Surface and bulk moving mesh methods based on equidistribution and alignment

By

Avary Kolasinski

Submitted to the Department of Mathematics and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Committee members

Weizhang Huang, Chair

Estela Gavosto

Agnieszka Mięlar

Suzanne Shontz

Erik Van Vleck

Date defended: May 2, 2019

The Thesis Committee for Avary Kolasinski certifies
that this is the approved version of the following thesis :

Surface and bulk moving mesh methods based on equidistribution and alignment

Weizhang Huang, Chair

Date approved: May 7, 2019

Abstract

In this dissertation, we first present a new functional for variational mesh generation and adaptation that is formulated by combining the equidistribution and alignment conditions into a single condition with only one dimensionless parameter. The functional is shown to be coercive which, when employed with the moving mesh partial differential equation method, allows various theoretical properties to be proved. Numerical examples for bulk meshes demonstrate that the new functional performs comparably to a similar existing functional that is known to work well but contains an additional parameter.

Variational mesh adaptation for bulk meshes has been well developed however, surface moving mesh methods are limited. Here, we present a surface moving mesh method for general surfaces with or without explicit parameterization. The development starts with formulating the equidistribution and alignment conditions for surface meshes from which, we establish a meshing energy functional. The moving mesh equation is then defined as the gradient system of the energy functional, with the nodal mesh velocities being projected onto the underlying surface. The analytical expression for the mesh velocities is obtained in a compact, matrix form, which makes the implementation of the new method on a computer relatively easy and robust. Moreover, it is analytically shown that any mesh trajectory generated by the method remains nonsingular if it is so initially. It is emphasized that the method is developed directly on surface meshes, making no use of any information on surface parameterization. A selection of two-dimensional and three-dimensional examples are presented.

Acknowledgements

I would first like to express my deep gratitude to my advisor, Professor Weizhang Huang, for his support, expert guidance, and encouragement throughout my research. Without his direction and persistent help, this dissertation and all other accomplishments during my graduate school career would have not been possible. In addition, I would like to thank the members of my thesis committee, Professor Estela Gavosto, Professor Agnieszka Międ̨lar, Professor Suzanne Shontz, and Professor Erik Van Vleck for their comments, thoughtful questions, and constructive criticism. Each one of them has been supportive and influential throughout my career. Specifically, I would like to thank Estela Gavosto and Agnieszka Międ̨lar for, on various occasions, taking time from their busy schedules to meet with me and offer new perspective and advice.

I would like to acknowledge my gratitude toward the office support staff: Gloria Prothe, Kerrie Breicheisen, and Lori Springs for their consistent help with paperwork, deadlines, and continuously keeping the department running smoothly.

I would also like to thank my parents, Ed and Sandy, and my sister and brother, Sidney and Taylor. They have supported me beyond belief throughout my life and have shaped me to become the person I am proud to be today. I truly cannot express how thankful I am for the unbound love and support my parents have given me not only during graduate school but my entire life. Additionally, I would like to personally thank my sister, Sidney, for being there for me as a truly amazing sister, roommate, and most importantly, best friend.

I am also deeply thankful for my fiancé, Justin, who was my rock during the most

difficult weeks I endured during graduate school. He believed in me at times when I was too tired to believe in myself and never went without reminding me how strong I was. His unwavering support and love during the last year especially has been crucial and I am quite lucky to have him by my side. I would also like to thank Justin's family for their encouragement the past few years. Specifically, Debbie for the support and patiently listening to my rants about the difficulties of research and teaching.

I wouldn't be in the position I am today without each and every person mentioned above, and some who were not mentioned. I am truly grateful for this experience and am looking forward to the next part of my research journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The solutions of partial differential equations (PDEs) arising in science and engineering can frequently have large variations occurring over small portions of the physical domain. A major challenge when solving such problems is how to appropriately resolve the solution behavior there. When finite difference or finite element methods are employed, a fine mesh is required in these particular regions of the physical domain. Typically, in one-dimension, using a uniform fine mesh is feasible, however, in higher dimensions this can become a substantial computational expense in terms of computer memory and processing time. This particular challenge has led to the study of mesh adaptation methods which consist of three mesh adaptation techniques. The h-method seeks to improve accuracy by adding more mesh points in regions of the domain with large solution variation. In this, the h is a standard notation representing the size of the mesh element. Although intuitive, this approach requires continuously adding elements and changing connectivity of the mesh for which, in some cases, can be expensive or even forbidding in terms of memory. The p-method (where the p stands for polynomial) seeks to improve the accuracy by increasing the polynomial degrees of the solution approximation in regions with large solution error. The p-method, unlike the h-method, does not add more points or change the connectivity of the mesh however, it is difficult to implement in many cases. The final method is the r-method (adaptive moving mesh method) where the r stands for relocation. This method seeks to improve the accuracy by moving mesh points into the regions that require a fine resolution. Adaptive moving mesh methods can attain a similar accuracy to that of a uniform mesh or an h-method with significantly less mesh points.

More specifically, adaptive moving mesh methods consist of two major components: the strategy used to move the mesh and the approach used to solve the system of mesh equations. The mesh movement strategy is typically performed either by solving a system of PDEs involving the mesh coordinate transformation or by doing a direct error-based minimization. Here we focus on the former and take a velocity-based variational approach to formulate the mesh strategy. To solve the system of mesh equations, we use a general ordinary differential equation (ODE) solver. In this variational approach, a (adaptive) mesh is generated as the image of a reference mesh under a coordinate transformation which is determined as the minimizer of a meshing functional. The meshing functional is typically designed to measure the difficulty in the numerical approximation of the physical solution and involves a user determined metric tensor or monitor function to control the movement. Due to the inherent ability of the method to easily incorporate user determined mesh requirements such as smoothness, orthogonality, adaptivity, alignment, etc., it has received considerable attention in the scientific computing community; e.g., see [8, 38, 42, 46, 55] and references therein. In addition to being a method for mesh generation and adaptation, this approach can also be used as a smoothing device for automatic mesh generation [22, 35] and a base for adaptive moving mesh methods [37, 38, 39, 45].

A number of meshing functionals have been developed from different problems and formulated based on different focused requirements. For example, Winslow [59] develops an equipotential method that is based on variable diffusion. Brackbill and Saltzmann [6] combine mesh concentration, smoothness, and orthogonality to create a functional. Dvinsky [17] develops a method based on the energy of harmonic mappings. Knupp [40] and Knupp and Robidoux [41] focus on the idea of conditioning the Jacobian matrix of the coordinate transformation. Huang [30] and Huang and Russell [38] have proposed two methods based on the so-called equidistribution and alignment conditions. Together, these two conditions completely characterize a uniform mesh. Huang [30] formulates a single energy functional (referred to as the existing functional hereafter) by averaging the equidistribution and align-

ment conditions with a dimensionless parameter [30]. The idea is that minimizing the existing energy functional will result in a mesh that closely satisfies both the equidistribution and alignment conditions.

Compared to the algorithmic development, very few theoretical results are known. For example, Dvinsky's meshing functional [17] is guaranteed to have a unique invertible minimizer by the theory of harmonic mappings between multidimensional convex domains. Winslow's functional [59] is known to have a unique minimizer due to its uniformly convexity and coercivity but its invertibility depends on the convexity of the domains. The existing functional based on the equidistribution and alignment condition has been proven to be both coercive and polyconvex and has minimizes. These results, however, are only at the continuous level.

At the discrete level, studies have typically been focused on one spatial dimension. However, Huang and Kamenski have recently proposed a theoretical study on variational mesh generation and adaptation at the discrete level for any dimension. In this method, the meshing functional is first discretized and then the mesh equation is defined as a modified gradient system of the discretized functional. This formulation provides an explicit, compact, and analytical formula for the mesh velocity, which makes the implementation of the method much easier and more robust (see Section 2.2). More importantly, several properties of the discrete MMPDE can be established; see [32] and/or Section 2.3 for detail. In particular, if the meshing functional satisfies the coercivity condition then the mesh trajectory of the discrete MMPDE stays nonsingular if it is so initially. Moreover, the altitudes and volumes of its elements are bounded below by positive numbers that depend only on the initial mesh, the number of elements, and the metric tensor. Furthermore, the value of the discrete functional converges as time increases which can be used as a stopping criteria in computation. Finally, Huang and Kamenski prove that the mesh trajectory has limiting meshes that are critical points of the discrete functional and satisfy the lower bounds on the element altitude and volume.

As mentioned, the existing functional satisfies the coercivity condition and hence, main-

tains the theoretical properties proposed by Huang and Kamenski. Although known to work well in many problems and clearly has significant proven theoretical advantages, the existing functional contains two parameters which can be considered large disadvantages. In particular, it is still unclear how to optimally choose either of the parameters. The performance of the existing functional does not seem sensitive to the value of the parameters however, their choice is still arbitrary and there is hardly a convincing guideline for choosing them. In Chapter 2, we formulate a new functional that combines the equidistribution and alignment conditions without introducing any new parameters. We prove that this new functional satisfies the coercivity condition and thus has similar properties to the existing functional when employed with the MMPDE method. Moreover, two-dimensional numerical results are presented to verify theoretical findings as well as demonstrate comparable performances of the two functionals.

Indeed, variational mesh adaptation for bulk meshes has been studied extensively, as discussed above, however, mesh adaptation methods for surface meshes are limited. There has been some work done on mesh movement and adaptation for surfaces. For example, Crestel et al. [11] present a moving mesh method for parametric surfaces by generalizing Winslow's meshing functional to Riemannian manifolds and taking into consideration the Riemannian metric associated with the manifolds. The method is simplified and implemented on a two-dimensional domain for surfaces that accept certain parameterizations. Weller et al. [7] and McRae et al. [48] solve a Monge-Ampére type equation on the surface of the sphere to generate optimally transported meshes that become equidistributed with respect to a suitable monitor function. MacDonald et al. [47] devise a moving mesh method for the numerical simulation of coupled bulk-surface reaction-diffusion equations on an evolving two-dimensional domain. They use a one-dimensional moving mesh equation in arclength to concentrate mesh points along the evolving domain boundary. Dassi et al. [13] generalize the higher embedding approach proposed in [44]. They modify the embedding map between the underlying surface and $\mathbb{R}^6$ to include more information associated with the physical

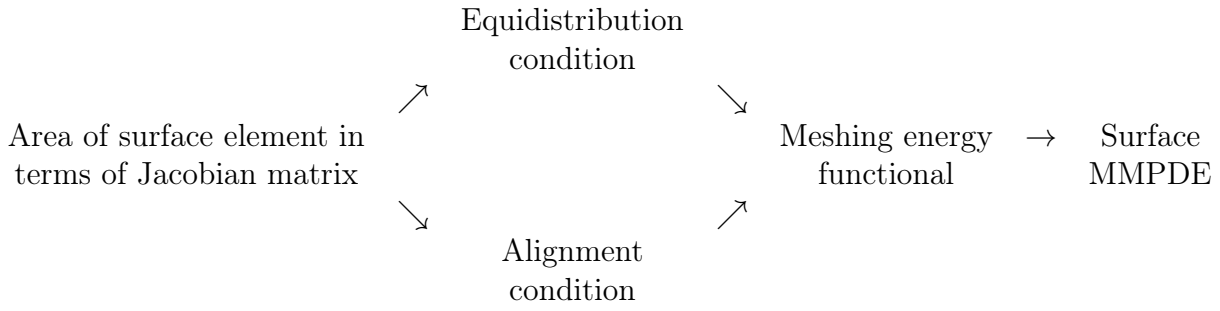
solution and its gradient. The idea behind this mapping is that it essentially approximates the geodesic length on the surface via a Euclidean length in $\mathbb{R}^6$. The mesh adapts in the Euclidean space and then is mapped back to the physical domain.

We, however, are interested in methods that can directly move simplicial meshes on general surfaces with or without analytical expressions. Such surface moving mesh methods can be used for adaptation and/or quality improvements of surface meshes and thus are useful for computational geometry and numerical solutions of partial differential equations (PDEs) defined on surfaces; e.g., see [15, 19, 56]. More specifically, the functionals based on equidistribution and alignment when combined with the MMPDE method for bulk meshes is known to work well hence, we are interested in developing an extension of this method to surface meshes.

The main challenges in the development of surface mesh movement come from the fact that the Jacobian matrix of the affine mapping between the reference element and any simplicial surface element is not square. This prevents us from using any bulk mesh results. To overcome these challenges (see Chapter 3), we start by connecting the area of the surface element in the Euclidean metric or a Riemannian metric with the Jacobian matrix of the corresponding affine mapping. This connection allows us to formulate the equidistribution and alignment conditions and ultimately, form a single meshing energy functional for surface meshes. This meshing functional is similar to Huang's functional [28, 30, 34, 43] for bulk meshes which has been proven to work well in a variety of problems. Following the MMPDE approach, we define the surface moving mesh equation as the gradient system of the meshing functional, with the nodal mesh velocities being projected onto the underlying surface. The analytical expression for the mesh velocities is obtained in a compact, matrix form, which makes the implementation of the new method on a computer relatively easy and robust. The steps in developing the surface moving mesh method are given in Table 1.1.

Several theoretical properties are obtained for the surface moving mesh method. In particular, it is proven that a surface mesh generated by the method remains nonsingular

Table 1.1: Steps in developing the surface moving mesh method.



for all time if it is so initially. Moreover, the altitudes and areas of the physical elements are bounded below by positive constants depending only on the initial mesh, the number of elements, and the metric tensor that is used to provide information on the size, shape, and orientation of the elements throughout the surface. Furthermore, limiting meshes exist and the meshing functional is decreasing along each mesh trajectory. These properties are verified in numerical examples.

It is emphasized that the new method is developed directly on surface meshes, making no use of any information on surface parameterization. It utilizes surface normal vectors to ensure that the mesh vertices remain on the surface while moving. Since the surface normal vectors can be computed even when the surface only has a numerical representation, the new method can apply to general surfaces with or without explicit parameterization. A large selection of two- and three-dimensional examples are presented in Chapter 3.

Chapter 2

Bulk Mesh

In this chapter we are going to describe two meshing functionals that are formulated from the equidistribution and alignment conditions (see (2.6) and (2.7) below). These conditions have been developed based on the concept of uniform meshes in some metric tensor [38]. They provide total control of the mesh element size, shape, and orientation of mesh elements through a metric tensor. The first meshing functionals to be described was first introduced in [30] and involves averaging functionals associated with the equidistribution and alignment conditions. It has a number of advantages and is known to work well in practice but involves two dimensionless parameters. Although the performance of the functional does not seem sensitive to the value of the parameters, the choices are still arbitrary and there is hardly a convincing guideline for choosing them.

The second functional is new. It is formulated by directly combining the equidistribution and alignment conditions into a single condition which, in turn, has eliminated one of the two parameters in the existing functional. We will show that the new functional satisfies the coercivity condition and has similar theoretical properties as the existing functional when employed with the MMPDE method. In particular, we will show that the mesh associated to the MMPDE remains nonsingular for all time if it is so initially. Moreover, the mesh trajectory has limiting meshes, all of which are nonsingular. Two-dimensional numerical results will be presented to verify theoretical findings as well as demonstrate comparable performances of the two functionals.

2.1 Meshing functionals based on equidistribution and alignment

2.1.1 Equidistribution and alignment

Let the physical domain, $\Omega \subset \mathbb{R}^d$, $d \geq 1$, be a bounded (not necessarily convex) polygonal or polyhedral domain and $\mathbb{M} = \mathbb{M}(\boldsymbol{x})$ be a given symmetric, uniformly positive definite metric tensor defined on Ω which satisfies

$$\underline{m} I \leq \mathbb{M}(\boldsymbol{x}) \leq \overline{m} I, \quad \forall \boldsymbol{x} \in \Omega, \tag{2.1}$$

where $\underline{m}$ and $\overline{m}$ are positive constants, I is the identity matrix, and the inequality is in terms of negative semi-definiteness. Our goal is to generate a simplicial mesh for Ω which is uniform with respect to the metric $\mathbb{M}$. Denote this target mesh by $\mathcal{T}_h = \{K\}$ and let N and N_v be the number of its elements and vertices, respectively. Assume that the reference element $\hat{K}$ has been chosen to be equilateral and unitary (i.e., $|\hat{K}| = 1$, where $|\hat{K}|$ denotes the volume of $\hat{K}$). For any element $K \in \mathcal{T}_h$ let $F_K : \hat{K} \subset \mathbb{R}^d \to K \subset \mathbb{R}^d$ be the affine mapping between them and $F_K' \in \mathbb{R}^{d \times d}$ be its Jacobian matrix. Denote the vertices of K by $\boldsymbol{x}_j^K$, $j = 0, ..., d$ and the vertices of $\hat{K}$ by $\boldsymbol{\xi}_j$, $j = 0, ..., d$. Then

$$\boldsymbol{x}_j^K = F_K(\boldsymbol{\xi}_j).$$

From this we have

$$\boldsymbol{x}_j^K - \boldsymbol{x}_0^K = F_K(\boldsymbol{\xi}_j - \boldsymbol{\xi}_0)$$

or

$$[\boldsymbol{x}_1^K - \boldsymbol{x}_0^K, \ldots, \boldsymbol{x}_d^K - \boldsymbol{x}_0^K] = F_K'[\boldsymbol{\xi}_1^K - \boldsymbol{\xi}_0^K, \ldots, \boldsymbol{\xi}_d^K - \boldsymbol{\xi}_0^K]$$

which gives $F_K' = E_K \hat{E}^{-1}$, where E_K and $\hat{E}$ are the edge matrices for K and $\hat{K}$, i.e.,

$$E_K = [\boldsymbol{x}_1^K - \boldsymbol{x}_0^K, \ldots, \boldsymbol{x}_d^K - \boldsymbol{x}_0^K], \quad \hat{E} = [\boldsymbol{\xi}_1^K - \boldsymbol{\xi}_0^K, \ldots, \boldsymbol{\xi}_d^K - \boldsymbol{\xi}_0^K].$$

It should be noted that since $\hat{K}$ is not degenerate, $\hat{E}^{-1} \in \mathbb{R}^{d\times d}$ exists. From this one can readily see that

$$|K| = \det\left(F_K'\right), \tag{2.2}$$

where $|K|$ denotes the volume of K in the Euclidean norm. Moreover, define

$$\mathbb{M}_K = \frac{1}{|K|}\int_K \mathbb{M}(\boldsymbol{x})d\boldsymbol{x} \tag{2.3}$$

and recall that the Riemannian distance in $\mathbb{M}_K$, denoted $\|\cdot\|_{\mathbb{M}_K}$, is given by

$$\|\boldsymbol{x}\|_{\mathbb{M}_K} = \sqrt{\boldsymbol{x}^T\mathbb{M}_K\boldsymbol{x}} = \sqrt{\left(\mathbb{M}_K^{\frac{1}{2}}\boldsymbol{x}\right)^T\left(\mathbb{M}_K^{\frac{1}{2}}\boldsymbol{x}\right)} = \left\|\mathbb{M}_K^{\frac{1}{2}}\boldsymbol{x}\right\|, \tag{2.4}$$

where $\|\cdot\|$ denotes the Euclidean metric. That is, the geometric properties of K in the metric $\mathbb{M}_K$ can be obtained from those of $\mathbb{M}_K^{\frac{1}{2}}$ in the Euclidean metric. Therefore

$$|K|_{\mathbb{M}_K} = |\mathbb{M}_K^{\frac{1}{2}}K| = \det(\mathbb{M}_K)^{\frac{1}{2}}|K| = \det\left(F_K'\right)\det\left(\mathbb{M}_K\right)^{1/2}, \tag{2.5}$$

where $|K|_{\mathbb{M}}$ denotes the volume of K in the metric $\mathbb{M}$.

With this in mind, we can define the equidistribution and alignment conditions that completely characterize a non-uniform mesh. Indeed, any non-uniform mesh can be viewed as a uniform one in some metric tensor. Using this viewpoint it is shown (e.g., see [38]) that a uniform mesh in the metric $\mathbb{M}$ satisfies that, $\forall K \in \mathcal{T}_h$,

$$\text{equidistribution:} \quad |K|\det(\mathbb{M}_K)^{\frac{1}{2}} = \frac{\sigma_h}{N}, \tag{2.6}$$

$$\text{alignment:} \quad \frac{1}{d}\text{tr}\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right) = \det\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right)^{\frac{1}{d}}, \tag{2.7}$$

where

$$\sigma_h = \sum_{K\in\mathcal{T}_h}|K|\det(\mathbb{M}_K)^{\frac{1}{2}}. \tag{2.8}$$

From (2.5) we can see that the equidistribution condition essentially requires that all of the elements have the same volume with respect to the metric $\mathbb{M}$. On the other hand, the left- and right-hand sides of the alignment condition (2.7) are the arithmetic mean and geometric mean of the eigenvalues of the matrix $(F'_K)^{-1}\mathbb{M}_K^{-1}(F'_K)^{-T}$, respectively. Thus, the condition implies that the eigenvalues of the matrix be equal, i.e.,

$$(F'_K)^{-1}\mathbb{M}_K^{-1}(F'_K)^{-T} = \theta_K I, \tag{2.9}$$

where θ_K is a positive constant. It can be shown [38] that geometrically, the condition (2.7) requires all elements K, when measured in the metric $\mathbb{M}_K$, to be similar to the reference element $\hat{K}$. Combining the equidistribution and alignment conditions, we see that if a mesh satisfies both of them then all of its elements have the same volume and are similar to the reference element, thus are uniform with respect to the metric $\mathbb{M}$.

2.1.2 The existing functional

We now describe the existing meshing functional based on the equidistribution and alignment conditions. To this end, first consider the equidistribution condition (2.6). From Hölder's inequality, for any $p > 1$ then

$$\left(\sum_{K\in\mathcal{T}_h} \frac{|K|\det(\mathbb{M}_K)^{\frac{1}{2}}}{\sigma_h} \cdot \left(\frac{1}{|K|\det(\mathbb{M}_K)^{\frac{1}{2}}}\right)^p\right)^{\frac{1}{p}} \geq \sum_{K\in\mathcal{T}_h} \frac{|K|\det(\mathbb{M}_K)^{\frac{1}{2}}}{\sigma_h} \cdot \left(\frac{1}{|K|\det(\mathbb{M}_K)^{\frac{1}{2}}}\right), \tag{2.10}$$

with equality if and only if

$$\frac{1}{|K|\det(\mathbb{M}_K)^{\frac{1}{2}}} = \text{constant}, \quad \forall K \in \mathcal{T}_h.$$

That is, minimizing the difference between the left-hand side and the right-hand side of (2.10) tends to make $1/(|K|\det(\mathbb{M}_K)^{\frac{1}{2}})$ constant for all $K \in \mathcal{T}_h$. Noticing that the right-hand side of (2.10) is N/σ_h, we can rewrite this inequality as

$$\sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}} \cdot \left(\frac{1}{|K|\det(\mathbb{M}_K)^{\frac{1}{2}}}\right)^p \geq \left(\frac{N}{\sigma_h}\right)^p \cdot \sigma_h \quad . \tag{2.11}$$

Since $\sigma_h \approx \int_\Omega \det(\mathbb{M})^{\frac{1}{2}} d\boldsymbol{x}$, it only weakly depends on the mesh so we can consider σ_h to be a constant. Therefore, we can use the left-hand side of (2.11) as the functional for the equidistribution condition. By (2.2), we thus have the equidistribution energy funtional[1]

$$I_{eq}(\mathcal{T}_h) = d^{\frac{dp}{2}} \sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}} \left(\det(F_K')^{-1}\det(\mathbb{M}_K)^{-\frac{1}{2}}\right)^p, \tag{2.12}$$

where $d^{\frac{dp}{2}}$ has been added to agree with the alignment energy functional (2.14) below and $|K|_{\mathbb{M}_K} = |K|\det(\mathbb{M}_K)^{\frac{1}{2}}$ is a weight.

We now consider the alignment condition (2.7). Recall that its left- and right-hand sides are the arithmetic and geometric mean of the eigenvalues of the matrix $(F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}$, respectively. By the arithmetic-mean geometric-mean inequality, we have

$$\frac{1}{d}\mathrm{tr}\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right) \geq \det\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right)^{\frac{1}{d}}, \tag{2.13}$$

with equality if and only if all of the eigenvalues are equal. From this, we have

$$\left(\mathrm{tr}\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right)\right)^{\frac{dp}{2}} \geq d^{\frac{dp}{2}} \left(\det(F_K')^{-1}\det(\mathbb{M}_K)^{-\frac{1}{2}}\right)^p$$

[1]In literature, typcally an energy functional refers to the continuous case, i.e., integration, whereas an energy function refers to the discrete case, i.e., summation. For the purpose of consistency, however, in this thesis we will refer to both as an energy functional.

and thus

$$\sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}}\left(\mathrm{tr}\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right)\right)^{\frac{dp}{2}} \geq \sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}} d^{\frac{dp}{2}}\left(\det(F_K')^{-1}\det(\mathbb{M}_K)^{-\frac{1}{2}}\right)^p,$$

where $p>0$ and $|K|\det(\mathbb{M}_K)^{\frac{1}{2}}$ have been added to agree with the equidistribution energy functional (2.12). Minimizing the difference of the left- and right-hand sides makes the mesh tend to satisfying the alignment condition. Therefore, we can define our alignment functional as

$$I_{ali}(\mathcal{T}_h) = \sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}}\mathrm{tr}\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right)^{\frac{dp}{2}} - d^{\frac{dp}{2}}|K|\det(\mathbb{M}_K)^{\frac{1}{2}}\left(\frac{1}{\det(F_K')\det(\mathbb{M}_K)^{\frac{1}{2}}}\right)^p. \tag{2.14}$$

We now have two functionals and want to obtain a mesh that tries to minimize both. One way to ensure this is to combine the two functionals into a single one. For example, we can average the equidistribution functional (2.12) and the alignment functional (2.14) with a dimensionless parameter $\theta\in[0,1]$, i.e.,

$$\begin{aligned} I_h(\mathcal{T}_h) &= \theta I_{ali}(\mathcal{T}_h) + (1-\theta)I_{eq}(\mathcal{T}_h) \\ &= \theta\sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}}\left(\mathrm{tr}\left((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\right)\right)^{\frac{dp}{2}} \\ &\quad + (1-2\theta)d^{\frac{dp}{2}}\sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}}\left(\det(F_K')^{-1}\det(\mathbb{M}_K)^{-\frac{1}{2}}\right)^p. \end{aligned} \tag{2.15}$$

This functional was first proposed in [30] in the continuous form. As one can notice, the equidistribution and alignment conditions are balanced in equation (2.15) by the dimensionless parameter θ, for which full alignment is achieved when $\theta=1$ and full equidistribution

is achieved when $\theta = 0$. For $0 < \theta \leq \frac{1}{2}$ and $p > 1$, the functional is coercive [38], i.e.,

$$G \geq \alpha \left[\text{tr} \left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T \right) \right]^q - \beta, \tag{2.16}$$

where the functional is written in the form

$$I_h = \sum_{K \in \mathcal{T}_h} |K| G,$$

(see (2.19) for more details) and $\alpha > 0$, $\beta \geq 0$, and $q > d/2$. More specifically, for the existing functional $\alpha = \theta \underline{m}^{d/2}$, $\beta = 0$, and $q = \frac{dp}{2}$ in (2.16). Coercivity gives rise to a number of theoretical properties. One important one being that the MMPDE mesh equation (see Section 2.2) associated with this functional has a mesh trajectory that stays nonsingular for all time if so initially and has element volumes and altitudes bounded away from zero [28]. The functional has also been successfully used for many problems.

2.1.3 The new functional

The existing functional (2.15) contains two dimensionless parameters which can be considered large disadvantages, especially since it is still unclear how to choose an optimal θ. Ideally we would like to take $\theta = 1/2$ to ensure (2.15) is convex, but, unfortunately, previous numerical experiments show that this choice of θ does not put enough emphasis on the equidistribution condition which controls the mesh concentration. Moreover, larger values of θ emphasize the alignment condition which produces a more regular mesh however, this regularity can also be achieved by choosing larger values of p [38]. It has been known experimentally that $\theta = 1/3$ and $p = 3/2$ work well for many problems but this relation between θ and p is not very clear. Here, we consider a new functional that eliminates the additional parameter θ. To this end, we first notice that (2.6) and (2.7) can be cast in a single condition. Indeed,

taking the determinant of both sides of (2.9), we obtain

$$\theta_K^d = \det((F_K')^{-T}\mathbb{M}_K^{-1}F_K'^{-1}) = \det(F_K')^{-2}\det(\mathbb{M}_K)^{-1} = |K|^{-2}\det(\mathbb{M}_K)^{-1},$$

which gives

$$|K|\det(\mathbb{M}_K)^{\frac{1}{2}} = \theta_K^{-\frac{d}{2}}.$$

Comparing this to the equidistribution condition (2.6) we get

$$\theta_K = \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}}.$$

Thus, we obtain a single condition

$$(F_K')^{-T}\,\mathbb{M}_K^{-1}\,(F_K')^{-1} = \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I, \quad \forall K \in \mathcal{T}_h$$

which directly combines the equidistribution and alignment conditions. From this, we can define a new functional as

$$I_h = \sum_{K\in\mathcal{T}_h} |K|\det(\mathbb{M}_K)^{\frac{1}{2}} \left\| (F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I \right\|_F^{2p}, \tag{2.17}$$

where σ_h is given in (2.8) and $\|\cdot\|_F$ is the Frobenius norm for matrices. Generally speaking, since we are working with $d \times d$ matrices, we can use any matrix norm and produce an equivalent form of the functional. We choose the Frobenius norm because it is convenient to compute. We remark that the weight, $|K|\det(\mathbb{M}_K)^{\frac{1}{2}}$, is chosen so that (2.17) is more comparable to (2.15) which includes the energy functional of a harmonic mapping as a special example. Moreover, this weight factor is used to emphasize the region where $\det(\mathbb{M})$ (error density) is large.

Minimizing (2.17) will then ensure that the mesh satisfies both the equidistribution and alignment conditions as closely as possible. Notice that this functional only contains one

parameter, p. In Section 2.3, it will be proven that this new functional has similar theoretical properties as the existing functional.

2.2 The moving mesh PDE and direct discretization

2.2.1 The moving mesh PDE solution strategy

In principle, we can directly minimize the two functionals (2.15) and (2.17) given in the last section, however, this direct minimization problem is too difficult due to their extreme nonlinearity. Instead, we will employ the moving mesh PDE (MMPDE) method [38] to find the minimizer. To be specific, we define the mesh equation as a modified gradient system of I_h, i.e.,

$$\frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T, \quad i = 1, \ldots, N_v, \tag{2.18}$$

where $\partial I_h/\partial \boldsymbol{x}_i$ is considered as a row vector, P_i is a positive scalar function used to make the equation have invariance properties, and $\tau > 0$ is a constant parameter used to adjust the time scale of mesh movement. It is interesting to notice that integrating (2.18) is equivalent to solving the minimization problem using the fastest descent method.

We consider functionals in a general form

$$I_h = \sum_{K\in\mathcal{T}_h} |K| G\left((F_K')^{-1}, \det\left(F_K'\right)^{-1}, \mathbb{M}_K\right), \tag{2.19}$$

where $G = G(\mathbb{J}, \det(\mathbb{J}), \mathbb{M})$ is a smooth function of three arguments,

$$\mathbb{J} = (F_K')^{-1} = \hat{E}E_K^{-1}, \qquad \det(\mathbb{J}) = \det(F_K')^{-1} = \frac{\det(\hat{E})}{\det(E_K)}, \qquad \mathbb{M} = \mathbb{M}_K.$$

For the existing functional (2.15), we have

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

Moreover, for the new functional (2.17), we have

$$G\left(\mathbb{J}, \det(\mathbb{J}), \mathbb{M}\right) = \sqrt{\det(\mathbb{M})} \left\| \mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I \right\|_F^{2p}. \tag{2.20}$$

Notice that G is a function of the physical nodes $\boldsymbol{x}_1^K, \cdots, \boldsymbol{x}_{d+1}^K$, that is

$$I_K\left(\boldsymbol{x}_1^K, \cdots, \boldsymbol{x}_{d+1}^K\right) = G\left(\left(F_K'\right)^{-1}, \det\left(F_K'\right)^{-1}, \mathbb{M}_K\right) \tag{2.21}$$

which gives

$$I_h\left(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{d+1}\right) = \sum_{K \in \mathcal{T}_h} |K| I_K\left(\boldsymbol{x}_1^K, \cdots, \boldsymbol{x}_{N_v}^K\right). \tag{2.22}$$

One of the keys to the MMPDE approach is to find the derivatives of I_h with respect to $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_v}$. In order to do so, we first obtain and assemble the elementwise derivatives of I_K with respect to $\boldsymbol{x}_1^K, \cdots, \boldsymbol{x}_{d+1}^K$ which requires scalar-by-matrix differentiation. To this end, let us first recall some notation and results for scalar-by-matrix derivatives.

2.2.2 Scalar-by-matrix derivatives

Let $f = f(A)$ be a scalar function of a matrix $A \in \mathbb{R}^{m \times n}$. Then the scalar-by-matrix derivative of f with respect to A is defined as

$$\frac{\partial f}{\partial A} = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{m1}} \\ \vdots & & \vdots \\ \frac{\partial f}{\partial A_{1n}} & \cdots & \frac{\partial f}{\partial A_{nm}} \end{bmatrix}_{n \times m} \quad \text{or} \quad \left(\frac{\partial f}{\partial A}\right)_{i,j} = \frac{\partial f}{\partial A_{j,i}}. \tag{2.23}$$

With this we can then define the chain rule with respect to the real parameter t as

$$\frac{\partial f}{\partial t} = \sum_{ij} \frac{\partial f}{\partial A_{j,i}} \frac{\partial A_{j,i}}{\partial t} = \sum_{ij} \left(\frac{\partial f}{\partial A}\right)_{i,j} \frac{\partial A_{j,i}}{\partial t} = \operatorname{tr}\left(\frac{\partial f}{\partial A} \frac{\partial A}{\partial t}\right). \tag{2.24}$$

We will use the following four lemmas throughout the applications of scalar-by-matrix differentiation. It should be noted that the first two lemmas can be verified directly, the third can be proven using the determinant expansion by minors, and the forth by differentiating $AA^{-1} = I$.

Lemma 2.2.1.

$$tr\left(A^T\right) = tr(A)\,,$$
$$tr(AB) = tr(BA)\,,$$
$$tr(ABC) = tr(CAB) = tr(BCA)\,.$$

Lemma 2.2.2.

$$\frac{\partial tr(A)}{\partial A} = I.$$

Lemma 2.2.3.

$$\frac{\partial \det(A)}{\partial A} = \det(A)A^{-1}.$$

Lemma 2.2.4.

$$\frac{\partial A^{-1}}{\partial t} = -A^{-1}\frac{\partial A}{\partial t}A^{-1}.$$

With these four lemmas, we can derive a number of identities which we will use in our application.

Corollary 2.2.1. *Assume $\mathbb{M}$ is independent of A. Then*

$$\frac{\partial tr(A\mathbb{M}A^T)}{\partial A} = 2\mathbb{M}A^T, \tag{2.25}$$
$$\frac{\partial tr(A^{-T}\mathbb{M}^{-1}A^{-1})}{\partial A} = -2A^{-1}A^{-T}\mathbb{M}^{-1}A^{-1}. \tag{2.26}$$

Moreover, assume A is independent of $\mathbb{M}$. Then

$$\frac{\partial tr(A\mathbb{M}A^T)}{\partial \mathbb{M}} = A^T A, \tag{2.27}$$

$$\frac{\partial tr(A\mathbb{M}^{-1}A^T)}{\partial \mathbb{M}} = -\mathbb{M}^{-1}A^T A\mathbb{M}^{-1}. \tag{2.28}$$

Proof. See Appendix A.1 for details. □

The above results give rise to expressions for $\frac{\partial G}{\partial \mathbb{J}}$ and $\frac{\partial G}{\partial r}$ which, as we have mentioned, are needed for the derivative computation of the discrete functional (2.19).

For the existing functional, recall that

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

Then the derivatives of G are given by

$$\begin{cases} \dfrac{\partial G}{\partial \mathbb{J}} &= dp\theta\sqrt{\det(\mathbb{M})}\left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}-1}\mathbb{M}^{-1}\mathbb{J}^T, \\ \dfrac{\partial G}{\partial \det(\mathbb{J})} &= p(1-2\theta)d^{\frac{dp}{2}}\det(\mathbb{M})^{\frac{1-p}{2}}\det(\mathbb{J})^{p-1}, \\ \dfrac{\partial G}{\partial \mathbb{M}} &= -\frac{\theta dp}{2}\sqrt{\det(\mathbb{M})}\left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}-1}\mathbb{M}^{-1}\mathbb{J}^T\mathbb{J}\mathbb{M}^{-1} \\ &\quad + \frac{\theta}{2}\sqrt{\det(\mathbb{M})}\left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}}\mathbb{M}^{-1} \\ &\quad + \frac{(1-2\theta)(1-p)d^{\frac{dp}{2}}}{2}\sqrt{\det(\mathbb{M})}\left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M}}}\right)^p\mathbb{M}^{-1}. \end{cases} \tag{2.29}$$

See Appendix A.2 for details.

