, \qquad \frac{\partial \eta}{\partial t} = -\frac{1}{\tau \sqrt{g_2}} \frac{\delta I}{\delta \eta}$$

and inverting gives equations for the mesh:

$$(26) \qquad \frac{\partial \mathbf{x}}{\partial t} = -\frac{\mathbf{x}_\xi}{\tau \sqrt{g_1} J} \left(\frac{\partial}{\partial \xi} \left[+\frac{1}{J g_1} \left(\mathbf{x}_\eta^T G_1 \mathbf{x}_\eta \right) \right] - \frac{\partial}{\partial \eta} \left[\frac{1}{J g_1} \left(\mathbf{x}_\xi^T G_1 \mathbf{x}_\eta \right) \right] \right)$$

$$(27) \qquad -\frac{\mathbf{x}_\eta}{\tau \sqrt{g_2} J} \left(\frac{\partial}{\partial \xi} \left[-\frac{1}{J g_2} \left(\mathbf{x}_\eta^T G_2 \mathbf{x}_\xi \right) \right] + \frac{\partial}{\partial \eta} \left[\frac{1}{J g_2} \left(\mathbf{x}_\xi^T G_2 \mathbf{x}_\xi \right) \right] \right).$$

This system is closed by the addition of additional adaptive boundary conditions. Typically this means the solution of a lower dimensional MMPDE on the boundary to provide Dirichlet conditions for the interior equations (26).

6.2. Geometric conservation laws. The geometric conservation law method [17], [16], [1], [2] is a velocity method based on a time derivative of the equidistribution equation which, when combined with extra conditions, gives equations for $\dot{\mathbf{x}}$. If the monitor function M is normalized to have a constant integral of unity over Ω_P, then integrating the equidistribution equation over an arbitrary set A we have

$$\int_{\mathbf{F}(A,t)} M(\mathbf{X}, t) \, d\mathbf{X} = \int_A d\boldsymbol{\xi}.$$

Differentiating this expression with respect to t, applying Reynold's transport theorem, and allowing the set A to be arbitrary, leads to the conservation law

$$(28) \qquad M_t + \nabla_\mathbf{X} \cdot (M \mathbf{V}) = 0,$$

where the mesh velocity is given by $\mathbf{V} = \mathbf{X}_t$. In one-dimension (28) specifies $\mathbf{V}$ but in higher dimensions further conditions must be added to give $\mathbf{V}$ uniquely. An often used condition is to require $\mathbf{V}$ to be a gradient $\mathbf{V}(\mathbf{X}, t) = \nabla_X \varphi$ for some $\varphi(\mathbf{X}, t)$, so that

$$(29) \qquad M_t + \nabla_\mathbf{X} \cdot (M \nabla_\mathbf{X} \varphi) = 0.$$

If M and M_t are both known, then this elliptic PDE can be solved for φ and hence $\mathbf{V}$ can be determined. The mesh can then be updated by solving the systems $\mathbf{X}_t = \mathbf{V}$. Note that while this equation is linear in φ it is not linear in the computational coordinates. The Monge–Ampère condition $\mathbf{X} = \nabla_{\boldsymbol{\xi}} P$ (or $\mathbf{V} = \nabla_{\boldsymbol{\xi}} P_t$) is closely related to the condition $\mathbf{V} = \nabla_X \varphi$ but as the gradients are with respect to different variables they are clearly distinct.

6.3. The method of Ceniceros and Hou. A mesh generation method due to Ceniceros and Hou [18] (and closely related to the earlier method of Winslow [48]) has been used to compute singular solutions of the Boussinesq equations. In this method the mesh is updated through the PDES

$$(30) \quad \varepsilon(I - \gamma\Delta_\xi)X_t = \nabla_\xi \cdot (M(X,Y)\nabla_\xi X), \quad \varepsilon(I - \gamma\Delta_\xi)Y_t = \nabla_\xi \cdot (M(X,Y)\nabla_\xi Y).$$

See [43], [42] for further applications. There are certain similarities between (30) and (18), but it involves solving a system of equations, separate equations need to be solved at the boundaries and there is no guarantee of mesh regularity or stability.

7. Implementation of the parabolic Monge–Ampère method. There are three key aspects for the useful implementation of the Parabolic Monge–Ampère method to mesh generation in the context of solving a PDE adaptively. Firstly, supposing that the Monitor function M is known explicitly (so that $M \equiv M(\mathbf{X}, t)$), how should (18) be discretized and solved efficiently. Secondly, if we assume that M is a function $M(u(\mathbf{X}, t))$ of the solution $u(\mathbf{X}, t)$ of an underlying PDE, how should the discretization of this equation be coupled to that of the underlying PDE. Thirdly, how should the choice of monitor function M be determined. The last of these three questions was discussed briefly in section 2. The second is very hard to answer in general, as it depends crucially on the nature of the underlying PDE (elliptic, hyperbolic, and/or systems) and the discretization strategy used to solve it (such as finite difference in the computational domain, finite volume or finite element in the physical domain, etc.). In [43], [44], and [38] reviews are given of different methods to do this. In this paper we will concentrate on the first question, but will give a partial answer to the second. A significant advantage of the approach that we use is that the spatial discretizations may all be done on a *uniform grid* in the computational space (although nonuniform grids can also be used if appropriate, for example, if extra refinement is needed at the boundaries of the domain).

