Project Number: LC1-AA5K

Quasiconformal Gradient Flows

An Major Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Weizhe Shen

Date: April 25, 2017

Approved:

Professor Luca Capogna, Advisor

1. Quasiconformal Maps
2. Gradient Flows

Abstract

The objective of this project was to quantify the difference between two planar domains with finitely prescribed matching points. To this end, we use gradient flows to obtain a homotopy of quasiconformal mappings between the two shapes, find a canonical minimizer for maximal infinitesimal stretching and rotation, and take the distortion of the minimizer as a measure of the difference between the two planar shapes. In our approach, numerical implementation is applied to get better insight of the long-time behavior of the gradient flow.

Contents

List of Figures

1 Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Luca Capogna, for his continuous guidance, instruction, and generous sharing of his wealth knowledge, and for all the enlightening and invaluable discussions throughout the completion of this Major Qualifying Project. I am especially grateful for Professor Capogna's creating this project when there was some issue with my original project, and for his great patience when I was lacking a sufficient foundation of knowledge with respect to this project in the beginning.

I would also like to thank Professor Xiaodan Zhou for her dedications and great devotion of time helping me with the process of reading graduate level papers and books. Thanks to Professor Zhou's correction of my work, there has been a substantial improvement of accuracy of my writing.

I would also like to extend my thanks to Professor Marcus Sarkis-Martins, for his suggestion and support on numerical implementations.

Finally, a special thank to Professor Maria Hempel, for her help in contacting Professor Capogna when she was not able to continue our original project on her medical leave. And best wishes to her health condition.

2 Introduction

In the book *On Growth and Form* by the Scottish mathematical biologist D'Arcy Wentworth Thompson [Thompson, 1942], a theory of transformation was introduced to characterize the change of shape and size during the growing process of an organism. In this theory, a grid method was applied to capture the descriptive and predictive properties underlying biological shapes via grid coordinates deformations.

The objective of this project was to generalize Thompson's problem mathematically and to come up with a new method to quantify the difference between two planar shapes (domains) with finitely prescribed matching points. To this end, we use quasiconformal maps between two planar shapes, preserving the landmark points. More specifically, the image of an infinitesimally small circle under such a map yields information on its local shear and distortion, and we consider various functional of these quantities to measure the difference in shapes between the two regions through the map.

Since there are infinitely many quasiconformal maps relating two given shapes, we need to find a canonical choice for such map to eliminate any ambiguity. In our approach, we use gradient flows to obtain a homotopy of quasiconformal mappings and simultaneously computing their maximal infinitesimal stretching and rotation at each time. As the flow decreases the mean distortion of the map, plausibly it may be used to find a canonical minimizer for such a functional. Once this is accomplished, we finally use the quasiconformal map with the smallest measure of shear as the optimal map, and we take the corresponding distortion as a measure of the difference between the two planar shapes. This approach is based on work of Capogna and Raich [Capogna and Raich, 2012], where the short time existence of the flow is proved. Since there are no results on the long time behavior we resort to numerical experiment to gain better insight on this problem.

There is a broad scientific literature on this problem. In particular, we were inspired by a recent paper of Gareth Wyn Jones and L. Mahadevan. In Jones and Mahadevan's work [Jones and Mahadevan, 2013], they aim to find a quasiconformal map that varies as little as possible in space and also minimizes an integrated squared-gradient of both shear and distortion. In the absence of any rigorous existence result, they use numerical experiments to analyze the problem.

3 Preliminaries on Quasiconformal Mappings in Plane

In this section we introduce the elementary definitions and important facts of conformal mappings and quasiconformal mappings. As most of our discussions are in complex forms, some basic review of complex variables and functions are also covered. The standard reference for this section includes [Ahlfors, 1953], [Ahlfors and Earle, 1966], [Capogna, 2014], [Jeffrey, 2005], [Jones and Mahadevan, 2013], [Recktenwald, 2004], and [Vuorinen, 1988].

3.1 Conformal Mappings

We shall start with the definition of *conformal mappings* in plane. The terminologies and the complex notations introduced in this subsection will be frequently used through this essay.

3.1.1 What is a Conformal Mapping in Plane

Definition 3.1. Let $u : \Omega \to \Omega'$ be a homeomorphism, where $\Omega, \Omega' \subset \mathbb{R}^2$ are two domains in the plane. Suppose $u(x,y) = (u^1(x,y), u^2(x,y))$. We say u is ***conformal*** if it satisfies the following conditions:

(1) $u \in C^1$, i.e., all first order partial derivatives of $u = (u^1, u^2)$ are continuous in Ω.

(2) The Jacobian determinant of u is non-zero for all points in Ω, i.e., $\forall (x,y) \in \mathbb{D}$,

$$J_u(x,y) = \det \begin{bmatrix} \frac{\partial u^1}{\partial x} & \frac{\partial u^1}{\partial y} \\ \frac{\partial u^2}{\partial x} & \frac{\partial u^2}{\partial y} \end{bmatrix} (x,y) \neq 0.$$

(3) $\forall (x,y) \in \mathbb{D}, \forall \vec{h} \in \mathbb{R}^2$,

$$\|\mathrm{d}u(x,y)\vec{h}\| = |\mathrm{d}u(x,y)|_o \, \|\vec{h}\|, \tag{3.1}$$

where du is the Jacobian matrix of u.

Remark. The matrix norm in equation (3.1) is referring to the *operator norm.* For a given 2×2 matrix A, the corresponding operator norm $|A|_o$ is defined via

$$|A|_o := \sup_{\substack{\|\vec{v}\|=1 \\ \vec{v} \in \mathbb{R}^2}} \|A\vec{v}\|.$$

The somewhat more familiar matrix norm to most us is the *Hilbert-Schmidt norm*, which is defined via

$$|A| := \sqrt{\sum_{i,j=1}^{2} A_{ij}^2}.$$

The definitions of the two norms can directly extend to n-dimensional case.

Let's see some specific examples of conformal mappings. A series of basic examples in general $\mathbb{R}^n$ case can be found in [Vuorinen, 1988].

Example 3.1. Define $u : \mathbb{R}^2 \to \mathbb{R}^2$ via $u(x, y) = (3x, 3y)$. This is a stretching mapping by a factor $k > 0$ (here, $k = 3$). For instance, the image of point $(-1, 2)$ is point $(-3, 6)$.

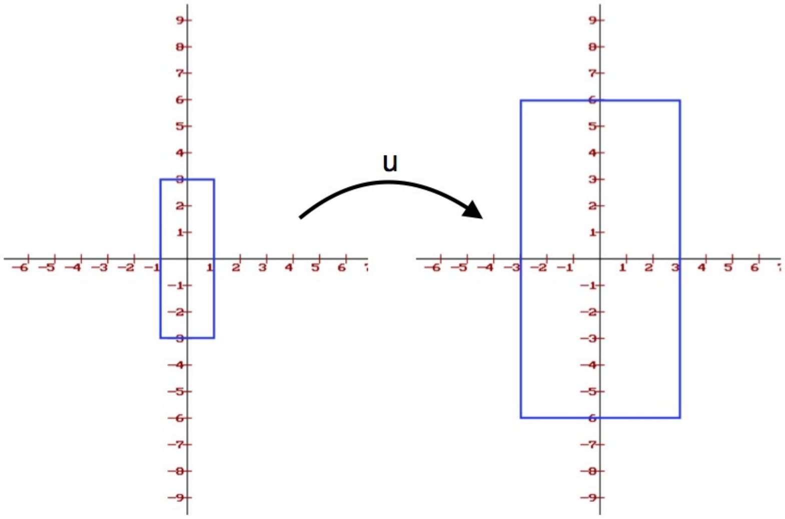


Figure 1: The image of a rectangle under the example stretching

Let's verify that u is conformal.

1. Here, $u^1 = 3x$ and $u^2 = 3y$. The first order partial derivatives are $\partial_x u^1 = 3, \partial_y u^1 = 0, \partial_x u^2 = 0$ and $\partial_y u^2 = 3$, all of which are continuous in $\mathbb{R}^2$.

2. From the first order partial derivatives that we have got above, it follows that the Jacobian determinant

$$J_u = \det \begin{bmatrix} \partial_x u^1 & \partial_y u^1 \\ \partial_x u^2 & \partial_y u^2 \end{bmatrix} = \det \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} = 9 \neq 0.$$

3. Let $\vec{h} = (h_1, h_2) \in \mathbb{R}^2$ be given. Then

$$\|\mathrm{d}u(x,y)\vec{h}\| = \left\| \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \right\| = 3 \left\| \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \right\| = 3\sqrt{h_1^2 + h_2^2} = 3\|\vec{h}\|,$$

and

$$|\mathrm{d}u(x,y)|_o \, \|\vec{h}\| = \sup_{\substack{\vec{v}=(v_1,v_2)\in\mathbb{R}^2 \\ \|\vec{v}\|=1}} \left\| \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right\| \|\vec{h}\| = \sup_{\substack{\vec{v}=(v_1,v_2)\in\mathbb{R}^2 \\ \|\vec{v}\|=1}} 3\,\|\vec{v}\|\,\|\vec{h}\| = 3\|\vec{h}\|.$$

Therefore,

$$\|\mathrm{d}u(x,y)\vec{h}\| = |\mathrm{d}u(x,y)|_o \, \|\vec{h}\|.$$

Hence, the stretching u with respect to factor $k = 3$ is a conformal mapping. Indeed, every stretching on the plane is conformal.

Example 3.2. Define $u : \mathbb{R}^2 \to \mathbb{R}^2$ via

$$u(x,y) = \begin{bmatrix} \cos(\frac{\pi}{3}) & -\sin(\frac{\pi}{3}) \\ \sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

This is a rotation mapping which rotates points counter-clockwise through an angle $\frac{\pi}{3}$ (radian) about the origin. For instance, the image of point $(2, 0)$ is point $(1, \sqrt{3})$.

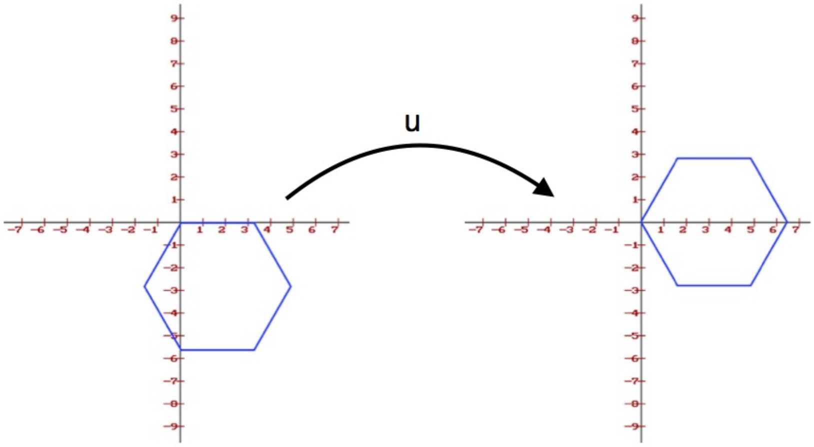


Figure 2: The image of a hexagon under the example rotation

Let's verify that u is conformal.

1. Here, $u^1 = \frac{1}{2}(x - \sqrt{3}y)$ and $u^2 = \frac{1}{2}(\sqrt{3}x + y)$. The first order partial derivatives are $\partial_x u^1 = \frac{1}{2}, \partial_y u^1 = -\frac{\sqrt{3}}{2}, \partial_x u^2 = \frac{\sqrt{3}}{2}$ and $\partial_y u^2 = \frac{1}{2}$, all of which are continuous in $\mathbb{R}^2$.

2. From the first order partial derivatives that we have got above, it follows that the Jacobian determinant
$$J_u = \det \begin{bmatrix} \partial_x u^1 & \partial_y u^1 \\ \partial_x u^2 & \partial_y u^2 \end{bmatrix} = \det \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} = 1 \neq 0.$$

3. Let $\vec{h} = (h_1, h_2) \in \mathbb{R}^2$ be given. Then
$$\|\mathrm{d}u(x,y)\vec{h}\| = \left\| \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \right\| = \frac{1}{2} \left\| \begin{bmatrix} h_1 - \sqrt{3}h_2 \\ \sqrt{3}h_1 + h_2 \end{bmatrix} \right\| = \frac{1}{2}\sqrt{4h_1^2 + 4h_2^2} = \|\vec{h}\|,$$
and
$$|\mathrm{d}u(x,y)|_o \|\vec{h}\| = \sup_{\substack{\vec{v}=(v_1,v_2)\in\mathbb{R}^2 \\ \|\vec{v}\|=1}} \left\| \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right\| \|\vec{h}\| = \sup_{\substack{\vec{v}=(v_1,v_2)\in\mathbb{R}^2 \\ \|\vec{v}\|=1}} \|\vec{v}\| \|\vec{h}\| = \|\vec{h}\|.$$
Therefore,
$$\|\mathrm{d}u(x,y)\vec{h}\| = |\mathrm{d}u(x,y)|_o \|\vec{h}\|.$$

Hence, the counter-clockwise rotation u with respect to an angle $\frac{\pi}{3}$ about the origin is a conformal mapping. Indeed, every rotation on the plane is conformal.

Before introducing more about conformal mappings, let us first review some algebra facts, which will help us in subsequent proofs.

Lemma 3.1. Let A be an $n \times n$ matrix, and let $\vec{v}, \vec{w} \in \mathbb{R}^n$ be two vectors. Then we have
$$\langle A\vec{v}, \vec{w} \rangle = \langle \vec{v}, A^T\vec{w} \rangle,$$
where $\langle \cdot, \cdot \rangle$ is the inner product.

Proof.
$$\begin{aligned} \langle A\vec{v}, \vec{w} \rangle &= \sum_{i=1}^{n} (A\vec{v})_i \vec{w}_i = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{ij}\vec{v}_j \right) \vec{w}_i = \sum_{i,j=1}^{n} A_{ij}\vec{v}_j\vec{w}_i = \sum_{j=1}^{n} \vec{v}_j \left(\sum_{i=1}^{n} A_{ij}\vec{w}_i \right) \\ &= \sum_{j=1}^{n} \vec{v}_j \left(\sum_{i=1}^{n} A^T_{ji}\vec{w}_i \right) = \sum_{j=1}^{n} \vec{v}_j (A\vec{w})_j \\ &= \langle \vec{v}, A^T\vec{w} \rangle. \end{aligned}$$

□

Lemma 3.2. Let A be an $n \times n$ symmetric matrix (i.e., $A = A^T$). Then

$$\operatorname{trace}\left(AA^T\right) = |A|^2 = \operatorname{trace}\left(A^T A\right).$$

Proof. Suppose $A = (a_{ij})_{n\times n}$. Then

$$\operatorname{trace}\left(AA^T\right) = \operatorname{trace}\left[\left(\sum_{k=1}^{n} a_{ik}a_{jk}\right)_{n\times n}\right] = \sum_{i,k=1}^{n} a_{ik}^2 = |A|^2$$

Since $|A^T|^2 = |A|^2$, then

$$\operatorname{trace}\left(AA^T\right) = \operatorname{trace}\left(A^T A\right).$$

□

From the third condition of the definition of conformal homeomorphisms, $\|\mathrm{d}u\vec{h}\| = |\mathrm{d}u|_o \|\vec{h}\|$, we do the following computations:

$$\begin{aligned}
\|\mathrm{d}u\vec{h}\|^2 &= |\mathrm{d}u|_o^2 \|\vec{h}\|^2 \\
\langle \mathrm{d}u\vec{h}, \mathrm{d}u\vec{h}\rangle &= |\mathrm{d}u|_o^2 \langle \vec{h}, \vec{h}\rangle \\
\langle (\mathrm{d}u)^T \mathrm{d}u\vec{h}, \vec{h}\rangle &= \langle |\mathrm{d}u|_o^2 \,\mathrm{Id}\vec{h}, \vec{h}\rangle \\
\langle \left[(\mathrm{d}u)^T \mathrm{d}u - |\mathrm{d}u|_o^2 \,\mathrm{Id}\right] \vec{h}, \vec{h}\rangle &= 0.
\end{aligned}$$

Since $\vec{h} \in \mathbb{R}$ is arbitrary, we have that

$$(\mathrm{d}u)^T \mathrm{d}u = |\mathrm{d}u|_o^2 \,\mathrm{Id}. \tag{3.2}$$

Then we take determinants of both sides of equation (3.2) and get

$$\begin{aligned}
\det\left((\mathrm{d}u)^T \mathrm{d}u\right) &= \det\left(|\mathrm{d}u|_o^2 \,\mathrm{Id}\right) \\
(\det \mathrm{d}u)^2 &= |\mathrm{d}u|_o^4 .
\end{aligned}$$

Thus, $|\mathrm{d}u|_o^2 = |\det \mathrm{d}u|$. Substituting for $|\mathrm{d}u|_o^2$ in equation (3.2) gives us

$$(\mathrm{d}u)^T \mathrm{d}u = |\det \mathrm{d}u| \,\mathrm{Id}. \tag{3.3}$$

We conclude that, when u is conformal, $\dfrac{\mathrm{d}u^T \mathrm{d}u}{|\det \mathrm{d}u|}$ equals the 2×2 identity matrix.

3.1.2 Complex Variables and Functions

As we are talking about mapping in plane, alternatively, it is natural to use the complex notations. Let's first recall some facts about complex variables and functions.

A complex number in the Cartesian representation is generally written as $z = x + iy$, and the correspongding complex conjugate is defined as $\overline{z} = x - iy$. Conversely, the real part and the imaginary part of a complex number can be represented as $x = \dfrac{z + \overline{z}}{2}$ and $y = \dfrac{z - \overline{z}}{2i}$, respectively. In conclusion, if we are given an ordered pair (x, y) as the real and imaginary parts of a complex number, we can find a unique pair $(z, \overline{z})$, and vice versa.

