A Comparative Numerical Study of Meshing Functionals for Variational Mesh Adaptation

Weizhang Huang[1], Lennard Kamenski[2,*], Robert D. Russell[3]

[1] *Department of Mathematics, University of Kansas, Lawrence, KS 66045, U.S.A.*

[2] *Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany*

[3] *Department of Mathematics, Simon Fraser University, Burnaby,*
BC V5A 1S6, Canada

Abstract. We present a comparative numerical study for three functionals used for variational mesh adaptation. One of them is a generalization of Winslow's variable diffusion functional while the others are based on equidistribution and alignment. These functionals are known to have nice theoretical properties and work well for most mesh adaptation problems either as a stand-alone variational method or combined within the moving mesh framework. Their performance is investigated numerically in terms of equidistribution and alignment mesh quality measures. Numerical results in 2D and 3D are presented.

AMS subject classifications: 65N50, 65K10

Key words: Variational mesh adaptation, Mesh adaptation, Moving mesh, Equidistribution, Alignment, Mesh quality measures.

1   Introduction

Variational mesh adaptation is an important type of mesh adaptation method and has received considerable attention from scientists and engineers; e.g., see the books [15, 19, 23, 24] and references therein. It also serves as the base of a number of commonly used adaptive moving mesh methods (e.g., see [5, 12, 14, 22]). In the variational approach, an adaptive mesh is generated as the image of a reference mesh under a coordinate transformation and such a coordinate transformation is determined as a minimizer of a certain meshing functional. A number of meshing functionals have been developed in the past (cf. the above mentioned books). For example, Winslow [25] proposed an equipotential method based on variable diffusion. Brackbill and Saltzman [3] developed a method by

combining mesh concentration, smoothness, and orthogonality. Dvinsky [6] used the energy of harmonic mappings as his meshing functional, while Knupp [20] and Knupp and Robidoux [21] developed functionals based on the idea of conditioning the Jacobian matrix of the coordinate transformation. More recently, Huang [7] and Huang and Russell [15] proposed functionals based on the so-called equidistribution and alignment conditions.

With variational mesh adaptation, the mesh concentration is typically controlled through a scalar or a matrix-valued function, often referred to as the metric tensor or monitor function and defined based on some error estimates and/or physical considerations. While most of the meshing functionals have been developed with physical or geometric intuitions and have various levels of success in the adaptive numerical solution of partial differential equations (PDEs) and other applications, there is only a limited understanding on how the metric tensor affects the behavior of the mesh. An attempt to alleviate this lack of understanding was made by Cao *et al.* [4] for a generalization of Winslow's variable diffusion functional. They showed that a significant change in an eigenvalue of the metric tensor along the corresponding eigendirection (first increasing and then decreasing, or vice versa) will result in adaptation of coordinate lines along this direction, although this adaptation competes with far more complicated effects, including those from changes in eigenvectors and other eigenvalues and the effects of the shapes of the physical and computational domains and the mesh point distribution on the boundaries. In [7,15], two functionals have been developed based directly on the equidistribution and alignment conditions. These two conditions provide a complete characterization of the mesh elements through the metric tensor [7]. Minimizing the functionals leads to meshes which tend to satisfy the conditions in an integral sense. Nevertheless, this characterization is only qualitative, and how closely the mesh satisfies the conditions depends on the boundary correspondence between the computational and physical domains and the mesh point distribution on the boundaries. Thus, numerical studies, especially comparative ones, are useful, and often necessary, in understanding how the mesh adaptation for those meshing functionals is controlled precisely by the metric tensor. There do exist a few comparative numerical studies for meshing functionals. For example, a gallery of (adaptive and non-adaptive) meshes is given in [19] for a number of meshing functionals. Some comparative meshes are given in [15] for the harmonic mapping functional [6] and the subsequent functional based on equidistribution and alignment [7].

The main objective of this work is to present a comparative study for three of the most appealing meshing functionals, a generalization of Winslow's variable diffusion functional (cf. (3.2)) and two functionals based on equidistribution and alignment (cf. (3.7) and (3.9)). They are selected because (3.2) and (3.7) have been known to work well for many problems (e.g., see [1, 2, 7, 13, 14, 22]) while (3.9) is similar to (3.7) and has some very nice theoretical properties (cf. §3.2). Another motivation is to present some three dimensional numerical results for those functionals. Critical for our study is to perform the substantial computations using the improved implementation of the variational methods introduced in [11]. In a sharp contrast to the situation in two dimensions, very little work

has been done with variational mesh adaptation and adaptive moving mesh methods in three dimensions (e.g., see [15,22]). It is particularly interesting to see how the functionals perform in this case.

An outline of the paper is given as follows. We describe the basic ideas of the variational mesh adaptation and its direct discretization (that is, first to discretize and then optimize) in §2. In §3 we introduce the three functionals to be studied for the numerical comparison, a generalization of Winslow's variable diffusion functional and two functionals based on equidistribution and alignment. Numerical results and example adaptive meshes are given in §4, followed by conclusions in §5.

2   Variational mesh adaptation

In variational mesh adaptation, an adaptive mesh is generated as the image of a reference mesh under a coordinate transformation. Denote the physical domain by $\Omega \subset \mathbb{R}^d$ $(d \geq 1)$, and assume that we are given a computational domain $\Omega_c \subset \mathbb{R}^d$ and a quasi-uniform mesh $\hat{\mathcal{T}}_{h_c}$ thereon (in this work we consider only simplicial meshes). In many situations we can choose Ω_c to be the unit square/cube or simply Ω. Denote the coordinate transformation by $x = x(\xi) \colon \Omega_c \to \Omega$ and its inverse by $\xi = \xi(x) \colon \Omega \to \Omega_c$. Such a coordinate transformation (more precisely, its inverse) is determined as a minimizer of a meshing functional. Most of the existing meshing functionals can be cast in a general form as

$$I[\xi] = \int_\Omega G(\mathbb{J}, \det(\mathbb{J}), \mathbb{M}, x)\, dx, \tag{2.1}$$

where G is a smooth function, $\mathbb{J}$ is the Jacobian matrix of $\xi = \xi(x)$, $\det(\mathbb{J})$ denotes the determinant of $\mathbb{J}$, and $\mathbb{M} = \mathbb{M}(x)$ is the metric tensor supplied by the user to control mesh concentration. We assume that $\mathbb{M} = \mathbb{M}(x)$ is a symmetric and uniformly positive definite d-by-d matrix-valued function on Ω. Notice that (2.1) is formulated in terms of the inverse coordinate transformation. One reason for this is that this form is less likely to produce singular meshes [6]. Another reason is that $\mathbb{M}$ is a function of x and thus finding the functional derivative of $I[\xi]$ will not directly involve the derivatives of $\mathbb{M}$. This is convenient since in practice $\mathbb{M}$ is known only at the vertices of a mesh and its derivatives are not cheap to find. The main disadvantage of the formulation in this form is that $\xi = \xi(x)$ (or its numerical approximation) does not give the physical mesh directly. This is remedied either by interchanging the roles of the independent and dependent variables in the Euler-Lagrange equation of $I[\xi]$ (e.g., see [15]) or, in a recently developed implementation (see below), computing the new physical mesh from a computational one using linear interpolation.

A minimizer of (2.1) can be found numerically in the MMPDE (moving mesh PDE) framework. A conventional implementation [15] is to find the functional derivative of (2.1) and then define the MMPDE as the gradient flow equation of the functional. Having been transformed by interchanging the roles of the dependent and independent variables, the MMPDE can be discretized on $\hat{\mathcal{T}}_{h_c}$ and a system of equations for the nodal

velocities is obtained. Finally, the new mesh is obtained by integrating the mesh equation over a time step.

