MOVING MESH FINITE ELEMENT METHODS BASED ON HARMONIC MAPS

RUO LI *, WENBIN LIU †, TAO TANG ‡, AND PINGWEN ZHANG §

Abstract. This paper is devoted to the applications of a class of moving mesh finite element methods based on harmonic maps. We review some recent work of the authors on solving PDES, variational inequalities and optimal control problems by use of the moving mesh techniques.

Key words. finite element method, mesh adaptivity, harmonic map, variational inequality, optimal control, *A posteriori error* estimates

1. Introduction. Adaptive mesh methods have been widely used in the last decade for solving partial differential equations that involve large solution variations, such as shock waves, boundary layers or wave fronts. In order to resolve such nonuniformities using a minimum number of mesh points, it is desirable to concentrate the mesh within the areas of large solution variation. In practice, there are three type of adaptive methods using finite element approach, namely h-method (mesh refinement), p-method (order enrichment) and r-method (mesh motion). The r-method, which will be studied in this work, is also known as moving mesh method. In the h-method, the overall method contains two parts, a solution algorithm and a mesh selection algorithm. In principle, these two parts are independent of each other in the sense that the change of the PDEs will affect the first part only. However, in most of the existing moving mesh methods, these two parts are strongly associated with each other such that the change of the PDEs will result in the rewritting of the whole code. In this work, we will use a moving finite element scheme which consists of two *independent* parts: a mesh-redistribution algorithm and a solution algorithm. The mesh-redistribution algorithm is independent of the solution algorithm which can be any of the standard codes for the given PDEs. In other words, the change of the PDEs will not affect the mesh-redistribution algorithm.

Mesh adaptation can be static or dynamic (moving). In the static case a new mesh is generated at a fixed time intervals. In a dynamic or a moving mesh adaptation [1, 5, 11, 15, 17, 19], the mesh changes continuously in space and time to adapt to the dynamic changes of a time-evolving solutions. It is a challenging problem to generate an efficient moving mesh in two or more dimensions, especially when the underlying solution develops complicated structures and becomes singular or nearly singular. In the past, several moving mesh techniques have been introduced, in which one of the most advocated methods is the one based on solving elliptic PDEs first proposed by Winslow [24]. Winslow's formulation requires the solution of a nonlinear, Poisson-like equation to generate a mapping from a regular domain in a parameter space Ω_c to an irregularly shaped domain in physical space Ω. By connecting points in the physical space corresponding to discrete points in the parameter space, the physical domain can be covered with a computation mesh suitable for the solution of finite difference/element equations. There are also many applications and extensions of Winslow's method, see e.g. Brackbill et al. [3, 4], Thompson et al. [23], and Ren and Wang [20].

The moving mesh methods to be used in this research will be based on *harmonic mapping*. The research of important impact on the present work is due to Dvinsky [6] where a useful framework for adaptive grid generation based on the harmonic maps was described. Our approach is similar to Dvinsky's scheme, except that

- (a): we will use the finite element approach and
- (b): a more detailed and robust mesh-moving strategy will be designed.

*School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China.

†CBS & Institute of Mathematics and Statistics, The University of Kent, Canterbury, CT2 7NF England, Email: W.B.Liu@ukc.ac.uk

‡Department of Mathematics, The Hong Kong Baptist University, Kowloon Tong, Hong Kong. Email: ttang@math.hkbu.edu.hk

§School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China. Email: pzhang@sxx0.math.pku.edu.cn

The finite element approach is found to be more reliable and efficient than the finite difference approach when solving problems with large variations or complex solution domains. In part (b), we make sure that the map between the logical and the physical domains are kept harmonic, even after long time of numerical integration. To this end, we construct the harmonic map between the physical mesh and the logical mesh by an *iteration* procedure. Each iteration step is to move the mesh *closer* to the harmonic map. The idea of iteration mapping was proposed by Tang and Trummer [22], Liu and Tang [18] and Ren and Wang [20].

Our recent study has extended the moving mesh methods to solve variational inequalities and optimal control problems. Finite element schemes loose accuracy when approximating variational inequalities due to the existence of free boundaries. In our work, computational meshes are constructed by combining harmonic mapping and sharper *a posteriori* error estimators which are normally used in h-version adaptive finite element approximation. These estimators depend solely on the discrete solution and data and impose no constraints between consecutive time steps. Selecting appropriate error monitors is one of the most important issues that have to be addressed in implementing the moving mesh methods. It seems that the commonly used gradient based monitors are not suitable in solving the variational inequalities and optimal control problems. In this paper, we will show that the monitors based on equivalent *a posteriori* error estimates can handle the free boundaries very well.

The paper is organized in two main parts. Section 2 introduces our moving mesh strategy. In particular, a mesh motion algorithm based on harmonic maps will be introduced. The second part is devoted to the application of our moving mesh schemes, with Section 3 the application for PDEs, Section 4 the application for variational inequalities and Section 5 the application for optimal control problems. It is observed from the numerical experiments that the mesh-redistribution strategy based on the harmonic maps adapts the mesh extremly well to the solution without producing skew elements for the multi-dimension computations.

