Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Applied Numerical Mathematics 47 (2003) 121–138

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

# Approaches for generating moving adaptive meshes: location versus velocity

Weiming Cao [a,1], Weizhang Huang [b,*,2], Robert D. Russell [c,3]

[a] *Department of Mathematics, University of Texas at San Antonio, San Antonio, TX 78249, USA*
[b] *Department of Mathematics, University of Kansas, Lawrence, KS 66045, USA*
[c] *Department of Mathematics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada*

**Abstract**

A variety of approaches for generating moving adaptive methods are summarized and compared. They basically fall into two groups: the velocity and the location based methods. The features, including the advantage and weakness, of each group are addressed. Brief numerical results are presented for several commonly used approaches to highlight their features and performance.


*Keywords:* Mesh adaptation; Mesh movement; Variational method; Lagrangian coordinate; Equidistribution

## 1. Introduction

Mesh adaptation plays an indispensable role in the efficient numerical simulation of many physical phenomenon. Among these we mention the occurrence of boundary and interior layers, blow-up, and moving interfaces. Over the past 20 years, with the development of various error estimate techniques, use of mesh adaptation has become standard practice in most numerical softwares.

The moving mesh method, or the $r$-refinement method, has been relatively less developed than other mesh adaptation techniques. For this method, the adaptive mesh is generally constructed as the image under an appropriate mapping of a fixed mesh over an auxiliary domain. By suitably defining the mapping, one may control various mesh properties, e.g., mesh density and alignment, that are desirable

* Corresponding author.
*E-mail addresses:* wcao@math.utsa.edu (W. Cao), huang@math.ku.edu (W. Huang), rdr@cs.sfu.ca (R.D. Russell).

[1] This author was supported in part by the Faculty Research Award (2001) of The University of Texas at San Antonio.
[2] This author was supported in part by the NSF (U.S.A) under Grant DMS-0074240.
[3] The author was supported in part by NSERC (Canada) through Grant OGP-0008781.

0168-9274/$30.00 
doi:10.1016/S0168-9274(03)00061-8

for the underlining applications. Although this method has been less popular than the local refinement (or $h$-refinement) method due to the difficulty in deriving suitable general governing equations for the adaptive mapping, it offers some distinct features useful in numerical computation. For example, it is in principle simple to implement, and much of the software based on fixed mesh methods can be easily extended to incorporate moving adaptive meshes. When there is no addition and deletion of grid points, difficulties from restarting the time integration procedure can be avoided. Moreover mesh movement fits naturally the salient feature of many time dependent problems, resulting in minimal numerical diffusion and dispersion.

During the past 20 years, and particularly the past 10 years, there have been a number of techniques proposed for generating moving adaptive meshes. Miller and Miller [27], Carlson and Miller [8,9], and Baines [1] developed moving finite element methods where nodal points are driven by the residual of the finite element approximation. Like Lynch [24], Johnson and Tezduyar [20] use an automatic mesh moving scheme in which the motion of the nodes is governed by the equations of linear elasticity, with boundary conditions for these equations imposed by the motion of the fluid boundaries and interfaces. Oden et al. [29] and Bank and Smith [2] propose updating the position of grid points by local averaging. Liao and coworkers [3,25,30] apply the concept of deformation map in differential geometry to the generation of moving adaptive meshes. Winslow [33], Thompson et al. [32], Brackbill and Saltzman [4], Dvinsky [12], Brackbill [5], Knupp [21] and Knupp and Robidoux [23] propose various elliptic equations or variational methods for defining the adaptive mapping. Recently, Huang [15] has proposed a new mapping equation for which the relation with error distribution is clearly formulated. Finally, the authors [7] have developed a type of moving mesh method based on the so-called geometric conservation law (GCL).

Most of these approaches can be classified into two groups. The first group is referred to as *the location based method* because it controls directly the location of mesh points, or in the continuous sense the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ from the auxiliary domain to the physical domain. A typical method in this group is a variational method which defines the mapping as the minimizer of a functional. The second group is referred to as *the velocity based method* since it targets directly the time derivative of the mapping $\boldsymbol{x}_t(\boldsymbol{\xi})$, or the mesh velocity. The second type of methods includes MFE, the deformation method, and the GCL method. For a velocity based method, the mesh equation is formulated for the mesh velocity, and the mesh point location is obtained by integrating the velocity field.

In this paper, we will summarize and compare the features of these two types of methods. We will focus on good representatives in each group, the GCL method and several typical variational methods. An early variational approach is the popular harmonic mapping method, and the GCL method reduces to the deformation method in a special case [7]. This paper is organized as follows. We first outline various velocity based methods in Section 2. In Section 3, we describe a number of variational formulations which have been considered in the past and the new variational formulation developed recently in [15]. Then we present several numerical examples in Section 4 to compare the adaptive meshes generated by the GCL method and the variational methods. Finally, conclusions are drawn in Section 5.

Throughout this paper, $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$ are used to denote the variables in the physical domain $\Omega$ and computational domain $\Omega_c$, respectively. $\boldsymbol{J} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$ is the Jacobian matrix for mapping $\boldsymbol{x}(\boldsymbol{\xi})$ from $\Omega_c$ to $\Omega$, and $J = \det(\boldsymbol{J})$. Also, $\nabla$ denotes the gradient operator with respect to the physical coordinate $\boldsymbol{x}$.

## 2. Velocity based approaches

### 2.1. The classical Lagrangian method

In fluid dynamics, Lagrangian coordinates form a moving coordinate system that is used to follow fluid particles. More precisely, if $\boldsymbol{u}(\boldsymbol{x},t)$ denotes the velocity of the fluid, $\boldsymbol{\xi}$ the reference coordinate of a fluid particle, and $\boldsymbol{x}(\boldsymbol{\xi},t)$ the position of the particle at time $t$, then the particle and therefore the Lagrangian coordinate lines evolve with

$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{u}. \tag{1}$$

A fortuitous property of Lagrangian coordinates is that convection terms are eliminated from the governing equation. For a viscous flow without inflow and outflow boundaries, the incompressibility of the fluid guarantees that the coordinate transformation between the Lagrangian and Eulerian coordinates is non-singular. However, it is well-known that the classical Lagrangian coordinates are rarely used directly with the widely used discretization methods such as finite differences and finite elements, because the mesh generated by the Lagrangian method is often very skew.

### 2.2. The moving finite element method

The moving finite element method (MFE) developed by Miller and Miller [27,28] also generates a moving mesh through a mesh velocity $\boldsymbol{x}_t$. Specifically, suppose a time dependent physical problem

$$\frac{\partial u}{\partial t} = \mathcal{L}u,$$

where $\mathcal{L}$ is a spatial differential operator, is given. The continuous version of the MFE can be viewed as determining a solution $u(\boldsymbol{x}(\boldsymbol{\xi},t),t)$ and $\boldsymbol{x}_t(\boldsymbol{\xi},t)$ by minimizing the $L^2$-norm of the residual involving $u$ and $\boldsymbol{x}_t$ over the entire space, viz.,

$$\min_{\boldsymbol{x}_t,\,\frac{Du}{Dt}} I_{\mathrm{mfe}}\left[\boldsymbol{x}_t, \frac{Du}{Dt}\right] \equiv \int_{\Omega} \left(\frac{Du}{Dt} - \nabla u \cdot \boldsymbol{x}_t - \mathcal{L}u\right)^2 W \,\mathrm{d}\boldsymbol{x},$$

where $\frac{D}{Dt}$ is the time derivative for $\boldsymbol{\xi}$ fixed and $W$ is a weight function. The classical version of MFE uses $W = 1$ [27,28] and the gradient weighted MFE (GWMFE) [8,9] uses $W = 1/(1+|\nabla u|^2)$. In the one dimensional case where $\mathcal{L}u = H(x,u,u_x)$ (e.g., $H = -uu_x$ for Burgers' equation), MFE produces a mesh speed $\boldsymbol{x}_t = -(\partial H)/(\partial u_x)$, which is identical to or at least an approximation to the mesh speed for the Lagrangian method [1].

