Journal of Computational Physics 227 (2008) 9841–9864

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An optimal robust equidistribution method for two-dimensional grid adaptation based on Monge–Kantorovich optimization

G.L. Delzanno [a,*], L. Chacón [a], J.M. Finn [a], Y. Chung [b], G. Lapenta [a]

[a] T-15 Plasma Theory Group, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[b] Department of Mathematics, Southern Methodist University, Dallas, TX, USA

ARTICLE INFO

Article history:
Received 26 February 2008
Received in revised form 7 July 2008
Accepted 31 July 2008
Available online 12 August 2008

PACS:
02.70.−c

Keywords:
Adaptive grid generation
Monge–Ampère equation
Monge–Kantorovich optimization
Grid tangling
Equidistribution
Newton–Krylov
Multigrid preconditioning
Moving meshes

ABSTRACT

A new cell-area equidistribution method for two-dimensional grid adaptation, based on Monge–Kantorovich optimization (or Monge–Kantorovich optimal transport), is presented. The method is based on a rigorous variational principle, in which the L_2 norm of the grid displacement is minimized, constrained *locally* to produce a prescribed positive-definite cell volume distribution. The procedure involves solving the Monge–Ampère equation: A single, nonlinear, elliptic scalar equation with no free parameters, and with proved existence and uniqueness theorems. We show that, for sufficiently small grid displacement, this method also minimizes the mean grid-cell distortion, measured by the trace of the metric tensor. We solve the Monge–Ampère equation numerically with a Jacobian-Free Newton–Krylov method. The ellipticity property of the Monge–Ampère equation allows multigrid preconditioning techniques to be used effectively, delivering a scalable algorithm under grid refinement. Several challenging test cases demonstrate that this method produces optimal grids in which the constraint is satisfied numerically to truncation error. We also compare this method to the well known deformation method [G. Liao, D. Anderson, Appl. Anal. 44 (1992) 285]. We show that the new method achieves the desired equidistributed grid using comparable computational time, but with considerably better grid quality than the deformation method.

1. Introduction

Equidistribution has traditionally been a fundamental guiding principle in grid generation, as is evidenced by the ample literature on the subject (see e.g. [1–4] and references therein). The problem is very simply posed: Generate a grid that equidistributes a given quantity along an arc (1D), a surface (2D), or a volume (3D). (Hereafter, we will refer to equidistribution with respect to *volumes* for arbitrary dimensionality.) The concept is most attractive due to its conceptual simplicity. Furthermore, in the context of error equidistribution, a rigorous connection exists between error equidistribution and minimization of total error [5,6].

In one dimension, equidistribution determines the grid uniquely [1,2,7,8]. This follows because, in 1D, only one unknown exists per cell, and it can be determined uniquely by specifying the Jacobian of the transformation at each cell. However, in two or more dimensions, equidistribution itself is not sufficient to determine the grid uniquely. There are many possible

* Corresponding author. Tel.: +1 5056672604; fax: +1 5056657150.
E-mail addresses: delzanno@lanl.gov (G.L. Delzanno), chacon@lanl.gov (L. Chacón), finn@lanl.gov (J.M. Finn), ychung@smu.edu (Y. Chung), lapenta@lanl.gov (G. Lapenta).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.07.020

grids that satisfy a given equidistribution principle, and therefore a possibility arises to select an optimal grid in some reasonable sense.

Historically, many approaches have been developed to determine good quality grids in two and higher dimensions. Variational methods have received a great deal of attention, as they provide a solid mathematical foundation, leading to Euler–Lagrange equations that govern the generation of the grid. There are several very good reviews on the subject [1,2,7,8]. Pioneering work by Winslow [9] used the "smoothness" integral (a measure of the trace of the metric tensor) to determine a set of Laplace equations that govern the generation of the grid (equipotential method). This method was very successful in allowing the modeler to generate grids for complicated boundaries. In a later development [10], Winslow proposed the variable diffusion method, in order to have control of the properties of the grid within the domain. Brackbill and Saltzman [11] used a cost function consisting of a combination of smoothness, orthogonality, and volume variation integrals to define a grid generation equation that incorporates properties of each. In this formulation, a term related to equidistribution is introduced by the volume variation integral. However, local equidistribution was not achieved due to competition among the different cost functions. Dvinsky [12] pioneered the use of harmonic maps for grid generation, and they have since been explored by a number of authors [4,13–15]. Harmonic maps can also be derived from variational principles, and they are very attractive because, in 2D under certain conditions, the existence of a solution is guaranteed [16,17]. However, they have important drawbacks, as we outline in the next paragraph. More recently, Huang [18] revisited the issue of variational equidistribution by providing rigorous integral measures of isotropy (or smoothness) and uniformity (or equidistribution). Again, equidistribution appears as a fundamental ingredient of the grid generation strategy.

Despite the attractiveness of these variational approaches – their mathematical soundness and the reasonable quality of the resulting grids – they suffer from various drawbacks. In particular, some require user-provided parameters to decide the relative contributions of the globally averaged terms in the cost function, and, when these parameters are chosen poorly, this may lead to mathematically ill-posed problems [19]. In addition, they result in as many Euler–Lagrange equations as dimensions considered, and these are strongly coupled and very nonlinear. They are therefore difficult to solve numerically (although, in the context of harmonic functionals, there has been recent progress employing state-of-the-art nonlinear algorithms [20]). The major disadvantage of some of these approaches is the following: Because the global grid property integrals compete against each other, the grid never truly satisfies any constraint, including equidistribution, to any predictable accuracy.

To resolve some of these issues, elliptic grid generation methods were first proposed by Thompson et al. [21], which evolved from the earlier work by Winslow [9]. In these approaches, non-homogeneous terms are added to Winslow's equipotential method. When properly specified, such terms allow good control of the properties of the grid. In [22], the source terms are found from a least squares (variational) fit of the inverse Jacobian matrix of the transformation to a target matrix with the desired properties. In Refs. [3,23], evidence that an elliptic approach can be made equivalent to an equidistribution principle by a proper choice of the non-homogeneous terms is presented. The attractiveness of these methods is that the equations remain essentially elliptic, and are therefore fairly tractable algorithmically. However, they still require as many coupled, nonlinear elliptic equations as the dimensionality of the problem, there is no rigorous existence and uniqueness theory, and for the most part they lack the mathematical soundness of variational principles.

Several authors have attempted to generalize the concept of grid equidistribution directly to multiple dimensions. One approach is to consider equidistribution along one-dimensional arcs in the multi-dimensional domain [3,24,25]. This method has the advantage that the task of multidimensional grid generation may be decomposed in a series of 1D equidistribution steps along coordinate arcs [3,26,27]. However, it has been shown [3,24] that the concept of arc equidistribution can only be satisfied locally in the domain, not globally, and that it generates fairly poor-quality grids (and may even fold the grid [24]).

The case for the need of cell-volume equidistribution to fix such smoothness problems has been made by various authors [28,23]. As we have argued earlier, in dimensions greater than unity equidistribution does not guarantee a unique solution. This implies that there is room for grid optimization. Recently, Kania [23] derived a set of non-homogeneous terms for Thompson's method that achieves volume equidistribution, and demonstrated that the generated grids are of good quality (although no *a posteriori* measure is given of how accurate the equidistribution principle is satisfied by the generated grids). Liao and Anderson [29] proposed an ODE-based equidistribution approach based on the work of Moser [30,31]. By suitably defining a flow velocity and an accompanying set of ODEs, it was demonstrated that the approach leads to an equidistributed grid. Then, based on ODE theory, they demonstrate that the solution obtained by the procedure exists and is unique, once the flow is prescribed. However, there is great latitude in choosing the flow, and for a fixed flow there is no evidence that the resulting grid is optimal in any sense.

In this paper, we propose a new approach for cell-volume equidistribution, based on *Monge–Kantorovich* optimization [32,33]. This method is based on a constrained minimization approach. Instead of minimizing a quantity consisting of a grid quality measure plus an equidistribution measure, this method involves minimizing a grid quality measure constrained *locally* by the equidistribution principle. This constraint is enforced by a *local* Lagrange multiplier. In this fashion, the method chooses the optimal grid which is compatible with the equidistribution principle. The minimization procedure results in a *single*, nonlinear, elliptic equation for the Lagrange multiplier with no tunable parameters, the *Monge–Ampère equation* [34]. This equation has been shown (see e.g. [35]) to have a unique solution in 2D and 3D. Our variational method achieves equidistribution up to truncation error. This is unlike existing elliptic or variational approaches which, besides requiring as many equations as the dimensionality of the problem, in general do not enforce equidistribution locally. Furthermore, in this work, we also establish a connection between the Monge–Kantorovich approach and the minimization of grid cell distortion (smoothness), which is a very desirable property in grid generation (see e.g. [11,24]).

Our approach combines the advantages of both variational and elliptic grid generation approaches. The variational character of our method is obvious, as it stems from a minimization procedure. The leading term of the resulting Euler–Lagrange equation is a Laplace operator (Section 2.2), and the linearized full operator is elliptic in nature (Section 3.2). Therefore, the method is suitable for modern, fast nonlinear solvers for elliptic equations. Specifically, we will demonstrate in this study the effectiveness of multigrid-preconditioned Newton–Krylov methods.

We would like to point out that the Monge–Ampère equation has been discussed before in the context of grid generation in Ref. [36], where its suitability for blow-up problems (with a developing point singularity) was shown. However, in that reference, the Monge–Ampère equation is parabolized to obtain an approximate solution (vs. the scalable, fully nonlinear algorithm proposed here). Further, there was no discussion of grid optimality in any sense.

The remainder of this paper is organized as follows. In Section 2 we formulate the problem and introduce our equidistribution approach based on Monge–Kantorovich optimization, and the resulting Monge–Ampère equation. In Section 3 we discuss certain properties of the Monge–Kantorovich approach, such as ellipticity and the connection with the minimization of grid distortion (maximization of grid smoothness). The numerical implementation of the equidistribution PDE is briefly discussed in Section 4. In Section 5, the Monge–Kantorovich approach is tested with several challenging examples. A comparison is also made with the deformation method [29] and with a method that minimizes grid-cell distortion. All the tests demonstrate the effectiveness and robustness of the Monge–Kantorovich approach in achieving optimally equidistributed grids. Conclusions are drawn in Section 6. The Appendix briefly reviews the deformation method [29].

2. Monge–Kantorovich optimization

2.1. Prescribing the Jacobian

The problem involves finding a one-to-one transformation in physical space according to a prescribed transformation Jacobian (or density or monitor function), i.e. to generate an adaptive grid with prescribed volumes. In this study, we focus on the 2D case. An example of the applicability of the method to 3D is presented in Ref. [37]. Let $X \subset \mathbb{R}^2$ be a bounded domain with boundary ∂X. We define a two-dimensional coordinate transformation in physical space between the coordinates of an initial grid $\mathbf{x} = (x, y)$ and the ones of the final grid $\mathbf{x}' = (x', y')$ as $\psi : X \to X$, i.e. $\mathbf{x}' = \psi(\mathbf{x})$. See Fig. 1. We will assume that the boundary ∂X maps to itself. The Jacobi matrix $\mathscr{J}$ is defined as

$$\mathscr{J} = \begin{pmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} \end{pmatrix}, \tag{1}$$

or $\mathscr{J}_{ij} = \partial x'_i / \partial x_j$. The Jacobian of the transformation ψ from $\mathbf{x} \to \mathbf{x}'$ is its determinant $\det[\nabla \mathbf{x}']$, which in 2D can also be written as $[x', y']$, where $[f, g] = (\partial f/\partial x)(\partial g/\partial y) - (\partial f/\partial y)(\partial g/\partial x)$ is the familiar Poisson bracket. Both the initial grid $\mathbf{x}$ and the final grid $\mathbf{x}'$ are mapped from the unit square $\boldsymbol{\xi} = (\xi, \eta) \in \Xi \equiv [0, 1] \times [0, 1]$, the logical space. We assume the mapping $\phi : \Xi \to X$, i.e. $\mathbf{x} = \phi(\boldsymbol{\xi})$, is given and $j(\boldsymbol{\xi})$ is its Jacobian. On the other hand, the mapping $\phi' : \Xi \to X$ giving $\mathbf{x}' = \phi'(\boldsymbol{\xi})$ is unknown and $j'(\boldsymbol{\xi})$

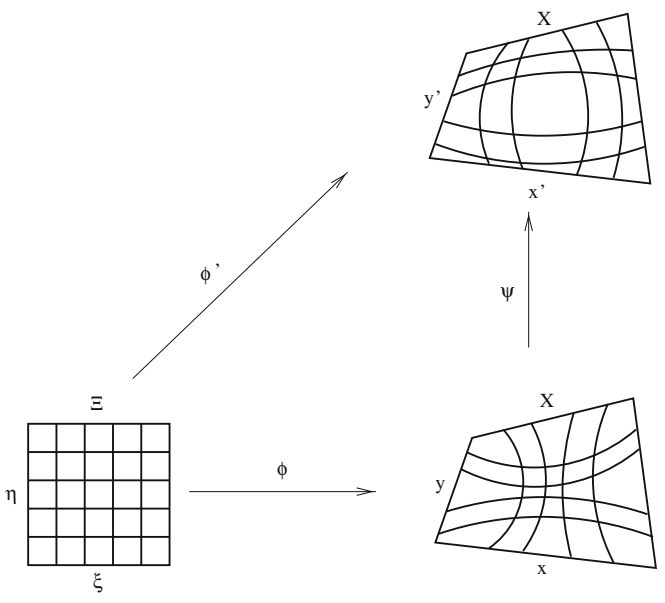


Fig. 1. Sketch of the mappings between the logical and physical spaces Ξ and X.

is its (prescribed) Jacobian. In what follows we will always use the lower case to refer to the Jacobian in terms of the logical variables ξ and the upper case to refer to the same Jacobian expressed in terms of the coordinates of the initial and final grids $\mathbf{x}, \mathbf{x}'$. For instance, we have

$$j'(\xi,\eta) = J'(x'(\xi,\eta), y'(\xi,\eta)) = J'(x',y') \tag{2}$$

for the transformation ϕ' from $\xi \to \mathbf{x}'$.