A much simpler implementation was proposed recently by Huang and Kamenski [11]. Instead of utilizing the MMPDE directly, the new implementation first discretizes the functional on the current mesh $\mathcal{T}_h$ and then, following the idea of the MMPDE, defines the mesh equation as the gradient equation of the discretized functional (with respect to the computational coordinates of the vertices). To be specific, denote by x_i, $\hat{\xi}_i$, and ξ_i, $i=1,...,N_v$ the coordinates of the vertices of the current physical mesh ($\mathcal{T}_h$), the reference mesh ($\hat{\mathcal{T}}_{hc}$), and the computational mesh ($\mathcal{T}_{hc}$), respectively. We assume that these meshes have the same numbers of the elements and vertices and the same connectivity. For any element $K \in \mathcal{T}_h$ (with vertices x_i^K, $i=0,...,d$), the corresponding element in $\mathcal{T}_{hc}$ is denoted by K_c (with vertices ξ_i^K, $i=0,...,d$). The edge matrices for K and K_c are defined as

$$E_K = [x_1^K - x_0^K,...,x_d^K - x_0^K], \quad E_{K_c} = [\xi_1^K - \xi_0^K,...,\xi_d^K - \xi_0^K].$$

Let ω_i be the element patch associated with vertex x_i (i.e., the collection of the elements containing x_i as a vertex). Then, the equation for the nodal velocities reads as

$$\begin{cases} \frac{d\xi_i}{dt} = \frac{P_i}{\tau} \sum_{K \in \omega_i} |K| v_{i_K}^K, & i=1,...,N_v, \quad t_n < t \le t_{n+1}, \\ \xi_i(t_n) = \hat{\xi}_i, & i=1,...,N_v, \end{cases} \quad (2.2)$$

where $|K|$ is the volume of K, $v_{i_K}^K$ is the local mesh velocity associated with vertex x_i in K, i_K denotes the local index of x_i in K, $\tau > 0$ is a constant parameter used to adjust the time scale of mesh movement, and $P=(P_1,...,P_{N_v})$ is a positive function used to make the mesh equation to have desired invariance properties. Although the parameter τ can be absorbed in P, using two parameters has the advantage that the role of each parameter is clear: τ for the time scale of mesh movement while P for the invariance properties. Ideally, τ should be chosen such that the mesh movement has the same scale as the physical equation. Unfortunately, there is no theoretical analysis for this yet and trial and error is still the most practical way to choose τ. Numerical experience shows that a value in the range $[0.01,0.1]$ seems to work well for most problems [15]. In our computation, we choose P such that the equation is invariant under the scaling transformation $\mathbb{M} \to c\mathbb{M}$ for all non-zero constants c (cf. §3): in variational mesh adaptation it is the relative distribution of $\mathbb{M}$ over the physical domain (instead of its absolute distribution) that determines the variation of the mesh density and therefore it is essential for the moving mesh equation to be invariant under the scaling transformation of $\mathbb{M}$.

The local velocities are given by

$$\begin{bmatrix} (v_1^K)^T \\ \vdots \\ (v_d^K)^T \end{bmatrix} = -E_K^{-1} \frac{\partial G}{\partial \mathbb{J}} - \frac{\partial G}{\partial \det(\mathbb{J})} \frac{\det(E_{K_c})}{\det(E_K)} E_{K_c}^{-1}, \quad v_0^K = -\sum_{j=1}^d v_j^K, \quad (2.3)$$

where the derivatives of G with respect to $\mathbb{J}$ and $\det(\mathbb{J})$ (see (3.3), (3.8) and (3.10) below) are evaluated as

$$\frac{\partial G}{\partial \mathbb{J}} = \frac{\partial G}{\partial \mathbb{J}} \left(E_{K_c} E_K^{-1}, \frac{\det(E_{K_c})}{\det(E_K)}, \mathbb{M}(x_K), x_K \right),$$

$$\frac{\partial G}{\partial \det(\mathbb{J})} = \frac{\partial G}{\partial \det(\mathbb{J})} \left(E_{K_c} E_K^{-1}, \frac{\det(E_{K_c})}{\det(E_K)}, \mathbb{M}(x_K), x_K \right).$$

The second equation in (2.3) is an inherent property resulting directly from the differentiation of the discretized meshing functional; it states that the centroid of K stays fixed if only the contribution from K is taken into account.

The above mesh equation should be modified properly for boundary vertices. For example, if ξ_i is a fixed boundary vertex, we replace the corresponding equation by

$$\frac{\partial \xi_i}{\partial t} = 0.$$

When ξ_i is allowed to move on a boundary curve/surface represented by

$$\phi(\xi) = 0,$$

then the mesh velocity $\frac{\partial \xi_i}{\partial t}$ needs to be modified such that its normal component along the curve or surface is zero, i.e.,

$$\nabla \phi(\xi_i) \cdot \frac{\partial \xi_i}{\partial t} = 0.$$

Mesh equation (2.2) defines the movement of the nodes of the computational mesh $\mathcal{T}_{hc}$ starting from the reference mesh $\hat{\mathcal{T}}_{hc}$ at t_n. The equation can be integrated in time to obtain the computational mesh at t_{n+1}. For notational simplicity, we denote the computational mesh at t_{n+1} by $\mathcal{T}_{hc}$ as well. Notice that the physical mesh $\mathcal{T}_h$ is fixed during the time integration from t_n to t_{n+1}. Meshes $\mathcal{T}_{hc}$ and $\mathcal{T}_h$ define a correspondence

$$\mathcal{T}_h = \Psi(\mathcal{T}_{hc}).$$

The new physical mesh is computed by means of linear interpolation as

$$\tilde{\mathcal{T}}_h = \Psi(\hat{\mathcal{T}}_{hc}),$$

where $\hat{\mathcal{T}}_{hc}$ is the reference mesh on Ω_c. The computational mesh plays the role of an intermediate variable.

Recall that the mesh concentration in variational mesh adaptation is controlled through the metric tensor $\mathbb{M} = \mathbb{M}(x)$. Such a metric tensor can be defined based on physical or geometric considerations or some error estimates. For example, for the L^2 norm of the error of piecewise linear interpolation on simplicial meshes, the optimal metric tensor [9, 16] (also see [15, (5.192)]) reads as

$$\mathbb{M} = \det(\alpha I + |H(u)|)^{-\frac{1}{d+4}} [\alpha I + |H(u)|], \tag{2.4}$$

where $H(u)$ is the Hessian of function u, $|H(u)|$ is the eigen-decomposition of $H(u)$ with the eigenvalues being replaced by their absolute values, and the regularization parameter $\alpha > 0$ is chosen such that

$$\int_\Omega \det(\mathbb{M})^{\frac{1}{2}} d\boldsymbol{x} \equiv \int_\Omega \det(\alpha I + |H(u)|)^{\frac{2}{d+4}} d\boldsymbol{x} = 2\int_\Omega \det(|H(u)|)^{\frac{2}{d+4}} d\boldsymbol{x}.$$

In practical computation, u is typically unknown, and only the approximations to its values at the vertices are available. For this reason (and even in situations where an analytical expression for u is available), the Hessian in (2.1) is replaced by an approximation obtained by a Hessian recovery technique from the nodal values of u or the approximations of the nodal values of u. A number of such techniques are known to produce nonconvergent recovered Hessians from a linear finite element approximation (e.g., see Kamenski [17]). Nevertheless, it is shown by Kamenski and Huang [18] that a linear finite element solution of an elliptic BVP converges at a second order rate as the mesh is refined if the recovered Hessian used to generate the adaptive mesh satisfies a closeness assumption. Numerical experiment shows that this closeness assumption is satisfied by the approximate Hessian obtained with commonly used Hessian recovery methods. We use a Hessian recovery method based on a least squares fit: a quadratic polynomial is constructed locally for each vertex via least squares fitting to neighboring nodal function values and an approximate Hessian at the vertex is then obtained by differentiating the polynomial.

3   Meshing functionals

Here we introduce the three meshing functionals used in the numerical study. A generalization of Winslow's variable diffusion functional and the two functionals based on equidistribution and alignment are selected because they are reasonably simple, have nice theoretical properties, and are known to work well for many problems.

3.1   Winslow's functional based on variable diffusion

The first functional is the variable diffusion proposed by Winslow [25]. It uses the system of elliptic PDEs

$$-\nabla \cdot (w\nabla \xi_i) = 0, \quad i=1,\dots,d,$$

for generating adaptive meshes, where $w = w(\boldsymbol{x}) > 0$ is the weight function. This system mimics a (steady-state) diffusion process with a heterogeneous diffusion coefficient $w(\boldsymbol{x})$. It is the Euler-Lagrange equation of the functional

$$I[\boldsymbol{\xi}] = \frac{1}{2}\int_\Omega \sum_{i=1}^{d} w(\boldsymbol{x})|\nabla \xi_i|^2 d\boldsymbol{x} = \frac{1}{2}\int_\Omega w(\boldsymbol{x})\operatorname{tr}(\mathbb{J}\mathbb{J}^T)d\boldsymbol{x}, \tag{3.1}$$

where $\mathrm{tr}(\cdot)$ is the trace of a matrix. A generalization of this functional to allow a diffusion tensor reads as

$$I[\boldsymbol{\xi}] = \frac{1}{2} \int_\Omega \mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\,d\boldsymbol{x}. \tag{3.2}$$

This functional has been used by a number of researchers, e.g., see Huang and Russell [13, 14], Li *et al.* [22], and Beckett *et al.* [2]. It is coercive and convex [15, Example 6.2.1]. Thus, under a suitable boundary condition (such as the Dirichlet boundary condition with $\partial\Omega_c$ being mapped onto $\partial\Omega$), the functional (3.2) has a unique minimizer.