A nice feature of the MFE is that the mesh attempts to follow a path corresponding to the smallest weighted $L^2$-norm of the residual of the discrete equations. In particular, if the physical PDE has a steady state solution, then the steady state mesh is a locally optimal one that produces the least error among all meshes with the same connectivity. The difficulty with MFE is that the mesh equations resulting from the minimization of functional $I_{\mathrm{mfe}}[\boldsymbol{x}_t, \frac{Du}{Dt}]$ can become degenerate, and its numerical computation requires careful regularization.

### 2.3. The GCL method and the deformation method

For any coordinate transformation $\boldsymbol{x}(\boldsymbol{\xi}, t)$ from $\Omega_c$ to $\Omega$, the mesh speed $\boldsymbol{x}_t$ and the time derivative of the Jacobian $J$ satisfy the identity

$$\nabla \cdot \boldsymbol{x}_t = \frac{1}{J}\frac{DJ}{Dt}. \tag{2}$$

This is the so-called geometric conservation law (GCL) [31].

We now consider the determination of the mapping $\boldsymbol{x}(\boldsymbol{\xi}, t)$ through its time derivative. Assume that we are given a monitor function $\rho(\boldsymbol{x}, t) > 0$ which reflects difficulties in the numerical approximation of the solution of the underlying problem. We want the cell size (area) in the adaptive mesh to be inversely proportional to $\rho(\boldsymbol{x}, t)$ at each time $t$. To this end, we require $\boldsymbol{x}(\boldsymbol{\xi}, t)$ to satisfy

$$\nabla \cdot \boldsymbol{x}_t = -\frac{1}{\rho}\frac{D\rho}{Dt}. \tag{3}$$

By comparing Eqs. (3) and (2), it follows readily that

$$\frac{D}{Dt}(\rho J) = 0, \tag{4}$$

which implies that

$$\rho J = \text{constant}, \tag{5}$$

over $\Omega$ for all the time if it is so initially. Note that (5) can be regarded as a multi-dimensional generalization of the well-known one dimensional equidistribution principle (defined in Section 3.1).

By the chain rule, (3) can be rewritten as

$$\nabla \cdot (\rho \boldsymbol{x}_t) + \frac{\partial \rho}{\partial t} = 0. \tag{6}$$

This equation is insufficient to determine the vector field $\boldsymbol{x}_t$. Motivation for finding supplementary equations is provided by the classic Helmholtz Decomposition Theorem for vectors: A continuous and differentiable vector field can be resolved into the orthogonal sum of the gradient of a scalar field and the curl of a vector field. Therefore, $\boldsymbol{x}_t$ can be determined by specifying both its divergence through (6) and its curl. In practice, it is often desirable on physical grounds to have the adaptive mesh follow the flow of some given vector field $\boldsymbol{u}(\boldsymbol{x}, t)$. Taking these considerations into account, we require $\boldsymbol{x}(\boldsymbol{\xi}, t)$ to satisfy

$$\nabla \times w(\boldsymbol{x}_t - \boldsymbol{u}) = 0, \tag{7}$$

where $w > 0$ is a weight function. This function, used only for defining (7), is different from those weight functions used for mesh adaptation. Different choices for $w$ lead to different curl conditions (7) for the vector field $\boldsymbol{x}_t$. In our computations we take $w = 1$ or $w = \rho$, but the possible utility of other choices cannot be ruled out.

Since the divergence and curl of a vector field are orthogonal in $L^2$, Eqs. (2) and (7) can be formulated as the minimization problem of the functional

$$I_{\text{gcl}}[\boldsymbol{x}_t] = \frac{1}{2}\int_{\Omega} \left| \nabla \cdot (\rho \boldsymbol{x}_t) + \frac{\partial \rho}{\partial t} \right|^2 + \left(\frac{\rho}{w}\right)^2 \left| \nabla \times w(\boldsymbol{x}_t - \boldsymbol{u}) \right|^2 \mathrm{d}\boldsymbol{x}, \tag{8}$$

where the weight $(\rho/w)^2$ is chosen to make the functional invariant under scaling of $\rho$ and $w$. Consider the boundary condition

$$\boldsymbol{x}_t \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega, \tag{9}$$

namely, a condition assuring no mesh points move in or out of the domain. Under this condition, it is not difficult to see that $\boldsymbol{x}_t$ is the minimizer of $I_{\text{gcl}}$ if and only if it is the solution of (6) and (7), cf. [7].

Eqs. (6), (7) and (9) form an elliptic system for the mesh velocity $\boldsymbol{x}_t$. Its solution exists and is smooth as long as the data and $\partial\Omega$ are smooth. Furthermore, by (5) the Jacobian of the mapping is determined by the monitor function $\rho(\boldsymbol{x}, t)$, and thus the mapping itself is non-singular, at least locally. However, like the Lagrange method or other moving mesh methods based on mesh velocity, this method can suffer from the tendency to produce increasingly skewed meshes.

*Deformation method.* The deformation map was introduced by Moser [10,26] in his study of volume elements of a compact Riemannian manifold to prove the existence of a $C^1$ diffeomorphism with a specified Jacobian. It has been adopted by Liao and coworkers [3,25,30] to define the deformation method for generating adaptive moving meshes. In our notation, this mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}, t)$ is determined from the system of equations [30]

$$\begin{aligned} \nabla \cdot (\rho \boldsymbol{x}_t) &= -\frac{\partial \rho}{\partial t} \quad \text{in } \Omega, \\ \nabla \times (\rho \boldsymbol{x}_t) &= 0 \qquad\quad \text{on } \partial\Omega. \end{aligned} \tag{10}$$

It is then easy to see that the deformation method is a special case of the GCL method with $\boldsymbol{u} = 0$ and $w = \rho$.

Note that both the deformation method and the GCL method control the Jacobian $J$ of the mapping in the same way, so the cell sizes produced by the two methods are the same. However, the extra freedom given by $\boldsymbol{u}$ and $w$ in the GCL method can be used to provide better control of the mesh behavior. For instance, in practical computation it is generally preferable to have an irrotational mesh velocity $\boldsymbol{x}_t$, which can result in less skewed grids. This can be achieved with the GCL method by choosing $w = 1$. But, on the other hand, an irrotational mesh velocity field is generally impossible with the deformation method, since

$$\nabla \times (\rho \boldsymbol{v}) = 0 \quad \text{implies} \quad \nabla \times \boldsymbol{v} = -\frac{1}{\rho} \nabla\rho \times \boldsymbol{v}, \tag{11}$$

and therefore $\nabla \times \boldsymbol{v}$ does not vanish.

Like the GCL method, the deformation method satisfies the relation (5) for any dimension. Thus, the Jacobian of the mapping stays positive, and the mapping itself is locally non-singular. This is a very advantageous feature of the deformation method (and the GCL method) since for most methods it is extremely difficult to prove that they produce non-singular coordinate transformations or meshes.

*Static version.* For many occasions, one needs to generate a mesh having a specified mesh topology and a prescribed mesh density distribution $m(\boldsymbol{x})$. An example is to produce an initial adaptive mesh which evenly distributes the interpolation error of the initial data. In this situation, we may define a time dependent monitor function by

$$\rho(\boldsymbol{x}, t) = (1 - t) + t\, m(\boldsymbol{x}), \quad 0 \leqslant t \leqslant 1,$$

and then use continuation, integrating the mesh equations from $t = 0$ to 1. A variant of this procedure can also be used to create the adaptive mesh at each time level.