Consider two complex planes, z-plane and w-plane. Let $w = f(z) = u(x, y) + iv(x, y)$ be a function that assigns each point $z = x + iy$ in the domain $\mathbb{D}$ on the z-plane to a unique point $w = u + iv$ on the w-plane. According to the one-to-one correspondence between (x, y) and $(z, \overline{z})$, we can describe the complex function as $w = w(z, \overline{z})$ (for convenience we still denote the function by w) and get the following partial derivatives:

$$
\begin{aligned}
\frac{\partial w}{\partial z} &= \frac{\partial u}{\partial x}\frac{\partial x}{\partial z} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial z} + \frac{\partial v}{\partial x}\frac{\partial x}{\partial z} + \frac{\partial v}{\partial y}\frac{\partial y}{\partial z} \\
&= \frac{1}{2}\frac{\partial u}{\partial x} + \frac{-i}{2i}\frac{\partial u}{\partial y} + \frac{i}{2}\frac{\partial v}{\partial x} + \frac{i}{2i}\frac{\partial v}{\partial y} \\
&= \frac{1}{2}(u_x - iu_y + iv_x + v_y), \\
\frac{\partial w}{\partial \overline{z}} &= \frac{1}{2}(u_x + iu_y + iv_x - v_y), \\
\frac{\partial \overline{w}}{\partial z} &= \frac{1}{2}(u_x - iu_y - iv_x - v_y), \\
\frac{\partial \overline{w}}{\partial \overline{z}} &= \frac{1}{2}(u_x + iu_y - iv_x + v_y).
\end{aligned}
$$

Definition 3.2. The system of the following two equations which involves the first order partial derivatives of $u(x, y)$ and $v(x, y)$,

$$
\begin{aligned}
\frac{\partial u}{\partial x} &= \frac{\partial v}{\partial y}, \\
\frac{\partial u}{\partial y} &= -\frac{\partial v}{\partial x},
\end{aligned}
$$

is called the ***Cauchy-Riemann equations***.

Remark. $\dfrac{\partial w}{\partial \overline{z}} = 0$ is an equivalent representation of the *Cauchy-Riemann equations*.

Indeed, if $\dfrac{\partial w}{\partial \overline{z}} = 0$, then both the real and imaginary parts of it are zeros, i.e., $u_x - v_y = 0$ and $u_y + v_x = 0$. Conversely, if the *Cauchy-Riemann equations* are satisfied, then $\dfrac{\partial w}{\partial \overline{z}} = 0 + i0 = 0$.

Definition 3.3. If $f(z)$ is defined at $z = z_0$ and in some open set containing z_0 (which is also called a *neighborhood* of z_0), we say f is ***differentiable*** at z_0 if the limit

$$f'(z_0) := \lim_{z \to z_0} \frac{f(z) - f(z_0)}{z - z_0}$$

exists. We call $f'(z_0)$ the ***derivative*** of f at z_0.

Definition 3.4. If, for some domain $\Omega \subset \mathbb{C}$, f is differentiable at every point of Ω, then f is said to be ***analytic*** or ***holomorphic*** in Ω.

After the review of complex variables and functions, we now introduce the following two theorems, which are not only basic but also extremely important in complex analysis. The detailed proofs and other related theorems and corollaries can be found in [Jeffrey, 2005].

Theorem 3.1. If the two real-valued functions $u(x, y)$ and $v(x, y)$ in $f(x+iy) = u(x, y) + iv(x, y)$ are both of C^1 class in the domain Ω (which means that their first order partial derivatives are continuous in Ω), and if the *Cauchy-Riemann equations* are satisfied, then f is holomorphic in Ω.

Theorem 3.2. A holomorphic function $w = f(z)$ is conformal at any point where $f'(z) \neq 0$. Conversely, every conformal mapping $w : \mathbb{C} \to \mathbb{C}$ that has continuous partial derivatives is holomorphic.

3.1.3 Local Properties under a Planar Mapping

In this small subsection we shall talk about properties of the image of an infinitesimally small circle under a planar mapping. The reason why we are concerned about it is that this image reflects local shear and distortion properties of the planar mapping. Although this section is mainly talking about conformal mappings, here we shall broaden our scope and start with a general planar mapping. The property of conformal mappings will be included in the discussion of general mappings, and we shall see that *conformality* is a rigid property which does not approve local shears.

Let's start with dw, the infinitesimal change of w. If there is only a change in dx, then $dw = w(x + dx, y) - w(x, y) = (w(x, y) + w_x(x, y)dx) - w(x, y)$, and thus,

$$dw = w_x dx.$$

In general, for $dw = w(x+dx, y+dy) - w(x,y)$, we add and subtract $w(x, y+dy)$ at the same time and do the similar computations to get

$$dw = w_x dx + w_y dy.$$

Likewise, when the mapping is represented by $w = w(z, \overline{z})$, we have

$$dw = w_z dz + w_{\overline{z}} d\overline{z}.$$

We notice that dw is an affine mapping, and that

$$\Big||w_z| - |w_{\overline{z}}|\Big||dz| \leqslant |dw| \leqslant \Big||w_z| + |w_{\overline{z}}|\Big||dz|.$$

An infinitesimally small circle on the complex plane can be described in the exponential form $dz = dre^{i\theta}$. We want to know the image of the circle under the complex map $w : \mathbb{C} \to \mathbb{C}$.

In the polar representation, $d\overline{z} = dre^{-i\theta}$. If we set $w_z = |w_z|e^{i\alpha}$ and $w_{\overline{z}} = |w_{\overline{z}}|e^{i\beta}$, where $|w_z|, |w_{\overline{z}}|$ are the moduli, and α, β are the arguments of w_z and $w_{\overline{z}}$, respectively, then

$$\begin{aligned} dw &= w_z dz + w_{\overline{z}} d\overline{z} \\ &= |w_z|e^{i\alpha} dre^{i\theta} + |w_{\overline{z}}|e^{i\beta} dre^{-i\theta} \\ &= dr(|w_z|e^{i(\alpha+\theta)} + |w_{\overline{z}}|e^{i(\beta-\theta)}) \\ &= dre^{i(\frac{\alpha+\beta}{2})}(|w_z|e^{i[\theta - \frac{\beta-\alpha}{2}]} + |w_{\overline{z}}|e^{-i[\theta - \frac{\beta-\alpha}{2}]}). \end{aligned}$$

For $Ae^{i\phi} + Be^{-i\phi}$, where $A, B \in \mathbb{R}$ and ϕ is a variable, we write it in the polar form and get

$$\begin{aligned} Ae^{i\phi} + Be^{-i\phi} &= A(\cos\phi + i\sin\phi) + B(\cos\phi - i\sin\phi) \\ &= (A+B)\cos\phi + i(A-B)\sin\phi. \end{aligned}$$

Let $x := (A+B)\cos\phi$, and also let $y := (A-B)\sin\phi$. Then we notice that

$$\frac{x^2}{(A+B)^2} + \frac{y^2}{(A-B)^2} = 1,$$

which represents an ellipse (if $B \neq 0$) or a circle (if $B = 0$). Since $dre^{i(\frac{\alpha+\beta}{2})}(|w_z|e^{i[\theta - \frac{\beta-\alpha}{2}]} + |w_{\overline{z}}|e^{-i[\theta - \frac{\beta-\alpha}{2}]})$ is of the same format of expression as $Ae^{i\phi} + Be^{-i\phi}$, we infer that, if $|w_{\overline{z}}| \neq 0$, the image of the circle $dz = dre^{i\theta}$ under the complex map $w : \mathbb{C} \to \mathbb{C}$ is an ellipse such that

1. its long axis is at an angle $\dfrac{\alpha+\beta}{2}$ to the horizontal, and
2. $\dfrac{|w_z|+|w_{\overline{z}}|}{|w_z|-|w_{\overline{z}}|}$ is its ratio of long axis to short axis.

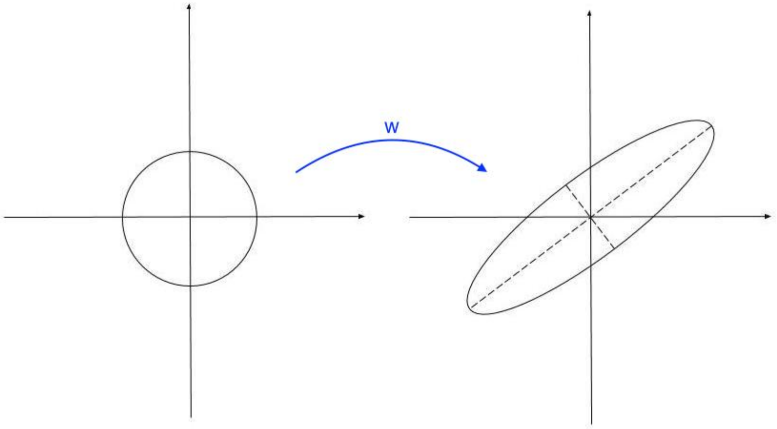


Figure 3: Mapping an infinitesimal circle onto an ellipse

Denote this ratio of axes by

$$D_w := \frac{|w_z|+|w_{\overline{z}}|}{|w_z|-|w_{\overline{z}}|}.$$

Remark. $D_w = 1$ if and only if $w_{\overline{z}} = 0$.

Indeed, if $D_w = 1$ (which means that the image is still a circle), then we have that $w_{\overline{z}} = 0$, the *Cauchy-Riemann equations.* Conversely, if $w_{\overline{z}} = 0$, then $D_w = 1$.

Moreover, we infer from the Cauchy-Riemann equations, theorems 3.1, and theorem 3.2 that, for conformal mappings, infinitesimal circles are to be mapped to circles, and that $D_w = 1$.

The maximal dilatation on $\Omega \subset \mathbb{C}$ is given by

$$K := \sup_{\Omega} D_w = \sup_{\Omega} \frac{|w_z|+|w_{\overline{z}}|}{|w_z|-|w_{\overline{z}}|},$$

and we will use it as the *dilatation* of a diffeomorphism when we solve the *Grötzsch Problem.*

3.1.4 Composite Functions

In this subsection we shall discuss about the composition of two complex functions and their derivatives. In addition, we shall prove that two

Let $\Omega \subset \mathbb{C}$ be open, and let $f : \Omega \to f(\Omega)$ and $g : f(\Omega) \to \mathbb{C}$ be two complex functions. Suppose that f^1 and f^2 are the coordinate functions of f, and that g^1 and g^2 are the coordinate functions of g.

From the previous discussion about complex functions , we know that

$$\begin{aligned} f_z &= \frac{1}{2}\left[(f^1_x + f^2_y) + i(f^2_x - f^1_y)\right], \\ f_{\overline{z}} &= \frac{1}{2}\left[(f^1_x - f^2_y) + i(f^1_y + f^2_x)\right], \\ \overline{f}_z &= \frac{1}{2}\left[(f^1_x - f^2_y) - i(f^1_y + f^2_x)\right], \\ \overline{f}_{\overline{z}} &= \frac{1}{2}\left[(f^1_x + f^2_y) - i(f^2_x - f^1_y)\right], \end{aligned}$$

from which we observe that

$$\begin{aligned} \overline{f}_z &= \overline{(f_{\overline{z}})}, \\ \overline{f}_{\overline{z}} &= \overline{(f_z)}. \end{aligned} \tag{3.4}$$

Using z and ζ to denote the variables on Ω and $f(\Omega)$, respectively, we obtain the following derivatives for the composition of g and f by the Chain Rule:

$$\begin{cases} (g \circ f)_z = (g_\zeta \circ f)f_z + (g_{\overline{\zeta}} \circ f)\overline{f}_z, \\ (g \circ f)_{\overline{z}} = (g_\zeta \circ f)f_{\overline{z}} + (g_{\overline{\zeta}} \circ f)\overline{f}_{\overline{z}}. \end{cases}$$

Substituting the results from equations (3.4) into the expressions of the two partial derivatives gives us

$$\begin{cases} (g \circ f)_z = (g_\zeta \circ f)f_z + (g_{\overline{\zeta}} \circ f)\overline{(f_{\overline{z}})}, \\ (g \circ f)_{\overline{z}} = (g_\zeta \circ f)f_{\overline{z}} + (g_{\overline{\zeta}} \circ f)\overline{(f_z)}. \end{cases} \tag{3.5}$$

The Jacobian determinant of f is

$$J_f = \det \begin{bmatrix} f_z & f_{\overline{z}} \\ \overline{f}_z & \overline{f}_{\overline{z}} \end{bmatrix} = \det \begin{bmatrix} f_z & f_{\overline{z}} \\ \overline{(f_{\overline{z}})} & \overline{(f_z)} \end{bmatrix} = |f_z|^2 - |f_{\overline{z}}|^2.$$

Solving for $g_\zeta \circ f$ and $g_{\overline{\zeta}} \circ f$ from the system of equations (3.5) gives

$$\begin{cases} g_\zeta \circ f = \dfrac{1}{J_f}\left[(g \circ f)_z\overline{(f_z)} - (g \circ f)_{\overline{z}}\overline{(f_{\overline{z}})}\right], \\ g_{\overline{\zeta}} \circ f = \dfrac{1}{J_f}\left[(g \circ f)_{\overline{z}} f_z - (g \circ f)_z f_{\overline{z}}\right]. \end{cases}$$

Now if $g = f^{-1}$, then $g \circ f = z$, and thus $g \circ f_z = 1$ and $g \circ f_{\overline{z}} = 0$. Hence,

$$\begin{cases} (f^{-1})_\zeta \circ f = \dfrac{\overline{(f_z)}}{J_f}, \\ (f^{-1})_{\overline{\zeta}} \circ f = -\dfrac{f_{\overline{z}}}{J_f}. \end{cases}$$

3.2 Statement of Riemann Mapping Theorem

One formulation of the famous *Riemann Mapping Theorem* is stated as follows. A complete investigation can be found in the first section of Chapter 6 in [Ahlfors, 1953].

Theorem 3.3 (Riemann Mapping Theorem)**.** Let Ω be a simply connected region in plane, and let D be the unit disk. Fix $\omega_1, \omega_2, \omega_3 \in \partial\Omega$ and $d_1, d_2, d_3 \in \partial D$, counterclockwise oriented.
(1) There is a conformal mapping from Ω to D.
(2) There is a unique conformal mapping $f : \Omega \to D$ that satisfies $f(\omega_i) = d_i$, $i = 1, 2, 3$.

Remark. The formal definition of simply connectedness requires background in fundamental group and group theory. We shall not precisely stating it here but instead give a general idea. Roughly speaking, a space X is said to be *simply connected* if any closed curve in X can shrink into a single point in X.

3.3 Grötzsch Problem

Consider two rectangles, R and R', with counterclockwise prescribed vertices O, A, B, C and O', A', B', C', respectively. Let the vertices O and O' coincide with the origins, and also let the sides of R and R' parallel to the corresponding axes, as indicated in Figure 4. Suppose that $|OA| = a, |OC| = b, |O'A'| = a'$, and $|O'C'| = b'$, where $|\cdot|$ denotes the length of a line segment, such that $\dfrac{a}{b} \neq \dfrac{a'}{b'}$; in other words, R and R' are not similar.

As we previously discussed about conformal mappings, we naturally ask *"Can there be a conformal diffeomorphism from R to R' that maps edges to edges, respectively?"* We will find the answer in the end of this section.

The very instuitive example of diffeomorphisms from R to R' that satisfies the above conditions is the affine mapping

$$u = \frac{a'}{a}x \quad \text{and} \quad v = \frac{b'}{b}y.$$

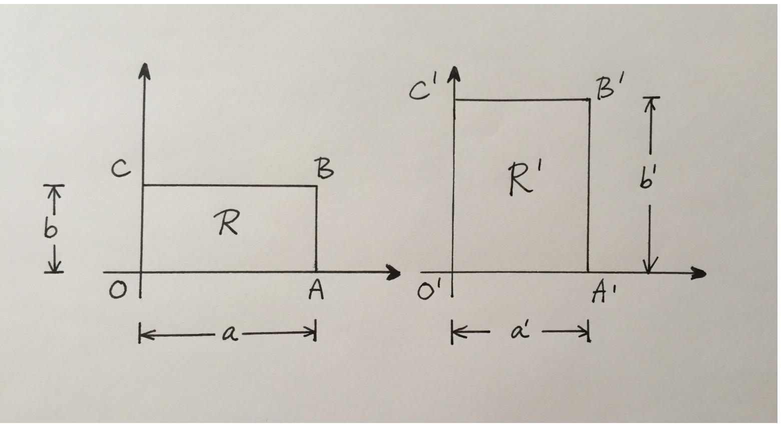


Figure 4: Two rectengles R and R'

Using $w = u + iv, x = \dfrac{z+\overline{z}}{2}$, and $y = \dfrac{z-\overline{z}}{2i}$, we can write the mapping differently:

$$w = \frac{1}{2}\left[\left(\frac{a'}{a}+\frac{b'}{b}\right)z + \left(\frac{a'}{a}-\frac{b'}{b}\right)\overline{z}\right].$$

Recall that the dilatation of a mapping is

$$K = \sup_{\Omega} \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|}.$$

Direct calculations give us the dilatation of the affine mapping as follows:

$$K = \begin{cases} \dfrac{a'b}{b'a} & \text{if } \dfrac{a}{b} < \dfrac{a'}{b'} \\ \dfrac{ab'}{ba'} & \text{if } \dfrac{a'}{b'} < \dfrac{a}{b}. \end{cases} \tag{3.6}$$

Note that the dilatation of such a affine mapping is greater than 1, as long as the two rectangles are not similar.

Theorem 3.4. The dilatation of every possible diffeomorphism from R to R' that maps edges to edges is greater than or equal to both $\dfrac{a'b}{b'a}$ and $\dfrac{ab'}{ba'}$.

Said differently, the dilatation of the affine mapping, (3.6), is the smallest among the dilatations of all possible competitors.

Proof. Let $w : R \to R'$ be a diffeomorphism that maps edged to edges. Let $w(x, y) = u(x, y) + iv(x, y)$.