2. Moving mesh methods based on harmonic maps. The basic idea of the moving mesh method is to equidistribute some functional of the numerical solution on the mesh. In 1-D case, it is called *equidistribution principle* as proposed by de Boor [2]. In the moving mesh method, the functional to be equidistributed is named *monitor*. The natural extension of the equidistribution principle is to construct the *harmonic maps* from physical domain Ω to the computational domain Ω_c by minimizing the energy functional

$$E(\vec{\xi}) = \sum_k \int_\Omega G^{ij} \frac{\partial \xi^k}{\partial x^i} \frac{\partial \xi^k}{\partial x^j} dx \tag{2.1}$$

in which the standard summation convention is assumed. In the functional (2.1), G^{ij} is the inverse matrix of the monitor and can be regarded as the metric matrix on Ω. The existence and uniqueness of harmonic maps are proved by Hamilton [10], Schoen and Yau [21]. A framework for adaptive grid generation based on the Hamilton-Schoen-Yau theory was proposed by Dvinsky [6]. In Li-Tang-Zhang [12], Dvinsky's method was extended to provide an efficient solver for mesh-redistribution. The key idea is to construct the harmonic map between the physical space and a parameter space by an *iteration* procedure. Each iteration step is to move the mesh *closer* to the harmonic map. In this paper, we only outline the basic idea of the moving mesh method – the detailed procedure, derivations and formulas for the moving mesh strategy based on harmonic mapping can be found in [12].

The Euler-Lagrange equation of the functional (2.1) is of the form

$$\frac{\partial}{\partial x^i} \left(G^{ij} \frac{\partial \xi^k}{\partial x^j} \right) = 0\,. \tag{2.2}$$

In principle, the desired harmonic mapping can be obtained numerically. However, in the numerical implementation what we need to compute is the inverse of the harmonic mapping instead of the harmonic mapping itself. The numerical approach to sway off the spot is to deform the given mapping to the harmonic one by solving the heat equation

$$\frac{\partial \xi^k}{\partial \tau} = \frac{\partial}{\partial x^i}\left(G^{ij}\frac{\partial \xi^k}{\partial x^j}\right). \tag{2.3}$$

Transforming back to the physical domain according to

$$\frac{\partial x^l}{\partial \tau} = -\frac{\partial x^l}{\partial \xi^k}\frac{\partial \xi^k}{\partial \tau}, \tag{2.4}$$

will lead to the map from $\vec{\xi}$ to $\vec{x}$.

In the first step, we prepare the mesh on the logical domain Ω_c. We choose a convex domain Ω_c as the logical domain. By solving the Poisson equation

$$\Delta\vec{\xi} = 0 \tag{2.5}$$

with Dirichlet boundary condition:

$$\vec{\xi}|_{\partial\Omega} = \vec{\xi}_b, \tag{2.6}$$

we obtain a mesh T_c in the logical domain. The above harmonic map is the same as the one originally proposed by Winslow [24]. Once the initial mesh $\vec{\xi}$ is obtained, it will be kept unchanged thoughout the computation. However, this initial mesh in Ω_c, denoted by $\vec{\xi}^{(0)}$, will be used as a reference grid. Once the solution u is computed at time step $t = t_n$, the inverse matrix of the monitor, G^{ij} (which in general depends on u), can be updated. By solving the Euler-Lagrange equation (2.2), we will obtain a mesh in the logical domain, denoted by $\vec{\xi}^*$. If the difference between this $\vec{\xi}^*$ and the initial mesh $\vec{\xi}^{(0)}$ is not small, we move the mesh in physical space and obtain the updated values for u in the resulting new grid based on the following princinple:

- (a) obtain the error of $\vec{\xi}$ between the solution of (2.2) and the fixed (initial) mesh in the logical domain,
- (b) obtain the direction and the magnitude of the movement for $\vec{x}$ by using the error of $\vec{\xi}$, and
- (c) update $\vec{u}$ on the new grid using some numerical interpolations.

This procedure is *repeated* until the difference between the $\vec{\xi}^*$ and the initial mesh $\vec{\xi}^{(0)}$ is sufficiently small. Then we can use some appropriate numerical methods, with the updated mesh in the physical space, to solve the given problems (such as PDEs, variational inequalities etc) to obtain the solution for $t = t_{n+1}$. This procedure is also illustrated by the following flowchart:

Solve the given problem (such as PDEs, variational inequalities etc) for $t = t_n$

$\Downarrow$

Update the mesh and solution based on an iteration procedure

$\Downarrow$

Solve the given problem (such as PDEs, variational inequalities etc) for $t = t_{n+1}$

In solving PDEs, the iteration procedure metioned above is given by Algorithm 1 in next section. In solving variational inequalities, an iteration procedure similar to Algorithm 1 can be found in Li-Liu-Tang [13].

In the remaining of this paper, we choose $G^{ij} = \frac{1}{M}\delta_{ij}$, where δ_{ij} is the Kronecker delta. We call the scalar function M as the monitor function.

3. Moving mesh methods for PDEs. Consider the following PDEs in a domain $\Omega \subset \mathbf{R}^n$

$$\vec{u}_t = L(\vec{u}) \quad \text{in} \quad \Omega \times [0,T] \tag{3.1}$$

with boundary and initial conditions

$$\begin{aligned} B\vec{u}|_{\partial\Omega} &= \vec{u}_b \quad \text{in} \quad \partial\Omega \times [0,T] \\ \vec{u}|_{t=0} &= \vec{u}_0 \quad \text{in} \quad \Omega\,. \end{aligned}$$

The complete iteration procedure of mesh-moving process can be summarized as the following:

Algorithm 1

(i) solve the Euler-Lagrange equation (2.2) to obtain $\vec{\xi}^*$;
(ii) judge if L_2-norm of $\vec{\xi}^* - \vec{\xi}^{(0)}$ is small. if yes, the iteration is over. otherwise, do (iii)-(vi).
(iii) using the difference $\vec{\xi}^* - \vec{\xi}^{(0)}$ to compute the mesh-moving vector $\delta\vec{x}$;
(iv) move the mesh to a new location based on the result in (iii);
(v) interpolate the numerical solutions from the old mesh to the new mesh;
(vi) go to (i).