## 3. Location based approaches

### 3.1. Mesh density and equidistribution in 1D

Consider the 1D (one-dimensional) case of an adaptive mapping $x(\xi)$ from a computational domain $\Omega_c$ to a physical domain $\Omega$. If the mesh on $\Omega_c$ is uniform then $\frac{\partial \xi}{\partial x}$ measures the density of the mesh on $\Omega$. Therefore, the mesh density is proportional to a prescribed function $m(x) > 0$ if

$$\frac{\partial \xi}{\partial x} = \text{const} \cdot m(x) \quad \text{on } \Omega. \tag{12}$$

This is the so-called equidistribution principle. It is equivalent to satisfying

$$\frac{\partial}{\partial x}\left( \left[m(x)\right]^{-1} \frac{\partial \xi}{\partial x} \right) = 0 \quad \text{on } \Omega,$$

which is exactly the Euler–Lagrange equation of the quadratic functional

$$I_{1\mathrm{D}}[\xi] = \int_{\Omega} \left[m(x)\right]^{-1} \left( \frac{\partial \xi}{\partial x} \right)^2 \mathrm{d}x. \tag{13}$$

### 3.2. Variational approaches in higher dimensions

*Knupp's methods.* A direct extension of the equidistribution principle (12) to higher dimensions would require

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}} = K(\boldsymbol{x}) \quad \text{on } \Omega, \tag{14}$$

where $K(\boldsymbol{x})$ is a prescribed $n \times n$ matrix describing various properties of the mapping. This is an over-determined system for the mapping $\boldsymbol{\xi}(\boldsymbol{x})$. Knupp [21,22] uses the least-squares principle to determine its inverse $\boldsymbol{x}(\boldsymbol{\xi})$ as the minimizer of the functional

$$I_{\mathrm{knp}}[\boldsymbol{\xi}] = \int_{\Omega} \left\| \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}} - K \right\|^2 \mathrm{d}\boldsymbol{x}, \tag{15}$$

where $\|\cdot\|$ is typically the Frobenius norm of matrix. A detailed discussion on how to choose the matrix $K$ is given in [22]; see also [23] for a broader discussion on algebraic properties of the Jacobian matrix. However, in addition to the Euler–Lagrange equation, $\boldsymbol{x}(\boldsymbol{\xi})$ must satisfy appropriate boundary conditions that specify the boundary correspondence between $\Omega$ and $\Omega_c$. It is far from clear how the boundary conditions affect the level of the minimization of $I_{\mathrm{knp}}$ and the overall behavior of the mapping.

*Winslow's variable diffusion functional.* Suppose that higher mesh concentrations are desired in regions with larger values of a given function $w(\boldsymbol{x}) > 0$. With the approach of Winslow [34], the mapping $\boldsymbol{\xi}(\boldsymbol{x})$ is defined as the minimizer of

$$I_{\text{win}}[\boldsymbol{\xi}] = \int_{\Omega} \frac{1}{w} \sum_i (\nabla \xi_i)^T (\nabla \xi_i)\, \mathrm{d}\boldsymbol{x}. \tag{16}$$

It can be viewed as a multi-dimensional extension of the functional (13) for one dimension.

*Thompson's method.* Following Winslow [33], Thompson et al. [32] also use a system of elliptic differential equations for generating body-fitted meshes. They propose to use the Poisson equations

$$\nabla^2 \xi_i = P_i(\boldsymbol{x}), \tag{17}$$

to control the mesh concentration and direction, where $P_i$, $1 \leqslant i \leqslant n$, are control functions. The system can be interpreted as the Euler–Lagrange equation of the quadratic functional

$$I_{\text{thm}}[\boldsymbol{\xi}] = \int_{\Omega} \sum_i \left( |\nabla \xi_i|^2 - P_i \xi_i \right) \mathrm{d}\boldsymbol{x}. \tag{18}$$

*The functional of Brackbill and Saltzman.* Brackbill and Saltzman [4] develop a popular variational method by combining mesh concentration, smoothness, and orthogonality. The functional associated with mesh concentration is

$$I_{\text{bs}}[\boldsymbol{\xi}] = \int_{\Omega} w(\boldsymbol{x}) J\, \mathrm{d}\boldsymbol{x}. \tag{19}$$

However, there are two major difficulties associated with the combined functional: the ellipticity of the Euler–Lagrange equation is not well characterized for all values of the weights that balance the functionals associated with concentration, smoothness, and orthogonality; and the various terms are not dimensionally homogeneous and must be re-scaled for each application; e.g., see [5] for more detailed discussion of the features of the functional.

*Harmonic mapping.* Dvinsky [12] uses a harmonic mapping for the purpose of adaptive mesh generation. Given an $n \times n$ symmetric positive definite matrix $G(\boldsymbol{x})$ with $g = \det(G)$, the mapping $\boldsymbol{\xi}(\boldsymbol{x})$ is defined as a harmonic mapping between $\Omega \subset \mathcal{R}^n$ equipped with metric $G$ and $\Omega_c \subset \mathcal{R}^n$ equipped with the Euclidean metric. This is equivalent to finding the minimizer of

$$I_{\text{hrm}}[\boldsymbol{\xi}] = \int_{\Omega} \sqrt{g} \sum_i (\nabla \xi_i)^T G^{-1} (\nabla \xi_i)\, \mathrm{d}\boldsymbol{x}. \tag{20}$$

*Brackbill's direction control functional.* Extending Winslow's variable diffusion method [34] and the harmonic mapping method, Brackbill [5] develops a functional which takes into consideration both mesh concentration and mesh alignment (or direction control). The functional can be written in a more general form (e.g., see [16,17])

$$I_{\text{brb}}[\boldsymbol{\xi}] = \int_{\Omega} \sum_i (\nabla \xi_i)^T G_i^{-1} (\nabla \xi_i)\, \mathrm{d}\boldsymbol{x}, \tag{21}$$

for suitable user defined functions $G_i$. These functions are generally required to be semi-positive definite. Depending upon the desired mesh properties, they may be selected in a variety of ways. For instance, if the mesh concentration is of primary concern, then it is natural to choose $G_i = w(\boldsymbol{x})I$, which simply leads to Winslow's functional. In the special case $G_i = \frac{1}{\sqrt{g}}G$, the functional $I_{\text{brb}}$ reduces to a harmonic mapping. If the mesh is expected to align with a certain prescribed direction field $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$, then $G_i$ can be chosen such that $(\nabla\xi_i)^T G_i^{-1}(\nabla\xi_i) = |\boldsymbol{v}_i \times \nabla\xi_i|^2$. If the mesh lines are expected to be highly orthogonal, one may first define an orthogonal reference mesh $\tilde{\boldsymbol{\xi}}(\boldsymbol{x})$ and then choose the monitor $G_i$ such that $(\nabla\xi_i)^T G_i^{-1}(\nabla\xi_i) = |\nabla\tilde{\xi}_i \times \nabla\xi_i|^2$. It is often advantageous to use a combination of these control functionals in order to achieve a balance among mesh properties.

Since the above methods are interpretable as variational generalizations of the one dimensional equidistribution principle (12) or (13), it is not surprising that they share more or less the feature of the one dimensional mesh adaptation, i.e., they have a certain degree of success to concentrate mesh points near the regions where the monitor function or the control function is large. However, due largely to the complication arising from boundary constraints, the relation between the monitor function and the mesh behavior is more complicated in higher dimensions. More importantly, most of these methods are formulated on the basis of intuitive and/or geometrical considerations. As a consequence, the choice of the monitor function, while being clearly defined for almost every method from such considerations, can be ad hoc in the sense that the chosen monitor function does not minimize or is not even related to a certain error bound. Furthermore, except for very special cases such as a harmonic mapping on a convex domain $\Omega_c$, the questions of existence and smoothness of a non-singular mapping $\boldsymbol{x}(\boldsymbol{\xi})$ in multi-dimensions are far from clear.