Consider now the new functional (2.17) where

$$G\left(\mathbb{J}, \det(\mathbb{J}), \mathbb{M}\right) = \sqrt{\det(\mathbb{M})}\left\|\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I\right\|_F^{2p}$$

as given in (2.20). Denoting $A = \left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I\right)$ we have

$$G = \sqrt{\det(\mathbb{M})}\ \|A\|_F^{2p} = \sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^p. \tag{2.30}$$

Then

$$\begin{aligned}
\frac{\partial G}{\partial t} &= \frac{\partial\left(\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^p\right)}{\partial t} \\
&= p\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^{p-1}\ \mathrm{tr}\left(\frac{\partial\ \mathrm{tr}\left(AA^T\right)}{\partial AA^T}\frac{\partial AA^T}{\partial t}\right) \\
&= p\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^{p-1}\mathrm{tr}\left(\frac{\partial A}{\partial t}A^T + A\frac{\partial A^T}{\partial t}\right) \\
&= p\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^{p-1}\mathrm{tr}\left(2A^T\frac{\partial A}{\partial t}\right).
\end{aligned} \tag{2.31}$$

From the definition of A, we have

$$\begin{aligned}
\mathrm{tr}\left(2A^T\frac{\partial A}{\partial t}\right) &= \mathrm{tr}\left(2\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I\right)\left(\frac{\partial\mathbb{J}}{\partial t}\mathbb{M}^{-1}\mathbb{J}^T + \mathbb{J}\mathbb{M}^{-1}\frac{\partial\mathbb{J}^T}{\partial t}\right)\right) \\
&= \mathrm{tr}\left(4\mathbb{M}^{-1}\mathbb{J}^T\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I\right)\frac{\partial\mathbb{J}}{\partial t}\right).
\end{aligned} \tag{2.32}$$

Combining (2.31) and (2.32) we obtain

$$\frac{\partial G}{\partial\mathbb{J}} = 4p\left\|\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I\right\|_F^{2(p-1)}\sqrt{\det(\mathbb{M})}\mathbb{M}^{-1}\mathbb{J}^T\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}} I\right). \tag{2.33}$$

In a similar fashion, consider

$$\begin{aligned}
\frac{\partial G}{\partial t} &= \frac{\partial\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^p}{\partial t} \\
&= \|A\|_F^{2p}\ \frac{\partial\sqrt{\det(\mathbb{M})}}{\partial t} + \sqrt{\det(\mathbb{M})}\ \frac{\partial\mathrm{tr}\left(AA^T\right)^p}{\partial t}.
\end{aligned} \tag{2.34}$$

The first term of (2.34) is

$$\begin{aligned}\|A\|_F^{2p}\,\frac{\partial\sqrt{\det(\mathbb{M})}}{\partial t} &= \frac{1}{2}\,\|A\|_F^{2p}\,\det(\mathbb{M})^{-\frac{1}{2}}\,\frac{\partial\det(\mathbb{M})}{\partial t}\\ &= \frac{1}{2}\,\|A\|_F^{2p}\,\det(\mathbb{M})^{-\frac{1}{2}}\ \mathrm{tr}\left(\frac{\partial\det(\mathbb{M})}{\partial\mathbb{M}}\frac{\partial\mathbb{M}}{\partial t}\right)\\ &= \frac{1}{2}\,\|A\|_F^{2p}\,\det(\mathbb{M})^{-\frac{1}{2}}\ \mathrm{tr}\left(\det(\mathbb{M})\mathbb{M}^{-1}\frac{\partial\mathbb{M}}{\partial t}\right). \end{aligned} \tag{2.35}$$

Moreover, the second term of (2.34) is

$$\begin{aligned}\sqrt{\det(\mathbb{M})}\ \frac{\partial\ \mathrm{tr}\left(AA^T\right)^p}{\partial t} &= p\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^{p-1}\ \frac{\partial\ \mathrm{tr}\left(AA^T\right)}{\partial t}\\ &= p\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^{p-1}\mathrm{tr}\left(\frac{\partial\ \mathrm{tr}\left(AA^T\right)}{\partial AA^T}\frac{\partial AA^T}{\partial t}\right)\\ &= p\sqrt{\det(\mathbb{M})}\ \mathrm{tr}\left(AA^T\right)^{p-1}\mathrm{tr}\left(2A^T\frac{\partial A}{\partial t}\right), \end{aligned} \tag{2.36}$$

where, using the definition of A we get

$$\begin{aligned}\frac{\partial AA^T}{\partial t} &= \mathrm{tr}\left(2A^T\frac{\partial A}{\partial t}\right)\\ &= \mathrm{tr}\left(2\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T-\left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}}I\right)\mathbb{J}\frac{\partial\mathbb{M}^{-1}}{\partial t}\mathbb{J}^T\right)\\ &= -\mathrm{tr}\left(2\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T-\left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d}}I\right)\mathbb{J}\mathbb{M}^{-1}\frac{\partial\mathbb{M}}{\partial t}\mathbb{M}^{-1}\mathbb{J}^T\right)\\ &= -\frac{1}{2}\mathrm{tr}\left(\frac{\partial G}{\partial\mathbb{J}}\mathbb{J}\mathbb{M}^{-1}\frac{\partial\mathbb{M}}{\partial t}\right), \end{aligned} \tag{2.37}$$

with $\frac{\partial G}{\partial\mathbb{J}}$ is given as in (2.33). Thus (2.35), (2.36), and (2.37) give

$$\frac{\partial G}{\partial\mathbb{M}} = \frac{1}{2}G\mathbb{M}^{-1}-\frac{1}{2}\frac{\partial G}{\partial\mathbb{J}}\mathbb{J}\mathbb{M}^{-1}.$$

Finally, one can readily see that

$$\frac{\partial G}{\partial\det(\mathbb{J})} = 0$$

since G does not depend on $\det(\mathbb{J})$. In summary, the derivatives of G for the new functional are given by

$$\begin{cases} \dfrac{\partial G}{\partial \mathbb{J}} &= 4p \left\| \mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\dfrac{\sigma_h}{N}\right)^{-\frac{2}{d}} I \right\|_F^{2(p-1)} \sqrt{\det(\mathbb{M})}\, \mathbb{M}^{-1}\mathbb{J}^T \left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \left(\dfrac{\sigma_h}{N}\right)^{-\frac{2}{d}} I \right), \\ \dfrac{\partial G}{\partial \det(\mathbb{J})} &= 0, \\ \dfrac{\partial G}{\partial \mathbb{M}} &= \dfrac{1}{2} G \mathbb{M}^{-1} - \dfrac{1}{2} \dfrac{\partial G}{\partial \mathbb{J}} \mathbb{J}\mathbb{M}^{-1}. \end{cases} \tag{2.38}$$

Note that in the above derivation, we have viewed σ_h as a constant since $\sigma_h \sim \int_\Omega \det(\mathbb{M})^{\frac{1}{2}} d\boldsymbol{x}$.

2.2.3 Analytical formulas for derivatives of discretized functional

Using the formulation (2.19), we can rewrite the mesh equation (2.18) in a compact form as

$$\frac{d\boldsymbol{x}_i}{dt} = \frac{P_i}{\tau} \sum_{K \in \omega_i} |K| \boldsymbol{v}_{i_K}^K, \quad i = 1, ..., N_v \tag{2.39}$$

where ω_i is the patch of elements having $\boldsymbol{x}_i$ as one of their vertices and i_K and $\boldsymbol{v}_i^K$ are the local index and velocity of $\boldsymbol{x}_i$ on K, respectively. The local velocities are given by

$$\begin{bmatrix} (\boldsymbol{v}_1^K)^T \\ \vdots \\ (\boldsymbol{v}_d^K)^T \end{bmatrix} = -G E_K^{-1} + E_K^{-1} \frac{\partial G}{\partial \mathbb{J}} \hat{E} E_K^{-1} + \frac{\partial G}{\partial \det(\mathbb{J})} \frac{\det(\hat{E})}{\det(E_K)} E_K^{-1}$$
$$- \frac{1}{d+1} \sum_{j=0}^{d} \operatorname{tr}\left(\frac{\partial G}{\partial \mathbb{M}} \mathbb{M}_{j,K} \right) \begin{bmatrix} \frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}} \\ \vdots \\ \frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}} \end{bmatrix},$$

$$(\boldsymbol{v}_0^K)^T = -\sum_{k=1}^{d} (\boldsymbol{v}_k^K)^T - \sum_{j=0}^{d} \operatorname{tr}\left(\frac{\partial G}{\partial \mathbb{M}} \mathbb{M}_{j,K} \right) \frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}},$$

where $\mathbb{M}_{j,K} = \mathbb{M}(\boldsymbol{x}_j^K)$, $\phi_{j,K}$ is a linear basis function associated with $\boldsymbol{x}_j^K$, and $\partial\phi_{j,K}/\partial\boldsymbol{x}$ is the gradient of $\phi_{j,K}$ as a row vector. Note that in order to calculate the above velocities, we need

$$G, \quad \frac{\partial G}{\partial \mathbb{J}}, \quad \frac{\partial G}{\partial \det(\mathbb{J})}, \quad \frac{\partial G}{\partial \mathbb{M}},$$

where the derivatives for the existing and the new functional are given in (2.29) and (2.38), respectively. See [33] for details or Section 3.3.2 for a similar derivation.

It is remarked that the mesh equation (2.39) needs to be modified for boundary vertices. For example, for corner or fixed boundary vertices, the corresponding equation is replaced by

$$\frac{d\boldsymbol{x}_i}{dt} = 0.$$

For other boundary vertices, the velocity should be modified so that they only slide along the boundary curve (in 2D) or surface (in 3D) represented by $\Phi(\boldsymbol{x}) = 0$. That is, the mesh velocities $d\boldsymbol{x}_i/dt$ need to be modified so that the normal component along the curve or surface is zero, i.e.,

$$\nabla\Phi(\boldsymbol{x}_i) \cdot \frac{d\boldsymbol{x}_i}{dt} = 0.$$

With appropriate modifications for boundary vertices and for a given metric tensor $\mathbb{M}$, (2.39) can be integrated for an adaptive mesh. We use Matlab's *ode15s* (a variable-order ODE solver based on the numerical differentiation formulas) in our computation.

2.3 Theoretical analysis of the new functional

2.3.1 Coercivity

In the continuous case, ensuring the existence of a minimizer is closely related to coercivity and convexity of the meshing functional. Loosely speaking, coercivity ensures that the functional grows rapidly as the norm of the input tends to infinity whereas convexity provides a kind of compactness property. It is still unclear if this relation between the existence of a

minimizer and the coercivity and convexity of the functional holds true in the discrete case however, coercivity in the discrete case gives rise to a number of important properties.

Theorem 2.3.1. *The new functional (2.17) with $p > 1$ is coercive, i.e., there exist positive constants α and β such that the function G defined in (2.20) satisfies*

$$G \geq \alpha \left\| \mathbb{J} \right\|_F^{4p} - \beta. \tag{2.40}$$

Proof. For notational simplicity, we denote $\gamma_h = \left(\frac{\sigma_h}{N}\right)^{-2/d}$. From the triangle inequality and Hölder's inequality, we have

$$\begin{aligned}
\left\| \mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T - \gamma_h I \right\|_F^{2p} &\geq \left(\left\| \mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T \right\|_F - \left\| \gamma_h I \right\|_F \right)^{2p} \\
&\geq 2^{1-2p} \left\| \mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T \right\|_F^{2p} - \gamma_h^{2p} \| I \|_F^{2p} \\
&= 2^{1-2p} \left\| \mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T \right\|_F^{2p} - \left(\gamma_h^2 d \right)^p .
\end{aligned}$$

Notice that for a $d \times d$ matrix A, we know that $\|A\|_2 \leq \|A\|_F \leq \sqrt{d}\|A\|_2$. With this, it follows

$$\|\mathbb{J}\mathbb{M}^{-1}\mathbb{J}\|_F \geq \|\mathbb{J}\mathbb{M}^{-1}\mathbb{J}\|_2 \geq \frac{1}{\overline{m}} \|\mathbb{J}\mathbb{J}^T\|_2 = \frac{1}{\overline{m}} \|\mathbb{J}\|_2^2 \geq \frac{1}{\overline{m}d} \|\mathbb{J}\|_F^2.$$

Combining the above results, we get

$$G \geq \underline{m}^{\frac{d}{2}} \|\mathbb{J}\mathbb{M}^{-1}\mathbb{J} - \gamma_h I\|_F^{2p} \geq \frac{2^{1-2p} \underline{m}^{\frac{d}{2}}}{\overline{m}^{2p} d^{2p}} \|\mathbb{J}\|_F^{4p} - \underline{m}^{\frac{d}{2}} (\gamma_h^2 d)^p.$$

Thus, G satisfies (2.40) with $\alpha = \frac{2^{1-2p} \underline{m}^{\frac{d}{2}}}{\overline{m}^{2p} d^{2p}}$ and $\beta = \underline{m}^{\frac{d}{2}} (\gamma_h^2 d)^p$. □

Thus the new functional is coercive. Unfortunately, it is not convex. As a consequence, there is no guarantee that the minimizer of I_h is unique. It does, however, have other important properties that are discussed in detail next.

2.3.2 Nonsingularity of the mesh trajectory

Consider the semi-discrete MMPDE (2.18) with the new functional (2.17). For a given metric tensor $\mathbb{M}$, which is independent of t and satisfies (2.1), the MMPDE will generate a mesh trajectory $\mathcal{T}_h(t)$, $t > 0$ for any given nonsingular initial mesh. We denote the minimum altitude of K in the metric $\mathbb{M}_K$ by $a_{K,\mathbb{M}_K}$. Moreover, proper modifications of the boundary vertices are required in practical computations however, the theoretical analysis for the MMPDE with or without these modifications is similar. Therefore, for simplicity we only consider the case without boundary modifications.

Corollary 2.3.1. *For any $t > 0$, the elements of the mesh trajectory of the semi-discrete MMPDE (2.18) with the new functional (2.17) satisfy*

$$a_{K,\mathbb{M}_K} \geq C_1 \overline{m}^{-\frac{d}{2(4p-d)}} N^{-\frac{4p}{d(4p-d)}}, \tag{2.41}$$

$$|K| \geq C_2 \overline{m}^{-\frac{d^2}{2(4p-d)}-\frac{d}{2}} N^{-\frac{4p}{(4p-d)}}, \tag{2.42}$$

for all $K \in \mathcal{T}_h(t)$, where C_1 and C_2 are constants give by

$$C_1 = \left(\frac{2^{6p}\ d!^{\frac{4p}{d}}\ \alpha}{d^{4p}(d+1)^{4p-\frac{2p}{d}}\ (\beta|\Omega| + I_h\left(\mathcal{T}_h(0)\right))} \right)^{\frac{1}{4p-d}}, \quad C_2 = \frac{C_1^d}{d!}, \tag{2.43}$$

and α and β are defined in the proof of Theorem 2.3.1. Moreover, $\mathcal{T}_h(t)$ is nonsingular for all $t > 0$ if it is nonsingular initially.

Proof. This is a consequence of Theorem 4.1 in [32] which is stated for a general coercive functional. A direct application of this theorem with $q = 2p$ and Theorem 2.3.1 in the previous subsection gives the desired result. □

The key components in the proof of Theorem 4.1 in [32] are the coercivity of the functional

and the decreasing energy along the mesh trajectory of (2.18). The latter can be seen from

$$\frac{dI_h}{dt} = \sum_i \frac{\partial I_h}{\partial \boldsymbol{x}_i}\frac{d\boldsymbol{x}_i}{dt} = -\sum_i \frac{P_i}{\tau}\frac{\partial I_h}{\partial \boldsymbol{x}_i}\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T = -\sum_i \frac{P_i}{\tau}\left\|\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right\|^2 \leq 0.$$

It should be noted that in general, this property cannot be guaranteed.

The role of the parameter p can be explained to some extent from the inequality (2.41). Indeed, from (2.41) we have

$$a_{K,\mathbb{M}_K} \geq C_1 \overline{m}^{-\frac{d}{2(4p-d)}} N^{-\frac{4p}{d(4p-d)}} \to C_1 N^{-\frac{1}{d}}, \quad p \to \infty.$$

Noticing that $N^{-\frac{1}{d}}$ represents the average diameter of the elements, the above inequality implies that the mesh becomes more uniform as p gets larger.

One may notice that the bounds in (2.41) and (2.42) depend on N and $\overline{m}$. This is natural since the elements becomes smaller for larger N. Moreover, from the equidistribution condition (2.6), we can see that $|K| \sim \det(\mathbb{M}_K)^{-\frac{1}{2}}$, thus we can expect the lower bounds for the altitudes and volumes of the elements to become smaller as $\overline{m}$ gets larger.

Consider now the fully discrete case. Let t_n, $n = 0, 1, \ldots$ denote the time levels with $t_n \to \infty$ as $n \to \infty$. Assume that we have chosen a one-step integration scheme for (2.39) such that the energy is decreasing, i.e.,

$$I_h(\mathcal{T}_h^{n+1}) \leq I_h(\mathcal{T}_h^n). \tag{2.44}$$

Then, Corollary 2.3.1 will also hold for the mesh sequence, $\mathcal{T}_h^n$, $n = 0, 1, \ldots$. It should be noted that many schemes such as Euler's and the backward Euler satisfy (2.44) with a sufficiently small but not diminishing time step; e.g., see [26, 33].

2.3.3 Limits of the mesh trajectory

A direct application of Theorem 4.3 in [32], which is stated for a general coercive functional, and Theorem 2.3.1 in Section 2.3.1, gives the following corollary.

Corollary 2.3.2. *The mesh trajectory of the semi-discrete MMPDE (2.18) with the new functional (2.17) has the following properties.*

(a) $I_h(\mathcal{T}_h(t))$ *has a limit as* $t \to \infty$, *i.e.,*

$$\lim_{t\to\infty} I_h(\mathcal{T}_h) = L.$$

(b) The mesh trajectory has limiting meshes, all of which are non-singular and satisfy the bounds given in Corollary 2.3.1.

(c) The limiting meshes are critical points of I_h, *i.e., they satisfy*

$$\frac{\partial I_h}{\partial \boldsymbol{x}_i} = 0, \quad i = 1, \ldots, N_v$$

The result in Corollary 2.3.2 ensures that as time increases, the values of the functional for the mesh trajectory converge. This is significantly beneficial since it can be used as a computational stopping criteria. It should be noted that in general, there is no guarantee the mesh trajectory converges. In order to guarantee this convergence, stronger requirements need to be placed on either the descent in the functional value or on the meshing functional; e.g., see [33] for more details or Section 3.4.2 for a similar discussion. Moreover, like Corollary 2.3.1, Corollary 2.3.2 also holds for the fully discrete case provided that the time step is sufficiently small and the scheme satisfies the energy decreasing condition (2.44).

To conclude this section, we recall that the existing functional (2.15) is also coercive for $p > 1$ and $\theta \in (0, 1/2]$. Thus, Corollary 2.3.1 and Corollary 2.3.2 apply to the existing functional as well.

2.4 Numerical examples

Here we present numerical results for three examples in two dimensions to demonstrate the theoretical findings discussed in Section 2.3. Two of the main focuses will be showing the positive lower bound of the element volumes and the monotonically decreasing energy functional. Additionally, we will provide and compare meshes associated with the new and existing functionals. In order to asses the quality of the generated meshes, we compare the linear interpolation error (*error*, measured in the L^2 norm), and the equidistribution (Q_{eq}), alignment (Q_{ali}), and geometric (Q_{geo}) mesh quality measures which are defined as

$$Q_{eq} = \sqrt{\frac{1}{N}\sum_{K\in\mathcal{T}_c} Q_{eq,K}^2}, \quad Q_{ali} = \sqrt{\frac{1}{N}\sum_{K\in\mathcal{T}_h} Q_{ali,K}^2}, \quad Q_{geo} = \sqrt{\frac{1}{N}\sum_{K\in\mathcal{T}_h} Q_{geo,K}^2}, \tag{2.45}$$

where

$$Q_{eq,K} = \frac{|K|\det(\mathbb{M}_K)^{\frac{1}{2}}}{\sigma_h/N}, \quad Q_{ali,K} = \frac{\operatorname{tr}\left((F_K')^T\mathbb{M}_K F_K'\right)}{d\det\left((F_K')^T\mathbb{M}_K F_K'\right)^{\frac{1}{d}}}, \quad Q_{geo,K} = \frac{\operatorname{tr}\left((F_K')^T F_K'\right)}{d\det\left((F_K')^T F_K'\right)^{\frac{1}{d}}}. \tag{2.46}$$

The equidistribution and alignment measures are indications of how closely the mesh satisfies the equidistribution condition (2.6) and the alignment condition (2.7), respectively. The closer these quality measures are to 1, the closer they are to a uniform mesh with respect to the metric $\mathbb{M}$. The geometric measure is the same as the alignment quality measure taking $\mathbb{M} = I$. It measures how skew the mesh is in the Euclidean metric. It should be noted that these, in a sense, measure the average quality measure over all elements K in the physical mesh.

We use $p = 3/2$ and $\theta = 1/3$ in the existing functional (2.15) and $p = 1$ in the new functional (2.17). The defined parameters p and θ for the existing functional are commonly used and known to work well for most problems. The choice for p in the new functional is based on the desire to ensure that (2.17) is a quadratic function of matrix entries, which, computationally, makes the MMPDE less difficult to solve. The parameter τ in the MMPDE

(2.39) is taken to be $\tau = 10^{-2}$. Additionally, for the positive function P_i in (2.39) we use $P_i = \det(\mathbb{M})^{\frac{p-1}{2}}$ for the existing functional and $P_i = \det(\mathbb{M})^{\frac{2}{d}}$ for the new functional to ensure, for both cases, that the MMPDE (2.39) is invariant under the scaling transformation of $\mathbb{M}$, i.e., $\mathbb{M} : \mathbb{M} \to c\mathbb{M}$ for any positive constant c. The two dimensional meshes for Example 2.4.1 and Example 2.4.2 are constructed on the domain $\Omega = (0,1) \times (0,1)$. We take the metric tensor as

$$\mathbb{M}_K = \det(|H_K|)^{\frac{-1}{d+4}} |H_K|,$$

where H_K is the recovered Hessian using least squares fitting to the function values at the mesh vertices and $|H_K| = Q\text{diag}(|\lambda_1|, ..., |\lambda_d|)Q^T$, assuming that $Q\text{diag}(\lambda_1, ..., \lambda_d)Q^T$ is the eigen-decomposition of H_K. It is known [38] that the above form of the metric tensor is optimal corresponding to the L^2-norm of linear interpolation on triangular meshes.

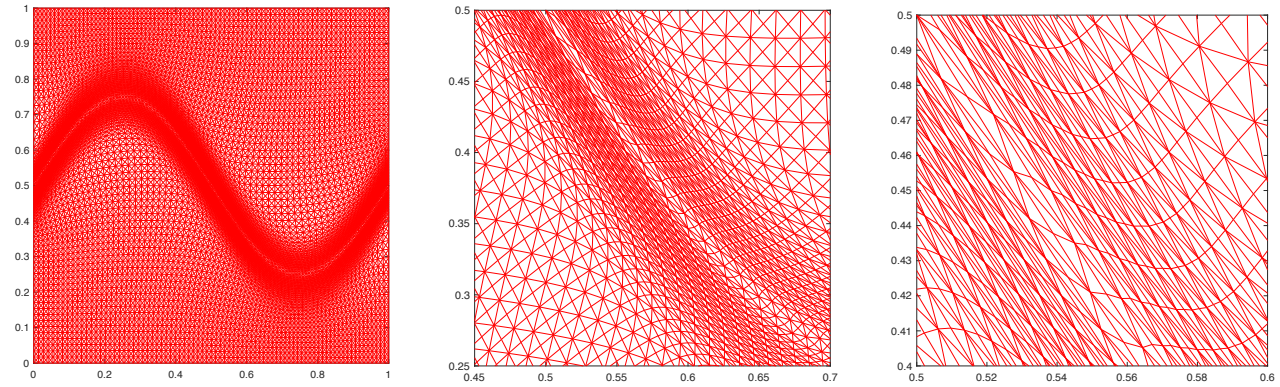
Example 2.4.1. In this example, we generate adaptive meshes for the sine wave modeled by

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

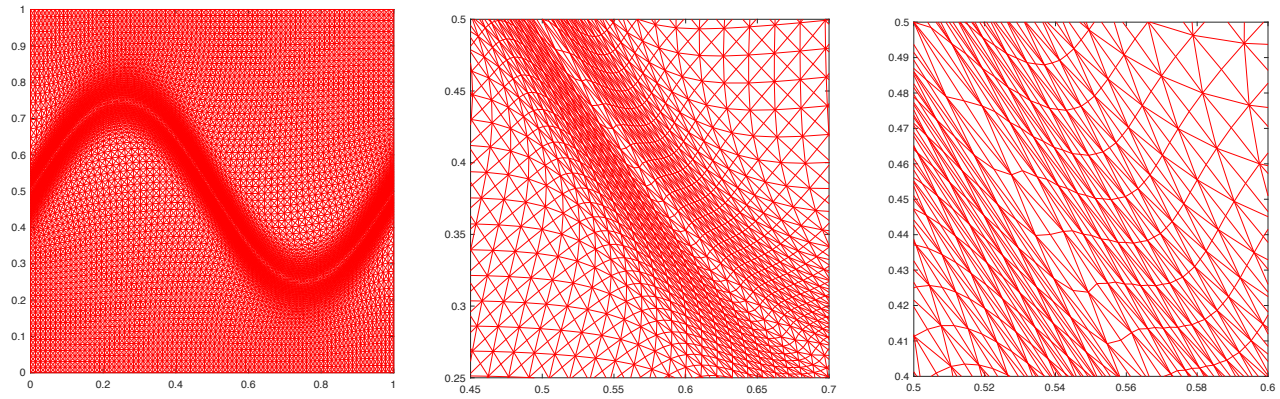
For the following results, we run to a final time of 5.0.

The example meshes and close-ups are given in Figure 2.1. The mesh associated with the new functional provides good shape and size adaptation. There is a high concentration of mesh elements in regions with large curvature near the interface. This is consistent with the fact that the metric tensor used is Hessian based. A closer look at the mesh shows that the elements are more skew (in the Euclidean metric) in the places with larger curvature. This is also shown in Table 2.1 with $Q_{geo} \approx 2$. On the other hand, Q_{ali} is close to 1, indicating that the mesh almost satisfies the alignment condition under the metric $\mathbb{M}$. Therefore, the mesh may seem skew in the Euclidean metric but is very regular in the metric $\mathbb{M}$.

While studying Table 2.1, we can also see that the equidistribution quality measure Q_{eq} for the new functional is close to 1, hence indicating that the mesh associated with the new functional is close to satisfying the equidistribution condition with respect to $\mathbb{M}$.



(a) New functional

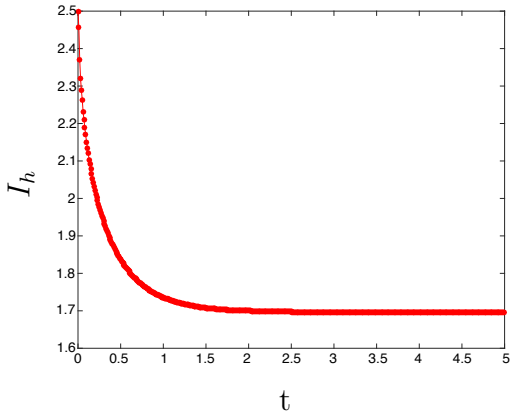


(b) Existing functional

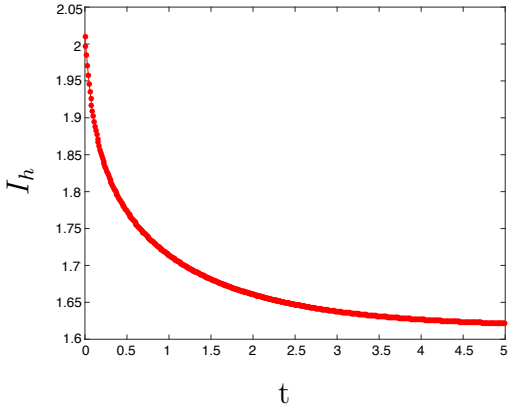
Figure 2.1: Example 2.4.1. Example meshes (left), close-ups near the inflection point (middle), and a closer version of the inflection point (right) with $N = 25600$.

Table 2.1: Mesh quality measures and the L^2 norm of linear interpolation error for Example 2.4.1.

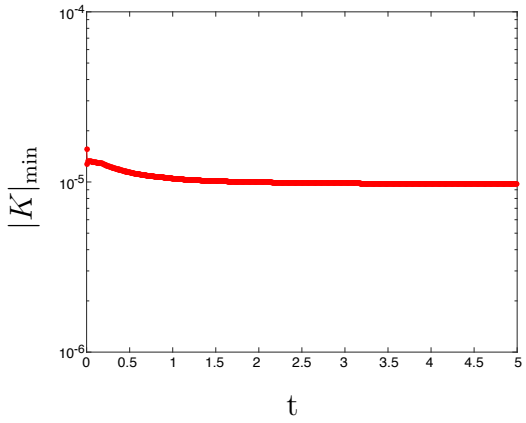
Functional	N	Q_{geo}	Q_{eq}	Q_{ali}	error
Existing	1600	1.684	1.065	1.041	5.563e-3
	6400	2.000	1.071	1.042	1.219e-3
	25600	1.986	1.081	1.039	3.038e-4
New	1600	1.593	1.088	1.028	6.077e-3
	6400	1.896	1.094	1.030	1.305e-3
	25600	2.019	1.091	1.030	3.138e-4



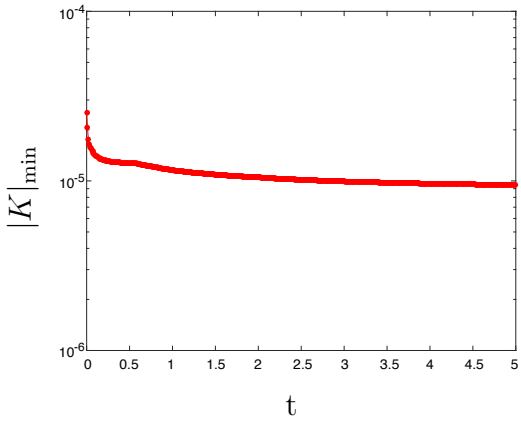
(a) New functional I_h



(b) Existing functional I_h



(c) New functional $|K|_{\min}$



(d) Existing functional $|K|_{\min}$

Figure 2.2: Example 2.4.1. The energy and minimum element volume are plotted as functions of t with $N = 25600$.

Therefore, with the alignment and equidistribution conditions close to being satisfied we can conclude that the mesh is almost uniform under the metric $\mathbb{M}$. The error value is a good indication that the mesh associated with the new functional is accurate. In this example, the error associated with the new functional is reasonably low. Moreover, as N increases, the numerical results show that the error decreases like $\mathcal{O}(N^{-1})$, a second-order convergence rate in terms of the average element diameter $\bar{h} = 1/\sqrt{N}$. This is consistent with the analysis of linear interpolation on triangle meshes.

As discussed in Section 2.3, theoretically we know that the I_h value is decreasing and $|K|$ is bonded below. To see these numerically, we plot I_h and $|K|_{min}$ as functions of t in Figure 2.2. The results are consistent with the theoretical predictions. Specifically, Figure 2.2(a) shows that I_h is decreasing while Figure 2.2(c) suggests that $|K|_{\min}$ is bounded below about 10^{-5}. It is interesting to observe that Figure 2.2(a) shows I_h decreasing faster at the beginning then leveling out more quickly when compared to the existing functional (Figure 2.2(b)). This shows that the energy is converging faster for the new functional than for the existing functional.

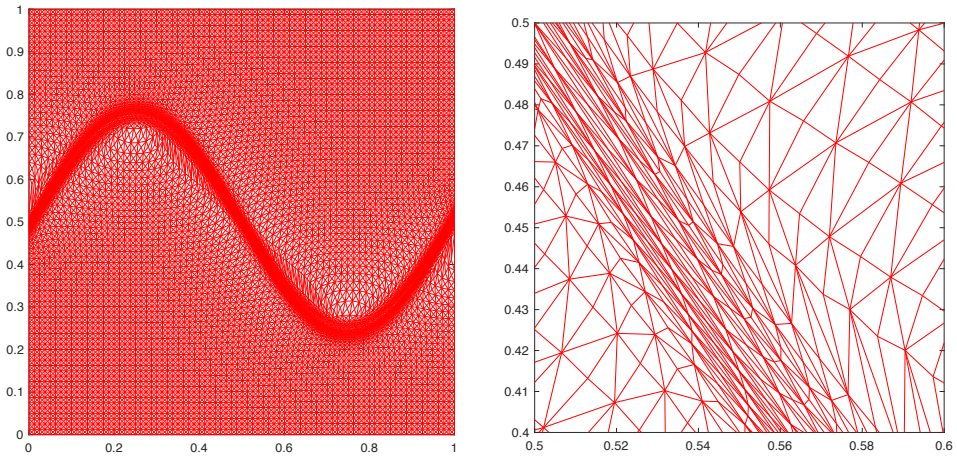
For comparison purposes, we also show the results obtained with the existing functional in Table 2.1, Figure 2.1, and Figure 2.2. From these, we see a high correlation. With respect to the mesh, both are very similar with high concentration near the interface where the function has large curvature. The quality measures Q_{geo}, Q_{eq}, and Q_{ali} are very similar as well. We further remark that the CPU time for both functionals are almost equivalent, differing at most by a few seconds. Hence, we can see that the two functionals are very comparable and both seem to work well in this example. To save space, we do not present numerical results comparing (2.15) and (2.17) with other meshing functionals. The interested reader is referred to [34] for additional numerical comparisons.

To show how both functionals perform in a more anisotropic example, we change the constant 30 in Example 2.4.1 to 100 and generate adaptive meshes. In this case, we run to a final time of 0.1. Figure 2.3 shows the meshes and close-ups. As we can see from

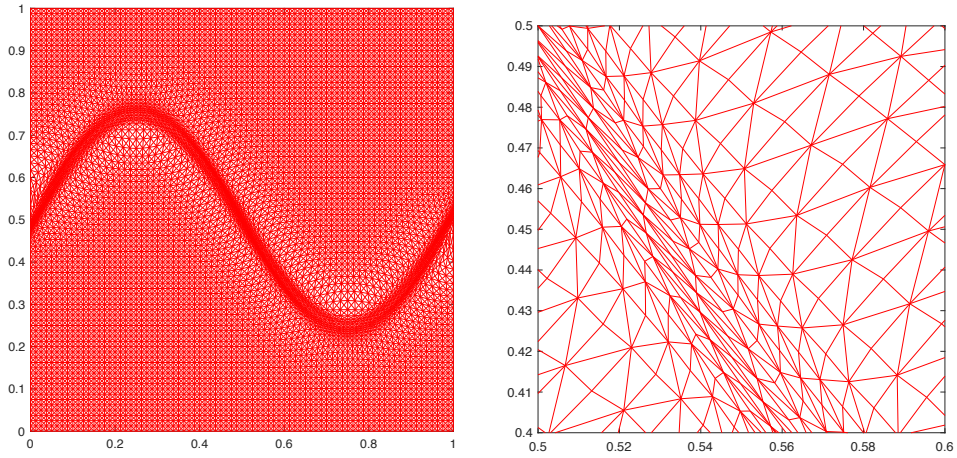
studying the meshes, the new functional provides a more adaptive mesh around the region with large curvature. That is, there is a higher concentration of mesh elements that are skew with respect to the Euclidean norm in this region. This is further confirmed by Table 2.2 where we see $Q_{geo} \approx 1.894$ for the new functional and $Q_{geo} \approx 1.279$ for the existing functional ($N = 25600$). It is also observed from Q_{eq} and Q_{ali} in Table 2.2 that the mesh associated with the new functional is slightly more uniform with respect to the metric tensor $\mathbb{M}$. Moreover, the interpolation error for the new functional is about half that of the existing functional for $N = 25600$. Overall, both functionals handle this more anisotropic example well and comparably. □

Table 2.2: Mesh quality measures and the L^2 norm of linear interpolation error for Example 2.4.1 with more anisotropic features.

Functional	N	Q_{geo}	Q_{eq}	Q_{ali}	error
Existing	1600	1.626	1.155	1.034	1.807e-2
	6400	1.548	1.312	1.058	3.942e-3
	25600	1.279	1.553	1.107	2.462e-3
New	1600	2.059	1.148	1.031	1.232e-2
	6400	2.203	1.249	1.028	2.616e-3
	25600	1.894	1.436	1.067	1.261e-3



(a) New functional



(b) Existing functional

Figure 2.3: Example 2.4.1 with more anisotropic features. Adaptive meshes (left) and close-ups near the inflection point (right) with $N = 25600$.

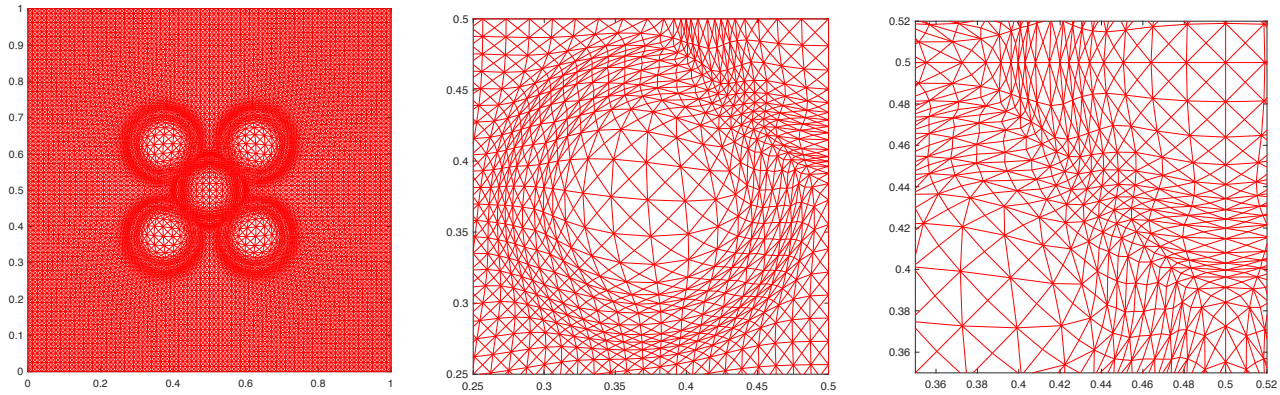
Example 2.4.2. In this example, we generate adaptive meshes for a five sphere figure modeled by

$$\begin{aligned} u(x,y) = &\tanh\left(30\left(X^2+Y^2-\frac{1}{8}\right)\right)+\tanh\left(30\left((X-0.5)^2+(Y-0.5)^2-\frac{1}{8}\right)\right)\\ &+\tanh\left(30\left((X-0.5)^2+(Y+0.5)^2-\frac{1}{8}\right)\right)\\ &+\tanh\left(30\left((X+0.5)^2+(Y-0.5)^2-\frac{1}{8}\right)\right)\\ &+\tanh\left(30\left((X+0.5)^2+(Y+0.5)^2-\frac{1}{8}\right)\right), \end{aligned}$$

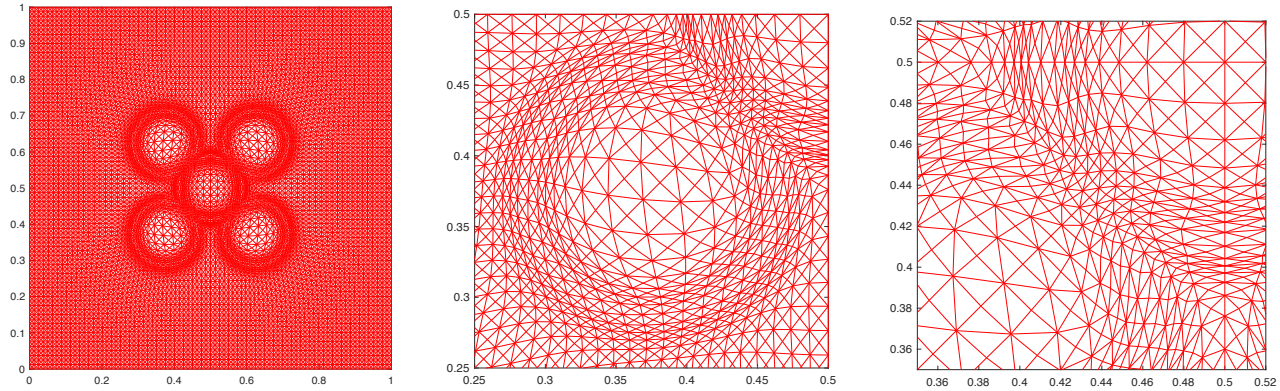
where $X = -2+4x$ and $Y = -2+4y$. We integrate the MMPDE to a final time of $t = 0.5$.

Figure 2.4 shows the meshes and close-ups of both functionals for this example. Studying the figure we see that the new functional provides a mesh with accurate shape and size adaptation. This can be further confirmed by the quality measures and the linear interpolation error given in Table 2.3. One may notice that the mesh has smaller values of Q_{geo} and thus is less skew than those in the previous example. This may be due to the fact that the function in this example is more isotropic than that in the previous example. Moreover, the linear interpolation error behaves like $\mathcal{O}(N^{-1})$, showing a second-order convergence rate.

Figure 2.5 shows the energy and minimum volume of the elements as functions of time. One can see that I_h is decreasing and converging faster for the new functional than for the existing functional, and that $|K|_{\min}$ is bounded by about 10^{-5}. Moreover, the results and performance of the new functional are similar to those with the existing functional. □



(a) New functional

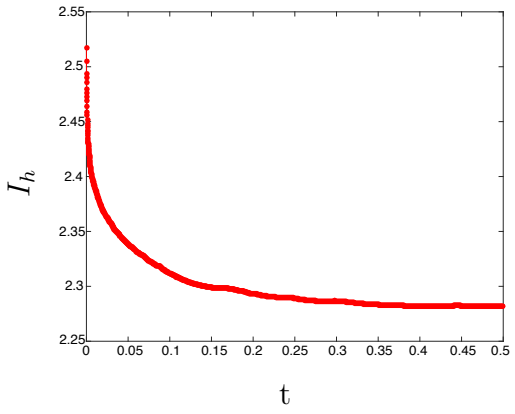


(a) Existing functional

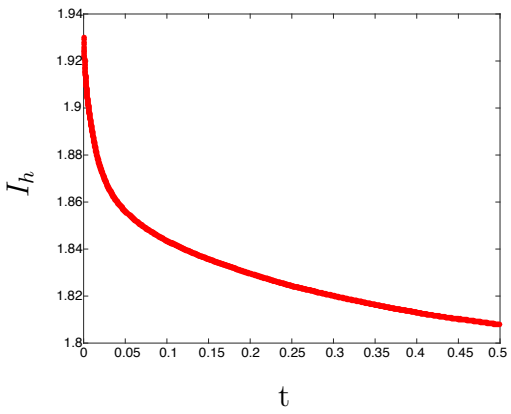
Figure 2.4: Example 2.4.2. Example meshes (left), close-ups near the circle meeting the boundary layer (middle), and a closer version of the circle meeting the boundary layer (right) with $N = 25600$.