The sketch in Fig. 1 shows the physical and logical domains and the relative mapping transformations. We envision positive, nonuniform density (monitor) functions $\rho(\mathbf{x})$ and $\rho'(\mathbf{x}')$. With appropriate grids for $\mathbf{x}$ and $\mathbf{x}'$, the corresponding density $\rho_0(\xi)$ on Ξ will be constant ($\rho_0 = 1$) and be equidistributed by a uniform grid on Ξ. Thus,

$$\rho'(x',y')\mathrm{d}x'\,\mathrm{d}y' = \rho(x,y)\mathrm{d}x\,\mathrm{d}y = \mathrm{d}\xi\,\mathrm{d}\eta. \tag{3}$$

We conclude $\rho(x,y) = 1/J(x,y)$ and $\rho'(x',y') = 1/J'(x',y')$. Accordingly, the density functions ρ and ρ' must satisfy the density normalization condition:

$$\int_X \rho(x,y)\mathrm{d}x\,\mathrm{d}y = \int_X \rho'(x',y')\mathrm{d}x'\,\mathrm{d}y' = \int_\Xi \mathrm{d}\xi\,\mathrm{d}\eta = 1. \tag{4}$$

The requirement that X maps to itself under the transformation ψ from $\mathbf{x} \to \mathbf{x}'$ leads to:

$$\int_\Xi J'\mathrm{d}\xi\,\mathrm{d}\eta = \int_\Xi J\mathrm{d}\xi\,\mathrm{d}\eta = V \Rightarrow$$
$$\int_X \left[\frac{J'(x'(x,y),y'(x,y))}{J(x,y)} - 1\right]\mathrm{d}x\,\mathrm{d}y = \int_X \left[\frac{\rho(x,y)}{\rho'(x'(x,y),y'(x,y))} - 1\right]\mathrm{d}x\,\mathrm{d}y = 0, \tag{5}$$

which simply states that the total volume of the physical domain remains the same under the transformation ψ. This condition must be satisfied by any map $\mathbf{x}'(\mathbf{x})$ that maps X to itself. Note that whereas Eq. (4) is a condition on the densities ρ and ρ', Eq. (5) is a condition on ρ, ρ' and the map ψ, i.e. on $\mathbf{x}'(\mathbf{x})$.

The aim of this study is to find the one-to-one and onto transformation $\psi : X \to X$ giving $\mathbf{x} \to \mathbf{x}'$ such that

$$[x',y'] = \frac{\rho(x,y)}{\rho'(x',y')} \quad \text{for all } (x,y) \in X. \tag{6}$$

Eq. (6) ensures that the final grid will be equidistributed according to the densities ρ and ρ'. Eq. (5) is a solvability condition for finding this transformation ψ. As we shall see (Section 4.3), enforcing Eq. (5) is crucial to improve the convergence rate of the numerical algorithm.

It is clear that, so posed, the problem has an infinite number of solutions. In what follows, we propose an approach to seek a map $\psi : \mathbf{x} \to \mathbf{x}'$ leading to a single grid which is optimal in a well-defined sense.

2.2. Monge–Kantorovich optimization: Minimization of the L_2 norm of the grid displacement

We begin with formulating the coordinate transformation in terms of a displacement function $\mathbf{p} = (p_x, p_y)$:

$$x' = x + p_x(x,y), \tag{7}$$
$$y' = y + p_y(x,y). \tag{8}$$

In this paper, we will restrict ourselves to physical domains characterized by a four-sided quadrilateral, in which each side of the unit square in logical space Ξ is mapped to a corresponding straight side in the physical space X. (Extension to domains with curved boundaries has been carried out in Ref. [37].) On the boundary segments, we require that the displacement $\mathbf{p}$ satisfies

$$\mathbf{p}\cdot\mathbf{n} = 0 \quad \text{on } \partial X, \tag{9}$$

where $\mathbf{n}$ is the unit vector normal to the boundary. This means that the boundary points are allowed to move only tangentially to the boundary, so that the boundary ∂X maps to itself. This is consistent with Eq. (5). Using Eqs. (7) and (8), the Jacobian can be written as

$$[x',y'] = \left(\frac{\partial x'}{\partial x}\frac{\partial y'}{\partial y} - \frac{\partial x'}{\partial y}\frac{\partial y'}{\partial x}\right) = \left(1 + \frac{\partial p_x}{\partial x}\right)\left(1 + \frac{\partial p_y}{\partial y}\right) - \frac{\partial p_y}{\partial x}\frac{\partial p_x}{\partial y}. \tag{10}$$

We now formulate a variational principle for which $\mathbf{p}$ is optimal in some sense, with the constraint of equidistribution built-in. We first consider a generic functional of the mapping

$$\mathscr{I}[\mathbf{x}'] = \int_X G(\mathbf{x}, \mathbf{x}', \mathbf{x}'_{x^i})\mathrm{d}x\,\mathrm{d}y, \tag{11}$$

where G is a scalar function, and $\mathbf{x}'_{x^i} = \partial\mathbf{x}'/\partial x^i$. The functional (11) is minimized if the mapping $\mathbf{x}'(\mathbf{x})$ obeys the well-known Euler–Lagrange equations.

Next, we construct a function G consisting of a term proportional to the square of the grid displacement $\mathbf{p}$ plus a constraint enforced with a local Lagrange multiplier $\lambda(x,y)$ (which ensures that the cell volumes of the final grid satisfy a given Jacobian J'/J equal to $\rho(x,y)/\rho'(x',y')$, i.e. equidistribution):

$$G(\mathbf{x},\mathbf{x}',\mathbf{x}'_{xi},\lambda) = \rho(x,y)\frac{(\mathbf{x}'-\mathbf{x})^2}{2} + \lambda(x,y)\left[\rho'(x',y')\left(\frac{\partial x'}{\partial x}\frac{\partial y'}{\partial y} - \frac{\partial x'}{\partial y}\frac{\partial y'}{\partial x}\right) - \rho(x,y)\right]. \tag{12}$$

In a time-stepping context, this leads to moving meshes with minimal grid velocities. This is of interest because excessive grid velocity may be a significant source of error stemming from the grid advective term [38].

The resulting Euler–Lagrange equations corresponding to the minimization of integral (11) with G given by (12) are

$$(x'-x)\rho + \lambda\widehat{J}\frac{\partial\rho'}{\partial x'} = [\lambda\rho', y'], \tag{13}$$

$$(y'-y)\rho + \lambda\widehat{J}\frac{\partial\rho'}{\partial y'} = -[\lambda\rho', x']. \tag{14}$$

(The boundary conditions ensure that the boundary terms due to the integrations by parts are zero.) Here, $\widehat{J} = J'/J$. We can write $\widehat{J} = [x', y']$, and note the relation $[f,g] = \widehat{J}(x,y)[f,g]_{\mathbf{x}'}$, where $[\cdot,\cdot]_{\mathbf{x}'}$ denotes the Poisson bracket taking derivatives with respect to $\mathbf{x}'$. Using this and $\rho(x,y)J(x,y) = \rho'(x'(x,y),y'(x,y))J'(x'(x,y),y'(x,y))$, we deduce

$$x'-x = \frac{\partial\lambda}{\partial x'}, \tag{15}$$

$$y'-y = \frac{\partial\lambda}{\partial y'}, \tag{16}$$

where $\lambda = \lambda(\mathbf{x}(\mathbf{x}'))$. We see that $\mathbf{x} = \psi^{-1}(\mathbf{x}')$ is a gradient with respect to $\mathbf{x}'$,

$$\mathbf{x} = \mathbf{x}' - \nabla_{\mathbf{x}'}\lambda = \nabla_{\mathbf{x}'}(\mathbf{x}'^2/2 - \lambda) = \nabla_{\mathbf{x}'}V(\mathbf{x}'). \tag{17}$$

The inverse $\mathbf{x}' \to \mathbf{x}$ exists because the Jacobian is positive. The inverse of a gradient map is a gradient map, and is in fact given by the Legendre transformation

$$\mathbf{x}' = \nabla U(\mathbf{x}), \tag{18}$$

$$U(\mathbf{x}) = \mathbf{x}\cdot\mathbf{x}'(\mathbf{x}) - V(\mathbf{x}'(\mathbf{x})). \tag{19}$$

(Here and elsewhere, ∇ equals $\nabla_{\mathbf{x}}$.) This fact, that the map $\psi : X \to X$ minimizing the L_2 norm of $\mathbf{p} = \mathbf{x}' - \mathbf{x}$ [with weight $\rho(\mathbf{x})$] is a gradient map, has been noted in the mathematics community [39,40], but we believe that the above is the most accessible derivation. Using $\mathbf{x}\cdot\mathbf{x}' - \mathbf{x}'^2/2 = \mathbf{x}^2/2 - \mathbf{p}^2/2$, we find $U = \mathbf{x}^2/2 - \mathbf{p}^2/2 + \lambda$ or

$$\mathbf{p} = \nabla\Phi \tag{20}$$

with $\Phi = \lambda - \mathbf{p}^2/2$.

Eq. (20), together with the constraint Eq. (6) and the boundary conditions $\mathbf{p}\cdot\mathbf{n} = 0$ on ∂X, define the Monge–Kantorovich optimal displacement approach. The equation holds for $\mathbf{p}$ of arbitrary magnitude, and has an important interpretation in a time-stepping context. In this case, the movement of the grid points by ψ defines a grid velocity proportional to $\mathbf{p}$, and the conclusions above guarantee that ψ moves the grid in an irrotational manner, i.e. it generates no vorticity.

By using Eqs. (20) and (6)–(8), we obtain the *Monge–Ampère equation* for Φ

$$\nabla^2\Phi + \frac{\partial^2\Phi}{\partial x^2}\frac{\partial^2\Phi}{\partial y^2} - \left(\frac{\partial^2\Phi}{\partial x\partial y}\right)^2 = \frac{\rho(x,y)}{\rho'(x',y')} - 1. \tag{21}$$

Note that the solvability condition (5) ensures that the integral of the right hand side is zero. The second and third terms on the left are the determinant of the Hessian matrix $H_{ij} = \partial^2\Phi/\partial x_i\partial x_j$, $i,j = 1,2$. Eq. (21) is to be solved with the boundary conditions of Section 4.3.1, or $\mathbf{n}\cdot\nabla\Phi = 0$, allowing one to obtain the new grid. Note that, unlike most other grid generation approaches, which require as many equations as dimensions, the Monge–Ampère equation is a single nonlinear equation for Φ with no adjustable parameters.

2.3. Direct and inverse approaches

Eq. (21) has two nonlinear aspects, namely the Hessian and the dependence of ρ' on $\mathbf{x}' = \mathbf{x} + \nabla\Phi$. If $|\rho' - 1|$ is large or varies over a short length scale, the latter nonlinearity can lead to numerical difficulties. For cases in which $|\rho - 1|$ is small ($\rho = 1$ is an important special case), it can be less difficult to solve the *inverse problem*. That is, we define $\mathbf{x} = \mathbf{x}' + \nabla\widehat{\Phi}(\mathbf{x}')$, leading to the inverse Monge–Ampère equation

$$\nabla^2_{\mathbf{x}'}\widehat{\Phi} + \frac{\partial^2\widehat{\Phi}}{\partial x'^2}\frac{\partial^2\widehat{\Phi}}{\partial y'^2} - \left(\frac{\partial^2\widehat{\Phi}}{\partial x'\partial y'}\right)^2 = \frac{\rho'(x',y')}{\rho(x,y)} - 1. \tag{22}$$

Again, condition (5) implies that the integral of the right hand side is zero. Let us specialize the discussion for now to the case $\rho(x,y)=1$. In this case, the Hessian is the only source of nonlinearity: The right-hand side does not contribute to the nonlinearity since x' and y' are now independent variables. The solution to Eq. (22) gives $\mathbf{x}$ as a function of $\mathbf{x}'$, i.e. the map ψ^{-1} of Fig. 1. To complete the computation, this map must be inverted numerically to obtain the direct map ψ.

In spite of the advantage of doing the inverse problem for $\rho=1$ (or ρ close to unity), there are two major disadvantages. First, since $\rho(\mathbf{x})$ is (almost) uniform, it makes more sense to discretize the (direct) Monge–Ampère equation on a uniform grid in $\mathbf{x}$. For $|\rho'-1|\sim 1$, particularly if $|\nabla_{\mathbf{x}'}\rho'|$ is large, discretizing Eq. (22) on a uniform grid in $\mathbf{x}'$ is seriously suboptimal, since a very fine uniform mesh will be required to resolve sharp features present in ρ'. Second, the interpolations involved in inverting ψ^{-1} to obtain ψ involve some loss of accuracy and incur some computational cost. Nevertheless, in Section 5.2 we use the inverse approach in order to compare with the deformation method [29], which in its original formulation was designed to solve the inverse problem.

For both direct and inverse approaches, we need to discretize the Monge–Ampère Eq. (21) in the logical (computational) mesh. A very important case where this is of the essence is when the Monge–Ampère equation is solved in a time stepping context, in which both $\rho(x,y)$ and $\rho'(x',y')$ can be far from unity, although ρ/ρ' is close to unity. Such a reformulation, in which all derivatives are with respect to ξ_i, can be readily performed as follows (where we will omit summation over repeated indices). For the *direct* approach, one notices that, in the logical space, $\nabla^2\Phi=\frac{1}{J}\partial_i(Jg^{ij}\partial_j\Phi)$, where $\partial_i\equiv\partial/\partial\xi_i$ and J, g^{ij} are the Jacobian and contravariant metric tensor of the map $\mathbf{x}(\boldsymbol{\xi})$, respectively. For the latter, $g^{ij}=(\partial\xi_i/\partial x_k)(\partial\xi_j/\partial x_k)$. Also, noticing that $[\partial_x\Phi,\partial_y\Phi]=[\xi,\eta][\partial_x\Phi,\partial_y\Phi]_\Xi=J^{-1}[\partial_x\Phi,\partial_y\Phi]_\Xi$, and that $\rho(x,y)=J^{-1}$, we find that the logical representation of the *direct* Monge–Ampère equation reads:

$$\partial_i(Jg^{ij}\partial_j\Phi)+[\partial_x\Phi,\partial_y\Phi]_\Xi=\frac{1}{\rho'(x',y')}-\frac{1}{\rho(x,y)}. \tag{23}$$

Here, the index Ξ indicates that the derivatives in the Poisson bracket are taken in the logical space. The components of $\nabla\Phi$ needed in the Poisson bracket can be expressed in terms of derivatives with respect to ξ_i by writing $\nabla\Phi=\partial_i\Phi\nabla\xi_i$, and using the fact that $\nabla\xi_i$ or $\partial\xi_i/\partial x_j$ is the inverse of $J_{ij}=\partial x_i/\partial\xi_j$. Equivalently, introducting a third dimension z, we have $\nabla\xi_i=\frac{\epsilon_{ij3}}{J}\partial\mathbf{x}/\partial\xi_j\times\hat{e}_z$, where ϵ_{ij3} is the Levi-Civita tensor. A similar argument for the *inverse* Monge–Ampère formulation (22) in terms of the metric tensor and Jacobian of the map $\mathbf{x}'(\xi)$ yields:

$$\partial_i(J'(g')^{ij}\partial_j\Phi)+[\partial_{x'}\Phi,\partial_{y'}\Phi]_\Xi=\frac{1}{\rho(x,y)}-\frac{1}{\rho'(x',y')} \tag{24}$$

with the components of $\nabla_{x'}\Phi$ found in a similar fashion.