We now present some numerical examples to demonstrate the performance of our moving mesh finite-element methods for solving time dependent PDEs.

EXAMPLE 3.1. Consider the wave equation

$$\frac{\partial U}{\partial t} - y\frac{\partial U}{\partial x} + x\frac{\partial U}{\partial y} = 0$$

on the unit circle with initial value

$$U(\vec{x},0) = \begin{cases} e^{32((x_1-1/2)^2+x_2^2)}, & \text{if} \quad (x_1-1/2)^2 + x_2^2 < 1/4, \\ e^{32((x_1+1/2)^2+x_2^2)}, & \text{if} \quad (x_1+1/2)^2 + x_2^2 < 1/4, \\ 0 & \text{elsewhere,} \end{cases}$$

and zero boundary condition.

The solution of Example 3.1 possesses a twin peak (of fixed shape) rotating counterclockwise around the origin. A linear finite element discretization based upon the moving mesh scheme is applied. The initial mesh is obtained from a quasi-uniform triangulation with 40×40 elements. As noted in [8] that some existing moving mesh techniques produce meshes with points sticking to the rotating peaks, causing the mesh to become increasingly skew until the computation eventually breaks down. It is observed from Figure 3.1 that the moving mesh scheme based on the harmonic maps has no such difficulty, and the mesh adapts extremely well to the solution without producing skew elements. For this problem, the monitor function is chosen as $\sqrt{1+6u^2+\|\nabla u\|_2^2}$.

EXAMPLE 3.2. Our second example is to compute a moving oblique shock. The governing equation for it is the Burgers' equation:

$$\frac{\partial U}{\partial t} = a\Delta U + UU_x + UU_y$$

defined in the unit square $\Omega = (0,1)\times(0,1)$. The initial condition and Dirichlet boundary condition are chosen such that the exact solution to the underlying problem is

$$U(x,y;t) = \Big(1 + \exp((x+y-t)/(2a)\Big)^{-1}.$$

In our computation we choose $a = 0.005$. It is noted that the smaller a is, the more convection dominates, and the higher the concentration of mesh points required around the wave front. Figure 3.2 shows the movement of a mesh of 30×30 elements, with the monitor function $\sqrt{1+\|\nabla u\|_2^2}$.

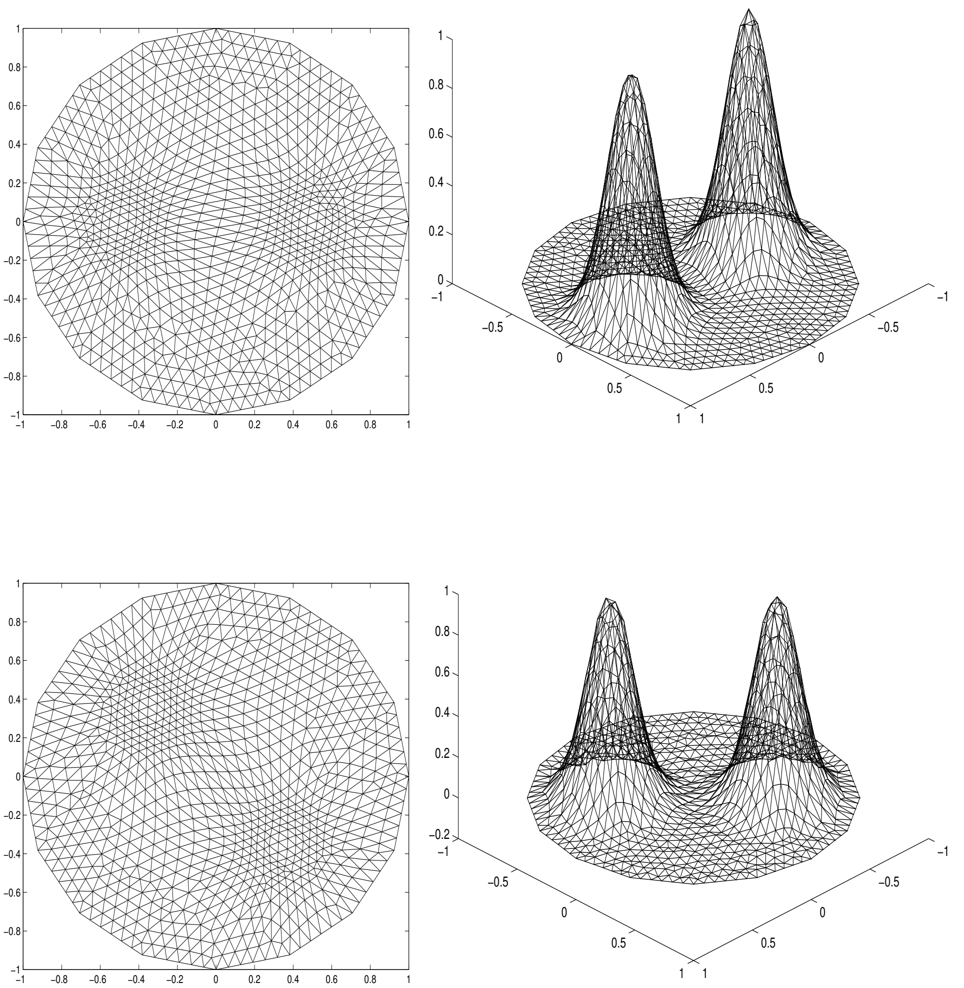

FIG. 3.1. *The adaptive meshes and the numerical solutions at* $t = 0$ *and* $t = 3\pi/8$ *for Example* 3.1.