Table 2.3: Mesh quality measures and the L^2 norm of linear interpolation error for Example 2.4.2.

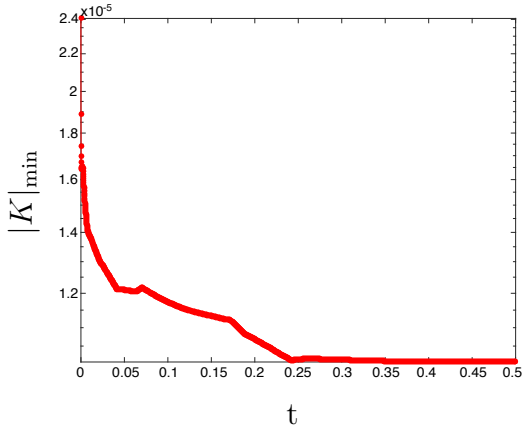
Functional	N	Q_{geo}	Q_{eq}	Q_{ali}	error
Existing	1600	1.051	1.134	1.056	6.954e-2
	6400	1.094	1.231	1.057	1.326e-2
	25600	1.122	1.342	1.040	3.068e-3
New	1600	1.031	1.188	1.026	6.946e-2
	6400	1.076	1.300	1.030	1.794e-2
	25600	1.137	1.370	1.030	3.310e-3



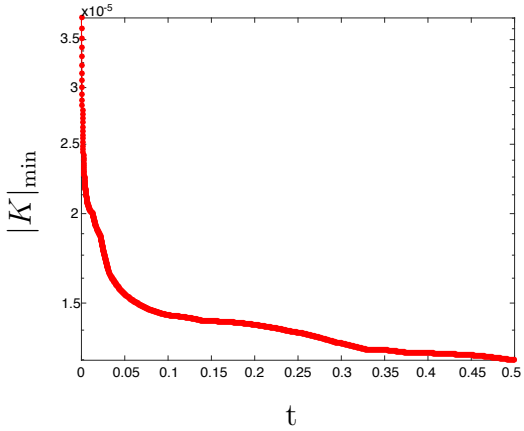
(a) New functional I_h



(b) Existing functional I_h



(c) New functional $|K|_{\min}$



(d) Existing functional $|K|_{\min}$

Figure 2.5: Example 2.4.2. The energy and minimum element volume are plotted as functions of t with $N = 25600$.

Example 2.4.3. In the final example, we solve the initial-boundary value problem of a special case of Burgers' equation

$$u_t = 10^{-3}\Delta u - uu_x - uu_y, \quad \text{in } \Omega = (-1,1)\times(-1,1)$$

subject to a homogeneous boundary condition and the initial condition

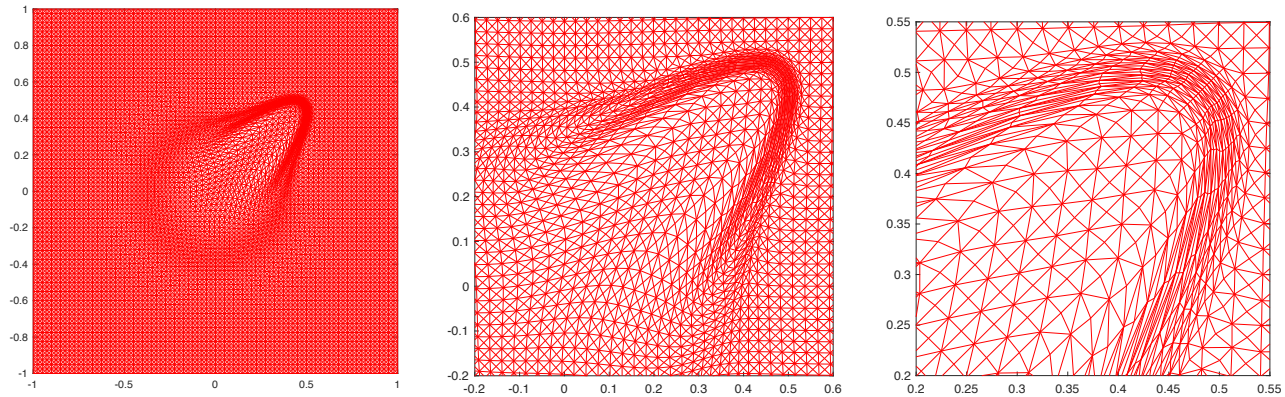
$$u(x,y,0) = e^{-36.8414(x^2+y^2)}, \quad \text{in } \Omega.$$

The partial differential equation is discretized in space using linear finite elements and in time using the fifth-order Radau IIA method [27]. It is solved with the mesh equation in an alternating manner [38]. For the following results, we start at $t = 0.25$ and run to a final time of $t = 1.25$.

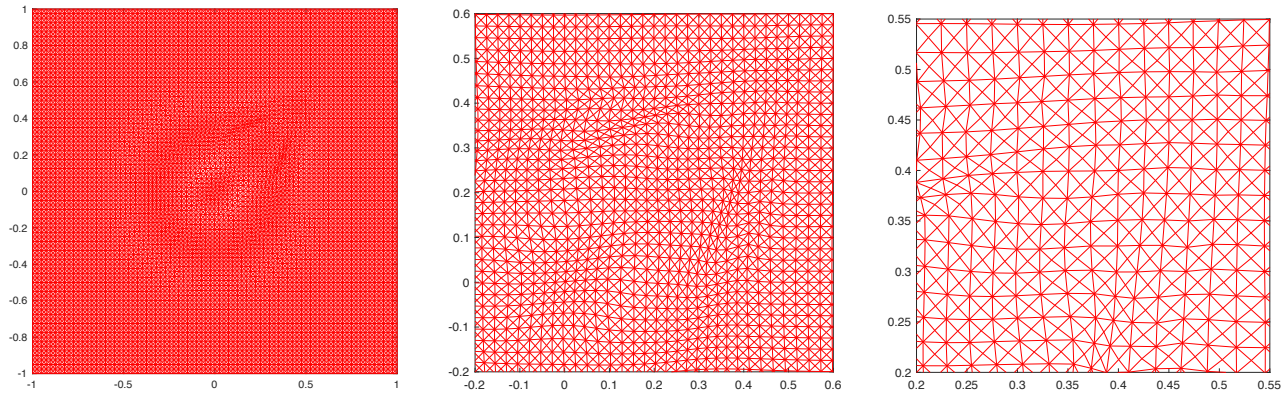
The meshes and close-ups for this example are given in Figure 2.6. Studying the figure we see that the new functional mesh is much more adaptive when compared to the existing functional mesh. The mesh associated with the new functional provides good shape and size adaptation. As seen in the close-ups, the concentration of mesh elements in the region with large curvature is high which, as we have seen in Example 2.4.1 and Example 2.4.2, is consistent with the Hessian based metric tensor. Moreover, the elements for the new functional are much more skew (with respect to the Euclidean metric) in the regions with larger curvature which is confirmed in Table 2.4 with $Q_{geo} \approx 17.01$. □

Table 2.4: Mesh quality measures for Example 2.4.3.

Functional	N	Q_{geo}	Q_{eq}	Q_{ali}
Existing	1600	1.502	5.696	1.842
	6400	1.934	14.20	2.391
	25600	1.677	31.77	3.426
New	1600	2.130	4.705	1.577
	6400	8.215	6.470	2.731
	25600	17.01	14.68	4.7111



(a) New functional



(b) Existing functional

Figure 2.6: Example 2.4.3. Example meshes (left), close-ups near the the tip (middle), and a closer version of the tip (right) with $N = 25600$.

2.5 Conclusion for the new functional

In the previous sections, we have introduced a new functional based on the equidistribution and alignment conditions. The functional is formulated by directly combining these two conditions into one with only a single parameter. It should be pointed out that (2.17) does not contain θ, a parameter that requires one to try to effectively balance the equidistribution and alignment conditions in (2.15). We have proven a number of theoretical results for this new functional at the discrete level which are similar to those of an existing functional that is also based on the equidistribution and alignment conditions but contains an additional parameter. For example, the new functional was proven to be coercive (Theorem 2.3.1). With this, it was then shown that the element altitude and volumes of the mesh trajectory of the discrete MMPDE associated with the new functional are bounded away from zero and the mesh trajectory stays nonsingular for all time if it is nonsingular initially (Corollary 2.3.1). Moreover, Corollary 2.3.2 states that the value of the meshing functional decreases monotonically along the mesh trajectory, while the latter has limiting meshes that are critical points of the meshing functional.

The numerical results shown in Section 2.4 demonstrated that the new functional produces correct mesh concentration and its performance is comparable to that of the existing functional which has been used successfully for various applications. In addition, the numerical results validated the theoretical properties of the new functional. It was shown that the meshing functional was monotonically decreasing and the minimum volume of the mesh element was bounded below as functions of time. From these results, we conclude that the new functional is similar to the existing functional in both numerical performance and theoretical properties.

Chapter 3

Surface Mesh

As mentioned in Chapter 1, mesh adaptation methods are well developed for bulk meshes however, methods on surfaces are limited. We are particularly interested in methods that adapt the mesh directly on a general surface with our without analytical expression. Here, we present such a method. The method can be viewed as a nontrivial extension of the MMPDE method that has been developed for bulk meshes discussed in Chapter 2 and demonstrated to work well for various applications; e.g. see [37, 38, 39]. It is emphasized that this method is developed directly on surface meshes, making no use of any information on surface parameterization.

The development starts with revealing the relation between the area of a surface element in the Euclidean or Riemannian metric and the the Jacobian matrix of the corresponding affine mapping. From this, we formulate two conditions that completely characterize a uniform mesh, the equidistribution and alignment conditions. These two conditions are then combined to establish a single meshing energy functional for which minimizing this functional will result in a mesh that closely satisfies these two conditions. In order to minimize the energy functional, we define the surface MMPDE equation as the gradient system of the energy functional in which the nodal mesh velocities are projected onto the underlying surface. Just as in the bulk mesh case, we obtain a compact, matrix form of the analytical expression for the mesh velocities making the implementation easy and robust.

A number of theoretical properties can then be proven. In particular, if the elements of the mesh trajectory of the surface MMPDE have positive areas initially then they will have positive areas for all time. This implies that there is no mesh tangling or closing. It is noted

that this MMPDE method utilizes surface normal vectors to ensure that the mesh vertices remain on the surface while moving. Thus, this new method can apply to general surfaces with or without explicit parameterization since the surface normal vectors can be computed even when the surface only has a numerical representation. A number of two- and three-dimensional examples are presented to verify the theoretical findings as well demonstrate the effectiveness of the method.

3.1 Equidistribution and alignment conditions for surface meshes

3.1.1 Area and affine mappings for surface elements

Let S be a bounded surface in $\mathbb{R}^d$ ($d \geq 2$). Assume that we have a mesh $\mathcal{T}_h = \{K\}$ on S and let N and N_v be the number of its elements and vertices, respectively. The elements K are surface simplexes in $\mathbb{R}^d$, i.e., they are $(d-1)$-dimensional simplexes in a d-dimensional space. Notice that their area in d dimensions is equivalent to their volume in $(d-1)$ dimensions. Assume that the reference element $\hat{K}$ has been chosen to be a $(d-1)$-dimensional equilateral and unitary simplex in a $(d-1)$-dimensional space. For $\hat{K}$ and any element $K \in \mathcal{T}_h$ let $F_K : \hat{K} \subset \mathbb{R}^{d-1} \to K \subset \mathbb{R}^d$ be the affine mapping between them and F_K' be its Jacobian matrix. Denote the vertices of K by $\boldsymbol{x}_j^K \in \mathbb{R}^d$, $j = 1, \ldots, d$ and the vertices of $\hat{K}$ by $\boldsymbol{\xi}_j \in \mathbb{R}^{d-1}$, $j = 1, \ldots, d$. Then

$$\boldsymbol{x}_j^K = F_K(\boldsymbol{\xi}_j), \quad j = 1, \ldots, d.$$

From this, we have

$$\boldsymbol{x}_j^K - \boldsymbol{x}_1^K = F_K' \left(\boldsymbol{\xi}_j - \boldsymbol{\xi}_1 \right), \quad j = 2, \ldots, d$$

or

$$\left[\boldsymbol{x}_2^K - \boldsymbol{x}_1^K, \ldots, \boldsymbol{x}_d^K - \boldsymbol{x}_1^K \right] = F_K' \left[\boldsymbol{\xi}_2 - \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d - \boldsymbol{\xi}_1 \right],$$

which gives $F'_K = E_K \hat{E}^{-1}$, where E_K and $\hat{E}$ are the edge matrices for K and $\hat{K}$, i.e.,

$$E_K = \left[\boldsymbol{x}_2^K - \boldsymbol{x}_1^K, \ldots, \boldsymbol{x}_d^K - \boldsymbol{x}_1^K\right], \quad \hat{E} = \left[\boldsymbol{\xi}_2 - \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_d - \boldsymbol{\xi}_1\right].$$

Notice that $\hat{E}$ is a $(d-1) \times (d-1)$ square matrix and its inverse exists since $\hat{K}$ is not degenerate. However, unlike the bulk mesh case, matrices $E_K, F'_K \in \mathbb{R}^{d\times(d-1)}$ are not square. This makes the formulation of adaptive mesh methods more difficult for surface than bulk meshes. Nevertheless, the approach is similar for both situations, as will be seen below.

In the following we can see that the area of the physical element $K \in \mathcal{T}_h$ can be determined using F'_K or E_K.

Lemma 3.1.1. *For any surface simplex K, there holds*

$$\frac{|K|}{|\hat{K}|} = \det\left(\left(F'_K\right)^T F'_K\right)^{1/2}, \tag{3.1}$$

where $|K|$ and $|\hat{K}|$ denote the area and the volume of the simplexes K and $\hat{K}$, respectively, and $\det(\cdot)$ *denotes the determinant of a matrix.*

Proof. From $F'_K = E_K \hat{E}^{-1}$, we have

$$\begin{aligned}
\det\left(\left(F'_K\right)^T F'_K\right)^{1/2} &= \det\left(\hat{E}^{-T} E_K^T E_K \hat{E}^{-1}\right)^{1/2} \\
&= \det\left(\hat{E}^{-T}\right)^{1/2} \det\left(E_K^T E_K\right)^{1/2} \det\left(\hat{E}^{-1}\right)^{1/2} \\
&= \det(\hat{E})^{-1} \det\left(E_K^T E_K\right)^{1/2} \\
&= \frac{1}{(d-1)!\,|\hat{K}|} \det\left(E_K^T E_K\right)^{1/2},
\end{aligned}$$

where we have used $|\hat{K}| = \frac{1}{(d-1)!}\det(\hat{E})$. Let the QR-decomposition of $E_K \in \mathbb{R}^{d\times(d-1)}$ be given by

$$E_K = Q_K \begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix},$$

where $Q_K \in \mathbb{R}^{d\times d}$ is a unitary matrix, $R_K \in \mathbb{R}^{(d-1)\times(d-1)}$ is an upper triangular matrix, and $\mathbf{0} \in \mathbb{R}^{1\times(d-1)}$ is a row vector of zeros. This decomposition indicates that K is formed by rotating the convex hull with edges formed by the column vectors of $\begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix}$. Thus, we have

$$|K| = \text{area}(E_K) = \text{area}\left(Q_K \begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix} \right) = \text{area}\left(\begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix} \right),$$

where we have used the fact that rotation, Q_K, does not change the area. Since the convex hull formed by the column vectors of $\begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix}$ lies on the $\boldsymbol{x}^{(1)} - \cdots - \boldsymbol{x}^{(d-1)}$ – plane, its area is equal to the $(d-1)$-dimensional volume of the convex hull formed by the column vectors of R_K in $(d-1)$-dimensions. Then,

$$|K| = \text{volume}_{(d-1)}(R_K) = \frac{1}{(d-1)!}\det(R_K) = \frac{1}{(d-1)!}\det(R_K^T R_K)^{1/2}.$$

On the other hand, we have

$$\det\left(E_K^T E_K\right) = \det\left(\begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix}^T Q_K^T Q_K \begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix} \right) = \det\left(\begin{bmatrix} R_K^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix} \right) = \det\left(R_K^T R_K\right).$$

Therefore,

$$\det\left(\left(F_K'\right)^T F_K'\right)^{1/2} = \frac{1}{(d-1)!\,|\hat{K}|}\det\left(E_K^T E_K\right)^{1/2} = \frac{1}{(d-1)!\,|\hat{K}|}\det\left(R_K^T R_K\right)^{1/2} = \frac{|K|}{|\hat{K}|}.$$

□

We now formulate the area of a surface element in a Riemannian metric using F_K' or E_K. The formula is similar to (3.1) and needed later in the development of algorithms for mesh adaptation. For this formulation we first consider a symmetric, uniformly positive

definite metric tensor $\mathbb{M}_K$ defined in (2.3). Recall (2.4) and the discussion thereafter, i.e., the geometric properties of K in the metric $\mathbb{M}_K$ can be obtained from those of $\mathbb{M}_K^{1/2}K$ in the Euclidean metric. This leads to the following lemma.

Lemma 3.1.2. *For any surface simplex K, there holds*

$$\frac{|K|_{\mathbb{M}_K}}{|\hat{K}|} = \det\left((F_K')^T \mathbb{M}_K F_K'\right)^{1/2}, \tag{3.2}$$

where $|K|_{\mathbb{M}_K}$ denotes the area of K in the metric $\mathbb{M}_K$.

Proof. The Jacobian matrix of the affine mapping from $\hat{K}$ to $\mathbb{M}_K^{1/2}K$ is given by

$$F'_{\mathbb{M}_K,K} = \left(\mathbb{M}_K^{1/2}E_K\right)\hat{E}^{-1} = \mathbb{M}_K^{1/2}F_K'.$$

From the discussion following (2.4), we know that the area of K in the metric $\mathbb{M}_K$ is equal to the area of $\mathbb{M}_K^{1/2}K$ in the Euclidean metric. Thus, from Lemma 3.1.1 we have

$$\begin{aligned}\frac{|K|_{\mathbb{M}_K}}{|\hat{K}|} &= \frac{|\mathbb{M}_K^{1/2}K|}{|\hat{K}|} = \det\left(\left(F'_{\mathbb{M}_K,K}\right)^T F'_{\mathbb{M}_K,K}\right)^{1/2} \\ &= \det\left(\left(\mathbb{M}_K^{1/2}F_K'\right)^T \mathbb{M}_K^{1/2}F_K'\right)^{1/2} = \det\left((F_K')^T \mathbb{M}_K F_K'\right)^{1/2}.\end{aligned}$$

□

The following lemma gives a lower bound for the area of K with respect to the metric $\mathbb{M}_K$ in terms of the minimum altitude of K with respect to $\mathbb{M}_K$.

Lemma 3.1.3. *Let $a_{K,\mathbb{M}_K}$ denote the minimum altitude of K with respect to $\mathbb{M}_K$. Then,*

$$|K|_{\mathbb{M}_K} \geq \frac{1}{(d-1)^{\frac{d-1}{2}}(d-1)!}\, a_{K,\mathbb{M}_K}^{d-1}. \tag{3.3}$$

Proof. From Lemma 3.1.2 and $F'_K = E_K \hat{E}^{-1}$, we have

$$\begin{aligned}
|K|_{\mathbb{M}_K} &= |\hat{K}| \det\left((F'_K)^T \mathbb{M}_K F'_K \right)^{1/2} \\
&= \frac{|\hat{K}|}{\det(\hat{E})} \det\left(E_K^T \mathbb{M}_K E_K \right)^{1/2} \\
&= \frac{1}{(d-1)!} \det\left(\left(\mathbb{M}_K^{1/2} E_K \right)^T \left(\mathbb{M}_K^{1/2} E_K \right) \right)^{1/2}.
\end{aligned}$$

Let the QR-decomposition of $\mathbb{M}_K^{1/2} E_K$ be denoted as

$$\mathbb{M}_K^{1/2} E_K = Q_K \begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix},$$

where $Q_K \in \mathbb{R}^{d \times d}$ is a unitary matrix, $R_K \in \mathbb{R}^{d-1 \times d-1}$ is an upper triangular matrix, and $\mathbf{0}$ is a $(d-1)$-dimensional row vector of zeros. This gives

$$\begin{aligned}
|K|_{\mathbb{M}_K} &= \frac{1}{(d-1)!} \det\left(\left(\mathbb{M}_K^{1/2} E_K \right)^T \left(\mathbb{M}_K^{1/2} E_K \right) \right)^{1/2} \\
&= \frac{1}{(d-1)!} \det\left([R_K^T \ \mathbf{0}^T] Q_K^T Q_K \begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix} \right)^{1/2} \\
&= \frac{1}{(d-1)!} \det(R_K^T R_K)^{1/2} \\
&= \frac{1}{(d-1)!} \prod_{i=1}^{d-1} s_i,
\end{aligned}$$

where s_i, $i = 1, \ldots, d-1$ denote the singular values of R_K. By [3, Lemma 5.12] we have that

$$s_i \geq \frac{a_{R_K}}{\sqrt{d-1}},$$

where a_{R_K} denotes the minimum altitude of the simplex formed by the columns of R_K.

Combining these, we get

$$|K|_{\mathbb{M}_K} \geq \frac{1}{(d-1)^{\frac{d-1}{2}}(d-1)!} a_{R_K}^{d-1}.$$

Since Q_K is a rotational matrix, the minimum altitude of K with respect to the metric $\mathbb{M}_K$ is the same as the minimum altitude of the convex hull formed by the columns of R_K i.e., $a_{K,\mathbb{M}_K} = a_{R_K}$. Thus, we have obtained (3.3). □

The relationship given in the above lemma between the area and minimum height will be used in the proof of the nonsingularity for surface meshes in Section 3.4. It is instructional to note that in two dimensions ($d = 2$), K is a line segement and both $|K|_{\mathbb{M}_K}$ and $a_{K,\mathbb{M}_K}$ represent the length of K in the metric $\mathbb{M}_K$ and are equal. In this case, the inequality (3.3) reduces to

$$|K|_{\mathbb{M}_K} \geq a_{K,\mathbb{M}_K},$$

which is very sharp. For $d = 3$, (3.3) becomes

$$|K|_{\mathbb{M}_K} \geq \frac{1}{4}\, a_{K,\mathbb{M}_K}^2,$$

which is not as sharp as in two dimensions. Indeed, when K is equilateral with respect to $\mathbb{M}_K$, we have [20]

$$|K|_{\mathbb{M}_K} = \frac{1}{\sqrt{3}}\, a_{K,\mathbb{M}_K}^2.$$

3.1.2 Equidistribution and alignment conditions for surface meshes

We can now define the equidistribution and alignment conditions characterizing a general nonuniform, simplicial surface mesh. As in the bulk mesh case, notice that any nonuniform mesh can be viewed as a uniform one in some metric tensor. Specifically, a mesh is uniform in some metric if all of the elements in the mesh have the same size and are similar to a reference element with respect to that metric. In this point of view, the equidistribution condition requires that all of the elements in the mesh have the same size. Mathematically,

this can be expressed as

$$|K|_{\mathbb{M}_K} = \frac{\sigma_h}{N}, \quad \forall K \in \mathcal{T}_h \tag{3.4}$$

where, as before, $|\cdot|_{\mathbb{M}_K}$ denotes the area of the surface with respect to the metric $\mathbb{M}_K$ and $\sigma_h = \sum_{K\in\mathcal{T}_h} |K|_{\mathbb{M}_K}$. Using Lemma 3.1.2 and recalling $|\hat{K}| = 1$, we have

$$|K|_{\mathbb{M}_K} = \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{1/2}, \qquad \sigma_h = \sum_{K\in\mathcal{T}_h} \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{1/2}.$$

Thus, the equidistribution condition (3.4) becomes

$$\det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{1/2} = \frac{\sigma_h}{N}, \quad \forall K \in \mathcal{T}_h. \tag{3.5}$$

The alignment condition, on the other hand, requires that all of the elements $K \in \mathcal{T}_h$ be similar to the reference element $\hat{K}$. Notice that any element K is similar to $\hat{K}$ if and only if $F_K : \hat{K} \to K$ is composed by dilation, rotation, and translation, or equivalently, F_K' is composed by dilation and rotation. Mathematically, F_K' can therefore be expressed as

$$F_K' = \alpha U \begin{bmatrix} I \\ \mathbf{0} \end{bmatrix} V^T, \tag{3.6}$$

where α is a constant representing dilation and $U \in \mathbb{R}^{d\times d}$ and $V \in \mathbb{R}^{(d-1)\times(d-1)}$ are orthogonal matrices representing rotation. This gives

$$\left(F_K'\right)^T F_K' = \alpha^2 V \begin{bmatrix} I & \mathbf{0}^T \end{bmatrix} U^T U \begin{bmatrix} I \\ \mathbf{0} \end{bmatrix} V^T = \alpha^2 I.$$

One can readily verify that

$$(\alpha^2)^{d-1} = \det\left(\left(F_K'\right)^T F_K'\right) \tag{3.7}$$

which gives

$$(d-1)\alpha^2 = \mathrm{tr}\left(\left(F'_K\right)^T F'_K\right) \tag{3.8}$$

and therefore

$$\det\left(\left(F'_K\right)^T F'_K\right)^{\frac{1}{d-1}} = \frac{1}{d-1}\mathrm{tr}\left(\left(F'_K\right)^T F'_K\right). \tag{3.9}$$

Consider now $\mathbb{M}_K$. From (3.6) we have

$$\mathbb{M}_K^{1/2}\ F'_K = \alpha\ \mathbb{M}_K^{1/2}\ U \begin{bmatrix} I \\ \mathbf{0} \end{bmatrix} V^T.$$

and thus, from (3.9) and the discussion following (2.4), we have

$$\frac{1}{d-1}\mathrm{tr}\left(\left(F'_K\right)^T \mathbb{M}_K F'_K\right) = \det\left(\left(F'_K\right)^T \mathbb{M}_K F'_K\right)^{\frac{1}{d-1}}, \quad \forall K \in \mathcal{T}_h \tag{3.10}$$

which is referred to as the alignment condition. Together, (3.5) and (3.10) completely characterize a uniform surface mesh with respect to the metric $\mathbb{M}_K$.

3.2 Surface energy functional

With these two conditions, we can now formulate a meshing functional similar to (2.15) in the bulk mesh case which averages (3.5) and (3.10). To do so, we first consider the alignment condition (3.10) and note that an equivalent condition is

$$\frac{1}{d-1}\mathrm{tr}\left[\left(\left(F'_K\right)^T \mathbb{M}_K F'_K\right)^{-1}\right] = \det\left[\left(\left(F'_K\right)^T \mathbb{M}_K F'_K\right)^{-1}\right]^{\frac{1}{d-1}}.$$

Notice that the left- and right-hand sides are the arithmetic mean and the geometric mean of the eigenvalues of the matrix $\left(\left(F'_K\right)^T \mathbb{M}_K F'_K\right)^{-1}$, respectively. The inequality of arithmetic

and geometric means gives

$$\frac{1}{d-1}\mathrm{tr}\left[\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{-1}\right] \geq \det\left[\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{-1}\right]^{\frac{1}{d-1}}, \tag{3.11}$$

with equality if and only if all of the eigenvalues are equal. From (3.11), for any general mesh which does not necessarily satisfy (3.10), we have

$$\mathrm{tr}\left[\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{-1}\right]^{d-1} \geq (d-1)^{d-1}\det\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{-1},$$

and therefore

$$\mathrm{tr}\left[\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{-1}\right]^{\frac{p(d-1)}{2}} - (d-1)^{\frac{p(d-1)}{2}}\det\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{-\frac{p}{2}} \geq 0,$$

where $p > 0$ is a dimensionless parameter which has been added to agree with the equidistribution energy functional below. Then, we define the alignment energy functional as

$$\begin{aligned} I_{ali} = \sum_{K\in\mathcal{T}_h}|\hat{K}|\det\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{\frac{1}{2}}\mathrm{tr}\left[\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{-1}\right]^{\frac{p(d-1)}{2}} \\ - (d-1)^{\frac{p(d-1)}{2}}\sum_{K\in\mathcal{T}_h}|\hat{K}|\det\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{\frac{1-p}{2}}, \end{aligned} \tag{3.12}$$

whose minimization will result in a mesh that closely satisfies the alignment condition (3.10). One may notice that $|\hat{K}|\det\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{\frac{1}{2}} = |K|_{\mathbb{M}_K}$ has been added as a weight.

Similarly, we consider the equidistribution condition (3.5). From Hölder's inequality, for any $p > 1$ we have

$$\sum_{K\in\mathcal{T}_h}\frac{|K|_{\mathbb{M}_K}}{\sigma_h}\cdot\frac{1}{\det\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{1/2}} \leq \left(\sum_{K\in\mathcal{T}_h}\frac{|K|_{\mathbb{M}_K}}{\sigma_h}\cdot\frac{1}{\det\left(\left(F_K'\right)^T\mathbb{M}_K F_K'\right)^{p/2}}\right)^{\frac{1}{p}}, \tag{3.13}$$

with equality if and only if

$$\det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1/2} = \text{constant}, \qquad \forall\ K \in \mathcal{T}_h.$$

That is, minimizing the difference between the left-hand side and the right-hand side of (3.13) tends to make $\det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1/2}$ constant for all $K \in \mathcal{T}_h$. Noticing that the left-hand side of (3.13) is simply N/σ_h, we can rewrite this inequality as

$$\left(\frac{N}{\sigma_h}\right)^p \cdot \sigma_h \leq \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{\frac{1-p}{2}}.$$

We can consider σ_h constant since $\sigma_h \approx \int_S \det(\mathbb{M}(\boldsymbol{x}))^{1/2} d\boldsymbol{x}$ and hence it only weakly depends on the mesh. Therefore, we define the equidistribution energy functional as

$$I_{eq} = (d-1)^{\frac{p(d-1)}{2}} \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{\frac{1-p}{2}}, \tag{3.14}$$

whose minimization will result in a mesh that closely satisfies the equidistribution condition.

We now have two functionals, one for each of equidistribution and alignment. Our goal is to formulate a single meshing functional for which minimizing will result in a mesh that closely satisfies both conditions. One way to ensure this is to average (3.12) and (3.14), that is, define $I_h = \theta I_{ali} + (1-\theta) I_{eq}$ for $\theta \in [0,1]$. This leads to

$$\begin{aligned} I_h = \theta \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{\frac{1}{2}} \operatorname{tr}\left[\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1}\right]^{\frac{p(d-1)}{2}} \\ + (1-2\theta)(d-1)^{\frac{p(d-1)}{2}} \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{\frac{1-p}{2}}, \end{aligned} \tag{3.15}$$

where $p > 1$ and $\theta \in [0,1]$ are dimensionless parameters, with the latter balancing the equidistribution and alignment conditions for which full alignment is achieved when $\theta = 1$ and full equidistribution is achieved when $\theta = 0$.

It should be noted that we can also formulate a functional similar to (2.17) in the surface mesh case, i.e.,

$$I_h = \sum_{K \in \mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K' \right)^{\frac{1}{2}} \left\| \left(\left(F_K'\right)^T \mathbb{M}_K F_K' \right)^{-1} - \left(\frac{\sigma_h}{N} \right)^{-\frac{2}{d-1}} I \right\|_F^{2p}. \tag{3.16}$$

In the bulk mesh case the associated functionals to (3.15) and (3.16) are proven to be theoretically and numerically comparable hence, for simplicity, we only consider (3.15) in numerical examples. For details on the formulation and properties of (3.16) see Appendix A.3.

We can write (3.15) as

$$I_h = \sum_{K \in \mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K' \right)^{\frac{1}{2}} \tilde{G}_K, \tag{3.17}$$

where

$$\tilde{G}_K = \theta \ \mathrm{tr}\left[\left(\left(F_K'\right)^T \mathbb{M}_K F_K' \right)^{-1} \right]^{\frac{p(d-1)}{2}} + (1-2\theta)(d-1)^{\frac{p(d-1)}{2}} \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K' \right)^{-\frac{p}{2}}. \tag{3.18}$$

We remark that for $0 < \theta \leq \frac{1}{2}$ and $p > 1$, $\tilde{G}_K$ is coercive, i.e.,

$$\tilde{G}\left(\mathbb{J}, \det\left(\mathbb{J}\right), \boldsymbol{x}\right) \geq \alpha \ \left(\mathrm{tr}\left[\left(\left(F_K'\right)^T \mathbb{M}_K F_K' \right)^{-1} \right] \right)^q - \beta, \qquad \forall \boldsymbol{x} \in S \tag{3.19}$$

where $q > (d-1)/2$, $\alpha > 0$, and $\beta \geq 0$ are constants. More specifically, for the surface functional (3.18) we have $\alpha = \theta$, $\beta = 0$, and $q = \frac{p(d-1)}{2}$. As we will see in Section 3.4, coercivity is an important property when proving mesh nonsingularity. It is also instructional to point out that the functional (3.15) is very similar to a Riemann sum of the meshing functional (2.15) for bulk meshes based on equidistribution and alignment. One of the main differences is that $\left(\left(F_K'\right)^T \mathbb{M}_K F_K' \right)$ cannot be simplified in (3.15) since it is not a square matrix as it is in the bulk mesh case. Additionally, the constant terms and exponents that contain d are $(d-1)$ in (3.15) instead of d in the bulk mesh case functional (2.15). The

functional (2.15) has been proven to work well for a variety of problems [38].

3.3 Surface moving mesh PDE

3.3.1 Gradient of meshing energy

Motivated by the functional (3.15), we consider meshing functionals in a general form (3.17), i.e.,

$$I_h = \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{\frac{1}{2}} \tilde{G}_K \equiv \sum_{K\in\mathcal{T}_h} G(\mathbb{J}_K, r_K), \tag{3.20}$$

where $\tilde{G}_K$ is a given smooth function of

$$\mathbb{J}_K = \left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1}, \quad r_K = \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1},$$

that is, $\tilde{G}_K = \tilde{G}(\mathbb{J}_K, r_K)$, and

$$G(\mathbb{J}_K, r_K) = |\hat{K}| r_K^{-\frac{1}{2}} \tilde{G}_K. \tag{3.21}$$

Indeed, a special example is (3.18) but $\tilde{G}_K$ can be chosen differently. Moreover, both $\mathbb{J}_K$ and r_K depend on the coordinates of the vertices of the physical element K and hence G is a function of them, i.e., $G\left(\mathbb{J}_K, r_K\right)$ can be expressed as

$$G\left(\mathbb{J}_K, r_K\right) = G_K\left(\boldsymbol{x}_1^K, \ldots, \boldsymbol{x}_d^K\right), \tag{3.22}$$

where $\boldsymbol{x}_i^K \in \mathbb{R}^d$ for $i = 1, \ldots, d$ are the coordinations of the vertices of K. As a consequence, the sum in (3.20) is a function of the coordinates of all vertices of the physical mesh $\mathcal{T}_h$, i.e.,

$$I_h\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}\right) = \sum_{K\in\mathcal{T}_h} G_K\left(\boldsymbol{x}_1^K, \ldots, \boldsymbol{x}_d^K\right), \tag{3.23}$$

where $\boldsymbol{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N_v$ are the coordinates of the vertices of the mesh with global indices. One of the underlying keys to our approach is to find the derivatives of I_h with respect to the physical coordinates $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_v}$ which requires elementwise derivatives of G_K with respect to $\boldsymbol{x}_1^K, \ldots, \boldsymbol{x}_d^K$. That is,

$$\frac{\partial I_h}{\partial \boldsymbol{x}_i} = \sum_{K \in \mathcal{T}_h} \frac{\partial G_K}{\partial \boldsymbol{x}_i} = \sum_{K \in \omega_i} \frac{\partial G_K}{\partial \boldsymbol{x}_{i_K}^K}, \quad i = 1, \ldots, N_v \tag{3.24}$$

where i_K denotes the local index of vertex $\boldsymbol{x}_i$ in K and ω_i is the element patch associated with $\boldsymbol{x}_i$.

Recall the properties of scalar-by-matrix differentiation in Section 2.2.2

$$\frac{\partial \mathrm{tr}\,(A)}{\partial A} = I, \qquad \frac{\partial A^{-1}}{\partial t} = -A^{-1}\frac{\partial A}{\partial t}A^{-1}, \quad \frac{\partial \det(A)}{\partial t} = \det(A)\ \mathrm{tr}\left(A^{-1}\frac{\partial A}{\partial t}\right) \tag{3.25}$$

where A is a square matrix. Using (3.25), we can find the expressions for $\frac{\partial G}{\partial \mathbb{J}}$ and $\frac{\partial G}{\partial r}$ which are needed to compute (3.24). For the functional (3.15), the first derivatives of G are given by

$$\begin{cases} \dfrac{\partial G}{\partial \mathbb{J}} &= \dfrac{\theta p(d-1)}{2}|\hat{K}|r^{-\frac{1}{2}}\mathrm{tr}(\mathbb{J})^{\frac{p(d-1)-2}{2}}I, \\ \\ \dfrac{\partial G}{\partial r} &= -\dfrac{\theta}{2}|\hat{K}|r^{-\frac{3}{2}}\mathrm{tr}(\mathbb{J})^{\frac{p(d-1)}{2}} + \dfrac{p-1}{2}(1-2\theta)(d-1)^{\frac{p(d-1)}{2}}|\hat{K}|r^{\frac{p-3}{2}}. \end{cases} \tag{3.26}$$

See Appendix A.4 for details.