3. Properties of Monge–Kantorovich optimization

We now proceed to establish certain important properties of the Monge–Kantorovich approach and the Monge–Ampère equation, Eq. (21). These include the relation with grid distortion (grid smoothness) and ellipticity.

3.1. Connection to optimal grid distortion or smoothness

Our purpose in this section is to establish a connection between the approach based on Monge–Kantorovich optimization and minimization of grid distortion. The latter can be quantified by the functional (11), with G given by:

$$G(\mathbf{x},\mathbf{x}',\mathbf{x}'_{x^i},\mu)=\frac{g_{11}+g_{22}}{2}\rho(x,y)-\mu(x,y)\left[\rho'(x',y')\left(\frac{\partial x'}{\partial x}\frac{\partial y'}{\partial y}-\frac{\partial x'}{\partial y}\frac{\partial y'}{\partial x}\right)-\rho(x,y)\right], \tag{25}$$

where $\mu(x,y)$ is another local Lagrange multiplier, which again enforces equidistribution locally. The *distortion* measure $(g_{11}+g_{22})$ is the trace of the covariant metric tensor, defined as $g=\mathscr{J}^{\mathrm{T}}\mathscr{J}$, where $\mathscr{J}$ is the Jacobi matrix of ψ, defined in Eq. (1). In component notation, $g_{ij}=\mathscr{J}^{\mathrm{T}}_{ik}\mathscr{J}_{kj}$, with repeated indices indicating summation. (That is, the Euclidean distance $\mathrm{d}x'_k\,\mathrm{d}x'_k$ equals $g_{ij}\,\mathrm{d}x_i\,\mathrm{d}x_j$.) The mean distortion in Eq. (25) (whose integral is related to the so-called smoothness measure of Ref. [11]) thus equals

$$g_{11}+g_{22}=\left(\frac{\partial x'}{\partial x}\right)^2+\left(\frac{\partial x'}{\partial y}\right)^2+\left(\frac{\partial y'}{\partial x}\right)^2+\left(\frac{\partial y'}{\partial y}\right)^2. \tag{26}$$

Variational principles based on smoothness have previously been used in grid generation. Examples are in papers by Winslow [9,10], Brackbill and Saltzman [11] and, more recently, Huang [18]. However, smoothness has not been used in the context of using a local Lagrange multiplier to enforce equidistribution exactly. In fact, in these references, a linear combination of the smoothness integral and an equidistribution integral was minimized, so that neither was exactly minimized.

In order to understand the effect of the minimization of G in Eq. (25), it is of interest to consider the eigenvalues of the metric tensor (λ_1 and λ_2), which are related to the elongation of a given computational cell in $\mathbf{x}'$ relative to the corresponding cell in $\mathbf{x}$ ($\lambda_1=\lambda_2$ implies that a square cell maps to a square cell). The trace of the metric tensor is equal to the sum of these eigenvalues,

$$g_{11} + g_{22} = \lambda_1 + \lambda_2, \tag{27}$$

while the local Jacobian constraint implies

$$\lambda_1 \lambda_2 = \left(\frac{\rho}{\rho'}\right)^2. \tag{28}$$

The constrained minimization outlined above can be understood in terms of the local minimization of $\lambda_1 + \lambda_2$ with the constraint $\lambda_1 \lambda_2 = C^2$ (with C a constant), which gives $\lambda_1 = \lambda_2 = C > 0$ (i.e. a square cell). Consequently, it follows that minimizing the trace of the metric tensor constrained by the local cell volume results in the minimal grid cell distortion in $\mathbf{x}'$ relative to $\mathbf{x}$ which is compatible with the equidistribution constraint.

The Euler–Lagrange equations resulting from minimizing (11) with (25) read:

$$\nabla \cdot (\rho \nabla x') + \mu \widehat{J} \frac{\partial \rho'}{\partial x'} = [\mu \rho', y'], \tag{29}$$

$$\nabla \cdot (\rho \nabla y') + \mu \widehat{J} \frac{\partial \rho'}{\partial y'} = -[\mu \rho', x']. \tag{30}$$

(Again the boundary terms obtained from integrating by parts are zero.) We refer to Eqs. (6), (29) and (30), with suitable boundary conditions, as the direct *optimal distortion method*. In Section 5, we will perform numerical calculations on the inverse optimal distortion method, which is obtained from Eqs. (6), (29) and (30) by letting $(\mathbf{x}, \mathbf{x}') \to (\mathbf{x}', \mathbf{x})$, $(\rho, \rho') \to (\rho', \rho)$, and $\widehat{J} \to 1/\widehat{J}$.

Similar manipulations to the ones conducted for Eqs. (15) and (16) lead to

$$\frac{1}{\rho} \nabla \cdot (\rho \nabla x') = \frac{\partial \mu}{\partial x'}, \tag{31}$$

$$\frac{1}{\rho} \nabla \cdot (\rho \nabla y') = \frac{\partial \mu}{\partial y'}, \tag{32}$$

or $\rho^{-1} \nabla \cdot (\rho \nabla \mathbf{x}') = \nabla_{\mathbf{x}'} \mu$. In general $\mathbf{x}'$ is not a gradient map. However, for $\rho = 1 + \mathrm{O}(\varepsilon)$ and $\rho' = 1 + \mathrm{O}(\varepsilon)$, we obtain $\mathbf{x}' = \mathbf{x} + \mathrm{O}(\varepsilon)$ or $\mathbf{p} = \mathrm{O}(\varepsilon)$, and it follows, to lowest order in ε, that

$$\nabla^2 \mathbf{p} = \nabla \mu. \tag{33}$$

Eq. (33) is of higher order than that of the Monge–Kantorovich approach, Eq. (20), and therefore requires more boundary conditions. We have found that specifying both the normal and tangential boundary conditions on $\mathbf{p}$ leads to a well-posed problem.

For $|\varepsilon| \ll 1$, we can establish a relationship between Monge–Kantorovich optimization and minimum distortion optimization. Suppose $\mathbf{p}_1 = \nabla \Phi$ is a solution to Eq. (21), with $\mathbf{p}_1 \cdot \mathbf{n} = 0$ on ∂X. Suppose further that we have a solution of Eq. (33) with $\nabla^2 \mathbf{p}_2 = \nabla \mu$ and with $\mathbf{p}_2 = \mathbf{p}_1$ on ∂X. In particular, this means that $\mathbf{p}_2 \cdot \mathbf{n} = 0$ on ∂X, but the tangential component $\mathbf{p}_2 \cdot \mathbf{t} \neq 0$ is specified also on ∂X ($\mathbf{t}$ being the unit vector tangential to the boundary). Defining $\mu = \nabla^2 \Phi$, we find $\nabla^2 [\mathbf{p}_2 - \nabla \Phi] = \nabla^2 [\mathbf{p}_2 - \mathbf{p}_1] = 0$, with $\mathbf{p}_2 - \mathbf{p}_1$ equal to zero on the boundary. We conclude that $\mathbf{p}_2 = \mathbf{p}_1$ on X. Therefore, the Monge–Kantorovich optimization solution $\mathbf{p}_1 = \nabla \Phi$ is also a solution to the optimal distortion method for $|\varepsilon| \ll 1$, and Eq. (21) will produce optimally smooth grids. However, since the optimal distortion method requires $\mathbf{p}$ (and not only $\mathbf{p} \cdot \mathbf{n}$) to be specified on ∂X, there are other solutions to the optimal distortion problem, with varying degrees of mean distortion Eq. (26). For example, as we shall see in Section 5.1 for an example with $\varepsilon \sim \mathrm{O}(1)$, the Monge–Kantorovich solution solution $\mathbf{p}_1$ with $\mathbf{p}_1 \cdot \mathbf{n} = 0$ on ∂X (which would be a minimum distortion solution for ε small) has less distortion near the boundary than the minimum distortion solution with $\mathbf{p}_2 = 0$ on ∂X.

3.2. Ellipticity of the linearized PDE

We proceed to show that the linearized equidistribution PDE, Eq. (21), is elliptic. This is of relevance for the applicability of multigrid methods in the solver algorithm (Section 4.2). As obtained in Section 2, the equidistribution nonlinear PDE reads:

$$F[\Phi] = 1 + \nabla^2 \Phi + \left[\frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial y}\right] = \frac{\rho(\mathbf{x})}{\rho'(\mathbf{x}')}. \tag{34}$$

Taking $\Phi = \Phi_0 + \delta\Phi$ and linearizing with respect to $\delta\Phi$, there results the linear operator:

$$L[\delta\Phi] = \nabla^2 \delta\Phi + \left[\frac{\partial \delta\Phi}{\partial x}, \frac{\partial \Phi_0}{\partial y}\right] + \left[\frac{\partial \Phi_0}{\partial x}, \frac{\partial \delta\Phi}{\partial y}\right].$$

Here, Φ_0 defines a mapping $\mathbf{x}'_0 = \mathbf{x} + \nabla \Phi_0$. Notice that the terms associated with the linearization of $\rho'(x', y')$ are disregarded since they involve only first derivatives of Φ and therefore do not affect the definition of ellipticity. Expanding the Poisson brackets, we find:

$$L(\delta\Phi) = \left(1 + \frac{\partial^2\Phi_0}{\partial y^2}\right)\frac{\partial^2\delta\Phi}{\partial x^2} + \left(1 + \frac{\partial^2\Phi_0}{\partial x^2}\right)\frac{\partial^2\delta\Phi}{\partial y^2} - 2\frac{\partial^2\Phi_0}{\partial x\partial y}\frac{\partial^2\delta\Phi}{\partial x\partial y}, \tag{35}$$

which can be cast in the standard quadratic form $A\partial^2\delta\Phi/\partial x^2 + C\partial^2\delta\Phi/\partial y^2 + 2B\partial^2\delta\Phi/\partial x\partial y$, with $A = (1 + \partial^2\Phi_0/\partial y^2)$, $C = (1 + \partial^2\Phi_0/\partial x^2)$, and $B = -\partial^2\Phi_0/\partial x\partial y$. The linear PDE in Eq. (35) is elliptic if and only if $AC - B^2 > 0$ [34]. However, it is straightforward to show that $AC - B^2 = F[\Phi_0] = \rho(\mathbf{x})/\rho'(\mathbf{x}'_0)$, which is positive by definition and so the ellipticity condition is satisfied. In most cases of interest, Φ_0 will correspond to a previous time step (in dynamic computations) or to a previous Newton iterate.

4. Numerical implementation

In this section, we discuss the details of the numerical implementation of the Monge–Kantorovich approach, Eq. (21). Here and in Section 5, for simplicity, we will consider $X = \Xi$ and the identity mapping between the logical space Ξ and the initial grid in physical space: $\mathbf{x} = \xi$. This implies $J = 1$.

4.1. Discretization of the equidistribution PDE, Eq. (21)

For the discretization of Eq. (21), we place Φ at the cell centers of the uniform logical grid and use ghost cells to enforce the boundary conditions. The Laplacian is discretized according to the usual 5-point stencil:

$$\nabla^2\Phi|_{i,j} = \frac{\Phi_{i+1,j} - 2\Phi_{i,j} + \Phi_{i-1,j}}{\Delta x^2} + \frac{\Phi_{i,j-1} - 2\Phi_{i,j} + \Phi_{i,j-1}}{\Delta y^2}, \tag{36}$$

where i and j label the x and y position of a generic cell center and $\Delta x, \Delta y$ are the width and height of each cell in the uniform initial grid (logical grid), respectively. The Hessian term in Eq. (21) (which can also be expressed as $[\partial_x\Phi, \partial_y\Phi]$) contains cross derivatives, and therefore requires a 9-point stencil for its discretization. For this, we compose two discrete first-order derivatives as follows. First, we define first-order derivatives at vertices $(i \pm 1/2, j \pm 1/2)$ as

$$\left.\frac{\partial\Phi}{\partial x}\right|_{i+1/2,j+1/2} \approx \frac{\Phi_{i+1,j} + \Phi_{i+1,j+1} - \Phi_{i,j} - \Phi_{i,j+1}}{2\Delta x}, \tag{37}$$

$$\left.\frac{\partial\Phi}{\partial y}\right|_{i+1/2,j+1/2} \approx \frac{\Phi_{i,j+1} + \Phi_{i+1,j+1} - \Phi_{i,j} - \Phi_{i+1,j}}{2\Delta y}. \tag{38}$$

These are introduced in similarly defined first-order derivatives at cell centers (i,j) (found by replacing $i \to i - 1/2$ and $j \to j - 1/2$ in the expressions above), to obtain the 9-point stencil discretization sought.

4.2. Newton–Krylov solver with multigrid preconditioning

In order to solve the nonlinear equation of the Monge–Kantorovich approach, Eq. (21), we use a nonlinear inexact Newton–Krylov solver. That is, we solve the nonlinear system (21), the discretized form of which is $\mathbf{G}(\mathbf{\Phi}) = \mathbf{0}$ (where $\mathbf{\Phi}$ is the vector containing the values of $\Phi(x,y)$ at cell centers). This is performed iteratively by solving successive linear systems of the form:

$$\left.\frac{\partial\mathbf{G}}{\partial\mathbf{\Phi}}\right|_k \delta\mathbf{\Phi}_k = -\mathbf{G}(\mathbf{\Phi}_k) \tag{39}$$

with $\mathbf{\Phi}_{k+1} = \mathbf{\Phi}_k + \beta\delta\mathbf{\Phi}_k$. The parameter β ($\leqslant 1$) damps the Newton update to extend the domain of convergence of Newton's method convergence in the face of very nonlinear systems. Here, β is determined using the Armijo rule [41].

Nonlinear convergence is determined by:

$$\|\mathbf{G}(\mathbf{\Phi}_k)\|_2 < \epsilon_a + \epsilon_r\|\mathbf{G}(\mathbf{\Phi}_0)\|_2 = \epsilon_t, \tag{40}$$

where $\|\cdot\|_2$ is the $\mathscr{L}_2$-norm (euclidean norm), $\epsilon_a = \sqrt{N}\times 10^{-15}$ (with N the total number of degrees of freedom, $N = n_x n_y$) is an absolute tolerance to avoid trying to converge to below roundoff, ϵ_r is the Newton relative convergence tolerance (set to 10^{-4} in this work), and $\mathbf{G}(\mathbf{\Phi}_0)$ is the initial residual. For our purposes, we use the identity $\mathbf{x}'(\mathbf{x}) = \mathbf{x}$, or $\mathbf{\Phi} = 0$, as the initial guess.