We have known that $\dfrac{\partial w}{\partial z} = \dfrac{1}{2}(u_x - iu_y + iv_x + v_y)$, $\dfrac{\partial w}{\partial \overline{z}} = \dfrac{1}{2}(u_x + iu_y + iv_x - v_y)$, and thus,

$$w_z + w_{\overline{z}} = u_x + iv_x, \tag{3.7}$$

$$|w_z|^2 - |w_{\overline{z}}|^2 = u_x v_y - u_y v_x = \begin{vmatrix} u_x & u_y \\ v_x & v_y \end{vmatrix} = J_w.$$

Since $u(0, y) = 0$, $u(a, y) = a'$, we integrate u(x,y) along the boundary of R and get

$$\int_{\partial R} u(x, y)\, dy = a'b.$$

By *Stokes' theorem*, we have that

$$\int_{\partial R} u(x, y)\, dy = \int_R \partial_x u(x, y)\, dxdy.$$

Since u_x is the real part of $w_z + w_{\overline{z}}$, then

$$a'b = \int_R \mathrm{Re}(w_z + w_{\overline{z}})\, dxdy.$$

If we let $p, q \in \mathbb{C}$, then $\mathrm{Re}(p + q) = \mathrm{Re}(p) + \mathrm{Re}(q) \leqslant \sqrt{\mathrm{Re}(p)^2} + \sqrt{\mathrm{Re}(q)^2} \leqslant \sqrt{\mathrm{Re}(p)^2 + \mathrm{Im}(p)^2} + \sqrt{\mathrm{Re}(q)^2 + \mathrm{Im}(q)^2} = |p| + |q|$. Therefore,

$$a'b \leqslant \int_R |w_z| + |w_{\overline{z}}|\, dxdy. \tag{3.8}$$

By *Hölder's Inequality*, we have that

$$\begin{aligned} a'b &\leqslant \int_R 1 \cdot (|w_z| + |w_{\overline{z}}|)\, dxdy \\ &\leqslant \left(\int_R 1^2\, dxdy\right)^{\frac{1}{2}} \left(\int_R (|w_z| + |w_{\overline{z}}|)^2\, dxdy\right)^{\frac{1}{2}} \\ &= [\mathrm{Area}(R)]^{\frac{1}{2}} \left(\int_R (|w_z| + |w_{\overline{z}}|)^2\, dxdy\right)^{\frac{1}{2}}. \end{aligned} \tag{3.9}$$

Let K denote the dilatation of w, and then $K = \sup\limits_R \dfrac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|}$. Thus,

$$\begin{aligned} |w_z| + |w_{\overline{z}}| &= \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|} \cdot (|w_z| - |w_{\overline{z}}|) \\ &\leqslant K\,(|w_z| - |w_{\overline{z}}|). \end{aligned}$$

Now, if we times $(|w_z| + |w_{\overline{z}}|)$ on both sides of $|w_z| + |w_{\overline{z}}| \leqslant K\,(|w_z| - |w_{\overline{z}}|)$, we get

$$\begin{aligned} (|w_z| + |w_{\overline{z}}|)^2 &\leqslant K\,\left(|w_z|^2 - |w_{\overline{z}}|^2\right) \\ &= K\,(J_w). \end{aligned}$$

Hence, we get

$$\begin{aligned} a'b &\leqslant \sqrt{\text{Area}(R)}\sqrt{\int_R K\left(J_w\right) dxdy} \\ &= \sqrt{K \cdot \text{Area}(R)}\sqrt{\int_R 1 \cdot \left(J_w\right) dxdy} \\ &= \sqrt{K \cdot \text{Area}(R)}\sqrt{\int_{R'} 1\, dxdy} \\ &= \sqrt{K \cdot \text{Area}(R) \cdot \text{Area}(R')} \\ &= \sqrt{K \cdot ab \cdot a'b'} \end{aligned} \tag{3.10}$$

and thus, $(a'b)^2 \leqslant K \cdot ab \cdot a'b'$, which gives

$$K \geqslant \frac{a'b}{b'a}.$$

If we revert the role of z and w, we shall obtain the other inequality

$$K \geqslant \frac{ab'}{ba'}.$$

Therefore, the affine mapping has the minimal dilatation among all diffeomorphisms from R to R' that map edges to edges. □

Now we ask another natural question: *"Is there any other diffeomorphism, satisfying all the conditions, that share the same dilatation as the affine mapping?"* The answer is given by the following theorem.

Theorem 3.5. If a diffeomorphism from R to R', mapping edges to edges, has a dilatation equal to (3.6), then it is the affine mapping. In other words, the affine mapping is the only such map that has the dilatation in (3.6).

Proof. Based on the inequalities we proved in Theorem 3.4, we now need to determine when each of these equalities holds.

There are three inequalities involved in the proof above; they are (3.8), (3.9), and (3.10). Let's discuss the condition for equality of each case.

First, for (3.8) to attain equality, we must have

$$\text{Re}(w_z + w_{\overline{z}}) = |w_z| + |w_{\overline{z}}|. \tag{3.11}$$

Since this equality involves $\mathrm{Re}(w_z+w_{\overline{z}}) = \mathrm{Re}(w_z)+\mathrm{Re}(w_{\overline{z}}) = \sqrt{\mathrm{Re}(w_z)^2}+\sqrt{\mathrm{Re}(w_{\overline{z}})^2} = \sqrt{\mathrm{Re}(w_z)^2+\mathrm{Im}(w_z)^2}+\sqrt{\mathrm{Re}(w_{\overline{z}})^2+\mathrm{Im}(w_{\overline{z}})^2} = |w_z|+|w_{\overline{z}}|$, we can conclude that

$$\mathrm{Im}(w_z) = 0 \quad \text{and} \quad \mathrm{Im}(w_{\overline{z}}) = 0. \tag{3.12}$$

Recall that $w_z = \dfrac{1}{2}(u_x - iu_y + iv_x + v_y)$ and $w_{\overline{z}} = \dfrac{1}{2}(u_x + iu_y + iv_x - v_y)$. Then (3.12) shows that

$$\begin{cases} w_z = \frac{1}{2}(u_x + v_y) \\ w_{\overline{z}} = \frac{1}{2}(u_x - v_y) \end{cases} \tag{3.13}$$

and

$$\begin{cases} u_y - v_x = 0 \\ u_y + v_x = 0. \end{cases} \tag{3.14}$$

The system (3.14) shows that $u_y = 0$ and $v_x = 0$.

To have equality in (3.9), we need that

$$|w_z| + |w_{\overline{z}}| = \text{constant in}\, R. \tag{3.15}$$

Combining the results from (3.11) and (3.15) gives us that

$$\mathrm{Re}(w_z + w_{\overline{z}}) = \text{constant in}\, R. \tag{3.16}$$

We have known in (3.7) that

$$w_z + w_{\overline{z}} = u_x + iv_x,$$

and therefore, (3.16) is equivalent to

$$u_x = \text{constant in}\, R. \tag{3.17}$$

Moreover, since $v_x = 0$, we actually have

$$w_z + w_{\overline{z}} = u_x.$$

Hence,

$$|w_z| + |w_{\overline{z}}| = \mathrm{Re}(w_z + w_{\overline{z}}) = u_x. \tag{3.18}$$

Last, to obtain equality in (3.10), we need

$$(|w_z| + |w_{\overline{z}}|)^2 = K\, \left(|w_z|^2 - |w_{\overline{z}}|^2\right).$$

Canceling $(|w_z| + |w_{\overline{z}}|)$ from both sides gives us that

$$|w_z| + |w_{\overline{z}}| = K\ (|w_z| - |w_{\overline{z}}|)\,.$$

By (3.15), we directly have $K\ (|w_z| - |w_{\overline{z}}|) = \text{constant in}\, R$, and as a result,

$$|w_z| - |w_{\overline{z}}| = \text{constant in}\, R. \tag{3.19}$$

Then (3.13), (3.17), and (3.19) together shows that

$$v_y = \text{constant in}\, R. \tag{3.20}$$

Using what we have obtained, we can deduce from the equation $(|w_z| + |w_{\overline{z}}|)^2 = K\ \left(|w_z|^2 - |w_{\overline{z}}|^2\right)$ that

$$\begin{aligned} |w_z|^2 - |w_{\overline{z}}|^2 &= K^{-1}(|w_z| + |w_{\overline{z}}|)^2 \\ \Rightarrow \frac{1}{4}(u_x + v_y)^2 - \frac{1}{4}(u_x - v_y)^2 &= K^{-1}(u_x)^2 \\ \Rightarrow u_x v_y &= K^{-1}(u_x)^2 \\ \Rightarrow u_x &= K v_y. \end{aligned}$$

All together, we have that

$$\begin{cases} u_y = 0, \quad u_x = \text{constant in } R \\ v_x = 0, \quad v_y = \text{constant in } R \\ u_x = K v_y. \end{cases}$$

Therefore, w must have the form

$$w = \alpha x + i\beta y,$$

where α and β are constants that match the values for the affine mapping. □

Now we can imply that there is no conformal diffeomorphism from R to R' that maps edges to edges, respectively. Indeed, the conformality would require the dilatation to be equal to 1, which is less than the dilation of the affine mapping – contradicting the fact that stated in Theorem 3.4.

3.4 Quasiconformal Mappings

The conformality of conformal mappings is a quite stringent property; though it is desired, in most cases, we cannot find a mapping to meet the requirements. For example, we have seen that there is no conformal diffeomorphism between non-similar rectangles that maps edges to edges, respectively. In similar cases where conformality is not achievable, we aim at finding a mapping that is, in some sense, as conformal as possible. Thus, we are going to talk about quasiconformal mappings and also make it more precise about how we quantify "how far" a quasiconformal mapping is from being conformal.

Let's first recall our previous discussion in section 3.1.2. If we have an infinitesimally small circle $dz = dre^{i\theta}$, then a complex mapping $w : \mathbb{C} \to \mathbb{C}$ will map it into an ellipse

$$dw = dre^{i(\frac{\alpha+\beta}{2})}(|w_z|e^{i[\theta-\frac{\beta-\alpha}{2}]} + |w_{\overline{z}}|e^{-i[\theta-\frac{\beta-\alpha}{2}]}),$$

whose ratio of long axis to short axis is

$$D_w = \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|}.$$

Moreover, we have seen that for conformal mappings, the ratio is 1 and the image is still a circle.

In this section we are going to talk about another special case in which the ratio D_w is bounded.

3.4.1 What is a Quasiconformal Mapping in Plane

Let's first introduce two terminologies.

We set

$$\mu := \frac{w_{\overline{z}}}{w_z} \tag{3.21}$$

and call it the *complex dilatation*, or the ***first complex dilatation***. From the modulus of μ, $|\mu| = \dfrac{|w_{\overline{z}}|}{|w_z|}$, we notice that

$$1 + |\mu| = \frac{|w_z| + |w_{\overline{z}}|}{|w_z|},$$
$$1 - |\mu| = \frac{|w_z| - |w_{\overline{z}}|}{|w_z|},$$

which gives that

$$\frac{1 + |\mu|}{1 - |\mu|} = \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|}.$$

Hence, we obtain a new representation of D_w, which is

$$D_w = \frac{1+|\mu|}{1-|\mu|}.$$

Next, we set

$$\nu := \frac{w_{\overline{z}}}{(\overline{w_z})} \tag{3.22}$$

and call it the ***second complex dilatation***. Since $|w_z| = |\overline{w_z}|$, then

$$|\mu| = |\nu|.$$

Definition 3.5. Let $\Omega \subset \mathbb{C}$ be open, and let $w : \Omega \to \mathbb{C}$ be a diffeomorphism. Then w is said to be ***quasiconformal*** if

$$D_w = \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|}$$

is bounded in Ω. Furthermore, if there exists a specific $K \in \mathbb{R}$ such that $D_w \leqslant K$, then w is said to be ***K-quasiconformal***.

Remark. For a K-quasiconformal mapping, the induced condition for $|\mu|$ and $|\nu|$ is that $|\mu| = |\nu| \leqslant \frac{K-1}{K+1} < 1$.

Therefore, if the dilatation of the mapping w,

$$K_w := \sup_{\Omega} D_w = \sup_{\Omega} \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|},$$

exists, then w is K_w-quasiconformal.

Remark. A 1-quasiconformal mapping is conformal.

Example 3.3. Let $w(x, y) := 9x + iy$. In complex notation (using w for convenience), one has that

$$w(z, \overline{z}) = 9(\frac{z+\overline{z}}{2}) + i(\frac{z-\overline{z}}{2i}) = 5z + 4\overline{z}.$$

Thus,

$$D_w = \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|} = \frac{5+4}{5-4} = 9.$$

Therefore, w is a quasiconformal mapping. Furthermore, it is 9-quasiconformal.

Now with the definition of quasiconformal mappings, we return to the *Grötzsch Problem* and conclude that there exists a K-quasiconformal mapping from R to R' if and only if

$$\frac{1}{K} \leqslant \frac{a'b}{b'a} \leqslant K.$$

Since this chain of inequalities is symmetric, we derive from it that the inverse of a quasiconformal mapping is still quasiconformal.

3.4.2 The Geometric Definition

We notice that in Definition 3.5 of quasiconformal mappings, we pre-require that the mapping w to be a diffeomorphism. In fact, this condition of differentiability is not natural, and sometimes, not necessary. In other words, this definition is not broad enough to include various examples that are useful in mathematics; we shall remove this condition of differentiability and give an equivalent definition of quasiconformal mappings.

Definition 3.6. Let $\Omega \subset \mathbb{C}$ be open, and let $w : \Omega \to \mathbb{C}$ be a homeomorphism. Let $z \in \Omega$ be a point, and let $r \in \mathbb{R}$ be positive. Set

$$L(z,r) := \sup_{\substack{z' \in \Omega \\ \|z-z'\| \leqslant r}} \|w(z) - w(z')\|,$$

and set

$$l(z,r) := \inf_{\substack{z' \in \Omega \\ \|z-z'\| \geqslant r}} \|w(z) - w(z')\|.$$

Define $H(z,w)$ via

$$H(z,w) := \limsup_{r \to 0} \frac{L(z,r)}{l(z,r)}.$$

We say w is ***quasiconformal*** if there exists a real number $H \geqslant 1$ such that

$$H(z,w) \leqslant H < \infty$$

for all $z \in \mathbb{C}$.

3.4.3 Beltrami and Conjugate Beltrami Equations

From (3.21) and (3.22), the definitions of the *first complex dilatation* and the *second complex dilatation* of w, we obtain the following partial differential equations for w:

$$\frac{\partial w}{\partial \overline{z}} = \mu(z,\overline{z}) \left(\frac{\partial w}{\partial z} \right),$$

$$\frac{\partial w}{\partial \overline{z}} = \nu(z,\overline{z}) \overline{\left(\frac{\partial w}{\partial z} \right)},$$

which are called the *Beltrami equation* and the *Conjugate Beltrami equation*, respectively.

Let $\kappa := \mathrm{Re}(\nu)$, the real part of ν, and let $\lambda := \mathrm{Im}(\nu)$, the imaginary part of ν. Using $w = u + iv, z = x + iy$, and $\nu = \kappa + i\lambda$, we obtain

$$\begin{cases} \dfrac{\partial w}{\partial \overline{z}} = \dfrac{1}{2}(u_x + iu_y + iv_x - v_y) \\ \qquad = \dfrac{1}{2}\left[\left(\dfrac{\partial u}{\partial x} - \dfrac{\partial v}{\partial y}\right) + i\left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right)\right], \\ \dfrac{\partial w}{\partial z} = \dfrac{1}{2}(u_x - iu_y + iv_x + v_y) \\ \qquad = \dfrac{1}{2}\left[\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) - i\left(\dfrac{\partial u}{\partial y} - \dfrac{\partial v}{\partial x}\right)\right], \\ \overline{\left(\dfrac{\partial w}{\partial z}\right)} = \dfrac{1}{2}\left[\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) + i\left(\dfrac{\partial u}{\partial y} - \dfrac{\partial v}{\partial x}\right)\right]. \end{cases}$$

Then the *Conjugate Beltrami* equation

$$\frac{\partial w}{\partial \overline{z}} = \nu(z, \overline{z})\overline{\left(\frac{\partial w}{\partial z}\right)}$$

can be expressed as

$$\frac{1}{2}\left[\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) + i\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] = (\kappa + i\lambda)\frac{1}{2}\left[\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + i\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right)\right].$$

Cancelling $\frac{1}{2}$ from both sides gives

$$\left[\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) + i\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] = (\kappa + i\lambda)\left[\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + i\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right)\right]. \qquad (3.23)$$

The real part of the right hand side of equation (3.23) is

$$\kappa\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) - \lambda\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right),$$

while the imaginary part of the right hand side is

$$\lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \kappa\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right).$$

Let them equal the two corresponding parts of the left hand side of equation (3.23), and we get a system of two equations

$$\begin{cases} (1 + \kappa)\dfrac{\partial v}{\partial y} + \lambda\left(-\dfrac{\partial u}{\partial y}\right) = (1 - \kappa)\dfrac{\partial u}{\partial x} - \lambda\dfrac{\partial v}{\partial x}, \\ \lambda\dfrac{\partial v}{\partial y} + (1 - k)\left(-\dfrac{\partial u}{\partial y}\right) = -\lambda\dfrac{\partial u}{\partial x} + (1 + \kappa)\dfrac{\partial v}{\partial x}. \end{cases}$$

Solve for $\dfrac{\partial v}{\partial y}$ and $\left(-\dfrac{\partial u}{\partial y}\right)$ to get

$$\begin{cases} \dfrac{\partial v}{\partial y} = \dfrac{[(\kappa-1)^2+\lambda^2]\left(\frac{\partial u}{\partial x}\right) - 2\lambda\left(\frac{\partial v}{\partial x}\right)}{1-(\kappa^2+\lambda^2)}, \\ \left(-\dfrac{\partial u}{\partial y}\right) = \dfrac{-2\lambda\left(\frac{\partial u}{\partial x}\right) + [(\kappa+1)^2+\lambda^2]\left(\frac{\partial v}{\partial x}\right)}{1-(\kappa^2+\lambda^2)}. \end{cases}$$

Therefore, we have rewritten the *Conjugate Beltrami* equation as a system of partial differential equations

$$\begin{bmatrix} \dfrac{\partial v}{\partial y} \\ -\dfrac{\partial u}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{(\kappa-1)^2+\lambda^2}{1-(\kappa^2+\lambda^2)} & \dfrac{-2\lambda}{1-(\kappa^2+\lambda^2)} \\ \dfrac{-2\lambda}{1-(\kappa^2+\lambda^2)} & \dfrac{(\kappa+1)^2+\lambda^2}{1-(\kappa^2+\lambda^2)} \end{bmatrix} \begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial x} \end{bmatrix}.$$

Now if we set

$$A := \frac{(\kappa-1)^2+\lambda^2}{1-(\kappa^2+\lambda^2)},$$
$$B := \frac{-2\lambda}{1-(\kappa^2+\lambda^2)},$$
$$C := \frac{(\kappa+1)^2+\lambda^2}{1-(\kappa^2+\lambda^2)},$$

we can rewrite the system of equations in a more concise form:

$$\begin{bmatrix} \dfrac{\partial v}{\partial y} \\ -\dfrac{\partial u}{\partial y} \end{bmatrix} = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial x} \end{bmatrix}. \tag{3.24}$$

We shall return to this form in next chapter.