3.3.2 Derivatives of the meshing functional with respect to the physical coordinates

From (3.24), we can see that we will need $\partial G_K/\partial \boldsymbol{x}_{i_K}^K$ to compute $\partial I_h/\partial \boldsymbol{x}_i$. The former can be obtained once we know the derivatives of G_K with respect to the coordinates of all

vertices of K, i.e.,

$$\frac{\partial G_K}{\partial[\boldsymbol{x}_1^K, \boldsymbol{x}_2^K, \ldots, \boldsymbol{x}_d^K]} = \begin{bmatrix} \dfrac{\partial G_K}{\partial \boldsymbol{x}_1^K} \\ \vdots \\ \dfrac{\partial G_K}{\partial \boldsymbol{x}_d^K} \end{bmatrix}.$$

To do so, let t be an entry of $[\boldsymbol{x}_1^K, \boldsymbol{x}_2^K, \ldots, \boldsymbol{x}_d^K]$. Using the chain rule we have

$$\frac{\partial G_K}{\partial t} = \operatorname{tr}\left(\frac{\partial G_K}{\partial E_K}\frac{\partial E_K}{\partial t}\right) + \operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\frac{\partial \mathbb{M}_K}{\partial t}\right).$$

Denote

$$\frac{\partial G_K}{\partial t}(I) = \operatorname{tr}\left(\frac{\partial G_K}{\partial E_K}\frac{\partial E_K}{\partial t}\right), \tag{3.27}$$

$$\frac{\partial G_K}{\partial t}(II) = \operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\frac{\partial \mathbb{M}_K}{\partial t}\right). \tag{3.28}$$

Now consider (3.27). When t is an entry of $[\boldsymbol{x}_2^K, \ldots, \boldsymbol{x}_d^K]$, recalling that $E_K = [\boldsymbol{x}_2^K - \boldsymbol{x}_1^K, \ldots, \boldsymbol{x}_d^K - \boldsymbol{x}_1^K]$, we have

$$\frac{\partial G_K}{\partial t}(I) = \operatorname{tr}\left(\frac{\partial G_K}{\partial E_K}\frac{\partial[\boldsymbol{x}_2^K, \ldots, \boldsymbol{x}_d^K]}{\partial t}\right),$$

which implies

$$\frac{\partial G_K}{\partial[\boldsymbol{x}_2^K, \ldots, \boldsymbol{x}_d^K]}(I) = \frac{\partial G_K}{\partial E_K}.$$

Moreover, for $t = \left(\boldsymbol{x}_1^K\right)^{(1)}$ (the first component of $\boldsymbol{x}_1^K$), we have

$$\frac{\partial G_K}{\partial \left(\boldsymbol{x}_1^K\right)^{(1)}}(I) = \operatorname{tr}\left(\frac{\partial G_K}{\partial E_K}\begin{bmatrix} -1 & -1 & \cdots & -1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}\right) = -\sum_i \left(\frac{\partial G_K}{\partial E_K}\right)_{i,1}.$$

We can obtain similar expressions for $\left(\boldsymbol{x}_1^K\right)^{(j)}$ for $j = 2, \ldots, d$. This gives

$$\frac{\partial G_K}{\partial \boldsymbol{x}_1^K}(I) = -\boldsymbol{e}^T \frac{\partial G_K}{\partial E_K}$$

where $\boldsymbol{e}^T = [1, \ldots, 1] \in \mathbb{R}^{1\times(d-1)}$. For (3.28), we assume that $\mathbb{M} = \mathbb{M}(\boldsymbol{x})$ is a piecewise linear function defined on the current mesh, i.e., $\mathbb{M} = \sum_{j=1}^{d} \mathbb{M}_{j,K}\phi_j^K$, where ϕ_j^K is the linear basis function associated with the vertex $\boldsymbol{x}_j^K$ for all $j = 1, \ldots, d$ and $\mathbb{M}_{j,K} = \mathbb{M}(\boldsymbol{x}_j^K)$. Denote the ith components of $\boldsymbol{x}$ and $\boldsymbol{x}_K$ by $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}_K^{(i)}$, respectively. Then, for any entry t of $[\boldsymbol{x}_1^K, \boldsymbol{x}_2^K, \ldots, \boldsymbol{x}_d^K]$, we have

$$\begin{aligned}
\frac{\partial G_K}{\partial t}(II) &= \operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\sum_{i=1}^{d}\frac{\partial \mathbb{M}_K}{\partial \boldsymbol{x}^{(i)}}\right)\frac{\partial \boldsymbol{x}_K^{(i)}}{\partial t} \\
&= \operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\sum_{i=1}^{d}\sum_{j=1}^{d}\mathbb{M}_{j,K}\frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}^{(i)}}\right)\frac{\partial \boldsymbol{x}_K^{(i)}}{\partial t} \\
&= \sum_{j=1}^{d}\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right)\frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}_K}{\partial t},
\end{aligned}$$

where we notice that $\partial\phi_{j,K}/\partial\boldsymbol{x}$ and $\partial\boldsymbol{x}_K/\partial t$ are a row and a column vector, respectively, and thus

$$\frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}_K}{\partial t}$$

is a dot product. From this and the identity $\boldsymbol{x}_K = (\boldsymbol{x}_1^K + \cdots + \boldsymbol{x}_d^K)/d$, we get

$$\frac{\partial G_K}{\partial [\boldsymbol{x}_2^K, \ldots, \boldsymbol{x}_d^K]}(II) = \frac{1}{d}\sum_{j=1}^{d}\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right)\begin{bmatrix}\dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}} \\ \vdots \\ \dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}\end{bmatrix}$$

and

$$\frac{\partial G_K}{\partial \boldsymbol{x}_1^K}(II) = \frac{1}{d}\sum_{j=1}^{d}\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right)\frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}.$$

Summarizing the above results, we have

$$\frac{\partial G_K}{\partial[\boldsymbol{x}_2^K,...,\boldsymbol{x}_d^K]} = \frac{\partial G_K}{\partial E_K} + \frac{1}{d}\sum_{j=1}^{d} \mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right) \begin{bmatrix} \dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}} \\ \vdots \\ \dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}} \end{bmatrix}, \tag{3.29}$$

$$\frac{\partial G_K}{\partial \boldsymbol{x}_1^K} = -\boldsymbol{e}^T \frac{\partial G_K}{\partial E_K} + \frac{1}{d}\sum_{j=1}^{d} \mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right) \frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}. \tag{3.30}$$

Notice that (3.30) can thus be rewritten as

$$\frac{\partial G_K}{\partial \boldsymbol{x}_1^K} = -\sum_{j=2}^{d} \frac{\partial G_K}{\partial \boldsymbol{x}_j^K} + \sum_{j=1}^{d} \mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right) \frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}. \tag{3.31}$$

Next, we establish the relations between

$$\frac{\partial G_K}{\partial E_K}, \quad \frac{\partial G_K}{\partial \mathbb{M}_K} \qquad \text{and} \qquad \frac{\partial G_K}{\partial \mathbb{J}}, \quad \frac{\partial G_K}{\partial r}.$$

First recall that $F_K' = E_K \hat{E}^{-1}$, thus

$$\mathbb{J} = \left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1} = \hat{E}\left(E_K^T \mathbb{M}_K E_K\right)^{-1} \hat{E}^T. \tag{3.32}$$

Let $E_K = E_K(t)$. Then we have

$$\begin{aligned} \frac{\partial G_K}{\partial t} &= \mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}} \frac{\partial \left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1}}{\partial t}\right) + \frac{\partial G_K}{\partial r} \frac{\partial \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1}}{\partial t} \\ &= \mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}} \hat{E} \frac{\partial \left(E_K^T \mathbb{M}_K E_K\right)^{-1}}{\partial t} \hat{E}^T\right) + \det(\hat{E})^2 \frac{\partial G_K}{\partial r} \frac{\partial \det\left(E_K^T \mathbb{M}_K E_K\right)^{-1}}{\partial t}. \end{aligned} \tag{3.33}$$

Consider the first term of (3.33). Using the properties of matrix derivatives (3.25) we get

$$\begin{aligned}
&\mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\frac{\partial\left(E_K^T\mathbb{M}_K E_K\right)^{-1}}{\partial t}\hat{E}^T\right)\\
&= -\mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\frac{\partial\left(E_K^T\mathbb{M}_K E_K\right)}{\partial t}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\right)\\
&= -\mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\left(\frac{\partial E_K^T}{\partial t}\mathbb{M}_K E_K + E_K^T\mathbb{M}_K\frac{\partial E_K}{\partial t}\right)\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\right).
\end{aligned}$$

Since $\frac{\partial G_K}{\partial \mathbb{J}}$, $\mathbb{M}_K$, and $\left(E_K^T\mathbb{M}_K E_K\right)^{-1}$ are all symmetric, it follows that

$$\begin{aligned}
&\mathrm{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\frac{\partial\left(E_K^T\mathbb{M}_K E_K\right)^{-1}}{\partial t}\hat{E}^T\right)\\
&\quad= -2\mathrm{tr}\left(\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}E_K^T\mathbb{M}_K\frac{\partial E_K}{\partial t}\right).
\end{aligned}$$

Consider now the second term of (3.33),

$$\begin{aligned}
&\det(\hat{E})^2\frac{\partial G_K}{\partial r}\frac{\partial\det\left(E_K^T\mathbb{M}_K E_K\right)^{-1}}{\partial t}\\
&= \frac{\det(\hat{E})^2}{\det\left(E_K^T\mathbb{M}_K E_K\right)}\frac{\partial G_K}{\partial r}\ \mathrm{tr}\left(\left(E_K^T\mathbb{M}_K E_K\right)\frac{\partial\left(E_K^T\mathbb{M}_K E_K\right)^{-1}}{\partial t}\right)\\
&= -\frac{\det(\hat{E})^2}{\det\left(E_K^T\mathbb{M}_K E_K\right)}\frac{\partial G_K}{\partial r}\ \mathrm{tr}\left(\frac{\partial\left(E_K^T\mathbb{M}_K E_K\right)}{\partial t}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\right)\\
&= -\frac{\det(\hat{E})^2}{\det\left(E_K^T\mathbb{M}_K E_K\right)}\frac{\partial G_K}{\partial r}\ \mathrm{tr}\left(\left(\frac{\partial E_K^T}{\partial t}\mathbb{M}_K E_K + E_K^T\mathbb{M}_K\frac{\partial E_K}{\partial t}\right)\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\right)\\
&= -2\frac{\det(\hat{E})^2}{\det\left(E_K^T\mathbb{M}_K E_K\right)}\frac{\partial G_K}{\partial r}\ \mathrm{tr}\left(\left(E_K^T\mathbb{M}_K E_K\right)^{-1}E_K^T\mathbb{M}_K\frac{\partial E_K}{\partial t}\right).
\end{aligned}$$

Therefore

$$\begin{aligned}
\frac{\partial G_K}{\partial E_K} &= -2\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}(E_K^T\mathbb{M}_K E_K)^{-1}E_K^T\mathbb{M}_K\\
&\quad - 2\frac{\det(\hat{E})^2}{\det\left(E_K^T\mathbb{M}_K E_K\right)}\frac{\partial G_K}{\partial r}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}E_K^T\mathbb{M}_K. \qquad (3.34)
\end{aligned}$$

Combining this with (3.30) we obtain

$$\begin{aligned}\begin{bmatrix}\dfrac{\partial G_K}{\partial \boldsymbol{x}_2^K}\\ \vdots \\ \dfrac{\partial G_K}{\partial \boldsymbol{x}_d^K}\end{bmatrix} =& -2\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}(E_K^T\mathbb{M}_K E_K)^{-1}E_K^T\mathbb{M}_K \\ &- 2\frac{\det(\hat{E})^2}{\det\left(E_K^T\mathbb{M}_K E_K\right)}\frac{\partial G_K}{\partial r}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}E_K^T\mathbb{M}_K \\ &+ \frac{1}{d}\sum_{j=1}^{d}\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right)\begin{bmatrix}\dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}\\ \vdots \\ \dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}\end{bmatrix}. \end{aligned} \tag{3.35}$$

We can compute $\frac{\partial G_K}{\partial \mathbb{M}_K}$ in a similar fashion. Let $\mathbb{M}_K = \mathbb{M}_K(t)$ and consider (3.33). Then, for the first term we have

$$\begin{aligned}&\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\frac{\partial\left(E_K^T\mathbb{M}_K E_K\right)^{-1}}{\partial t}\hat{E}^T\right) \\ &= -\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\frac{\partial\left(E_K^T\mathbb{M}_K E_K\right)}{\partial t}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\right) \\ &= -\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}E_K^T\frac{\partial \mathbb{M}_K}{\partial t}E_K\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\right) \\ &= -\operatorname{tr}\left(E_K\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}E_K^T\frac{\partial \mathbb{M}_K}{\partial t}\right).\end{aligned}$$

The second term of (3.33) is then

$$\begin{aligned}
&\det(\hat{E})^2 \frac{\partial G_K}{\partial r} \frac{\partial \det\left(E_K^T \mathbb{M}_K E_K\right)^{-1}}{\partial t} \\
&= \frac{\det(\hat{E})^2}{\det\left(E_K^T \mathbb{M}_K E_K\right)} \frac{\partial G_K}{\partial r} \operatorname{tr}\left(\left(E_K^T \mathbb{M}_K E_K\right) \frac{\partial \left(E_K^T \mathbb{M}_K E_K\right)^{-1}}{\partial t}\right) \\
&= -\frac{\det(\hat{E})^2}{\det\left(E_K^T \mathbb{M}_K E_K\right)} \frac{\partial G_K}{\partial r} \operatorname{tr}\left(\frac{\partial \left(E_K^T \mathbb{M}_K E_K\right)}{\partial t} \left(E_K^T \mathbb{M}_K E_K\right)^{-1}\right) \\
&= -\frac{\det(\hat{E})^2}{\det\left(E_K^T \mathbb{M}_K E_K\right)} \frac{\partial G_K}{\partial r} \operatorname{tr}\left(E_K^T \frac{\partial \mathbb{M}_K}{\partial t} E_K \left(E_K^T \mathbb{M}_K E_K\right)^{-1}\right) \\
&= -\frac{\det(\hat{E})^2}{\det\left(E_K^T \mathbb{M}_K E_K\right)} \frac{\partial G_K}{\partial r} \operatorname{tr}\left(E_K \left(E_K^T \mathbb{M}_K E_K\right)^{-1} E_K^T \frac{\partial \mathbb{M}_K}{\partial t}\right).
\end{aligned}$$

Therefore

$$\begin{aligned}
\frac{\partial G_K}{\partial \mathbb{M}_K} = &-E_K (E_K^T \mathbb{M}_K E_K)^{-1} \hat{E}^T \frac{\partial G_K}{\partial \mathbb{J}} \hat{E} (E_K^T \mathbb{M}_K E_K)^{-1} E_K^T \\
&- \frac{\det(\hat{E})^2}{\det\left(E_K^T \mathbb{M}_K E_K\right)} \frac{\partial G_K}{\partial r} E_K \left(E_K^T \mathbb{M}_K E_K\right)^{-1} E_K^T,
\end{aligned} \tag{3.36}$$

which we can use in the last term of (3.35).

Finally, we derive the relations between

$$\frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}, \quad j = 1, ..., d \qquad \text{and} \qquad E_K.$$

First, note that the basis functions satisfy

$$\sum_{i=1}^{d} \phi_{i,K} = 1 \qquad \text{and} \qquad \sum_{i=1}^{d} \boldsymbol{x}_i^K \phi_{i,K} = \boldsymbol{x}.$$

Eliminating $\boldsymbol{x}_1^K$ yields

$$\boldsymbol{x} - \boldsymbol{x}_1^K = \sum_{i=2}^{d} (\boldsymbol{x}_i^K - \boldsymbol{x}_1^K) \phi_{i,K}.$$

Then differentiating with respect to $\boldsymbol{x}^{(k)}$ gives

$$\boldsymbol{e}_k = \frac{\partial(\boldsymbol{x} - \boldsymbol{x}_1^K)}{\partial \boldsymbol{x}^{(k)}} = \frac{\partial}{\partial \boldsymbol{x}^{(k)}} \left(\sum_{i=2}^{d} (\boldsymbol{x}_i^K - \boldsymbol{x}_1^K) \phi_{i,K} \right) = \sum_{i=2}^{d} (\boldsymbol{x}_i^K - \boldsymbol{x}_1^K) \frac{\partial \phi_{i,K}}{\partial \boldsymbol{x}^{(k)}},$$

where $\boldsymbol{e}_k$ is the k^{th} unit vector in $\mathbb{R}^d$. Hence we have

$$I = E_K \begin{bmatrix} \dfrac{\partial \phi_{2,K}}{\partial \boldsymbol{x}} \\ \vdots \\ \dfrac{\partial \phi_{d,K}}{\partial \boldsymbol{x}} \end{bmatrix},$$

which gives

$$E_K^T E_K \begin{bmatrix} \dfrac{\partial \phi_{2,K}}{\partial \boldsymbol{x}} \\ \vdots \\ \dfrac{\partial \phi_{d,K}}{\partial \boldsymbol{x}} \end{bmatrix} = E_K^T$$

and thus

$$\begin{bmatrix} \dfrac{\partial \phi_{2,K}}{\partial \boldsymbol{x}} \\ \vdots \\ \dfrac{\partial \phi_{d,K}}{\partial \boldsymbol{x}} \end{bmatrix} = (E_K^T E_K)^{-1} E_K^T, \qquad \frac{\partial \phi_{1,K}}{\partial \boldsymbol{x}} = -\sum_{j=2}^{d} \frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}. \tag{3.37}$$

Summarizing the above we have

$$\begin{aligned}\begin{bmatrix}\dfrac{\partial G_K}{\partial \boldsymbol{x}_2^K}\\ \vdots \\ \dfrac{\partial G_K}{\partial \boldsymbol{x}_d^K}\end{bmatrix} =& -2\left(E_K^T\mathbb{M}_K E_K\right)^{-1}\hat{E}^T\frac{\partial G_K}{\partial \mathbb{J}}\hat{E}(E_K^T\mathbb{M}_K E_K)^{-1}E_K^T\mathbb{M}_K \\ & -2\frac{\det(\hat{E})^2}{\det\left(E_K^T\mathbb{M}_K E_K\right)}\frac{\partial G_K}{\partial r}\left(E_K^T\mathbb{M}_K E_K\right)^{-1}E_K^T\mathbb{M}_K \\ & +\frac{1}{d}\sum_{j=1}^{d}\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right)\begin{bmatrix}\dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}\\ \vdots \\ \dfrac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}\end{bmatrix}, \end{aligned} \tag{3.38}$$

$$\frac{\partial G_K}{\partial \boldsymbol{x}_1^K} = -\sum_{j=2}^{d}\frac{\partial G_K}{\partial \boldsymbol{x}_j^K}+\sum_{j=1}^{d}\operatorname{tr}\left(\frac{\partial G_K}{\partial \mathbb{M}_K}\mathbb{M}_{j,K}\right)\frac{\partial \phi_{j,K}}{\partial \boldsymbol{x}}, \tag{3.39}$$

where $\frac{\partial G_K}{\partial \mathbb{J}}$ and $\frac{\partial G_K}{\partial r}$ are given in (3.26), $\partial\phi_{j,K}/\partial\boldsymbol{x}$ for $j=1,\ldots,d$ are given in (3.37), and $\frac{\partial G_K}{\partial \mathbb{M}}$ is given in (3.36). Having computed $\partial G_K/\partial \boldsymbol{x}_j^K$ $(j=1,...,d)$ for all elements using (3.38) and (3.39), we can obtain $\partial I_h/\partial \boldsymbol{x}_i$ from (3.24).

3.3.3 Surface moving mesh equations

As mentioned above, we employ the MMPDE method to minimize the meshing functional (3.15) or a more general form (3.20). An MMPDE is a mesh equation that involves mesh speed. There are various formulations of MMPDEs; we focus here on the approach where the surface MMPDE is defined as a modified gradient system of the meshing functional. A distinct feature for surface meshes, other than bulk meshes, is that the nodes need to stay on the surface during movement. By Section 3.3.2 we may assume that we have the matrix

$$\frac{\partial I_h}{\partial \boldsymbol{x}_i}, \quad i=1,\cdots,N_v.$$

Let $\Phi(\boldsymbol{x}) = 0$ denote the surface, where Φ can be defined through an analytical expression or a numerical representation such as by spline functions. Then for the vertices to stay on the surface we should have $\Phi(\boldsymbol{x}_i) = 0$ for all $i = 1, \ldots, N_v$ or at least

$$\frac{d\boldsymbol{x}_i}{dt} \cdot \nabla\Phi(\boldsymbol{x}_i) = 0 \tag{3.40}$$

where $d\boldsymbol{x}_i/dt$ is the nodal mesh velocity. Following the MMPDE approach, we would define the mesh equation as the gradient system of I_h, i.e.,

$$\frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T, \quad i = 1, ..., N_v \tag{3.41}$$

where P_i is a positive scalar function used to make the equation have desired invariance properties and $\tau > 0$ is a constant parameter used for adjusting the time scale of mesh movement. Obviously, this does not satisfy (3.40). Here we propose to project the velocities in (3.41) onto the surface and define the surface moving mesh equation as

$$\frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left[\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T - \left(\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T \cdot \boldsymbol{n}_i\right)\boldsymbol{n}_i\right], \quad i = 1, ..., N_v \tag{3.42}$$

where $\boldsymbol{n}_i = \nabla\Phi(\boldsymbol{x}_i)/\|\nabla\Phi(\boldsymbol{x}_i)\|$ is the unit normal to the surface at $\boldsymbol{x}_i$ and the difference inside the square bracket is the projection of the vector $\partial I_h/\partial \boldsymbol{x}_i$ onto the tangential plane of the surface at $\boldsymbol{x}_i$. Notice that this surface MMPDE inherently ensures that (3.40) be satisfied or, in words, *the nodes stay on the surface during the mesh movement*. Moreover, it is important to note that (3.42) only utilizes the unit normal vectors of the surface whose computation does not require explicit parameterization or analytical expression of the surface. As mentioned, for surfaces without an analytical expression, spline functions may be used to approximate the gradient for (3.42). It should be noted that (3.40) is a weak imposition. A stabilized

version is given by

$$\frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left[\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T - \left(\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T \cdot \boldsymbol{n}_i\right)\boldsymbol{n}_i\right] - \frac{\Phi}{\delta\tau\,|\nabla\Phi|}\boldsymbol{n}_i, \tag{3.43}$$

where $\delta \geq 0$, and the last term, in a sense, determines the projection method of the nodes. That is, when $\delta = 0$ then $\Phi(\boldsymbol{x}_i(t)) = 0$ and hence, the nodes are directly projected to the surface. On the other hand, when $\delta = \infty$, (3.43) is equivalent to (3.42) and so the nodes are projected onto the tangential plane. For a direct formulation of (3.42) using Lagrange multipliers and a more detailed discussion of (3.43), see Appendix A.5.

Using (3.24) we can rewrite the above equation in a compact form as

$$\frac{d\boldsymbol{x}_i}{dt} = \frac{P_i}{\tau}\sum_{K\in\omega_i} \boldsymbol{v}_{i_K}^K, \quad i = 1, ..., N_v \tag{3.44}$$

where $\boldsymbol{v}_{i_K}^K \in \mathbb{R}^d$ is the local mesh velocities contributed by K to $\boldsymbol{x}_{i_K}^K$ and has the expressions

$$\boldsymbol{v}_{i_K}^K = -\left(\frac{\partial G_K}{\partial \boldsymbol{x}_{i_K}^K}\right)^T + \left(\left(\frac{\partial G_K}{\partial \boldsymbol{x}_{i_K}^K}\right)^T \cdot \boldsymbol{n}_{i_K}\right)\boldsymbol{n}_{i_K}, \tag{3.45}$$

and $\partial G_K/\partial \boldsymbol{x}_{i_K}^K$ is given in (3.38) and (3.39).

The surface MMPDE (3.44) must be modified properly for boundary vertices when S has a boundary. For fixed boundary vertices, the corresponding equation is replaced by

$$\frac{d\boldsymbol{x}_i}{dt} = 0.$$

The velocities for other boundary vertices should be modified such that they slide on the boundary which is defined on a case-by-case basis.

With proper modification of the boundary vertices, the system (3.44) can be integrated in time. To do so, one first starts by calculating the edge matrices E_K for all elements and $\hat{E}$ for the reference element. One can then readily calculate (3.26) which is needed for

(3.38) and (3.39). Then one can integrate (3.44) in time. For this work we use Matlab's ODE solvers ode45 and ode15s. The explicit scheme, ode45, implements a 4(5)-order Runge-Kutta method with a variable time step. The implicit scheme, ode15s, is a variable time step and variable-order solver based on the numerical differentiation formulas of orders 1 to 5. All of the numerical examples in this paper use ode45 although both ode45 and ode15s have been tested and proven to work very well in computation.

3.4 Theoretical Properties

3.4.1 Equivalent measure of minimum height

We begin the theoretical analysis by establishing the relation between $\left\| \left((F_K')^T \mathbb{M}_K F_K' \right)^{-1} \right\|$ and the minimum altitude of K with respect to $\mathbb{M}_K$.

Lemma 3.4.1. *There holds*

$$\frac{\hat{a}^2}{a_{K,\mathbb{M}_K}^2} \leq \left\| \left((F_K')^T \mathbb{M}_K F_K' \right)^{-1} \right\| \leq \frac{(d-1)^2 \hat{a}^2}{a_{K,\mathbb{M}_K}^2}, \tag{3.46}$$

where $\hat{a}$ is the altitude of $\hat{K}$ and $a_{K,\mathbb{M}_K}$ is the minimum altitude of K with respect to the metric $\mathbb{M}_K$.

Proof. First of all, we have

$$\left\| \left((F_K')^T F_K' \right)^{-1} \right\| = \left\| \left(\hat{E}^{-T} E_K^T E_K \hat{E}^{-1} \right)^{-1} \right\| = \left\| \hat{E} \left(E_K^T E_K \right)^{-1} \hat{E}^T \right\|.$$

Now, consider the QR decomposition of E_K

$$E_K = Q_K \begin{bmatrix} R_K \\ \mathbf{0} \end{bmatrix},$$

where $Q_K \in \mathbb{R}^{d \times d}$ is a unitary matrix, $R_K \in \mathbb{R}^{(d-1)\times(d-1)}$ is an upper triangular matrix, and

$\mathbf{0}$ is a $(d-1)$-dimensional row vector of zeros. With this we have

$$\begin{aligned}\left\|\hat{E}\left(E_K^T E_K\right)^{-1}\hat{E}^T\right\| &= \left\|\hat{E}\left([R_K^T\ \mathbf{0}^T]Q_K^T Q_K\begin{bmatrix}R_K\\ \mathbf{0}\end{bmatrix}\right)^{-1}\hat{E}^T\right\| \\ &= \left\|\hat{E}R_K^{-1}R_K^{-T}\hat{E}^T\right\| \\ &= \left\|\left(R_K\hat{E}^{-1}\right)^{-1}\left(R_K\hat{E}^{-1}\right)^{-T}\right\|.\end{aligned}$$

By [33, Lemma 4.1] we have

$$\frac{\hat{a}^2}{a_{R_K}^2} \le \left\|\left(R_K\hat{E}^{-1}\right)^{-1}\left(R_K\hat{E}^{-1}\right)^{-T}\right\| \le \frac{(d-1)^2\hat{a}^2}{a_{R_K}^2},$$

where a_{R_K} is the minimum altitude of the simplex formed by the columns of R_K. Since Q_K is a rotation matrix, a_{R_K} is the same as a_K, the minimum altitude of K with respect to the Euclidean metric. Combining the above results, we get

$$\frac{\hat{a}^2}{a_K^2} \le \left\|\left(\left(F_K'\right)^T F_K'\right)^{-1}\right\| \le \frac{(d-1)^2\hat{a}^2}{a_K^2}.$$

The inequality (3.46) follows from this and the observation that the geometric properties of K with respect to the metric $\mathbb{M}_K$ are the same as those of $\mathbb{M}_K^{1/2}K$ with respect to the Euclidean metric. □

Lemma 3.4.1 indicates that if $\hat{K}$ is chosen to satisfy $|\hat{K}| = \mathcal{O}(1)$ then

$$\left\|\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1}\right\| \sim a_{K,\mathbb{M}_K}^{-2}. \tag{3.47}$$

3.4.2 Mesh nonsingularity and existence of limiting meshes

We now consider the MMPDE (3.44). Recall that the velocities for the boundary vertices need to be modified in order for them to stay on the boundary. However, the analysis is

similar with or without modifications. Hence, for simplicity we do not consider modifications in the analysis. We also note that for theoretical purposes, we assume that $\hat{K}$ is taken to satisfy $|\hat{K}| = \frac{1}{N}$ instead of being unitary as we have been considering thus far. This change does not affect the actual computation. However, since typically we expect $|K| = \mathcal{O}(1/N)$, the assumption $|\hat{K}| = \frac{1}{N}$ will likely lead to $F'_K = \mathcal{O}(1)$ and thus $I_h(\mathcal{T}_h(0))$ (the value of I_h on the initial mesh $\mathcal{T}_h(0)$) stays $\mathcal{O}(1)$. On the other hand, if $|\hat{K}| = 1$ (unitary), we have $F'_K = \mathcal{O}(1/N)$ and $I_h(\mathcal{T}_h(0))$ will depend strongly on N.

In the following analysis, the mesh at time t is denoted by $\mathcal{T}_h(t) = (\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_{N_v}(t))$.

Theorem 3.4.1. *Assume that the meshing functional in the form (3.20) satisfies the coercivity condition as given in (3.19), i.e.,*

$$\tilde{G}\left(\mathbb{J}, \det\left(\mathbb{J}\right), \boldsymbol{x}\right) \geq \alpha \ \left(tr\left[\left(\left(F'_K\right)^T \mathbb{M}_K F'_K\right)^{-1}\right]\right)^q - \beta, \qquad \forall \boldsymbol{x} \in S \tag{3.48}$$

where $q > (d-1)/2$, $\alpha > 0$, and $\beta \geq 0$ are constants. We also assume that $\hat{K}$ is equilateral and $|\hat{K}| = \frac{1}{N}$. Then if the elements of the mesh trajectory of the MMPDE (3.44) have positive areas initially, they will have positive areas for all time. Moreover, their minimum altitudes in the metric $\mathbb{M}_K$ and their areas in the Euclidean metric are bounded below by

$$a_{K,\mathbb{M}_K} \geq C_1 \ \left[I_h(\mathcal{T}_h(0)) + \beta \bar{m}^{d/2} |S|\right]^{-\frac{1}{2q-d+1}} \ N^{-\frac{2q}{(d-1)(2q-d+1)}}, \tag{3.49}$$

$$|K| \geq C_2 \ \left[I_h(\mathcal{T}_h(0)) + \beta \bar{m}^{d/2} |S|\right]^{-\frac{d-1}{2q-d+1}} \ N^{-\frac{2q}{2q-d+1}} \ \overline{m}^{\,-\frac{d}{2}}, \tag{3.50}$$

where

$$C_1 = \left(\frac{\alpha \ d^{\frac{q(d-2)}{d-1}} \ (d-1)!^{\frac{2q-d+1}{d-1}}}{(d-1)^{\frac{d-1+2q}{2}}} \right)^{\frac{1}{2q-d+1}}, \qquad C_2 = \frac{C_1^{d-1}}{(d-1)^{\frac{d-1}{2}}(d-1)!}. \tag{3.51}$$

Proof. From (3.42) we have

$$\begin{aligned}\frac{dI_h}{dt} &= \sum_i \frac{\partial I_h}{\partial \boldsymbol{x}_i}\frac{d\boldsymbol{x}_i}{dt} = -\sum_i \frac{P_i}{\tau}\frac{\partial I_h}{\partial \boldsymbol{x}_i}\left[\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T - \left(\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T \cdot \boldsymbol{n}_i\right)\boldsymbol{n}_i\right] \\ &= -\sum_i \frac{P_i}{\tau}\left[\left\|\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right\|^2 - \left(\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T \cdot \boldsymbol{n}_i\right)^2\right] \\ &\leq 0.\end{aligned}$$

This implies $I_h\left(\mathcal{T}_h(t)\right) \leq I_h\left(\mathcal{T}_h(0)\right)$ for all t. From coercivity (3.19) and Lemma 3.4.1, we get

$$\begin{aligned}I_h\left(\mathcal{T}_h(t)\right) &\geq \alpha \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{1/2} \left(\operatorname{tr}\left[\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1}\right]\right)^q - \beta \bar{m}^{d/2}|S| \\ &\geq \alpha \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{1/2} \left\|\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1}\right\|^q - \beta \bar{m}^{d/2}|S| \\ &\geq \alpha \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{1/2} \frac{\hat{a}^{2q}}{a_{K,\mathbb{M}_K}^{2q}} - \beta \bar{m}^{d/2}|S|.\end{aligned}$$

By Lemma 3.8, $|\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{1/2} = |K|_{\mathbb{M}_K} \geq \frac{1}{(d-1)^{\frac{d-1}{2}}(d-1)!}\, a_{K,\mathbb{M}_K}^{d-1}$, thus

$$I_h\left(\mathcal{T}_h(t)\right) + \beta \bar{m}^{d/2}|S| \geq \frac{\alpha \hat{a}^{2q}}{(d-1)^{\frac{d-1}{2}}(d-1)!} \sum_{K\in\mathcal{T}_h} \frac{1}{a_{K,\mathbb{M}_K}^{2q-d+1}}\,, \tag{3.52}$$

and therefore

$$a_{K,\mathbb{M}_K}^{2q-d+1} \geq \frac{\alpha \hat{a}^{2q}}{(d-1)^{\frac{d-1}{2}}(d-1)!} \left(I_h\left(\mathcal{T}_h(0)\right) + \beta \bar{m}^{d/2}|S|\right)^{-1}. \tag{3.53}$$

Moreover, from the assumption that $\hat{K}$ is equilateral and $|\hat{K}| = \frac{1}{N}$ it follows that

$$\hat{a} = \frac{\sqrt{d}\ (d-1)!^{\frac{1}{d-1}}}{\sqrt{d-1}\ d^{\frac{1}{2(d-1)}}}\ N^{-\frac{1}{d-1}}. \tag{3.54}$$

Combining (3.53) and (3.54) we get

$$a_{K,\mathbb{M}_K} \geq C_1 \left[I_h(\mathcal{T}_h(0)) + \beta \bar{m}^{d/2}|S|\right]^{-\frac{1}{2q-d+1}} N^{-\frac{2q}{(d-1)(2q-d+1)}}, \tag{3.55}$$

where

$$C_1 = \left(\frac{\alpha \; d^{\frac{q(d-2)}{d-1}} \; (d-1)!^{\frac{2q-d+1}{d-1}}}{(d-1)^{\frac{d-1+2q}{2}}} \right)^{\frac{1}{2q-d+1}}$$

which gives (3.49).

Furthermore, we have

$$\begin{aligned} \frac{a_{K,\mathbb{M}_K}^{d-1}}{(d-1)^{\frac{d-1}{2}}(d-1)!} &\leq |K|_{\mathbb{M}_K} = |\hat{K}| \det \left((F_K')^T \, \mathbb{M}_K F_K' \right)^{1/2} \\ &\leq \overline{m}^{\, d/2} |\hat{K}| \det \left((F_K')^T \, F_K' \right)^{1/2} = \overline{m}^{\, d/2} |K|. \end{aligned}$$

Then (3.50) follows from the above inequality and (3.49).

Finally, from (3.38) and (3.39) it is not difficult to see that the magnitude of the mesh velocities is bounded from above when $|K|$ is bounded from below. As a consequence, the mesh vertices will move continuously with time and $|K|$ cannot jump over the bound to become negative. Hence, $|K|$ will stay positive if so initially. □

From the proof we have seen that the key points are the energy decreasing property and the coercivity of the meshing functional. The former is satisfied by the MMPDE (3.44) by design while the latter is an assumption for the meshing functional. We emphasize that the result holds for any functional satisfying the coercivity condition (3.19).

On the other hand, the condition (3.19) is satisfied by the meshing functional (3.15) for $0 < \theta \leq \frac{1}{2}$ and $p > 1$ (with $q = (d-1)p/2$ and $\beta = 0$ in Theorem 3.4.1). It is interesting to point out that the role of the parameter p can be explained from (3.49). Indeed, for this case the inequality (3.49) becomes

$$a_{K,\mathbb{M}_K} \geq C_1 \; \left[I_h(\mathcal{T}_h(0)) + \beta \bar{m}^{d/2} |S| \right]^{-\frac{1}{(d-1)(p-1)}} \; N^{-\frac{p}{(d-1)(p-1)}} \to C_1 \; N^{-\frac{1}{d-1}}, \quad p \to \infty. \quad (3.56)$$

Since $N^{-\frac{1}{d-1}}$ represents the average diameter of the elements, the above inequality implies that the mesh becomes more uniform as p gets larger. In numerical computation, we take

$p = 3/2$, which has been found to work well for all examples we have tested.

Theorem 3.4.2. *Under the assumptions of Theorem 3.4.1, for any nonsingular initial mesh, the mesh trajectory $\{\mathcal{T}_h(t), t > 0\}$ of MMPDE (3.44) has the following properties.*

1. *$I_h(\mathcal{T}_h(t))$ has a limit as $t \to \infty$, i.e.,*

$$\lim_{t\to\infty} I_h(\mathcal{T}_h(t)) = L. \tag{3.57}$$

2. *The mesh trajectory has limiting meshes, all of which are nonsingular and satisfy (3.49) and (3.50).*

3. *The limiting meshes are critical points of I_h, i.e., they satisfy*

$$\frac{\partial I_h}{\partial \boldsymbol{x}_i} = 0, \quad i = 1, \ldots, N_v. \tag{3.58}$$

Proof. The proof is very much the same as that for [32, Theorem 4.3] for the bulk mesh case. The key ideas to the proof are the monotonicity and boundedness of $I_h(\mathcal{T}_h(t))$ and the compactness of $\overline{S}$. With these holding for the surface mesh case, one can readily prove the three properties. □

It is remarked that the above two theorems have been obtained for the MMPDE (3.44) which is semi-discrete in the sense that it is discrete in space and continuous in time. A fully discrete scheme can be obtained by applying a time-marching scheme to (3.44). More specifically, consider the time integration of (3.44) and denote the time instants by t_n for $n = 0, 1, \ldots$ where we assume that $t_n \to \infty$ as $n \to \infty$. For integrating the MMPDE (3.44), we are interested in methods of the form

$$\mathcal{T}_h^{n+1} = \Psi(\mathcal{T}_h^n), \quad n = 0, 1, \ldots \tag{3.59}$$

for which integrating from t_n to t_{n+1} can be carried out in more than one step. One of the key points to the proof of Theorem 3.4.1 is the monotonically decreasing property of the energy functional. Therefore, for the fully discrete case we must assume that the scheme satisfies

$$I_h(\mathcal{T}_h^{n+1}) \leq I_h(\mathcal{T}_h^n), \quad n = 0, 1, \dots \ . \tag{3.60}$$

It should be noted that many schemes satisfy (3.60) including the forward and backward Euler schemes, and algebraically stable Runge-Kutta schemes (such as Gauss and Radau IIA schemes) under a time-step restriction involving a local Lipschitz bound of the Hessian matrix of I_h (e.g., [29, 54]).

Theorem 3.4.3. *Assume the assumptions of Theorem 3.4.1 are satisfied and that a numerical scheme in the form (3.59) is applied to the MMPDE (3.44). Furthermore, assume that the resulting mesh sequence $\{T_h^n\}_{n=0}^{\infty}$ satisfies the energy decreasing property given by (3.60). If the time step is sufficiently small but not diminishing and the elements of the mesh trajectory have positive areas initially then they will have positive areas for all $t_n > 0$. Moreover, the minimum altitudes in the metric $\mathbb{M}$ and the element areas are bounded away form zero by (3.49) and (3.50), respectively.*

Proof. We only need to show that the volumes of the elements will stay positive if the time step is sufficiently small but not diminishing after which the proof is similar to Theorem 3.4.1 for the semi-discrete case. To this end, assume G has continuous derivatives up to the third order. Similar to the discussion in the last paragraph of Theorem 3.4.1, when the mesh satisfies (3.49) and (3.50) we can show that the right-hand side of (3.44) as well as its gradient and Hessian are bounded by bounds independent of time and individual elements. After which, it can be shown that there exists $\delta t_0 > 0$ that only depends on the bounds mentioned above and thus is not diminishing such that if $t_{n+1} - t_n \leq \delta t_0$ then $\|\boldsymbol{x}_j^{n+1} - \boldsymbol{x}_j^n\|$ for $j = 1, \dots, N_v$ do not exceed a fixed fraction of the minimal altitude and, in the case an implicit scheme is used for (3.59), Newton's (or some other) iteration for the resulting

nonlinear algebraic equations converges. This then guarantees that the elements of the mesh trajectory will remain nonsingular during the current time step. The same argument can be repeated for each time step since the new mesh satisfies (3.49) and (3.50) and therefore, the volumes of the elements stay positive for $t_n > 0$. □

Extending Theorem 3.4.2 to the fully discrete case we are able to prove the following.