### *3.3. A new variational approach*

In the afore-mentioned variational methods, the adaptive mapping $\boldsymbol{\xi}(\boldsymbol{x})$ is determined directly from the monitor function $G(\boldsymbol{x})$. $G$ may be chosen with intent that an estimate or an indicator of the discretization error is more or less evenly distributed over the adaptive mesh. However, as mentioned before, the precise relation between the mesh and the monitor function (and therefore the discretization error) is unclear. In practical computations one may observe a qualitative improvement in solution accuracy using the above types of mesh generation, but it is unclear whether or not the mesh is optimal in terms of minimization of the actual error. To deal with this issue, Huang [15] formulates a functional based more directly on error distribution. He observes that the local error distribution $E(\boldsymbol{x})$ for a discretization can often be written in the form

$$E(\boldsymbol{x}) = \sqrt{\mathrm{d}\boldsymbol{\xi}^T \boldsymbol{J}^T G \boldsymbol{J}\, \mathrm{d}\boldsymbol{\xi}},$$

where $G(\boldsymbol{x})$ is an $n \times n$ symmetric positive definite matrix. For instance, $G$ is the Hessian of the interpolation function when the $L^2$-norm is used to measure the error for linear interpolation [11]. According to the equidistribution principle, an ideal adaptive mesh would have $E(\boldsymbol{x})$ uniformly distributed over the entire domain. This is equivalent to having

$$A \equiv \boldsymbol{J}^{-1} G^{-1} \boldsymbol{J}^{-T} = cI, \tag{22}$$

for some constant $c$. If $\lambda_i, 1 \leqslant i \leqslant n$, are the eigenvalues of $A$, (22) is true if and only if the following two criteria are satisfied:

$$\text{Isotropy Criterion:}\quad \lambda_1 = \cdots = \lambda_n; \tag{23}$$

$$\text{Uniformity Criterion:}\quad \sqrt{\prod \lambda_i} = \text{constant}. \tag{24}$$

Noting that

$$\sum_i \lambda_i = \operatorname{tr}(A) = \sum_i \bigl(\nabla \xi^i\bigr)^T G^{-1} \nabla \xi^i,$$

$$\prod_i \lambda_i = \det(A) = \frac{1}{J^2 g},$$

where $g$ is the determinant of $G$ and using the arithmetic-geometric mean inequality, Huang shows that (23) is fulfilled if and only if the inequality

$$\frac{n^{n/2}}{J} \leqslant \sqrt{g}\Bigl(\sum_i (\nabla \xi_i)^T G^{-1} \nabla \xi_i\Bigr)^{n/2}, \tag{25}$$

becomes an equality. Consequently, the isotropy criterion (23) may be satisfied approximately by minimizing the integral of the difference of both sides of (25), or equivalently the functional

$$I_{\text{iso}}[\boldsymbol{\xi}] = \frac{1}{2}\int_{\Omega} \sqrt{g}\Bigl(\sum_i (\nabla \xi_i)^T G^{-1} \nabla \xi_i\Bigr)^{n/2} \mathrm{d}\boldsymbol{x}. \tag{26}$$

The uniformity condition (24) is equivalent to satisfying $J\sqrt{g} = \text{const}$, which in turn is satisfied if and only if the inequality

$$\int_{\Omega_c} \mathrm{d}\boldsymbol{\xi} \leqslant \Bigl[\int_{\Omega} \frac{\sqrt{g}}{(J\sqrt{g})^q}\,\mathrm{d}\boldsymbol{x}\Bigr]^{1/q}, \tag{27}$$

where $q > 1$ is any real number, is an equality (see [13, p. 143]). Thus we may achieve (24) by minimizing the functional

$$I_{\text{ep}}[\boldsymbol{\xi}] = \int_{\Omega} \frac{\sqrt{g}}{(J\sqrt{g})^q}\,\mathrm{d}\boldsymbol{x}. \tag{28}$$

To balance between the goals of achieving both (23) and (24), the combined functional

$$I_{\text{new}}[\boldsymbol{\xi}] = \theta \int_{\Omega} \sqrt{g}\Bigl(\sum_i (\nabla \xi_i)^T G^{-1} \nabla \xi_i\Bigr)^{nq/2} \mathrm{d}\boldsymbol{x} + (1-2\theta) n^{nq/2} \int_{\Omega} \frac{\sqrt{g}}{(J\sqrt{g})^q}\,\mathrm{d}\boldsymbol{x}, \tag{29}$$

is defined, where $0 \leqslant \theta \leqslant 1$ is a parameter. By (25) this functional is non-negative for all $\theta \in [0,1]$, and when $\theta \in (0, 1/2)$, both terms in $I_{\text{new}}$ are non-negative. By shifting the parameter $\theta$, one may adjust the relative emphasis on mesh isotropy or mesh uniformity.

### *3.4. Boundary conditions*

There are at least two natural ways to specify the boundary conditions for the admissible mapping in the minimization of the above functionals. The first is to use the natural boundary conditions

$$\nabla \xi_i \cdot \nabla \xi_j = 0 \quad \text{for } i \neq j. \tag{30}$$

This essentially requires that the mesh lines of constant $\xi_i$ $(i = 1, \ldots, n)$ are orthogonal at the boundary.

The second is to use a lower-dimensional mesh equation to determine a Dirichlet type boundary condition. In two dimensions, given a boundary segment $\Gamma$ of $\partial\Omega$ and corresponding boundary segment $\Gamma_c$ of $\Omega_c$, denote by $s$ and $\zeta$ the arc-length coordinates of $\Gamma$ and $\Gamma_c$, respectively. Then one requires that the restriction of the mapping $\boldsymbol{\xi}(\boldsymbol{x})$ on $\Gamma$, denoted by $s(\zeta)$, should satisfy

$$M(s)\frac{\partial s}{\partial \zeta} = \text{const}, \tag{31}$$

where $M(s) = \boldsymbol{\tau}^T G \boldsymbol{\tau}$ is the projection of the higher dimensional monitor function onto the boundary tangential direction $\boldsymbol{\tau}$.

### *3.5. Moving mesh equation*

The above variational approaches have only been described for generating one adaptive mesh for a specified monitor function. When one solves time dependent problems, an adaptive mesh is needed for each time step. One way to do this is to simply use the monitor function defined at each time level to generate a corresponding adaptive mesh by the above procedure. This is generally not an efficient procedure and can cause abrupt mesh change in time. To provide smoother mesh movement and increased efficiency, Huang and Russell [16,17] use the gradient flow equation

$$\frac{\partial \boldsymbol{\xi}}{\partial t} = -\frac{1}{\tau}\frac{\delta I}{\delta \boldsymbol{\xi}}, \tag{32}$$

where $\tau > 0$ is a time smoothing factor, to define the time dependent mapping $\boldsymbol{\xi}(\boldsymbol{x}, t)$. This is a parabolic system whose limit as $\tau \to 0$ is the Euler–Lagrange equation for minimizing the functional $I$. By moderating $\tau$, one may balance the effects of mesh adaptation and mesh movement in time. The larger $\tau$, the smoother the mesh moving, and the smaller $\tau$, the more accurate the mesh adaptation.

## 4. Numerical examples

In this section, we present a short numerical comparison of four mesh adaptation methods: the GCL method (8), Winslow's method (16), the harmonic mapping method (20), and the method based on the new functional (29). They are so selected partly because of our research interest and also because we believe they are good representatives in each group, the location based methods or the velocity based ones. But we would like to point out that many other methods, such as the MFE, the deformation method, and Knupp's Jacobian-weighted variational method, also have their potentials and deserve further investigations.

We now briefly describe the implementations of the four methods. For the GCL method, we typically take the computational domain $\Omega_c$ to be the physical domain $\Omega$ and the initial mapping $\boldsymbol{x}(\boldsymbol{\xi}, 0) = \boldsymbol{\xi}$. Given a fixed mesh on $\Omega_c$, we approximate $\boldsymbol{x}(\boldsymbol{\xi}, t)$ by continuous piecewise linear polynomials at each time $t$. Solving the minimization problem for $I_{\text{gcl}}$ in (8) involves solving a system of ODEs for the unknowns $\boldsymbol{x}(\boldsymbol{\xi}_j, t)$ at the nodal points of the mesh. The mapping $\boldsymbol{x}(\boldsymbol{\xi}, t)$ is calculated by integrating the ODEs with a standard time integrator, here a Singly Diagonally Implicit Runge–Kutta (SDIRK) method [7].