4 Methods in Jones-Mahadevan Paper

In this chapter, we shall summarize the work of Jones and Mahadevan in their recent paper [Jones and Mahadevan, 2013] which inspires our work in the next chapter. Specifically, we shall explain their approach to the problem and extend the mathematical ideas underlying the core part.

4.1 Problem Description

In [Jones and Mahadevan, 2013], the authors want to quantify the difference between two planar shapes (domains) with finitely prescribed matching points. To this end, they construct quasiconformal mappings between two planar shapes, preserving the landmark points. Additionally, boundary points are supposed to remain on the boundary of the target shape (not necessarily fixed). In this way, the parameter field underlying an appropriate quasiconformal manifests the difference as desired.

Let's translate the problem into mathematical language (notations that have been introduced in the preliminary chapter are continued to be used here.)

Given two planar shapes, Ω_0 and Ω_1, together with a finite number of landmark points, say $\{(\hat{x}_i, \hat{y}_i)\}_{i=1}^N$ for some $N \in \mathbb{N}$, in the domain shape Ω_0, and correspondingly, $\{(\hat{u}_i, \hat{v}_i)\}_{i=1}^N$ in the target shape Ω_1, a quasiconformal mapping $w = (u, v) : \Omega_0 \to \Omega_1$ that preserves the matching of all the landmark points is desired. Further, since solving the conjugate Beltrami equation,

$$\frac{\partial w}{\partial \overline{z}} = \nu(z, \overline{z})\overline{\left(\frac{\partial w}{\partial z}\right)},$$

can give a quasiconformal mapping, with $\kappa := \mathrm{Re}(\nu)$ and $\lambda := \mathrm{Im}(\nu)$, a distribution of $\kappa(x, y)$ and $\lambda(x, y)$ satisfying

$$\begin{cases} \hat{u}_i = u(\hat{x}_i, \hat{y}_i; \kappa, \lambda; \partial\Omega_1) \\ \hat{v}_i = v(\hat{x}_i, \hat{y}_i; \kappa, \lambda; \partial\Omega_1) \end{cases}$$

for all $i \in \{1, 2, ..., N\}$ is desired.

4.2 Optimization Schemes

Theoretically, the continuity of $\kappa(x, y)$ and $\lambda(x, y)$ together with the finite number of constrains will produce infinitely many eligible quasiconformal mappings. However, most of them do not reflect the difference between the two regions correctly. So, among all possibles mappings, the optimal one is desired. A natural question is: what is the criterion to select the optimal mapping? The construction of a criterion takes time and effort. We shall describe the proposals tested in [Jones and Mahadevan, 2013] and explain the related mathematical idea.

Recall that if $|w_{\overline{z}}| \neq 0$, the image of the circle $dz = dre^{i\theta}$ under a complex map $w : \mathbb{C} \to \mathbb{C}$ is an ellipse with

$$D_w := \frac{|w_z| + |w_{\overline{z}}|}{|w_z| - |w_{\overline{z}}|}.$$

as its ratio of long axis to short axis. With the previously defined second complex dilatation, $\nu := \frac{w_{\overline{z}}}{(\overline{w_z})}$, the ratio can also be expressed as

$$D_w = \frac{1 + |\nu|}{1 - |\nu|}.$$

To read the differences between the shapes of two domains from a collection of possible quasiconformal mappings, the extreme one with minimum maximum degree of local shear is concerned. Thus, the first proposal of Jones and Mahadevan is that the maximum value of $|\nu|$ on Ω_0 be minimized. However, in their trials, perturbations of κ and λ (the real and imaginary parts of ν, respectively) result in an unchanged maximum value of $|\nu|$, and thus, the scheme fails to find a direction towards a solution. Next, the proposal to minimize $[\int_{\Omega_0} |\lambda|^p \, \mathrm{d}S]^{\frac{1}{p}}$ is experimented numerically. Considering the relation between the L^p norm and the L^∞ norm, when p goes to infinity, the value will approach the maximum value of $|\nu|$. New issues appear in this trial: the optimal solution has special values of $|\nu|$ near landmark points, whereas positions of these constraints are supposed to be arbitrary. After they minimize the gradient of $|\nu| = \left|\sqrt{\kappa^2 + \lambda^2}\right|$, they obtain an optimal solution with constant $|\nu|$ across Ω_0. As the local shear property of a quasiconformal mapping can be revealed in the corresponding ν, the optimal mapping will have a constant local shear. The problem with that scheme is that the angle of local shear cannot be controlled. To avoid drastic variation of the angle of local shear, Jones and Mahadevan turn to minimize the gradients of both κ and λ. So, they set the objective function

$$\mathcal{I} = \int_{\Omega_0} (|\nabla\kappa|^2 + |\nabla\lambda|^2) \, \mathrm{d}S.$$

In this case, a pair of functions (κ, λ) that minimizes the objective function is desired. Also, the matching of landmark points is required in the process. Then, with the pair

(κ, λ), the optimal mapping can be solved from the conjugate Beltrami equation, $w_{\overline{z}} = \nu(z, \overline{z})\overline{(w_z)}$. This is the finally settled scheme.

4.3 Variational Principle

The last scheme introduce above is theoretically feasible. However, deriving a quasiconformal mapping from a conjugate Beltrami equation is non-trivial. A further method to compute a solution is needed in current step. We shall discuss about it in this section. Besides the work of Jones and Mahadevan, the standard reference here also includes [Garabedian, 1970].

In the subsection 3.4.3, we have converted the conjugate Beltrami equation,

$$\frac{\partial w}{\partial \overline{z}} = \nu(z, \overline{z})\overline{\left(\frac{\partial w}{\partial z}\right)},$$

into a system of partial differential equations as follows:

$$\begin{bmatrix} \dfrac{\partial v}{\partial y} \\ -\dfrac{\partial u}{\partial y} \end{bmatrix} = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial x} \end{bmatrix}, \tag{4.1}$$

where

$$\begin{cases} A = \dfrac{(\kappa-1)^2+\lambda^2}{1-(\kappa^2+\lambda^2)}, \\ B = \dfrac{-2\lambda}{1-(\kappa^2+\lambda^2)}, \\ C = \dfrac{(\kappa+1)^2+\lambda^2}{1-(\kappa^2+\lambda^2)}. \end{cases}$$

Based on that, a functional is introduced in Jones-Mahadevan paper as follows:

$$\mathcal{F}[u, v] := \frac{1}{2}\int_{\Omega_0} \left(A|\nabla u|^2 + 2B\nabla u \cdot \nabla v + C|\nabla v|^2\right)\, dS. \tag{4.2}$$

In the corresponding appendix of [Jones and Mahadevan, 2013], it is proved that if a homeomorphism $w = (u, v) : \Omega_0 \to \Omega_1$ minimizes $\mathcal{F}$ in (4.2), then it is also a solution to the system of equations in (4.1). Therefore, with a known value of second complex dilatation, $\nu = \kappa + i\lambda$, if we can find the minimizer function of $\mathcal{F}$, then we simultaneously derive a quasiconformal mapping from a conjugate Beltrami equation.

Let's consider the minimization of the functional $\mathcal{F}$ through variational method. Suppose a minimizer exists. Define a one-variable function f via

$$f(s) := \mathcal{F}[u + s\tilde{u}, v + s\tilde{v}],$$

where $\tilde{w} := (\tilde{u}, \tilde{v})$ is an admissible test mapping from Ω_0 to Ω_1. If (u, v) is a minimizer among all competitors (i.e., possible homeomorphisms from Ω_0 to Ω_1), then

$$\mathcal{F}[u, v] \leqslant \mathcal{F}[u + s\tilde{u}, v + s\tilde{v}], \quad \forall s,$$

same as

$$f(0) \leqslant f(s), \quad \forall s.$$

Hence we must have

$$\begin{aligned} 0 = f'(0) &= \lim_{s \to 0} \frac{f(s) - f(0)}{s} \\ &= \lim_{s \to 0} \frac{\mathcal{F}[u + s\tilde{u}, v + s\tilde{v}] - \mathcal{F}[u, v]}{s}. \end{aligned}$$

As $\mathcal{F}[u, v] = \frac{1}{2} \int_{\Omega_0} \left(A|\nabla u|^2 + 2B\nabla u \cdot \nabla v + C|\nabla v|^2 \right) \, dS$, one shall have

$$\begin{aligned} 0 &= \frac{d}{ds} \int_{\Omega_0} [A(\nabla u + s\nabla\tilde{u}) \cdot (\nabla u + s\nabla\tilde{u}) + 2B(\nabla u + s\nabla\tilde{u}) \cdot (\nabla v + s\nabla\tilde{v}) \\ &\qquad\qquad + C(\nabla v + s\nabla\tilde{v}) \cdot (\nabla v + s\nabla\tilde{v})]\Big|_{s=0} \, dS \\ &= \int_{\Omega_0} \frac{d}{ds} [A(\nabla u + s\nabla\tilde{u}) \cdot (\nabla u + s\nabla\tilde{u}) + 2B(\nabla u + s\nabla\tilde{u}) \cdot (\nabla v + s\nabla\tilde{v}) \\ &\qquad\qquad + C(\nabla v + s\nabla\tilde{v}) \cdot (\nabla v + s\nabla\tilde{v})]\Big|_{s=0} \, dS \\ &= \int_{\Omega_0} [A\nabla\tilde{u} \cdot (\nabla u + s\nabla\tilde{u}) + A(\nabla u + s\nabla\tilde{u}) \cdot \nabla\tilde{u} \\ &\qquad\qquad + 2B\nabla\tilde{u} \cdot (\nabla v + s\nabla\tilde{v}) + 2B(\nabla u + s\nabla\tilde{u}) \cdot \nabla\tilde{v} \\ &\qquad\qquad + C\nabla\tilde{v} \cdot (\nabla v + s\nabla\tilde{v}) + C(\nabla v + s\nabla\tilde{v}) \cdot \nabla\tilde{v}]\Big|_{s=0} \, dS \\ &\overset{s=0}{=} \int_{\Omega_0} [2A\nabla\tilde{u} \cdot \nabla u + 2B\nabla\tilde{u} \cdot \nabla v + 2B\nabla\tilde{v} \cdot \nabla u + 2C\nabla\tilde{v} \cdot \nabla v] \, dS \\ &= 2 \int_{\Omega_0} [\nabla\tilde{u} \cdot (A\nabla u + B\nabla v) + \nabla\tilde{v} \cdot (B\nabla u + C\nabla v)] \, dS \end{aligned}$$

So, if (u, v) is a minimizer of $\mathcal{F}$, then for every possible admissible test mapping $(\tilde{u}, \tilde{v}) : \Omega_0 \to \Omega_1$ that is not independent at the boundary, one has that

$$\int_{\Omega_0} [\nabla\tilde{u} \cdot (A\nabla u + B\nabla v) + \nabla\tilde{v} \cdot (B\nabla u + C\nabla v)] \, dS = 0.$$

Let

$$\delta\mathcal{F} := \int_{\Omega_0} [\nabla\tilde{u} \cdot (A\nabla u + B\nabla v) + \nabla\tilde{v} \cdot (B\nabla u + C\nabla v)] \, dS.$$

Then, in Jones and Mahadevan's method, a minimizer $w = (u, v)$ is found by setting

$$\delta\mathcal{F} = 0.$$

In trial changes, with a guess of $\nu = \kappa + i\lambda$, a series of perturbations of κ and λ are made in order to minimize

$$\mathcal{I} = \int_{\Omega_0} \left(|\nabla\kappa|^2 + |\nabla\lambda|^2\right) dS$$

with the constraints that the solution of each corresponding conjugate Beltrami equation, $w = (u, v)$, which is solved from

$$\delta\mathcal{F} = 0,$$

satisfies that

$$w(\hat{x_i}, \hat{y_i}) = (\hat{u_i}, \hat{v_i}), \quad i = 1, ..., N.$$

4.4 Numerical Implementation

Numerically, the scheme is implemented through a finite-element method.

The first step is to introduce a triangulation of the domain Ω_0. Suppose there are N_I nodes in the interior and N_B nodes on the boundary. In addition, suppose the image of the N_I interior vertices under $w = (u, v)$ are indexed as

$$\{(u_1, v_1), (u_2, v_2), ..., (u_{N_I}, v_{N_I})\}.$$

For boundary points, we first parametrize the boundary of Ω_1 by $u_B : [0, 2\pi] \to \Omega_1$, and then for each point j of the N_B nodes on $\partial\Omega_0$ ($j = 1, 2, ..., N_B$), give it a specific value $\theta_j \in [0, 2\pi]$ of the parametrizing variable. After that, suppose the image of the N_B boundary vertices under $w = (u, v)$ are indexed as

$$\{(u_B(\theta_1), v_B(\theta_1)), (u_B(\theta_2), v_B(\theta_2)), ..., (u_B(\theta_{N_B}), v_B(\theta_{N_B}))\}.$$

Let $\{\phi_i(x, y)\}_{i=1}^{N_I}$ and $\{\phi_{Bj}(x, y)\}_{j=1}^{N_B}$ be two collections piecewise affine basis functions. That is, for each interior point i, ϕ_i is affine on its corresponding triangles such that $\phi_i(i) = 1$ and $\phi_i(k) = 0$ for $k \neq i$; each ϕ_{Bj} has the same property with respect to boundary nodes. If the triangulation mesh size is h, then the corresponding discretized expressions of the mapping are

$$u^h(x,y) := \sum_{i=1}^{N_I} u_i\phi_i(x,y) + \sum_{j=1}^{N_B} u_B(\theta_j)\phi_{Bj}(x,y),$$
$$v^h(x,y) := \sum_{i=1}^{N_I} v_i\phi_i(x,y) + \sum_{j=1}^{N_B} v_B(\theta_j)\phi_{Bj}(x,y).$$

Likewise, discretize κ and λ by

$$\kappa^h(x,y) := \sum_{t=1}^{N_I+N_B} \kappa_t\phi_t(x,y)$$
$$\lambda^h(x,y) := \sum_{t=1}^{N_I+N_B} \lambda_t\phi_t(x,y).$$

According to the discretization of (u,v), the admissible test mapping $(\tilde{u},\tilde{v})$ mentioned in previous is obtained by making perturbations to the coefficients $\{u_i\}_{i=1}^{N_I}$, $\{v_i\}_{i=1}^{N_I}$, and $\{\theta_j\}_{j=1}^{N_B}$, and thus,

$$\tilde{u}^h = \sum_{i=1}^{N_I} \tilde{u}_i\phi_i(x,y) + \sum_{j=1}^{N_B} \tilde{\theta}_j u'_B(\theta_j)\phi_{Bj}(x,y),$$
$$\tilde{v}^h = \sum_{i=1}^{N_I} \tilde{v}_i\phi_i(x,y) + \sum_{j=1}^{N_B} \tilde{\theta}_j v'_B(\theta_j)\phi_{Bj}(x,y).$$

Recall that the goal is to find the minimizer of the discretized

$$\tilde{\mathcal{I}} = \int_{\Omega_0} \left(|\nabla\kappa^h|^2 + |\nabla\lambda^h|^2\right) dS$$

with the constraints that the solution of each corresponding conjugate Beltrami equation, $w = (u,v)$, which is solved from

$$\delta\mathcal{F} = 0,$$

satisfies that

$$w(\hat{x_i},\hat{y_i}) = (\hat{u_i},\hat{v_i}), \quad i = 1,...,N.$$

Considering the expression of $\delta\mathcal{F}$,

$$\delta\mathcal{F} = \int_{\Omega_0} \left[\nabla\tilde{u}\cdot(A\nabla u + B\nabla v) + \nabla\tilde{v}\cdot(B\nabla u + C\nabla v)\right] dS = 0,$$

we first do some necessary calculations with respect to the discretized terms as follows:

$$\begin{cases} \nabla u^h = \sum_{i=1}^{N_I} u_i \nabla\phi_i + \sum_{j=1}^{N_B} u_B(\theta_j)\nabla\phi_{Bj} \\ \nabla v^h = \sum_{i=1}^{N_I} v_i \nabla\phi_i + \sum_{j=1}^{N_B} v_B(\theta_j)\nabla\phi_{Bj} \\ \nabla \tilde{u}^h = \sum_{i=1}^{N_I} \tilde{u}_i \nabla\phi_i + \sum_{j=1}^{N_B} \tilde{\theta}_j u'_B(\theta_j)\nabla\phi_{Bj} \\ \nabla \tilde{v}^h = \sum_{i=1}^{N_I} \tilde{v}_i \nabla\phi_i + \sum_{j=1}^{N_B} \tilde{\theta}_j v'_B(\theta_j)\nabla\phi_{Bj} \end{cases}$$