Theorem 3.4.4. *Under the assumptions of Theorem 3.4.3, for any nonsingular initial mesh, the mesh trajectory* $\{\mathcal{T}_h^n, n = 0, 1, \dots\}$ *of the scheme (3.59) applied to the MMPDE (3.44) has the following properties.*

1. $I_h(\mathcal{T}_h^n)$ *has a limit as* $n \to \infty$, *i.e.,*

$$\lim_{n\to\infty} I_h(\mathcal{T}_h^n) = L. \tag{3.61}$$

2. *The mesh trajectory has limiting meshes, all of which are nonsingular and satisfy (3.49) and (3.50).*

3. *If we further assume that the scheme satisfies a stronger property of monotonically decreasing energy,*

$$\begin{cases} I_h(\mathcal{T}_h^{n+1} \le I_h(\mathcal{T}_h^n), & n = 0, 1, \dots, \\ I_h(\mathcal{T}_h^{n+1} \le I_h(\mathcal{T}_h^n), & \text{if } \mathcal{T}_h^n \text{ is not a critical point,} \end{cases} \tag{3.62}$$

then the limiting meshes are critical points of I_h, *i.e., they satisfy (3.58).*

Proof. The proofs are very similar to those in Theorem 3.4.2. See [32] for more details. □

Notice Theorem 3.4.2 and Theorem 3.4.4 state that the values of the functional for the mesh trajectory converge which, as mentioned previously, can be used as a stopping criteria in computations. However, in general there is no guarantee that the mesh trajectory converges.

In order to guarantee convergence, a stronger requirement on the decent in the functional value or on the meshing functional is needed. For a more detailed discussion, see [32].

3.5 Numerical experiments

3.5.1 Definition of curvature for curves and surfaces

In the numerical results we will consider a metric tensor $\mathbb{M}$ based on curvature. In order to define such an $\mathbb{M}$ we must first derive the definition of curvature for both a curve in two-dimensions and a surface in three-dimensions. We will begin by deriving the curvature of a curve in three-dimensions then extending this to a surface in three-dimensions. To this end, let the curve in space be defined by $\gamma = \gamma(t) : I \to \mathbb{R}^3$ where $I \subset \mathbb{R}$. The curvature of γ measures the rate at which γ is turning. This can be described by the position and size of γ'' relative to γ' given by

$$\kappa(t) = \frac{\|\gamma' \times \gamma''\|}{\|\gamma'\|^3}. \tag{3.63}$$

The term $\|\gamma'\|^3$ in the denominator is used to ensure the value of the curvature does not change under reparametrization. One can use dimensional analysis to see this or, more rigorously, denote the reparametrization by $\beta(t) = \gamma(\phi(t))$. Then

$$\beta'(t) = \phi'(t)\gamma'(\phi(t)) \tag{3.64}$$

and

$$\beta''(t) = \phi''(t)\gamma'(\phi(t)) + \phi'(t)^2\gamma''(\phi(t)) \tag{3.65}$$

which gives

$$\beta'(t) \times \beta''(t) = \phi'(t)^3\gamma'(\phi(t)) \times \gamma''(\phi(t)). \tag{3.66}$$

Therefore

$$\frac{\|\beta' \times \beta''\|}{\|\beta'\|^3} = \frac{|\phi'(t)|^3\|\gamma'(\phi(t)) \times \gamma''(\phi(t))\|}{|\phi'(t)|^3\|\gamma'(\phi(t))\|^3} = \frac{\|\gamma'(\phi(t)) \times \gamma''(\phi(t))\|}{\|\gamma'(\phi(t))\|^3} = \frac{\|\gamma' \times \gamma''\|}{\|\gamma'\|^3} \tag{3.67}$$

and thus the curvature is independent of reparametrization.

From (3.63) one can readily see that the curvature κ for a curve is measured as the change in the tangential direction. The tangential direction is used because the tangent of a curve is a unique vector whereas the normal of a curve is a plane hence it is natural to define the curvature in terms of the unique vector. In the numerical results we only consider curves in $\mathbb{R}^2$ however, κ preserves definition (3.63) where $z = 0$ as we will see in the following examples.

Example 3.5.1. In our first example we consider a line given by

$$\gamma(t) = (t, ct + b, 0)$$

where c and b are constants. From (3.63) we have

$$\begin{aligned}\kappa(t) &= \frac{\|\gamma' \times \gamma''\|}{\|\gamma'\|^3} \\ &= \frac{\|(1, c, 0) \times (0, 0, 0)\|}{\|(1, c, 0)\|^3} \\ &= 0.\end{aligned}$$

Thus a line has 0 curvature which is consistent with the definition of curvature, i.e., curvature measures the rate at which a curve is turning. □

Example 3.5.2. In our second example we consider a circle with radius r and a counterclockwise parametrization given by

$$\gamma(t) = (r\cos(t), r\sin(t), 0).$$

By (3.63) we have

$$\begin{aligned}\kappa(t) &= \frac{\|\gamma' \times \gamma''\|}{\|\gamma'\|^3} \\ &= \frac{\|(-r\sin(t), r\cos(t), 0) \times (-r\cos(t), -r\sin(t), 0)\|}{\|(-r\sin(t), r\cos(t), 0)\|^3} \\ &= \frac{1}{r}.\end{aligned}$$

Thus a circle has constant curvature $\kappa(t) = \frac{1}{r}$. □

Example 3.5.3. In the last example we consider an ellipse given by

$$\gamma(t) = (a\cos(t), b\sin(t), 0)$$

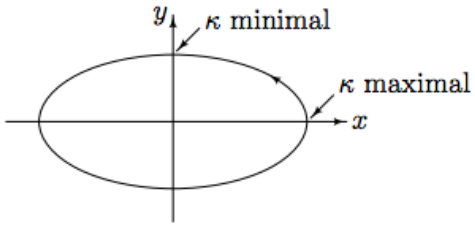
where $a, b > 0$. From (3.63) we have

$$\begin{aligned}\kappa(t) &= \frac{\|\gamma' \times \gamma''\|}{\|\gamma'\|^3} \\ &= \frac{\|(-a\sin(t), b\cos(t), 0) \times (-a\cos(t), -b\sin(t), 0)\|}{\|(-a\sin(t), b\cos(t), 0)\|^3} \\ &= \frac{ab}{(a^2\sin^2(t) + b^2\cos^2(t))^{3/2}},\end{aligned}$$

and therefore

$$\kappa(t) = \frac{ab}{(a^2\sin^2(t) + b^2\cos^2(t))^{3/2}}.$$

Notice that when $a > b$, κ attains its maximal value $\frac{a}{b^2}$ when $\sin(t) = 0$, i.e., the denominator is minimal. Furthermore, κ attains its minimal value $\frac{b}{a^2}$ when $\cos(t) = 0$, i.e., where the denominator is maximal.



This is consistent with the fact that the curvature of a line is 0 (Example 3.5.1) and hence the regions of the ellipse that are more linear will have smaller values of κ compared to those regions of the ellipse that have more curve, i.e., less linear. □

We would like to extend (3.63) from a space curve to a surface in $\mathbb{R}^3$. In order to do so, the definition must now be formulated in terms of the normal of the surface due to its uniqueness (in the same way that we used the tangent for the curve). To this end, let $\sigma : U \to \mathbb{R}^3$ denote the surface, $p = (u_0, v_0) \in U \subset \mathbb{R}^2$ a point, and $T_\sigma p$ the tangent space of σ at the point p. Then it follows that the unit normal vector at p is

$$N = \frac{\sigma_u \times \sigma_v}{\|\sigma_u \times \sigma_v\|}, \tag{3.68}$$

where $\sigma_u = \sigma_u(p)$ and $\sigma_v = \sigma_v(p)$. It is important to notice that $\|N\| = N^T N = 1$ and therefore

$$2\left(\frac{\partial N}{\partial u}\right)^T N = 0 \qquad \text{and} \qquad 2\left(\frac{\partial N}{\partial v}\right)^T N = 0.$$

From this we know that N_u, N_v are orthogonal to N and thus $N_u, N_v \in T_\sigma p$.

As mentioned above, to define the curvature κ of a surface we want to measure the change in N however, $N = N(u, v)$ thus we must consider both N_u and N_v. This could be done by using the Jacobian matrix for N however, this derivation does not remain the same under reparametrization. To ensure that reparametrization does not affect the curvature, we consider the mapping $W : T_\sigma p \to T_\sigma p$ defined by

$$W(\sigma_u) = -N_u = a\sigma_u + b\sigma_v \qquad \text{and} \qquad W(\sigma_v) = -N_v = c\sigma_u + d\sigma_v. \tag{3.69}$$

where $a, b, c, d \in \mathbb{R}$ and it should be noted that $\text{span}\{\sigma_u, \sigma_v\} = T_p\sigma$ and $-N_u, -N_v \in T_p\sigma$.

By (3.69) we see that W has the following matrix representation

$$W = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

thus we can write (3.69) as

$$\begin{bmatrix} \sigma_u & \sigma_v \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} -N_u & -N_v \end{bmatrix}.$$

This gives

$$\begin{bmatrix} \sigma_u^T \\ \sigma_v^T \end{bmatrix} \begin{bmatrix} \sigma_u & \sigma_v \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} \sigma_u^T \\ \sigma_v^T \end{bmatrix} \begin{bmatrix} -N_u & -N_v \end{bmatrix}$$

and therefore

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} \|\sigma_u\|^2 & \sigma_u^T\sigma_v \\ \sigma_v^T\sigma_u & \|\sigma_v\|^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sigma_u^T N_u & -\sigma_u^T N_v \\ -\sigma_v^T N_u & -\sigma_v^T N_v \end{bmatrix}. \tag{3.70}$$

Thus the shape operator W defined by (3.69) where

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} \|\sigma_u\|^2 & \sigma_u^T\sigma_v \\ \sigma_v^T\sigma_u & \|\sigma_v\|^2 \end{bmatrix}^{-1} \begin{bmatrix} \sigma_u^T W(\sigma_u) & \sigma_u^T W(\sigma_v) \\ \sigma_v^T W(\sigma_u) & \sigma_v^T W(\sigma_v) \end{bmatrix}. \tag{3.71}$$

As discussed earlier, we use W rather than the Jacobian to ensure that the curvature is unchanged under reparameterization. The idea behind using W is that a reparametrization will change N_u and N_v but will also change σ_u and σ_v. It turns out that these changes are directly related and so the map remains essentially the same. This implies that the shape operator is more directly related to a geometric property of the surface than the vectors N_u and N_v.

Theorem 3.5.1. *The shape operator W is unchanged under reparametrization which preserve orientation and it changes to $-W$ under reparametrization which reverse orientation.*

Proof. : See [53] for details. □

With the shape operator W, we can now define the principal curvatures. The principal curvatures, denote as κ_1 and κ_2, are the eigenvalues of the shape operator W. It follows by Theorem 3.5.1 that under reparameterization the principal curvatures are unchanged in absolute value. The principal curvatures lead to the definition of the Gaussian curvature

$$\kappa_G = \kappa_1 \cdot \kappa_2 \tag{3.72}$$

and the mean curvature

$$\kappa_M = \frac{\kappa_1 + \kappa_2}{2}. \tag{3.73}$$

To see the difference between (3.72) and (3.73) we consider the following examples.

Example 3.5.4. Consider a cylinder defined by

$$\sigma(u, v) = (\cos(v), \sin(v), u).$$

Then we have

$$\sigma_u = (0, 0, 1), \qquad \sigma_v = (-\sin(v), \cos(v), 0), \qquad N = (-\cos(v), \sin(v), 0),$$

and hence

$$N_u = (0, 0, 0) \qquad \text{and} \qquad N_v = (-\sin(v), -\cos(v), 0).$$

By (3.70) it follows that

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

One can readily see that the eigenvalues of W are $\kappa_1 = 0$ and $\kappa_2 = 1$. This is consistent

with the fact that the normal direction associated to κ_1 is vertical and hence the principal curvature is 0. Moreover, the normal direction associated to κ_2 is that of a planar circle with radius 1 and hence the principal curvature is 1 by Example 3.5.2. The Gaussian (3.72) and mean (3.73) curvatures are thus

$$\kappa_G = 0 \qquad \text{and} \qquad \kappa_M = \frac{1}{2}.$$

□

Notice that (3.63), (3.72), and (3.73) are defined for curves and surfaces that are parameterized parametrically. The naturally question that arises is how can we define curvature for a curve or surface with a Cartesian representation. For planar curves the following theorem holds true.

Theorem 3.5.2. *Consider the planar curve, $\Phi(x, y) = 0$. Then*

$$\kappa = \frac{(-\Phi_y, \Phi_x) \cdot Hess(\Phi) \cdot (-\Phi_y, \Phi_x)^T}{|\nabla \Phi|^3} \tag{3.74}$$

where κ is the curvature of Φ at the point (x, y), ∇ is the gradient, and Hess is the Hessian.

Proof. See [24] for details. □

We can examine (3.74) by using Example 3.5.2 as follows.

Example 3.5.5. Consider a circle of radius r given by

$$\Phi(x, y) = x^2 + y^2 - r^2.$$

Then

$$(-\Phi_y, \Phi_x) = (-2y, 2x), \quad \nabla \Phi = (2x, 2y) \quad \text{and} \quad \text{Hess}(\Phi) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

Therefore, (3.74) gives

$$\kappa = \frac{(-2y, 2x) \cdot \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \cdot (-2y, 2x)^T}{|(2x, 2y)|^3} = \frac{8x^2 + 8y^2}{(4x^2 + 4y^2)^{3/2}} = \frac{1}{(x^2 + y^2)^{1/2}} = \frac{1}{r},$$

which is consistent with the calculation in Example 3.5.2. □

The results for Example 3.5.3 are similar.

For surfaces in $\mathbb{R}^3$ with Cartesian representation, the Gaussian and mean curvature are defined as in the following theorem.

Theorem 3.5.3. *Let $\Phi(x, y, z) = 0$ define an implicit surface. Then*

$$\kappa_G = \frac{\nabla\Phi \cdot Hess^*(\Phi) \cdot \nabla\Phi}{|\nabla\Phi|^4}, \tag{3.75}$$

and

$$\kappa_M = \frac{\nabla\Phi \cdot Hess(\Phi) \cdot \nabla\Phi^T - |\nabla\Phi|^2 \, tr(Hess(\Phi))}{2\,|\nabla\Phi|^3}, \tag{3.76}$$

where $tr(\cdot)$ is the trace, ∇ is the gradient, Hess is the Hessian, and $Hess^*$ is the adjoint of the Hessian.

Proof. See [24] for details. □

We can examine (3.74) in Example 3.5.4 as follows.

Example 3.5.6. Consider a cylinder defined by

$$\Phi(x, y, z) = x^2 + y^2 - 1.$$

Then

$$\nabla\Phi = (2x, 2y, 0) \quad \text{and} \quad \text{Hess}(\Phi) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

which gives that $\text{tr}(\text{Hess}(\Phi)) = 4$ and $\text{Hess}(\Phi)^*$ is the zero matrix. Then it readily follows that $\kappa_G = 0$ and

$$\kappa_M = \frac{(2x, 2y, 0) \cdot \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot (2x, 2y, 0)^T - 4\left|(2x, 2y, 0)\right|^2}{2\left|(2x, 2y, 0)\right|^3} = \frac{-4x^2 - 4y^2}{(4x^2 + 4y^2)^{3/2}} = -\frac{1}{2}.$$

Notice that the mean curvature here is negative but $\nabla\Phi = (2x, 2y, 0)$ is the outward pointing normal. Thus, as one would expect, the mean curvature vector defined as $\frac{\kappa_M}{|\nabla\Phi|}\nabla\Phi$ points into the cylinder and has magnitude $\frac{1}{2}$. The magnitude of the curvature is consistent with Example 3.5.4. □

3.5.2 Numerical results

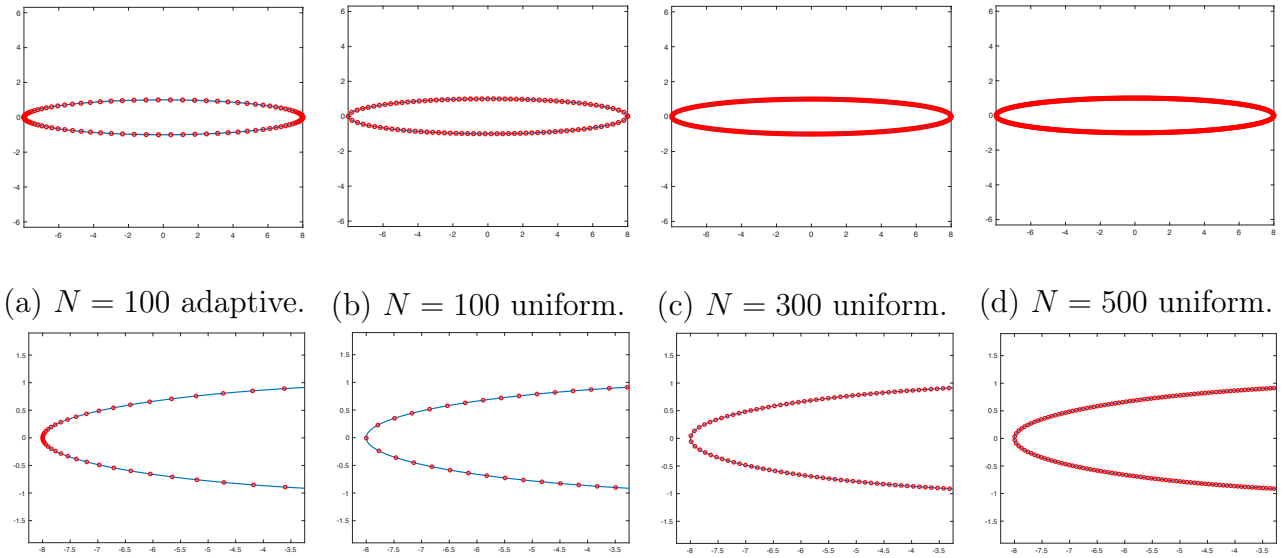
As mentioned in Chapter 1, mesh adaptation has been proven to be an extremely useful tool due to its ability to concentrate elements in specific regions of the domain or, now with the method described above, the surface. One of the main advantages to using an adaptive mesh rather than a uniform mesh is that fewer elements are required to accurately represent a curve or a surface. To see this we compare an adaptive mesh with a curvature based metric tensor to a uniform mesh for the ellipse defined by

$$\Phi(x, y) = \frac{x^2}{64} + y^2 - 1.$$

More specifically, Figure 3.1(a) and (e) show the adaptive mesh for the ellipse using a total of $N = 100$ elements. Indeed, the concentration of elements is higher in the regions with larger curvature and smaller in the more linear regions of the curve which is consistent with the curvature based metric tensor used. With this concentration, the adaptive mesh represents the geometry of the curve well with only 100 total elements. However, when a uniform mesh is used, significantly more elements are required to attain a similar concentration of elements in the curved regions (Figure 3.1(e)).

From Figure 3.1(b) and (f), one can readily see that curve is not well represented when a uniform mesh with a total of 100 elements is used. The elements are equidistant along the curve making the regions with larger curvature poorly represented. When we increase the number of elements to 300, although better than $N = 100$, the uniform mesh can still be seen to be a less than ideal representation of the curve, specifically in the regions with large curvature. Finally, when $N = 500$, we can see a similar concentration in the curved regions to that of the adaptive mesh. The concentration of elements at $x = -8$ and 8 (Figure 3.1(h)) is similar to the concentration with the adaptive mesh (Figure 3.1(e)). There are more elements in the linear regions of the curve when using the uniform mesh with $N = 500$ compared to the adaptive mesh however, this is not necessary since these more linear regions can be well represented with few elements. Thus, the curve has a similar representation when using an adaptive mesh with 100 total elements or a uniform mesh with 500 total elements. For computational purposes, one can see that it is a significant advantage to use the adaptive mesh method compared to a uniform mesh as shown in Figure 3.1.

With this in mind, we present numerical results for a selection of two- and three-dimensional examples to demonstrate the performance of the surface moving mesh method described in the previous sections. The main focus will be on showing how our method can be used for mesh smoothing and concentration. To assess the quality of the generated meshes, we compare the equidistribution (Q_{eq}) and alignment (Q_{ali}) mesh quality measures



(a) $N = 100$ adaptive. (b) $N = 100$ uniform. (c) $N = 300$ uniform. (d) $N = 500$ uniform.

(e) Zoomed in (a). (f) Zoomed in (b). (g) Zoomed in (c). (h) Zoomed in (d).

Figure 3.1: Adaptive and uniform meshes for the ellipse with $N = 100$ for the adaptive mesh and $N = 100, 300$, and 500 for the uniform meshes.

which are defined as

$$Q_{eq} = \max_{K \in \mathcal{T}_h} \frac{\det\left((F_K')^T \mathbb{M}_K F_K'\right)^{\frac{1}{2}}}{\sigma_h / N}, \tag{3.77}$$

and

$$Q_{ali} = \max_{K \in \mathcal{T}_h} \frac{\operatorname{tr}\left[\left((F_K')^T \mathbb{M}_K F_K'\right)^{-1}\right]}{(d-1)\det\left((F_K')^T \mathbb{M}_K F_K'\right)^{-\frac{1}{d-1}}}. \tag{3.78}$$

These measures are indications of how closely the mesh satisfies the equidistribution condition (3.5) and the alignment condition (3.10), respectively. The closer these quality measures are to 1, the closer they are to a uniform mesh with respect to the metric $\mathbb{M}_K$. It should be noted that the alignment condition does not apply to the two-dimensional case where a "surface" is actually a curve. Mathematically, when $d = 2$, $(F_K')^T \mathbb{M}_K F_K'$ is a number and hence (3.10) is always satisfied.

For all computations we use $p = 3/2$ and $\theta = 1/3$ in the meshing functional (3.15). This choice has been known to work well in bulk mesh applications. Interestingly, we have found that it also works well for all surface mesh examples we have tested. We take $\tau = 0.01$,

$dt = 0.01$, and

$$P_i = \det\left(\mathbb{M}(\boldsymbol{x}_i)\right)^{\frac{p(d-1)-d}{2}}.$$

The latter is to ensure that the MMPDE (3.44) be invariant under scaling transformations of $\mathbb{M}$. For all of the results, we run to a final time of 1.0.

We choose two forms of $\mathbb{M}_K$. The first is $\mathbb{M}_K = I$, which will ensure the mesh moves to become as uniform as possible with respect to the Euclidean norm. The second is a curvature-based metric tensor defined as a scalar matrix $\mathbb{M}_K = (k_K + \epsilon)\, I$, where k_K is the mean curvature and ϵ is machine epsilon. The mean curvature is defined [24] for a curve $\Phi(x, y) = 0$ in $\mathbb{R}^2$ as (3.74) given by

$$k = \left| \frac{\Phi_{xx}\Phi_y^2 - 2\Phi_{xy}\Phi_x\Phi_y + \Phi_x^2\Phi_{yy}}{\left(\Phi_x^2 + \Phi_y^2\right)^{\frac{3}{2}}} \right|$$

and for a surface $\Phi(x, y, z) = 0$ in $\mathbb{R}^3$ as (3.76) given by

$$k = \left| \frac{D_1 + D_2 + D_3 - D_4}{2\left(\Phi_x^2 + \Phi_y^2 + \Phi_z^2\right)^{3/2}} \right|,$$

where

$$\begin{aligned}
D_1 &= \Phi_x\left(\Phi_x\Phi_{xx} + \Phi_y\Phi_{xy} + \Phi_z\Phi_{xz}\right), \\
D_2 &= \Phi_y\left(\Phi_x\Phi_{xy} + \Phi_y\Phi_{yy} + \Phi_z\Phi_{yz}\right), \\
D_3 &= \Phi_z\left(\Phi_x\Phi_{xz} + \Phi_y\Phi_{yz} + \Phi_z\Phi_{zz}\right), \\
D_4 &= \left(\Phi_x^2 + \Phi_y^2 + \Phi_z^2\right)\left(\Phi_{xx} + \Phi_{yy} + \Phi_{zz}\right).
\end{aligned}$$

We would like to explore more metric tensors in future work but will focus on these two for the numerical results in this thesis.

Not all of the examples have an analytic expression for Φ thus, in these cases, the normal vectors must approximated using spline functions. The initial mesh for all of the examples

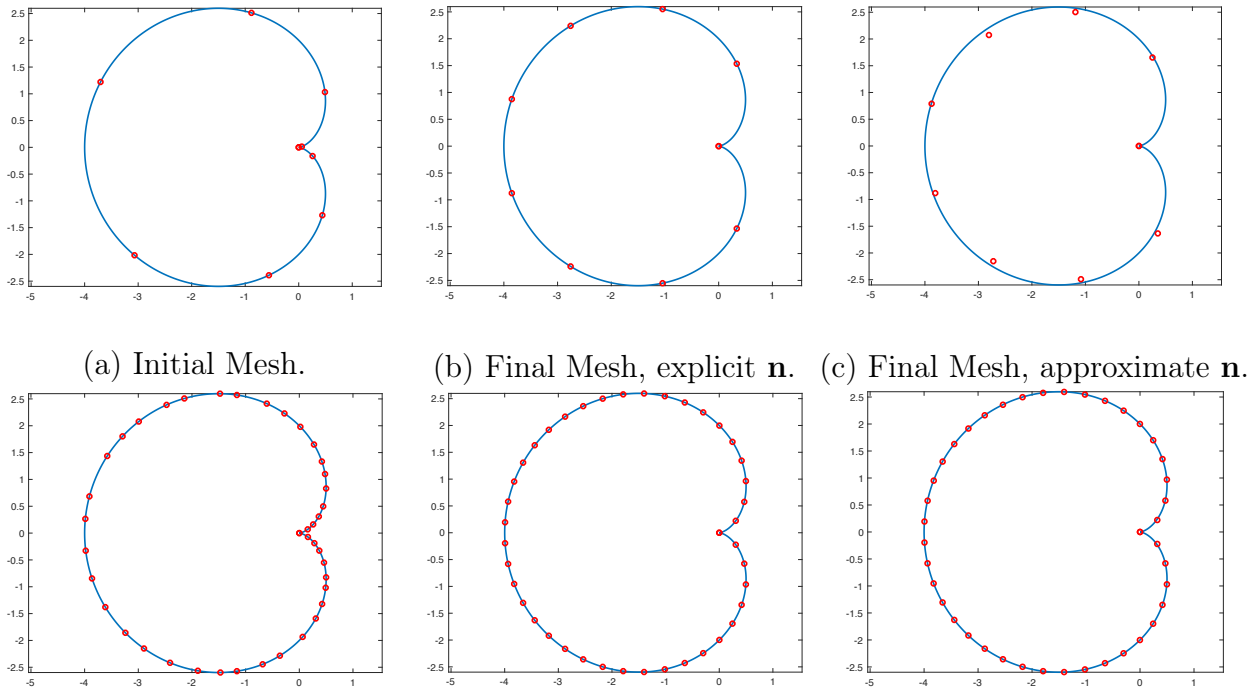
provided is taken to be fine enough so that this approximation does not affect the mesh movement however, if the initial mesh is too coarse, the nodes can move from the surface during adaptation. To see this consider the cardioid defined by

$$\Phi(x, y) = \left(x^2 + y^2\right)^2 + 4x(x^2 + y^2) - 4y^2. \tag{3.79}$$

We adapt the mesh on the curve for both $N = 10$ and $N = 40$. Here, we consider the metric tensor associated to the Euclidean metric (Figure 3.2(a) - (f)). For the curvature based metric tensor, see Figure 3.16. In this example, we fix the node $\boldsymbol{x}_1 = (0, 0)$. Figure 3.2 compares the final meshes using normal vectors calculated explicitly to the final meshes using normal vectors approximated via spline functions for the Euclidean metric.

From Figure 3.2 we see that the mesh adapts from a very nonuniform initial mesh (Figure 3.2(a) and (d)) to a uniform final mesh (Figure 3.2(b),(c),(e), and (f)) which is consistent with the fact that the metric tensor corresponds to the Euclidean metric. However, for $N = 10$, we see the nodes move off of the curve when the normal vectors are approximated (Figure 3.2(c)) whereas when the normal vectors are calculated explicitly, the nodes remain on the curve (Figure 3.2(b)). For $N = 40$, the nodes remain on the curve when the normal vectors are approximated and calculated explicitly as shown in Figure 3.2(e) and (f).

This example shows that the initial mesh must be fine enough in order for the nodes to remain on the surface during movement when the normal vectors are approximated. When the initial mesh is fine, however, the final meshes when using approximate normal vectors and analytical normal vectors are identical. When a Cartesian representation is available, we will note whether the normal vectors are approximated or calculated explicitly however, the initial meshes are fine enough that there is no difference in the final mesh between the approximate and explicit normal vectors.



(a) Initial Mesh. (b) Final Mesh, explicit $\mathbf{n}$. (c) Final Mesh, approximate $\mathbf{n}$.

(d) Initial Mesh. (e) Final Mesh, explicit $\mathbf{n}$. (f) Final Mesh, approximate $\mathbf{n}$.

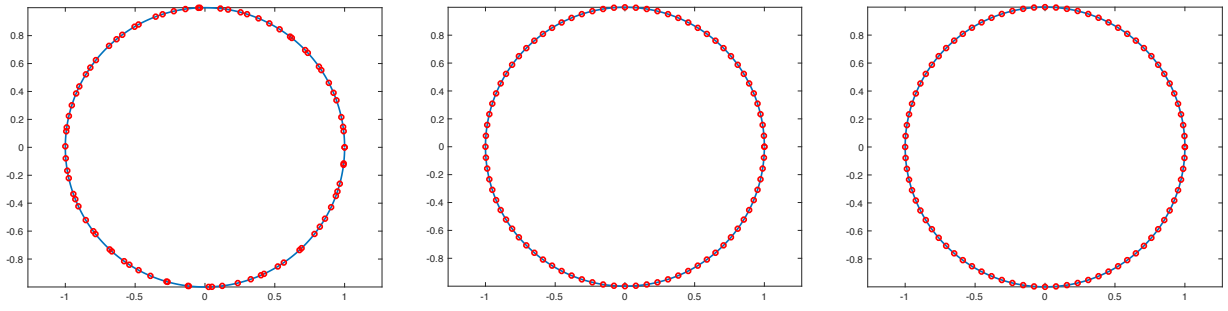
Figure 3.2: Meshes of $N = 10$ and $N = 40$ for the cardioid using analytical and approximate normal vector with the Euclidean metric tensor.

Example 3.5.7. For the first example, we generate adaptive meshes for the unit circle in two dimensions,

$$\Phi(x, y) = x^2 + y^2 - 1.$$

We take $N = 80$ and fix the node $\boldsymbol{x}_1 = (1, 0)$. The normal vectors are calculated explicitly in this example.

Figure 3.3 shows the meshes for this example. Studying the figures we see that the initial mesh Figure 3.3(a) is very nonuniform but the final meshes Figure 3.3(b) and (c) have adapted to be equidistant along the curve. Moreover, the final meshes for both $\mathbb{M}_K = I$ (Figure 3.3 (b)) and $\mathbb{M}_K = (k_K + \epsilon)I$ (Figure 3.3(c)) adapt the mesh in the same manner. This is consistent with the fact that the curvature of a circle is constant thus the nodes do not concentrate in one particular region of the curve. The final meshes in both cases provide good size adaptation and are more uniformly distributed along the curve when compared with the initial mesh. This can be further supported assessing the mesh quality measure for which Q_{eq} improves from 7.509604 to 1.000004 for both cases of $\mathbb{M}_K$. The fact that $Q_{eq} \approx 1$ indicates that the mesh is close to satisfying the equidistribution condition (3.5) and hence the mesh is almost uniform with respect to the metric tensor $\mathbb{M}_K$. It can also be seen that the nodes remain on the curve Φ, which is an inherent feature of the new surface moving mesh method and indeed an important one when adapting a mesh on a curve. □



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.3: Example 3.5.7. Meshes of $N = 80$ are obtained for $\Phi(x, y) = x^2 + y^2 - 1$.

Example 3.5.8. The second two-dimensional example is the ellipse defined by

$$\Phi(x, y) = \frac{x^2}{64} + y^2 - 1.$$

In this example we take $N = 60$ and fix the node $\boldsymbol{x}_1 = (8, 0)$. The normal vectors in this example are approximated via spline functions.

The initial nodes (Figure 3.4(a)) are randomly distributed along the curve. However, for $\mathbb{M}_K = I$, the final mesh (Figure 3.4(b)) is equidistant along the ellipse providing a much more uniform mesh. This can also be seen in Q_{eq} which improves from 5.497002 initially to 1.026912 in the final mesh.

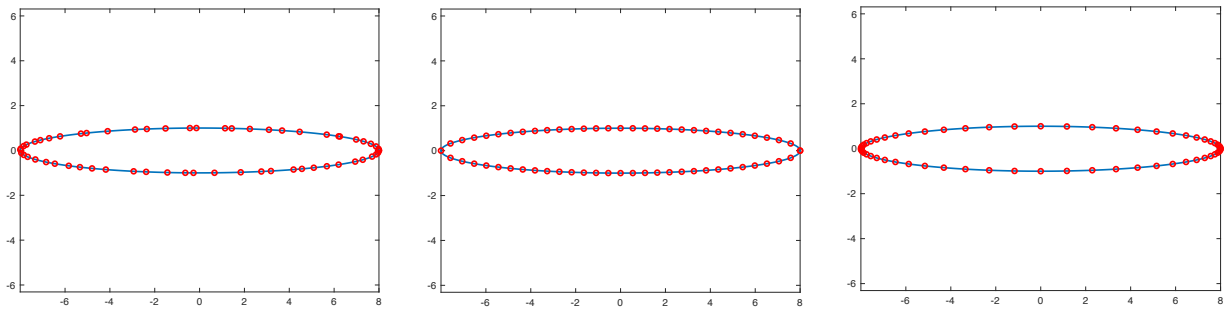
Now considering the curvature-based metric tensor (Figure 3.4(c)), we can see a high concentration of elements near the regions of the ellipse with large curvature. This is consistent with the equidistribution principle which requires higher concentration in the regions with larger determinant of the metric tensor (larger mean curvature in the current situation). The mean curvature is large in the regions of the ellipse close to $x = -8, 8$ and almost 0 for $x \in (-2, 2)$. From Figure 3.4(c) we can see that the adaptation with $\mathbb{M}_K = (k_K + \epsilon)I$ provides a mesh that represents the shape of the curve much better than other two meshes. The improvement of Q_{eq} from 5.126216 to 1.015848 indicates that the final mesh is almost uniform with respect to the curvature-based metric tensor. □

Example 3.5.9. For the next two-dimensional example, we generate adaptive meshes for the sine curve defined by

$$\Phi(x, y) = 4\sin(x) - y.$$

In this example we take $N = 60$ and fix the end nodes $\boldsymbol{x}_1 = (0, 0)$ and $\boldsymbol{x}_{61} = (2\pi, 0)$. We calculate the normal vectors explicitly in this example.

Figure 3.5 shows the meshes for this example. From Figure 3.5(a) and (b) we see that for $\mathbb{M}_K = I$, the mesh becomes much more uniform. This is consistent with the fact that for



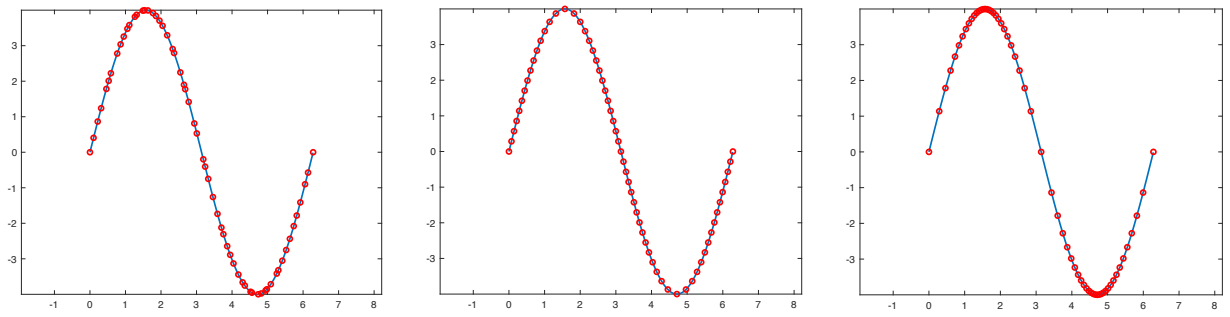
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.4: Example 3.5.8. Meshes of $N = 60$ are obtained for $\Phi(x, y) = \dfrac{x^2}{64} + y^2 - 1$.

$\mathbb{M} = I$, the minimization of the meshing functional will make the mesh more uniform with respect to the Euclidean norm. The observation can be further supported by assessing the mesh quality measures for which Q_{eq} measure improves from 4.183312 to 1.002906 indicating that the final mesh satisfies the equidistribution condition (3.5) closely.

Now studying Figure 3.5(c) where $\mathbb{M}_K = (k_K + \epsilon)I$ is used, we see that there is a high concentration of mesh elements in regions with large curvature, i.e., the hill at $y = 4$ and cup at $y = -4$, which is consistent with the use of the curvature-based metric tensor. Moreover, the equidistribution measure Q_{eq} improves from 6.254755 to 1.007493. This indicates that although the mesh may seem nonuniform in the Euclidean metric, it is almost uniform in the metric $\mathbb{M}_K$.

As discussed in Section 3.4, theoretically we know that the value of I_h is decreasing and $|K|$ is bounded below. To see these numerically, we plot I_h and $|K|_{\min}$ as functions of t in Figure 3.6, where $|K|_{\min}$ denotes the minimum area of K over all elements in $\mathcal{T}_h$. The numerical results are shown to be consistent with the theoretical predictions. Specifically, for $\mathbb{M}_K = I$, Figure 3.6(a) shows that I_h is decreasing and bounded below by 9.535. Additionally, Figure 3.6(b) suggests that $|K|_{\min}$ is bounded below by 0.235 which is the value of $|K|_{\min}$ of the initial mesh. As we see, $|K|_{\min}$ first increases and then converges to about $0.285 \approx \frac{|S|}{N}$. The reason is because in the final mesh, the elements are close to being uniform with respect



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.5: Example 3.5.9. Meshes of $N = 60$ are obtained for $\Phi(x, y) = 4\sin(x) - y$.

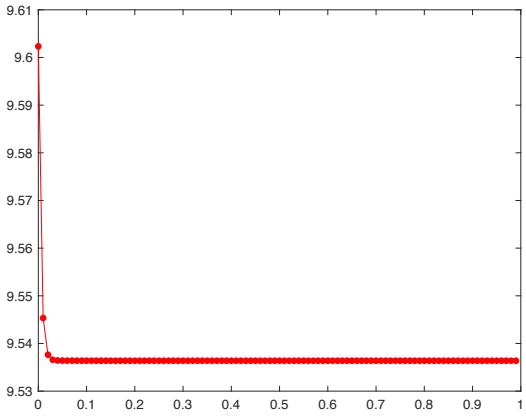
to the Euclidean metric and thus $|K| \approx \frac{|S|}{N}$ for all K. Since the initial mesh is nonuniform, we expect an increase in $|K|_{\min}$ as the mesh is becoming more uniform. Moreover, as the mesh reaches the limiting mesh trajectory around $t = 0.05$, we see that $|K|_{\min}$ converges as shown in Figure 3.6(b).

