OPTIMAL MASS TRANSPORT FOR
ADAPTIVITY AND IMAGE REGISTRATION

by

Mohamed H. Mahmoud Sulman
B.Sc., Al-Neelain University, 1993
M.Sc., Tsinghua University, 2000
M.Sc., Simon Fraser University, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the Department
of
Mathematics

© Mohamed H. Mahmoud Sulman 2008
SIMON FRASER UNIVERSITY
Summer 2008

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
APPROVAL

Name: Mohamed H. Mahmoud Sulman
Degree: Doctor of Philosophy
Title of thesis: Optimal Mass Transport for Adaptivity and Image Registration

Examinining Committee: Dr. Manfred Trummer
Chair

Dr. Robert D. Russell, Senior Supervisor

Dr. JF Williams, Senior Supervisor

Dr. Mirza Faisal Beg, Supervisor

Dr. Steven J. Ruuth, SFU Examiner

Dr. Raymond J. Spiteri, External Examiner

Date Approved: August 6, 2008
Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Revised: Fall 2007
Abstract

Finding a coordinate transformation is a fundamental task in diverse fields of mathematical applications. In this thesis, we propose new techniques of computing coordinate transformations for two different applications. In the first application, the transformations are computed to generate adaptive meshes that are suitable for solving time-dependent partial differential equations in two or more spatial dimensions. For the second application, the transformations are developed for an elastic image registration. Registration means establishing a coordinate transformation between two or more images taken, for example, at different times or from different viewpoints.

The transformations are formulated based on solving the optimal mass transport problem, also known as the Monge–Kantorovich problem. This problem concerns finding the best way of moving a pile of material from one site to another with minimum transportation cost. Two different methods are described to compute the transformations for each application. In the first method, the solution of the problem is obtained as the steady-state solution of a parabolic partial differential equation. The second method is a velocity based method in which a velocity field is obtained using the fluid dynamics formulation of the $L^2$ Monge–Kantorovich problem.

The optimal mass transport approach for computing coordinate transformations has a number of useful features. The existence and uniqueness of the transformation are guaranteed from the Kantorovich theory. Moreover, it can be characterized as the gradient of a convex function, i.e., it is rotation free. A number of theoretical issues for computing these transformations are addressed. Several numerical experiments are presented to show the performance of the proposed approach for both adaptive grid generation and image registration for medical applications.
Acknowledgments

I owe a lot to my supervisor Dr. Robert D. Russell, for his time, encouragement and constant support during the entire course of my study at Simon Fraser university. His wise guidance is gratefully acknowledged.

My enormous gratitude is given to my cosupervisor Dr. JF Williams, for his overwhelming generosity with time, suggestions and discussions during the stage of writing the thesis. He was kind enough to provide the matlab code for the PMA method.

I sincerely thank Dr. M. Faisal Beg for providing the data sets on the 2D and 3D medical images appear in this thesis. I deeply acknowledge his valuable discussions and suggestions on the material in Chapter 5.

Special thanks go to Dr. Wilfrid Gangbo, for the valuable discussions and useful suggestions on the material of Section 3.1.3 during the optimal mass transport conference at IPAM, UCLA. I thank IPAM for the partial financial support during my stay at UCLA. Also special thanks go to Dr. Weiming Cao, for making the code for the GCL method available. I also thank Dr. Jim Verner, for proofreading and making some comments on various parts of this thesis.

My sincere gratitude goes to the staff and faculty of the Mathematics Department at Simon Fraser University, for their kind support throughout my studies. I also thank Xiangmin Xu and Colin Macdonald and my colleagues in the PIMS computer lab for their help, friendship and great company during this journey of my study at Simon Fraser University.