For the three location based methods, the formulations have been given in the above sections in terms of the mapping $\boldsymbol{\xi}(\boldsymbol{x})$ from $\Omega$ to $\Omega_c$. If it is computed directly, one then needs to invert the mapping to define the adaptive mesh on $\Omega$ from the fixed one on $\Omega_c$, so it is often more convenient to instead derive the equations for the inverse mapping $\boldsymbol{x}(\boldsymbol{\xi})$. This is done using the chain rule and various relationships between the covariant and contravariant base vectors, e.g., see [14]. In the time dependent case it is then straightforward to either solve for $\boldsymbol{x}(\boldsymbol{\xi})$ iteratively or by integrating the moving mesh PDE (32) in time to find the successively updated mesh.

We consider three examples with $\Omega = (0,1) \times (0,1)$ and the monitor function defined as follows:

**Example A.**

$$\rho(x,y) = 1 + 10\exp\left(-50\left(y - \frac{1}{2} - \frac{1}{4}\sin(2\pi x)\right)^2\right). \tag{33}$$

**Example B.**

$$\rho(x,y) = 1 + 10\exp\left(-50\left|\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 - \left(\frac{1}{4}\right)^2\right|\right). \tag{34}$$

**Example C.**

$$\rho(x,y,t) = 1 + 10\exp\left(-50\left|\left(x - \frac{1}{2} - \frac{1}{4}\cos(2\pi t)\right)^2 + \left(y - \frac{1}{2} - \frac{1}{4}\sin(2\pi t)\right)^2 - \left(\frac{1}{10}\right)^2\right|\right). \tag{35}$$

Examples A and B require generating an adaptive mesh for the specified monitor function. Example C requires generating a moving adaptive mesh from the time dependent monitor function.

In all three examples, we set $w = 1$, $\boldsymbol{u} = 0$, and use the non-crossing boundary condition (9) for the GCL method. For Winslow's method, we set $w = \rho$ and use one-dimensional equidistribution (31) for the boundary condition. For the harmonic mapping method, we choose the monitor function $G = Q * \mathrm{Diag}(\rho, 1/\rho) * Q^T$, where $Q = \frac{1}{\|\nabla\rho\|^2}(\nabla\rho, (\nabla\rho)^{\perp})$ is the matrix defined with the gradient vector of $\rho$. For the method based on the new functional (29), we choose $G = \rho I$, $q = 2$ and $\theta = 0.1$, and use the boundary condition from (31). (The choice of $q = 2$ and $\theta = 0.1$ generally works well by suitably emphasizing mesh uniformity – also see [15].) In all cases we use the computational domain $\Omega_c = \Omega$ and a uniform $40 \times 40$ mesh on it.

The normalized quantity

$$EP(\boldsymbol{x}) = \frac{J\rho}{\int_{\Omega} \rho \, \mathrm{d}\boldsymbol{x}},$$

is used to measure how close the mesh distribution comes to equidistributing the monitor function. For the GCL method, from (4) this benchmarking quantity should be 1 over the entire domain since it is initially. For the method based on the new functional (29), we also have $EP(\boldsymbol{x}) = 1$ in the limit $\theta \to 0$. In the cases of Winslow's method and the harmonic mapping method, there are no theoretical results on the quantitative relation between the monitor function and the mesh density. Nevertheless, $w$ and $\rho$ have traditionally been used to qualitatively control the mesh density, and $EP$ provides some indication on how reliably they perform.

It should be pointed out that although only the equidistribution measure $EP(\boldsymbol{x})$ is considered here, additional measures of other mesh properties such as the regularity should be included for the purpose of convergence analysis, e.g., see [19] for asymptotic analysis for the interpolation error on adaptive meshes.

We present in Fig. 1 the adaptive meshes obtained by the four moving mesh methods for Example A. In this and the other examples, the region where the monitor function $\rho$ takes its maximum values (and an ideal mesh would be most concentrated) is highlighted as a thick curve in the figures. Note that in this example all four methods produce the higher mesh concentration in regions of higher $\rho$ values. We also list in Table 1 the maximum and minimum values of $EP$. The GCL method has the smallest deviation from constant 1, followed by the method based on the new functional (29), then Winslow's method, and finally the harmonic mapping method. This is not surprising since the GCL method satisfies (4), and the

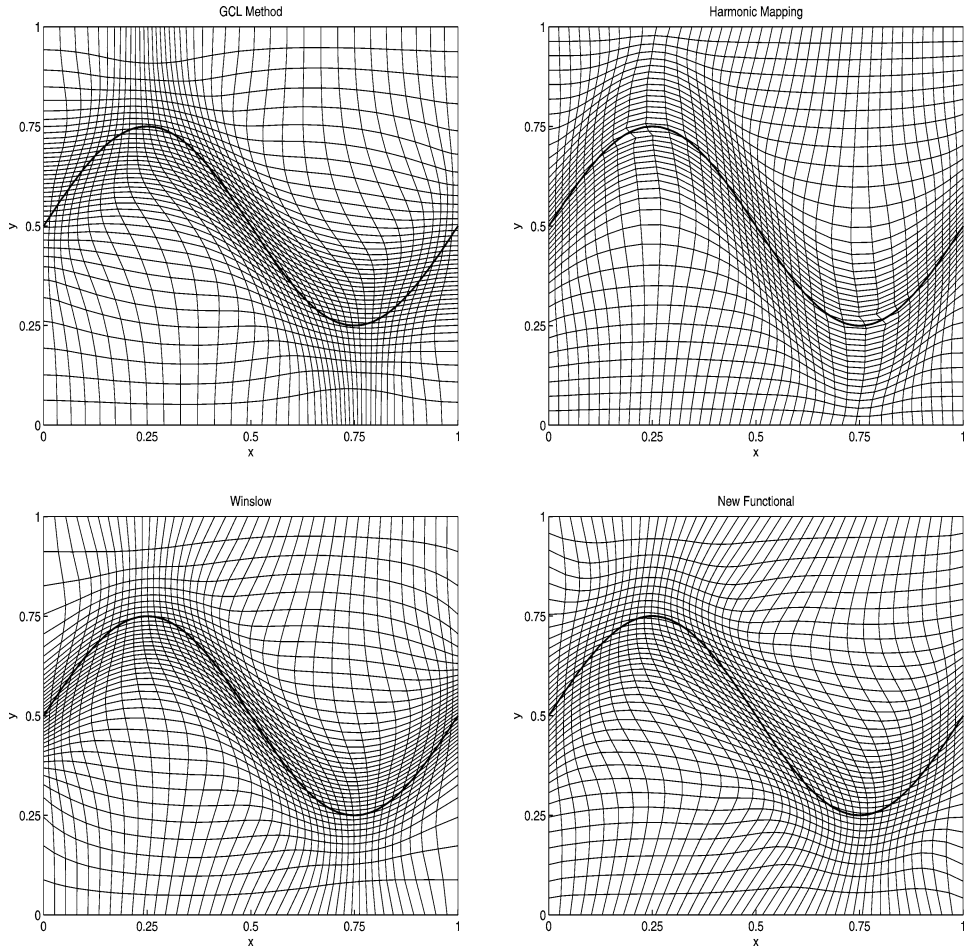


Fig. 1. Example A: The adaptive meshes obtained by the GCL method (top left), Winslow's method (bottom left), the harmonic mapping method (top right), and the method based on the new functional (bottom right).

new functional takes into account the equidistribution of $\sqrt{g} = \rho$. For the method based on Winslow's functional and the harmonic mapping method, even though there is no quantitative relation between $J$ and $\rho$, $EP$ is still reasonably constant. The methods all perform well in this case.