7.1. Discretizing the parabolic Monge–Ampère equation (18). For integration of (18) in time together with the Neumann boundary conditions (16) we use a semidiscrete method, taking a finite difference/spectral method in the computational space to lead to a set of ordinary differential equations (in time), and then solving these equations using a *fully explicit* time-stepping method such as `ode45` or any other appropriate ODE solver. In calculating a mesh to solve a PDE, it is unnecessary to use a highly accurate method for solving (18) (as it is the solution of the PDE which matters rather than the mesh itself), and this greatly simplifies the methods that can be employed. The solution procedure is motivated in part by the methods outlined in [18] to solve (30) and had no difficulty in solving all the examples in the next section in a very reasonable period of time. We describe the method for solving this system on the assumption that we are working in two-dimensions, though there are no theoretical difficulties of extending it to higher dimensions. To discretize (18) in space we impose a uniform grid of mesh size $\Delta\xi$ on the computational space and assume that Q and Q_t take point values $Q_{i,j}(t), \quad i,j = 0 \ldots N$, etc. on this grid, with indices 0

and N corresponding to mesh points on the boundary. To discretize the differential operator $|H(Q)|$ we use a nine point stencil, and interior to the domain Ω_C, use central differencing to evaluate all second derivatives in $|H(Q)|$. For the boundary points, the central differences are replaced by Taylor series expansions, using the respective conditions $Q_\xi = 0$ or $Q_\xi = 1$, etc. The gradient ∇Q is calculated similarly, so that the right-hand side of (18) can be determined. To determine the values of Q_t at the mesh points we then invert the operator $(I - \gamma \Delta_{\boldsymbol{\xi}})$. As the system is posed on a uniform (rectangular) mesh in the computational domain, this inversion was done very rapidly by using a fast spectral solver based upon the discrete cosine transform (invoking the Neumann boundary conditions for Q_t). The value of γ must be chosen to give a balance between a smooth mesh and one which equidistributes M. Typically we take $\gamma = 1/\max(M)$. In the calculation of Q_t (and of the value of u_t for the underlying solution), it may be appropriate to build in some (adaptive) temporal rescaling, and this can be relatively easily incorporated into the (ordinary) differential equations for the point values of $Q_{i,j}$ by means of an appropriate Sundman transformation [32], [11].

It is important to apply some form of *smoothing* and *regularization* to the monitor function to allow for more regular meshes [34], [5], [39]. We have found it necessary to use two forms of regularization to derive the meshes presented in section 8. The first is to use a low pass filter applied to the point values of the monitor function to spread large local variations over several mesh points. Following [18] we iterate the following fourth-order filter twice,

$$M_{i,j} \leftarrow \frac{4}{16} M_{i,j} + \frac{2}{16} \left(M_{i+1,j} + M_{i-1,j} + M_{i,j-1} + M_{i,j+1} \right)$$
$$+ \frac{1}{16} \left(M_{i+1,j+1} + M_{i-1,j-1} + M_{i+1,j-1} + M_{i-1,j+1} \right),$$

to smooth the point values $M_{i,j}$ of the monitor function. Secondly, a common problem with r-adaptive methods is that the mesh points move into regions of singular behavior only, leaving a relatively sparse number of mesh points in other regions. Such behavior can be avoided by using meshes in which 50% of the mesh points are placed in regions where the monitor function is large, and the remainder in other regions [9]. This can be done [5], [8] by taking the original monitor function M and replacing it with the function $\widehat{M}$ defined by

$$\widehat{M} = M + \int_{\Omega_P} M(\mathbf{X}, t)\, d\mathbf{X}.$$

Applying the various procedures above to generate a moving mesh for which the function $M(\mathbf{X}, t) = M(\nabla Q, t)$ is known explicitly leads to a system of ordinary differential equations for the evolution of the mesh potential Q and hence of the mesh itself. We solve these using an explicit method (typically the MATLAB routine `ode23` or `ode45`), as the strongly nonlinear nature of the equations makes implicit time-stepping too expensive in general. We apply this in two stages.

1. Starting from an initially uniform mesh in the physical space, we adapt this to equidistribute the monitor function *at the initial time* over Ω_P. To do this we set $M_0(\mathbf{X}) \equiv M(\mathbf{X}, 0)$, $Q(\boldsymbol{\xi}, 0) = (\xi^2/2 + \eta^2/2)$ and solve (18) with M fixed to equal the function M_0, with $\varepsilon = 1$ for $0 < t < T$, where T is a fixed time. In this first calculation t is an *artificial time* during which the uniform mesh evolves toward an equidistributed mesh. It follows from Lemma 5.2

that such a mesh exists and is stable. The relatively large value of ε prevents this calculation from being unnecessarily stiff, and T is chosen large enough to allow the mesh to relax toward the equidistributed state.

2. We then solve (18) with the time-dependent monitor function $M(\mathbf{X}, t)$ with t now *actual time*. For this calculation we now set $\varepsilon = 0.01$. It follows from the results in Lemma 5.3 that the mesh resulting from taking the gradient of the solution of the parabolic Monge–Ampère problem will, as time evolves, be ε-close to a mesh which exactly equidistributes $M(\mathbf{X}, t)$, provided that M does not change too rapidly with time). As this procedure starts from a mesh which exactly equidistributes $M(\mathbf{X}, 0)$, the resulting differential equations are not especially stiff.

7.2. Coupling the parabolic Monge–Ampère equation to the underlying partial differential equation. In general, the function M is not known explicitly, but is a given function of the solution of the underlying PDE which we assume has the form.

$$ u_t = f(t, \mathbf{X}, u, \nabla_X u, \Delta_X u). \tag{31} $$

This equation is coupled with (18) to evolve the solution and mesh together. This can be done in various ways, see [43], [31] for reviews of these. The *quasi-Lagrangian* approaches (which avoid interpolation) allow directly for the mesh movement and express the PDE in Lagrangian variables

$$ \dot{u} - \nabla_X u \cdot \dot{\mathbf{X}} = f(t, \mathbf{X}, u, \nabla_X u, \Delta_X u), \tag{32} $$

where the $\dot{u}$, $\dot{\mathbf{X}}$ indicates differentiation with respect to time with the computational variable $\boldsymbol{\xi}$ fixed. The two equations (18) and (32) can then both be discretized on the *computational mesh* and solved *simultaneously*. This procedure is often used in calculations in one spatial dimension [33], but is harder to use in higher dimensions as the coupled system can be rather stiff and the equations are very nonlinear. This method has the advantage that there is no need to interpolate a solution from one mesh to the next as the mesh evolves and also that the mesh can inherit useful dynamical properties of the solution such as scaling structures [10], [8], [2]. Alternatively the two equations (18) and (32) can be solved *alternately* [31], [36], [18]. This reduces the stiffness problems but can lead to a lag in the mesh movement. As an alternative to the quasi-Lagrangian method, a *rezoning method* can be used [43]. In such methods equations similar to (18) are solved to advance the mesh by one time step. The current solution u is then interpolated onto this new mesh in the physical domain, and the original PDE (31) solved on the new mesh in this domain (often using a finite element or finite volume method). The advantage of this method is that standard software can be used to solve the PDE, but at the expense of an interpolation step.