For the case with $\mathbb{M}_K = (k_K + \epsilon)I$, the numerical results are again consistent with the theoretical predictions. Figure 3.6(c) shows that I_h is decreasing for all time and bounded below by 15.5. This figure also shows that at around $t = 0.15$, I_h begins to converge. In Figure 3.6 (d), $|K|_{\min}$ has similar properties to Figure 3.6(b). That is, we see an initial increase in $|K|_{\min}$ after which, the value converges to 0.11 starting at around $t = 0.15$. Furthermore, Figure 3.6(d) suggests that $|K|_{\min}$ is bounded below by the initial value of 0.045. □

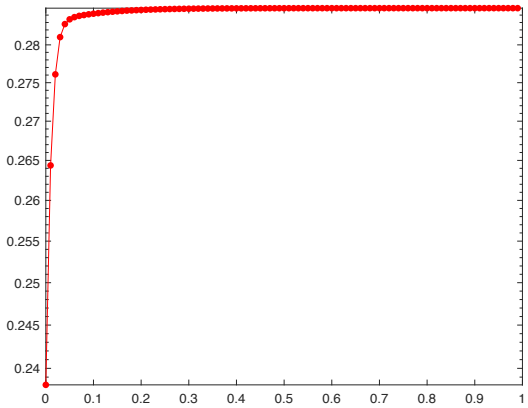
Example 3.5.10. As the final two-dimensional example, we generate adaptive meshes for the lemniscate defined by

$$\Phi(x, y) = (x^2 + y^2)^2 - 4(x^2 - y^2).$$

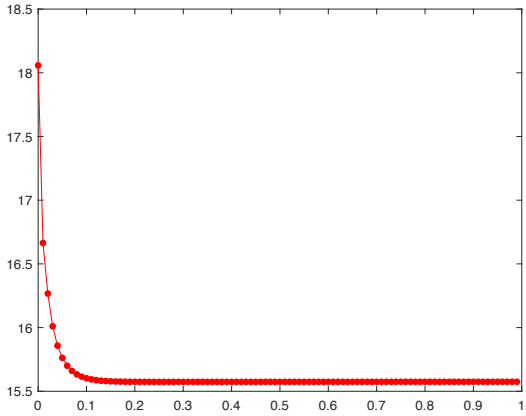
In this example we adapt the mesh on the curve for both $N = 60$ and $N = 120$. In both situations, we fix the node $\boldsymbol{x}_1 = (2, 0)$. The normal vectors are calculated explicitly.



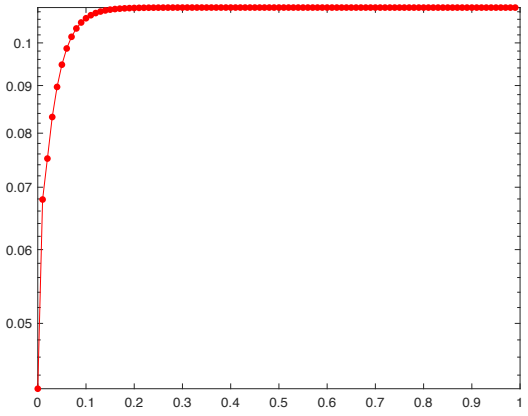
(a) I_h, $\mathbb{M}_K = I$



(b) $|K|_{\min}$, $\mathbb{M}_K = I$



(c) I_h, $\mathbb{M}_K = (k_K + \epsilon)I$

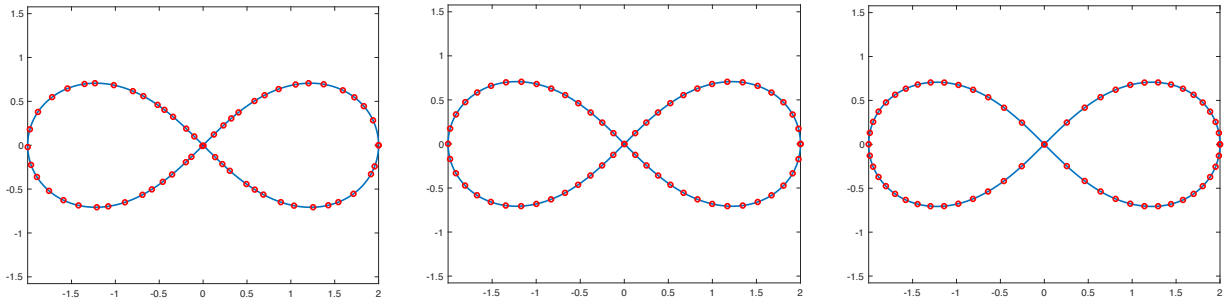


(d) $|K|_{\min}$, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.6: Example 3.5.9. I_h and $K_{\min}$ plotted as functions of t for $\Phi(x, y) = 4\sin(x) - y$.

From Figure 3.7 we see that for $N = 60$ the mesh adapts from a very nonuniform initial mesh (Figure 3.7(a)) to a uniform final mesh (Figure 3.7(b)) when considering the metric tensor corresponding to the Euclidean metric. The nodes are equidistant apart while remaining on the curve. This improvement in uniformity can be further supported by the equidistribution quality measure which improves from 2.083287 for the initial mesh to 1.002549 for the final mesh.

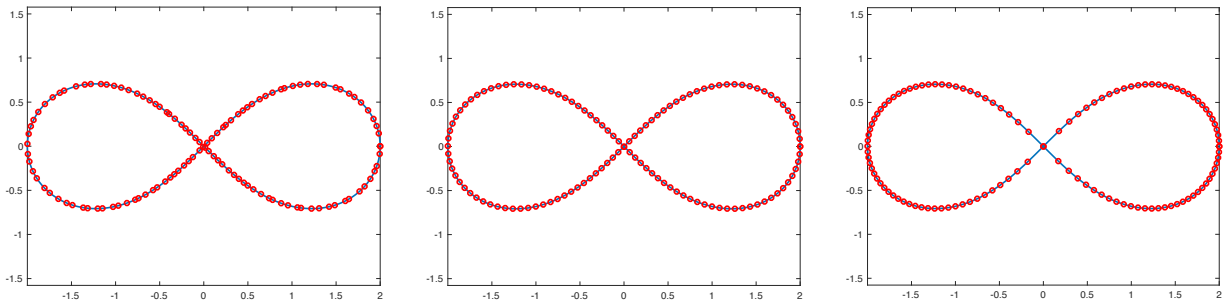
We see a similar result when the curvature-based metric tensor is used (Figure 3.7(c)). A higher concentration of nodes occurs in the circular regions with larger curvature compared to the cross section which has smaller curvature (i.e., the linear regions). It is not a significant difference in concentration but this is consistent with the fact that the curvature of the lemniscate is close to but not exactly constant. The equidistribution quality measure improves from 3.364232 to 1.001011 indicating that the final mesh is much more uniform with respect to the curvature-based $\mathbb{M}_K$ than the initial mesh.



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.7: Example 3.5.10. Meshes of $N = 60$ are obtained for the lemniscate $\Phi(x, y) = (x^2 + y^2)^2 - 4(x^2 - y^2)$.

For $N = 120$, Figure 3.8 shows similar findings. When considering the Euclidean metric, we see the mesh, Figure 3.8(a), is very nonuniform initially and adapts to a equidistant spacing of the nodes along the curve. This is further supported in the quality measures for which the equidistribution measure improves from 2.552134 to 1.002167 indicating that the final mesh is close to satisfying the equidistribution condition.



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.8: Example 3.5.10. Meshes of $N = 120$ are obtained for the lemniscate $\Phi(x, y) = (x^2 + y^2)^2 - 4(x^2 - y^2)$.

The curvature-based metric tensor results in a similar adaptation as before. Figure 3.8(c) shows the final mesh has adapted in such a way where there is a higher concentration of nodes in those regions of the curve with larger curvature, i.e., circular regions. Comparatively, there are fewer nodes in the cross section which has smaller curvature. The difference in concentration can be clearly seen in Figure 3.8(c) with $N = 120$ nodes. The adaptation is consistent with the curvature of the lemniscate, which is close to but not exactly constant. This improvement in uniformity can be further supported by the equidistribution quality measure which improves from 8.023253 for the initial mesh to 1.001855 for the final mesh. □

Example 3.5.11. Let us now consider surfaces in $\mathbb{R}^3$. In this first example, we consider adaptive meshes for the torus defined by

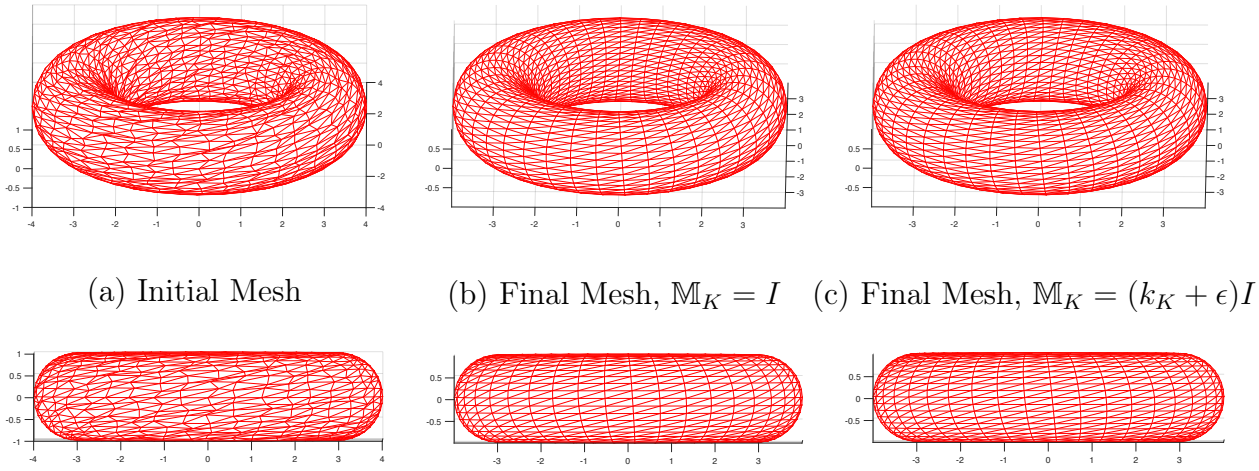
$$\Phi(x, y, z) = \left(2 - \sqrt{x^2 + y^2}\right)^2 + z^2 - 1,$$

where $x, y \in [-3, 3]$, and $z \in [-1, 1]$. We take $N = 3200$. The normal vectors are approximated using spline functions in this example.

Figure 3.9 shows the meshes for this example in two different views. Studying Figure 3.9(a), the initial mesh, and Figure 3.9(b), the final mesh with $\mathbb{M}_K = I$, we can see that

the final mesh provides a more uniform distribution of the nodes. That is consistent with the use of the metric tensor $\mathbb{M} = I$ whose goal is to make the mesh as uniform as possible in the Euclidean norm. This can also be confirmed from the equidistribution and alignment quality measures. The equidistribution measure for the initial mesh is 15.50150 and for the final mesh 1.332488. Similarly, the initial alignment quality measure is 30.63276 compared to that of the final mesh which is 1.920701.

For the curvature-based metric tensor, we see similar results to that of the Euclidean metric. That is, the final mesh for the curvature-based metric tensor, Figure 3.9(c), looks identical to the final mesh for the Euclidean metric, Figure 3.9(b). This is because the absolute value of the mean curvature of a torus is close to constant and hence, the elements do not concentrate in any particular region of the surface. □



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

(d) side view of (a) (e) side view of (b) (f) side view of (c)

Figure 3.9: Example 3.5.11. Meshes of $N = 3200$ are obtained for the surface $\Phi(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$.

Example 3.5.12. The second three-dimensional example is the cylinder defined by

$$\Phi(x, y, z) = x^2 + y^2 - 1,$$

where $z \in [-2, 2]$. For this example we take $N = 3200$. In this example, the normal vectors are calculated explicitly. Two boundary nodes were fixed, $\boldsymbol{x}_1 = (0, 1, -2)$ and $\boldsymbol{x}_1 = (0, 1, 2)$, but the remaining boundary nodes were allowed to slide along the boundary. Although the cylinder has constant curvature like Example 3.5.11, this example shows the adaptation on a surface with a boundary.

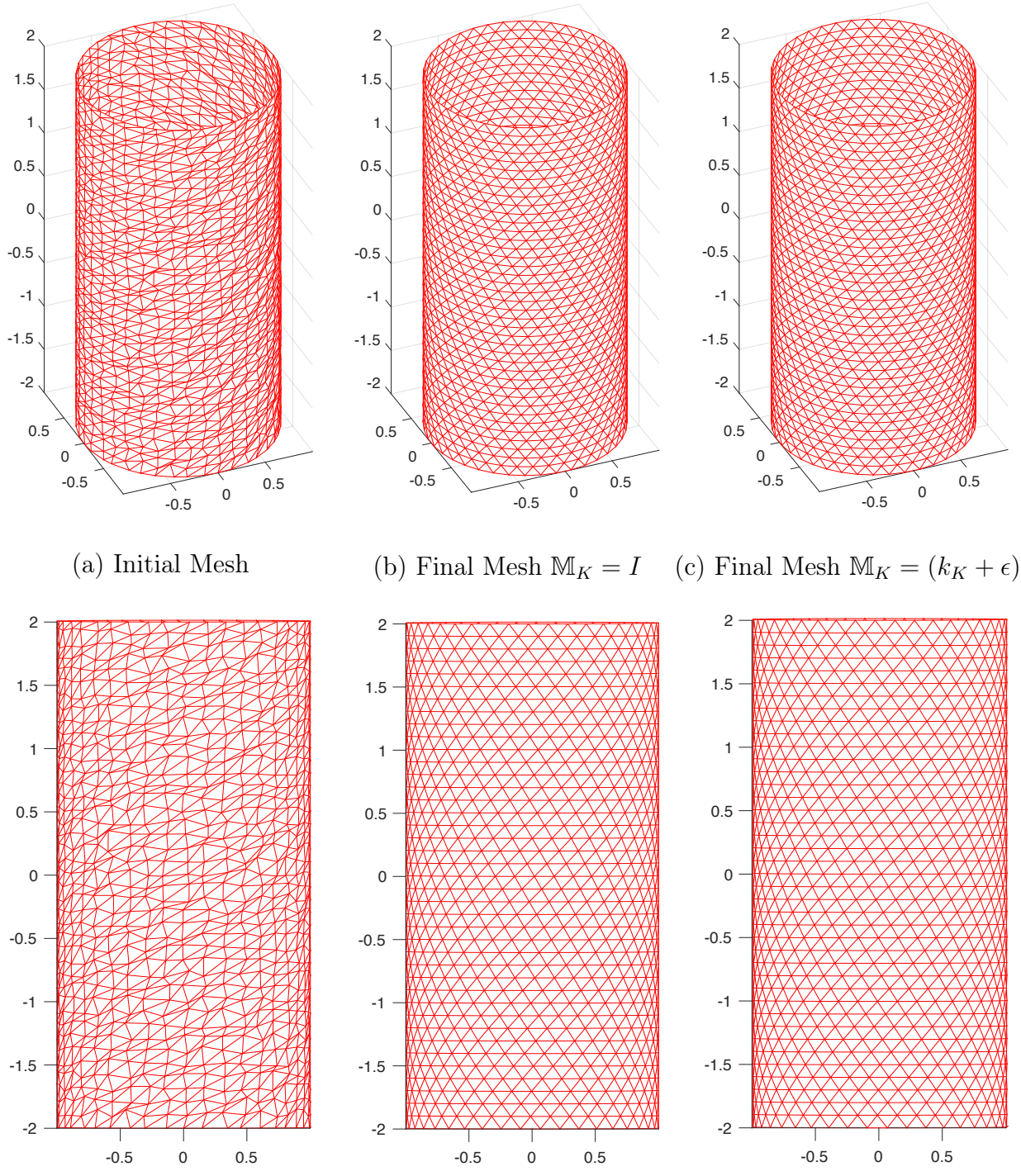
Figure 3.10 shows the adaptive meshes for the cylinder in two different views. For both $\mathbb{M}_K = I$ and $\mathbb{M}_K = (k_K + \epsilon)I$, the mesh becomes much more uniform and identical. This is consistent with the constant curvature of the cylinder hence the nodes do not concentrate in any specific region of the surface. The equidistribution quality measure improves from 19.07656 to 1.054857 and the alignment quality measure from 23.35403 to 1.192268. The fact that the final quality measures for both conditions are close to 1 indicates that the final meshes are close to satisfying conditions (3.5) and (3.10). □

Example 3.5.13. Our next example is the sine surface in three dimensions defined by

$$\Phi(x, y, z) = \sin(x + y) - z,$$

where $x \in [-2, 2]$, $y \in [\frac{\pi}{2}, \frac{3\pi}{2}]$, and $z \in [-1, 1]$. For this example we take $N = 3200$ and fix the boundary nodes. The normal vectors are approximated using spline functions.

Figure 3.11 shows the adaptive meshes for this examples in two different views. It is clear in Figure 3.11, when $\mathbb{M}_K = I$, the mesh becomes more uniform in the Euclidean metric from the initial mesh Figure 3.11(a) to the final mesh Figure 3.11(b). The top view of the surface, Figure 3.11(d) and (e), further confirm this. It is also supported by the improvement of the quality measures from $Q_{eq} = 4.234781$ to 1.669880 and $Q_{ali} = 6.643755$ to 1.702617.

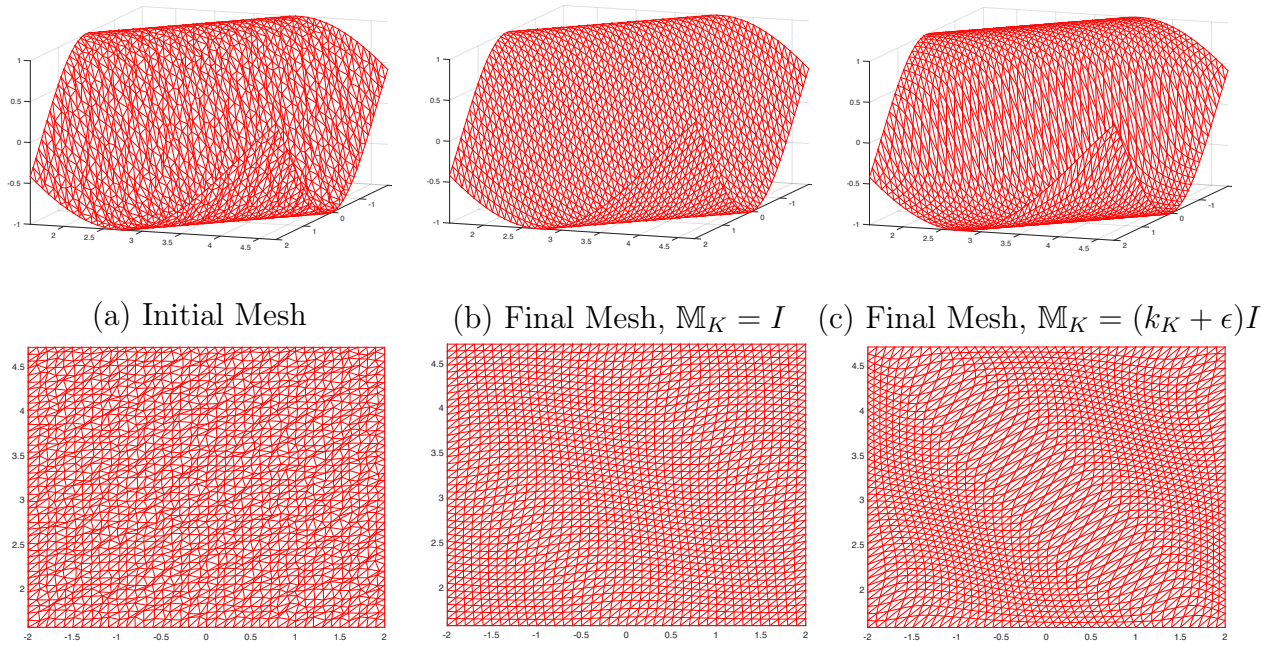


(a) Initial Mesh (b) Final Mesh $\mathbb{M}_K = I$ (c) Final Mesh $\mathbb{M}_K = (k_K + \epsilon)I$

(d) side view of (a) (e) side view of (b) (f) side view of (c)

Figure 3.10: Example 3.5.12. Meshes of $N = 3200$ are plotted for $\Phi(x, y, z) = x^2 + y^2 - 1$.

When $\mathbb{M}_K$ is curvature-based, we see a similar result to Example 3.5.9. That is, Figure 3.11(c) and (f) show that the elements are more concentrated in those regions of the surface with larger curvature, i.e., the dip when $z = -1$ and the hill when $z = 1$. The quality measures with respect to the metric tensor improve from $Q_{eq} = 21.696868$ to $Q_{eq} = 1.634091$ and $Q_{ali} = 6.527829$ to $Q_{ali} = 2.586702$. The final quality measure for the equidistribution condition is close to 1 hence indicating that the final mesh is close to satisfying (3.5). The final quality measure for the alignment condition is not as close to 1 as the equidistribution condition. Recall that θ in the meshing functional (3.15) balances equidistribution and alignment and the choice $\theta = 1/3$ has been used in the computation. Further computations show that increasing θ will improve the alignment quality but worsen the equidistribution quality, and vice versa. This suggests that a perfectly uniform mesh cannot be obtained by minimizing (3.15) for the curvature-based metric tensor for this example.



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

(d) top view of (a) (e) top view of (b) (f) top view of (c)

Figure 3.11: Example 3.5.13. Meshes of $N = 3200$ for the surface $\Phi(x, y, z) = \sin(x + y) - z$.

Finally, we would like to take a look at the changes of I_h and $|K|_{\min}$ along the mesh trajectory. As we recall from Section 3.4, $|K|$ is bounded from below and I_h is decreasing. These can be seen numerically for $\mathbb{M}_K = I$ in Figure 3.12(a) and Figure 3.12(b). Similar to what we saw in Example 3.5.9, Figure 3.12(a) shows that I_h is always decreasing and at around $t = 0.10$ begins to converge. In Figure 3.12(b) we see an initial increase in the $|K|_{\min}$ value and then it begins to converge to $4.64\times10^{-3} \approx \frac{|S|}{N}$ at $t = 0.10$. This initial increase, as discussed above, is due to the nonuniformity of the initial mesh. That is, the initial mesh is very nonuniform and therefore $|K|_{\min}$ can be very small whereas when the mesh is adapted, the mesh becomes more uniform and hence the values of $|K| \approx \frac{|S|}{N}$ become almost identical. This implies that the value of $|K|_{\min}$ is likely to increase as the mesh adapts.

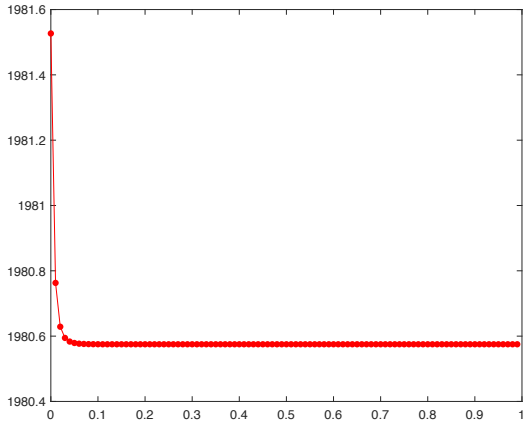
For the case with $\mathbb{M}_K = (k_K + \epsilon)I$, Figure 3.12(c) and (d) show similar findings. In Figure 3.12(c) we see that I_h is decreasing for all time and converging beginning at around $t = 0.15$. Figure 3.12(d) shows $|K|_{\min}$ initially increases then begins to converge to about 2.0×10^{-4}. Furthermore, $|K|_{\min}$ is bounded below by the initial $|K|_{\min}$ value of 0.90×10^{-4}. These numerical results for the curvature based metric tensor further support the theoretical predictions. □

Example 3.5.14. Our final example explores the sphere and ellipsoid defined by an icosahedral initial mesh (see [58] for more details). We begin with the sphere

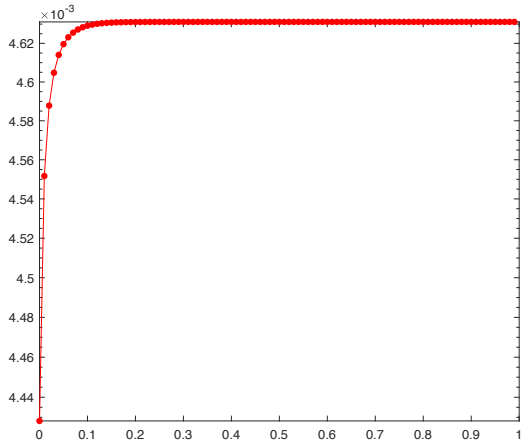
$$\Phi(x, y, z) = x^2 + y^2 + z^2 - 1.$$

For this example we take $N = 1280$. The normal vectors are calculated explicitly for all surfaces in this example.

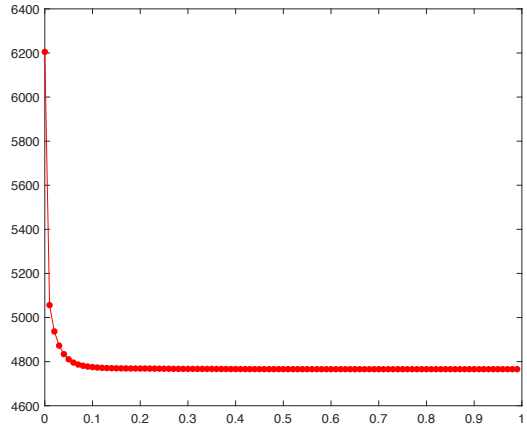
As we see from Figure 3.13(a), the initial mesh is close to being uniform however, there is a very slight difference in the final mesh Figure 3.13(b) when $\mathbb{M}_K = I$. Indeed, this slight adaptation can be seen in the quality measures which change from $Q_{eq} = 1.068461$ and $Q_{ali} = 1.025691$ for the initial mesh to $Q_{eq} = 1.289843$ and $Q_{ali} = 1.025972$ for the final



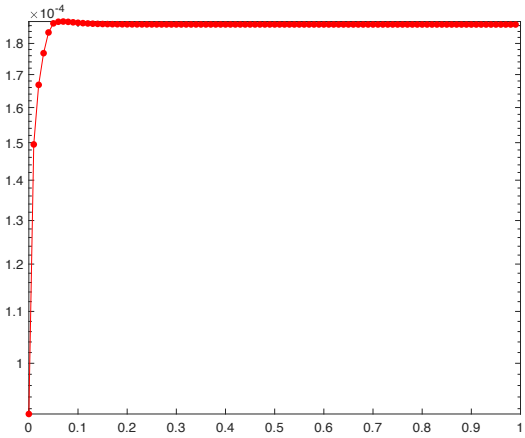
(a) I_h, $\mathbb{M}_K = I$



(b) $|K|_{\min}$, $\mathbb{M}_K = I$



(c) I_h, $\mathbb{M}_K = (k_K + \epsilon)I$



(d) $|K|_{\min}$, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.12: Example 3.5.13. I_h and $K_{\min}$ are plotted as functions of t for $\Phi(x, y, z) = \sin(x + y) - z$.

mesh. The difference in the quality measures indicates that the initial icosahedral mesh is almost uniform and so the moving mesh method does not affect the mesh significantly.

We further this example to consider adaptive meshes for the ellipsoid defined by

$$\Phi(x, y, z) = x^2 + y^2 + \frac{z^2}{4} - 1.$$

We move the mesh on the surface for both $N = 1280$ and $N = 5120$.

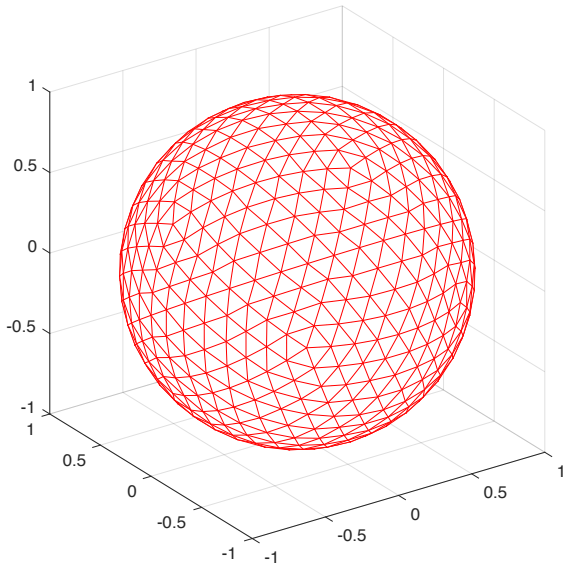
First considering $N = 1280$, Figure 3.14 shows the meshes for this example in two different views. Studying Figure 3.14(a) and Figure 3.14(d), the initial mesh, and Figure 3.14(b) and Figure 3.14 (e), the final mesh with $\mathbb{M}_K = I$, we can see that the final mesh adapts to provide a higher concentration of elements in the middle region of the ellipsoid and fewer elements near the tips. The quality measures improve from $Q_{eq} = 1.724289$ and $Q_{ali} = 1.453207$ for the initial mesh to $Q_{eq} = 1.571401$ and $Q_{ali} = 1.102655$ for the final mesh. Although the initial mesh is close to uniform, the final mesh adapts in such a way to satisfy the equidistribution and alignment condition on the surface. However, this is not an accurate representation of the shape thus we consider a curvature-based metric tensor.

In our numerical experiments, when $\mathbb{M}_K = (k_K + \epsilon)I$ is used, we saw the mesh adapt in a similar way as with the Euclidean metric. This is because the curvature of the ellipsoid does not change significantly at the tips thus not many nodes move there. With this in mind, we altered the curvature-based metric tensor to concentrate more mesh elements at the tips of the ellipsoid by redefining $\mathbb{M}_K$ as

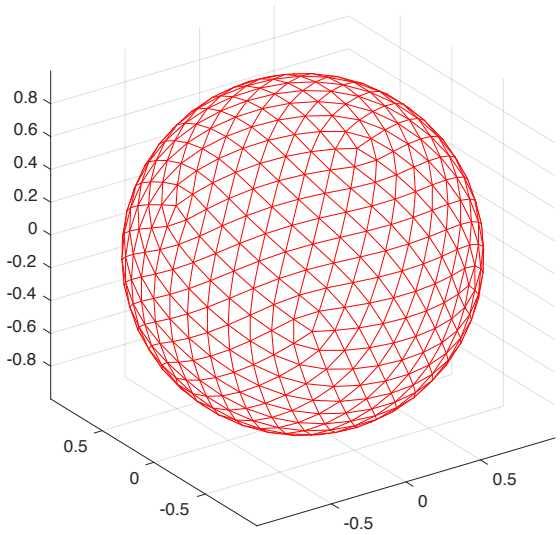
$$\tilde{\mathbb{M}}_K = \mathbb{M}_K + \left(\frac{1}{\sqrt{(z_K - 2)^2 + \epsilon}} + \frac{1}{\sqrt{(z_K + 2)^2 + \epsilon}} \right) I. \tag{3.80}$$

Figure 3.14(c) and Figure 3.14(f) show the final mesh using this altered metric tensor. As we can see, the mesh elements have concentrated at the tips of the ellipsoid thus better representing the shape of the surface. The equidistribution quality measure changes from 1.374300 initially to 1.967482 whereas the alignment quality measure from 1.453207 to 1.262156. Simi-

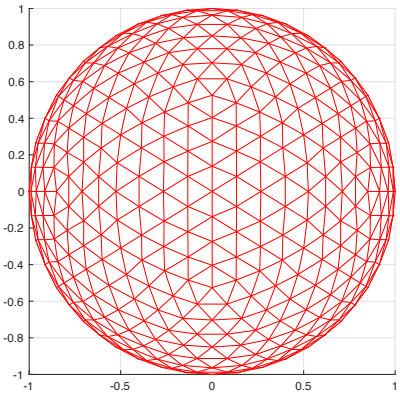
lar results are seen with a finer mesh in Figure 3.15 for both the Euclidean metric and altered curvature-based metric. □



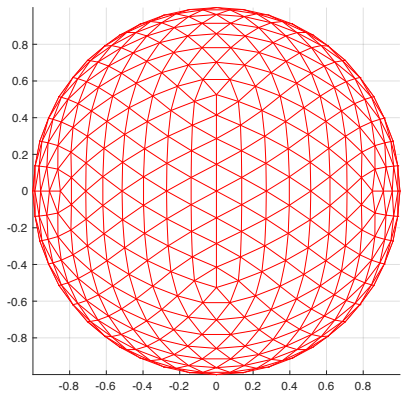
(a) Initial Mesh



(b) Final Mesh $\mathbb{M}_K = I$

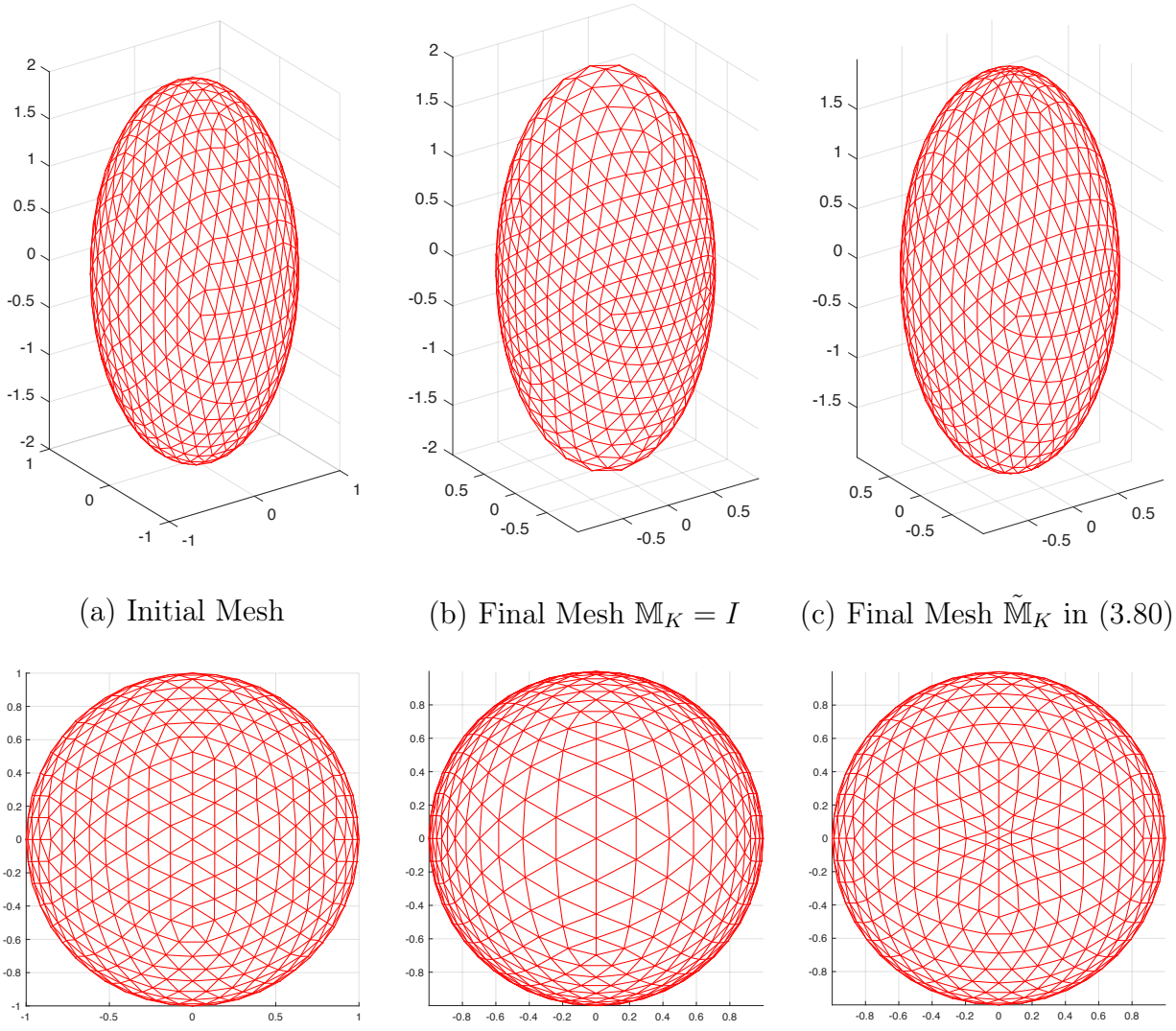


(c) top view of (a)



(d) top view of (b)

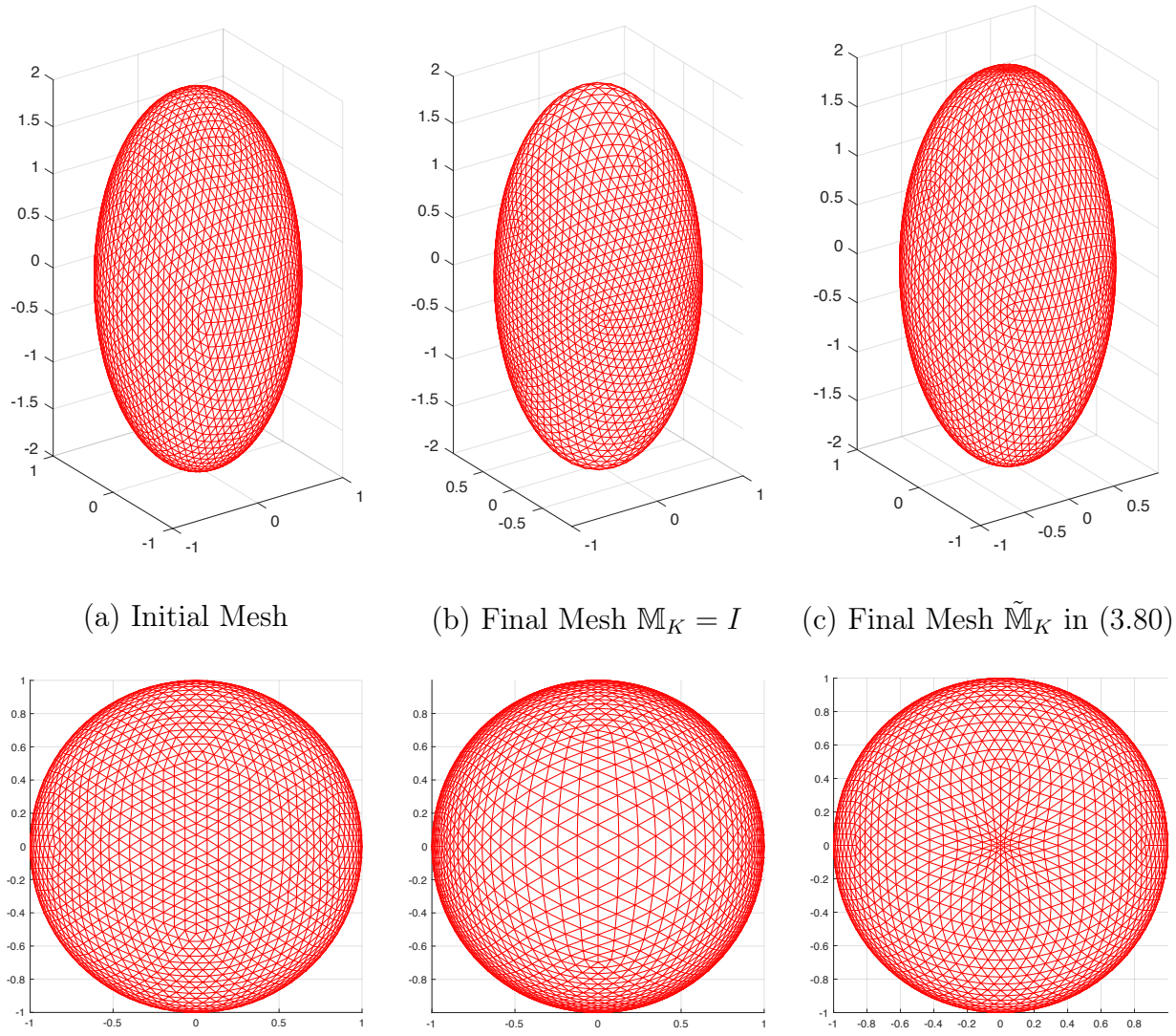
Figure 3.13: Example 3.5.14. Meshes of $N = 1280$ are plotted for $\Phi(x, y, z) = x^2+y^2+z^2-1$.



(a) Initial Mesh (b) Final Mesh $\mathbb{M}_K = I$ (c) Final Mesh $\tilde{\mathbb{M}}_K$ in (3.80)

(d) top view of (a) (e) top view of (b) (f) top view of (c)

Figure 3.14: Example 3.5.14. Meshes of $N = 1280$ are plotted for $\Phi(x, y, z) = x^2+y^2+\frac{z^2}{4}-1$.