Table 1
Maximum and Minimum values of $EP$

| | Example A | | Example B | |
|---|---|---|---|---|
| | max($EP$) | min($EP$) | max($EP$) | min($EP$) |
| GCL | 1.059 | 0.823 | 1.096 | 0.899 |
| Winslow | 2.380 | 0.473 | 1.355 | 0.463 |
| Harmonic | 4.706 | 0.404 | 15.82 | 0.453 |
| New Fnl | 1.280 | 0.568 | 1.400 | 0.620 |

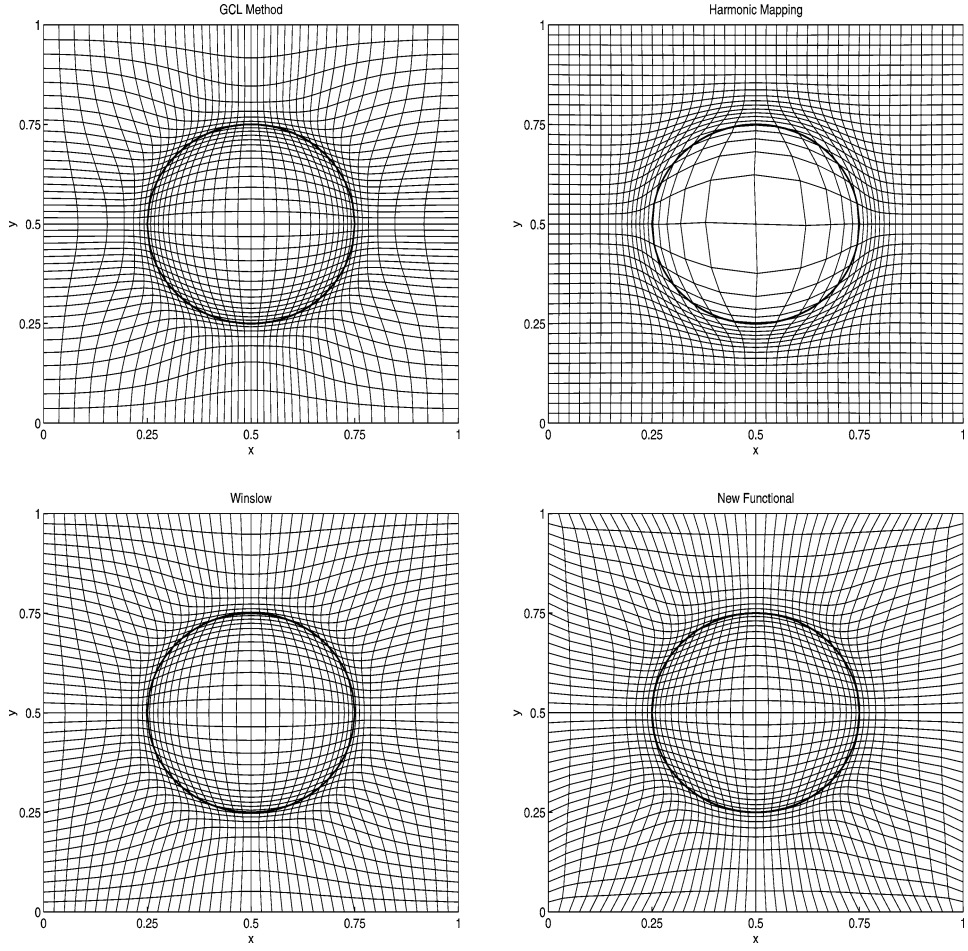


Fig. 2. Example B: The adaptive meshes and the distributions of the benchmarking quantity $EP$ obtained by the GCL method (top left), Winslow's method (bottom left), the harmonic mapping method (top right), and the method based on the new functional (bottom right).

Fig. 2 shows the results for Example B, and the extreme values of *EP* are given in Table 1. In this example, all the methods except the harmonic mapping method perform well, and the meshes are concentrated around the places of higher $\rho$ values. For the harmonic mapping, the circle with the highest mesh concentration is slightly outside the circle with the maximum $\rho$ values, and the mesh in the central region is much sparser than in the outer region although $\rho$ takes approximately the same value 1 in both places. This phenomenon has also been observed in [6], where it is attributed to the so-called *two-dimensional effects* of the method. However, the precise relation between the monitor function and the mesh distribution is still unclear for this method. A detailed study of this issue is clearly important for determining the breadth of practical applicability of the harmonic mapping method.

Finally, Figs. 3–6 show for Example C the adaptive meshes at different time instances for the four methods. For Winslow's method and the new functional (29), the higher mesh concentration region follows closely the evolution of function $\rho$. The benchmarking quantity is always bounded between 0.54 and 1.86 for the case of Winslow's method, and between 0.68 and 1.31 for the new functional. For the

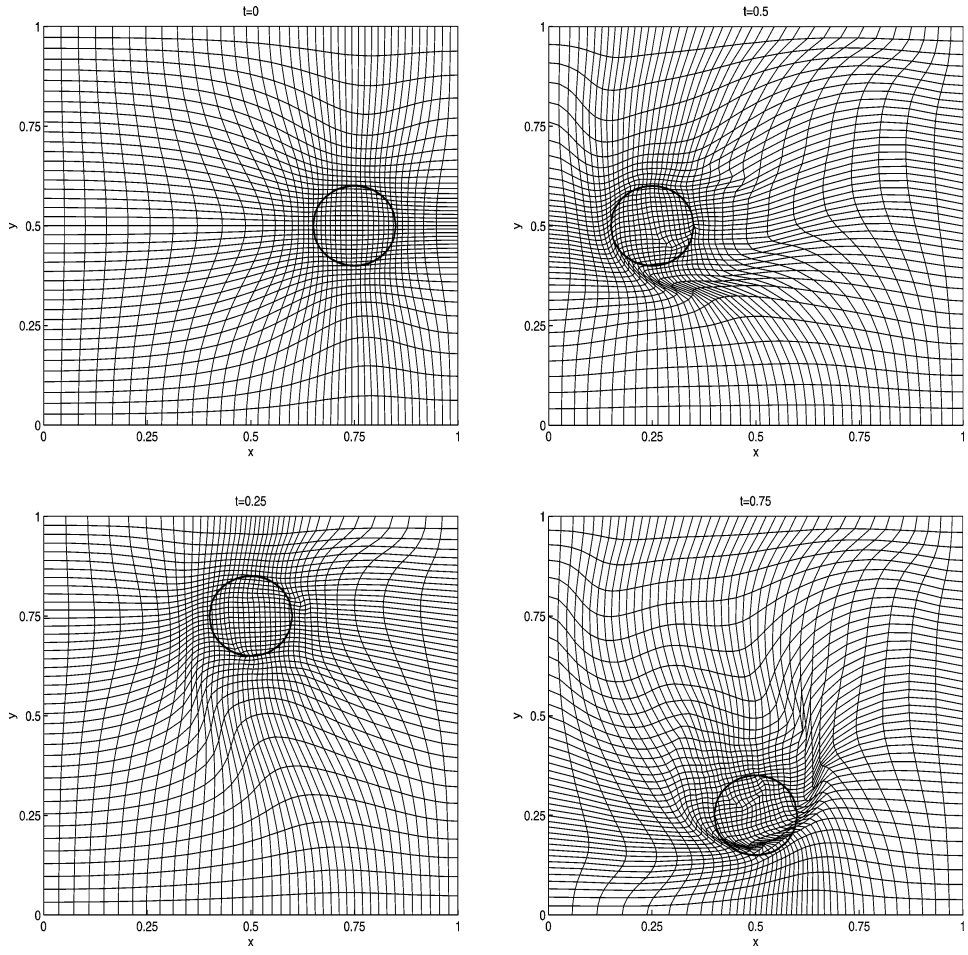


Fig. 3. Example C: The adaptive meshes and the distributions of the benchmarking quantity *EP* obtained by the GCL method at $t = 0, 0.25, 0.5, 0.75$.