We now briefly detail the implementation of the *simultaneous solution method* solving the mesh equation (18) and the PDE (32) by using finite differences in the *computational* domain. To do this we associate each mesh point in the fixed computational mesh with a solution point $U_{i,j}(t)$. To discretize (32) we transform the derivatives in the physical domain to ones in the computational domain. For example, in two-dimensions, if we have general transformation F with Jacobian J, then (see [18] for details) the derivatives of u can be expressed in terms of the computational

variables in the following manner:

(33)
$$u_x = \frac{1}{J}\left(y_\eta u_\xi - y_\xi u_\eta\right),$$

$$u_y = \frac{1}{J}\left(-x_\eta u_\xi + x_\xi u_\eta\right),$$

$$u_{xx} = \frac{1}{J}\left(y_\eta \left(J^{-1}y_\eta u_\xi\right)_\xi - y_\eta \left(J^{-1}y_\xi u_\eta\right)_\xi - y_\xi \left(J^{-1}y_\eta u_\xi\right)_\eta + y_\xi \left(J^{-1}y_\xi u_\eta\right)_\eta\right),$$

$$u_{yy} = \frac{1}{J}\left(x_\eta \left(J^{-1}x_\eta u_\xi\right)_\xi - x_\eta \left(J^{-1}x_\xi u_\eta\right)_\xi - x_\xi \left(J^{-1}x_\eta u_\xi\right)_\eta + x_\xi \left(J^{-1}x_\xi u_\eta\right)_\eta\right).$$

As the computational domain has a uniform mesh, and the Jacobian $J = |H(Q)|$ may be determined directly from Q, it follows that (33) can be discretized to high accuracy in space. The solution u is then determined by solving (18) and (32) together using the Matlab routine `ode45`. We give examples of this in the next section.

8. Examples. To demonstrate the effectiveness of this approach, we now describe a sequence of examples in two-dimensions. We consider two types of examples: Generating time-dependent grids based on given time-evolving monitor functions $M(\mathbf{X}, t)$ in two-dimensions, and the adaptive solution of two PDEs, one with a moving front and the other exhibiting finite time blow-up.

8.1. Examples with a prescribed monitor function. We have chosen the first of these examples to compare with other methods presented in the literature and described in section 6. All examples have been run in Matlab on a desktop computer in under five minutes.

Example 1. We first take a case motivated by [17] with a monitor function localized over a moving circle of the form

$$M_1(x, y, t) = 1 + 5\exp(-50|(x - 1/2 - 1/4\cos(2\pi t))^2$$
$$+ (y - 1/2 - \sin(2\pi t)/2)^2 - 0.01|).$$

This is described in [17] as a severe test of a moving mesh method, and the meshes calculated from this using the (velocity based) geometric conservation law (GCL) method [17] have a high degree of skewness. To compute a corresponding moving mesh using the parabolic Monge–Ampère method (and also the solutions in all of the examples in this subsection) we use a uniform computational mesh to solve (18) with $N = 30$ mesh points in each direction and mapping the unit square to the unit square. We see from this calculation that the resulting mesh closely follows the moving circle with now evidence of skewness or irregularity. Note the orthogonality and regularity of the mesh at the boundary of the domain. Notice further that in Figure 2 the solution partially exits the domain with no ill consequence and that the grid at $t = 10$ is virtually indistinguishable from that at $t = 0$. This is to be contrasted with a similar figure in [17] in which the grid is significantly distorted after the circle has passed out of the domain. It is clear from these figures that the mesh generated has excellent regularity. Observe the high degree of mesh uniformity close to the solution maximum.

Example 2. In an example taken directly from [17] we consider a monitor function localized on a sine-wave of the form

$$M_2(x, y, t) = 1 + 5\exp(-50|y - 1/2 - 1/4\sin(2\pi x)\sin(2\pi t)|)$$

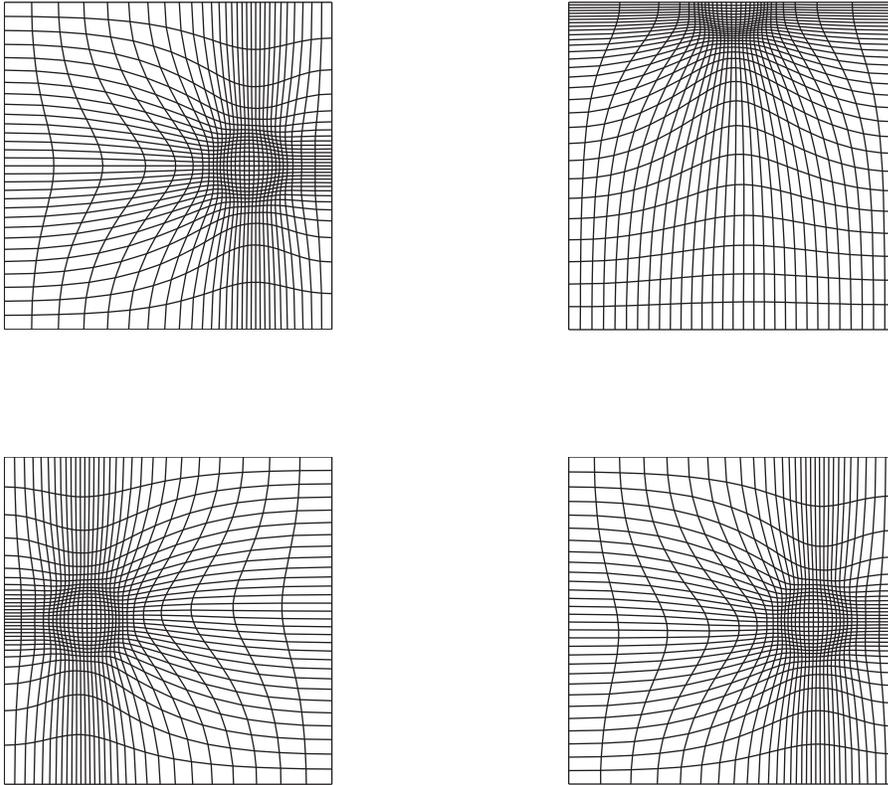and proceed to use exactly the same method as described in Example 1.