Such linear systems are solved iteratively with Krylov methods, which only require matrix–vector products to proceed. Because the linear system matrix is a Jacobian matrix, such matrix–vector products can be implemented Jacobian-free using the Gateaux derivative:

$$\left.\frac{\partial\mathbf{G}}{\partial\mathbf{\Phi}}\right|_k \mathbf{v} = \lim_{\epsilon\to 0}\frac{\mathbf{G}(\mathbf{\Phi}_k + \epsilon\mathbf{v}) - \mathbf{G}(\mathbf{\Phi}_k)}{\epsilon}, \tag{41}$$

where in practice a small but finite ϵ is employed [41]. Thus, the evaluation of the Jacobian-vector product only requires the function evaluation $\mathbf{G}(\mathbf{\Phi}_k + \epsilon\mathbf{v})$, and there is no need to form or store the Jacobian matrix.

An inexact Newton method [42] is used to adjust the convergence tolerance of the Krylov method at every Newton iteration according to the size of the current Newton residual, as follows:

$$\|J_k\delta\mathbf{\Phi}_k + \mathbf{G}(\mathbf{\Phi}_k)\|_2 < \eta_k\|\mathbf{G}(\mathbf{\Phi}_k)\|_2, \tag{42}$$

where η_k is the inexact Newton parameter and $J_k = \frac{\partial \mathbf{G}}{\partial \mathbf{\Phi}}|_k$ is the Jacobian matrix. Thus, the convergence tolerance of the Krylov method is loose when the Newton state vector $\mathbf{\Phi}_k$ is far from the nonlinear solution, but tightens as $\mathbf{\Phi}_k$ approaches the solution. Hence, the linear solver works the hardest when the Newton state vector is closest to the nonlinear root. Superlinear convergence rates of the inexact Newton method are possible if the sequence of η_k is chosen properly [41]. Here, we employ the same prescription as in [43]:

$$\begin{aligned}
\eta_k^A &= \gamma\left(\frac{\|\mathbf{G}(\mathbf{\Phi}_k)\|_2}{\|\mathbf{G}(\mathbf{\Phi}_{k-1})\|_2}\right)^{\alpha},\\
\eta_k^B &= \min[\eta_{\max}, \max(\eta_k^A, \gamma\eta_{k-1}^{\alpha})],\\
\eta_k &= \min\left[\eta_{\max}, \max\left(\eta_k^B, \gamma\frac{\epsilon_t}{\|\mathbf{G}(\mathbf{\Phi}_k)\|_2}\right)\right],
\end{aligned}$$

with $\alpha = 1.5$, $\gamma = 0.9$, and $\eta_{\max} = 0.9$. The convergence tolerance ϵ_t is defined in Eq. (40). In this prescription, the first step ensures superlinear convergence (for $\alpha > 1$), the second avoids volatile decreases in η_k, and the last avoids oversolving in the last Newton iteration. We also use a quadratic line-search backtracking algorithm [41] for added robustness of the nonlinear solver.

A further advantage of Krylov methods is that they can be preconditioned by considering the alternate systems $J_kP_k^{-1}P_k\delta\mathbf{\Phi}_k = -\mathbf{G}_k$ (right preconditioning) or $P_k^{-1}J_k\delta\mathbf{\Phi}_k = -P_k^{-1}\mathbf{G}_k$ (left preconditioning). Such a preconditioning step can be straightforwardly and efficiently implemented in the Krylov algorithm as two consecutive matrix–vector products, and has the potential of substantially improving the convergence properties of the Krylov iteration if $P_k^{-1} \approx J_k^{-1}$. Here, we use multigrid (MG) right preconditioning, which has been shown in many applications [20,44–49,43] to deliver optimal, scalable convergence rates. In fact, for some of the examples considered in Section 5, we have observed that the MG-preconditioned approach results in a two order of magnitude improvement in the iteration count and an order of magnitude speedup in the CPU time vs. the unpreconditioned one. While multigrid methods as solvers are very sensitive to the details of the smoother and the restriction and prolongation operators [50], as preconditioners they have shown remarkable robustness even with low-order interpolation operators [44–48,51]. Here, we use a V(4,4) multigrid cycle with damped Jacobi as smoother (with damping parameter $\omega = 0.7$ unless otherwise noted), agglomeration for restriction, and bilinear prolongation.

The ellipticity property of the linearized equidistribution PDE Eq. (21) guarantees the stability of the Jacobi smoother. Indeed, it can be shown that, for the linearized version of Eq. (21), namely Eq. (35), the spectral radius r of the Jacobi iteration matrix is $|r| < 1$ when:

$$|2 + \nabla^2\Phi_0| > 2|\partial^2\Phi_0/\partial x\partial y|, \tag{43}$$

where Φ_0 corresponds to the previous Newton iterate on which the linearization is conducted. Alternatively, we can write Eq. (43) as

$$|A + C| > 2|B| \tag{44}$$

with A, B, C defined in Section 3.2. Squaring Eq. (44) and subtracting $4AC$ results in

$$(A - C)^2 > 4(B^2 - AC). \tag{45}$$

The ellipticity property of the linearized equidistribution PDE (Section 3.2) implies that the right hand side of the equation above is always negative, thereby ensuring that the inequality is always satisfied. This proves that the spectral radius of the Jacobi iteration matrix will always be smaller than unity, and therefore that a damped Jacobi iteration can be an effective multigrid smoother.

4.3. Numerical implementation issues

There are several important numerical details that have the potential of derailing the viability of a given solution algorithm of the Monge–Kantorovich equation. In the context of our finite-difference implementation, these are: (1) the numerical treatment of the boundary conditions $\mathbf{n}\cdot\nabla\Phi = 0$, and (2) the numerical enforcement of the solvability condition Eq. (5). We proceed to discuss these in some detail.

4.3.1. Boundary conditions

Our finite-difference implementation places finite-volume faces at physical boundaries. The unknown potential Φ is placed at cell centers. The boundary conditions $\mathbf{n}\cdot\nabla\Phi = 0$ are imposed via ghost cells. Ghost cells across boundary faces are filled trivially by extrapolation using this homogeneous Neumann boundary condition. In practice, this simply implies that, taking the $x = 1$ boundary face, $\Phi_{n_x+1,j} = \Phi_{n_x,j}$, and similarly for other faces. Here, n_x and n_y are the number of cells in each direction and the boundary face $x = 1$ corresponds to $i = n_x + 1/2$. Similarly, the face $x = 0$ has $i = 1/2$. However, since

the discretization of the Monge–Kantorovich equation requires a 9-point stencil, corner ghost cells need to be filled as well. To fill these, we have found it crucial to enforce that the value in the ghost corner be equal to the corner value within the domain (e.g. for the northwest corner, we have $\Phi_{n_x+1,n_y+1} = \Phi_{n_x,n_y}$). It is easy to show that, with this choice and the previous recipe for face ghost cells, the following consistency condition is enforced at the corners of the domain:

$$\Phi_{n_x+1,n_y+1} = \Phi_{n_x+1,n_y} = \Phi_{n_x,n_y+1} = \Phi_{n_x,n_y}.$$

In essence, this enforces the corners to remain fixed.

4.3.2. Solvability condition

The solvability condition Eq. (5) states that the physical volume mapped by $\mathbf{x}'$ and $\mathbf{x}$ be the same. The implications of this normalization requirement are different depending on whether one is using a direct or an inverse Monge–Kantorovich formulation (Section 2.3), assuming that $\rho(x,y)$ equals unity. In an *inverse* formulation, where $\mathbf{x} \leftrightarrow \mathbf{x}'$ in Eq. (5), the right-hand side of Eq. (22), $\rho'(x',y') - 1$, depends only on the independent variables x', y'. In this case, the solvability condition is automatically satisfied by the density normalization condition, Eq. (4), and nothing changes during the nonlinear iterations. In a *direct* formulation, the right-hand side of Eq. (21), $1/\rho'(x',y') - 1$, depends on the new variables, i.e. on Φ, so that the solvability condition is a constraint on the mapping $\mathbf{x}'(\mathbf{x})$, i.e. on Φ. In order to take this solvability contraint into account and to improve convergence of the solver for the direct case, we have normalized the right-hand side of Eq. (21) by letting $1/\rho'(x',y') - 1 \to C/\rho'(x',y') - 1$ and finding C so that Eq. (5) is satisfied for each nonlinear function call. Although convergence may be achieved without this renormalization, we have found that this renormalization improves the convergence and robustness of the solver appreciably. By performing this normalization at every nonlinear call, the solver corrects for small changes in the normalization constant due to numerical errors (small changes in the total volume) which occur as the map evolves. While performing the normalization procedure at every function evaluation increases the cost per call, the corresponding reduction in linear and nonlinear iterations easily offsets this increase.

4.4. Equidistribution diagnostic

In order to check the enforcement of the equidistribution constraint, one must define the discretized volumes of the deformed cells. Again, we assume $\rho(x,y) = J(x,y) = 1$, i.e. the old physical grid and the logical grid are identical.

Since the shape of the new cell is deformed, we numerically compute the Jacobian $\det[\nabla \mathbf{x}']$ using the four-point approximation [52–54]. For a given cell in the new grid coordinate system, let us denote the four vertices of the cell by (x_i', y_i'), $i = 1,2,3,4$ counterclock-wise from the lower left corner. See Fig. 2. We compute the derivatives at each cell center by

$$\frac{\Delta x'}{\Delta x} = \frac{(x_2' - x_1' + x_3' - x_4')}{2\Delta x}, \tag{46}$$

$$\frac{\Delta x'}{\Delta y} = \frac{(x_4' - x_1' + x_3' - x_2')}{2\Delta y}, \tag{47}$$

$$\frac{\Delta y'}{\Delta x} = \frac{(y_2' - y_1' + y_3' - y_4')}{2\Delta x}, \tag{48}$$

$$\frac{\Delta y'}{\Delta y} = \frac{(y_4' - y_1' + y_3' - y_2')}{2\Delta y}. \tag{49}$$

Then we compute the numerical Jacobian at the center of the cell as

$$J_{\text{num}} = \frac{\Delta x'}{\Delta x}\frac{\Delta y'}{\Delta y} - \frac{\Delta x'}{\Delta y}\frac{\Delta y'}{\Delta x}, \tag{50}$$

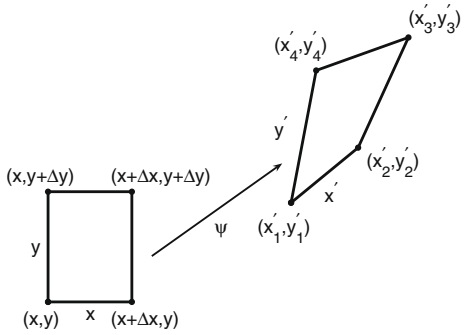


Fig. 2. Sketch of the mapping transformation for a single cell. The spatial discretization adopted in Eqs. (46)–(49) and the definition of the numerical Jacobian, Eq. (50), are such that $J_{\text{num}}\Delta x\Delta y$ equals the area of the polygon connecting the four vertices of the deformed cell.

and $J_{\text{num}}\Delta x\Delta y$ is the cell volume in 2D. Notice that, with this choice of the discretization, the area of the deformed cell corresponds to the natural geometrical area of the polygon connecting the four vertices of the cell.

4.5. Implementation algorithm

Here we summarize the implementation algorithm for the nonlinear solution of the Monge–Kantorovich approach, Eq. (21). For notation, see Section 4.2.

(1) Start with a guess $\mathbf{\Phi} = \mathbf{\Phi}_0$, the values of Φ at cell centers.
(2) Evaluate the nonlinear residual, $\mathbf{G}(\mathbf{\Phi}_k)$.
(3) Check convergence [Eq. (40)].
(4) If not converged, compute the Newton update $\delta\mathbf{\Phi}_k$ by solving $J_k\delta\mathbf{\Phi}_k = -\mathbf{G}(\mathbf{\Phi}_k)$.
(5) Compute the new Newton state $\mathbf{\Phi}_{k+1} = \mathbf{\Phi}_k + \beta\delta\mathbf{\Phi}_k$. Go back to (2).
(6) If converged, compute the vertices of the new grid $\mathbf{x}' = \mathbf{x} + \nabla\Phi$ by using Eqs. (37) and (38).

In order to solve iteratively for the Newton update using Krylov methods, the evaluation of the residual $\mathbf{G}(\mathbf{\Phi})$ is essential. We report its algorithm below:

(1) Given $\mathbf{\Phi}$, fill the ghost cells as discussed in Section 4.3.1.
(2) Compute the vertices of the new grid $\mathbf{x}' = \mathbf{x} + \nabla\Phi$ by using Eqs. (37) and (38).
(3) Compute the cell centers of the new grid as

$$\mathbf{x}'_{ij} = \frac{\mathbf{x}'_{i+1/2,j+1/2} + \mathbf{x}'_{i+1/2,j-1/2} + \mathbf{x}'_{i-1/2,j+1/2} + \mathbf{x}'_{i-1/2,j-1/2}}{4}. \tag{51}$$

(4) Compute $\rho'(\mathbf{x}')$ at cell centers. Apply renormalization of Section 4.3.2.
(5) Obtain $\mathbf{G}(\mathbf{\Phi})$, the discretized form of $\nabla^2\mathbf{\Phi} + \left[\frac{\partial\mathbf{\Phi}}{\partial x}, \frac{\partial\mathbf{\Phi}}{\partial y}\right] - \frac{\rho(\mathbf{x})}{\rho'(\mathbf{x}')} + 1$, at cell centers (the discretization of the first two terms is discussed in Section 4.1).

5. Results

In this Section, we will apply the Monge–Kantorovich approach to several challenging tests. First, we will solve the inverse problem, Eq. (22), to compare the Monge–Kantorovich approach to the optimal distortion method (inverse approach), in order to test the connection between the two approaches established in Section 3.1 for an example where $\varepsilon \sim \mathrm{O}(1)$.