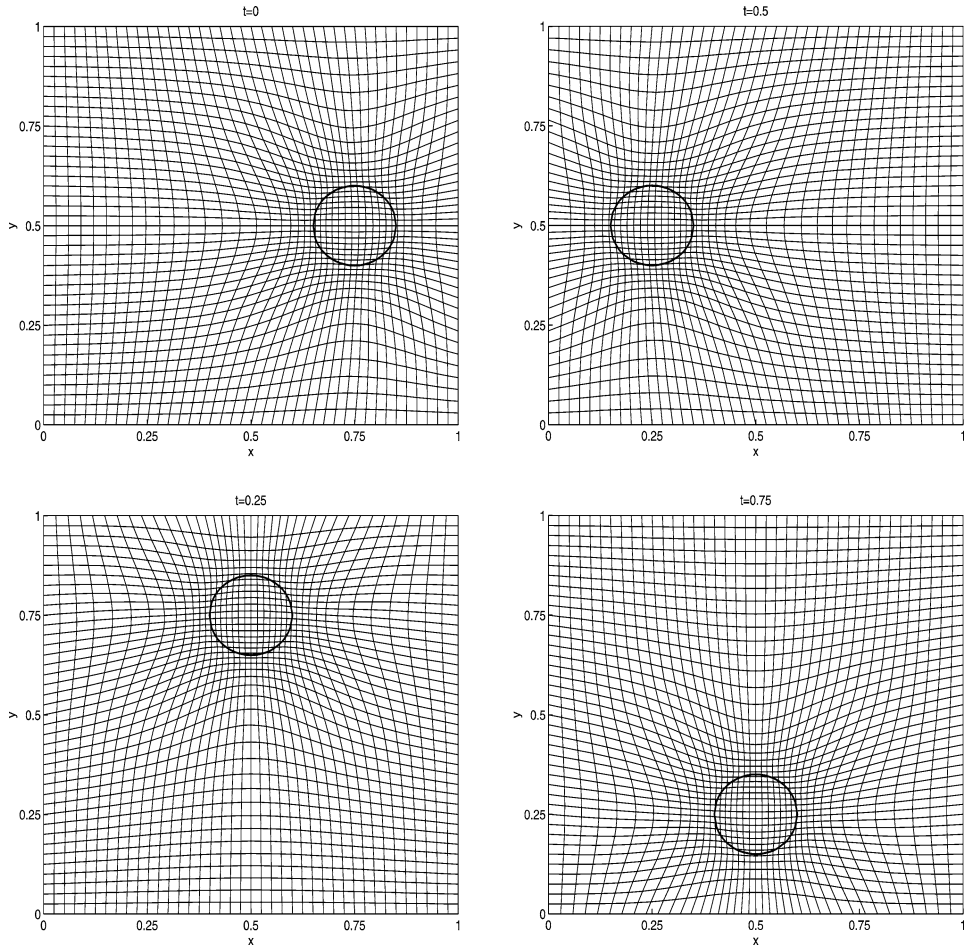


Fig. 4. Example C: The adaptive meshes and the distributions of the benchmarking quantity *EP* obtained by Winslow's method at $t = 0, 0.25, 0.5, 0.75$.

harmonic mapping method, the mesh concentration region is again slightly away from the desired region, although it moves in time at the same speed as $\rho$. In the case of the GCL method, however, even though the *EP* is not far away from 1 (between 0.4 and 2.1 for all time), the adaptive mesh becomes increasingly skew and the mesh concentration becomes slowly misplaced. In part, this difficulty can be overcome by taking care to avoid the degeneracy of the mesh, e.g., either by changing the mesh connectivity [18] or by generating the adaptive meshes at selected times by the static version of the GCL method.

## 5. Discussion

We have considered a variety of moving mesh methods which fall into two groups: (i) velocity based methods such as the moving finite element method, the method based on the geometric conservation law, and the deformation method, and (ii) the location based methods, which include mainly Winslow's

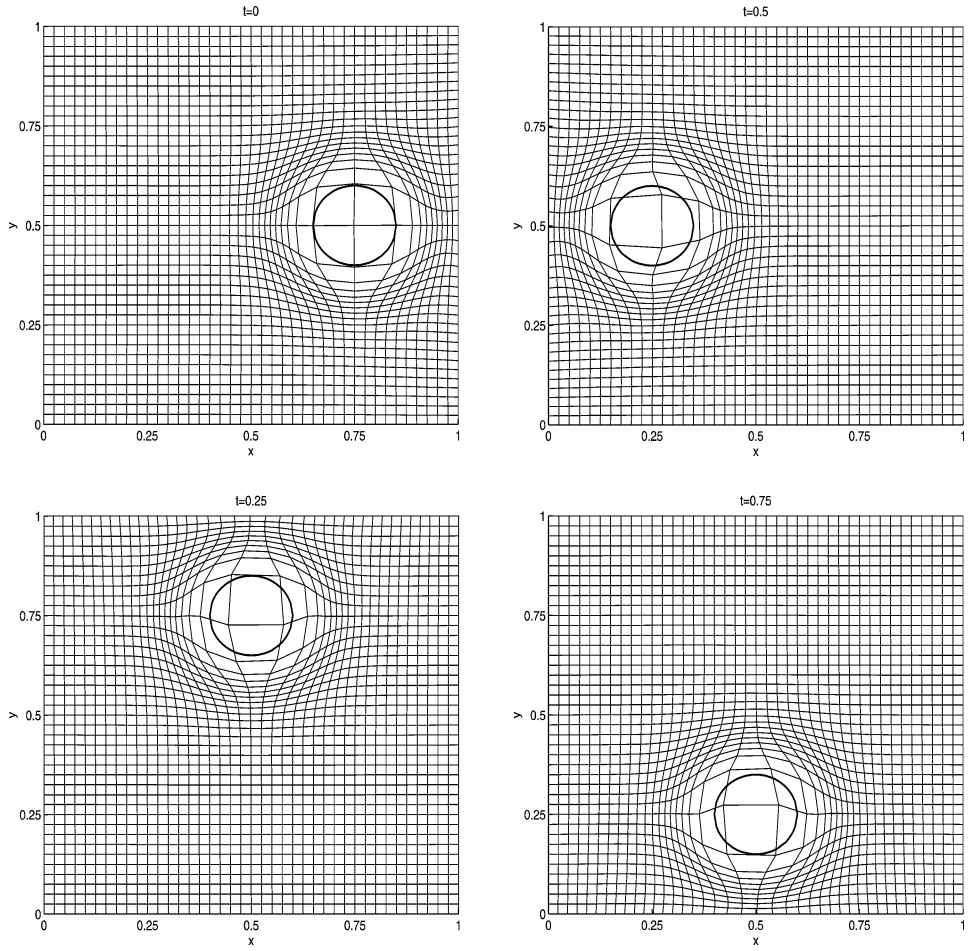


Fig. 5. Example C: The adaptive meshes and the distributions of the benchmarking quantity *EP* obtained by the harmonic mapping method at $t = 0, 0.25, 0.5, 0.75$.

variable diffusion method, the harmonic mapping method, Brackbill's direction control method, and the method based on a new functional developed recently in [15]. There are two major differences between these two types of approaches. First, with the location based methods, the mesh behavior is controlled by a monitor function through a global integral of the mapping, which, except for the new functional (29), can make it difficult to predict precisely the effects of monitor functions (and consequently to devise a monitor function to realize the desired mesh behavior). For a velocity based method and the functional (29), the relationship between the mesh density and the monitor function is clearer, making it possible to construct adaptive meshes based on the error or residual distribution. Second, with the velocity based methods the adaptive mesh is the result of time integration of the mesh velocity fields. Thus, the mesh history for previous times will influence the mesh behavior at the current time, and the moving mesh can easily become increasingly skewer, just as for the classical Lagrangian method. In contrast, for the location based methods the adaptive mesh at a given time is essentially determined by the monitor function at that time level. As a result, it is relatively stable over the long run.