FIG. 2. *Example 1. Snapshots at $t = 0, 1/4$ in the top row and $t = 1/2, 10$ in the bottom. As the solution is advected about, the grid follows the maxima of the solution and does not ever fall behind or become distorted.*

In this example, presented in Figure 3, we again see that the PMA method has generated a very regular and periodic (in time) mesh. This is again in stark contrast to the GCL method which resulted in nonsmooth and nonperiodic meshes.

Example 3. In this example we consider a monitor function localized on the support of two travelling linear fronts moving toward each other and show the resulting mesh in Figure 4. For this we take

$$x_0 = t, \quad y_0 = 0.2 + t/2,$$
$$u_0 = \gamma \operatorname{sech}(\lambda(x - x_0 + y - y_0)),$$
$$x_1 = 1 - t, \quad y_1 = 0.8 - t/2,$$
$$u_1 = \gamma \operatorname{sech}(\lambda(x - x_1 + y - y_1)),$$
$$M_3(x, y, t) = 1 + u_0 + u_1$$

with $\gamma = 5$ and $\lambda = 100$. We note that the fronts pass through each other without generating spurious oscillations in the mesh. Note also that the solution points do not follow the front (moving in and out of the region of high mesh density), but that the *density* of the solution points does. Observe that the mesh automatically aligns itself along the front. A close-up of the mesh close to the front is given in Figure 5. This shows very good resolution of the local structure close to the front and a smooth transition from a uniform mesh to one refined at the front. If γ is large, then close
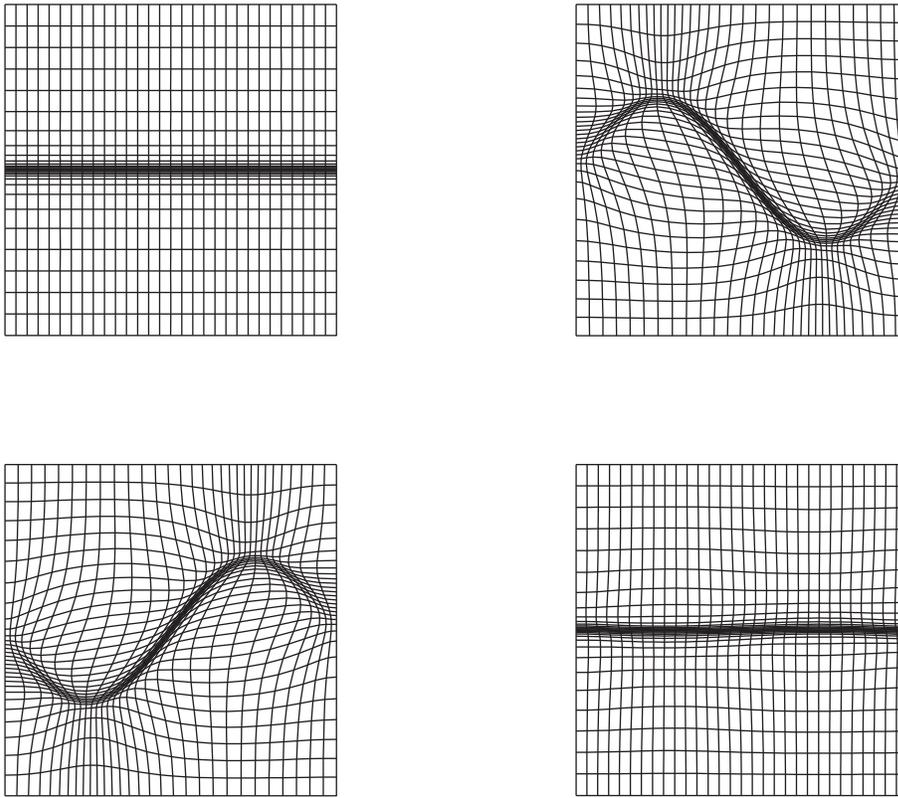
FIG. 3. *Example* 2. *Snapshots at* $t = 0, 1/3$ *in the top row and* $t = 2/3, 1$ *in the bottom. In this figure we see both the smoothness and the periodic (in time) forms of the mesh.*

to the front we see a mesh compression proportional to γ orthogonal to the front, with a local skewness of $s = \gamma$. The PMA method in this case thus generates a smooth mesh, for which the degree of skewness can be controlled via the choice of γ. More details of this calculation are given in [47]. We can also see the effects of mesh orthogonality close to the boundaries of the domain. As mentioned earlier, it is in principle possible to reduce the loss of resolution here by increasing the value of M close to the boundary, although we have not done this here.

Example 4. In this example we look at a monitor function localized on the support of two fronts meeting at an angle. The behavior of the mesh close to the front is similar to that in the last example. Significantly there is no mesh tangling as the two fronts intersect. The full meshes are presented in Figure 6 and in Figure 7 we show a close-up of the intersection region showing the high degree of mesh regularity there.