For this functional, we can find the derivatives of G with respect to $\mathbb{J}$ and $\det(\mathbb{J})$ needed in (2.3),

$$\begin{cases} G = \frac{1}{2}\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T), \\ \frac{\partial G}{\partial \mathbb{J}} = \mathbb{M}^{-1}\mathbb{J}^T, \\ \frac{\partial G}{\partial \det(\mathbb{J})} = 0. \end{cases} \tag{3.3}$$

The interested reader is referred to [11] for the derivation.

The balancing function in (2.2) is chosen as $P = \det(\mathbb{M})^{\frac{1}{d}}$. With this choice, (2.2) is invariant under the scaling transformation $\mathbb{M} \to c\mathbb{M}$.

3.2   Functionals based on equidistribution and alignment

The other functionals are based on the equidistribution and alignment conditions. These conditions provide a full mathematical characterization of a non-uniform mesh. Indeed, any non-uniform mesh can be viewed as a uniform one in the metric specified by a tensor. Moreover, a mesh is uniform in the metric specified by the metric tensor $\mathbb{M} = \mathbb{M}(\boldsymbol{x})$ if and only if it satisfies the equidistribution and alignment conditions associated with $\mathbb{M}$ [10, 15]. In the continuous form, they are

$$\text{equidistribution:} \quad \det(\mathbb{J})^{-1}\det(\mathbb{M})^{\frac{1}{2}} = \frac{\sigma}{|\Omega_c|}, \quad \forall \boldsymbol{x} \in \Omega \tag{3.4}$$

$$\text{alignment:} \quad \frac{1}{d}\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) = \det(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)^{\frac{1}{d}}, \quad \forall \boldsymbol{x} \in \Omega, \tag{3.5}$$

where

$$\sigma = \int_\Omega \det(\mathbb{M})^{\frac{1}{2}}\,d\boldsymbol{x}. \tag{3.6}$$

These conditions require the mesh elements to have the same size (equidistribution) and be equilateral (alignment) in the metric $\mathbb{M}$, respectively. The alignment condition also implies that the elements are aligned with $\mathbb{M}$ in the sense that the principal directions of the circumscribed ellipsoid of each element coincide with the eigen-directions of $\mathbb{M}$ while the lengths of the principal axes of the ellipsoid are reciprocally proportional to the square roots of the eigenvalues of $\mathbb{M}$.

The first functional based on equidistribution and alignment, proposed in [7], is

$$I[\boldsymbol{\xi}] = \theta \int_\Omega \sqrt{\det(\mathbb{M})} \left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) \right)^{\frac{dp}{2}} d\boldsymbol{x} + (1-2\theta)d^{\frac{dp}{2}} \int_\Omega \sqrt{\det(\mathbb{M})} \left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M})}} \right)^p d\boldsymbol{x},$$

(3.7)

where $\theta \in (0,1)$ and $p > 0$ are dimensionless parameters. Loosely speaking, the first and second terms correspond to the equidistribution and alignment conditions, respectively. The terms are dimensionally homogeneous and the balance between them is controlled by the dimensionless parameter θ. For $0 < \theta \le \frac{1}{2}$, $dp \ge 2$, and $p \ge 1$, the functional is coercive and polyconvex and has a minimizer [15, Example 6.2.2]. Moreover, for $\theta = \frac{1}{2}$ and $dp = 2$ it reduces to

$$I[\boldsymbol{\xi}] = \frac{1}{2} \int_\Omega \sqrt{\det(\mathbb{M})} \, \mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) d\boldsymbol{x},$$

which is exactly the energy functional of a harmonic mapping from Ω to Ω_c (cf. [6]).

For the functional (3.7), we have

$$\begin{cases} G = \theta \sqrt{\det(\mathbb{M})} \left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) \right)^{\frac{dp}{2}} + (1-2\theta)d^{\frac{dp}{2}} \sqrt{\det(\mathbb{M})} \left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M})}} \right)^p, \\ \frac{\partial G}{\partial \mathbb{J}} = dp\theta \sqrt{\det(\mathbb{M})} \left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) \right)^{\frac{dp}{2}-1} \mathbb{M}^{-1}\mathbb{J}^T, \\ \frac{\partial G}{\partial \det(\mathbb{J})} = p(1-2\theta)d^{\frac{dp}{2}} \det(\mathbb{M})^{\frac{1-p}{2}} \det(\mathbb{J})^{p-1}. \end{cases}$$

(3.8)

In the computation, we use $(p,\theta) = (2,\frac{1}{3})$. $p = 2$ is the smallest integer to satisfy $dp \ge 2$ for $d = 1, 2$, and 3. The choice of $\theta = 1/3$ is in the range $(0,1/2]$ for the functional to be polyconvex while giving a bigger weight to the equidistribution condition. This set of the values works well for all tested problems. The balancing function in (2.2) is chosen to be $P = \det(\mathbb{M})^{\frac{p-1}{2}}$, so that (2.2) is invariant under the scaling transformation $\mathbb{M} \to c\,\mathbb{M}$.

The second functional based on equidistribution and alignment reads as

$$I[\boldsymbol{\xi}] = \theta_1 \int_\Omega \sqrt{\det(\mathbb{M})} \left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) \right)^{\frac{dp}{2}} d\boldsymbol{x}$$

$$+ \theta_2 d^{\frac{dp(d-2)}{2(d-1)}} \int_\Omega \det(\mathbb{M})^{\frac{1}{2}(1-\frac{dp}{d-1})} \det(\mathbb{J})^{\frac{dp}{d-1}} \left(\mathrm{tr}(\mathbb{J}^{-T}\mathbb{M}\mathbb{J}^{-1}) \right)^{\frac{dp}{2(d-1)}} d\boldsymbol{x}$$

$$+ (\theta_3 - \theta_1 - \theta_2) d^{\frac{dp}{2}} \int_\Omega \sqrt{\det(\mathbb{M})} \left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M})}} \right)^p d\boldsymbol{x}$$

$$+ \frac{\theta_4}{\sigma^{p+\nu}} \int_\Omega \sqrt{\det(\mathbb{M})} \left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M})}} \right)^{-\nu} d\boldsymbol{x},$$

(3.9)

where $p > 1$, $\nu > 0$, and $\theta_i > 0$ ($i = 1,\ldots,4$) are parameters. The first three terms in (3.9) are dimensionally homogeneous in $\mathbb{M}$ and $\mathbb{J}$ while the last term has the same dimension in $\mathbb{M}$ as the other terms. This functional was proposed in [15, (6.120)] to avoid singularity

of the coordinate transformation. Indeed, if $\theta_3 - \theta_1 - \theta_2 > 0$, then the functional is coercive and polyconvex and has a minimizer satisfying $\det(\mathbb{J}) > 0$ in Ω [15, Example 6.2.3].

In the computation, we choose $\theta_1 = \theta_2 = \frac{1}{3}$, $\theta_3 = 1$, $\theta_4 = 0.1$, $p = 2$, $\nu = 1$, and the balancing function $P = \det(\mathbb{M})^{\frac{p-1}{2}}$. These choices are based on the functional (3.7) and the desire to keep the fourth term relatively small.