Now we can obtain that

$$\begin{aligned}
&\int_{\Omega_0} \left[\nabla\tilde{u}^h \cdot (A\nabla u^h + B\nabla v^h)\right] dS \\
&= \int_{\Omega_0} \left(\sum_{m=1}^{N_I} \tilde{u}_m \nabla\phi_m + \sum_{n=1}^{N_B} \tilde{\theta}_n u'_B(\theta_n)\nabla\phi_{Bn}\right) \left[A\left(\sum_{i=1}^{N_I} u_i\nabla\phi_i + \sum_{j=1}^{N_B} u_B(\theta_j)\nabla\phi_{Bj}\right)\right. \\
&\qquad\qquad \left. + B\left(\sum_{i=1}^{N_I} v_i\nabla\phi_i + \sum_{j=1}^{N_B} v_B(\theta_j)\nabla\phi_{Bj}\right)\right] dS \\
&= \int_{\Omega_0} \left(\sum_{m=1}^{N_I} \tilde{u}_m \nabla\phi_m + \sum_{n=1}^{N_B} \tilde{\theta}_n u'_B(\theta_n)\nabla\phi_{Bn}\right) \left[\nabla\phi_i\left(A\sum_{i=1}^{N_I} u_i + B\sum_{i=1}^{N_I} v_i\right) dS\right. \\
&\qquad\qquad \left. + \nabla\phi_{Bj}\left(A\sum_{j=1}^{N_B} u_B(\theta_j) + B\sum_{j=1}^{N_B} v_B(\theta_j)\right)\right] dS \\
&= \int_{\Omega_0} \left(\sum_{m=1}^{N_I} \tilde{u}_m \nabla\phi_m + \sum_{n=1}^{N_B} \tilde{\theta}_n u'_B(\theta_n)\nabla\phi_{Bn}\right) \left[\sum_{i=1}^{N_I} (Au_i + Bv_i)\nabla\phi_i\right. \\
&\qquad\qquad \left. + \sum_{j=1}^{N_B} [Au_B(\theta_j) + Bv_B(\theta_j)]\nabla\phi_{Bj}\right] dS \\
&= \sum_{m=1}^{N_I} \tilde{u}_m \int_{\Omega_0} \nabla\phi_m \left[\sum_{i=1}^{N_I} (Au_i + Bv_i)\nabla\phi_i + \sum_{j=1}^{N_B} [Au_B(\theta_j) + Bv_B(\theta_j)]\nabla\phi_{Bj}\right] dS \\
&\quad + \sum_{n=1}^{N_B} \tilde{\theta}_n \int_{\Omega_0} u'_B(\theta_n)\nabla\phi_{Bn} \left[\sum_{i=1}^{N_I} (Au_i + Bv_i)\nabla\phi_i + \sum_{j=1}^{N_B} [Au_B(\theta_j) + Bv_B(\theta_j)]\nabla\phi_{Bj}\right] dS.
\end{aligned}$$

Likewise,

$$
\begin{aligned}
&\int_{\Omega_0} \left[\nabla \tilde{v}^h \cdot \left(B \nabla u^h + C \nabla v^h \right) \right] \, dS \\
&= \sum_{m=1}^{N_I} \tilde{v}_m \int_{\Omega_0} \nabla \phi_m \left[\sum_{i=1}^{N_I} \left(B u_i + C v_i \right) \nabla \phi_i + \sum_{j=1}^{N_B} \left[B u_B(\theta_j) + C v_B(\theta_j) \right] \nabla \phi_{Bj} \right] dS \\
&\quad + \sum_{n=1}^{N_B} \tilde{\theta}_n \int_{\Omega_0} v_B'(\theta_n) \nabla \phi_{Bn} \left[\sum_{i=1}^{N_I} \left(B u_i + C v_i \right) \nabla \phi_i + \sum_{j=1}^{N_B} \left[B u_B(\theta_j) + C v_B(\theta_j) \right] \nabla \phi_{Bj} \right] \, dS
\end{aligned}
$$

Combine them together to get the discretized counterpart of $\delta\mathcal{F}$:

$$\begin{aligned}
\delta\mathcal{F}^h &= \int_{\Omega_0} \left[\nabla\tilde{u}^h \cdot (A\nabla u^h + B\nabla v^h) + \nabla\tilde{v}^h \cdot (B\nabla u^h + C\nabla v^h)\right] dS \\
&= \sum_{m=1}^{N_I} \tilde{u}_m \int_{\Omega_0} \nabla\phi_m \left[\sum_{i=1}^{N_I} (Au_i + Bv_i)\,\nabla\phi_i + \sum_{j=1}^{N_B} \left[Au_B(\theta_j) + Bv_B(\theta_j)\right] \nabla\phi_{Bj}\right] dS \\
&\qquad + \sum_{m=1}^{N_I} \tilde{v_m} \int_{\Omega_0} \nabla\phi_m \left[\sum_{i=1}^{N_I} (Bu_i + Cv_i)\,\nabla\phi_i + \sum_{j=1}^{N_B} \left[Bu_B(\theta_j) + Cv_B(\theta_j)\right] \nabla\phi_{Bj}\right] dS \\
&\qquad + \sum_{n=1}^{N_B} \tilde{\theta}_n \int_{\Omega_0} \nabla\phi_{Bn} \left[Au'_B(\theta_n) \left(\sum_{i=1}^{N_I} u_i\nabla\phi_i + \sum_{j=1}^{N_B} u_B(\theta_j)\nabla\phi_{Bj}\right)\right. \\
&\qquad\qquad \left. + \; Bu'_B(\theta_n) \left(\sum_{i=1}^{N_I} v_i\nabla\phi_i + \sum_{j=1}^{N_B} v_B(\theta_j)\nabla\phi_{Bj}\right)\right] dS \\
&\qquad + \sum_{n=1}^{N_B} \tilde{\theta}_n \int_{\Omega_0} \nabla\phi_{Bn} \left[Bv'_B(\theta_n) \left(\sum_{i=1}^{N_I} u_i\nabla\phi_i + \sum_{j=1}^{N_B} u_B(\theta_j)\nabla\phi_{Bj}\right)\right. \\
&\qquad\qquad \left. + \; Cv'_B(\theta_n) \left(\sum_{i=1}^{N_I} v_i\nabla\phi_i + \sum_{j=1}^{N_B} v_B(\theta_j)\nabla\phi_{Bj}\right)\right] dS \\
&= \sum_{m=1}^{N_I} \tilde{u}_m \int_{\Omega_0} \nabla\phi_m \left[\sum_{i=1}^{N_I} (Au_i + Bv_i)\,\nabla\phi_i + \sum_{j=1}^{N_B} \left[Au_B(\theta_j) + Bv_B(\theta_j)\right] \nabla\phi_{Bj}\right] dS \\
&\qquad + \sum_{m=1}^{N_I} \tilde{v}_m \int_{\Omega_0} \nabla\phi_m \left[\sum_{i=1}^{N_I} (Bu_i + Cv_i)\,\nabla\phi_i + \sum_{j=1}^{N_B} \left[Bu_B(\theta_j) + Cv_B(\theta_j)\right] \nabla\phi_{Bj}\right] dS \\
&\qquad + \sum_{n=1}^{N_B} \tilde{\theta}_n \int_{\Omega_0} \left[[Au'_B(\theta_n) + Bv'_B(\theta_n)]\nabla\phi_{Bn} \left(\sum_{i=1}^{N_I} u_i\nabla\phi_i + \sum_{j=1}^{N_B} u_B(\theta_j)\nabla\phi_{Bj}\right)\right. \\
&\qquad\qquad \left. + \; [Bu'_B(\theta_n) + Cv'_B(\theta_n)]\nabla\phi_{Bn} \left(\sum_{i=1}^{N_I} v_i\nabla\phi_i + \sum_{j=1}^{N_B} v_B(\theta_j)\nabla\phi_{Bj}\right)\right] dS.
\end{aligned}$$

If for each $m \in \{1, 2, ..., N_I\}$ and $n \in \{1, 2, ..., N_B\}$ we set

$$\begin{aligned}
S_m^1 &:= \int_{\Omega_0} \nabla\phi_m \left[\sum_{i=1}^{N_I} (Au_i + Bv_i)\,\nabla\phi_i + \sum_{j=1}^{N_B} \left[Au_B(\theta_j) + Bv_B(\theta_j)\right] \nabla\phi_{Bj}\right] dS, \\
S_m^2 &:= \int_{\Omega_0} \nabla\phi_m \left[\sum_{i=1}^{N_I} (Bu_i + Cv_i)\,\nabla\phi_i + \sum_{j=1}^{N_B} \left[Bu_B(\theta_j) + Cv_B(\theta_j)\right] \nabla\phi_{Bj}\right] dS,
\end{aligned}$$

and

$$S_n^3 := \int_{\Omega_0} \left[[Au_B'(\theta_n) + Bv_B'(\theta_n)] \nabla \phi_{Bn} \left(\sum_{i=1}^{N_I} u_i \nabla \phi_i + \sum_{j=1}^{N_B} u_B(\theta_j) \nabla \phi_{Bj} \right) \right.$$
$$\left. + \, [Bu_B'(\theta_n) + Cv_B'(\theta_n)] \nabla \phi_{Bn} \left(\sum_{i=1}^{N_I} v_i \nabla \phi_i + \sum_{j=1}^{N_B} v_B(\theta_j) \nabla \phi_{Bj} \right) \right] dS,$$

then we obtain $2N_I + N_B$ finite-element equations

$$S_m^1 = 0,$$
$$S_m^2 = 0,$$
$$S_n^3 = 0,$$

for $2N_I + N_B$ coefficients: $\{u_i\}_{i=1}^{N_I}$, $\{v_i\}_{i=1}^{N_I}$, and $\{\theta_j\}_{j=1}^{N_B}$.

Remark. The A, B, C shown above in this subsection are actually generated from the corresponding κ^h and λ^h. They can be understood as A^h, B^h, C^h.

With the above notations, we can restate the optimization process. Now the objective is to find the minimizer of

$$\tilde{\mathcal{I}} = \int_{\Omega_0} \left(|\nabla \kappa^h|^2 + |\nabla \lambda^h|^2 \right) dS$$

via perturbations of coefficients $u_i, v_i, \theta_j, \kappa_t$, and λ_t, such that

$$S_m^1 = 0, S_m^2 = 0, S_n^3 = 0,$$

and also that

$$(u^h, v^h)(\hat{x_i}, \hat{y_i}) = (\hat{u_i}, \hat{v_i}), \quad i = 1, ..., N.$$

After introducing the algorithm, the optimization problem now can be implemented via computer programs such as triangulation code, mesh generator, and optimization code. For a deep understanding of the programs that have been applied in Jones and Mahadevan's work, please find [Persson and Strang, 2004] and [Gill et al., 2005], which are from the original code authors. Moreover, a step-by-step description of the optimization procedure can be found in [Jones and Mahadevan, 2013], Appendix B.

5 Alternative Approach

We observe from Jones and Mahadevan's method that their optimization process involves 4^{th} degree of derivatives, and therefore, when it comes to numerical implementations, the error will be multiplied. So, we want to come up with an alternative approach such that a lower order of derivative will be involved in order to reduce the error in approximation.

In this chapter, we shall present a different method that quantifies the difference between two planar shapes with finitely prescribed matching points. Inspired by the work in [Jones and Mahadevan, 2013] and [Capogna and Raich, 2012], here we also use quasiconformal maps to reflect information about local shear and distortion, and we consider various functional of these quantities to measure the difference. To eliminate ambiguity, we aim to find a canonical quasiconformal map relating two given shapes among infinitely many choices and then take the corresponding distortion as a measure of the difference between the two planar shapes.

5.1 Notations

In this section we introduce several important notations that will be used in the rest of this chapter when describing the idea of our approach.

For a planar mapping u, let

$$g := \frac{\mathrm{d}u^T \mathrm{d}u}{|\det \mathrm{d}u|}.$$

The ***dilation*** of u is defined as $\sqrt{\operatorname{trace}(g)}$. By Lemma 3.2, we have that

$$\sqrt{\operatorname{trace}(g)} = \frac{|\mathrm{d}u|}{(\det \mathrm{d}u)^{\frac{1}{2}}}.$$

Define the ***distortion tensor*** of u at a point of differentiability $x \in \mathbb{R}^2$ via

$$S(g) := g - \frac{\operatorname{trace}(g)}{n}\mathrm{Id}.$$

Now if we denote the eigenvalues of g by $0 \leqslant \lambda_1 \leqslant \lambda_2$, then

$$\lambda_1\lambda_2 = \det(g) = \det\left(\frac{\mathrm{d}u^T \mathrm{d}u}{|\det \mathrm{d}u|}\right) = \frac{\det(\mathrm{d}u^T \mathrm{d}u)}{(\det \mathrm{d}u)^2} = \frac{(\det \mathrm{d}u)^2}{(\det \mathrm{d}u)^2} = 1.$$

Further, if g is diagonal, i.e.,

$$g = \Lambda := \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

then

$$\begin{aligned} S(\Lambda) &= \Lambda - \frac{\text{trace}(\Lambda)}{2}\text{Id} \\ &= \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} - \frac{\lambda_1+\lambda_2}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{\lambda_1-\lambda_2}{2} & 0 \\ 0 & -\frac{\lambda_1-\lambda_2}{2} \end{bmatrix}. \end{aligned}$$

Thus,

$$|S(\Lambda)|^2 = \left(\frac{\lambda_1-\lambda_2}{2}\right)^2 + \left(-\frac{\lambda_1-\lambda_2}{2}\right)^2 = \frac{(\lambda_1-\lambda_2)^2}{2} = \frac{1}{2}(\lambda_1+\lambda_2)^2 - 2\lambda_1\lambda_2.$$

$$|S(\Lambda)|^2 = \frac{1}{2}\left[\text{trace}(\Lambda)^2 - 4\right].$$

As g is a symmetric matrix (i.e., $g^T = g$), we can find a 2×2 matrix Q with $Q^TQ = \text{Id}$ such that $QgQ^T = \Lambda$, and thus,

$$\text{trace}(g) = \text{trace}(g\text{Id}) = \text{trace}(gQ^TQ) = \text{trace}(QgQ^T) = \text{trace}(\Lambda) = \lambda_1 + \lambda_2.$$

We observe that $S(g) = \left(g - \dfrac{\text{trace}(g)}{2}\text{Id}\right)$ is also symmetric. Moreover,

$$QS(g)Q^T = QgQ^T - Q\frac{\text{trace}(g)}{2}\text{Id}Q^T = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} - \begin{bmatrix} \frac{\lambda_1+\lambda_2}{2} & 0 \\ 0 & \frac{\lambda_1+\lambda_2}{2} \end{bmatrix} = S(\Lambda).$$

Hence, for a general g, we have

$$\begin{aligned} |S(g)|^2 &= \text{trace}(S(g)S(g)^T) \\ &= \text{trace}(S(g)S(g)^TQ^TQ) \\ &= \text{trace}(QS(g)S(g)^TQ^T) \\ &= \text{trace}(QS(g)Q^TQS(g)^TQ^T) \\ &= \text{trace}\left[\left(QS(g)Q^T\right)\left(QS(g)Q^T\right)^T\right] \\ &= \text{trace}[S(\Lambda)S(\Lambda)^T] \\ &= |S(\Lambda)|^2 \end{aligned}$$

Therefore,

$$|S(g)|^2 = \frac{1}{2}\left[\text{trace}(\Lambda)^2 - 4\right] = \frac{1}{2}\left[\text{trace}(g)^2 - 4\right].$$

With the discussions of g and $S(g)$ above, one can conclude that a planar diffeomorphism u is *conformal* if and only if $S(g) = \mathbf{0}$ (2×2 zero matrix) and if and only if trace$(g) = 2$ ($g = \mathrm{Id}$ indeed).

We have introduced the definition of a quasiconformal mapping in plane in section 3.4. Now let's introduce an equivalent definition, which will be the start point of our approach to quantify the difference between two given planar shapes.

Definition 5.1. A diffeomorphism $u : \Omega_0 \to \Omega_1$ is said to be *quasiconformal* if

$$\mathbb{K}(u, \Omega_0) := \left\| \frac{|\mathrm{d}u|}{(\det \mathrm{d}u)^{\frac{1}{2}}} \right\|_{L^\infty} < \infty.$$

We call $\mathbb{K}(u, \Omega_0)$ the *maximal dilation* of u.

In general, $\mathbb{K}(u, \Omega_0) \geqslant \sqrt{2}$. The equality holds if and only if u is conformal.

5.2 Scheme Frame

We first restate here that our objective is to obtain a quantified difference between these two shapes. Given two planar shapes Ω_0 and Ω_1 and a finite number of landmark points, we can build infinitely many quasiconformal mappings between Ω_0 and Ω_1 with these landmark points matched. Suppose $\{u_\alpha\}_{\alpha \in J}$ is the collection of all those possible quasiconformal mappings. For each u_α, there is a corresponding maximal dilation $\mathbb{K}(u_\alpha, \Omega_0)$, which measures the distortion of the mapping. Roughly speaking, our idea is to first find a u_β in $\{u_\alpha\}_{\alpha \in J}$ such that $\mathbb{K}(u_\beta, \Omega_0) \leqslant \mathbb{K}(u_\alpha, \Omega_0)$ for all $\alpha \in J$, and then we use the dilation of the minimizer, $\mathbb{K}(u_\beta, \Omega_0)$, as the quantified difference between these two planar shapes.

5.3 L^p Gradient Flow Approach

In order to find the minimizer of $\left\| \frac{|\mathrm{d}u|}{(\det \mathrm{d}u)^{\frac{1}{2}}} \right\|_{L^\infty}$,we can equivalently turn to find the minimizer of

$$\left\| \frac{|\mathrm{d}u|^2}{(\det \mathrm{d}u)} \right\|_{L^\infty}.$$

We first study the corresponding L^p problem. That is, we want to find a critical mapping of the following functional

$$\int_{\Omega_0} \frac{|\mathrm{d}u|^{2p}}{(\det \mathrm{d}u)^p}\, dz.$$

By classical variational method, we can obtain the Euler-Lagrange equation for this functional, which is a system of two equations

$$(L_p u)^i = 0, \quad i = 1, 2,$$

where

$$(L_p u)^i = 2p\partial_j \left([\text{trace}(g)]^{p-1}(du^{-1})^T S(g)\right)_{ij}, \quad i, j = 1, 2,$$

and the explicit forms of g and $S(g)$ can be found in the above subsection 5.1. More detailed explanation about this variational problem part can be found in the 8th chapter of [Capogna, 2014].

Now we let $p \to \infty$ in order to go back to the original problem. That is, we want to solve the system of equations

$$(L_\infty u)^i = 0, \quad i = 1, 2. \tag{5.1}$$

However, the major obstacle to achieve our goal in this process is that we currently are not clear about the existence of equation (5.1).