Second, we will compare the performance of the Monge–Kantorovich approach against the deformation method of Liao and Anderson [29]. (Again we will use the inverse formulation for the Monge–Kantorovich approach, in order to compare with the deformation method, which in its original formulation could deal only with the inverse problem, i.e. the Jacobian specified as $J(x, y)$ [29].) This method (reviewed in the Appendix) finds cell-area equidistribution by requiring the grid to be generated by means of a specified flow. The deformation procedure is designed specifically to take advantage of certain ODE theorems to prove that, once a consistent flow is chosen, the mapping is one-to-one. See the Appendix for more details. The method requires the solution of a second order system of time-dependent ODEs [the initial value problem defined by Eqs. (A.4), (A.5)] with a given velocity field [Eqs. (A.6) and (A.7)]. To solve numerically these equations, we use a second-order Runge–Kutta method. This is a very simple – and not very accurate – method, but we will use small time steps so that typically grid errors dominate most of the computations.

Finally, we will solve the direct Monge–Ampère equation, Eq. (21), for three very challenging examples.

All the methods have been coded in Fortran 90 and results, including accuracy and performance, are obtained with a 2.4 GHz Intel Xeon processor.

In order to check the accuracy of the various methods, we will compare the numerical Jacobian in the new coordinate system to the prescribed Jacobian, namely, we test if $J' = J_{\text{num}}$ at the cell centers (recall that $\mathbf{x} = \xi$ so that $J = 1$). For this purpose, we define the total error of the scheme as:

$$\text{Error} = \sqrt{\sum_{i,j} |J_{\text{num}} - J'|^2_{i,j}\Delta x\Delta y}. \tag{52}$$

We will also assess the quality of the adapted grid by measuring the global displacement of the grid points [according to Eq. (12)]

$$\|\mathbf{p}\|_2 = \sqrt{\sum_{i,j} (\rho|\mathbf{p}|^2)_{i,j}\Delta x\Delta y}, \tag{53}$$

as well as the global distortion of the cells [according to Eq. (25)]

$$\|g_{11}+g_{22}\|_1=\sum_{i,j}(\rho|g_{11}+g_{22}|)_{ij}\Delta x\Delta y. \tag{54}$$

5.1. Comparison of the Monge–Kantorovich approach with the optimal distortion method

First, we qualitatively compare the Monge–Kantorovich approach against the optimal distortion method, without the assumption $\varepsilon \ll 1$ (Section 3.1). In both cases, we solve the inverse problem. For the Monge–Kantorovich approach, we solve Eq. (22) to obtain the mapping $\mathbf{x}(\mathbf{x}')$ and then invert it numerically to obtain $\mathbf{x}'(\mathbf{x})$. For the inverse optimal distortion method, we solve Eqs. (6), (29) and (30) with $(\mathbf{x},\mathbf{x}')\to(\mathbf{x}',\mathbf{x})$, $(\rho,\rho')\to(\rho',\rho)$ and $\widehat{J}\to 1/\widehat{J}$ and again invert the mapping $\mathbf{x}(\mathbf{x}')$ numerically. We consider $\rho(x,y)=1$ and the following functional form of the density function on the new grid

Example 1

$$\rho'(x',y')=\frac{C}{2+\cos[8\pi\sqrt{(x'-0.5)^2+(y'-0.5)^2}]}, \tag{55}$$

where C is the normalization factor such that Eq. (4) is satisfied. Note that $\varepsilon\sim|\rho'-1|$ is of order unity (see Fig. 3).

The results for the Monge–Kantorovich approach are presented in Table 1. By looking at the equidistribution error as a function of $N=n_x\times n_y$ (with $n_x=n_y$), one notices that the error scales as $\Delta x^2\sim n_x^{-2}$, consistent with the truncation error of the numerical scheme. Also, note that the method converges with practically the same number of linear and nonlinear iterations for all the grids considered. This is consistent with the observation (third column) that the scaling of the CPU time (ignoring the small cost of inversion) with the total number of grid cells N is linear (optimal).

Fig. 4 shows the grid generated by the Monge–Kantorovich approach (solid line) and an optimal distortion method with $\mathbf{p}=0$ on ∂X (dashed line) for a 32×32 grid. First of all, note that the grids are finer in the regions in Fig. 3 where ρ' is larger. It is also clear that the two grids are practically indistinguishable in the interior region, but noticeably different near the four corners. These differences can be attributed to the $\mathbf{p}=0$ boundary condition (fixed boundary points). For the Monge–Kantorovich method, only the normal component of $\mathbf{p}$ is zero. These boundary conditions require the adapted grid to curve noticeably near the boundary, much more so than the Monge-Kantorovich grid. (We have also checked numerically that,

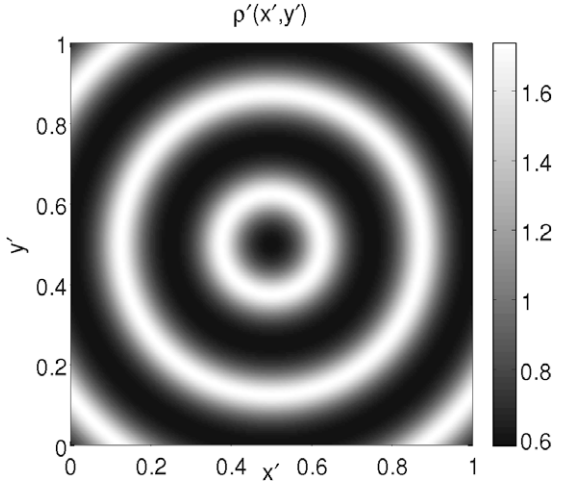


Fig. 3. $\rho'(x',y')$ for Example 1. Note that $\varepsilon\sim|\rho'-1|$ is of order unity, and the ratio $\rho'_{\max}/\rho'_{\min}$ is 3.

Table 1
Example 1, inverse formulation

Number of cells	Error	CPU time (s)	$\|\mathbf{p}\|_2^{MK}$	$\|g_{11}+g_{22}\|_1^{MK}$	Newton iterations	GMRES iterations
16×16	9.64×10^{-2}	0.1	0.0173	1.449	3	3
32×32	2.28×10^{-2}	0.4	0.0173	1.466	4	4
64×64	5.78×10^{-3}	1.3	0.0173	1.470	4	4
128×128	1.46×10^{-3}	4.9	0.0174	1.470	4	4
256×256	3.67×10^{-4}	19	0.0174	1.471	4	4

Grid convergence study for the Monge–Kantorovich approach, reporting the equidistribution error, the CPU time (ignoring the cost of inversion), the grid quality measures $\|\mathbf{p}\|_2^{MK}$ and $\|g_{11}+g_{22}\|_1^{MK}$, and the number of linear and nonlinear iterations as functions of $N=n_x\times n_y(n_x=n_y)$.

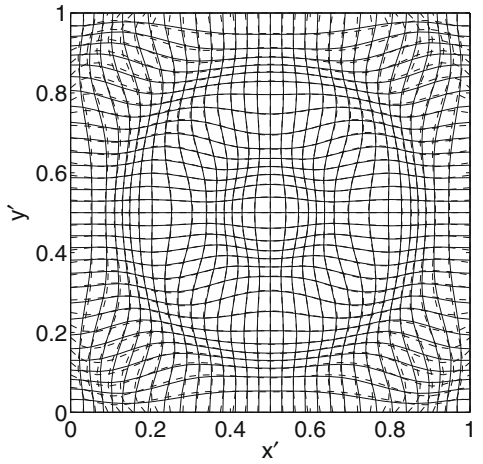


Fig. 4. Example 1, inverse formulation. Comparison of the new grid configuration for ρ' as in Eq. (55) and Fig. (3), obtained by the Monge–Kantorovich approach (solid line) and the optimal distortion method (dashed line). The grid has $n_x = n_y = 32$.

when p_x and p_y on ∂X obtained from the Monge–Kantorovich approach are used as boundary conditions for the optimal distortion method, the grids generated with the two methods are essentially identical.) We also notice that the newly generated grids are symmetric about $x' = 0.5$, about $y' = 0.5$, and about the 45° axes, as required by the symmetry properties held by the density ρ' and the boundary. Overall, we can conclude that the two approaches produce very similar grids despite the fact that this example has $\varepsilon \sim O(1)$. This is consistent with the results of Section 3.1, and suggests that even for $\varepsilon \sim O(1)$, the Monge–Kantorovich approach very nearly minimizes the grid distortion.

5.2. Comparison of the Monge–Kantorovich approach with the deformation method

Second, we compare the Monge–Kantorovich approach with the deformation method of Liao and Anderson [29] in terms of accuracy [Eq. (52)], quality of the grid defined by Eqs. (53) and (54), and computational time needed to achieve the solution. Again we consider the inverse problem, Eq. (22). To the best of our knowledge, the deformation method is the only other method in the literature that is designed specifically to satisfy the equidistribution constraint. The method is reviewed in the Appendix, to which we refer the interested reader for some details on our numerical implementation.

In order to cast the results in the correct perspective, it is useful to identify the sources of error associated with the implementation of the deformation method. These are:

- the spatial discretization of the integrals (A.6) and (A.7), for which we use a second-order numerical scheme;
- the temporal discretization of the ODE system (A.4), for which we use the second-order Runge–Kutta method;
- the spatial interpolation from the reference grid to the time-dependent physical grid (see the Appendix), for which we use cubic splines. This source of error is generally negligible since it is of higher order than the spatial discretization of Eqs. (A.6) and (A.7).

Since we are using an inverse formulation, which obtains ψ^{-1}, there is an additional source of interpolation error in obtaining ψ, i.e. $\mathbf{x}'(\mathbf{x})$. We ignore this source of error because it is in common with the inverse Monge–Kantorovich approach.

We investigate these sources of error by performing numerical experiments on Example 1 for differing number of cells in the computational domain and by changing the temporal time step Δt. Results are presented in Tables 2 and 3, where we report the error between the numerical and analytical Jacobian [as defined in Eq. (52)], some measures of the quality of the grid, and the computational time needed to converge to the desired grid. By inspection of Table 2 ($\Delta t = 0.1$), one notices

Table 2
Example 1, inverse formulation

Number of cells	Error	CPU time (s)	$\frac{\|\mathbf{p}\|_2}{\|\mathbf{p}\|_2^{Mk}} - 1$ (%)	$\frac{\|g_{11}+g_{22}\|_1}{\|g_{11}+g_{22}\|_1^{Mk}} - 1$ (%)
16×16	1.16×10^{-1}	0.02	+24	+1
32×32	3.54×10^{-2}	0.1	+28	+2
64×64	9.89×10^{-3}	0.4	+30	+3
128×128	2.84×10^{-3}	1.5	+30	+3
256×256	1.18×10^{-3}	6	+30	+3

Grid convergence study for the deformation method with $\Delta t = 0.1$. The grid quality measures $\|\mathbf{p}\|_2$ and $\|g_{11} + g_{22}\|_1$ are expressed relative to the corresponding quantities obtained with the Monge–Kantorovich approach shown in Table 1.

Table 3
Example 1, inverse formulation

Number of cells	Error	CPU time (s)	$\frac{\|\mathbf{p}\|_2}{\|\mathbf{p}\|_2^{MK}} - 1$ (%)	$\frac{\|g_{11}+g_{22}\|_1}{\|g_{11}+g_{22}\|_1^{MK}} - 1$ (%)
16×16	1.16×10^{-1}	0.2	+24	+1
32×32	3.53×10^{-2}	0.9	+28	+2
64×64	9.64×10^{-3}	3.4	+30	+3
128×128	2.46×10^{-3}	13.6	+30	+3
256×256	6.21×10^{-4}	55	+30	+3

Grid convergence study for the deformation method with $\Delta t = 0.01$. The grid quality measures $\|\mathbf{p}\|_2$ and $\|g_{11} + g_{22}\|_1$ are expressed in terms of variation with respect to the corresponding quantities obtained with the Monge–Kantorovich approach in Table 1.

that the error decreases a little more slowly than n_x^{-2} when the number of grid points is doubled in each direction from the 16×16 grid to the 128×128 grid. However, from the 128×128 to the 256×256 grid the error decreases only by a factor of 2.4 and remains roughly constant for more refined grids (not shown). This indicates that the spatial discretization error is the dominant source of error for $n_x \leqslant 128$, but for the more refined grids the temporal discretization error is important. At the same time, by looking at Table 3 (obtained with $\Delta t = 0.01$), it is clear that the error decreases roughly as n_x^{-2} for all the cases considered. Therefore we can conclude that $\Delta t = 0.01$ is small enough to make the temporal discretization error negligible for the grids considered in this Table. We mention at this point that we did not find any equidistribution errors or performance comparisons reported in the literature for the deformation method. By comparing Table 1 with Tables 2 and 3, we see that the Monge–Kantorovich approach and the deformation method are comparable (for equal grid size) in terms of equidistribution error. Another observation from Tables 2 and 3 is that the computational time (ignoring the cost of inversion) needed to obtain the desired grid scales linearly (optimally) with the number of grid points. Notice also that, for the 256×256 grid, the deformation method with $\Delta t = 0.01$ is roughly three times slower than the multigrid preconditioned Monge–Kantorovich approach (Table 1).

For comparison, in Fig. 5 we have superimposed the 32×32 grids obtained by the Monge–Kantorovich approach (solid line) and the deformation method (dashed line). Clearly, the differences in the grid are much larger than the corresponding differences in the equidistribution error, which says that the two methods do not converge to the same grid. By looking at the fourth and fifth columns of Table 3 for the deformation method (expressed in terms of variation with respect to the values obtained with the Monge–Kantorovich approach), we see that the Monge–Kantorovich approach produces better quality grids. For instance, for the 128×128 grid, the norm of the displacement of the grid points is about 30% more while the overall distortion of the grid is about 3% more for the deformation method (which is significant given the smoothness of the target grid density; below we consider a much more challenging case which will show far more distortion with the deformation method). These differences are evident in Fig. 5. It is not surprising that the deformation method does not produce optimal grids, since it is not formulated in terms of a variational principle.