For this functional, we have

$$
\left\{
\begin{aligned}
G &= \theta_1 \sqrt{\det(\mathbb{M})} \left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) \right)^{\frac{dp}{2}} \\
&\quad + \theta_2 d^{\frac{dp(d-2)}{2(d-1)}} \det(\mathbb{M})^{\frac{1}{2}\left(1 - \frac{dp}{d-1}\right)} \det(\mathbb{J})^{\frac{dp}{d-1}} \left(\operatorname{tr}(\mathbb{J}^{-T}\mathbb{M}\mathbb{J}^{-1}) \right)^{\frac{dp}{2(d-1)}} \\
&\quad + (\theta_3 - \theta_1 - \theta_2) d^{\frac{dp}{2}} \left(\sqrt{\det(\mathbb{M})} \right)^{1-p} \det(\mathbb{J})^p \\
&\quad + \frac{\theta_4}{\sigma^{p+\nu}} \left(\sqrt{\det(\mathbb{M})} \right)^{1+\nu} \det(\mathbb{J})^{-\nu}, \\
\frac{\partial G}{\partial \mathbb{J}} &= \theta_1 dp \sqrt{\det(\mathbb{M})} \left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T) \right)^{\frac{dp}{2}-1} \mathbb{M}^{-1}\mathbb{J}^T \\
&\quad - \frac{\theta_2 dp}{d-1} d^{\frac{dp(d-2)}{2(d-1)}} \det(\mathbb{M})^{\frac{1}{2}\left(1 - \frac{dp}{d-1}\right)} \det(\mathbb{J})^{\frac{dp}{d-1}} \left(\operatorname{tr}(\mathbb{J}^{-T}\mathbb{M}\mathbb{J}^{-1}) \right)^{\frac{dp}{2(d-1)}-1} \mathbb{J}^{-1}\mathbb{J}^{-T}\mathbb{M}\mathbb{J}^{-1}, \\
\frac{\partial G}{\partial \det(\mathbb{J})} &= \frac{\theta_2 dp}{d-1} d^{\frac{dp(d-2)}{2(d-1)}} \det(\mathbb{M})^{\frac{1}{2}\left(1 - \frac{dp}{d-1}\right)} \det(\mathbb{J})^{\frac{dp}{d-1}-1} \left(\operatorname{tr}(\mathbb{J}^{-T}\mathbb{M}\mathbb{J}^{-1}) \right)^{\frac{dp}{2(d-1)}} \\
&\quad + (\theta_3 - \theta_1 - \theta_2) p d^{\frac{dp}{2}} \left(\sqrt{\det(\mathbb{M})} \right)^{1-p} \det(\mathbb{J})^{p-1} \\
&\quad - \frac{\theta_4 \nu}{\sigma^{p+\nu}} \left(\sqrt{\det(\mathbb{M})} \right)^{1+\nu} \det(\mathbb{J})^{-\nu-1}.
\end{aligned}
\right.
\tag{3.10}
$$

4 Numerical experiments

In the following we consider a number of examples in two and three dimensions. For a given function we consider $\mathbb{M}$ defined in (2.4) which is optimal for minimizing the L^2 norm of the linear interpolation error of this function and compare meshes obtained from using the functionals of Winslow (3.2) (W), Huang (3.7) (H), and Huang and Russell (3.9) (HR).

To assess the quality of the generated meshes, we compare the L^2 norm of the linear interpolation error and the equidistribution and alignment mesh quality measures, which describe how far the mesh is from being uniform in the metric defined by $\mathbb{M}$. The element-wise quality measures are based on (3.4) and (3.5) and defined as

$$
Q_{eq,K} = \frac{\det(\mathbb{J}_K)^{-1} \det(\mathbb{M}_K)^{\frac{1}{2}}}{\sigma / |\Omega_c|}, \qquad Q_{ali,K} = \frac{\operatorname{tr}(\mathbb{J}_K \mathbb{M}_K^{-1} \mathbb{J}_K^T)}{d \det(\mathbb{J}_K \mathbb{M}_K^{-1} \mathbb{J}_K^T)^{\frac{1}{d}}},
\tag{4.1}
$$

while for the overall mesh quality measures we take their root-mean-squared values,

$$
Q_{eq} = \sqrt{\frac{1}{N} \sum_{K \in \mathcal{T}_h} Q_{eq,K}^2}, \qquad Q_{ali} = \sqrt{\frac{1}{N} \sum_{K \in \mathcal{T}_h} Q_{ali,K}^2}.
\tag{4.2}
$$

If the mesh is uniform with respect to $\mathbb{M}$, then $Q_{eq} = Q_{ali} = 1$; if the mesh is far from being uniform with respect to $\mathbb{M}$, then Q_{eq} and Q_{ali} will become large. In other words, these quality measures describe how well the volume (measured by Q_{eq}) and the shape and orientation (measured by Q_{ali}) of mesh elements correspond to the desired size and shape prescribed by $\mathbb{M}$ (see [8] for more details on the mesh quality measures).

4.1 Two dimensions

First, we consider two dimensional meshes constructed for the unit square $\Omega = (0,1) \times (0,1)$ and the test functions

Example 4.1.
$$u = \tanh(-30[y - 0.5 - 0.25\sin(2\pi x)]).$$

Example 4.2.
$$u = \tanh(25y) - \tanh(25(x - y - 0.5)).$$

Example meshes, close-ups, as well as the mesh quality measures and the L^2 interpolation error are given in Figs. 1 and 2.

For these examples, all three functionals provide good size and shape adaptation. A closer look at the mesh quality measures shows that, although all three functionals provide comparable meshes which are reasonably close to the prescribed metric tensor, meshes constructed using H and HR functionals have better correspondence to the prescribed metric tensor. In both two-dimensional examples, H and HR functionals provide very similar grids which are closer to the prescribed size and shape (that is, smaller values of Q_{eq} and Q_{ali}). This is also reflected in the error of the linear interpolation: HR functional (3.9) provides the smallest error, followed by H functional (3.7) and then W functional (3.2). It seems that W functional is a bit too aggressive in moving nodes towards the neighborhood of areas of interest, providing a higher density of the nodes along the anisotropic features of the given function while coarsening out the mesh nearby, leading to a steeper element size gradation. Interestingly, for both examples the convergence of the linear interpolation error for W functional (Figs. 1f and 2f) slows down near $N = 10^4$ and returns to the order $\mathcal{O}(N^{-1})$ as the mesh is refined (N is the number of mesh elements). It is unclear to us what causes this for W functional.

For Example 4.1 we also compute adaptive meshes for the metric tensor which is optimal for the H^1 semi norm error (see, e.g., [10] for details on the metric tensor). The results shown in Fig. 3 are essentially the same as in Fig. 1: HR functional (3.9) provides the smallest error, followed by H functional (3.7) and then W functional (3.2). For this metric tensor, the L^2 error for the H and HR functionals (Fig. 3f) is slightly larger than in Fig. 1f, which is not surprising, since the metric tensor chosen for Fig. 1 is optimal for the L^2 error. Thus, adding the equidistribution property to the meshing functional seems to help to obtain meshes which are closer to fulfilling the prescribed properties. Interestingly, the L^2 error for the W functional seems to be a bit smaller if the H^1 metric

tensor is used. This can be explained by the fact that the W functional does not incorporate the equidistribution property and, thus, doesn't exactly generate a mesh which is uniform with respect to the provided metric tensor. Thus, it is not quite clear if one is able to generate the optimal mesh when using the W functional.

4.2   Three dimensions

In three dimensions, we consider the unit cube $\Omega = (0,1) \times (0,1) \times (0,1)$ and the following test functions.

Example 4.3.

$$
\begin{aligned}
u = & \tanh\left(30\left[(4x-2.0)^2+(4y-2.0)^2+(4z-2.0)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-2.5)^2+(4y-2.5)^2+(4z-2.5)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-2.5)^2+(4y-1.5)^2+(4z-2.5)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-1.5)^2+(4y-2.5)^2+(4z-2.5)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-1.5)^2+(4y-1.5)^2+(4z-2.5)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-2.5)^2+(4y-2.5)^2+(4z-1.5)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-2.5)^2+(4y-1.5)^2+(4z-1.5)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-1.5)^2+(4y-2.5)^2+(4z-1.5)^2-0.1875\right]\right) \\
& +\tanh\left(30\left[(4x-1.5)^2+(4y-1.5)^2+(4z-1.5)^2-0.1875\right]\right).
\end{aligned}
$$

Example 4.4.

$$
u = \tanh\left(-30[z-0.5-0.25\sin(2\pi x)\sin(\pi y)]\right).
$$

Example 4.5.

$$
u = \tanh\left(-30\left\{z-\tanh\left(-30[y-0.5-0.25\sin(2\pi x)]\right)\right\}\right).
$$

Adaptive mesh examples (slice and clip cuts) and numerical results are given in Figs. 4 to 6.