To continue in this method, we introduce a time variable t based on the work of [Capogna, 2014], build a quasiconformal diffeomorphism u_0 that satisfies all the required conditions, and then apply the *gradient flow approach* to study the following initial value problem

$$\begin{cases} \partial_t u = -L_p u & \text{in} \quad \Omega_0 \times (0, T), \\ u = u_0, & \text{at} \quad \Omega_0 \times \{t = 0\}. \end{cases}$$

It has been proved in [Capogna and Raich, 2012] that in a short time, we can obtain a homotopy class of quasiconformal mappings with the same domain and range. Since the flow decreases the mean distortion over time, for a feasible value of T, we let $t \to T$ in order to find a canonical minimizer of mean dilation. After that we let $p \to \infty$ so as to go back to the L^∞ problem. Then the distortion of the minimizer shall be used as a measure of the difference between the two planar shapes that we want to compare.

Since the long-time behavior of the gradient flow is not clear, we resort to numerical experiment with MATLAB programming to study the problem.

In our numerical trials with MATLAB, we observe that, since $(L_p u)^i$ $(i = 1, 2)$ are second order, nonlinear, system of differential operators, it will take a long time to process the implementation in MATLAB. So, we want to take a linear flow approach, which will be more efficient.

Since the algorithm and programming of the numerical implementation in MATLAB corresponding to the L^p flow can be build on from the one corresponding to the linear flow approach that we are going to apply, and since most of us may be more familiar with this linear flow, we shall first explain the numerical part of the linear flow and then revisit the nonlinear flow.

5.4 Heat Gradient Flow Approach

To begin with, we first present how we find this alternative linear flow approach and why these two methods are equivalent.

5.4.1 Relation with the L^p Gradient Flow

For $U : D_1 \to D_2$, a quasiconformal diffeomorphism, the inverse mapping, U^{-1}, is well-defined, and

$$\begin{aligned} U^{-1}(U(z)) &= z, \quad \forall z \in D_1, \\ U(U^{-1}(\zeta)) &= \zeta, \quad \forall \zeta \in D_2. \end{aligned}$$

By the changing of variable $\zeta = U(z)$, we obtain that

$$\int_{D_2} \left| \frac{\mathrm{d}[U^{-1}(\zeta)]}{\mathrm{d}\zeta} \right|_o^2 \mathrm{d}\zeta = \int_{D_1} \left| \frac{\mathrm{d}[U^{-1}(U(z))]}{\mathrm{d}\zeta} \right|_o^2 \det\left(\frac{\mathrm{d}U(z)}{\mathrm{d}z} \right) \mathrm{d}z.$$

Since

$$\begin{cases} \dfrac{\mathrm{d}[U^{-1}(U(z))]}{\mathrm{d}\zeta} = \left(\dfrac{\mathrm{d}U(z)}{\mathrm{d}z} \right)^{-1}, \\ \det\left(\dfrac{\mathrm{d}U(z)}{\mathrm{d}z} \right) = \dfrac{1}{\det\left(\frac{\mathrm{d}U(z)}{\mathrm{d}z}\right)^{-1}}, \end{cases}$$

then

$$\int_{D_2} \left| \frac{\mathrm{d}[U^{-1}(\zeta)]}{\mathrm{d}\zeta} \right|_o^2 \mathrm{d}\zeta = \int_{D_1} \frac{\left| \left(\frac{\mathrm{d}U(z)}{\mathrm{d}z}\right)^{-1} \right|_o^2}{\det\left[\left(\frac{\mathrm{d}U(z)}{\mathrm{d}z}\right)^{-1} \right]} \mathrm{d}z.$$

That is,

$$\int_{D_2} \left| \mathrm{d}_\zeta (U^{-1}) \right|_o^2 \mathrm{d}\zeta = \int_{D_1} \frac{|\,(\mathrm{d}_z U)^{-1}\,|_o^2}{\det\left[(\mathrm{d}_z U)^{-1}\right]} \mathrm{d}z.$$

Moreover, in the 2×2 matrix case,

$$\frac{|\,(\mathrm{d}_z U)^{-1}\,|_o^2}{\det\left[(\mathrm{d}_z U)^{-1}\right]} = \frac{|\mathrm{d}_z U|_o^2}{\det\left[\mathrm{d}_z U\right]}.$$

It follows that

$$\int_{D_2} \left|\mathrm{d}_\zeta(U^{-1})\right|_o^2 \,\mathrm{d}\zeta = \int_{D_1} \frac{|\mathrm{d}_z U|_o^2}{\det\left[\mathrm{d}_z U\right]}\,\mathrm{d}z, \tag{5.2}$$

which shows that minimizing the Dirichlet energy of U^{-1} (the left hand said) is the same as minimizing the L^2 norm of the mean dilation of $\mathrm{d}_z U$ (the right hand side).

In the two-dimensional case, the relation between the Hilbert-Schmidt norm and the operator norm is that, if A is a real-valued 2×2 matrix, then

$$\frac{|A|^2}{\det A} = \frac{|A|_o^2}{\det A} + \frac{1}{\frac{|A|_o^2}{\det A}}. \tag{5.3}$$

Since the function $K \mapsto K + \frac{1}{K}$ is monotonic in $[1, \infty)$, then the minimization problems in the Hilbert-Schmidt norm and in the operator norm shall have the equivalent solution. Hence, by the change of variable, we obtain an identity between mean dilation and Dirichlet energy. To put it differently, in order to minimize the mean dilation of a quasiconformal diffeomorphism, we can instead minimize the Dirichlet energy of the inverse map.

Since the inverse of a quasiconformal diffeomorphism is also quasiconformal, we finally settle at a new (and equivalent) problem to find the minimizer of

$$\int_{\Omega_0} |\mathrm{d}u|^2 \,\mathrm{d}z,$$

with boundary condition prescribed by having Ω_0 being mapped to Ω_1, instead of minimizing

$$\int_{\Omega_0} \frac{|\mathrm{d}u|^2}{\det\left(\mathrm{d}u\right)}\,\mathrm{d}z.$$

By variational method, we obtain the Euler-Lagrange equation in this case, which is a system of heat equations

$$\Delta u = 0.$$

Then we introduce time variable t and apply the gradient flow method to study the initial value problem

$$\begin{cases} \partial_t u = \Delta u \quad \text{in} \quad \Omega_0 \times (0, T), \\ u = u_0 \quad \text{at} \quad \Omega_0 \times \{t = 0\}. \end{cases}$$

We want to find a canonical minimizer of mean dilation, and then we take the dilation of the minimizer as a quantified difference between two planar shapes.

Likewise, we shall apply numerical implementations to find the desired quantified difference value. We will first introduce common numerical methods dealing with heat equations, and then study the heat gradient flow in numerical experiments.

5.4.2 The One-Dimensional Heat Equation

In the study of partial differential equations (PDE), the topic about heat equations is basic and appears in the beginning part of many PDE books. Roughly speaking, a familiar two-dimensional heat equation describes how temperature gradually changes (decreases, in most cases) over time at different locations of a given area. We shall introduce our algorithms together with implementations on MATLAB in this subsection. We start with the one-dimensional heat equation, and the schemes involved will be the essential foundation for us to tackle the further two-dimensional case.

The general format of the heat equation is

$$u_t - \alpha \Delta u = 0, \tag{5.4}$$

where $u = u(x, t)$ is a real-valued function of the spacial variable x and the time variable t, α is a constant named thermal diffusivity (which is a property of the material of the object in the practical sense), and the Laplacian is taken with respect to x.

In our one-dimensional problem, x is in $\mathbb{R}$, and we may assume that the thermal diffusivity is 1 for convenience. Thus, the heat equation becomes

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 \leqslant x \leqslant L,\, t \geqslant 0, \tag{5.5}$$

where L is a constant number. Furthermore, the equation is subject to some initial and boundary conditions.

Recall the definition of derivatives in Calculus. For a one-dimensional function f, we obtain the slope of the secant line passing through points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ by computing

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

(without loss of generality, we assume that $x_1 < x_2$.) If the distance between x_1 and x_2 is sufficiently small (i.e., $\Delta x := x_2 - x_1$ is sufficiently close to 0 from the right), we roughly take the *difference quotient*

$$\frac{f(x_1 + \Delta x) - f(x_1)}{\Delta x}$$

as the slope of the tangent line (if not vertical) at the point $(x_1, f(x_1))$. The idea of the *difference quotient* inspires us to turn the continuous differential equation problem into a discrete approximation.

In the numerical implementation, we require the time domain to be finite, say, $[0,T]$. We divide the spacial domain $[0,L]$ and the time domain $[0,T]$ uniformly into (N_1-1) and $(M-1)$ sub-intervals, respectively ($N_1, M \in \mathbb{Z}_+$). Therefore, we get N_1 nodes on $[0,L]$, of which the distance between each pair of adjacent nodes, Δx, is equal to $\frac{L}{N_1-1}$. Denote these spacial nodes by

$$x_i = (i-1)\Delta x, \quad i = 1, 2, ..., N_1.$$

Similarly, we obtain the uniformly divided time domain $[0,T]$ with the following M nodes

$$t_m = (m-1)\Delta t, \quad m = 1, 2, ..., M,$$

where $\Delta t = \frac{T}{M-1}$. Here we start the subscripts of the two sequences both from 1 because we want to keep consistent with the indexing format of MATLAB.

For convenience, we use the notation u_i^m instead to denote $u(x_i, t_m)$, the value of u evaluated at location node x_i and time node t_m. Assume the initial condition is $u(x,0) = f_0(x)$, and the boundary conditions are $u(0,t) = u_0$ and $u(L,t) = u_L$. Then, from *Taylor series* and the difference quotient that we discussed above, we have that

$$\left.\frac{\partial u}{\partial t}\right|_{x_i,t_m} = \frac{u_i^{m+1} - u_i^m}{\Delta t} + \mathcal{O}(\Delta t). \tag{5.6}$$

Remark. Alternatively, we can use $\left.\frac{\partial u}{\partial t}\right|_{x_i,t_m} = \frac{u_i^{m+1}-u_i^{m-1}}{2\Delta t} + \mathcal{O}(\Delta t^2)$, the *first order central difference*, or $\left.\frac{\partial u}{\partial t}\right|_{x_i,t_m} = \frac{u_i^m - u_i^{m-1}}{\Delta t} + \mathcal{O}(\Delta t)$, the *first order backward difference*, while the difference quotient we used in (5.6) is called the *first order forward difference.* To learn more about the outcomes through the other two kinds of differences and the one we have chosen, please read the article [Recktenwald, 2004].

The discrete approximation for the quadratic derivative $\frac{\partial^2 u}{\partial x^2}$ (i.e., the left side of the heat equation (5.5)) is not trivial. We shall process it in the following steps.

We will be evaluating at time node t_m; for convenience, we shall omit specific mention of t_m in the parentheses after u when no confusion will be caused. We first get the Taylor series

$$u(x_{i+1}) = u(x_i + \Delta x) = u(x_i) + \Delta x \left.\frac{\partial u}{\partial x}\right|_{x_i} + \frac{(\Delta x)^2}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i} + \frac{(\Delta x)^3}{3!}\left.\frac{\partial^3 u}{\partial x^3}\right|_{x_i} + \cdots \tag{5.7}$$

The substitution of Δx with $-\Delta x$ gives

$$u(x_{i-1}) = u(x_i - \Delta x) = u(x_i) - \Delta x \left.\frac{\partial u}{\partial x}\right|_{x_i} + \frac{(\Delta x)^2}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i} - \frac{(\Delta x)^3}{3!}\left.\frac{\partial^3 u}{\partial x^3}\right|_{x_i} + \cdots \tag{5.8}$$

We add the two equations (5.7) and (5.8) together to get

$$u(x_{i+1}) + u(x_{i-1}) = 2u(x_i) + (\Delta x)^2 \frac{\partial^2 u}{\partial x^2}\bigg|_{x_i} + \mathcal{O}(\Delta x^2),$$

from which we obtain the difference approximation of $\frac{\partial^2 u}{\partial x^2}$ as follows:

$$\frac{\partial^2 u}{\partial x^2}\bigg|_{x_i, t_m} = \frac{u_{i-1}^m - 2u_i^m + u_{i+1}^m}{(\Delta x)^2} + \mathcal{O}(\Delta x^2). \tag{5.9}$$

If both Δt and Δx are sufficiently small, we can omit the two error parts of (5.6) and (5.9) and then substitute them into the heat equation (5.5) to obtain

$$\frac{u_i^{m+1} - u_i^m}{\Delta t} = \frac{u_{i-1}^m - 2u_i^m + u_{i+1}^m}{(\Delta x)^2},$$

and finally,

$$u_i^{m+1} = u_i^m + \frac{\Delta t}{(\Delta x)^2}\left(u_{i-1}^m - 2u_i^m + u_{i+1}^m\right).$$

Therefore, the values of u evaluated at three adjacent location nodes at the same time node t_m can be combined to compute $u(x_i, t_{m+1})$.

5.4.3 The Two-Dimensional Heat Equation Implementation

In the two-dimensional case, the function u is of the spatial variable (x, y) and the time variable t. Still, we assume that the thermal diffusivity is 1 for convenience. Then our heat equation is

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}. \tag{5.10}$$

In this case, we continue using the uniform divisions of the spatial interval $[0, L]$ in the x-direction and of the time domain $[0, T]$. In addition, we assume the spatial interval in the y-direction is $[0, H]$ and we divide it uniformly into $(N_2 - 1)$ sub-intervals $(N_2 \in \mathbb{Z}_+)$. Therefore, we get N_2 nodes on $[0, H]$, and the distance between each pair of adjacent nodes is $\Delta y := \frac{H}{N_2 - 1}$. Denote these spacial nodes by

$$y_j = (j-1)\Delta y, \quad j = 1, 2, ..., N_2.$$

Moreover, we use the notation $u_{i,j}^m$ to denote $u(x_i, y_j, t_m)$, the value of u evaluated at location node (x_i, y_j) and time node t_m.

We directly take advantage of the results we have got from the one-dimensional part and obtain the following discrete approximation equation

$$\frac{u_{i,j}^{m+1} - u_{i,j}^m}{\Delta t} = \frac{u_{i-1,j}^m - 2u_{i,j}^m + u_{i+1,j}^m}{(\Delta x)^2} + \frac{u_{i,j-1}^m - 2u_{i,j}^m + u_{i,j+1}^m}{(\Delta y)^2}.$$

Hence,

$$u_{i,j}^{m+1} = u_{i,j}^{m} + \frac{\Delta t}{(\Delta x)^2}(u_{i-1,j}^{m} - 2u_{i,j}^{m} + u_{i+1,j}^{m}) + \frac{\Delta t}{(\Delta y)^2}(u_{i,j-1}^{m} - 2u_{i,j}^{m} + u_{i,j+1}^{m}). \quad (5.11)$$

In order to solve the two-dimensional equation numerically in MATLAB, we shall do some slight change to the equation (5.11) and make some assumptions which can help simplify our computation as well as to make the problem numerically doable.

First, we set

$$a := \frac{\Delta t}{(\Delta x)^2}, b := \frac{\Delta t}{(\Delta y)^2}, c := 1 - 2a - 2b.$$

Then the equation (5.11) becomes

$$u_{i,j}^{m+1} = a(u_{i-1,j}^{m} + u_{i+1,j}^{m}) + b(u_{i,j-1}^{m} + u_{i,j+1}^{m}) + cu_{i,j}^{m}.$$

As the values of Δx and Δy are both very small, we may set them to be equal in the MATLAB programming (i.e., $\Delta x = \Delta y$). Thus, $a = b, c = 1 - 4a$ and

$$u_{i,j}^{m+1} = a(u_{i-1,j}^{m} + u_{i+1,j}^{m} + u_{i,j-1}^{m} + u_{i,j+1}^{m}) + cu_{i,j}^{m}. \quad (5.12)$$

This equation will be the core part of our MATLAB program.

As for the initial condition and boundary conditions, we assume that

$$\begin{aligned} u(x,y,0) &= \sin(\frac{\pi x}{L}) + \sin(\frac{\pi y}{H}), \\ u(0,0,t) &= 0, \\ u(L,H,t) &= 0; \end{aligned}$$

in addition, we set $L = H = 1$ and will be computing in the time interval $[0, 0.2]$.

The complete MATLAB source code is presented below. This program not only gives a huge matrix of values of u evaluated at each node but also provides an animation of the gradually decreasing temperature on different locations of the given region over time. We also present several screen shots from the animation in Figure 5.

```
% The Two-Dimensional Heat Equation Implementation with MATLAB
clear;
close all;
clc;

xL = 1;                %
```

```
yL = xL;            % spatial & time domain
tEnd = 0.2;         %

xNum = 50;              %    spacial:
yNum = xNum;            % number of nodes

dx = xL/(xNum-1);           %     spacial:
dy = yL/(yNum-1);           % distance b/w nodes

tNum = 100;                                  %
dt = tEnd/(tNum-1);                          %     time:
while dt≥(dx*dy)^2/(2*((dx)^2+(dy)^2))       % number of nodes
    tNum = tNum+10;                          % distance b/w nodes
    dt = tEnd/(tNum-1);                      % (satisfying stability)
end                                          %

a = dt/(dx)^2;        %
b = dt/(dy)^2;        % coefficients
c = 1 - 2*a - 2*b;    %

x = linspace(0,xL,xNum)';
y = linspace(0,yL,yNum);
t = linspace(0,tEnd,tNum);
U = zeros(xNum,yNum,tNum);

U(:,:,1) = sin(pi*x/xL)*sin(pi*y/yL);   % initial condition

%DISCRETE APPROXIMATION
for m=1:tNum-1
    for i = 2:xNum-1
        for j = 2:yNum-1
            U(i,j,m+1) = a*(U(i-1,j,m)+U(i+1,j,m)) ...
                + b*(U(i,j-1,m)+U(i,j+1,m)) ...
                + c*U(i,j,m);
        end
    end
end

% PLOTTING PART
for m=1:tNum
    [Y,X]=meshgrid(y,x);
    surf(X,Y,U(:,:,m));
    colormap(jet);
    shading interp;
    caxis([0,1]);
    colorbar;
    axis([0 1 0 1 0 1]);
    pause(0.001);       % pause to give a dynamic visualization
end
```

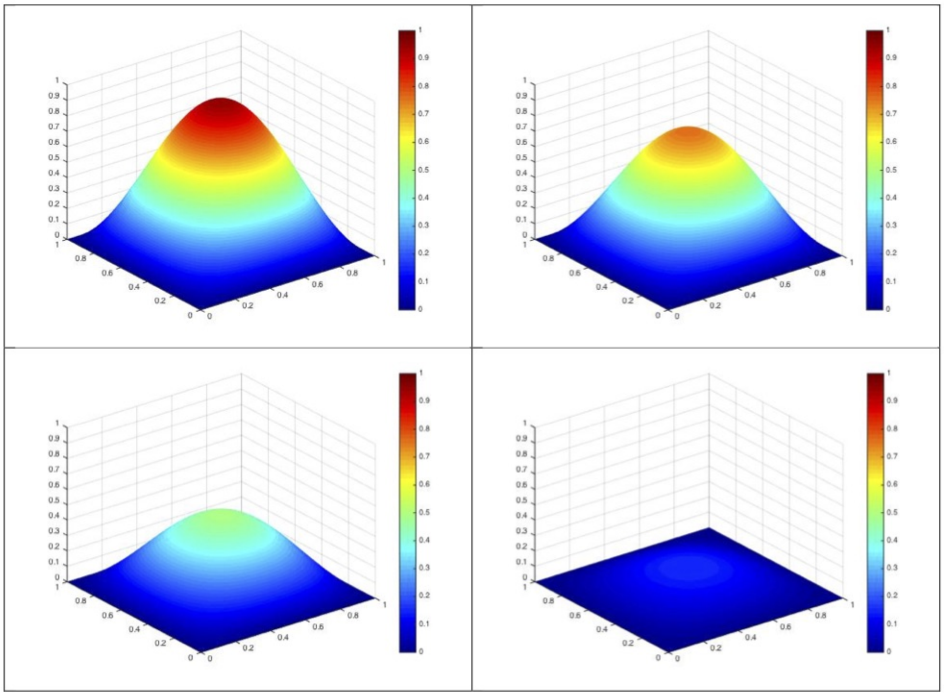


Figure 5: Heat distribution over time

5.4.4 MATLAB Simulation of the Heat Gradient Flow

With all these discussion about heat equations above, now we can start to set up the model for the heat gradient flow.