Last but not least, I acknowledge the great support of my parents and wife Maalim Eltigani. Without her, the accomplishment of this work would not be possible.
Contents

Approval ii
Abstract iii
Acknowledgments iv
Contents v
List of Tables viii
List of Figures ix
Glossary xii

1 Introduction 1
  1.1 Literature Review ............................................. 1
  1.2 Contributions of The Thesis .................................. 4
      1.2.1 Outline of the Thesis ................................. 7

2 Background and Motivation 8
  2.1 One Dimensional Adaptation ................................. 8
      2.1.1 The Equidistribution Principle .......................... 10
      2.1.2 Adaptation Function ................................... 11
  2.2 Higher Dimensional Adaptation ............................. 15
      2.2.1 The GCL Adaptive Mesh Method ....................... 18
2.2.2 The PMA Adaptive Mesh Method ........................................... 21
2.3 The Monge–Kantorovich Problem ........................................... 23
2.4 The Fluid Dynamics Framework of the MKP ................................. 25
2.5 Image Registration .............................................................. 27
  2.5.1 AHT Image Registration Method ....................................... 29
  2.5.2 LDDMM Image Registration Method ................................... 30

3 The MKP Adaptive Mesh Methods .............................................. 32
  3.1 The PMKP Adaptive Mesh Method ......................................... 32
    3.1.1 Boundary and Initial conditions ................................... 35
    3.1.2 The Stopping Criterion .............................................. 35
    3.1.3 Existence and Uniqueness Results .................................. 36
    3.1.4 Implementation of the PMKP Method ................................ 41
  3.2 The FDMKP Adaptive Mesh Method ....................................... 44
    3.2.1 Implementation of FDMKP Method .................................. 44
    3.2.2 The Saddle Point Problem Formulation ............................. 45
    3.2.3 The Saddle Point Problem Solution Procedure ..................... 46

3.3 Solution Procedure for the Physical Problem ............................. 49
  3.3.1 Finite Difference Scheme ............................................. 50

4 Numerical Experiments .......................................................... 55

5 The MKP Elastic Registration Methods ...................................... 90
  5.1 MKP Registration Framework ............................................. 90
  5.2 The PMKP Registration Method ......................................... 93
    5.2.1 Computational Details ............................................ 93
  5.3 FDMKP Image Registration Method ..................................... 94
  5.4 Numerical Experiments .................................................... 96
    5.4.1 Numerical Examples ............................................... 96
    5.4.2 Summary of the Numerical Results ................................ 110

6 Concluding Remarks and Future Directions ................................ 118
List of Tables

4.1 Example 4.1. The maximum and minimum values of the mesh quality measure function $E(x, y, t)$ ................................................. 59
4.2 Example 4.4. The maximum and minimum values of the mesh quality measure $E(x, y, t)$ for the adaptation function (4.5) ........... 66
4.3 Example 4.4. The maximum and minimum values of the mesh quality measure $E(x, y, t)$ with respect to the adaptation function (4.6) ... 66
4.4 Example 4.6. The change of the $L^2$ norm of the error with $\delta x$ .... 86
4.5 Example 4.6. The change of the $L^2$ norm of the error with the number of the grid points given at some selected times. ................. 86

5.1 Typical average run-time results for some 3D data sets for PMKP....... 116
5.2 Overlay error, relative error, and $L^2$-Kantorovich distance for some sample 3D datasets. .................................................. 116
5.3 Left Hippocampus Segmentation Results ..................................... 116
5.4 Right Hippocampus Segmentation Results ................................... 117
List of Figures

2.1 Equidistributed mesh using the gradient adaptation function 13
2.2 Equidistributed mesh using the arc-length adaptation function 14
2.3 Representation of mesh adaptation as a coordinate transformation $x = x(\xi, t)$ from $\Omega_c$ to $\Omega$ for the 2D case. 15
2.4 Transformation of two squares, $\phi$ transforms a template image $\rho_0$ (left) to a target image $\rho_1$ (right). 28
2.5 Three images of a human brain given as a template image $\rho_0$ (left), a reference (target) image $\rho_1$ (middle) and a deformed template (right). The image was extracted from the web in the year 2007 during the development of the thesis. 29