4. Moving mesh methods for variational inequalities. Although the moving mesh methods have been applied extensively to partial differential equations with *large solution gradients*, there seems no existing work on implementing the moving mesh method to approximate variational inequalities either static or evolutional. Therefore, as the first step we will apply the moving mesh methods to the static obstacle problem before we attack more complicated case of evolutional problems. To this end, we will investigate the moving mesh adaptive finite element approximation for the following elliptic obstacle problem:

$$\begin{cases} -\mathrm{div}(A\nabla u) \geq f, \\ u \geq \phi, \\ (-\mathrm{div}(A\nabla u) - f)(u - \phi) = 0\,, \end{cases} \tag{4.1}$$

where A is a given matrix function, and f and ϕ are given functions. For ease of presentation we only consider a special obstacle problem where $\phi = 0$. The general case of nonzero obstacle function can be handled by using the transformation $u^* = u - \phi$.

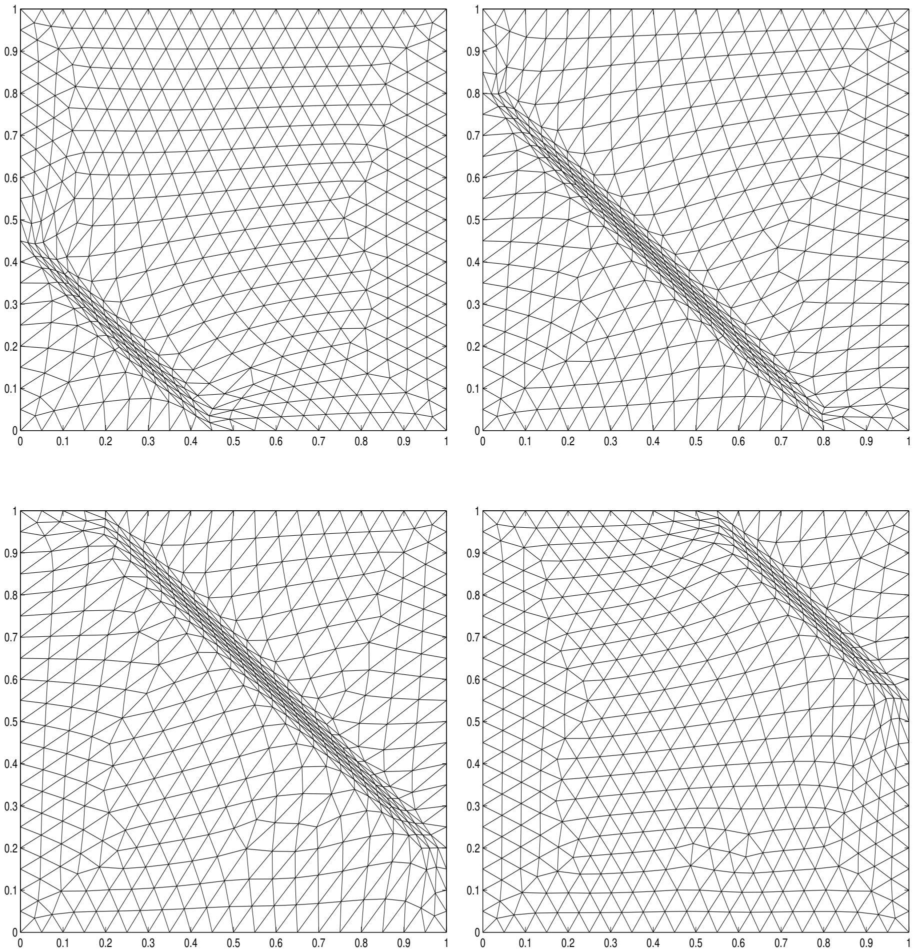

FIG. 3.2. *The adaptive meshes at* $t = 0.45$ *(top left),* 0.8 *(top right),* 1.2 *(bottom left) and* 1.65 *(bottom right) for Example* 3.2.

4.1. Finite element method for obstacle problem. Let Ω be a bounded open set in $\mathbf{R}^2$ with a Lipschitz boundary $\partial\Omega$. Let $f \in L^2(\Omega), g \in H^1(\Omega)$ and $V_g = \{v \in H^1(\Omega) : v - g \in H_0^1(\Omega)\}$. Let

$$a(u, v) = \int_\Omega (A\nabla u, \nabla v)_{\mathbf{R}^2}, \qquad \forall u, v \in H^1(\Omega),$$
$$(f, v) = \int_\Omega fv, \qquad \forall v \in H^1(\Omega).$$

The variational form of the obstacle problem (4.1) reads: For given f, g, find $u \in K_g$ such that

$$a(u, v - u) \geq (f, v - u) \qquad \forall v \in K_g. \tag{4.2}$$

Let K_g^h be a closed convex set in V_g^h. Then the finite element approximation of (4.2) reads: Find $u_h \in K_g^h$ such that

$$a(u_h, v_h - u_h) \geq (f, v_h - u_h) \qquad \forall v_h \in K_g^h. \tag{4.3}$$