$$x_0 = t, \quad y_0 = 0.2 + t/2,$$
$$u_0 = \gamma \operatorname{sech}(\lambda(x - x_0 + y - y_0)),$$
$$x_1 = 1 - t, \quad y_1 = 0.8 - t,$$
$$u_1 = \gamma \operatorname{sech}(\lambda(x - x_1 + (y - y_1)/2)),$$
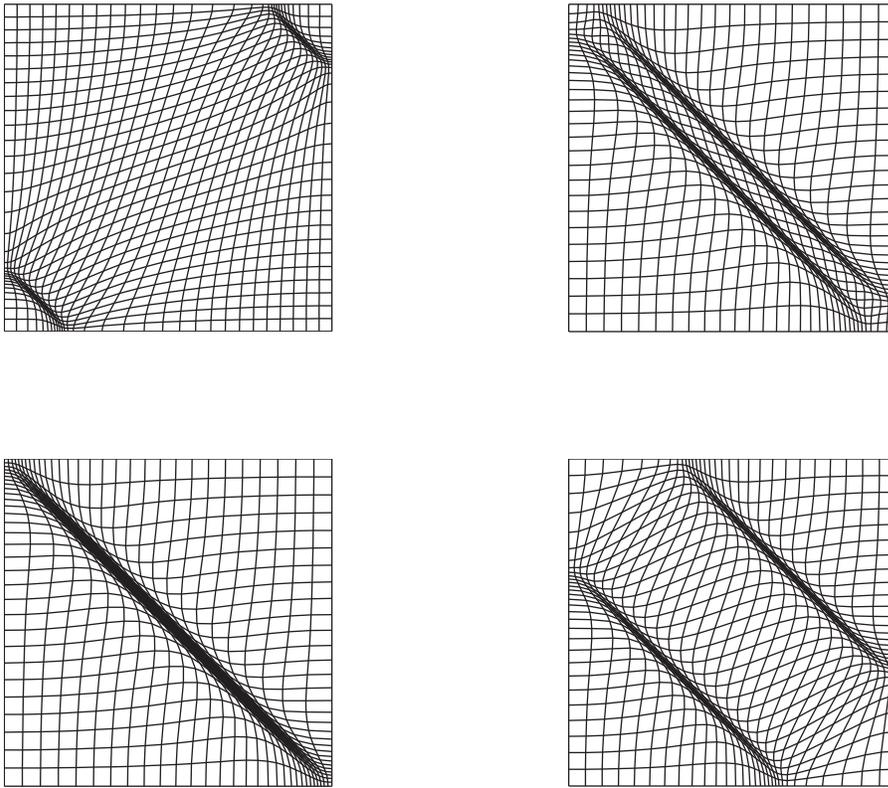$$M_4(x, y, t) = 1 + u_0 + u_1.$$

FIG. 4. *Example 3. Snapshots at $t = 0, 0.495$ in the top row and $t = 0.525, 0.75$ in the bottom. Here we have simulated two fronts passing through each other in parallel, and see no difficulties with the resulting mesh. Note the way that the mesh automatically aligns itself parallel to the front.*
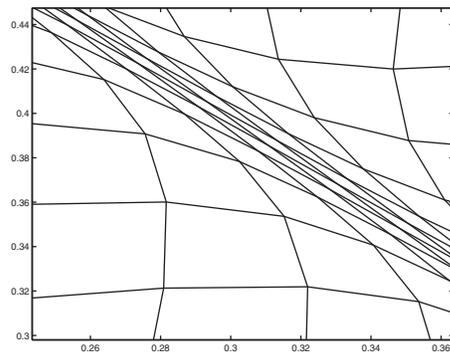

FIG. 5. *Example 3. A close-up of the mesh close to the front, showing the transition from a uniform mesh to one compressed by a factor γ orthogonal to the front.*

8.2. Solution-dependent monitor functions.

Example 5 (Burgers' equation). A classical problem leading to the formation of sharp fronts is Bergers' equation given by

$$(34) \qquad u_t + \frac{1}{2}\left(u^2\right)_x + \frac{1}{2}\left(u^2\right)_y = \nu \Delta u, \quad \nu \ll 1.$$
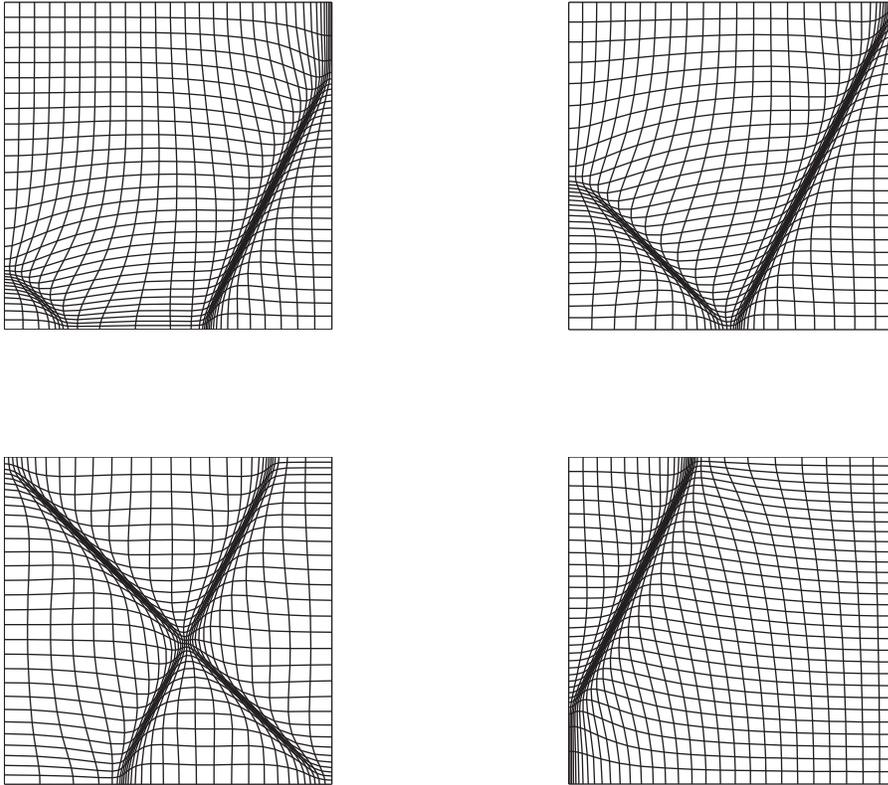
FIG. 6. *Example 4. Snapshots at $t = 0, 0.2$ in the top row and $t = 0.525, 1.4$ in the bottom. Here we have simulated two linear traveling fronts passing through each other at an angle and see no difficulties in generating and moving the mesh.*
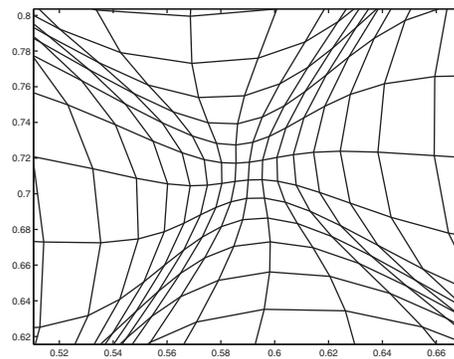