3.1 Sketch of a smooth strictly convex domain $\Omega_c$ and a square domain $\Omega$ for the 2D case. 39
3.2 A nine point stencil used in approximating the second order derivatives. 42

4.1 Example 4.1. PMKP method, the plots of the adaptive meshes and the equidistribution measure function $E_{adp}(x, y, t)$ 57
4.2 Example 4.1. FDMKP method, the plots of the adaptive meshes and the equidistribution measure function $E_{adp}(x, y, t)$ 58
4.3 The plot of the physical solution (4.3) 60
4.4 Example 4.2. Adaptive meshes and the function $E_{adp}(x, y)$. 61
4.5 The loglog plot $L^2$ norm of the residual 62
4.6 Example 4.3. The PMKP method, plot of adaptive meshes of size $41 \times 41$ and the function $E_{adp}(x, y)$. 63
4.7 Example 4.3. The FOMKP method, plot of adaptive meshes of size $41 \times 41$ and the function $E_{adp}(x, y)$. 64

4.8 Example 4.4. The PMKP method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.5). 67

4.9 Example 4.4. The FDMKP method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.5). 68

4.10 Example 4.4. The PMA method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.5). 69

4.11 Example 4.4. The GCL method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.5). 70

4.12 Example 4.4. The PMKP method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.6). 71

4.13 Example 4.4. The FDMKP method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.6). 72

4.14 Example 4.4. The PMA method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.6). 73

4.15 Example 4.4. The GCL method, adaptive meshes and the function $E_{adp}(x, y, t)$ obtained for the adaptation function (4.6). 74

4.16 Example 4.5. Adaptive mesh generated for the solution $u = ce^{-c^2(x^2+y^2)}, c = 100$, using the adaptation function (4.9). 76

4.17 Example 4.5. $u = ce^{-c^2(x^2+y^2)}, c = 100$ plots in the physical and computational domains using the adaptation function (4.9). 77

4.18 Example 4.5. Adaptive mesh generated for the solution $u = ce^{-c^2(x^2+y^2)}, c = 100$, using the adaptation function (4.8). 78

4.19 Example 4.5. $u = ce^{-c^2(x^2+y^2)}, c = 100$ plots in the physical and computational domains using the adaptation function (4.8). 79

4.20 Example 4.6. The numerical solution of the 2D Burgers' equation and the adaptive mesh computed using the adaptation function (4.8). 83

4.21 Example 4.6. The numerical solution of the 2D Burgers' equation and the adaptive meshes, computed using the adaptation function (4.9). 84
Example 4.6. The $L^2$ norm of the errors in the numerical solution of the 2D Burgers' equation.

Example 4.6. Plot of the $L^2$ norm of the error as a function of the number of the grid points and the maximum and minimum of the grid spacing.

Example 4.6. The plot of the $L^2$ norm of the error for the PMKP and PMA methods.

Example 4.7. 3D adaptive meshes

Example 4.8. 3D adaptive mesh

Example 5.1. Registration of two concentric circles

Example 5.1. Displacement vector field, numerical and exact solution for the optimal mapping

Example 5.1. Registration of a circle to a C-shape

Example 5.1. Deformations of a circle to a C-shape

Example 5.1. Registration of two Macaque cortex images

Example 5.1. Deformations of a Macaque cortex

Example 5.1. Registration of two 3D hearts of size 240 x 240 x 80

Example 5.1. 3D registration of two heart images

Example 5.1. 3D registration of a heart to six other hearts

Example 5.1. 3D registration of a left hippocampus

Example 5.1. 3D registration of a right hippocampus

Example 5.1. Segmentation of three coronal slices of left hippocampus

Example 5.1. Segmentation of three coronal slices of right hippocampus
Glossary

\( D^2 \)  
Hessian matrix \( D^2 f = (\partial_{x_i x_j} f)_{i,j} \subset \mathbb{R}^{d \times d} \).

\( L^2 \) \text{MKP}  
the \( L^2 \) Monge–Kantorovich problem.