One of the important issues requiring some attention is the monitor function M. There are several excellent papers in this direction, including Brackbill [3] and Cao et al. [7]. Most of the works in this direction are concerned with the solutions with large gradients, so in these works it is natural to use the solution gradients in the monitor function. However, it is quite complicated to choose a proper monitor in solving the obstacle problem (4.1). The standard monitor function involving gradients can not produce a satisfactory mesh, as to be seen later. One of the important conclusions from the existing theoretical analysis and numerical experiment is that in general the maximum approximation error is concentrated around free boundaries (i.e., the boundary of the coincidence set). Thus it is essential to use sharper *a posteriori* error estimators, which can identify the free boundaries, as error indicators in h-version adaptive schemes in order to refine the meshes correctly around the free boundary.

We first introduce two *a posteriori* error estimates for the finite element approximation of the obstacle problem with zero obstacle. The first estimate can be found in Chen and Nochetto [9].

THEOREM 4.1. *Let* $K_g = \{v \in V_g : v \geq 0\}$ *and* $K_g^h = \{v_h \in V_g^h : v_h \geq 0\}$. *Assume that* $g \in C^2(\bar{\Omega})$. *Let* u *and* u_h *be the solutions of* (4.2) *and* (4.3), *respectively. Then*

$$|u - u_h|_{1,\Omega}^2 \leq C(\eta_1^2 + \eta_2^2 + \eta_3^2), \tag{4.4}$$

where

$$\eta_1^2 = \sum_\tau h_\tau^2 \int_\tau (f + div(A\nabla u_h))^2,$$
$$\eta_2^2 = \sum_{l\cap\partial\Omega=\emptyset} h_l \int_l [(A\nabla u_h)\cdot \mathbf{n}]^2,$$
$$\eta_3^2 = \int_{\partial\Omega} |(A\nabla u_h)\cdot\mathbf{n}||g - \pi_h g|,$$

where h_l *is the size of an edge* l, *and* $[(A\nabla u_h)\cdot\mathbf{n}]$ *denotes the jump of* A*-normal derivative on an interior edge* l, *defined by*

$$[(A\nabla u_h)\cdot\mathbf{n}]_l = (A\nabla u_h|_{\tau_l^1} - A\nabla u_h|_{\tau_l^2})\cdot\mathbf{n}, \quad l = \bar{\tau}_l^1 \cap \bar{\tau}_l^2,$$

where $\mathbf{n}$ *is the outer normal vector of* τ_l^1.

The main disadvantage of the above estimate is that it only provides a upper bound for the error. It is clear that in general the estimate cannot be a lower bound since one could change the value of f in the coincidence set provided $f \leq 0$ without changing that of the solution.

By exploiting the fact that the obstacle problem may be equivalent to the following nonlinear equation: For given f, g, find $u \in K_g$ such that

$$a(u,v) + (f\chi_u, v) = (f,v) \qquad \forall v \in H_0^1(\Omega), \tag{4.5}$$

where

$$\chi_u(x) = \begin{cases} 1 & u(x) = 0 \\ 0 & u(x) > 0 \end{cases}$$

sharper *a posteriori* error estimates can be derived. The equivalence certainly holds if the solution $u \in H^2(\Omega)$ and the free boundary has Lebesgue measure zero, see [16].

THEOREM 4.2. *Assume that* A *is a constant matrix. Let* u, u_h *be the solutions of* (4.2) *and* (4.3) *with* $f \leq 0$ *on* Ω^-, *respectively. Let* $K_g^h = \{v \in V_g^h : v_h \geq 0\}$, *and* $g \in H^2(\Omega)$. *Assume that* $u \in W^{1,p}(\Omega)$ *with* $p > n$ *and that there is an* $\beta \in (0,2]$ *such that* $\|u - u_h\|_{0,\infty,\Omega} \leq Ch^\beta |ln(h)|$. *Then*

$$c(\hat{\eta}_1^2 + \hat{\eta}_2^2 + \eta_3^2 + \epsilon_1) \leq |u - u_h|_{1,\Omega}^2 \leq C(\hat{\eta}_1^2 + \hat{\eta}_2^2 + \eta_3^2 + \epsilon_2), \tag{4.6}$$

where

$$\hat{\eta}_1^2 = \sum_\tau h_\tau^2 \int_\tau (f(1 - \chi_{u_h}^h))^2, \quad \hat{\eta}_2^2 = \sum_{l\cap\partial\Omega=\emptyset} h_l \int_l [(A\nabla u_h)\cdot\mathbf{n}]^2,$$
$$\chi_{u_h}^h = \frac{h^\alpha}{h^\alpha + u_h}, \qquad 0 < \alpha < \beta,$$

ϵ_1 and ϵ_2 are some higher order terms when the solution is in $H^2(\Omega)$ and the free boundary is consist of finite pieces of smooth curves.

It is seen that the estimates in Theorem 4.2 are sharper than that given by Theorem 4.1, in the sense that the inequalities (4.6) provide an equivalent *a posteriori* error estimate.