FIG. 7. *Example 4. A close-up of the region where the two fronts intersect.*

This equation has been used as a benchmark for a number of different moving mesh algorithms [49], [44] with the initial and boundary values chosen so that (34) has the exact solution given by

$$u(x, y, t) = \left(1 + e^{(x+y-t)/2\nu}\right)^{-1},$$

which has a sharp moving front. We consider coupling this system to the PMA algorithm to generate a moving mesh which can both compute an approximation to this solution and follow the front as it evolves over the time interval $t \in [1/4, 2]$. To do this we discretize (34) in the Lagrangian form

$$(35) \qquad u_t = \nu \Delta u - \left(\frac{1}{2} \left(u^2 \right)_x + \frac{1}{2} \left(u^2 \right)_y \right) + \dot{X} u_x + \dot{Y} u_y, \quad \nu \ll 1$$

with all discretizations made in the computational variables. The conservation form of the equation above is used for this calculation so that, for example, the advective term u_x^2 is discretized as

$$\left(u^2 \right)_x = \frac{1}{J} \left[y_\eta \left(u^2 \right)_\xi - y_\xi \left(u^2 \right)_\eta \right]$$

with similar discretizations for the other advective terms. The equation posed in the computational variables is then coupled to the discretized form of PMA equation with the values of the approximation U to $u(x, y)$ and of Q given on the mesh vertices. In these computations we take a $N \times N$ computational mesh, a viscosity of $\nu = 0.005$, and use the arc-length monitor function

$$M = \sqrt{1 + \alpha \left(u_x^2 + u_y^2 \right)}$$

with $\alpha = 1$. A number of different strategies were considered to evolve this coupled system forward in time, but in practice a simple scheme which solved the PMA equation and (35) *simultaneously* using a simple forward Euler method proved effective with a time-step Δt as given in [49] determined by the CFL condition. In the PMA equation we used $\varepsilon = 1, \gamma = 0.335$ to find the initial mesh when $t = 1/4$ and then $\varepsilon = 0.01, \gamma = \sqrt{\max(M)}$ to follow the front up to $t = 2$. More details are given in [47]. (Note also that solving the PMA equation coupled to (35), using an alternating solution strategy coupled with an upwinding discretization, has also been shown to be effective [40].) In Figure 8 we present the results of a series of computations using this method, when $N = 40$, showing both the solution for (34) and the corresponding mesh. In this figure we see excellent resolution of the solution at the front.

A quantitative measure of the error in the computed solution at $t = 1$ can be given by determining the L_2-norm of the difference between it and the exact solution. We consider this error both for a uniform and a moving mesh for various values of N.

N	Error on a uniform mesh	Error on a moving mesh
20	6e−2	8e−3
40	2e−2	2e−3
80	6e−3	1e−3

(See [40] for a similar table obtained by using the alternating, upwind method PMA and [49] using a harmonic mapping method to generate the mesh coupled to a finite volume discretization of (34).) We see a useful reduction in the error by using the PMA method in this case, similar to that observed in [49].

Example 6 (Blow-up). In this example we consider a problem, in which the mesh is coupled to a rapidly time-evolving solution of a PDE, by considering a classical example of a nonlinear parabolic PDE for $u(\mathbf{X}, t)$ which admits solutions which *blow up* in a finite time T (given sufficiently large initial data) [10], [8] so that $|u|_\infty$ as $t \to T$. Such solutions develop a large and increasingly narrow peak as t tends to T. It is not uncommon to consider solutions which change by ten orders of magnitude,
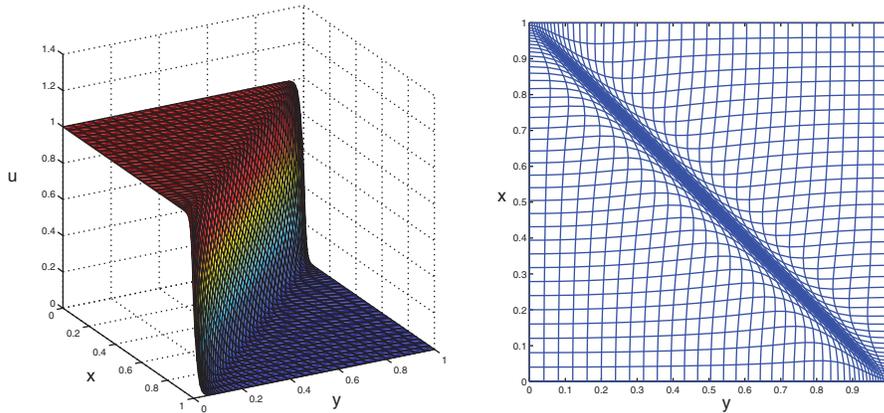
FIG. 8. *Example 5. The solution of Burgers' equation and the corresponding mesh at the time* $t = 1$.

with a reduction in the solution length scale by a similar amount. It is essential to use an adaptive method to capture such behavior accurately. The equation that we consider is

$$
(36) \quad
\begin{cases}
u_t = \Delta u + u^3, & \mathbf{X} = (x, y) \in \Omega = (0, 1)^2, \\
u(\mathbf{X}, t) = 0, & \mathbf{X} \in \partial\Omega, \\
u(\mathbf{X}, 0) = 5 \exp\left(-25(x - 0.45)^2 - 25(y - 0.35)^2\right).
\end{cases}
$$

We augment this problem with the PMA equation (18) where the monitor function is taken as

$$
M(\mathbf{X}, t) = u(\mathbf{X}, t)^4 + \int_{\Omega_P} u(\mathbf{X}', t)^4 \, d\mathbf{X}'.
$$