Third, we compare the Monge–Kantorovich approach (inverse method) and the deformation method (with $\Delta t = 0.01$) for the following very challenging example

Example 2

$$\rho'(x', y') = C\left[1 + \frac{9}{1 + [10r\cos(\theta - 20r^2)]^2}\right] \tag{56}$$

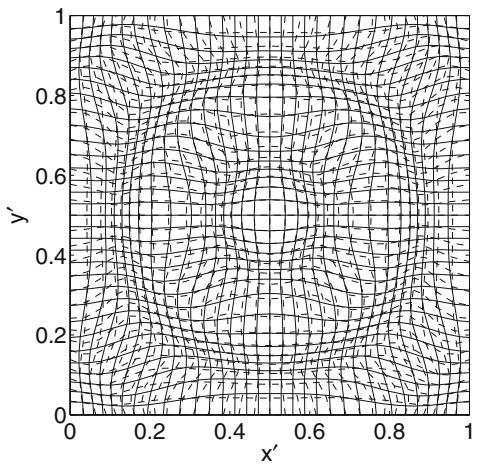


Fig. 5. Example 1, inverse formulation. Comparison of the new grid configuration obtained by the Monge–Kantorovich approach (solid line) and the deformation method (dashed line, $\Delta t = 0.01$) using 32×32 cells.

with

$$r = \sqrt{(x' - 0.7)^2 + (y' - 0.5)^2},$$
$$\theta = \tan^{-1}\left(\frac{y' - 0.5}{x' - 0.7}\right). \tag{57}$$

This example corresponds to a spiral centered at (0.7,0.5) with very tight arms (see Fig. 6). In this case, the ratio between the maximum and minimum values of $\rho'(x', y')$ is about 9. Table 4 shows the grid convergence study for the Monge–Kantorovich approach. Notice that the equidistribution error for coarse grids is quite large, and does not scale quadratically with doubling the number of grid points in each direction until the grid is sufficiently refined. For instance, from the 16×16 grid to the 32×32 grid the error decreases only by a factor of 1.9, while from the 256×256 grid to the 512×512 grid the error decreases by a factor of 4.1. The computational time (ignoring the cost of inversion) needed for convergence, however, does scale optimally with the number of grid cells, $\propto N$, consistent with a constant number of linear and nonlinear iterations for all the grids considered. Fig. 7 (left) shows the grid $\mathbf{x}(\mathbf{x}')(\psi^{-1})$ obtained by the Monge–Kantorovich approach for the 64×64 grid. That is, the grid lines in $\mathbf{x}$ are obtained by applying ψ^{-1} to a uniform grid in $\mathbf{x}'$. Fig. 7 (right) shows the grid obtained by inverting the $\mathbf{x}(\mathbf{x}')$ mapping, and showing the image under ψ in $\mathbf{x}'$ of a uniform grid in $\mathbf{x}$. Clearly the grid on the right is finer in regions of higher density ρ' in Fig. 6.

We have also investigated the performance of the deformation method for Example 2 with $\Delta t = 0.01$. The results are presented in Table 5. The equidistribution error is slightly worse than that of the Monge–Kantorovich approach and has not yet reached the quadratic scaling with grid refinement for large grids: From the 256×256 grid to the 512×512 grid, the error decreases by a factor of 3.2. The computational time (ignoring the cost of inversion) to reach the adapted grid scales optimally with the number of grid points and is in fact close to that of Example 1. Table 5 also shows some grid quality measures expressed in terms of variation with respect to the same quantities calculated with the Monge–Kantorovich approach. In terms of these measures, the grid obtained with the deformation method is worse: For example, for the 256×256 grid, the grid-point displacement and the grid-cell distortion are respectively 5% and 15% higher than those of the Monge–Kantorovich approach. This fact can be clearly seen in Fig. 8, which shows the new grid configuration obtained with the deformation method (left) and the related grid $\mathbf{x}'(\mathbf{x})$ obtained by inversion (right). By comparing Figs. 7 and 8, it is clear than the

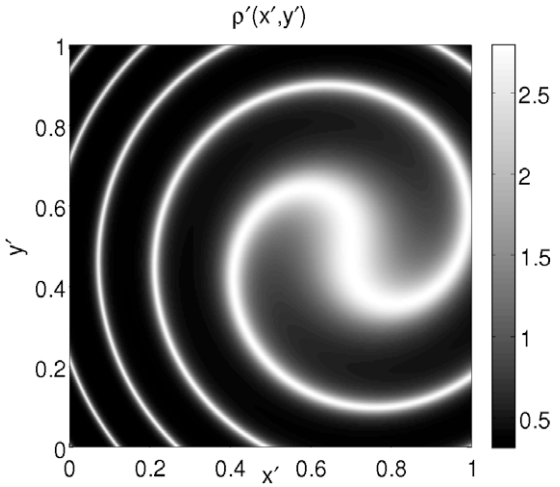


Fig. 6. $\rho'(x', y')$ for Example 2, which has $\rho'_{\max}/\rho'_{\min} \approx 9$.

Table 4
Example 2, inverse formulation

Number of cells	Error	CPU time (s)	$\|\mathbf{p}\|_2^{MK}$	$\|g_{11} + g_{22}\|_1^{MK}$	Newton iterations	GMRES iterations
16×16	4.22×10^{-1}	0.2	0.0766	1.528	5	6
32×32	2.17×10^{-1}	0.5	0.0801	1.578	6	7
64×64	9.45×10^{-2}	1.8	0.0810	1.612	6	7
128×128	2.88×10^{-2}	6.9	0.0815	1.622	6	7
256×256	7.16×10^{-3}	27	0.0817	1.624	6	7
512×512	1.76×10^{-3}	109	0.0818	1.625	6	7

Grid convergence study for the Monge–Kantorovich approach, reporting the equidistribution error, the CPU time (ignoring the cost of inversion), the grid quality measures $\|\mathbf{p}\|_2^{MK}$ and $\|g_{11} + g_{22}\|_1^{MK}$, and the number of linear and nonlinear iterations.

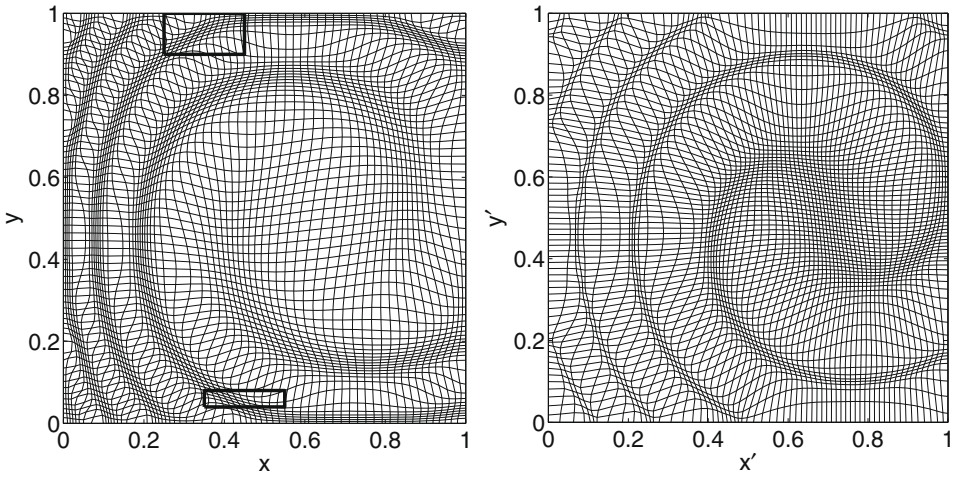
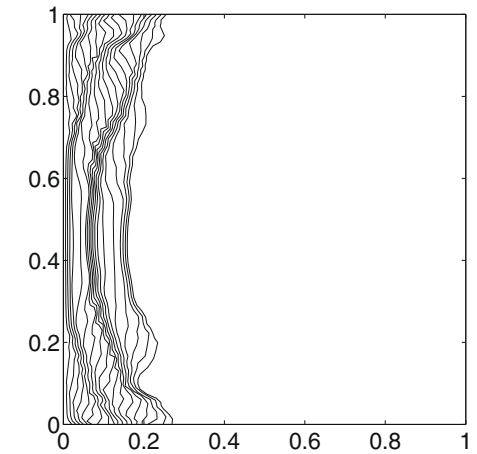


Fig. 7. Example 2, inverse formulation. Monge–Kantorovich approach: New grid configuration $\mathbf{x}(\mathbf{x}')$ with 64×64 cells (left) and related grid $\mathbf{x}'(\mathbf{x})$ obtained by inversion (right). The boxes indicated in bold are expanded in scale in Fig. 9 (left).

Table 5
Example 2, inverse formulation

Number of cells	Error	CPU time (s)	$\frac{\|\mathbf{p}\|_2}{\|\mathbf{p}\|_2^{MK}} - 1$ (%)	$\frac{\|g_{11}+g_{22}\|_1}{\|g_{11}+g_{22}\|_1^{MK}} - 1$ (%)
16×16	5.28×10^{-1}	0.2	+22	+9
32×32	2.83×10^{-1}	0.9	+6	+11
64×64	1.40×10^{-1}	3.3	+5	+13
128×128	5.41×10^{-2}	13.6	+5	+14
256×256	1.98×10^{-2}	55	+5	+15
512×512	6.24×10^{-3}	231	+5	+15

Grid convergence study for the deformation method ($\Delta t = 0.01$). The grid quality measures $\|\mathbf{p}\|_2$ and $\|g_{11} + g_{22}\|_1$ are expressed in terms of variation with respect to the corresponding quantities obtained with the Monge–Kantorovich approach, Table 4.



two methods converge to different grids and the grid cells obtained with the deformation method are much more distorted (relative to square cells) than those of the Monge–Kantorovich approach. Again, this is not surprising since the deformation method [with the velocity field given by Eqs. (A.6) and (A.7)] does not minimize grid point displacement or grid cell distortion. Fig. 9 compares the grid obtained by the Monge–Kantorovich approach (left) and by the deformation method (right) in the regions $x, y \in [0.25, 0.45] \times [0.9, 1]$ and $x, y \in [0.35, 0.55] \times [0.04, 0.08]$. One can see how, for this example, the deforma-

tion method produces some very highly elongated cells while the Monge–Kantorovich approach tends to minimize cell distortion. We also notice that, for this example, the grid obtained with the deformation method is folded in $x, y \in [0.35 - 0.55] \times [0.04, 0.08]$.

A comment on grid folding (tangling) is in order. While prescribing a positive Jacobian ensures an untangled grid in the limit to the continuum, finite resolution may in fact allow grid tangling. Fundamentally, the reason is that one can define a tangled grid cell with positive average volume (as measured according to Section 4.4). The issue then becomes a matter of robustness of a given grid equidistribution procedure against grid tangling when finite resolution is employed. In this light, and as demonstrated by the previous example, the Monge–Kantorovich approach proposed here has a definite advantage, since it is guaranteed to minimize grid displacement (and grid distortion approximately) for a given number of degrees of freedom. Clearly, a tangled grid cell – even with positive average volume – counts against both principles, since it necessarily requires larger displacements and more distortion than an untangled one. Thus, while we cannot offer a proof at this time that the method will always result in an untangled grid, numerical experiments demonstrate that it performs robustly, much more so than the deformation method.

5.3. Additional challenging tests

The results presented so far have highlighted the effectiveness of the Monge–Kantorovich approach in achieving a grid with prescribed volumes: The constraint Eq. (6) is satisfied up to discretization truncation errors, the computational time needed to achieve the solution scales linearly with the number of grid points, and the adapted grid is optimal with respect

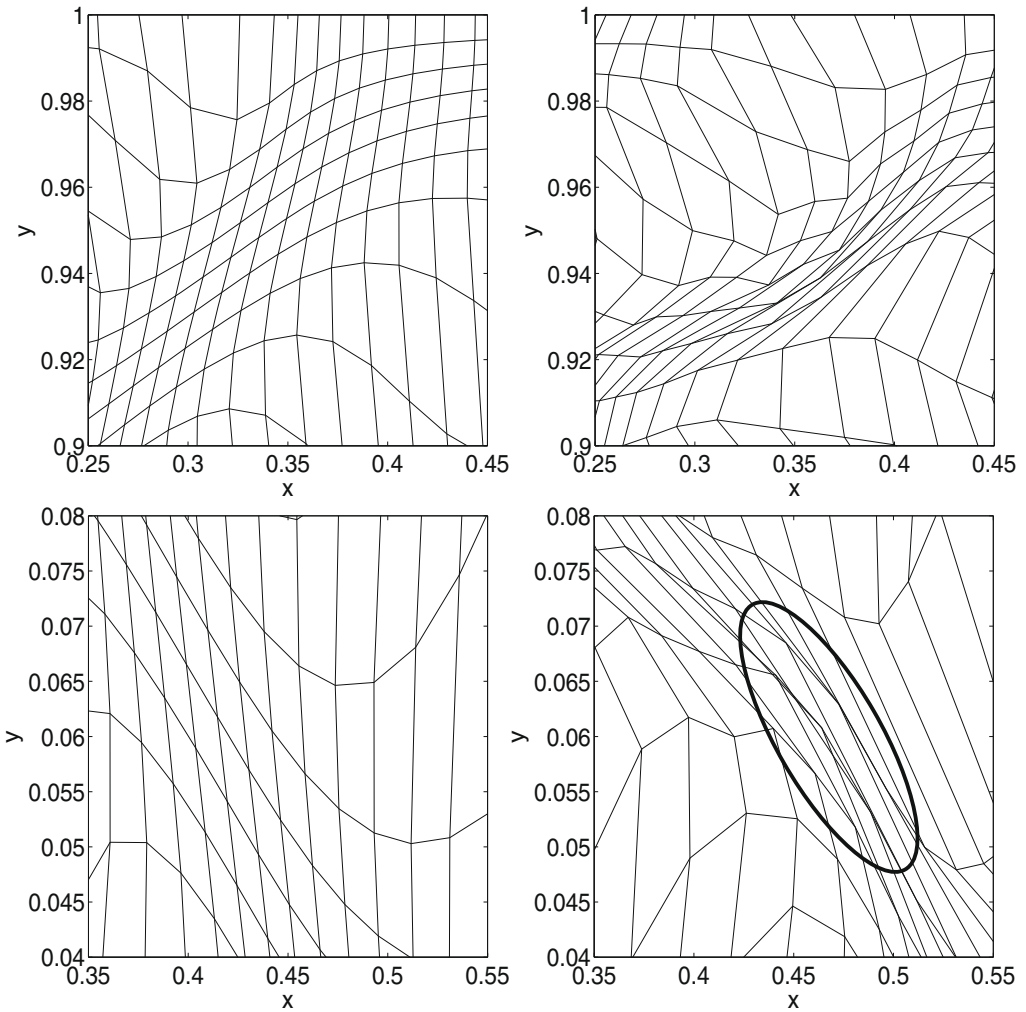


Fig. 9. Example 2, inverse formulation. Comparison of the new grid configuration $\mathbf{x}(\mathbf{x}')$ obtained with the Monge–Kantorovich approach (left) and the deformation method (right) for $x, y \in [0.25 - 0.45; 0.9 - 1]$ and $x, y \in [0.35 - 0.55; 0.04 - 0.08]$. In the area indicated with the bold curve, the grid obtained with the deformation method is folded.

to the L_2 norm of the displacement of the grid points and close to optimal with respect to grid distortion, measured as the trace of the metric tensor (Section 3.1). In this subsection, we further test the robustness of the Monge–Kantorovich approach by solving the *direct problem* for three challenging cases. Clearly, the solution of the direct problem is more challenging since $\rho'(\mathbf{x}') = \rho'(\mathbf{x} + \nabla\Phi)$ is a functional of the unknown Φ and this is an additional source of nonlinearity of Eq. (21). For these challenging tests, with $|\rho' - 1|$ of order unity and with a wide range of scales, this source of nonlinearity presents more difficulties that the Hessian in Eq. (21).