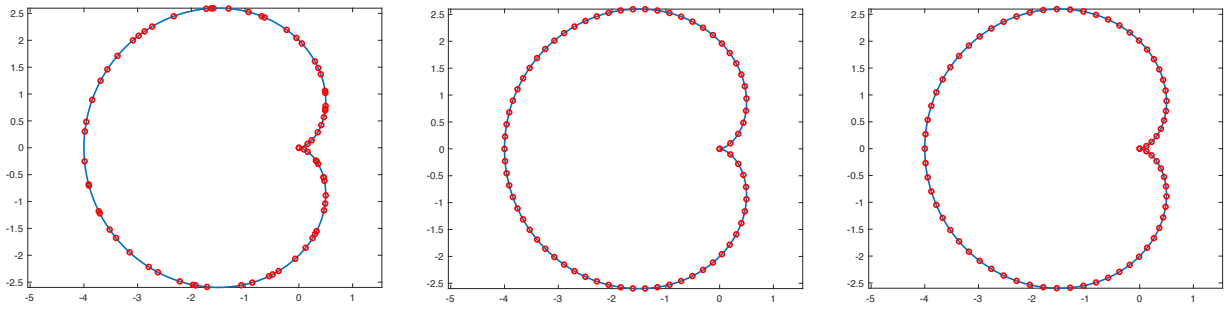


(a) Initial Mesh (b) Final Mesh $\mathbb{M}_K = I$ (c) Final Mesh $\tilde{\mathbb{M}}_K$ in (3.80)

(d) top view of (a) (e) top view of (b) (f) top view of (c)

Figure 3.15: Example 3.5.14. Meshes of $N = 5120$ are plotted for $\Phi(x, y, z) = x^2 + y^2 + \frac{z^2}{4} - 1$.

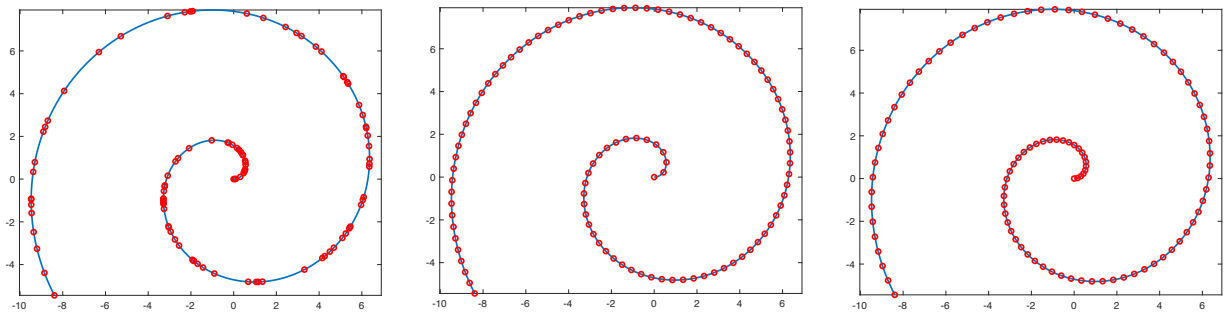
For completeness, we include additional 2D and 3D examples (see Figure 3.16 through Figure 3.29). For all of the two-dimensional examples, unless otherwise specified, we use a total of $N = 100$ elements and take $dt = 0.01$ and $\tau = 0.001$. For all of the 3-dimensional examples, unless otherwise specified, we take $N = 3872$ total elements with $dt = 0.01$ and $\tau = 0.001$. We run to a final time of 1.0 in both cases. As mentioned above, many of the following examples do not have an explicit Cartesian representation, i.e., $\Phi(x, y) = 0$ or $\Phi(x, y, z) = 0$. In these cases we use spline functions to approximate the normal vector required for the MMPDE method and approximate the curvature for the curve [49] and the surface [14] via open source codes. All of the meshes show an accurate adaptation based on the metric tensor used, similar to the examples discussed above.



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

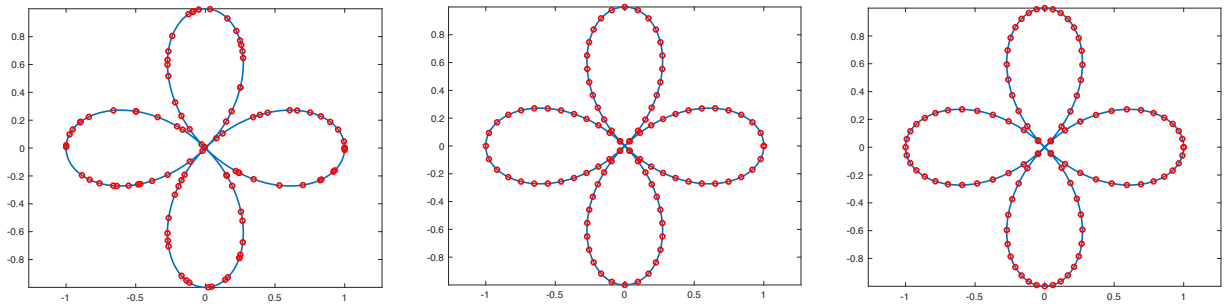
Figure 3.16: Meshes for the cardioid (3.79) with $N = 70$.

[]



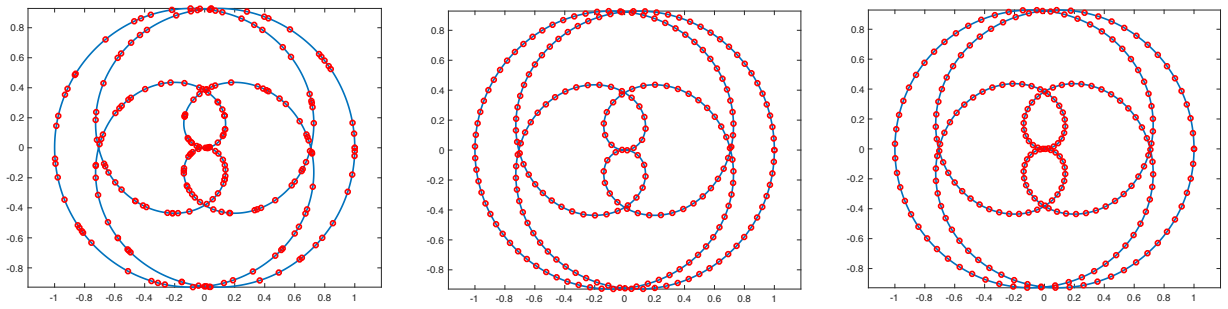
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.17: Meshes for the spiral defined by $x(\theta) = \theta\cos(\theta)$, $y(\theta) = \theta\sin(\theta)$ for $\theta \in [0, 10]$.



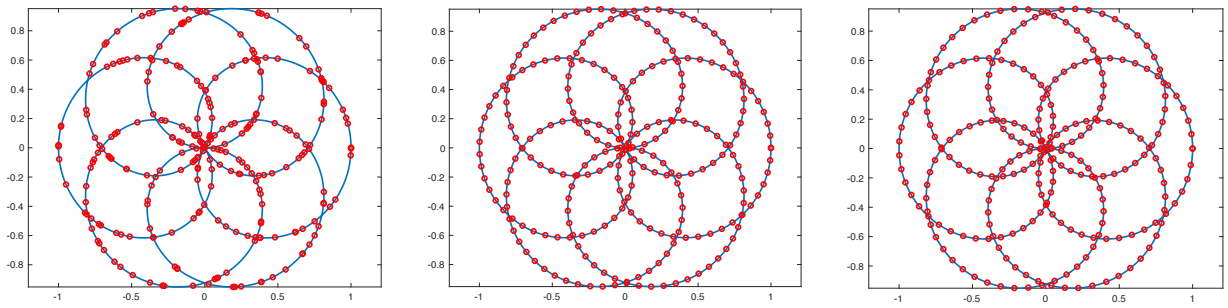
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.18: Meshes for four-petal rose defined by $x(\theta) = \cos(2\theta)\cos(\theta)$, $y(\theta) = \cos(2\theta)\sin(\theta)$ for $\theta \in [0, 2\pi]$.



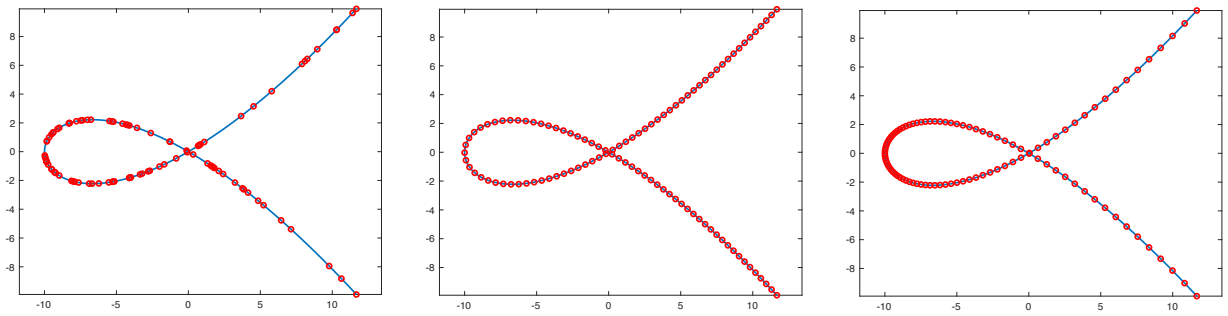
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.19: Meshes for the flower defined by $x(\theta) = \cos\left(\frac{\theta}{4}\right)\cos(\theta)$, $y(\theta) = \cos\left(\frac{\theta}{4}\right)\sin(\theta)$ for $\theta \in [0, 8\pi]$ with $N = 220$.



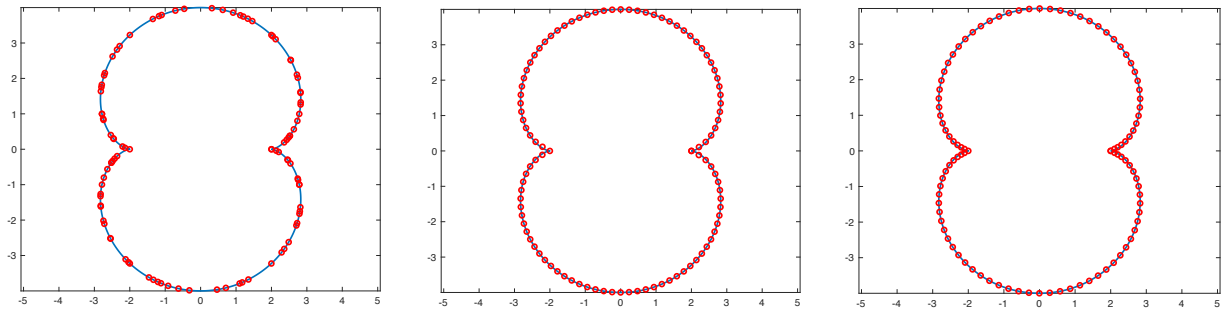
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.20: Meshes for the rose defined by $x(\theta) = \cos\left(\frac{3\theta}{4}\right)\cos(\theta)$, $y(\theta) = \cos\left(\frac{3\theta}{4}\right)\sin(\theta)$ for $\theta \in [0, 8\pi]$ with $N = 260$.



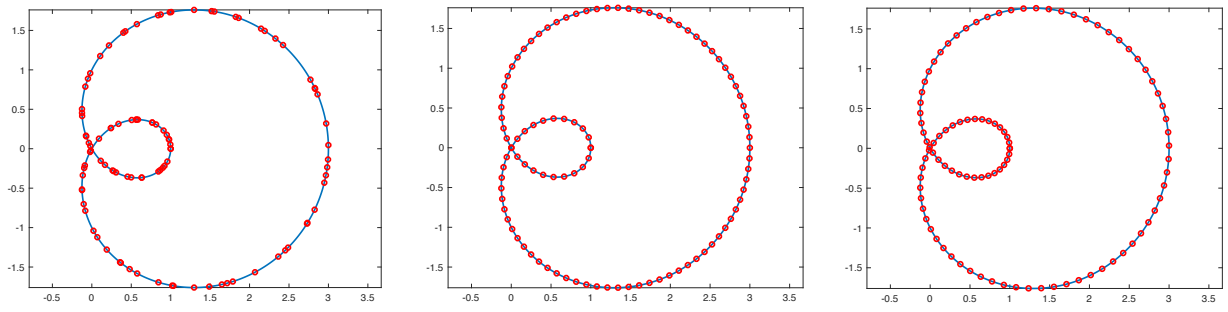
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.21: Meshes for the ribbon defined by $x(\theta) = -10 - 3\theta^2$, $y(\theta) = \theta x(\theta)$ for $\theta \in [-0.85, 0.85]$.



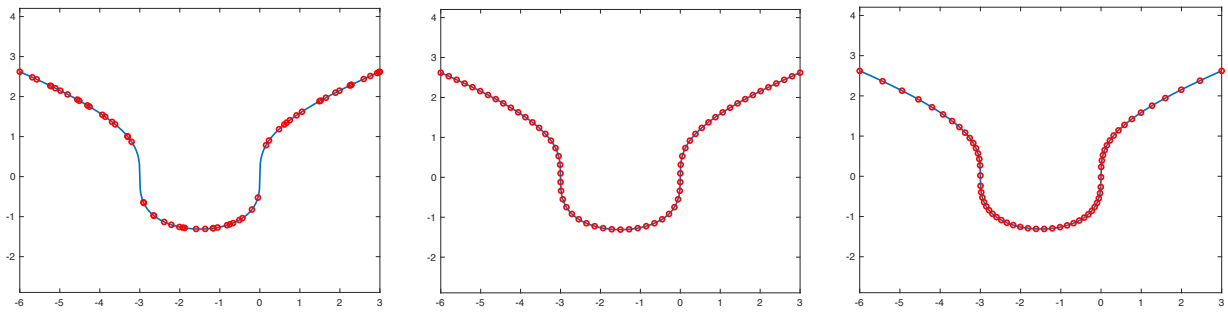
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.22: Meshes for the nephroid defined by $x(\theta) = 3\cos(\theta) - \cos(3\theta)$, $y(\theta) = 3\sin(\theta) - \sin(3\theta)$ for $\theta \in [0, 2\pi]$.



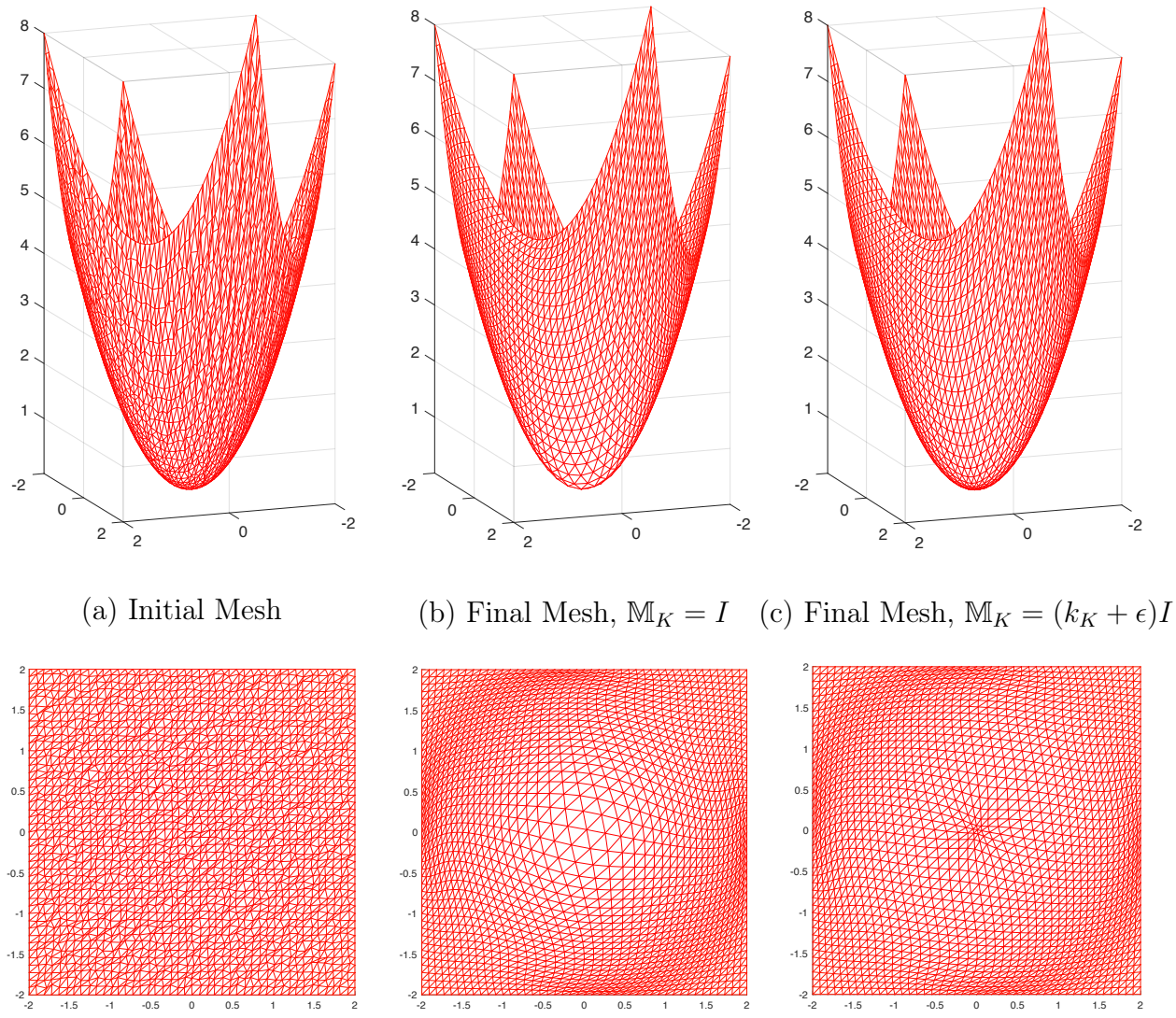
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

Figure 3.23: Meshes for limaçon defined by $x(\theta) = (-1 + 2\cos(\theta))\cos(\theta)$, $y(\theta) = (-1 + 2\cos(\theta))\sin(\theta)$ for $\theta \in [0, 2\pi]$.



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

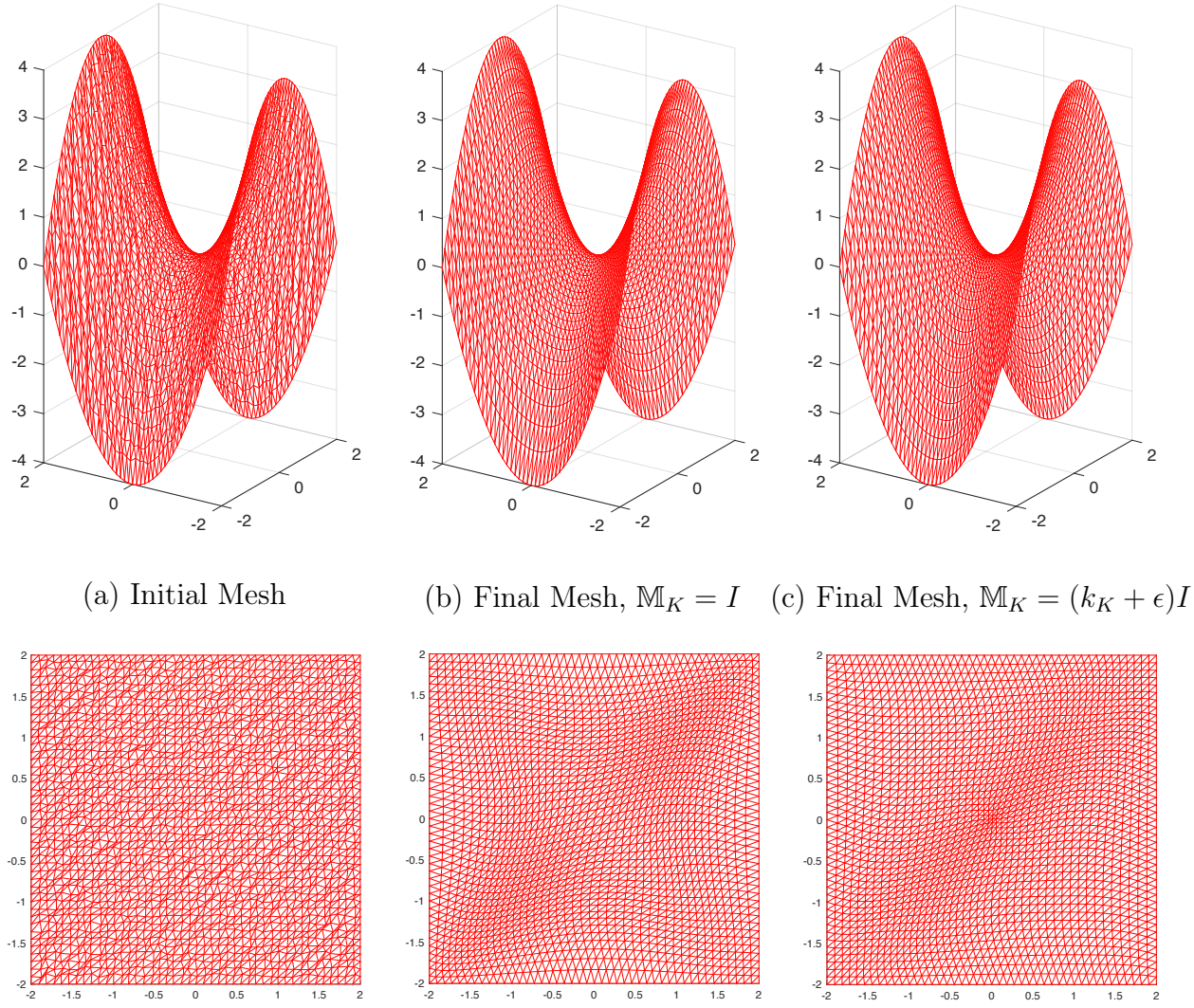
Figure 3.24: Meshes for the dip defined by $\Phi(x, y) = x^2 + 3x - y^3$ with $N = 60$.



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

(d) aerial view of (a) (e) aerial view of (b) (f) aerial view of (c)

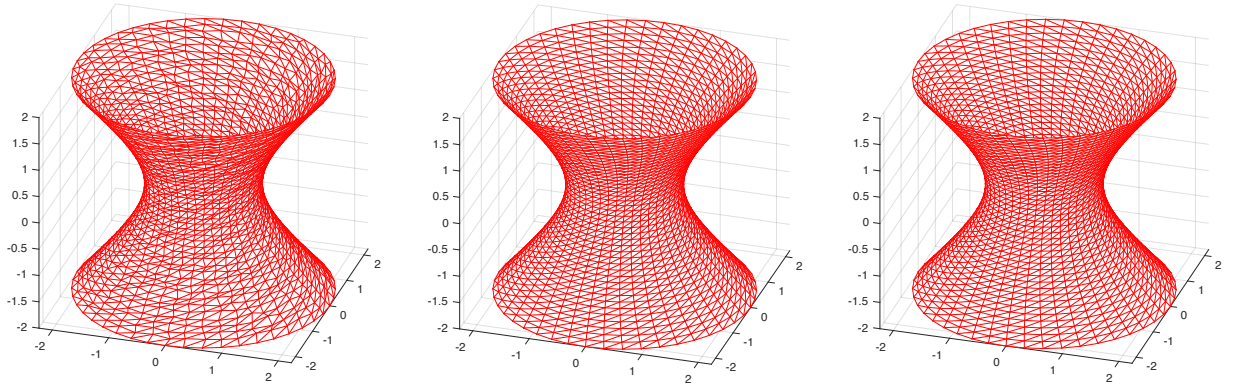
Figure 3.25: Meshes for the paraboloid defined by $\Phi(x, y, z) = x^2 + y^2 - z$.



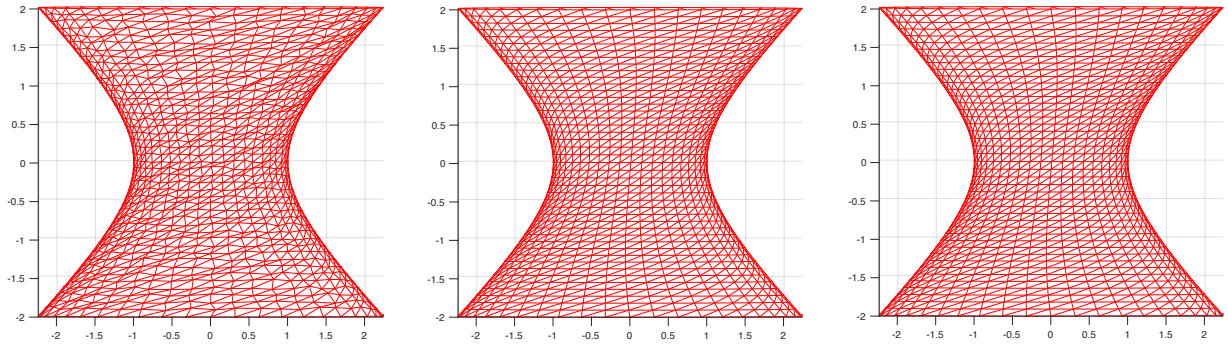
(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

(d) aerial view of (a) (e) aerial view of (b) (f) aerial view of (c)

Figure 3.26: Meshes for the saddle defined by $\Phi(x, y, z) = -x^2 + y^2 - z$.

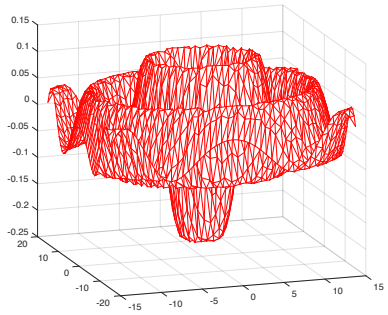


(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

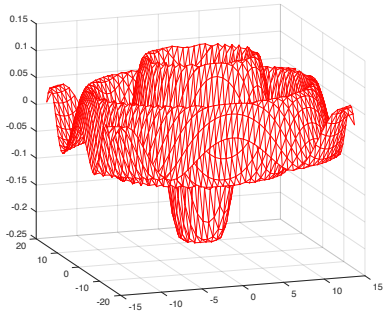


(d) aerial view of (a) (e) aerial view of (b) (f) aerial view of (c)

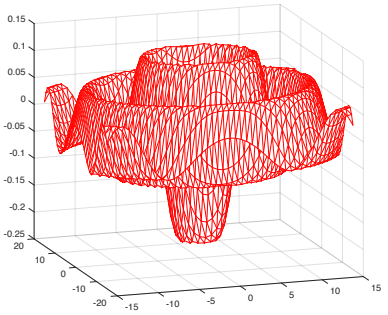
Figure 3.27: Meshes for the hyperboloid defined by $\Phi(x, y, z) = x^2 + y^2 - z^2$.



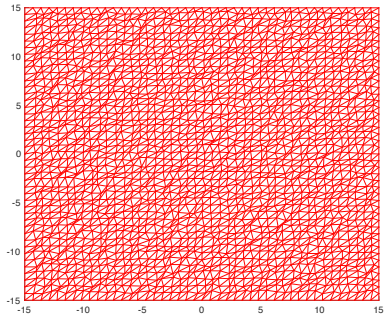
(a) Initial Mesh



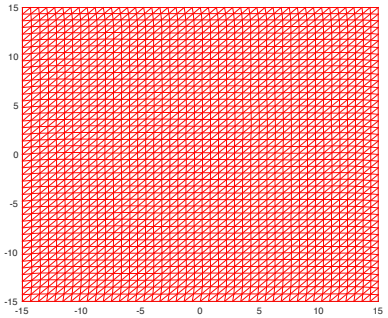
(b) Final Mesh, $\mathbb{M}_K = I$



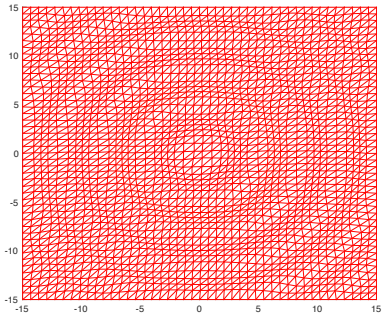
(c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$



(d) aerial view of (a)

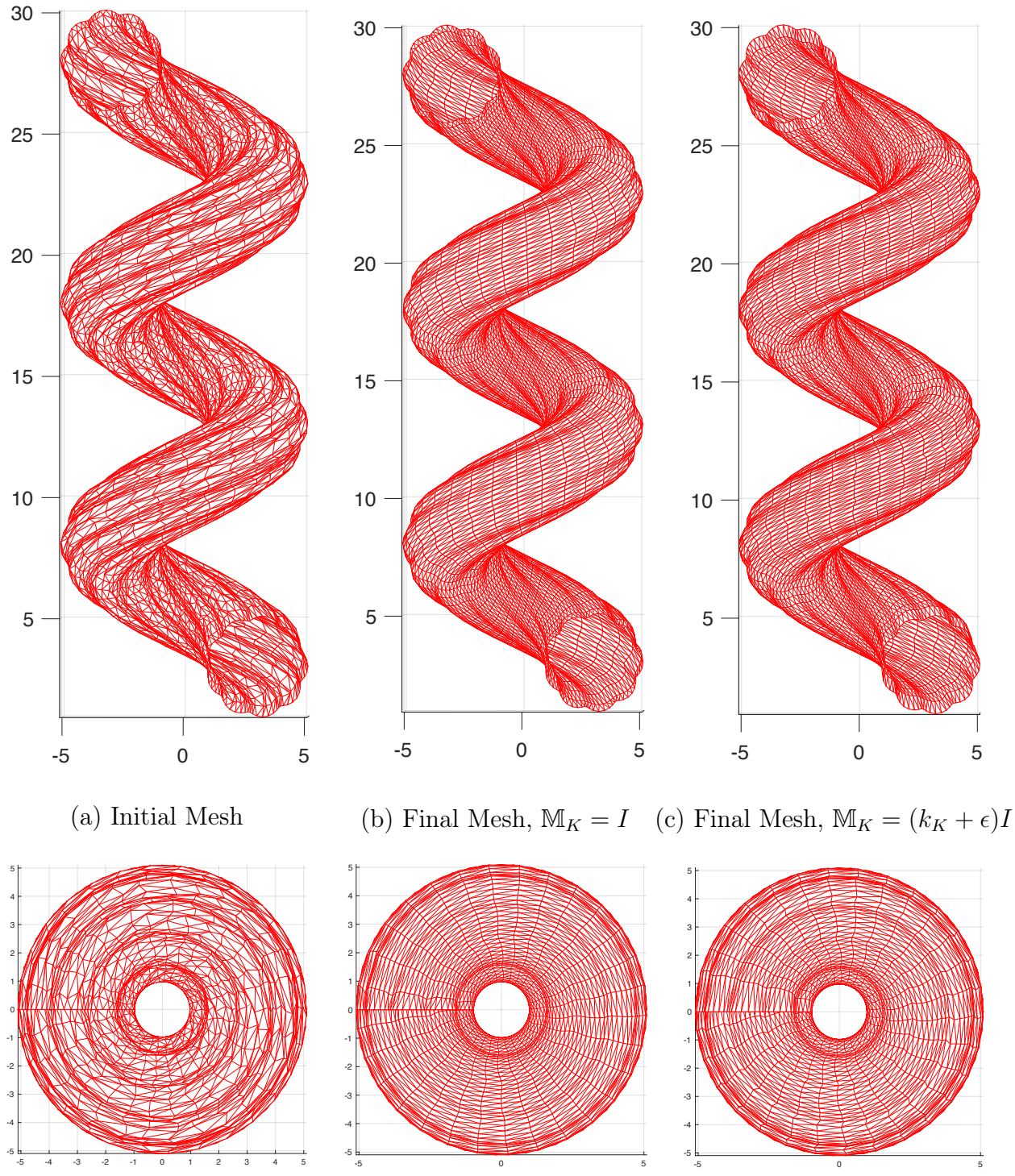


(e) aerial view of (b)



(f) aerial view of (c)

Figure 3.28: Meshes for the ripple defined by $\Phi(x, y, z) = \frac{\sin\left(\sqrt{x^2+y^2+16}\right)}{\sqrt{x^2+y^2+16}}$.



(a) Initial Mesh (b) Final Mesh, $\mathbb{M}_K = I$ (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$

(d) aerial view of (a) (e) aerial view of (b) (f) aerial view of (c)

Figure 3.29: Meshes for the cavatappi with $N = 10952$ defined by
$x(\theta, \gamma) = \left(3 + 2\cos\left(\frac{\pi}{35}\theta\right) + 0.1\cos\left(\frac{2\pi}{7}\theta\right)\right)\cos\left(\frac{\pi}{30}\gamma\right)$,
$y(\theta, \gamma) = \left(3 + 2\cos\left(\frac{\pi}{35}\theta\right) + 0.1\cos\left(\frac{2\pi}{7}\theta\right)\right)\sin\left(\frac{\pi}{30}\gamma\right)$,
$z(\theta, \gamma) = 3 + 2\sin\left(\frac{\pi}{35}\theta\right) + 0.1\sin\left(\frac{2\pi}{7}\theta\right) + \frac{\gamma}{6}$.

3.6 Conclusions for surface mesh adaptation

In this chapter, we have proposed a direct approach for surface mesh movement and adaptation that can be applied to a general surface with or without analytical expressions. We did so by first proving the relation (3.2) between the area of a surface element in a Riemannian metric and the Jacobian matrix of the affine mapping between the reference element and any simplicial surface element. From this we formulated the equidistribution and alignment conditions as given in (3.5) and (3.10), respectively. These two conditions enabled us to formulate a surface meshing functional that is similar to a discrete version of Huang's functional (2.15) for bulk meshes [30]. The surface functional satisfies the coercivity condition (3.19) for $\theta \in (0, 1/2]$ and $p > 1$.

We defined the surface MMPDE (3.42) as the gradient system of the meshing functional, which utilizes surface normal vectors to inherently ensure that the mesh vertices remain on the surface during movement. Equations (3.38) and (3.39) give explicit, compact formulas for the mesh velocities making the time integration of the surface MMPDE (3.44) relatively easy to implement. Moreover, we showed that this surface MMPDE satisfies the energy decreasing property, which is one of the keys to proving Theorem 3.4.1. This theorem is an important theoretical result as it states that the surface mesh remains nonsingular for all time if it is so initially. We then proved Theorem 3.4.2 that states the mesh has limiting meshes, all of which are nonsingular. Finally, both Theorem 3.4.1 and Theorem 3.4.2 were proven for the fully discrete case in Theorem 3.4.3 and Theorem 3.4.4.

A point of emphasis is that the new method is developed directly on surface meshes thus, making no use of any information on surface parameterization. As mentioned, the MMPDE (3.42) only depends on surface normal vectors which can be computed even when the surface has a numerical representation. This allows the new method to be applied to general surfaces with or without explicit parameterization.

The numerical results presented in this work demonstrated that this new approach to surface mesh movement is successful. In all of the examples, the final mesh was seen to be

much more uniform with respect to both cases of the metric tensor $\mathbb{M}_K = I$ and $\mathbb{M}_K = (k_K + \epsilon)I$ which was supported by the mesh quality measures. Moreover, the theoretical properties were numerically verified in Example 3.5.9 and Example 3.5.13 as we showed that I_h is decreasing and $|K|$ is bounded below.

Chapter 4

Conclusions

4.1 Conclusions

Variational mesh adaptation has proven to be an extremely useful tool however, Huang's functional (2.15), which is known to work well in a variety of problems, involves two dimensionless parameters. Although the parameters do not seem to affect the success of the method, the optimal values are still unknown. To overcome this, in Chapter 2 we introduced a new functional based on the equidistribution and alignment conditions. The formulation of the new functional directly combines the equidistribution and alignment conditions into a single condition with only one parameter. In particular, (2.17) does not contain the parameter θ from (2.15) which requires one to try to effectively balance the equidistribution and alignment conditions. Various theoretical results for the new functional that are similar to those of an existing functional were proven. More specifically, the new functional was shown to be coercive (Theorem 2.3.1). With coercivity we could then prove Corollary 2.3.1 which states that the element altitude and volumes of the mesh trajectory of the discrete MMPDE associated with the new functional are bounded away from zero. Moreover, if the initial mesh is nonsingular then the mesh trajectory remains nonsingular for all time. Furthermore, Corollary 2.26 showed the existence of limiting meshes that are critical points of the meshing functional and all nonsingular.

Numerical results in Section 2.4 demonstrated that the new functional is comparable to that of the existing functional. The new functional produces correct mesh concentration and adaptation. In addition, the theoretical findings for the new functional were validated

through numerical results. More specifically, it was shown that the meshing functional was monotonically decreasing and the minimum volume of the mesh element was bounded below as functions of time. From these results, we conclude that the new functional is similar to the existing functional in both numerical performance and theoretical properties.

After studying mesh adaptation for bulk meshes, the natural question that arises is if these methods apply to surfaces. There are a number of mesh adaptation methods for surfaces that have been proposed, all of which have strict requirements. In Chapter 3, we proposed a direct approach for surface mesh movement and adaptation that can be applied to a general surface with or without analytical expressions. We did so by first establishing the relation (3.2) between the area of a surface element in a Riemannian metric and the Jacobian matrix of the affine mapping between the reference element and any simplicial surface element. With (3.2) and the concept of a uniform mesh we were then able to derive the equidistribution and alignment conditions as given in (3.5) and (3.10), respectively. We then combined these two conditions into a single surface meshing functional that is similar to Huang's functional (2.15) for bulk meshes. This surface meshing functional was then shown to satisfy the coercivity condition (3.19) for $\theta \in (0, 1/2]$ and $p > 1$.

In order to minimize this meshing functional, we defined the surface MMPDE (3.42) as a modified gradient system of the meshing functional. This MMPDE utilizes surface normal vectors to project the nodes onto the tangential space and hence ensure that the mesh vertices remain on the surface during movement. We then derived explicit, compact formulas for the mesh velocities given by (3.38) and (3.39). This makes implementation and time integration of the surface MMPDE (3.44) relatively easy. Furthermore, we showed that this surface MMPDE satisfies the energy decreasing property, a key property to proving Theorem 3.4.1 which states that if the surface mesh is initially nonsingular then it will remain nonsingular for all time. We then proved Theorem 3.4.2 stating the existence of a limiting meshes, all of which are nonsingular. Finally, we extended these two theorems to the fully discrete case as given in Theorem 3.4.3 and Theorem 3.4.4.

A point of emphasis is that the new method makes no use of any information on surface parameterization since it is developed directly on surface meshes. As mentioned, the MMPDE (3.42) only depends on surface normal vectors which can be computed even when the surface has a numerical representation. This allows the new method to be applied to general surfaces with or without explicit parameterization.

The numerical results presented in Section 3.5 demonstrated that this new approach to surface mesh movement is successful. In all of the examples, the final mesh was seen to be much more uniform with respect to both cases of the metric tensor $\mathbb{M}_K = I$ and $\mathbb{M}_K = (k_K + \epsilon)I$ which was supported by the mesh quality measures. Moreover, the theoretical properties were numerically verified in Example 3.5.9 and Example 3.5.13 as we showed that I_h is decreasing and $|K|$ is bounded below. It was also shown that, assuming the initial mesh is fine enough, spline functions can be used successfully to approximate the normal vector required in computation. This proves the theoretical assumption that meshes without explicit parameterization work well with our method without any additional assumptions or requirements. Finally, throughout various examples in two-dimensions we have also showed that the proposed method successfully adapts meshes with crossings.

4.2 Future Research

In order to better understand the performance of the new functional, more work and a variety of examples are necessary. Specifically, one of the main disadvantages of the new functional is that it is not convex whereas the existing functional is known to be polyconvex and can be made convex with the special choice of the parameter θ ($\theta = 1/2$). With this in mind, it is hard to say how the non-convexity of the new functional affects the numerics. For the examples we tested, we did not experience any difficulty with computation or CPU time but this may be a topic for further investigations.