It is shown in [10], [8] that this choice of monitor function (which is derived a priori by considerations of the symmetries of the underlying PDE) leads to a mesh which automatically inherits the correct dynamic length scale of the underlying solution so that the mesh potential $Q(\xi, \eta, t)$ takes the form

$$
Q(\xi, \eta, t) = L(t)\hat{Q}(\xi, \eta),
$$

where $L(t)$ is a natural length scale for the problem. As a direct consequence of this the *relative local truncation error* of the solution close to the peak becomes independent of the solution scale [8]. (This analysis is applicable to a much wider class of problems with scaling symmetries.) The previous analysis of mesh skewness then implies that there should be good mesh regularity close to the evolving solution peak. As described in section 7, we solve this problem in the computational domain Ω_C using a finite difference method applied to the Lagrangian form of the problem with a rescaling of the Laplacian to account for the change in the mesh geometry. In this calculation an adaptive time-step is also used as the blow-up time is approached, employing the Sundman transformation to derive this. In Figure 9 we show the final grid both over
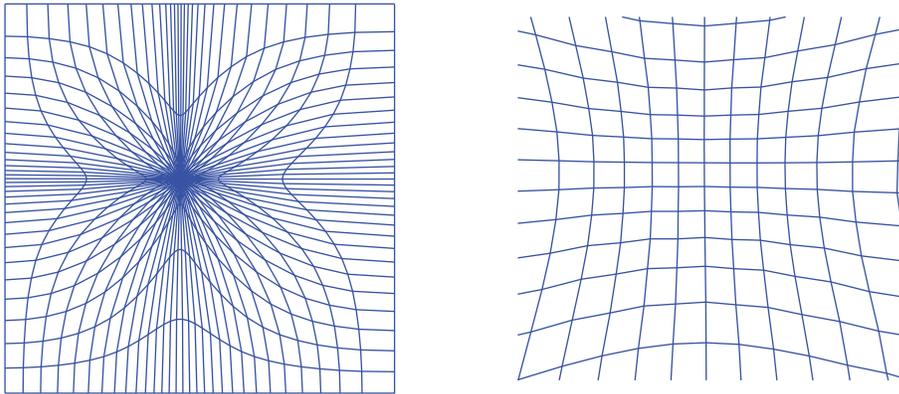
FIG. 9. *Final grid for Example* 6. *Left: entire grid when* $|u|_\infty = 1e15$. *Right: detail near the blow-up point. The grid is quite regular in the vicinity of the singularity.*

the whole domain and also close to the peak near the center as well as the initial and final solutions. The integration was performed until $|u|_\infty = 10^{15}$, for which the peak has approximate length scale of 10^{-15}. Over the course of the evolution we see a mesh compression of a factor of 10^{12}. The final mesh shows a similar gradation of mesh elements from size $\mathcal{O}(10^{-2})$ to size $\mathcal{O}(10^{-14})$. However, if we study the mesh close to the solution peak, as illustrated in Figure 9, we see that this shows a strong degree of local regularity, with no evidence of long thin elements in the region where the solution gradients are very large. This is exactly as predicted by the previous theory. (Away from the solution peak the mesh is less degree of smoothness; however in this region the solution gradients are much smaller than in the peak, and the local truncation errors are thus lower.)

9. Conclusions. In this paper we have presented a new technique for generating a mesh close to a uniform one by solving the parabolic Monge–Ampère equation (18). The numerical advantages of this method are that it is computationally simple (with automatic treatment of boundaries) and computationally efficient, requiring only the solution of a scalar problem. It can be initialized with a uniform mesh. It can also be analyzed analytically, with guaranteed local convergence and stability (without mesh crossing) for all dimensions d, the ability to find a solution close to an equidistributed one, and good scaling and rotation properties. The examples presented indicate that the method copes well with a variety of different geometries, delivering smooth regular meshes, and it can be coupled to the solution of certain types of PDE with relative ease and without the need for interpolation. The main disadvantages of this method at present are the limited choices of monitor function M (which must be scalar valued), the lack of direct skewness control, and the forced orthogonality of the mesh at the boundaries.

In subsequent papers we will study the coupling of this method to further problems with moving fronts and methods for estimating and controlling the skewness of the meshes that result in general applications.

REFERENCES

[1] M. J. Baines, M. E. Hubbard, and P. K. Jimack, *A moving mesh finite strategy for the adaptive solution of time-dependent partial differential equations with moving boundaries*, Appl. Numer. Math., 54 (2005), pp. 450–469.

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

[3] M. J. Baines and S. L. Wakelin, *Equidistribution and the Legendre transformation*, Univ. of Reading, Reading, PA, Numer. Anal. Rep., 4/91, 1991.

[4] G. I. Barenblatt, *Scaling, self-similarity and intermediate asymptotics*, Cambridge University Press, Cambridge, 1996.

[5] G. Beckett and J. A. Mackenzie, *Convergence analysis of finite difference approximations on equidistributed grids to a singularly perturbed boundary value problem*, Appl. Numer. Math, 35 (2000), pp. 87–109.

[6] J. D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numer. Math., 84 (2000).

[7] Y. Brenier, *Polar factorization and monotone rearrangement of vector-valued functions*, Comm. Pure Appl. Math., 44 (1991), pp. 375–417.

[8] C. Budd and J. W. Williams, *Parabolic Monge-Ampère methods for blow-up problems in several spatial dimensions*, J. Phys. A, 39 (2006), pp. 5425–5444.

[9] C. J. Budd, R. Carretero-Gonzalez, and R. D. Russell, *Precise computations of chemotactic collapse using moving mesh methods*, J. Comput. Phys., 202 (2005), pp. 462–487.

[10] C. J. Budd, W. Z. Huang, and R. D. Russell, *Moving mesh methods for problems with blow-up*, SIAM J. Sci. Comput., 17 (1996), pp. 305–327.

[11] C. J. Budd, B. Leimkuhler, and M. D. Piggott, *Scaling invariance and adaptivity*, Appl. Numer. Math., 39 (2001), pp. 261–288.

[12] C. J. Budd and M. D. Piggott, *Geometric integration and its applications*, in Handbook of Numerical Analysis, P. G. Ciarlet and F. Cucker, eds., North-Holland, Amsterdam, 2005.

[13] C. J. Budd, M. D. Piggott, and J. F. Williams, *Adaptive numerical methods and the geostrophic coordinte transformation*, Monthly Weather Review, submitted.