\( L^p(\Omega) \)  
\( \{ f : \Omega \to \mathbb{R} \mid (\int_\Omega |f|^p dx)^{\frac{1}{p}} < \infty, 1 \leq p < \infty \} \).

\( L^p_{\text{loc}}(\Omega) \)  
\( \{ f : \Omega \to \mathbb{R} \mid f \in L^p(\Lambda), \text{for each } \Lambda \subset \Omega \} \).

\( W^{1,p}_{\text{loc}}(\Omega) \)  
Sobolev space \( \{ f \mid f \text{ and its weak partial derivatives } \partial_{x_i} f \in L^p_{\text{loc}}(\Omega), i = 1, \cdots d \} \).

\( \Omega \)  
physical domain, \( \Omega \subset \mathbb{R}^d \).

\( \Omega_c \)  
computational domain, \( \Omega \subset \mathbb{R}^d \).

\( \mathbb{R} \)  
real numbers.

\( \mathbb{R}^d \)  
d-dimensional real Euclidean space.

\( \mathbb{R}^{d \times d} \)  
d \times d real matrices.

\( \nabla \)  
\( \nabla f = (\partial_{x_1} f, \cdots, \partial_{x_d} f)^T \) for \( f : \mathbb{R}^d \to \mathbb{R} \), \( \nabla \phi = (\partial_{x_i} \phi_i)_{i,j} \in \mathbb{R}^{d \times d} \) for \( \phi : \mathbb{R}^d \to \mathbb{R}^d \).

\( \partial \Omega \)  
boundary of \( \Omega \).

\( \partial_{x_i} \)  
partial derivative with respect to \( x_i \).

\( \Delta \)  
Laplace operator, \( \Delta f = \sum_{i=1}^d \partial_{x_i x_i} f \).

\text{GCL}  
goemetric conservation law.

\text{MAE}  
Monge-Ampère equation.

\text{MKP}  
Monge–Kantorovich problem.

\text{MRI}  
magnetic resonance imaging.
Chapter 1

Introduction

1.1 Literature Review

The use of adaptive methods is fundamental to the numerical solution of differential equations that involve large solution variations. Using uniform meshes to accurately resolve the physical solution in these regions of rapid solution variation requires an extremely fine mesh which makes the computations prohibitively expensive and inefficient in particular for two or higher dimensional problems. During the last two decades a broad range of adaptive mesh methods have been developed for various types of problems arising in many areas such as computational fluid dynamics (CFD), combustion processes, and heat transfer. Adaptive methods seek to provide efficient numerical solutions with high accuracy. There are three major classes of adaptive methods. The first one, is the $p$-method. Where, the polynomial degree of the shape function is increased or decreased according to the smoothness of the numerical solution of the physical problem. For instance, a cubic shape function can be used instead of a linear one. The second class, is the $h$-method which involves local refinement or coarsening of the spatial mesh by adding or removing points from the mesh based on a posterior error estimators or indicators to achieve certain accuracy in the numerical solution of the physical problem. The $h$-method is generally easy and straightforward to apply. Several local refinement methods have been developed, for example see [AFMW92, BS90]. Disadvantages of the local refinement method are
its complex data structure and its being forced to either use one-step time integrator or continually restart a multistep method. The third class, is the r-method which is also known as the moving mesh method. For this method, the number of mesh points is fixed and the nodes are continuously moving in time in such a way that sufficient mesh points remain concentrated in regions where the variation of the solution is large. The mesh concentration is controlled by some measure of the solution variation or error indicator called the adaptation function (or the monitor function). In this thesis, we are interested in applications of the third class of these adaptive methods.

For standard moving mesh methods, the adaptive mesh is constructed by computing a continuous time-dependent coordinate transformation, \( x = x(\xi, t) \), defined from a computational domain \( \Omega_c \) to a physical domain \( \Omega \).