One of the difficulties in applying the moving mesh techniques to the obstacle problem is that on the coincidence set the error estimators may be zero. This is a good feature for h-version refinement as it increases computational efficiency. However, in the moving mesh methods the monitor will be used as metric matrix, which should be positive defined, and no-zero at any point. In case the error estimator is zero at some point, we need to introduce a small positive number c within the estimator. The magnitude of c will decide the density of the nodes on the coincidence set. If c is too large, then there will have too many elements inside the set, which are not needed, as main approximation error is concentrated around the free boundary. On the other hand, if c is too small, there will have a large jump of density of elements over the free boundary as the monitor is relatively large just outside the coincidence set and near zero inside the set. This, as observed in our numerical experiments, leads to very singular meshes and higher approximation error around the free boundary. We overcome this difficulty by smoothing the monitors around free boundary. One of the strategies for smoothing the monitors is as follows: we first interpolate the monitor M from $L^2(\Omega)$ into $H_{1,h}(\Omega)$, i.e. from piecewise constant to piecewise linear, and we then project it back into $L^2(\Omega)$. An alternative approach for smoothing the monitor can be found in Li-Liu-Tang [13].

4.2. Numerical examples. In this subsection, we consider one numerical example by using three different types of monitor functions:

Monitor I: Gradient Monitor

$$M = \sqrt{1+\alpha|\nabla u|^2}$$

where α is a positive constant. This monitor has been used by many authors, see e.g. [3, 7, 12].

Monitor II: Monitor based on a non-equivalent *a posterior* error estimator

$$M|_{\text{on } \tau} = \sqrt{h_\tau^2 \int_\tau f^2 + \sum_{l:\text{ edge of } \tau} h_l \int_l [(A\Delta u_h)\cdot \mathbf{n}]^2}\,.$$

As indicated in Theorem 4.1 the above monitor function provides an upper bound for the finite element approximation to the variational inequality (4.2).

Monitor III: Monitor based on an equivalent *a posterior* error estimator

$$M|_{\text{on } \tau} = \sqrt{c + \eta_{1,\tau}^2 + \sum_{l:\text{ edge of } \tau} \eta_{2,l}^2}\,,$$

where c is a positive constant, $\eta_{1,\tau}$ is the integration in every element

$$\eta_{1,\tau}^2 = h_\tau^2 \int_\tau (f(1-\chi_{u_h}^h))^2,$$

and η_2 is the integration on every side

$$\eta_{2,l}^2 = h_l \int_l [(A\Delta u_h)\cdot \mathbf{n}]^2.$$

The reason for adding a positive constant c in the definition for Monitor III is that the monitor is to be used as metric matrix so it can not be zero at any point.

EXAMPLE 4.1. *We choose a square physical domain: $\Omega = [-1.5,1.5]\times[-1.5,1.5]$. The obstacle function ϕ is zero. The function f is chosen as*

$$f = \begin{cases} \frac{2}{r} - 4 & \text{if } r \geq 0.5 \\ 4^{0.95} - \left(\frac{2}{r}\right)^{0.95} & \text{if } r < 0.5. \end{cases}$$

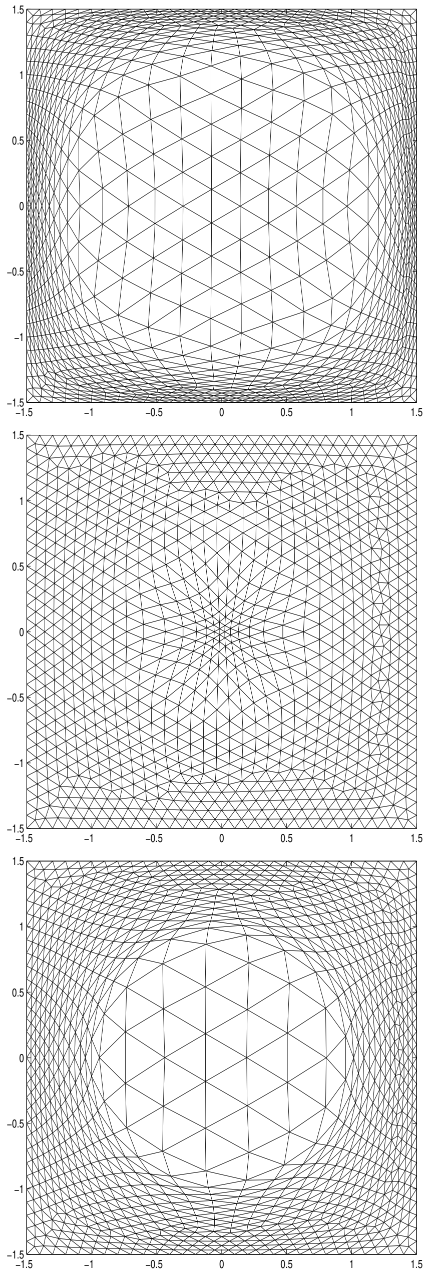

FIG. 4.1. *From top to bottom: the meshes for Example* 4.1 *obtained by by using Monitors I (gradient monitor), II (non-equilent monitor) and III (equivalent monitor), respectively.*

The exact solution is

$$u = \begin{cases} (r-1)^2 & r \geq 1 \\ 0 & r < 1 \end{cases}$$

where $r = \sqrt{x^2 + y^2}$. The free boundary is $u(x, y) = 0$ along the circle $r = 1$.

We compare the performance of the different monitor functions in Figure 4.1, with 30×30 elements used. It is seen from Figure 4.1 that the mesh refinement scheme based on the gradient monitor yields a wrong coincident set; the mesh refinement scheme based on the non-equivalent *a posterior* error estimates (Monitor II) refines the meshes almost uniformly everywhere in the computational domain and thus can not recognize the coincident set. On the other hand, the moving mesh scheme based on the equivalent *a posterior* error estimates (Monitor III) only refines the domain outside the coincidence set and therefore gives a clear description of the coincident set.