Suppose that we have a function $U : D_1 \to D_2$ such that

$$U(x, y, t) = \left(U^1(x, y, t), U^2(x, y, t)\right),$$

where D_1, D_2 are two regions on the plane, and (U^1, U^2) is the pair of coordinate functions of U. Same as the condition of the previous two-dimensional heat equation, here x and y are the spatial variables, while t is the time variable. What's more, in our simplified case, the domain D_1 is the unit square with vertices $(0,0),(1,0),(1,1),(0,1)$, and the image D_2 is the square with vertices $(0,0),(1,0),(1,2),(0,2)$. The model we are going to set up is supposed to give us a dynamic visualization of the change of

$$\frac{|\mathrm{d}U|^2}{\det(\mathrm{d}U)} \tag{5.13}$$

over time, under the condition that

$$\begin{cases} \partial_t U^1 = \Delta U^1, \\ \partial_t U^2 = \Delta U^2, \end{cases} \tag{5.14}$$

where the Laplacian is taken with respect to (x,y). What's more, the initial function at time $t=0$ is

$$\begin{cases} U^1(x,y,0) = x + \text{noise}, \\ U^2(x,y,0) = 2y + \text{noise}. \end{cases}$$

Remark. The noise we use in the MATLAB program has zero-value boundary entries and very small entries inside. The construction is listed in the source code from line 34 to line 47.

We have studied the two equations in (5.14) in the preceding part and know how to implement the numerical approximation in MATLAB. As for the term in (5.13) that we want to visualize, since

$$\mathrm{d}U = \begin{bmatrix} U^1_x & U^1_y \\ U^2_x & U^2_y \end{bmatrix},$$

we shall apply the idea of *difference quotient* again to each partial derivative and then combine them together into (5.13).

Let's consider another problem. Although we can use the discrete approximation from the previous implementation, the boundary condition in this example model calls for our attention, and we shall discuss it in detail.

In our example, $U : D_1 \to D_2$, we first parametrize ∂D_1 by $\gamma : [0,1] \to \partial D_1$, and then we want

$$U(\gamma(s),t) \in \partial D_2, \quad \forall s \in [0,1], \forall t \in [0,T].$$

That is, we want all boundary points mapped onto boundary points at any time node. Hence, if we denote by $\vec{n}(U(\gamma(s),t))$ the unit normal vector at point $U(\gamma(s),t) \in \partial D_2$, we have that, $\forall s, t$,

$$\begin{aligned} &(1) \quad \frac{\mathrm{d}}{\mathrm{d}s} U(\gamma(s), t) \perp \vec{n}(U(\gamma(s), t)), \\ &(2) \quad \frac{\mathrm{d}}{\mathrm{d}t} U(\gamma(s), t) \perp \vec{n}(U(\gamma(s), t)). \end{aligned}$$

Thus, the corresponding inner product is 0 in each case, and we obtain, $\forall s, t$,

$$\langle \mathrm{d}U \mid_{(\gamma(s),t)} \dot{\gamma}(s), \vec{n} \mid_{U(\gamma(s),t}\rangle = 0, \tag{5.15}$$

$$\langle \partial_t U \mid_{(\gamma(s),t)}, \vec{n} \mid_{U(\gamma(s),t}\rangle = 0. \tag{5.16}$$

In most cases, it takes some time to compute the unit normal vector, but since our D_1 and D_2 are rectangles with sides parallel to axes, there are in total only four normal vectors we are going to use:

$$\vec{n}_1 := (-1, 0), \vec{n}_2 := (0, -1), \vec{n}_3 := (1, 0), \vec{n}_4 := (0, 1).$$

In our model, we assume that horizontal sides are mapped to horizontal sides, and that vertical sides are mapped to vertical sides. Let's take the bottom side of D_1 (i,e, the line segment connecting $(0, 0)$ and $(0, 1)$) for example. On the lower horizontal side of the boundary, the second coordinate is constant, and therefore $\dot{\gamma}(s) = (1, 0)^T$. In addition, the normal vector at any point on the bottom side is $\vec{n}_2 = (0, -1)$. Then the equation (5.15) becomes

$$\langle \begin{bmatrix} \partial_x U^1 & \partial_y U^1 \\ \partial_x U^2 & \partial_y U^2 \end{bmatrix}_{(\gamma(s),t)} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \rangle = 0, \forall s, t.$$

So,

$$\langle \begin{bmatrix} \partial_x U^1 \\ \partial_y U^2 \end{bmatrix}_{(\gamma(s),t)}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \rangle = 0,$$

$$\partial_x U^2 \mid_{(\gamma(s),t)} = 0, \forall s, t,$$

which implies that U^2 is constant in x over the lower horizontal side. Further, we consider the equation (5.16), which now can be written as

$$\langle \frac{\partial}{\partial t} \begin{bmatrix} U^1(\gamma(s), t) \\ U^2(\gamma(s), t) \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \rangle = 0, \forall s, t.$$

Hence,

$$\frac{\partial}{\partial t} U^2(\gamma(s), t) = 0, \forall s, t,$$

from which we conclude that U^2 is also constant in time t over the lower horizontal side.

Likewise, we can discuss the cases of the rest three sides of D_1. In summary, we obtain the result that

- U^1 is constant in y and in t over vertical sides,
- U^2 is constant in x and in t over horizontal sides.

Note that our discrete approximation only produces the interior points but tells nothing about how boundary points changes over time. Now with the result we have got in the above paragraph, we can combine it with the discrete step and notations we used in the previous MATLAB program to extend the approximation to boundaries. In our case, we shall obtain

1. Over the left vertical side (i.e., $j = 1$),
   $U^1(i,1,m+1) = U^1(i,2,m+1)$ and $U^2(i,1,m+1) = U^2(i,1,m)$.
2. Over the lower horizontal side (i.e., $i = 1$),
   $U^1(1,j,m+1) = U^1(1,j,m)$ and $U^2(1,j,m+1) = U^2(2,j,m+1)$.
3. Over the right vertical side (i.e., $j = \mathrm{yNum}$),
   $U^1(i,\mathrm{yNum},m+1) = U^1(i,\mathrm{yNum}-1,m+1)$ and
   $U^2(i,\mathrm{yNum},m+1) = U^2(i,\mathrm{yNum},m)$.
4. Over the upper horizontal side (i.e., $i = \mathrm{xNum}$),
   $U^1(\mathrm{xNum},j,m+1) = U^1(\mathrm{xNum},j,m)$ and
   $U^2(\mathrm{xNum},j,m+1) = U^2(\mathrm{xNum}-1,j,m+1)$.

With the complete information of the discrete approximation of all points in the domain, we now can implement our model in MATLAB. The source code for this example model is presented as follows, and some screen shots from the resulting dynamic graph are listed in Figure 6.

```
% The Heat Flow Model Implementation with MATLAB
clear;
close all;
clc;

xL = 1;              %
```

```
yL = xL;            % spatial & time domain
tEnd = 0.12;        %

xNum = 20;                  %    spacial:
yNum = xNum;                % number of nodes

dx = xL/(xNum-1);           %     spacial:
dy = yL/(yNum-1);           % distance b/w nodes

tNum = 10;                                       %
dt = tEnd/(tNum-1);                              %     time:
while dt≥(dx*dy)^2/(2*((dx)^2+(dy)^2))           % number of nodes
    tNum = tNum+10;                              % distance b/w nodes
    dt = tEnd/(tNum-1);                          % (satisfying stability)
end                                              %

a = dt/(dx)^2;        %
b = dt/(dy)^2;        % coefficients
c = 1 - 2*a - 2*b;    %

x = linspace(0,xL,xNum)';
y = linspace(0,yL,yNum);
t = linspace(0,tEnd,tNum);

U1 = zeros(xNum,yNum,tNum);
U2 = zeros(xNum,yNum,tNum);

f = zeros(1,xNum);
g = zeros(1,yNum);
for i=2:xNum-1
    f(i) = exp(-1/(i-1)^2)*exp(-1/(xNum-i)^2);
end
for j=2:yNum-1
    g(j) = exp(-1/(j^2))*exp(-1/(yNum-j)^2);
end
N = randn(xNum,yNum)/1000;
for j=1:yNum                                         %
    for i=1:xNum                                     %
        N(i,j) = f(i)*g(j)*N(i,j);                   %   noise matrix
    end                                              %
end                                                  %

```

```
for j=1:yNum                                %
    for i=1:xNum                            %
        U1(i,j,1) = x(i) + N(i,j);          % initial condition with noise
        U2(i,j,1) = 3*y(j) + N(i,j);        %
    end                                     %
end                                         %



for m=1:tNum-1
    for j = 2:yNum-1
        for i = 2:xNum-1
            U1(i,j,m+1) = a*(U1(i-1,j,m)+U1(i+1,j,m)) ...
                + b*(U1(i,j-1,m)+U1(i,j+1,m)) ...
                + c*U1(i,j,m);
            U2(i,j,m+1) = a*(U2(i-1,j,m)+U2(i+1,j,m)) ...
                + b*(U2(i,j-1,m)+U2(i,j+1,m)) ...
                + c*U2(i,j,m);
        end
    end

    for i=1:xNum
        U1(i,1,m+1)=U1(i,2,m+1);
        U2(i,1,m+1)=U2(i,1,m);
        U1(i,yNum,m+1)=U1(i,yNum-1,m+1);
        U2(i,yNum,m+1)=U2(i,yNum,m);
    end

    for j=1:yNum
        U1(1,j,m+1)=U1(1,j,m);
        U2(1,j,m+1)=U2(2,j,m+1);
        U1(xNum,j,m+1)=U1(xNum,j,m);
        U2(xNum,j,m+1)=U2(xNum-1,j,m+1);
    end
end

% ENTRIES OF dU
for m=1:tNum
    for j = 1:yNum-1
        for i = 1:xNum-1
            U1x(i,j,m)=(U1(i+1,j,m)-U1(i,j,m))/dx;
            U2x(i,j,m)=(U2(i+1,j,m)-U2(i,j,m))/dx;
            U1y(i,j,m)=(U1(i,j+1,m)-U1(i,j,m))/dy;
            U2y(i,j,m)=(U2(i,j+1,m)-U2(i,j,m))/dy;
        end
    end
end

for m=1:tNum
    for j = 1:yNum-1
        for i = 1:xNum-1
            du=[U1x(i,j,m),U1y(i,j,m);U2x(i,j,m),U2y(i,j,m)];
            v(i,j,m)=(norm(du))^2/(det(du));   % that we want to visulize
        end
    end
end

% ANIMATION
for m=1:10
    [Y,X]=meshgrid(y(1:yNum-1),x(1:xNum-1));
    mesh(X,Y,v(:,:,m));
    colormap(jet);
    shading interp;
    caxis([0,1]);
    axis([0 1 0 1 0 5]);
    pause(0.1);
end
```

5.4.5 Example Tests

Here we test our MATLAB program via three examples.

In our tests, we want to measure the difference between the unit square with a rectangle, a parallelogram, and a shape obtained by moving the upper right vertex of a unit square from coordinates $(1,1)$ to $(2,2)$ (suppose the lower left vertex is at the origin). In our approach, we need to first construct an initial quasiconformal mapping between the unit square and each of the three shapes. Thus, we call the unit square as our reference domain. The rest three shapes are called given images.

These shapes can be found in figure 7.

For the case of the rectangle which has width 1 and length 3, the initial map we construct is a one-direction stretching with function expression $u(x,y) = (x,3y)$. From this initial map, we want to build the optimal map which is the minimizer of the

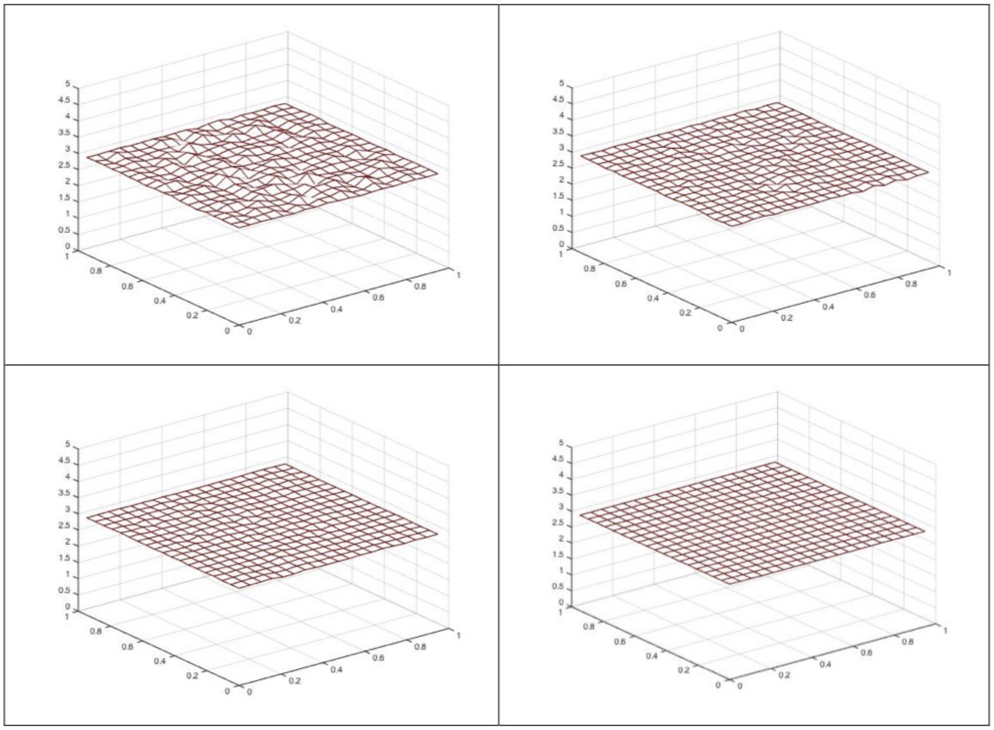


Figure 6: Heat Flow Model Visualization

mean dilation. Since the initial mapping u here is harmonic, and therefore satisfies the Euler-Lagrange equation, it is already a minimizer of mean dilation. We can also read this information from the corresponding graph which plots the change of distortions: It is a horizontal line, which means that the gradient flow does not produce any change. We read that the dilation of the optimal mapping is 2.96, which measures the distortion of the mapping, and as we mentioned before, we use it as the quantified difference between the unit square and the rectangle.

In the second example, we want to measure the difference between the unit square and the parallelogram in figure 7. In this example, the initial map we construct is $u(x, y) = (x, x + y)$, which is also harmonic, and thus satisfies the Euler-Lagrange equation. So, it is already a minimizer of mean dilation. Similar to the first example, we do not see dilation changes produced by the flow. And we take the dilation as the quantified difference between these two shapes.

The third example is a little bit different. In figure 7, we can see that most of the lower left part of the given image coincide with the unit square. In this example, the

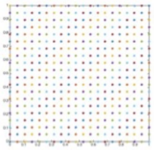
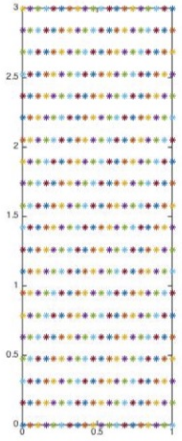
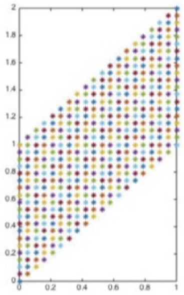
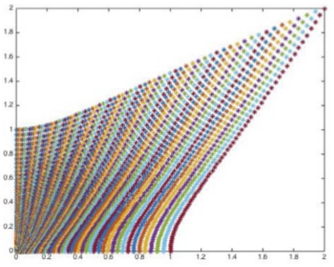
Reference Domain	Given Image	Quantified Difference
		2.96
		2.22
		1.46

Figure 7: Three Example Tests via MATLAB Program

initial function we construct is of expression $u(x,y) = \big((x^2+y^2)x, (x^2+y^2)y\big)$. It is not harmonic, and thus is not an optical one. So in this case the gradient flow decreases the dilation over time. When the dilation is stable, we read the value, which is 1.46, and we take it as the quantified difference.