As in two dimensions, all three functionals provide good size and shape adaptation, with Q_{eq} and Q_{ali} being reasonably small. The best mesh size control Q_{eq} is given by HR functional (Figs. 4g, 5g and 6g), although for the considered examples, HR has a slightly worse mesh alignment quality Q_{ali} than the others (Figs. 4h, 5h, and 6h).

A closer look at the example meshes (slice cuts) reveals that, as in 2D, W functional —based on variable diffusion— is noticeably more aggressive in moving nodes toward the steep features or, alternatively, one can say that the functionals (3.7) and (3.9) based

(a) Winslow's (3.2)

(b) Huang's (3.7)

(c) Huang and Russell's (3.9)

(d) Q_{eq} vs. N

(e) Q_{ali} vs. N
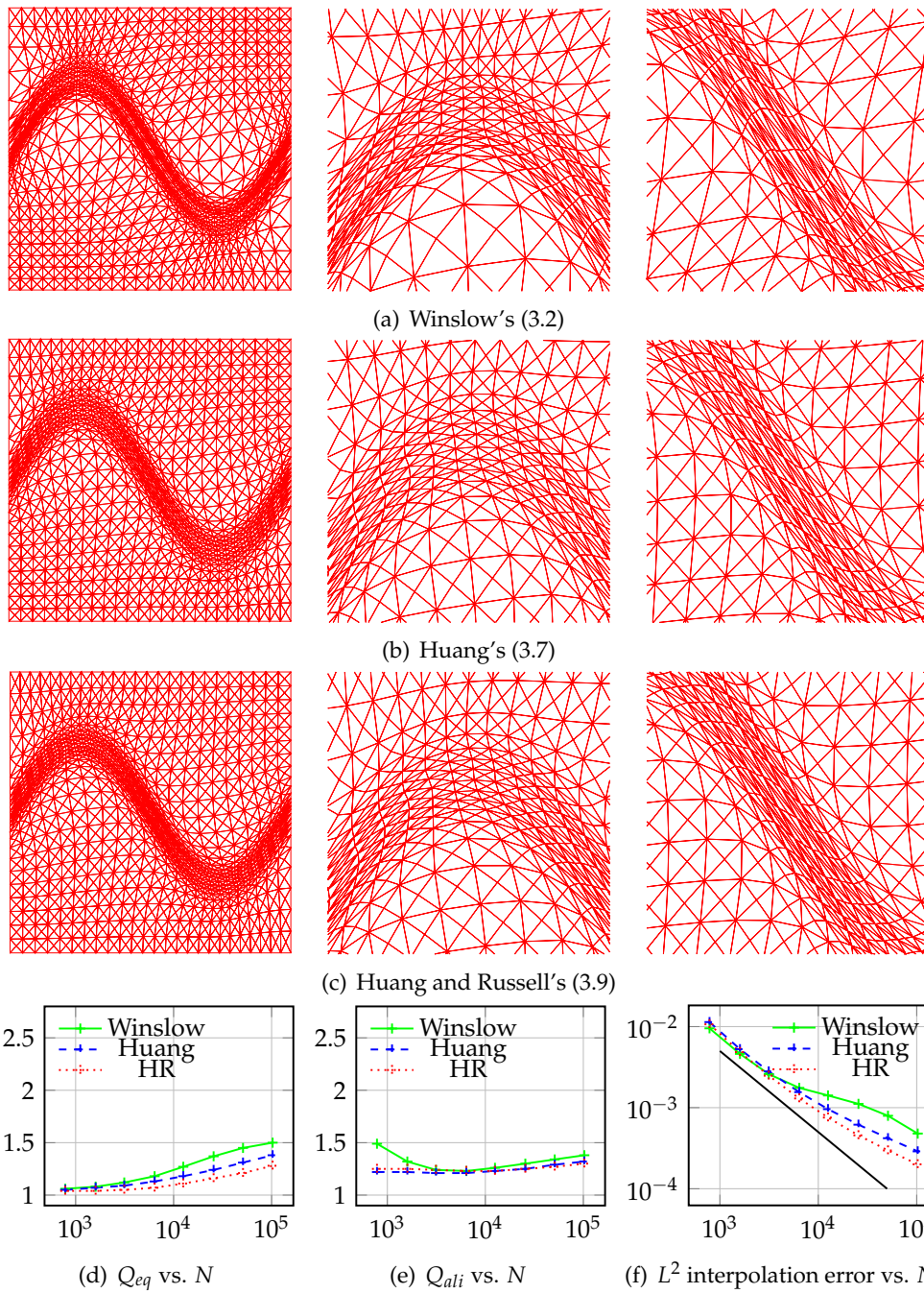
(f) L^2 interpolation error vs. N

Figure 1: Example 4.1: example meshes (left), close-ups near the wave tip (middle) and in the middle (right), mesh quality measures, and L^2 interpolation error (black line represents N^{-1}).

(a) Winslow's (3.2)

(b) Huang's (3.7)

(c) Huang and Russell's (3.9)

(d) Q_{eq} vs. N

(e) Q_{ali} vs. N
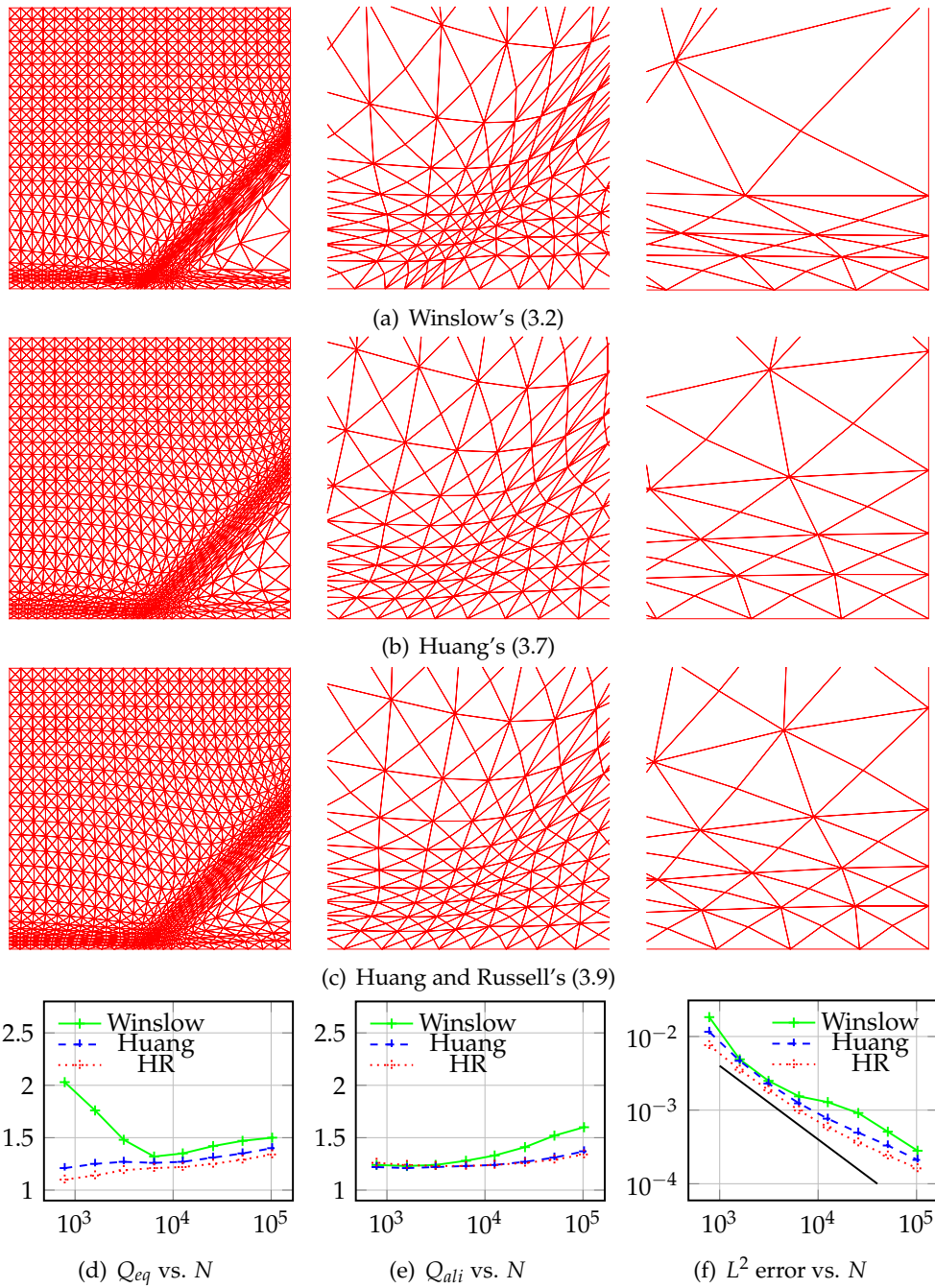
(f) L^2 error vs. N

Figure 2: Example 4.2: example meshes (left), close-ups near the wave meeting the boundary layer (middle) and in the right bottom corner (right), mesh quality measures, and L^2 interpolation error (black line represents N^{-1}).