We start by solving Example 2 with the direct method and refer to this case as Example 3. The results are reported in Table 6. It is evident that, for fine grids, the equidistribution error scales as n_x^{-2} for $n_x = n_y$ (similarly to Example 2). For instance, from the 256×256 grid to the 512×512 grid the error decreases by a factor of 3.9. The error is about half the error in Table 4, which uses the inverse formulation. This is reasonable, because the inverse formulation uses a non-optimal uniform grid in $\mathbf{x}'$. The direct method uses a uniform grid in $\mathbf{x}$, which is optimal because $\rho = 1$.

The computational time for convergence scales linearly with the number of grid cells N and is roughly 3 times higher than the time required for the solution of the inverse problem in Example 2 (although the computational time reported in Table 4 does not include the relatively small time needed for the inversion of the mapping). The number of linear and nonlinear iterations remains roughly constant for all the grids considered, and are about double the number required in Table 4. Fig. 10 shows the new grid obtained with 64×64 cells. When comparing Fig. 10 with Fig. 7 (right), small differences can be noticed. These are due to the fact that, for the 64×64 grid, the equidistribution error is a few percent. These differences vanish with grid refinement.

For our next example, we choose the following expression for the density:

Example 4

$$\rho'(x', y') = C \frac{\exp\left(-2\sqrt{(x' - 0.5)^2 + (y' - 0.5)^2}\right)}{\sqrt{(x' - 0.7)^2 + (y' - 0.7)^2}}. \tag{58}$$

Notice that this choice of the ρ' is singular at $x' = y' = 0.7$, and $J'/J = 1/\rho'$ (recall $\rho = 1$) goes to zero there. In this case, the deformation method cannot be used because its formulation requires a strictly positive Jacobian [29]. We have found numerically that, when this condition is violated, the deformation method yields folded grids around the singular point. On the other hand, the Monge–Kantorovich approach performs robustly. (In this case, we set $\omega = 0.5$ as the Jacobi MG smoother

Table 6
Example 3, direct formulation

Number of cells	Error	CPU time (s)	Newton iterations	GMRES iterations
16×16	1.80×10^{-1}	0.5	13	15
32×32	1.07×10^{-1}	1.6	11	16
64×64	3.92×10^{-2}	4.9	9	14
128×128	1.23×10^{-2}	19	8	15
256×256	3.44×10^{-3}	79	9	15
512×512	8.93×10^{-4}	317	9	15

Grid convergence study for the Monge–Kantorovich approach, using the prescribed density ρ' of Example 2.

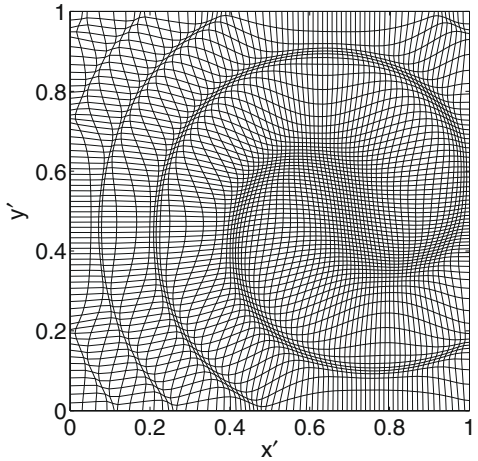


Fig. 10. Example 3, direct formulation. Monge–Kantorovich approach: New grid configuration using 64×64 cells.

damping parameter.) Results from a grid-convergence study are summarized in Table 7. First, one notices from the second column of Table 7 that J'_{max}/J'_{min}, the ratio of the maximum to the minimum value of the Jacobian $J' = 1/\rho'$, increases sharply with the total number of cells, being almost two orders of magnitude higher for the case of 256×256 cells. [From Eq. (50), J'_{min} and J'_{max} are proportional to the maximum and minimum volumes of cells formed by joining vertices with straight lines and, as $n_x = n_y$ increases, the vertices nearest to the singularity get even closer to it.] Second, the equidistribution error scales as n_x^{-2} with grid refinement for coarse grids, but the scaling is lost as the grid is refined. The reason is that, while finer grids attempt to resolve the singularity by placing more points around it, the resulting error grows faster than the available resolution (since the singularity is non-differentiable), and the n_x^{-2} scaling of the equidistribution error is lost. However, the computational time required for the convergence still scales optimally with the number of grid points (and is very similar to that of Example 3). The number of Newton iterations required for convergence is always about 7, while the number of GMRES iterations grows slightly from 10 to saturate at 16. This is an indication that this singular example is more challenging for the solver than previous examples, but still manageable. Fig. 11 shows the adapted grid for the case of 64×64 cells (left) and a detail near the singular point with $x, y \in [0.6, 0.8]$ (right).

As our last challenging example, we consider a density which corresponds to the luminosity of an image. We refer to this case as Example 5. We choose the well-known image of Lena. See Fig. 12. This image has become very popular in the image processing community because it contains a mixture of fine details and high and low contrast regions, which makes it suitable for testing imaging algorithms. We use it here with the purpose of testing our grid generation algorithm against a prescribed density function with significant amount of structure. Areas of higher luminosity of the image correspond to larger values of the density function and therefore to areas where the grid will be more refined. Areas of lower luminosity correspond to areas where the grid will be less refined. Thus, when the grid is plotted with white lines against a black background, a positive image is reproduced by the grid. Fig. 13 shows the new grid obtained with 64×64 cells (white lines against a black background). With this level of resolution, one can barely recognize the presence of Lena in the picture. Fig. 14 shows the new grid obtained with 200×200 cells. With this higher resolution, the picture is reproduced very well and many details such as the plume can be recognized. The performance of the method is presented in Tables 8 and 9. Table 8 is obtained by using a low resolution (200×200 pixel) image of Lena, while Table 9 is obtained by the higher resolution image

Table 7
Example 4, direct formulation

Number of cells	J'_{max}/J'_{min}	Error	CPU time (s)	Newton iterations	GMRES iterations
16×16	137	1.07×10^{-2}	0.3	7	10
32×32	356	3.04×10^{-3}	1	6	12
64×64	1127	8.21×10^{-4}	4	7	13
128×128	2829	2.10×10^{-4}	17.4	7	16
256×256	8886	7.78×10^{-5}	70	7	16

Grid convergence study for the Monge–Kantorovich approach.

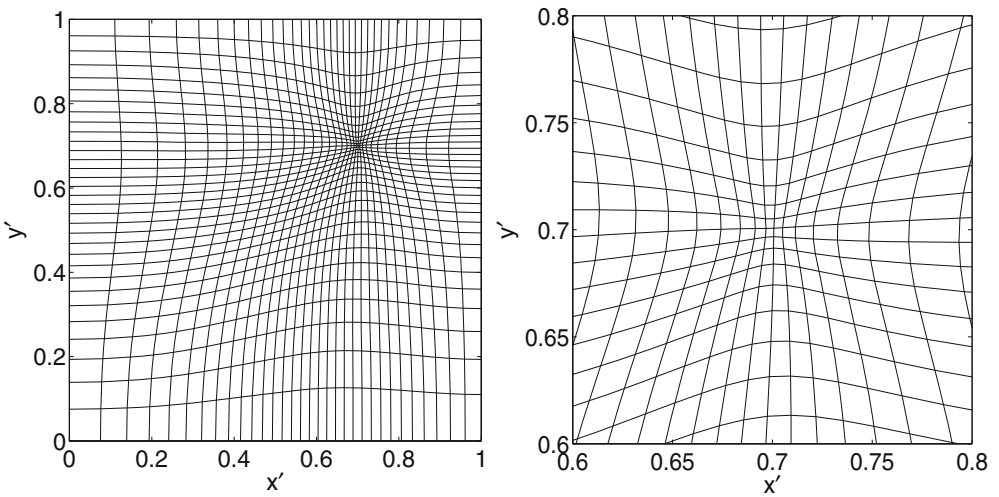


Fig. 11. Example 4, direct formulation. Monge–Kantorovich approach: new grid configuration using 64×64 cells (left); zoom in $x, y \in [0.6, 0.8]$ (right).

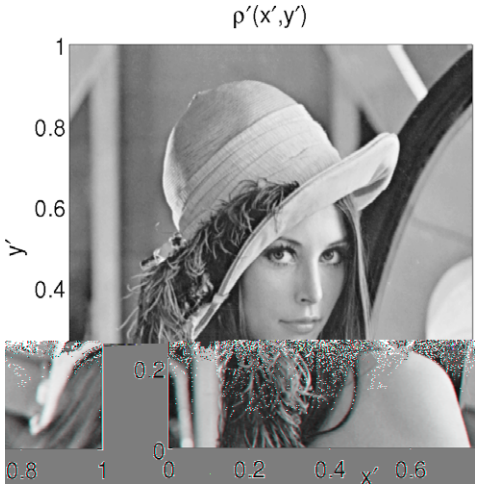


Fig. 12. Example 5, direct formulation. $\rho'(x',y')$ prescribed according to the image of Lena.

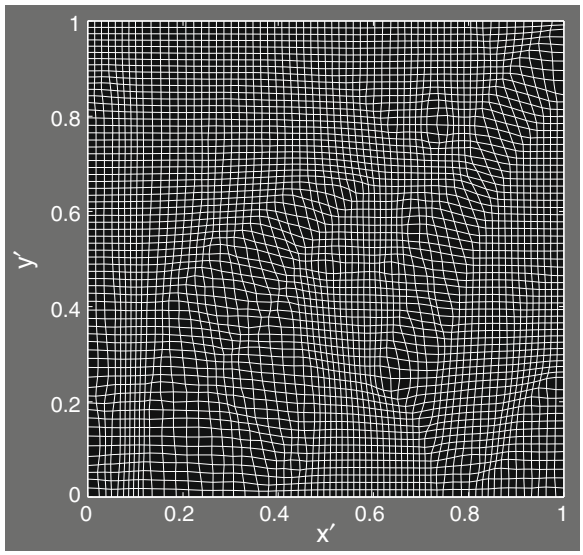


Fig. 13. Example 5, direct formulation. Monge–Kantorovich approach: New grid configuration using 64×64 cells for Lena.

(512×512 pixel) shown in Fig. 12. By comparing Tables 8 and 9, one can see that, for the high resolution image, more nonlinear iterations are required when the grid is refined. On the other hand, in terms of the number of linear iterations, the method performs well in both cases and requires about one GMRES iteration for each Newton iteration. This is an indication that the multigrid preconditioning is working properly. The fact that the high resolution image requires more Newton iterations under grid refinement is not surprising considering that there are sharp gradients in the image which are now resolved better by the prescribed monitor function. Finally, we wish to emphasize that the initial grid is the identity $\mathbf{x}'(\mathbf{x}) = \mathbf{x}$, or $\Phi = 0$. In a time stepping context, the grid (i.e. Φ) from the previous time step would provide a good first guess and require fewer Newton iterations.

6. Conclusions

We have presented an effective and robust approach for cell-area equidistribution in 2D. The new method, which is based on Monge–Kantorovich optimization, is obtained by a minimization of the L_2 norm of the grid displacement, constrained to satisfy a given distribution of cell volumes.

Table 8
Example 5, direct formulation

Number of cells	IRES iterations
32×32	
64×64	
128×128	
256×256	

Grid convergence study for the M

This method is based o has no free
parameters, and solutions with a Jaco-
bian-Free Newton–Krylov itioning can
be effectively used in order th the num-
ber of grid points).

We have compared the l onge–Kant-
orovich approach produces e L_2 norm of
the displacement of the gr distribution
error comparable to that of by success-
fully applying it to very cha ases. Specif-
ically, we showed that fo he Monge–
Kantorovich approach for t

We have also shown th tent with a
grid-distortion-minimizatio the metric
tensor.

Finally, we point out tha 2D and to a
cube in 3D [37]. In light o eters in the

Monge–Kantorovich variational formulation. In earlier variational grid generation approaches, suitable user-specified weights were needed to ensure grid quality. While, in principle, grid quality should be considered in connection to a given physical problem of interest [55], there are case studies in the literature which are so ill-posed that they produce poor-quality grids for most applications and mesh generators of interest. (See e.g. the Rogue's Gallery in Ref. [56].) For such cases, it is very convenient that the grid generation algorithm has some grid-quality measures "built-in". (General application-independent grid-quality measures are available in terms of local geometric properties of the grid mapping; see e.g. Ref. [8].) The Monge–Kantorovich approach presented here features such "built-in" measures, as it minimizes grid-point displacement (constrained to the equidistribution principle), and it features a connection with the minimization of grid distortion. It is the lack of adjustable parameters (other than the monitor function to be equidistributed) and these "built-in" measures of grid quality that make the Monge–Kantorovich approach so promising for robust grid adaptation.

Acknowledgment

The authors are very grateful to Jerry Brackbill, Darryl Holm, Alex Dragt, Brent Wohlberg, and Richard Chartrand for stimulating discussions. GLD wishes to thank Caterina Baravalle for constant support. This work was funded by the Laboratory Directed Research and Development program (LDRD), US Department of Energy Office of Science, Office of Fusion Energy Sciences, under the auspices of the National Nuclear Security Administration of the US Department of Energy by Los Alamos National Laboratory, operated by Los Alamos National Security LLC under Contract DE-AC52-06NA25396.

Appendix

In this Appendix, we outline a method of finding a map with a given Jacobian, and an adaptive grid generation method, the *deformation method*, based on it. The main idea of this method was first introduced by Moser [30] for the volume elements on a Riemannian manifold without a boundary. Later, it was extended by Banayaga [57] and by Dacoragna and Moser [31] to manifolds with a boundary. Liao and Anderson [29] suggested using this method to generate a grid with given volume elements, and named the grid generation strategy *the deformation method*. More recently, Cho and Jun [58] have implemented this method for shell finite element problems. More applications of the deformation method can be found in Refs. [59–61].