Roughly speaking, in one spatial dimension, the transformation \( x = x(\xi, t) \) is determined by equally distributing the adaptation (monitor) function \( \rho(x, t) \) in each subinterval. Based on this equidistribution principle, Russell and his coworkers introduced the moving mesh partial differential equation (MMPDE) method (see [HRR94a], [HRR94b], [HR97], [BHR96], and [HR96]). In one spatial dimension, the MMPDE approach has proven to be an efficient and reliable approach for mesh adaptation. However, in higher dimensions, satisfying the equidistribution principle alone is not sufficient to uniquely determine the coordinate transformation, since it is one equation in more than one unknown. Thus, the coordinate transformation is required to satisfy some additional conditions, and this has lead to a broad range of moving mesh strategies for higher dimensional problems.

Motivated by the original MMPDE approach, Huang and Russell [HR99] proposed a moving mesh strategy based on a gradient flow equation for two dimensional problems. Cao, Huang and Russell in [CHR02] described a two-dimensional moving method based on the geometric conservation law (GCL). The GCL was first introduced by Thomas and Lombard [TL79] to maintain global and local conservation of (cell) volume elements under time-dependent mappings that result from boundary motion. For this method, the Jacobian of the transformation \( x = x(\xi, t) \) is first specified by relating it to the adaptation function. Then the mesh movement is controlled
by a velocity field which is uniquely determined by combining an additional condition imposed on its curl and the GCL. In [BW06] Budd and Williams have derived a moving mesh method based on solving a parabolic Monge-Ampère equation (PMA). For the PMA method, the adaptive mesh is generated as the image of the coordinate transformation \( \mathbf{x} = \mathbf{x}(\xi, t) \). The transformation, unlike the GCL method, is directly computed as the gradient of a smooth scalar potential function. Several other adaptive mesh methods have been developed for two dimensional problems, for example see [BMRS02, HS94, CH01] and the references therein.

Coordinate transformation techniques play a major role in adaptive grid generation. Another area of applications where these techniques are equally important is in image registration, also known as image matching or mapping. It is one of the most important tasks in image processing, where one is interested in comparing information given by different images. The image registration problem appears in many different application areas, for example computer vision, geophysics, and medical imaging. An overview of image registration can be found in [Bro92, MV98, MF93, PMV03], and the references therein.

Registration is the process of aligning two or more images. The alignment is often achieved by finding a coordinate transformation that best matches the two images in some specified sense. For example, consider the case where the two images are taken of a patient using two different sensors. Suppose that the first, a CT scan (computed tomography) can clearly see the structures of the patient, i.e., the bones and the gross anatomy, whereas the second scan uses a sensor which is sensitive to radio nucleic activity such as PET (positron emission tomography) which can clearly localize specific metabolic activity but only indirectly determine a limited number of normal structures. Since the two images may be taken at different resolutions, from different perspectives, and at different times, it is not a simple task to overlay the two images directly. However, successful registration can identify the structural sites of metabolic activity (such as tumors) to overlay the images algorithmically. In this case, registration involves finding a transformation which matches the structures found by both sensors, thus allowing the clinician to overlay different physiological aspects of the same physical region.
In general, image registration algorithms can be classified based on the coordinate transformation used in matching the images. The first class is the rigid registration where only rotations and translations of the coordinates are allowed (for example, see [DSR+99, MCV+97, VdEMPV95]). The second class is nonrigid or elastic registration; for this class, the transformations allow local warping of image features, thus providing support for local deformations. There are a number of different nonrigid transformation approaches in the literature. These approaches, include Landmark based methods (for example, see [Thi96, JM00, CRD+99, Boo99, BMTY03, VG05]), methods based on modeling the transformations as deformations of physical bodies (solids, fluid), and large deformation diffeomorphism based methods (for example, see [Bro81, SLP+96, SD03, RSH+99, LHG+05, HBHH01, GRB93, JM00, DSR+99, CNPE94, CJM97]). In this thesis we are concerned with the class of elastic registration methods. In [AHT03] an elastic registration method is described based on ideas from the theory of the optimal mass transport problem. This problem was first proposed by Monge [Mon81] in 1781, and it describes the most efficient way of moving a pile of soil from one configuration to another one with minimal transportation cost. A modern mathematical formulation of the Monge problem was described by Kantorovich in 1942 (cf.[Kan48] for the English version) defining the Monge-Kantorovich problem (MKP). The MKP has recently become a central focus of many research areas and has been extensively studied by many authors leading to the so-called Monge-Kantorovich theory. This problem has numerous applications in diverse fields of science, for example, fluid dynamics, functional analysis, probability theory, image analysis, economics and meteorology. For more recent study on this problem see, for example, [Sud79], [Amb03], [ACB+03] [CFM02], [FM02], [GM95], [GS98] and [EG99].