(a) Winslow's (3.2)

(b) Huang's (3.7)

(c) Huang and Russell's (3.9)

(d) Q_{eq} vs. N

(e) Q_{ali} vs. N
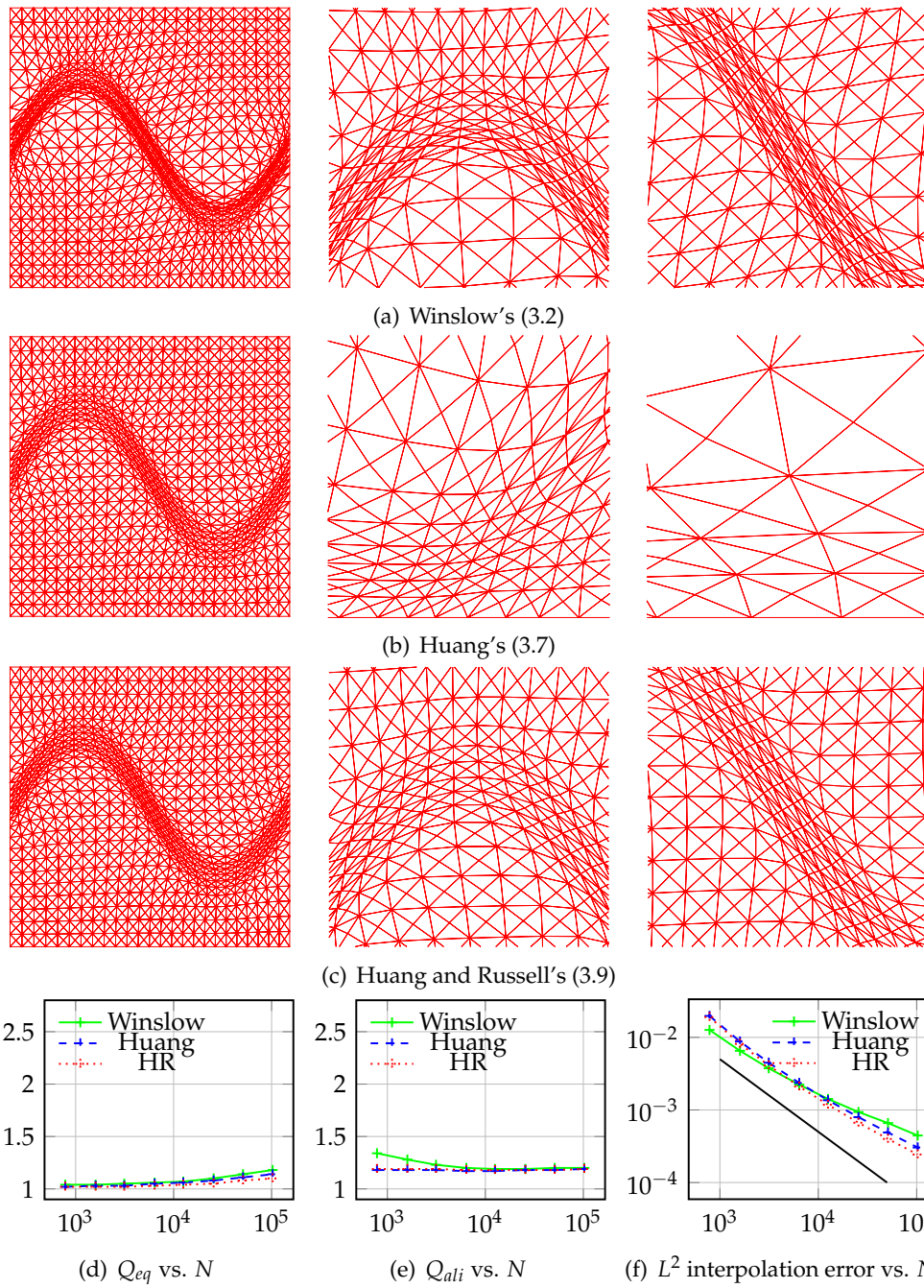
(f) L^2 interpolation error vs. N

Figure 3: Example 4.1: using a metric tensor for the H^1 semi-norm: example meshes (left), close-ups near the wave tip (middle) and in the middle (right), mesh quality measures, and L^2 interpolation error (black line represents N^{-1}).
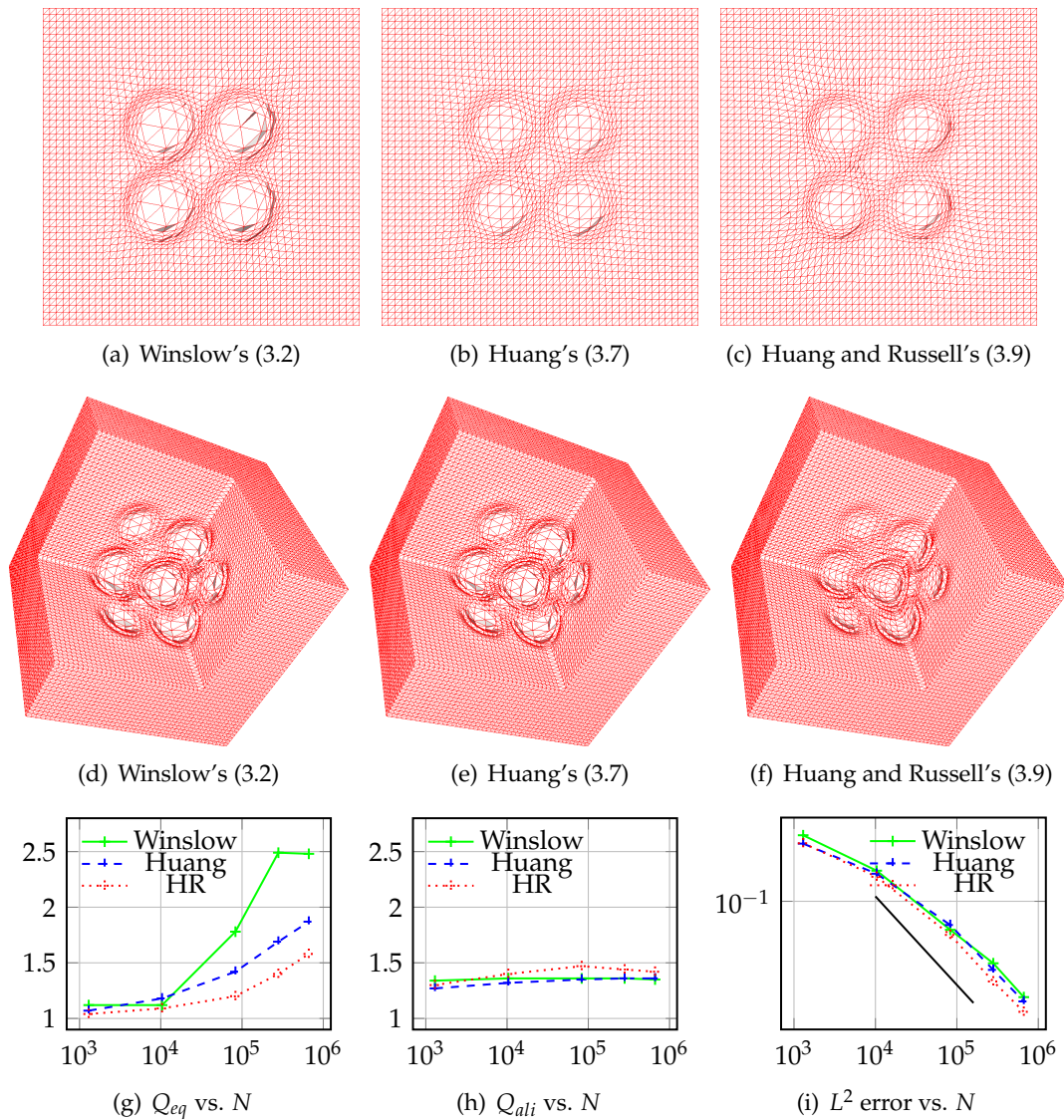
(a) Winslow's (3.2)   (b) Huang's (3.7)   (c) Huang and Russell's (3.9)

(d) Winslow's (3.2)   (e) Huang's (3.7)   (f) Huang and Russell's (3.9)

(g) Q_{eq} vs. N (h) Q_{ali} vs. N (i) L^2 error vs. N

Figure 4: Example 4.3: The top row: slice cuts of the meshes. The middle row: clip cuts of the meshes. The bottom row: Q_{eq}, Q_{ali}, and the L^2 norm of the linear interpolation error (black line represents $N^{-2/3}$).

on equidistribution and alignment distribute the nodes with the better correspondence with the given $\mathbb{M}$. For coarse meshes, all three functionals provide similar results (see convergence plots in Figs. 4i, 5i, 6i); however, for fine meshes, sizing of mesh elements obtained by means of W functional is not quite as good as for H and HR functionals, as indicated by a larger Q_{eq}.