The problem addressed in Ref. [29] requires the right hand side of Eq. (6) to be dependent only on the coordinates of the initial grid. With the choice $\rho = 1$ used in this paper, this is equivalent to solving the inverse problem, as discussed in Section 2.3. Therefore, we describe below the specific algorithm suggested by Liao and Anderson [29] in terms of the inverse formulation and using the notation adopted in this paper. The aim is to construct $\mathbf{x}(x', y')$ that satisfies

$$\det[\nabla \mathbf{x}]_{x'} = [x, y]_{x'} = \rho'(x', y') \quad \text{for all } (x, y) \in X, \tag{A.1}$$

where the notation x' indicates that the operators are acting on the $\mathbf{x}'$ coordinates. The method of Ref. [29] unnecessarily constrains the points of the computational boundary to be fixed under the mapping transformation. This is rather restrictive: We require only that each of the four sides of the boundary map to its corresponding side, but otherwise allow the boundary points to move tangential to the boundary. The algorithm outlined in Ref. [29] is as follows:

Step 1: Find a vector field $\mathbf{v} : X \to \mathbb{R}^2$ such that

$$\nabla_{x'} \cdot \mathbf{v}(x', y') = \rho'(x', y') - 1 \equiv g(x', y'), \quad (x', y') \in X, \tag{A.2}$$

$$\mathbf{v} \cdot \mathbf{n} = 0, \quad (x', y') \in \partial X, \tag{A.3}$$

where $\mathbf{n}$ is the outward normal vector to the boundary.

Step 2: Solve the following equation for $\vec{\phi}(t, \mathbf{x}') : [0, 1] \times X \to X \subset \mathbb{R}^2$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{\phi}(t, \mathbf{x}') = \frac{\mathbf{v}(\vec{\phi}(t, \mathbf{x}'))}{t + (1 - t)\rho'(\vec{\phi}(t, \mathbf{x}'))} = \mathbf{v}_t, \tag{A.4}$$

$$\vec{\phi}(t = 0, \mathbf{x}') = \mathbf{x}'. \tag{A.5}$$

where t is an artificial time parameter. Then, $\vec{\phi}(t = 1, \mathbf{x}')$ yields the final position obtained by the deformation of the initial grid position (x', y'), namely $\mathbf{x}(\mathbf{x}') = \vec{\phi}(t = 1, \mathbf{x}')$. The fact that the required Jacobian is indeed obtained was originally proved by Moser [30,31]. Liao and Anderson [29] construct a solution of Eq. (A.2) which has symmetry between x' and y' by using $\mathbf{v} = (v_1, v_2)$

$$v_1(x', y') = \frac{1}{2}\left[\int_0^{x'} g(t, y')\mathrm{d}t - h(x')\int_0^1 g(t, y')\mathrm{d}t + h'(y')\int_0^{x'}\int_0^1 g(s, t)\mathrm{d}t\,\mathrm{d}s\right], \tag{A.6}$$

$$v_2(x', y') = \frac{1}{2}\left[\int_0^{y'} g(x', t)\mathrm{d}t - h(y')\int_0^1 g(x', t)\mathrm{d}t + h'(x')\int_0^{y'}\int_0^1 g(t, s)\mathrm{d}t\,\mathrm{d}s\right], \tag{A.7}$$

where $h \in C^1$ is any function on $[0, 1]$ such that

$$h(0) = h'(0) = h'(1) = 0, \quad h(1) = 1. \tag{A.8}$$

Notice that this velocity field is not irrotational.

The existence and uniqueness of the solution of the initial value problem in Eq. (A.4) has been proved [30,31]. Therefore, the existence of a solution of the problem posed in (A.1) is clear. However, this solution is clearly not unique. In particular, it depends on the function h of Eqs. (A.6)–(A.8).

We have found, by numerical simulations implementing different choices of the function h, that solutions corresponding to different functions h do change the solution, but do not appreciably affect the accuracy of the Jacobian found, as long as the boundary conditions (A.8) are satisfied.

With regard to the numerical implementation of the deformation method, we notice that the right hand side of Eq. (A.4) involves the velocity field $\mathbf{v}(\vec{\phi}(t,\mathbf{x}'))$ which depends on the solution $\vec{\phi}(t,\mathbf{x}')$. Thus, as $\vec{\phi}(t,\mathbf{x}')$ evolves with time, the new velocity field $\mathbf{v}(\vec{\phi}(t,\mathbf{x}'))$ must be updated in accordance with Eqs. (A.6) and (A.7). That is, the integrations in Eqs. (A.6) and (A.7) have upper limits depending on $\vec{\phi}(t,\mathbf{x}')$. A direct implementation of this method, which re-evaluates the velocity integral as needed, has been found impractical even for moderate grids. Instead, a better strategy is to evaluate the velocity integral once, at the beginning of the calculation, on a reference grid in the domain $[0,1]\times[0,1]$, and then interpolate (using cubic splines) $\mathbf{v}(\vec{\phi}(t,\mathbf{x}'))$ as necessary. The resulting algorithm has been found to be practical and very efficient.

References

[1] J.F. Thompson, Survey of dynamically-adaptive grids in the numerical solution of partial differential equations, Applied Numerical Mathematics 1 (1) (1985) 3–27.
[2] P.R. Eiseman, Adaptive grid generation, Computer Methods in Applied Mechanics and Engineering 64 (1–3) (1987) 321–376.
[3] D.A. Anderson, Equidistribution schemes, Poisson generators, and adaptive grids, Applied Mathematics and Computation 24 (3) (1987) 211–227.
[4] W. Huang, Y. Ren, R.D. Russell, Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, SIAM J. Numer. Anal. 31 (3) (1994) 709–730.
[5] M.J. Baines, Least squares and approximate equidistribution in multidimensions, Numerical Methods for Partial Differential Equations 15 (5) (1999) 605–615.
[6] G. Lapenta, Variational grid adaptation based on the minimization of local truncation error: Time independent problems, Journal of Computational Physics 193 (2004) 159.
[7] J.F. Thompson, Z.A. Warsi, C.W. Mastin, Numerical Grid Generation: Foundations and Applications, North-Holland, New York, 1985.
[8] V.D. Liseikin, Grid Generation Methods, Springer, Berlin, New York, 1999.
[9] A. Winslow, Numerical solution of the quasi-linear poisson equation in a nonuniform triangle mesh, Journal of Computational Physics 1 (1967) 149.
[10] A. Winslow, Adaptive Mesh Zoning by the Equipotential Method, UCID-19062, Lawrence Livermore National Laboratory, 1981.
[11] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in 2 dimensions, Journal of Computational Physics 46 (3) (1982) 342–368.
[12] A.S. Dvinsky, Adaptive grid generation from Harmonic maps, Journal of Computational Physics 95 (1991) 450–476.
[13] J.U. Brackbill, An adaptive grid with directional control, Journal of Computational Physics 108 (1993) 38–50.
[14] W. Cao, W. Huang, R.D. Russell, An r-adaptive finite element method based upon moving mesh pdes, Journal of Computational Physics 149 (1999) 221–244.
[15] W. Cao, W. Huang, R.D. Russell, A study of monitor functions for two-dimensional adaptive grid generation, SIAM J. Sci. Comput. 20 (6) (1999) 1978–1994.
[16] R. Hamilton, Harmonic maps of manifolds with boundary, Lecture Notes in Mathematics, vol. 471, Springer-Verlag, New York, 1975.
[17] R. Schoen, S.T. Yau, On univalent harmonic maps between surfaces, Inventiones Mathematicae 44 (1978) 265–278.
[18] W. Huang, Variational mesh adaptation: isotropy and equidistribution, Journal of Computational Physics 174 (2001) 903–924.
[19] R.I. Kreis, H.A. Hassan, F.C. Thames, Application of a variational method for generating adaptive grids, in: AIAA 23rd Aerospace Sciences Meeting, Reno, NV, 1985, pp. 85–0487.
[20] L. Chacón, G. Lapenta, A fully implicit, nonlinear adaptive grid strategy, Journal of Computational Physics 212 (2) (2006) 703.
[21] J.F. Thompson, F.C. Thames, C.W. Mastin, Automatic numerical generation of body fitted curvilinear coordinate system for field containing any number of arbitrary two dimensional bodies (appl. to airfoils), Journal of Computational Physics 15 (3) (1974) 299–319.
[22] P.M. Knupp, Jacobian-weighted elliptic grid generation, SIAM Journal on Scientific Computing 17 (6) (1996) 1475–1490.
[23] L. Kania, Elliptic adaptive grid generation and area equidistribution, International Journal for Numerical Methods in Fluids 30 (5) (1999) 481–491.
[24] W. Huang, D.M. Sloan, A simple adaptive grid method in two dimensions, SIAM Journal on Scientific Computing 15 (4) (1994) 776–797.
[25] J. Castillo, E.M. Pedersen, Solution adaptive direct variational grids for fluid flow calculations, Journal of Computational and Applied Mathematics 67 (2) (1996) 343–370.
[26] T.F. Chen, H.D. Yang, Numerical construction of optimal adaptive grids in two spatial dimensions, Computers and Mathematics with Applications 39 (12) (2000) 101–120.
[27] J.-C. Yang, B.K. Soni, Structured adaptive grid generation, Applied Mathematics and Computation 65 (1–3) (1994) 265–278.
[28] D.A. Anderson, Adaptive grid scheme controlling cell area/volume, in: AIAA 25rd Aerospace Sciences Meeting, Reno, NV, 1987, pp. 87–0202.
[29] G. Liao, D. Anderson, A new approach to grid generation, Applicable Analysis 44 (1992) 285–297.
[30] J. Moser, On volume elements on a manifold, Transactions of the American Mathematical Society 120 (2) (1965) 286–294.
[31] B. Dacorogna, J. Moser, On a partial-differential equation involving the Jacobian determinant, Annales de l'Institut Henri Poincaré. Analyse non Linéaire 7 (1) (1990) 1–26.
[32] G. Monge, Mémoire sur la théorie des déblais at des remblais, Histoire de l'Académie Royale des Sciences de Paris (1781) 666–704.
[33] L.V. Kantorovich, On the translocation of masses, C.R. (Doklady) Acad. Sci. URSS (N.S.) 37 (1942) 199–201.
[34] L.C. Evans, Partial Differential Equations, American Mathematical Society, 1999.
[35] L. Caffarelli, L. Nirenberg, J. Spruck, The Dirichlet problem for nonlinear 2nd-order elliptic equations I. Monge–Ampère equation, Communications on Pure and Applied Mathematics 37 (3) (1984) 369–402.
[36] C.J. Budd, J.F. Williams, Parabolic Monge–Ampère methods for blow-up problems in several spatial dimensions, Journal of Physics A 39 (2006) 5425.
[37] J.M. Finn, G.L. Delzanno, L. Chacón, Grid generation and adaptation by Monge–Kantorovich optimization in two and three dimensions, in: 17th International Meshing Roundtable, Pittsburgh, PA, USA (accepted for publication).
[38] S. Li, L. Petzold, Moving mesh methods with upwinding schemes for time dependent PDEs, Journal of Computational Physics 131 (1997) 368–377.
[39] Y. Brenier, Polar factorization and monotone rearrangement of vector-valued functions, Communications on Pure and Applied Mathematics 44 (1991) 375.
[40] L.C. Evans, Partial differential equations and Monge–Kantorovich mass transfer, in: S.T. Yau (Ed.), Current Developments in Mathematics, 1997.
[41] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.

[42] R. Dembo, S. Eisenstat, R. Steihaug, Inexact Newton methods, Journal of Numerical Analysis 19 (1982) 400.
[43] L. Chacón, D.A. Knoll, A 2D high-β Hall MHD implicit nonlinear solver, Journal of Computational Physics 188 (2) (2003) 573–592.
[44] D.A. Knoll, W.J. Rider, Multigrid preconditioned Newton–Krylov method, SIAM Journal on Scientific Computing 21 (2) (1999) 691–710.
[45] W.J. Rider, D.A. Knoll, G.L. Olson, A multigrid Newton–Krylov method for multimaterial equilibrium radiation diffusion, Journal of Computational Physics 152 (1) (1999) 164–191.
[46] D.A. Knoll, V.A. Mousseau, On Newton–Krylov multigrid methods for the incompressible Navier–Stokes equations, Journal of Computational Physics 163 (1) (2000) 262–267.
[47] M. Pernice, M.D. Tocci, A multigrid-preconditioned Newton–Krylov method for the incompressible Navier–Stokes equations, SIAM Journal on Scientific Computing 23 (2) (2001) 398–418.
[48] L. Chacón, D.C. Barnes, D.A. Knoll, G.H. Miley, An implicit energy-conservative 2d Fokker–Planck algorithm. II. Jacobian-free Newton–Krylov solver, Journal of Computational Physics 157 (2) (2000) 654–682.
[49] L. Chacón, D.A. Knoll, J.M. Finn, An implicit, nonlinear reduced resistive MHD solver, Journal of Computational Physics 178 (1) (2002) 15–36.
[50] W.L. Briggs, A Multigrid Tutorial, SIAM, Philadelphia, 1987.
[51] D.A. Knoll, G. Lapenta, J.U. Brackbill, A multilevel iterative field solver for implicit, kinetic, plasma simulation, Journal of Computational Physics 149 (2) (1999) 377–388.
[52] J.U. Brackbill, H.M. Ruppel, FLIP: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, Journal of Computational Physics 65 (2) (1986) 314–343.
[53] J.U. Brackbill, FLIP MHD: a particle-in-cell method for magnetohydrodynamics, Journal of Computational Physics 96 (1) (1991) 163–192.
[54] D. Sulsky, J.U. Brackbill, A numerical method for suspension flow, Journal of Computational Physics 96 (2) (1991) 339–368.
[55] G.A. Hansen, R.W. Douglass, A. Zardecki, Mesh Enhancement: Selected Elliptic Methods, Foundations, and Applications, Imperial College Press, 2005.
[56] P.M. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC-Press, 1993.
[57] A. Banyaga, Formes volume sur es varietes a bord, Enseignement Mathematics 20 (1974) 127–131.
[58] M. Cho, S. Jun, r-Adaptive mesh generation for shell finite element analysis, Journal of Computational Physics 199 (1) (2004) 291–316.
[59] F. Liu, S.H. Ji, G.J. Liao, Adaptive grid method and its application to steady Euler flow calculations, SIAM Journal on Scientific Computing 20 (3) (1998) 811–825.
[60] B. Semper, G. Liao, A moving grid finite-element method using grid deformation, Numerical Method for PDEs 11 (1995) 603–615.
[61] P. Bochev, G. Liao, G.D. Pena, Analysis and computation of adaptive moving grids by deformation, Numerical Method for PDEs 12 (1996) 489–506.