To demonstrate the convergence of the moving mesh method with Monitor III, we compute Example 4.1 with 20×20, 30×30 and 40×40 elements, and display the adaptive meshes and the solution errors in Figure 4.2. Indeed we observe from Figure 4.2 that the moving mesh methods with Monitor III have increased the accuracy significantly. The leading errors at the free boundary are eliminated with the moving mesh method.

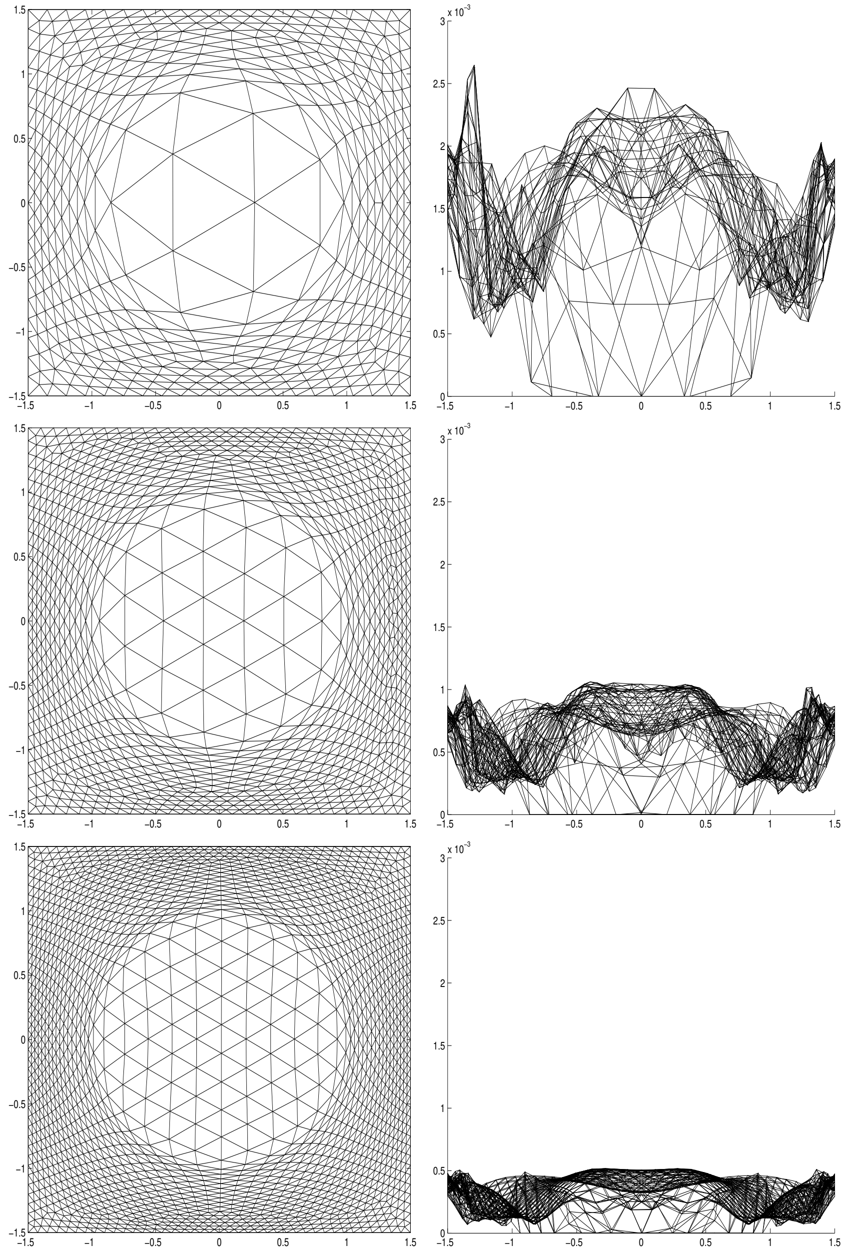

FIG. 4.2. *Adaptive meshes and solution errors with* 20×20, 30×30 *and* 40×40 *grid points (from top to bottom), respectively, for Example* 4.1.

5. Moving mesh methods for optical control problems. Optimal control has been a very important research area for the last fifty years. A large class of optical control problems can be symbolically written in the following form: (OCP)

$$
\begin{cases} \min\limits_{u\in U_{ad}} J(y,u) \\ \\ \text{subject to} \quad A(y,u)=0, \end{cases} \tag{5.1}
$$

where

- J is the objective functional,
- y is the state variable in a state space Y,
- u is the control variable in a control space U,
- U_{ad} is the admissible set of the control,
- and A is a suitable operator on $Y \times U_{ad}$.

Here the state equation $A(y,u) = 0$ can be an ordinary differential equation (ODE), a partial differential equation (PDE), a system of ODEs or PDEs, or a variational inequality, etc., and may incorporate with state constraints. An example of optimal control for parabolic type system is given below.

EXAMPLE 5.1. A linear quadratic control problem:

$$
\begin{cases} \min\limits_{u(t)\in K} \left\{ \dfrac{1}{2}\displaystyle\int_0^T \Big(\int_\Omega (y-z_d)^2 dx + \int_{\Omega_U} u^2 dx\Big) dt \right\} \\ \\ y_t(x,t) - \operatorname{div}(A\nabla y(x,t)) = (f+Bu)(x,t), \quad x\in\Omega, \quad t\in[0,T], \\ \\ y(x,t)=0, \quad x\in\partial\Omega, \quad t\in[0,T], \\ \\ y(x,0)=y_0(x), \quad x\in\Omega, \end{cases} \tag{5.2}
$$

where K is a closed convex set and B is a linear operator.