Altogether, the linear interpolation error (Figs. 4i, 5i, and 6i) suggests that HR functional provides the best mesh, followed by H and W functionals. One may notice from

(a) Winslow's (3.2)   (b) Huang's (3.7)   (c) Huang and Russell's (3.9)

(d) Winslow's (3.2)   (e) Huang's (3.7)   (f) Huang and Russell's (3.9)

(g) Q_{eq} vs. N (h) Q_{ali} vs. N (i) L^2 error vs. N

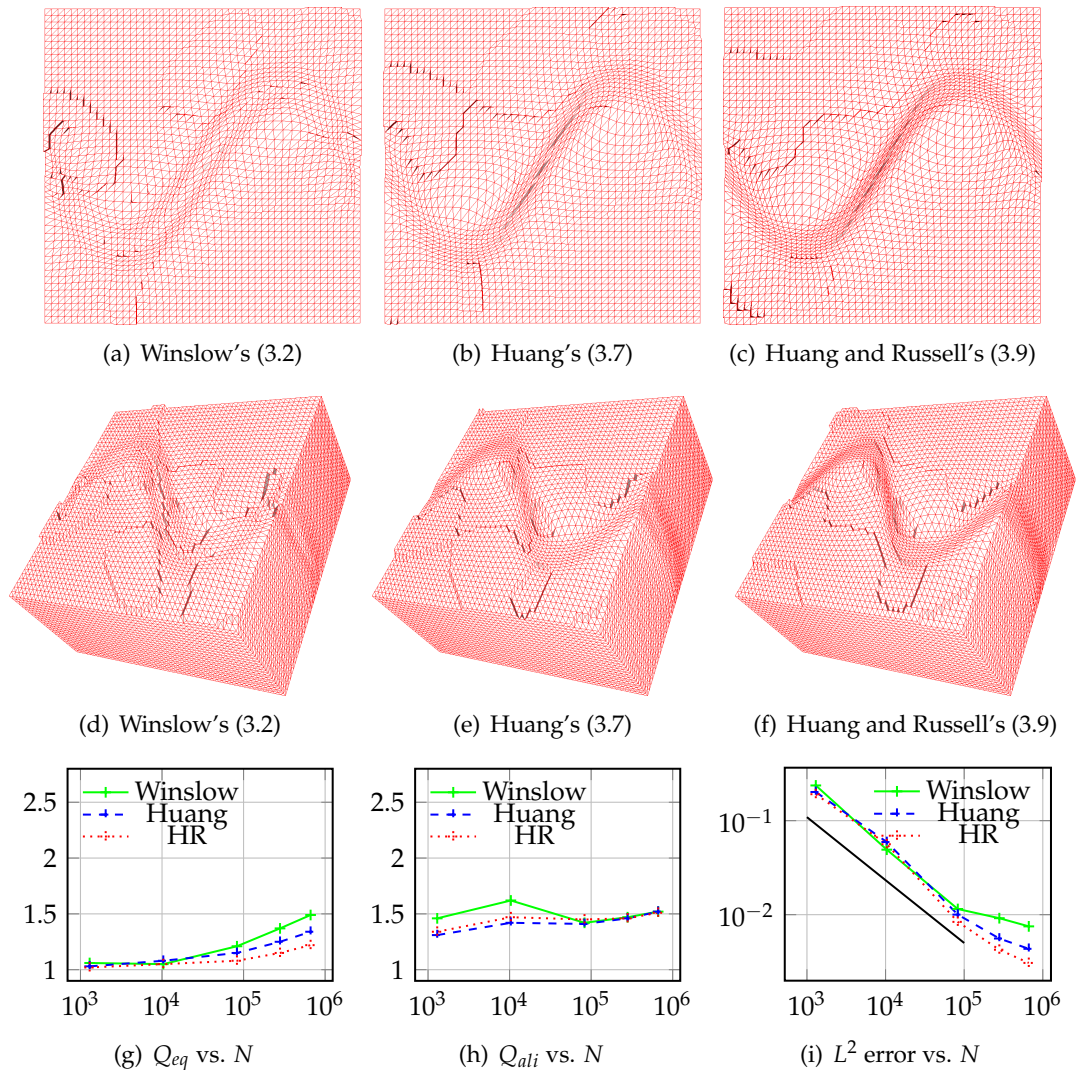Figure 5: Example 4.4: The top row: slice cuts of the meshes. The middle row: clip cuts of the meshes. The bottom row: Q_{eq}, Q_{ali}, and the L^2 norm of the linear interpolation error (black line represents $N^{-2/3}$).

Figs. 5i and 6i that the convergence of the linear interpolation error for W functional slows down near $N = 10^5$ for Examples 4.4 and 4.5, although it seems to improve as the mesh is refined (Fig. 6i). The reason for this behaviour is not clear to us. On the other hand, W functional has the simplest form and seems to be more economic to compute than the other two. From tentative comparison, mesh generation using W functional uses about one fifth to an half of the CPU time used with H or HR functional. Qualitatively, this is not difficult to understand since W functional is convex whereas the others are not (although they are polyconvex).
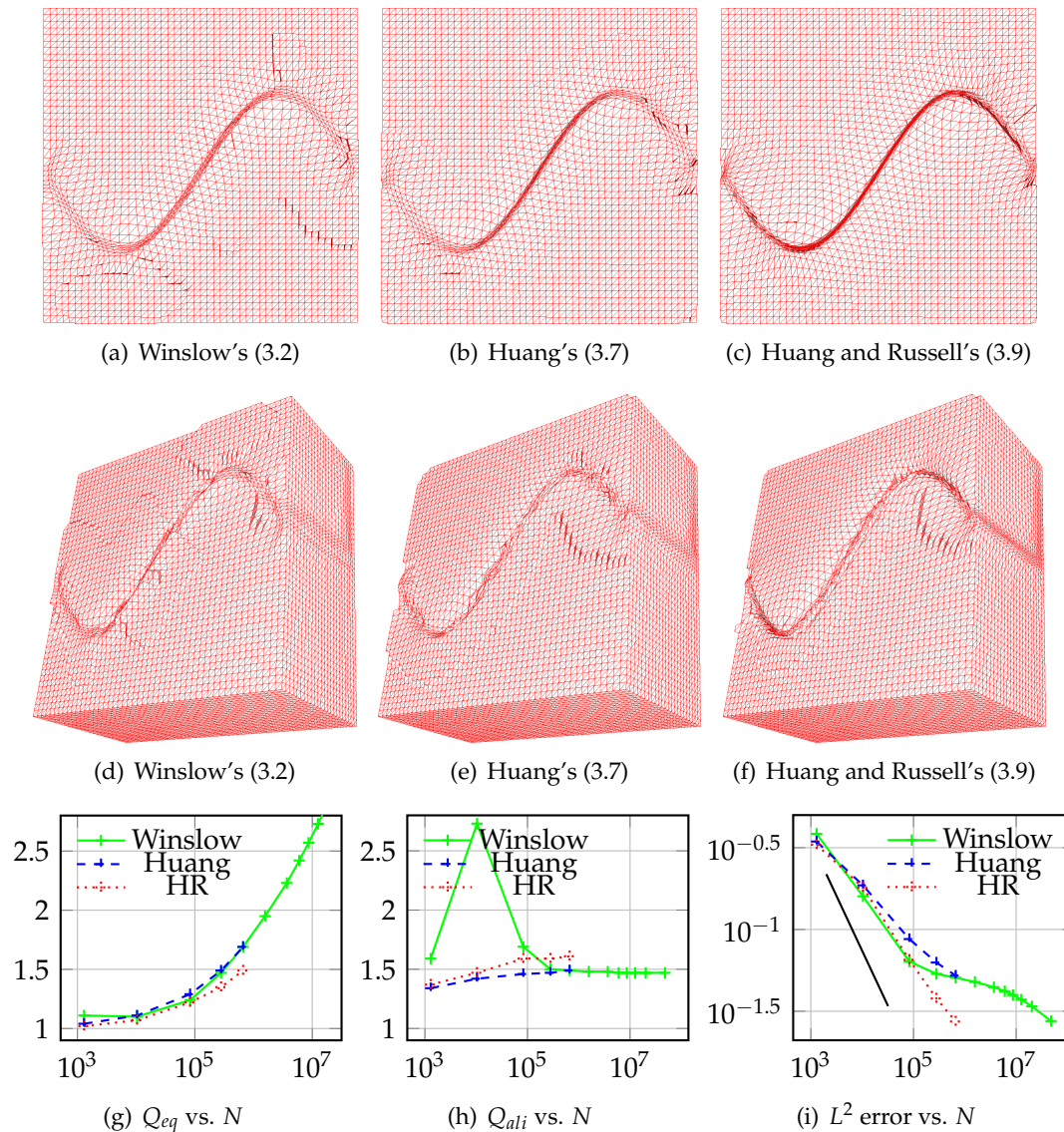
(a) Winslow's (3.2)   (b) Huang's (3.7)   (c) Huang and Russell's (3.9)