5.5 Revisiting the L^p Gradient Flow: MATLAB Simulation

After investigating on the numerical approximation of the heat flow, we now introduce the implementation of the L^p flow. Recall that the initial value problem we are working on in this case is

$$\begin{cases} \partial_t u = -L_p u \quad \text{in} \quad \Omega_0 \times (0,T), \\ u = u_0, \qquad \text{at} \quad \Omega_0 \times \{t=0\}. \end{cases}$$

In the heat flow approach, the counterpart problem is

$$\begin{cases} \partial_t u = \Delta u \quad \text{in} \quad \Omega_0 \times (0,T), \\ u = u_0 \quad \text{at} \quad \Omega_0 \times \{t=0\}. \end{cases}$$

As we can see, the difference between the two systems above is the right hand side of the first equation. In later numerical approximations, we shall mainly focus on the core part but omit some side parts since they have been discussed in the heat flow model.

In the algorithm of this model, we continue use the difference quotients to approximate derivatives. The difference is that, here we use *first order central difference* instead of *first order forward difference* in order to further reduce the possible error.

Recall that the explicit form of $L_p u$ is

$$(L_p u)^i = 2p\partial_j \left([\text{trace}(g)]^{p-1}(du^{-1})^T S(g)\right)_{ij}, \quad i,j = 1,2,$$

where

$$g = \frac{\mathrm{d}u^T \mathrm{d}u}{|\det \mathrm{d}u|},$$

and

$$S(g) = g - \frac{\text{trace}(g)}{n}\text{Id}.$$

Suppose $u = (u^1, u^2)$. The Jacobian matrix of u

$$\mathrm{d}u = \begin{bmatrix} \partial_x u^1 & \partial_y u^1 \\ \partial_x u^2 & \partial_y u^2 \end{bmatrix}$$

will be approximated via replacing each entry with corresponding difference quotient. After that step, the corresponding approximations of trace(g), $(du^{-1})^T$, and $S(g)$ are all ready. Now we set

$$\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} := [\text{trace}(g)]^{p-1}(du^{-1})^T S(g).$$

Then we obtain the approximation of $L_p u$ as follows

$$(L_p u)^1 = 2p(\partial_x F_{11} + \partial_y F_{12}),$$

$$(L_p u)^2 = 2p(\partial_x F_{21} + \partial_y F_{22}).$$

The four derivatives in the above expression of $L_p u$ shall all be approximated by corresponding difference quotients. Then, with $\partial_t u$ approximated by difference quotient, we can complete the numerical implementation of

$$\begin{cases} \partial_t u = -L_p u & \text{in} \quad \Omega_0 \times (0, T), \\ u = u_0, & \text{at} \quad \Omega_0 \times \{t = 0\}. \end{cases}$$

in Matlab. The explicit core part can be found in the following source code from line 68 to line 107.

In addition to the difference of the core computation part between the two flows, another different part is about the boundary condition.

We first consider nodes at the four corners. We take one corner for example, which is shown in figure 8. We set the value of u at node $(1,1), (2,1)$ and $(1,2)$ be constant over time. For the node $(2,2)$, at each time step, the value of this node is the average of the value at node $(1,1)$ and $(3,3)$. The rest three corner conditions will follow the same pattern. The explicit information can be found in the source code from line 109 to line 157.

For the rest boundary nodes, we use the term, "the first layer", to refer to the outmost boundary. We set the value at the nodes of the first layer to be constant over time. For the second layer, we set the value at each time step to be equal to the average of the value of adjacent nodes from the first layer and the third layer at the immediate previous time step. That is, for example,

$$u(5, 2, t+1) = \frac{u(5, 1, t) + u(5, 3, t)}{2},$$

where t denoted the time step. The complete part can be found in the source code below from between line 159 and line 186.

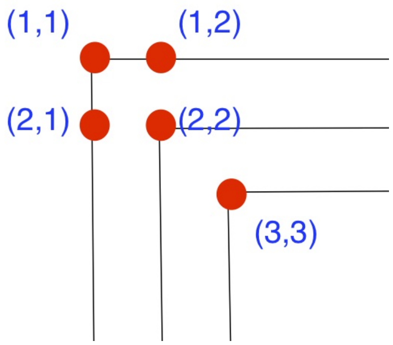


Figure 8: Corner Condition Example

Beside the computation part, we also have a change in the form of the visualization of the change of distortion. Instead of a plotting which is similar to figure 6, the visualized result in this case is a 2D plot with the value of distortion versus time. The result is in figure 9, and we can observe that the heat gradient flow does decrease the value of the distortion over time as desired.

```
% The Lp Flow Model Implementation with MATLAB
clear;
close all;
clc;

xL = 1;             %
yL = xL;            %   length
tEnd = 0.5;         %

p = 2;              %   value of p

xNum = 30;          %
yNum = xNum;        %   number of nodes
tNum = 50;          %

epsilon = exp(-100);

dx = xL/(xNum-1);           %
dy = yL/(yNum-1);           %   unit distance
```

```
dt = tEnd/(tNum-1);          %

x = linspace(0,xL,xNum)';          %
y = linspace(0,yL,yNum);           % vector of coords of nodes
t = linspace(0,tEnd,tNum);         %

f = zeros(1,xNum);
g = zeros(1,yNum);
for i=2:xNum-1
    f(i) = exp(-2/(dx*(i-1)^2)) * exp(-2/(dx*(xNum-i)^2));
end
for j=2:yNum-1
    g(j) = exp(-2/(dy*(j-1)^2)) * exp(-2/(dy*(yNum-j)^2));
end
N = randn(xNum,yNum)/10;
for j=1:yNum                                         %
    for i=1:xNum                                     %
        N(i,j) = f(i)*g(j)*N(i,j);                   %   noise matrix
    end                                              %
end                                                  %

% initial condition with noise
for j=1:yNum
    for i=1:xNum
        U1(i,j,1) = 100*x(i) + N(i,j);
        U2(i,j,1) = 100*(y(j)+x(i)) + N(i,j);
    end
end

U1x = zeros(xNum-1,yNum-1,tNum-1);
U1y = zeros(xNum-1,yNum-1,tNum-1);
U2x = zeros(xNum-1,yNum-1,tNum-1);
U2y = zeros(xNum-1,yNum-1,tNum-1);

v = zeros(xNum-1,yNum-1,tNum-1);

F11 = zeros(xNum-1,yNum-1,tNum-1);
F12 = zeros(xNum-1,yNum-1,tNum-1);
F21 = zeros(xNum-1,yNum-1,tNum-1);
F22 = zeros(xNum-1,yNum-1,tNum-1);

F11_x = zeros(xNum-2,yNum-2,tNum-1);
```

```
F12_y = zeros(xNum-2,yNum-2,tNum-1);
F21_x = zeros(xNum-2,yNum-2,tNum-1);
F22_y = zeros(xNum-2,yNum-2,tNum-1);

for m=1:tNum-1

    for j=2:yNum-1
        for i=2:xNum-1
            %  matrix consisting of difference quotients
            U1x(i,j,m) = (U1(i+1,j,m)-U1(i-1,j,m))/(2*dx);
            U1y(i,j,m) = (U1(i,j+1,m)-U1(i,j-1,m))/(2*dy);
            U2x(i,j,m) = (U2(i+1,j,m)-U2(i-1,j,m))/(2*dx);
            U2y(i,j,m) = (U2(i,j+1,m)-U2(i,j-1,m))/(2*dy);

            % Jacobian matric
            dU = [U1x(i,j,m),U1y(i,j,m); U2x(i,j,m),U2y(i,j,m)];

            dU_inv_T = transpose(inv(dU+ epsilon * eye(2)));    %
            g = (transpose(dU)*dU)/(abs(det(dU))+epsilon);      % for Lp(U)
            S_g = g - (trace(g)/2)*eye(2) ;                     %

            F = trace(g)^(p-1) * dU_inv_T * S_g;

            F11(i,j,m) = F(1,1);
            F12(i,j,m) = F(1,2);
            F21(i,j,m) = F(2,1);
            F22(i,j,m) = F(2,2);
        end
    end

    for j=3:yNum-2
        for i=3:xNum-2
            %  matrix consisting of difference quotients
            F11_x(i,j,m) = (F11(i+1,j,m)-F11(i-1,j,m))/(2*dx);
            F12_y(i,j,m) = (F12(i,j+1,m)-F12(i,j-1,m))/(2*dy);
            F21_x(i,j,m) = (F21(i+1,j,m)-F21(i-1,j,m))/(2*dx);
            F22_y(i,j,m) = (F22(i,j+1,m)-F22(i,j-1,m))/(2*dy);
        end
    end

    for j=3:yNum-2
        for i=3:xNum-2
```

```
            U1(i,j,m+1) = U1(i,j,m) - dt*(-2*p)*(F11_x(i,j,m)+F12_y(i,j,m));
            U2(i,j,m+1) = U2(i,j,m) - dt*(-2*p)*(F21_x(i,j,m)+F22_y(i,j,m));
        end
    end

    % Corner Conditions
    % NW corner of the grid
    U1(1,1,m+1)=U1(1,1,1);
    U1(1,2,m+1)=U1(1,2,1);
    U1(2,1,m+1)=U1(2,1,1);

    U2(1,1,m+1)=U2(1,1,1);
    U2(1,2,m+1)=U2(1,2,1);
    U2(2,1,m+1)=U2(2,1,1);

    U1(2,2,m+1)=(U1(3,3,m+1)+U1(1,1,m+1))/2;
    U2(2,2,m+1)=(U2(3,3,m+1)+U2(1,1,m+1))/2;

    %  NE corner of the grid
    U1(1,yNum,m+1)=U1(1,yNum,1);
    U1(2,yNum,m+1)=U1(2,yNum,1);
    U1(1,yNum-1,m+1)=U1(1,yNum-1,1);

    U2(1,yNum,m+1)=U2(1,yNum,1);
    U2(2,yNum,m+1)=U2(2,yNum,1);
    U2(1,yNum-1,m+1)=U2(1,yNum-1,1);

    U1(2,yNum-1,m+1)=(U1(3,yNum-2,m+1)+U1(1,yNum,m+1))/2;
    U2(2,yNum-1,m+1)=(U2(3,yNum-2,m+1)+U2(1,yNum,m+1))/2;

    %  SW corner of the grid
    U1(xNum,1,m+1)=U1(xNum,1,1);
    U1(xNum-1,1,m+1)=U1(xNum-1,1,1);
    U1(xNum,2,m+1)=U1(xNum,2,1);

    U2(xNum,1,m+1)=U2(xNum,1,1);
    U2(xNum-1,1,m+1)=U2(xNum-1,1,1);
    U2(xNum,2,m+1)=U2(xNum,2,1);

    U1(xNum-1,2,m+1)=  (U1(xNum,1,m+1)+U1(xNum-2,3,m+1))/2;
    U2(xNum-1,2,m+1)=  (U2(xNum,1,m+1)+U2(xNum-2,3,m+1))/2;


    %  SE corner of the grid
    U1(xNum,yNum,m+1)=U1(xNum,yNum,1);
    U1(xNum,yNum-1,m+1)=U1(xNum,yNum-1,1);
    U1(xNum-1,yNum,m+1)=U1(xNum-1,yNum,1);

    U2(xNum,yNum,m+1)=U2(xNum,yNum,1);
    U2(xNum,yNum-1,m+1)=U2(xNum,yNum-1,1);
    U2(xNum-1,yNum,m+1)=U2(xNum-1,yNum,1);

    U1(xNum-1,yNum-1,m+1)= (U1(xNum,yNum,m+1)+U1(xNum-2,yNum-2,m+1))/2;
    U2(xNum-1,yNum-1,m+1)= (U2(xNum,yNum,m+1)+U2(xNum-2,yNum-2,m+1))/2;

    % Boundary Conditions
    for i=3:xNum-2
        U1(i,1,m+1) = U1(i,1,1);
        U2(i,1,m+1) = U2(i,1,1);

        U1(i,2,m+1) = (U1(i,1,m)+U1(i,3,m))/2;
        U2(i,2,m+1) = (U2(i,1,m)+U2(i,3,m))/2;

        U1(i,yNum,m+1)=U1(i,yNum,1);
        U2(i,yNum,m+1)=U2(i,yNum,1);

        U1(i,yNum-1,m+1) = (U1(i,yNum,m)+U1(i,yNum-2,m))/2;
        U2(i,yNum-1,m+1) = (U2(i,yNum,m)+U2(i,yNum-2,m))/2;
    end

    for j=3:yNum-2
        U1(1,j,m+1) = U1(1,j,1);
        U2(1,j,m+1) = U2(1,j,1);

        U1(2,j,m+1) = (U1(1,j,m)+U1(3,j,m))/2;
        U2(2,j,m+1) = (U2(1,j,m)+U2(3,j,m))/2;

        U1(xNum,j,m+1)=U1(xNum,j,1);
        U2(xNum,j,m+1)=U2(xNum,j,1);

        U1(xNum-1,j,m+1) = (U1(xNum,j,m)+U1(xNum-2,j,m))/2;
        U2(xNum-1,j,m+1) = (U2(xNum,j,m)+U2(xNum-2,j,m))/2;
    end

    for j=2:yNum-1
        for i=2:xNum-1
            du=[U1x(i,j,m),U1y(i,j,m);U2x(i,j,m),U2y(i,j,m)];
            v(i,j,m)=(norm(du))^2/(det(du));     % distortion
        end
    end
end

% Change of the value of distortion over time
int_v = zeros(1,tNum-1);
for m=1:tNum-1
    for j=1:yNum-2
        for i=1:xNum-2
            int_v(m) = int_v(m) + abs(v(i,j,m))^(p)*dx*dy;
        end
    end
    int_v(m)=(int_v(m))^(1/p);
end
time = linspace(0,1,tNum-1);
figure
plot(time,int_v,'*');
```

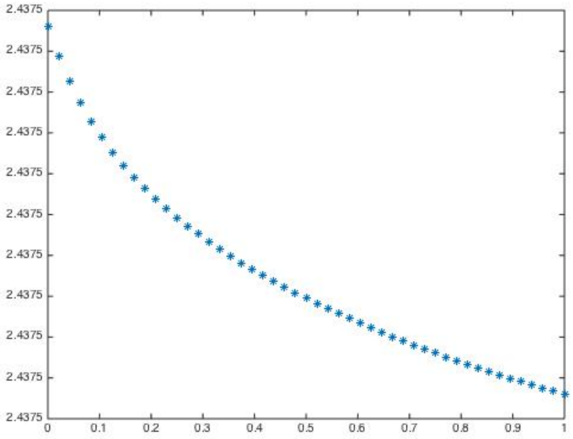


Figure 9: Change of Distortion over Time

6 Conclution

In conclusion, in order to quantify the difference between two given shapes on the plane with a finite number of prescribed matching points, among infinitely many quasiconformal maps relating the two shapes, we use gradient flows to obtain a homotopy of quasiconformal mappings, decreasing the mean distortion of the map with the flow, find a canonical minimizer for such a functional, and then use the distortion of the minimizer as the quantified difference between the two planar shapes.

The gradient flows that we use in our approach include the L^p flow and the heat flow. Both of them provide plausible solutions to the posed problem. As they are nonlinear and linear flows, respectively, the time that the numerical implementation takes to process are different.

Our work is inspired by [Jones and Mahadevan, 2013]. We come up with these alternative methods because we want to lower the order of derivatives to reduce the error from the numerical approximations. Our idea is plausible, but since there is no specific example of implementation results in [Jones and Mahadevan, 2013], we are not able to make a direct comparison to check if our method creates better accuracy or not.

The core numerical method we use in our implementation is that we use difference quotients to approximate derivatives. Actually, the finite-element method, which is also used in [Jones and Mahadevan, 2013], is supposed to give a higher accuracy and a better result. This project can be extended via using this method.

Additionally, another possible improvement that can be made is that, in our approach, we have to construct a quasiconformal mapping at the first step as the initial mapping to start with, which is not easy in general cases. So, an improved approach which can give the quantified difference between two given shapes without a prerequisite initial mapping will directly provide the answer and help save the time of the whole process of implementation.

References

[Ahlfors, 1953] Ahlfors, L. V. (1953). *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable.* McGraw-Hill, 2$^{\text{nd}}$ edition

[Ahlfors and Earle, 1966] Ahlfors, L. V. and Earle, C. J. (1966). *Lectures on Quasiconformal Mappings*, volume 3. D. Van Nostrand, Princeton, New Jersey.

[Capogna, 2014] Capogna, L. (2014). L^{∞}-Extremal Mappings in AMLE and Teichmüller Theory. In *Fully Nonlinear PDEs in Real and Complex Geometry and Optics*, pages 1–46. Springer.

[Capogna and Raich, 2012] Capogna, L. and Raich, A. (2012). An Aronsson Type Approach to Extremal Quasiconformal Mappings. *Journal of Differential Equations*, 253(3):851–877.

[Garabedian, 1970] Garabedian, P. R. (1970). A Method of Canonical Coordinates for Flow Computations. *Communications on Pure and Applied Mathematics*, 23(3):313–327.

[Gill et al., 2005] Gill, P. E., Murray, W., and Saunders, M. A. (2005). Snopt: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review*, 47(1):99–131.

[Jeffrey, 2005] Jeffrey, A. (2005). *Complex Analysis and Applications*, volume 10. CRC Press, 2$^{\text{nd}}$ edition.

[Jones and Mahadevan, 2013] Jones, G. W. and Mahadevan, L. (2013). Planar Morphometry, Shear and Optimal Quasi-Conformal Mappings. In *Proc. R. Soc. A*. The Royal Society.

[Persson and Strang, 2004] Persson, P.-O. and Strang, G. (2004). A Simple Mesh Generator in MATLAB. *SIAM Review*, 46(2):329–345.

[Recktenwald, 2004] Recktenwald, G. W. (2004). Finite-Difference Approximations to the Heat Equation. *Class Notes*.

[Thompson, 1942] Thompson, D. W. (1942). *On Growth and Form.* The University Press, Cambridge [Eng.].

[Vuorinen, 1988] Vuorinen, M. (1988). *Conformal Geometry and Quasiregular Mappings*, volume 1319 of *Lecture Notes in Math.* Springer Berlin Heidelberg.