We are currently working on problems of this type using the moving mesh methods. Numerical approximation for optimal control is an emerging research area, which may involve optimization, grid adaptivity, error estimators etc. Optimal control problems very often have free or moving boundaries, and thus seem to be suitable in applying the moving mesh strategies. Moreover, it was found that efficiency of optimization algorithms used in solving the discretized problems play a very important role in improving performance of computations for optimal control. The detailed results on the optimal control computations will be reported elsewhere.

EXAMPLE 5.2. A convex optimal control problem

$$
\begin{cases} \min\limits_{u\in K} \Big\{ g(y) + h(u) \Big\}, \\ \\ -\operatorname{div}(A\nabla y) = f + Bu \quad \text{in } \Omega, \ y|_{\partial\Omega} = 0, \end{cases} \tag{5.3}
$$

where g and h are given convex functionals, K is a closed convex set and B is a continuous lenear operator.

In [14], both upper bounds and lower bounds are established with special attention to their equivalence. It is observed that numerical schemes based on harmonic mapping and the error estimators can resolve the problems with large solution gradients or free boundaries very efficiently.

Acknowledgments. This research was supported by Special Funds for Major State Basic Research Projects of China, British EPSRC (Grant number: GR/L67387), Hong Kong RGC (Grant number: 2033/99P) and HKBU FRG (Grant numbers: 98-99/II-14 and I-32).

REFERENCES

[1] Adjerid S., Flaherty, JE. A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations. *SIAM Journal on Numerical Analysis* 1986; **23**: 778-796.

[2] de Boor C. *Good approximation by splines with variable knots, II, in Conf. On the Numerical Solution of Differential Equations.* Lecture Notes in Math. 363. Springer-Verlag:Berlin, 1973; 12-20.

[3] Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics* 1982; **46**:342-368.

[4] Brackbill IU. An adaptive grid with direction control, *Journal of Computational Physics* 1993; **108**: 38-50.

[5] Ceniceros HD, Hou TY. An efficient dynamically adaptive mesh for potentially singular solutions. Submitted to JCP, 2000.

[6] Dvinsky AS. Adaptive grid generation from harmonic maps on Riemannian manifolds. *Journal of Computational Physics* 1991; **95**:450-476.

[7] Cao WM, Huang WZ, Russell RD. A study of monitor functions for two dimensional adaptive mesh generation, *SIAM Journal on Scientific Computing* 1999; **20**:1978-1994.

[8] Cao WM, Huang WZ, Russell RD. An r-adaptive finite element method based upon moving mesh PDEs. *Journal of Computational Physics* 1999; **149**:221-244.

[9] Chen ZM, Nochetto RH. Residual type a posteriori error estimates for elliptic obstacle problems. *Numer. Math.* 2000; **84**: 527–548.

[10] Hamilton R. *Harmonic maps of manifolds with boundary.* Lecture Notes in Math. 471. Springer-Verlag:New-York, 1975.

[11] Huang WZ, Russell RD. *Moving mesh strategy based on a gradient flow equation for two-dimensional problems. SIAM Journal on Scientific Computing* 1999; **20**: 998-1015.

[12] Li R, Tang T, Zhang PW. Moving mesh methods in multiple dimensions based on harmonic maps. Submitted to JCP (under revision), 2000.

[13] Li R, Liu WB, Tang T. Moving Mesh Finite Element Approximations for Variational Inequality I: Static Obstacle Problem, Submitted, 2000. `http://www.math.hkbu.edu.hk/~ttang/`.

[14] Li R, Liu WB, H.-P. Ma Tang T. Mesh adaptivity for optimal control problems, in preperation.

[15] Liao G, Liu F, de la Pena C, Peng D, Osher S. Level-set-based deformation methods for adaptive grids. *Journal of Computational Physics* 2000; **159**: 103-122.

[16] Liu WB, Yan N. A posteriori error estimates for a class of variational inequalities. To appear.

[17] Miller K, Miller RN. Moving finite elements I. *SIAM Journal on Numerical Analysis* 1981; **18**: 1019-1032.

[18] Liu WB, Tang T. Spectral methods for singular perturbation problems. *Proceedings of Symposia in Applied Mathematics, AMS* 1994; **48**:323-326.

[19] Petzold LR. *Observations on an adaptive moving grid method for one-dimensional systems of partial differential equations. Applied Numerical Mathematics* 1987; **3**: 347-360.

[20] Ren W, Wang XP. An iterative grid redistribution method for singular problems in multiple dimensions. *Journal of Computational Physics*, 2000. To appear.

[21] Schoen R, Yau ST. On univalent harmonic maps between surfaces. *Inventiones Mathematicae* 1978. **44**:265.

[22] Tang T, Trummer MR. Boundary layer resolving pseudospectral methods for singular perturbation problems. *SIAM Journal on Scientific Computing* 1996; **17**:430-438.

[23] Thompson JF, Warsi ZUA, Mastin CW. *Numerical Grid Generation.* North Holland, New York, 1985.

[24] Winslow A. Numerical solution of the quasi-linear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics* 1967; **1**:149-172.