(d) Winslow's (3.2)   (e) Huang's (3.7)   (f) Huang and Russell's (3.9)

(g) Q_{eq} vs. N (h) Q_{ali} vs. N (i) L^2 error vs. N

Figure 6: Example 4.5: The top row: slice cuts of the meshes. The middle row: clip cuts of the meshes. The bottom row: Q_{eq}, Q_{ali}, and the L^2 norm of the linear interpolation error (black line represents $N^{-2/3}$).

5 Conclusions

Among the three functionals in this study, Huang and Russell's functional consistently provides the best mesh for piecewise linear interpolation in both two and three dimensions. In all examples it leads to the best equidistribution quality and the smallest interpolation error. Interestingly, while it results in the best mesh alignment quality in two dimensions, the functional gives a slightly worse mesh alignment than the other two

functionals in three dimensions.

Meshes obtained by means of Winslow's functional have the worst mesh equidistribution (element sizing) quality and the largest interpolation error in four out of five examples, although in three dimensions its mesh alignment is quite good and even better than that of the meshes obtained using Huang and Russell's functional. A possible explanation to this behavior could be the fact that this functional does not have an explicit mechanism to control the equidistribution property.

The behavior of Huang's functional is somewhere in between Winslow's and Huang and Russell's functionals: both in mesh quality measures and interpolation error. It provides better mesh sizing than Winslow's functional but not quite as good as Huang and Russell's. On the other hand, it provides the best (or very close to the best) mesh alignment in all examples.

While being able to produce correct and good quality mesh concentration, Winslow's functional seems to have the tendency to move more points toward the area of interest and is slightly less reliable than the other two functionals especially when the mesh is fine. On the other hand, it has a very simple form and is more economic to compute than the others. It can be a good choice for mesh adaptation at least for coarser meshes, for which all of the three functionals produce comparable meshes.

Finally, it should be pointed out that the numerical experiment we conducted in this work is limited and more work is needed to have an extensive and more complete understanding of the behavior of the meshing functionals especially in three dimensions. Moreover, the newly developed implementation of the variational methods in [11] has been crucial to the current study to perform substantial computations in two and three dimensions. It is our hope that it can serve as an efficient tool for use in future studies of mesh adaptation and movement.

Acknowledgments

References

[1] G. Beckett, J. A. Mackenzie, A. Ramage D. M. Sloan. Computational solution of two-dimensional unsteady PDEs using moving mesh methods. J. Comput. Phys., 182: 478–495, 2002.
[2] G. Beckett, J. A. Mackenzie M. L. Robertson. A moving mesh finite element method for the solution of two-dimensional Stefan problems. J. Comput. Phys., 168: 500–518, 2001.
[3] J. U. Brackbill and J. S. Saltzman. Adaptive zoning for singular problems in two dimensions. J. Comput. Phys., 46: 342–368, 1982.
[4] W. Cao, W. Huang and R. D. Russell. A study of monitor functions for two dimensional adaptive mesh generation. SIAM J. Sci. Comput., 20: 1978–1994, 1999.

 [5] H. D. Ceniceros and T. Y. Hou. An efficient dynamically adaptive mesh for potentially sin-
     gular solutions. J. Comput. Phys., 172: 609–639, 2001.
 [6] A. S. Dvinsky. Adaptive grid generation from harmonic maps on Riemannian manifolds. J.
     Comput. Phys., 95: 450–476, 1991.
 [7] W. Huang. Variational mesh adaptation: isotropy and equidistribution. J. Comput. Phys. ,
     174: 903–924, 2001.
 [8] W. Huang. Measuring mesh qualities and application to variational mesh adaptation. SIAM
     J. Sci. Comput., 26: 1643–1666, 2005.
 [9] W. Huang. Metric tensors for anisotropic mesh generation. J. Comput. Phys. , 204: 633–665,
     2005.
[10] W. Huang. Mathematical principles of anisotropic mesh adaptation. Comm. Comput. Phys.,
     1: 276–310, 2006.
[11] W. Huang and L. Kamenski. A geometric discretization and a simple implementation for
     variational mesh generation and adaptation. J. Comput. Phys. (Submitted), arXiv:1410.7872,
     2014.
[12] W. Huang, Y. Ren and R. D. Russell. Moving mesh partial differential equations (MMPDEs)
     based upon the equidistribution principle. SIAM J. Numer. Anal. , 31: 709–730, 1994.
[13] W. Huang and R. D. Russell. A high dimensional moving mesh strategy. Appl. Numer.
     Math., 26: 63–76, 1998.
[14] W. Huang and R. D. Russell. Moving mesh strategy based upon a gradient flow equation
     for two dimensional problems. SIAM J. Sci. Comput., 20: 998–1015, 1999.
[15] W. Huang and R. D. Russell. Adaptive Moving Mesh Methods. Springer, New York, 2011.
     Applied Mathematical Sciences Series, Vol. 174.
[16] W. Huang and W. Sun. Variational mesh adaptation II: error estimates and monitor func-
     tions. J. Comput. Phys., 184: 619–648, 2003.
[17] L. Kamenski. Anisotropic Mesh Adaptation Based on Hessian Recovery and A Posteriori
     Error Estimates. Verlag Dr. Hut, München, 2009. PhD Thesis, Technische Universität Darm-
     stadt.
[18] L. Kamenski and W. Huang. How a nonconvergent recovered Hessian works in mesh adap-
     tation. SIAM J. Numer. Anal., 52:1692–1708, arXiv:1211.2877, 2014.
[19] P. Knupp and S. Steinberg. Fundamentals of Grid Generation. CRC Press, Boca Raton, 1994.
[20] P. M. Knupp. Jacobian-weighted elliptic grid generation. SIAM J. Sci. Comput., 17: 1475–
     1490, 1996.
[21] P. M. Knupp and N. Robidoux. A framework for variational grid generation: conditioning
     the Jacobian matrix with matrix norms. SIAM J. Sci. Comput., 21: 2029–2047, 2000.
[22] R. Li, T. Tang, and P. W. Zhang. Moving mesh methods in multiple dimensions based on har-
     monic maps. J. Comput. Phys., 170: 562–588, 2001.
[23] V. D. Liseikin. Grid Generation Methods. Springer, Berlin, 1999.
[24] J. F. Thompson, Z. A. Warsi, and C. W. Mastin. Numerical Grid Generation: Foundations
     and Applications. North-Holland, New York, 1985.
[25] A. M. Winslow. Adaptive mesh zoning by the equipotential method. Technical Report UCID-
     19062, Lawrence Livemore Laboratory, 1981 (unpublished).