The future research for the surface mesh method is a natural continuation of the work described above. The first goal is to implement more numerical examples with various

intricate curves and surfaces. Moreover, most of the numerical examples thus far are limited to having fixed boundary points. It would be useful to provide more interesting examples for which the points are allowed to move along the boundary. The monitor functions we used in the examples are limited to simple scalar metric tensors. It will be interesting to see how an anisotropic metric tensor, such as one based on the shape map, affects mesh movement and quality. It is then necessary to develop the method to include adapting the nodes to improve the solutions of PDEs defined on surfaces. This would require concentrating mesh elements to account for the curvature of the surface as well as the solution of the PDE. The final step in this research is to expand the method to moving surfaces which will be extremely useful for a variety of applications.

References

[1] T. Apel. Anisotropic finite elements: local estimates and applications. B. G. Teubner Stuttgart, 1999.

[2] T. Apel and M. Dobrowolski. Anisotropic interpolation with applications to the finite element method. *Computing*, 47: 277-293, 1992.

[3] J.-D. Boissonnat, F. Chazal, and M. Yvinec. Geometric and Topological Inference. Cambridge University Press, 2018.

[4] H. Borouchaki, P. L. George, P. Hecht, P. Laug, and E. Saletl. Delaunay mesh generation governed by metric specification: Part I. Algorithms. *Finite Elem. Anal. Des.*, 25: 61-83, 1997.

[5] F. J. Bossen and P. S. Heckbert. A pliant method for anisotropic mesh generation. *5th International Meshing Roundtable, Sandia National Laboratories*, 63-74, 1996.

[6] J. U. Brackbill and J. S. Saltzman. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.*, 46: 342-368, 1982.

[7] P. Browne, C. J. Budd, M. Cullen, and H. Weller. Mesh adaptation on the sphere using optimal transport and the numerical solution of a Monge-Ampére type equation. *J. Comput. Phys.*, 308: 102-123, 2016.

[8] G. F. Carey. Computational Grids: Generation, Adaptation, and Solution Strategies. Taylor and Francis, Washinton, DC, 1997.

[9] M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *Int. J. Numer. Meth. Fluids*, 25: 475-491, 1997.

[10] J. Cavalcante-Neto, M. Freitas, D. Siqueira, and C. Vidal. An adaptive parametric surface mesh generation method guided by curvature. Proceedings of the 22nd International Meshing Roundtable, Springer International Publishing. 425-443, 2014.

[11] B. Crestel, R. D. Russell, and S. Ruuth. Moving mesh methods on parametric surfaces. *Procedia Engineering*, 124: 148-160, 2015.

[12] E. F. D'Azevedo and R. B. Simpson. On optimal triangular meshes for minimizing the gradient error. *Numer. Math.*, 59: 321-348, 1991.

[13] F. Dassi, S. Perotto, H. Si, and T. Streckenbach. A priori anisotropic mesh adaptation driven by a higher dimensional embedding. *Computer-Aided Design*, 85: 111-122, 2017.

[14] A. Dastan. Gaussian and mean curvatures calculation on a triangulated 3D surface. *MathWorks*, https://www.mathworks.com/matlabcentral/fileexchange/61136-gaussian-and-mean-curvatures-calculation-on-a-triangulated-3d-surfaces_tid=FX_rc2_behav. Updated 1/21/2017.

[15] A. Demlow and G. Dziuk. An adaptive finite element method for the Laplace-Beltrami Operator on implicitly defined surfaces. *SIAM J. Numer. Anal.*, 45: 421-442, 2007.

[16] A. S. Dvinsky. Adaptive grid generation from harmonic maps on Riemannian manifolds. *J. Comput. Phys.*, 95: 450-476, 1991.

[17] G. Dziuk. Finite elements for the Beltrami operator on arbitrary surfaces. Springer, New York. Lecture Notes in Math., Vol. 1357, 2006.

[18] G. Dziuk and C. Elliott. Finite element methods for surface PDEs. *Acta Numerica*, 22: 289-396, 2013.

[19] G. Dziuk and C. Elliott. Surface finite elements for parabolic equations. *J. Comput. Math.*, 25: 385-407, 2007.

[20] J. Emert and R. Nelson. Volume and surface area for polyhedra and polytopes. *Math. Mag.*, 70: 365-371, 1997.

[21] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numer. Math.*, 89: 641-667, 2001.

[22] L. A. Freitag and C. Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Numer. Meth. Engrg.*, 40: 3979-4002, 1997.

[23] M. J. Gander and R. D. Haynes. Domain decomposition approaches for mesh generation via the equidistribution principle. *SIAM J. Numer. Anal.*, 50: 2111-2135, 2012.

[24] F. Goldman. Curvature formulas for implicit curves and surfaces. *Comp. Aided Geometric Design.*, 22: 632-658, 2005.

[25] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles. *Int. J. Numer. Meth. Fluids*, 32: 725-744, 2000.

[26] E. Hairer and C. Lubich. Energy-diminishing integration of gradient systems. *IMA J. Numer. Anal.*, 34: 452-461, 2014.

[27] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996. Stiff and differential-algebraic problems.

[28] R. D. Haynes and F. Kwok. Discrete analysis of domain decomposition approaches for mesh generation via the equidistribution principal. *Math. Comp.* 86: 233-273, 2017.

[29] E. Hairer and C. Lubich. Energy-diminishing integration of gradient systems. *IMA J. Numer. Anal.,* 34: 452-461, 2014.

[30] W. Huang. Variational mesh adaptation: isotropy and equidistribution. *J. Comput. Phys.*, 174: 903-924, 2001.

[31] W. Huang. Metric tensors for anisotropic mesh generation. *J. Comput. Phys.*, 204: 633-665, 2005.

[32] W. Huang and L. Kamenski. On the mesh nonsingularity of the moving mesh PDE method. *Math. Comp.*, 87:1887-1911, 2018.

[33] W. Huang and L. Kamenski. A geometric discretization and a simple implementation for variational mesh generation and adaptation. *J. Comput. Phys.*, 301: 322-337, 2015.

[34] W. Huang, L. Kamenski, and R. D. Russell. A comparative numerical study of meshing functionals for variational mesh adaptation. *J. Math. Study*, 48: 168-186, 2015.

[35] W. Huang, L. Kamenski, and H. Si. Mesh smoothing: an MMPDE approach. Research note at the 24th International Meshing Roundtable, WIAS Preprint No. 2130, 2015.

[36] W. Huang, Y. Ren, and R. D. Russell. Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle. *SIAM J. Numer. Anal.*, 31: 707-730, 1994.

[37] 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.

[38] W. Huang and R. D. Russell. Adaptive Moving Mesh Methods. Springer, New York. Applied Mathematical Sciences Series, Vol. 174, 2011.

[39] W. Huang, Y. Ren, and R. D. Russell. Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle. *SIAM J. Numer. Anal.*, 31: 707-730, 1994.

[40] P. M. Knupp. Jacobian-weighted elliptic grid generation. *SIAM J. Sci. Comput.*, 17: 1475-1490, 1996.

[41] 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.

[42] P. Knupp and S. Steinberg. Fundamentals of Grid Generation. CRC Press, Boca Raton, 1994.

[43] A. Kolasinski and W. Huang. A new function for variational mesh generation and adaptation based on equidistribution and alignment conditions. *Comput. Math. Appl.*, 75: 2044-2058, 2018.

[44] B. Lévy and N. Bonneel. Variational anisotropic surface meshing with Voronoi parallel linear enumeration. The 21st International Meshing Roundtable, Springer-Verlag, 344-366, 2012.

[45] R. Li, T. Tang, and P. W. Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *J. Comput. Phys.*, 170: 562-588, 2001.

[46] V. D. Liseikin. Grid Generation Methods. Springer, Berlin, 1999.

[47] G. MacDonald, J. A. Mackenzie, M. Nolan, and R. H. Insall. A computational method for the coupled solution of reaction-diffusion equations on evolving domains and manifolds: Application to a model of cell migration and chemotaxis. *J. Comput. Phys.*, 309: 207-226, 2016.

[48] A. T. T. McRae, C. J. Cotter, and C. J. Budd. Optimal-transport-based mesh adaptivity on the plane and sphere using finite elements. *SIAM J. Sci. Comput.*, 40: A1121-A1148, 2018.

[49] A. Mjaavatten. Curvature of a 2D or 3D curve. *MathWorks*, https://www.mathworks.com/matlabcentral/fileexchange/69452-curvature-of-a-2d-or-3d-curve. Updated 11/26/2018.

[50] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comput. Phys.*, 72: 449-466, 1997.

[51] J. D. Pryce. On the convergence of iterated remeshing. *IMA J. Numer. Anal.*, 9: 315-335, 1989.

[52] J. Remacle, X. Li, M. S. Shephard, and J. E. Flaherty. Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods. *Internat. J. Numer. Methods Engrg.*, 62: 899-923, 2005.

[53] H. Schlichtkrull. Curves and Surfaces- Lecture Notes for Geometry 1. *Department of Mathematics. University of Copenhagen.*

[54] A.M. Stuart and A.R. Humphries. Dynamical systems and numerical analysis. Volume 2 of Cambridge Monographs on Applied and Computational Mathematics. *Cambridge University Press,* Cambridge, 1996.

[55] J. F. Thompson, Z. A. Warsi, and C. W. Mastin. Numerical Grid Generation: Foundations and Applications. North-Holland, New York, 1985.

[56] N. Tuncer and A. Madzvamuse. Projected finite elements for systems of reaction-diffusion equations on closed evolving spheroidal surfaces. *Comm. Comp. Phys.*, 21: 718-747, 2017.

[57] Yu. V. Vassilevski, V. G. Dyadechko, and K. N. Lipnikov. Hessian-based anisotropic mesh adaptation in domains with discrete boundaries. *Russian J. Numer. Anal. Math. Modelling*, 20: 391-402, 2005.

[58] N. Wang and J. Lee. Geometric properties of the icosahedral-hexagonal grid on the two-sphere. *SIAM J. Sci. Comput.*, 33: 2536-2559, 2011.

[59] A. M. Winslow. Adaptive mesh zoning by the equipotential method. Technical Report UCID-19062, Lawrence Livermore Laboratory, 1981.

[60] X. Xu, W. Huang, R. D. Russell, and J. F. Williams. Convergence of de Boor's algorithm for generation of equidistributing meshes. *IMA J. Numer. Anal.*, 31: 558-596, 2011.

[61] S. Yamakawa and K. Shimada. High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. *9th International Meshing Roundtable, Sandia National Laboratories*, 263-273, 2000.

Appendix A

Appendix

A.1 Proof of Corollary 2.2.1

Corollary A.1.1. *Assume $\mathbb{M}$ is independent of A. Then*

$$\frac{\partial tr(A\mathbb{M}A^T)}{\partial A} = 2\mathbb{M}A^T, \tag{A.1}$$

$$\frac{\partial tr(A^{-T}\mathbb{M}^{-1}A^{-1})}{\partial A} = -2A^{-1}A^{-T}\mathbb{M}^{-1}A^{-1}. \tag{A.2}$$

Moreover, assume A is independent of $\mathbb{M}$. Then

$$\frac{\partial tr(A\mathbb{M}A^T)}{\partial \mathbb{M}} = A^TA, \tag{A.3}$$

$$\frac{\partial tr(A\mathbb{M}^{-1}A^T)}{\partial \mathbb{M}} = -\mathbb{M}^{-1}A^TA\mathbb{M}^{-1}. \tag{A.4}$$

Proof. Let t be an entry of A. Then using Lemma 2.2.1 and Lemma 2.2.2, we have

$$\begin{aligned}
\frac{\partial \mathrm{tr}(A\mathbb{M}A^T)}{\partial t} &= \mathrm{tr}\left(\frac{\partial \mathrm{tr}(A\mathbb{M}A^T)}{\partial (A\mathbb{M}A^T)}\frac{\partial (A\mathbb{M}A^T)}{\partial t}\right) = \mathrm{tr}\left(\frac{\partial (A\mathbb{M}A^T)}{\partial t}\right) \\
&= \mathrm{tr}\left(\frac{\partial A}{\partial t}\mathbb{M}A^T + A\mathbb{M}\frac{\partial A^T}{\partial t}\right) = \mathrm{tr}\left(\frac{\partial A}{\partial t}\mathbb{M}A^T\right) + \mathrm{tr}\left(A\mathbb{M}\frac{\partial A^T}{\partial t}\right) \\
&= \mathrm{tr}\left(\mathbb{M}A^T\frac{\partial A}{\partial t}\right) + \mathrm{tr}\left(\frac{\partial A}{\partial t}\mathbb{M}A^T\right) = \mathrm{tr}\left(2\mathbb{M}A^T\frac{\partial A}{\partial t}\right).
\end{aligned}$$

Applying the chain rule (2.24) gives (A.1).

Alternatively, using Lemma 2.2.1 and Lemma 2.2.4, we have

$$\begin{aligned}\frac{\partial \mathrm{tr}(A^{-T}\mathbb{M}^{-1}A^{-1})}{\partial t} &= \mathrm{tr}\left(\frac{\partial \mathrm{tr}(A^{-T}\mathbb{M}^{-1}A^{-1})}{\partial A^{-T}}\frac{\partial A^{-T}}{\partial t}\right)\\ &= \mathrm{tr}\left(2\mathbb{M}^{-1}A^{-1}\frac{\partial A^{-T}}{\partial t}\right) = \mathrm{tr}\left(2\mathbb{M}^{-1}A^{-1}\left(-A^{-1}\frac{\partial A^{T}}{\partial t}A^{-T}\right)\right)\\ &= \mathrm{tr}\left(-2A^{-1}\frac{\partial A}{\partial t}A^{-1}A^{-T}\mathbb{M}^{-1}\right) = \mathrm{tr}\left(-2A^{-1}A^{-T}\mathbb{M}^{-1}A^{-1}\frac{\partial A}{\partial t}\right),\end{aligned}$$

which gives (A.2).

In a similar fashion, using Lemma 2.2.1 and Lemma 2.2.2, we have

$$\begin{aligned}\frac{\partial \mathrm{tr}(A\mathbb{M}A^{T})}{\partial t} &= \mathrm{tr}\left(\frac{\partial \mathrm{tr}(A\mathbb{M}A^{T})}{\partial (A\mathbb{M}A^{T})}\frac{\partial (A\mathbb{M}A^{T})}{\partial t}\right) = \mathrm{tr}\left(\frac{\partial (A\mathbb{M}A^{T})}{\partial t}\right)\\ &= \mathrm{tr}\left(A\frac{\partial \mathbb{M}}{\partial t}A^{T}\right) = \mathrm{tr}\left(A^{T}A\frac{\partial \mathbb{M}}{\partial t}\right),\end{aligned}$$

which gives (A.3).

Finally, using Lemma 2.2.1, Lemma 2.2.2, and Lemma 2.2.4, we have

$$\begin{aligned}\frac{\partial \mathrm{tr}\left(A\mathbb{M}^{-1}A^{T}\right)}{\partial t} &= \mathrm{tr}\left(\frac{\partial \mathrm{tr}\left(A\mathbb{M}^{-1}A^{T}\right)}{\partial \left(A\mathbb{M}^{-1}A^{T}\right)}\frac{\partial \left(A\mathbb{M}^{-1}A^{T}\right)}{\partial t}\right)\\ &= \mathrm{tr}\left(A\frac{\partial \mathbb{M}^{-1}}{\partial t}A^{T}\right)\\ &= \mathrm{tr}\left(-A\mathbb{M}^{-1}\frac{\partial \mathbb{M}}{\partial t}\mathbb{M}^{-1}A^{T}\right)\\ &= \mathrm{tr}\left(-\mathbb{M}^{-1}A^{T}A\mathbb{M}^{-1}\frac{\partial \mathbb{M}}{\partial t}\right),\end{aligned}$$

which gives (A.4). □

A.2 Derivatives of existing functional for bulk mesh

Consider (2.19) for the existing functional (2.15), i.e.,

$$I_h = \sum_{K \in \mathcal{T}_h} |K| G\left(\mathbb{J}, \det(\mathbb{J}), \mathbb{M}\right),$$

where

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

To find $\frac{\partial G}{\partial \mathbb{J}}$ consider

$$\begin{aligned}
\frac{dG}{dt} &= \mathrm{tr}\left(\frac{\partial G}{\partial \mathbb{J}} \frac{\partial \mathbb{J}}{\partial t}\right) \\
&= \mathrm{tr}\left(\theta\ |K| \det(\mathbb{M})^{\frac{1}{2}} \frac{\partial \mathrm{tr}\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T\right)^{\frac{dp}{2}}}{\partial \mathbb{J}} \frac{\partial \mathbb{J}}{\partial t}\right) \\
&= \mathrm{tr}\left(\frac{dp}{2}\theta\ |K| \det(\mathbb{M})^{\frac{1}{2}} \mathrm{tr}\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T\right)^{\frac{dp-1}{2}} \frac{\partial \mathrm{tr}\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T\right)}{\partial \mathbb{J}} \frac{\partial \mathbb{J}}{\partial t}\right) \\
&= \mathrm{tr}\left(dp\theta\ |K| \det(\mathbb{M})^{\frac{1}{2}} \mathrm{tr}\left(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T\right)^{\frac{dp-1}{2}} \mathbb{M}^{-1}\mathbb{J}^T \frac{\partial \mathbb{J}}{\partial t}\right),
\end{aligned}$$

where we used Corollary 2.2.1. Similarly, to find $\frac{\partial G}{\partial \det(\mathbb{J})}$ consider

$$\begin{aligned}
\frac{dG}{dt} &= \mathrm{tr}\left(\frac{\partial G}{\partial \det(\mathbb{J})} \frac{\partial \det(\mathbb{J})}{\partial t}\right) \\
&= \mathrm{tr}\left((1-2\theta)|K| d^{\frac{dp}{2}} \det(\mathbb{M})^{\frac{1-p}{2}} \frac{\partial \det(\mathbb{J})^p}{\partial \det(\mathbb{J})} \frac{\partial \det(\mathbb{J})}{\partial t}\right) \\
&= \mathrm{tr}\left(p(1-2\theta)|K| d^{\frac{dp}{2}} \det(\mathbb{M})^{\frac{1-p}{2}} \det(\mathbb{J})^{p-1} \frac{\partial \det(\mathbb{J})}{\partial t}\right).
\end{aligned}$$

Finally, to find $\frac{\partial G}{\partial \mathbb{M}}$ consider the first term of G, denoted as G_1, i.e.,

$$G_1 = \theta \det(\mathbb{M})^{\frac{1}{2}} \left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}}.$$

Then

$$\begin{aligned}
\frac{dG_1}{dt} &= \operatorname{tr}\left(\frac{\partial G_1}{\partial \mathbb{M}}\frac{\partial \mathbb{M}}{\partial t}\right) \\
&= \operatorname{tr}\left(\theta|K|\left(\frac{\partial \det(\mathbb{M})^{\frac{1}{2}}}{\partial \mathbb{M}}\left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}} + \det(\mathbb{M})^{\frac{1}{2}}\frac{\partial\left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}}}{\partial \mathbb{M}}\right)\frac{\partial \mathbb{M}}{\partial t}\right)\right) \\
&= \operatorname{tr}\left(\frac{1}{2}\theta|K|\det(\mathbb{M})^{-\frac{1}{2}}\frac{\partial \det(\mathbb{M})}{\partial \mathbb{M}}\left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}}\frac{\partial \mathbb{M}}{\partial t}\right) \\
&\quad + \operatorname{tr}\left(\frac{dp}{2}\theta|K|\det(\mathbb{M})^{\frac{1}{2}}\left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}-1}\frac{\partial \operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)}{\partial \mathbb{M}}\frac{\partial \mathbb{M}}{\partial t}\right).
\end{aligned}$$

Then by Lemma 2.2.3 and Corollary 2.2.1 we have

$$\begin{aligned}
\frac{dG_1}{dt} &= \operatorname{tr}\left(\frac{1}{2}\theta|K|\det(\mathbb{M})^{\frac{1}{2}}\left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}}\mathbb{M}^{-1}\frac{\partial \mathbb{M}}{\partial t}\right) \\
&\quad - \operatorname{tr}\left(\frac{dp}{2}\theta|K|\det(\mathbb{M})^{\frac{1}{2}}\left(\operatorname{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}-1}\mathbb{M}^{-1}\mathbb{J}^T\mathbb{J}\mathbb{M}^{-1}\frac{\partial \mathbb{M}}{\partial t}\right).
\end{aligned}$$

Considering now the second term of G, denoted as G_2,i.e.,

$$G_2 = (1-2\theta)d^{\frac{dp}{2}}\det(\mathbb{M})^{\frac{1-p}{2}}\det(\mathbb{J})^p,$$

we have

$$\begin{aligned}
\frac{dG_2}{dt} &= \operatorname{tr}\left(\frac{\partial G_2}{\partial \mathbb{M}}\frac{\partial \mathbb{M}}{\partial t}\right) \\
&= \operatorname{tr}\left((1-2\theta)|K|d^{\frac{dp}{2}}\det(\mathbb{J})^p\frac{\partial \det(\mathbb{M})^{\frac{1-p}{2}}}{\partial \mathbb{M}}\frac{\partial \mathbb{M}}{\partial t}\right) \\
&= \operatorname{tr}\left(\frac{1-p}{2}(1-2\theta)|K|d^{\frac{dp}{2}}\det(\mathbb{J})^p\det(\mathbb{M})^{\frac{1-p}{2}-1}\frac{\partial \det(\mathbb{M})}{\partial \mathbb{M}}\frac{\partial \mathbb{M}}{\partial t}\right) \\
&= \operatorname{tr}\left(\frac{1-p}{2}(1-2\theta)|K|d^{\frac{dp}{2}}\det(\mathbb{J})^p\det(\mathbb{M})^{\frac{1-p}{2}}\mathbb{M}^{-1}\frac{\partial \mathbb{M}}{\partial t}\right).
\end{aligned}$$

Thus with $\frac{\partial G}{\partial \mathbb{M}} = \frac{\partial G_1}{\partial \mathbb{M}} + \frac{\partial G_2}{\partial \mathbb{M}}$ we obtain the derivative $\frac{\partial G}{\partial \mathbb{M}}$.

Therefore the first derivatives of G are as given in (2.29), i.e.,

$$
\begin{cases}
\dfrac{\partial G}{\partial \mathbb{J}} &= dp\theta\sqrt{\det(\mathbb{M})}\left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}-1}\mathbb{M}^{-1}\mathbb{J}^T, \\
\dfrac{\partial G}{\partial \det(\mathbb{J})} &= p(1-2\theta)d^{\frac{dp}{2}}\det(\mathbb{M})^{\frac{1-p}{2}}\det(\mathbb{J})^{p-1}, \\
\dfrac{\partial G}{\partial \mathbb{M}} &= -\frac{\theta dp}{2}\sqrt{\det(\mathbb{M})}\left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}-1}\mathbb{M}^{-1}\mathbb{J}^T\mathbb{J}\mathbb{M}^{-1} \\
&\quad + \frac{\theta}{2}\sqrt{\det(\mathbb{M})}\left(\mathrm{tr}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}}\mathbb{M}^{-1} \\
&\quad + \frac{(1-2\theta)(1-p)d^{\frac{dp}{2}}}{2}\sqrt{\det(\mathbb{M})}\left(\dfrac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M}}}\right)^p\mathbb{M}^{-1}.
\end{cases}
\tag{A.5}
$$

A.3 Formulation and coercivity of the new meshing functional for surface meshes

To formulate the surface energy functional similar to (2.17) for bulk meshes, consider the alignment condition (3.10) and note that this implies all of the eigenvalues of $(F'_K)^T\,\mathbb{M}_K F'_K$ are equal, i.e.,

$$(F'_K)^T\,\mathbb{M}_K F'_K = \theta_K I,$$

where θ_K denotes the eigenvalue. This gives

$$\det\left((F'_K)^T\,\mathbb{M}_K F'_K\right) = \theta_K^{\frac{d-1}{2}}. \tag{A.6}$$

Comparing (A.6) to the equidistribution condition (3.5) we get

$$\theta = \left(\frac{\sigma_h}{N}\right)^{\frac{2}{d-1}},$$

and therefore

$$(F'_K)^T\,\mathbb{M}_K F'_K = \left(\frac{\sigma_h}{N}\right)^{\frac{2}{d-1}} I. \tag{A.7}$$

This gives rise to the energy functional

$$I_h = \sum_{K\in\mathcal{T}_h} |\hat{K}| \det\left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{\frac{1}{2}} \left\| \left(\left(F_K'\right)^T \mathbb{M}_K F_K'\right)^{-1} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right\|_F^{2p} \tag{A.8}$$

as given in (3.16).

As with (3.15), we can write (A.8) as (3.20), i.e.,

$$I_h = \sum_{K\in\mathcal{T}_h} G(\mathbb{J}, r) = \sum_{K\in\mathcal{T}_h} |\hat{K}| r^{-\frac{1}{2}} \left\| \mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right\|_F^{2p} \tag{A.9}$$

To find the expressions for $\frac{\partial G}{\partial \mathbb{J}}$ and $\frac{\partial G}{\partial r}$ which are needed to compute (3.44), we use the scalar-by-matrix properties given in (3.25). First, let us consider $\frac{\partial G}{\partial \mathbb{J}}$ and let

$$A = \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I\right).$$

Then

$$\begin{aligned}
\frac{\partial G}{\partial t} &= \operatorname{tr}\left(\frac{\partial G}{\partial A^T A}\frac{\partial A^T A}{\partial t}\right) \\
&= \operatorname{tr}\left(|\hat{K}| r^{-\frac{1}{2}} \frac{\partial \|A\|_F^{2p}}{\partial A^T A}\frac{\partial A^T A}{\partial t}\right) \\
&= \operatorname{tr}\left(p|\hat{K}| r^{-\frac{1}{2}} \|A\|_F^{2(p-1)} \frac{\partial \operatorname{tr}\left(A^T A\right)}{\partial A^T A}\frac{\partial A^T A}{\partial t}\right) \\
&= \operatorname{tr}\left(p|\hat{K}| r^{-\frac{1}{2}} \|A\|_F^{2(p-1)} \frac{\partial A^T A}{\partial t}\right).
\end{aligned}$$

Now using the definition of A we obtain the following

$$\begin{aligned}
\frac{\partial G}{\partial t} &= \operatorname{tr}\left(p|\hat{K}|r^{-\frac{1}{2}} \left\| \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right) \right\|_F^{2(p-1)} \frac{\partial \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right)^T \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right)}{\partial t} \right) \\
&= \operatorname{tr}\left(2p|\hat{K}|r^{-\frac{1}{2}} \left\| \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right) \right\|_F^{2(p-1)} \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right) \frac{\partial \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right)}{\partial t} \right) \\
&= \operatorname{tr}\left(2p|\hat{K}|r^{-\frac{1}{2}} \left\| \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right) \right\|_F^{2(p-1)} \left(\mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right) \frac{\partial \mathbb{J}}{\partial t} \right).
\end{aligned}$$

Similarly, let us consider $\frac{\partial G}{\partial r}$. That is

$$\begin{aligned}
\frac{\partial G}{\partial t} &= \operatorname{tr}\left(\frac{\partial G}{\partial r} \frac{\partial r}{\partial t} \right) \\
&= \operatorname{tr}\left(|\hat{K}| \left\| \mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right\|_F^{2p} \frac{\partial r^{-\frac{1}{2}}}{\partial r} \frac{\partial r}{\partial t} \right) \\
&= \operatorname{tr}\left(-\frac{1}{2}|\hat{K}|r^{-\frac{3}{2}} \left\| \mathbb{J} - \left(\frac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right\|_F^{2p} \frac{\partial r}{\partial t} \right).
\end{aligned}$$

Therefore, for the functional (A.8), the derivatives of G are given by

$$\begin{cases}
\dfrac{\partial G}{\partial \mathbb{J}} &= 2p|\hat{K}|r^{-\frac{1}{2}} \left\| \left((F_K')^T \mathbb{M}_K F_K' \right)^{-1} - \left(\dfrac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right\|_F^{2(p-1)} \\
&\qquad \cdot \left(\left((F_K')^T \mathbb{M}_K F_K' \right)^{-1} - \left(\dfrac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right), \\
\dfrac{\partial G}{\partial r} &= -\dfrac{1}{2}|\hat{K}|r^{-\frac{3}{2}} \left\| \left((F_K')^T \mathbb{M}_K F_K' \right)^{-1} - \left(\dfrac{\sigma_h}{N}\right)^{-\frac{2}{d-1}} I \right\|_F^{2p}.
\end{cases} \tag{A.10}$$

A.4 First derivatives of surface meshing functional

Consider (3.20) for the surface energy functional (3.15), i.e.,

$$I_h = \sum_{K\in\mathcal{T}_h} G(\mathbb{J}, r) = \sum_{K\in\mathcal{T}_h} \theta|\hat{K}|r^{-\frac{1}{2}}\mathrm{tr}\,(\mathbb{J})^{\frac{p(d-1)}{2}} + (1-2\theta)(d-1)^{\frac{p(d-1)}{2}}|\hat{K}|r^{\frac{p-1}{2}}.$$

Using the scalar-by-matrix properties (3.25) we can find $\frac{\partial G}{\partial \mathbb{J}}$ and $\frac{\partial G}{\partial r}$. To find $\frac{\partial G}{\partial \mathbb{J}}$ consider

$$\begin{aligned}
\frac{dG}{dt} &= \mathrm{tr}\left(\frac{\partial G}{\partial \mathbb{J}}\frac{\partial \mathbb{J}}{\partial t}\right) \\
&= \mathrm{tr}\left(\theta\ |\hat{K}|r_K^{-\frac{1}{2}}\ \frac{\partial \mathrm{tr}\,(\mathbb{J})^{\frac{p(d-1)}{2}}}{\partial \mathbb{J}}\ \frac{\partial \mathbb{J}}{\partial t}\right) \\
&= \mathrm{tr}\left(\frac{\theta p(d-1)}{2}\ |\hat{K}|r_K^{-\frac{1}{2}}\ \mathrm{tr}\,(\mathbb{J})^{\frac{p(d-1)-2}{2}}\ \frac{\partial \mathrm{tr}\,(\mathbb{J})}{\partial \mathbb{J}}\ \frac{\partial \mathbb{J}}{\partial t}\right) \\
&= \mathrm{tr}\left(\frac{\theta p(d-1)}{2}\ |\hat{K}|r_K^{-\frac{1}{2}}\ \mathrm{tr}\,(\mathbb{J})^{\frac{p(d-1)-2}{2}}\ I\ \frac{\partial \mathbb{J}}{\partial t}\right).
\end{aligned}$$

Similarly, to find $\frac{\partial G}{\partial r}$ consider

$$\begin{aligned}
\frac{dG}{dt} &= \mathrm{tr}\left(\frac{\partial G}{\partial r}\frac{\partial r}{\partial t}\right) \\
&= \mathrm{tr}\left(\left[\theta\ |\hat{K}|\ \mathrm{tr}\,(\mathbb{J})^{\frac{p(d-1)}{2}}\ \frac{\partial r^{-\frac{1}{2}}}{\partial r} + (1-2\theta)(d-1)^{\frac{p(d-1)}{2}}|\hat{K}|\frac{\partial r^{\frac{p-1}{2}}}{\partial r}\right]\ \frac{\partial r}{\partial t}\right) \\
&= \mathrm{tr}\left(\left[-\frac{\theta}{2}\ |\hat{K}|\ r^{-\frac{3}{2}}\mathrm{tr}\,(\mathbb{J})^{\frac{p(d-1)}{2}} + \frac{p-1}{2}(1-2\theta)(d-1)^{\frac{p(d-1)}{2}}|\hat{K}|r^{\frac{p-3}{2}}\right]\ \frac{\partial r}{\partial t}\right).
\end{aligned}$$

Therefore the first derivatives of G are as given in (3.26), i.e.,

$$\begin{cases}
\dfrac{\partial G}{\partial \mathbb{J}} &= \dfrac{\theta p(d-1)}{2}|\hat{K}|r^{-\frac{1}{2}}\mathrm{tr}(\mathbb{J})^{\frac{p(d-1)-2}{2}}I, \\[2ex]
\dfrac{\partial G}{\partial r} &= -\dfrac{\theta}{2}|\hat{K}|r^{-\frac{3}{2}}\mathrm{tr}(\mathbb{J})^{\frac{p(d-1)}{2}} + \dfrac{p-1}{2}(1-2\theta)(d-1)^{\frac{p(d-1)}{2}}|\hat{K}|r^{\frac{p-3}{2}}.
\end{cases} \tag{A.11}$$

A.5 Formulation of surface MMPDE

The surface MMPDE (3.42) is intuitively derived from the MMPDE approach for bulk meshes with a slight alternation which ensures that the nodes remain on the surface during movement by projecting the velocities onto the tangential space. However, it should be noted that (3.42) can also be formulated directly using Lagrange multipliers. That is, for a given i for $i = 1, \ldots, N_v$, we want to solve the minimization problem given by

$$\min_{\boldsymbol{x}_i} I_h(\boldsymbol{x}_i) \quad \text{s.t.} \quad \Phi(\boldsymbol{x}_i) = 0. \tag{A.12}$$

By the method of Lagrange multipliers, we define

$$\mathcal{L}(\boldsymbol{x}_i, \lambda) = I_h(\boldsymbol{x}_i) + \lambda \Phi(\boldsymbol{x}_i) \tag{A.13}$$

and hence we obtain the system of equations given by

$$\begin{cases} \left(\dfrac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T + \lambda \nabla \Phi(\boldsymbol{x}_i) = 0 \\ \Phi(\boldsymbol{x}_i) = 0, \quad i = 1, \ldots, N_v. \end{cases} \tag{A.14}$$

Employing a modified gradient decent method to (A.14) we obtain

$$\frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left(\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T + \lambda \nabla \Phi\right) \tag{A.15}$$

such that $\Phi(\boldsymbol{x}_i) = 0$. Since $\boldsymbol{x}_i = \boldsymbol{x}_i(t)$, instead of requiring $\Phi(\boldsymbol{x}_i) = 0$, we impose a weaker condition to (A.15) given by

$$\nabla \Phi \cdot \frac{d\boldsymbol{x}_i}{dt} = 0. \tag{A.16}$$

Applying (A.16) to the MMPDE (A.15) we obtain

$$0 = \nabla\Phi \cdot \frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left(\nabla\Phi \cdot \left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T + \lambda \left\|\nabla\Phi\right\|^2\right) \tag{A.17}$$

which gives

$$\lambda = -\frac{\nabla\Phi}{\left\|\nabla\Phi\right\|^2} \cdot \left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T. \tag{A.18}$$

Inserting λ into (A.15) we obtain (3.42), i.e.,

$$\frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left[\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T - \left(\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T \cdot \boldsymbol{n}_i\right)\boldsymbol{n}_i\right] \tag{A.19}$$

where $\boldsymbol{n}_i = \frac{\nabla\Phi}{|\nabla\Phi|}$ is the outward normal vector to the surface Φ at the point $\boldsymbol{x}_i$.

As previously mentioned, we imposed a weak condition (A.16) to obtain (3.42) however, we can impose a stronger condition to obtain a stabilized MMPDE. That is, we impose

$$\Phi(\boldsymbol{x}_i(t + \delta\tau)) = 0 \tag{A.20}$$

to (A.15) where $\delta \geq 0$ is a constant. In this, we are not requiring that the nodes be directly projected onto the surface at time t, i.e., $\Phi(\boldsymbol{x}_i) = 0$, but instead be projected onto the surface at some time $t + \delta\tau$. This is called a delayed projection which is a stronger imposition than (A.16). By Taylor's Expansion on $\boldsymbol{x}_i$ in (A.20) we have

$$0 = \Phi(\boldsymbol{x}_i(t + \delta\tau)) = \Phi\left(\boldsymbol{x}_i(t) + \delta\tau\frac{d\boldsymbol{x}_i}{dt} + \mathcal{O}((\delta\tau)^2)\right). \tag{A.21}$$

Moreover, by Taylor's Expansion on Φ we get

$$0 = \Phi(\boldsymbol{x}_i(t)) + \delta\tau\nabla\Phi \cdot \frac{d\boldsymbol{x}_i}{dt}. \tag{A.22}$$

Combining (A.22) with (A.15) we obtain

$$\nabla\Phi \cdot \frac{d\boldsymbol{x}_i}{dt} = -\frac{\Phi(\boldsymbol{x}_i)}{\delta\tau} = -\frac{P_i}{\tau}\left(\nabla\Phi \cdot \left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T + \lambda \left\|\nabla\Phi\right\|^2\right) \tag{A.23}$$

which then gives

$$\frac{1}{P_i\delta}\Phi(\boldsymbol{x}_i) - \nabla\Phi \cdot \left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T = \lambda \left\|\nabla\Phi\right\|^2. \tag{A.24}$$

Therefore, by (A.15) we have

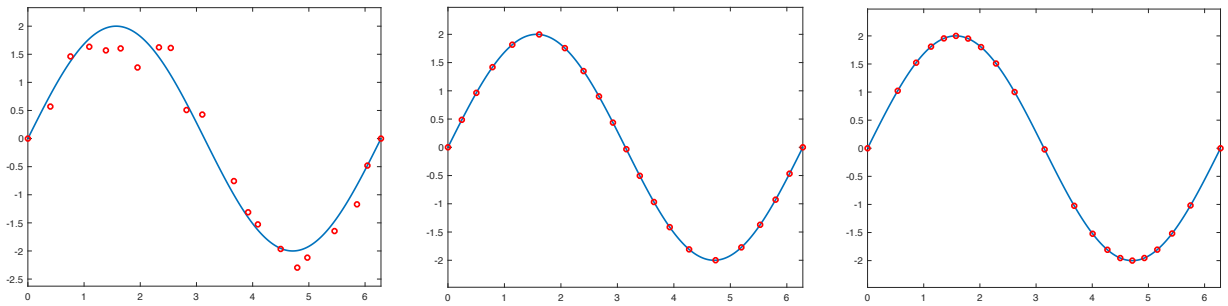
$$\frac{d\boldsymbol{x}_i}{dt} = -\frac{P_i}{\tau}\left[\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T - \left(\left(\frac{\partial I_h}{\partial \boldsymbol{x}_i}\right)^T \cdot \boldsymbol{n}_i\right)\boldsymbol{n}_i\right] - \frac{\Phi}{\delta\tau\left\|\nabla\Phi\right\|}\,\boldsymbol{n}_i \tag{A.25}$$

where $\boldsymbol{n}_i = \frac{\nabla\Phi}{|\nabla\Phi|}$ as before. The role of the last term can be seen through the different values of δ. That is, from (A.25) we can see that if $\delta = 0$ then $\Phi(\boldsymbol{x}_i(t)) = 0$ and hence we project the nodes directly onto the surface. If $\delta = \infty$ then (A.25) gives (3.42), i.e., we project the nodes onto the tangential space. For all values $0 < \delta < \infty$, (A.25) uses a delayed projection of the nodes onto the surface. It should be noted that we use $\delta\tau$ as the delay parameter so that dimension of τ in all of the terms in (A.25) agree. Furthermore, an explicit parameterization of the curve is required for the delayed projection method and cannot be approximated for obvious reasons.

To see the effectiveness of the delayed projection method, consider Figure A.1, i.e.,

$$\Phi(x, y) = 2\sin(x) - y.$$

In this, the initial mesh shows the nodes off of the surface however, when (A.25) is employed, the nodes are projected directly onto the surface and the method continues as the examples in Section 3.5.2. Here we use $N = 20$, $dt = 0.01$, $\tau = 0.01$, and $\delta = 0.001$. It should be noted, however, that δ depends on N and the initial mesh. That is, if the initial mesh is fine (i.e., N is large) and the nodes are close in distance, δ must be chosen to be sufficiently small otherwise the delayed projection will result in a singular mesh.



(a) Initial Mesh. (b) Final Mesh, $\mathbb{M}_K = I$. (c) Final Mesh, $\mathbb{M}_K = (k_K + \epsilon)I$.

Figure A.1: Meshes for $\Phi(x, y) = 2\sin(x) = y$ with $N = 20$ using the delayed projection MMPDE.