[14] L. A. Caffarelli, *The regularity of mappings with a convex potential*, J. Amer. Math. Soc., 5 (1992), pp. 99–104.

[15] L. A. Caffarelli, *Boundary regularity of maps with convex potentials*, Ann. Math., 3 (1996), pp. 453–496.

[16] W. Cao, W. Huang, and R. D. Russell, *An r-adaptive finite element method based upon moving mesh pdes*, J. Comput. Phys., 149 (1999), pp. 221–244.

[17] W. Cao, W. Huang, and R. D. Russell, *A moving mesh method based on the geometric conservation law*, SIAM J. Sci. Comput., 24 (2002), pp. 118–142.

[18] H. D. Ceniceros and T. Y. Hou, *An efficient dynamically adaptive mesh for potentially singular solutions*, J. Comput. Phys., 172 (2001), pp. 609–639.

[19] S. Chynoweth and M. J. Baines, *Legendre transform solutions to semi-geostrophic frontogenesis*, in Finite element analysis in fluids, T. J. Chung and G. R. Kerr, eds., 1989, pp. 697–703.

[20] S. Chynoweth and M. J. Sewell, *Dual variables in semigeostrohic theory*, Proc. R. Soc. Lond. Ser. A, 424 (1989), pp. 155–186.

[21] M. J. P. Cullen, *Implicit finite difference methods for modelling discontinuous atmospheric flows*, J. Comput. Phys., 81 (1989), pp. 319–348.

[22] M. J. P. Cullen, J. Norbury, and R. J. Purser, *Generalized Lagrangian solutions for atmospheric and oceanic flows*, SIAM J. Appl. Math., 51 (1991), pp. 20–31.

[23] M. J. P. Cullen and R. J. Purser, *An extended Lagrangian theory of semi-geostrophic frontogenesis*, J. Atmos. Sci., 41 (1984), pp. 1477–1497.

[24] M. J. P. Cullen, *A mathematical theory of large-scale atmosphere/ocean flow*, Imperial College Press, London, 2006.

[25] C. deBoor, *Good approximations by splines with variable knots* II, 1973.

[26] W. Gangbo and R. J. McCann, *The geometry of optimal transport*, Acta Math., 177 (1996), pp. 113–161.

[27] W. Huang, *Practical aspects of formulation and solution of moving mesh partial differential equations*, J. Comput. Phys., 171 (2001), pp. 753–775.

[28] W. Huang, *Variational mesh adaption: Isotropy and equidistribution*, J. Comput. Phys., 174 (2001), pp. 903–924.

[29] W. Huang, *Measuring mesh qualities and application to variational mesh adaption*, SIAM J. Sci. Comput., 26 (2005), pp. 1643–1666.

[30] W. HUANG, *Metric tensors for anisotropic mesh generation*, J. Comput. Phys., 204 (2005), pp. 663–665.

[31] W. HUANG, *Anisotropic mesh adaption and movement*, in *Adaptive Computations: Theory and Algorithms*, T. Tang and J. Xu, eds., Science Press, Bejing, 2007.

[32] W. HUANG AND B. LEIMKUHLER, *The adaptive Verlet method*, SIAM J. Sci. Comput., 18 (1997), pp. 239–256.

[33] W. HUANG, Y. REN, AND R. D. RUSSELL, *Moving mesh partial differential equations (MM-PDES) based on the equidistribution principle*, SIAM J. Numer. Anal., 31 (1994), pp. 709–730.

[34] W. HUANG AND R. D. RUSSELL, *Analysis of moving mesh partial differential equations with spatial smoothing*, SIAM J. Numer. Anal., 34 (1997), pp. 1106–1126.

[35] W. HUANG AND R. D. RUSSELL, *A high dimensional moving mesh strategy*, Appl. Numer. Math., 26 (1999), pp. 63–76.

[36] W. HUANG AND R. D. RUSSELL, *Moving mesh strategy based on a gradient flow equation for two-dimensional problems*, SIAM J. Sci. Comput., 20 (1999), pp. 998–1015.

[37] W. HUANG AND R. D. RUSSELL, *Adaptive mesh movement — the MMPDE approach and its applications*, J. Comput. Appl. Math, 128 (2001), pp. 383–398.

[38] J. A. MACKENZIE AND W. R. MEKWI, *On the use of moving mesh methods to solve PDEs*, in Adaptive Computations: Theory and Algorithms, T. Tang and J. Xu, eds., Science Press, Bejing, 2007.

[39] J. A. MACKENZIE AND M. L. ROBERTSON, *A moving mesh method for the solution of the one-dimensional phase-field equations*, J. Comput. Phys., 181 (2002), pp. 526–544.

[40] M. H. SULMAN, *Optimal mass transport for adaptivity and image registration*, Ph.D. Thesis, SFU, Burnaby, BC, Canada, 2007.

[41] M. J. SEWELL, *On Legendre transformations and umbilic catastrophes*, Math. Proc. Cambridge Philos. Soc., 83 (1978), pp. 273–288.

[42] C. SULEM AND P. L. SULEM, *The nonlinear Schrodinger equation: Self focusing and wave collapse*, Springer-Verlag, New York, 1999.

[43] T. TANG, *Moving mesh methods for computational fluid dynamics*, Contemp. Math., 383 (2005), pp. 141–173.

[44] T. TANG AND J. XU, editors, *Adaptive Computations: Theory and Algorithms*, Science Press, Bejing, 2007.

[45] J. F. THOMPSON, Z. U. A. WARSI, AND C. W. MASTIN, *Numerical Grid Generation*, North-Holland, 1985.

[46] C. VILLANI, *Topics in Optimal Transportation*, Graduate Studies in Mathematics 58, AMS, Providence, RI, 2003.

[47] E. WALSH, C. J. BUDD, AND J. F. WILLIAMS, *The PMA method for grid generation applied to problems with moving fronts*, in preparation.

[48] A. M. WINSLOW, *Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh*, J. Comput. Phys., 2 (1967), pp. 149–172.

[49] Z.-R. ZHANG AND T. TANG, *An adaptive mesh redistribution algorithm for convection-dominated problems*, Comm. Pure Appl. Anal., 1 (2002), pp. 341–357.