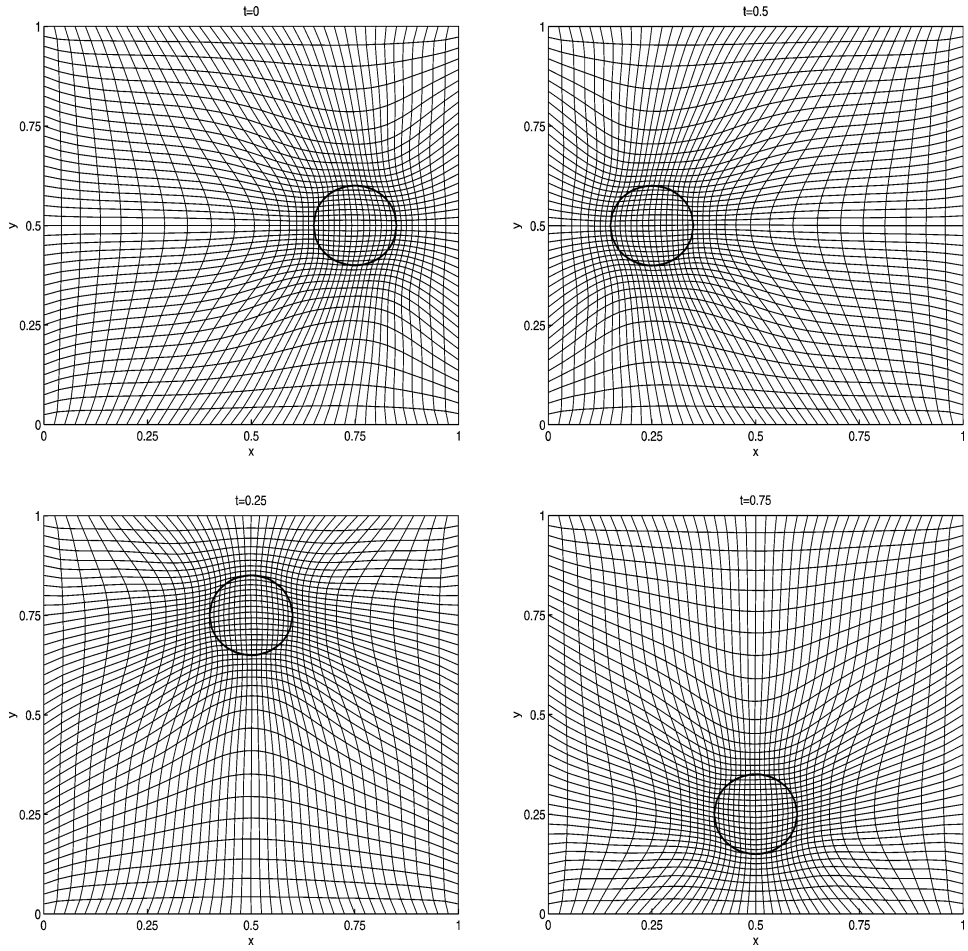


Fig. 6. Example C: The adaptive meshes and the distributions of the benchmarking quantity *EP* obtained by the method based on the new functional at $t = 0, 0.25, 0.5, 0.75$.

As well, it is possible to formulate the functionals with the location based approach to control various mesh properties (such as orthogonality).

A brief numerical comparison of several of these moving mesh methods has been presented to demonstrate these points.

## References

[1] M.J. Baines, Moving Finite Elements, Oxford University Press, New York, 1994.
[2] R.E. Bank, K. Smith, Mesh smoothing using a posteriori error estimates, SIAM J. Numer. Anal. 37 (1997) 921–935.
[3] P. Bochev, G. Liao, G. Pena, Analysis and computation of adaptive moving grids by deformation, Numer. Meth. PDEs 12 (1996) 489–506.
[4] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (1982) 342–368.

[5] J.U. Brackbill, An adaptive grid with direction control, J. Comput. Phys. 108 (1993) 38–50.
[6] W. Cao, W. Huang, R.D. Russell, A study of monitor functions for two dimensional adaptive mesh generation, SIAM J. Sci. Comput. 20 (1999) 1978–1994.
[7] W. Cao, W. Huang, R.D. Russell, A moving mesh method based on the geometric conservation law, SIAM J. Sci. Comput. 24 (2002) 118–142.
[8] N. Carlson, K. Miller, Design and application of a gradient-weighted moving finite element code, Part I, in 1D, SIAM J. Sci. Comput. 19 (1998) 728–765.
[9] N. Carlson, K. Miller, Design and application of a gradient-weighted moving finite element code, Part II, in 2D, SIAM J. Sci. Comput. 19 (1998) 766–798.
[10] B. Dacorogna, J. Moser, On a partial differential equation involving the Jacobian determinant, Ann. Inst. Henri Poincaré Anal. Non Linéaire 7 (1990) 1–26.
[11] E.F. D'Azevedo, R.B. Simpson, On optimal triangular meshes for minimizing the gradient error, Numer. Math. 59 (1991) 321–348.
[12] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (1991) 450–476.
[13] G.H. Hardy, J.E. Littlewood, G. Pólya, Inequalities, 2nd Edition, Cambridge University Press, Cambridge, 1967.
[14] W. Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, J. Comput. Phys. 171 (2001) 753–775.
[15] W. Huang, Variational mesh adaptation: isotropy and equidistribution, J. Comput. Phys. 174 (2001) 903–924.
[16] W. Huang, R.D. Russell, A high dimensional moving mesh strategy, Appl. Numer. Math. 26 (1997) 63–76.
[17] W. Huang, R.D. Russell, Moving mesh strategy based upon a gradient flow equation for two-dimensional problems, SIAM J. Sci. Comput. 20 (1999) 998–1015.
[18] W. Huang, D.M. Sloan, Generate nondegenerate meshes using the GCL moving mesh method and by changing the mesh connectivity, in preparation.
[19] W. Huang, W. Sun, Variational mesh adaptation II: Error estimates and monitor functions, J. Comp. Phys. 184 (2003) 619–648.
[20] A.A. Johnson, T.E. Tezduyar, Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces, Comput. Meth. Appl. Mech. Engrg. 119 (1994) 73–94.
[21] P.M. Knupp, Mesh generation using vector-fields, J. Comput. Phys. 119 (1995) 142–148.
[22] P.M. Knupp, Jacobian-weighted elliptic grid generation, SIAM J. Sci. Comput. 17 (1996) 1475–1490.
[23] P.M. Knupp, N. Robidoux, A framework for variational grid generation: conditioning the Jacobian matrix with matrix norms, SIAM J. Sci. Comput. 21 (2000) 2029–2047.
[24] D.R. Lynch, Unified approach to simulation on deforming elements with application to phase change problems, J. Comput. Phys. 47 (1982) 387–411.
[25] G.J. Liao, D. Anderson, A new approach to grid generation, Appl. Anal. 44 (1992) 285–298.
[26] J. Moser, On the volume elements of a manifold, Trans. Amer. Math. Soc. 120 (1965) 286–294.
[27] K. Miller, R.N. Miller, Moving finite elements I, SIAM J. Numer. Anal. 18 (1981) 1019–1032.
[28] K. Miller, Moving finite elements II, SIAM J. Numer. Anal. 18 (1981) 1033–1057.
[29] J.T. Oden, T. Stroubolis, P. Devloo, Adaptive finite element methods for the analysis of inviscid compressible flow, Part I: Fast refinement, unrefinement and moving mesh methods for unstructured meshes, Comput. Methods Appl. Mech. Engrg. 59 (1986) 327–362.
[30] B. Semper, G. Liao, A moving grid finite-element method using grid deformation, Numer. Methods in PDEs 11 (1995) 603–615.
[31] P.D. Thomas, C.K. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA J. 17 (1979) 1030–1037.
[32] J.F. Thompson, Z.A. Warsi, C.W. Mastin, Numerical Grid Generation, North-Holland, Amsterdam, 1985.
[33] A. Winslow, Numerical solution of the quasi-linear Poisson equation in a nonuniform triangle mesh, J. Comput. Phys. 1 (1967) 149–172.
[34] A. Winslow, Adaptive mesh zoning by the equipotential method, Lawrence Livemore Laboratory Report UCID-19062, 1981, unpublished.