1.2 Contributions of The Thesis

MKP Methods for 2D and 3D Adaptive Grid Generation

Consider the problem of generating an adaptive mesh on a physical domain $\Omega \subset \mathbb{R}^d, d = 2$ or 3. Assume that the adaptation function $\rho(x, t), x \in \Omega$, is given for all time.
CHAPTER 1. INTRODUCTION

5

t ∈ [0, 1]. The adaptive mesh at a fixed but arbitrary time t is obtained as an image of a coordinate transformation \( x : \Omega_c \rightarrow \Omega, \ x = x(\xi, t) \), where \( \Omega_c \subset \mathbb{R}^d, d = 2 \) or 3, is the computational domain.

We propose two different approaches to determine the transformation \( x = x(\xi, t) \) based on solving the \( L^2 \) Monge–Kantorovich problem. More precisely, we specify two positive density functions \( \rho_0 \) and \( \rho_1 \) defined on \( \Omega_c \) and \( \Omega \) and find the transformation \( x = x(\xi, t) \) as the minimizer of the \( L^2 \) Kantorovich distance defined as

\[
\inf \left( \int_{\Omega_c} |x(\xi) - \xi|^2 \rho_0(\xi) d\xi \right),
\]

among all transformations \( x(\xi, t) \) that realizes the transfer of \( \rho_0 \) to \( \rho_1 \).

Our first method for determining \( x = x(\xi, t) \) is based on finding the steady-state solution of the parabolic Monge–Ampère equation

\[
\frac{\partial \Psi}{\partial \tau} = \log \left( \frac{\rho_1(\nabla \Psi) \det D^2 \Psi}{\rho_0(\xi)} \right),
\]

where \( \Psi \) is a scalar potential function, \( \nabla \Psi \) is the spatial gradient of \( \Psi \) with respect to the variable \( \xi \) and \( D^2 \Psi \) is the Hessian matrix of \( \Psi \). In this case, the transformation is obtained by taking the spatial gradient of \( \Psi \) with respect to the variable \( \xi \).

A number of theoretical issues, such as the existence, uniqueness, convergence and stopping criterion of the solution of the nonlinear partial differential equation (1.2), are investigated in this thesis.

Our second method for determining the coordinate transformation is a velocity based method. The velocity field is found by solving the \( L^2 \) Monge–Kantorovich problem in a fluid dynamics framework. In this method, we solve the minimization problem,

\[
\inf_{\mathbf{v}, \rho} \int_0^1 \int_\Omega \rho(\mathbf{x}, \tau) |\mathbf{v}(\mathbf{x}, \tau)|^2 \ d\mathbf{x} d\tau, \tag{1.3}
\]

over all the velocity and density fields \( \mathbf{v}(\mathbf{x}, \tau) \) and \( \rho(\mathbf{x}, \tau), \tau \in [0, 1] \) that satisfy the continuity equation

\[
\frac{\partial \rho}{\partial \tau} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{1.4}
\]

and the initial and final conditions

\[
\rho(\xi, 0) = \rho_0(\xi), \quad \rho(\mathbf{x}, 1) = \rho_1(\mathbf{x}). \tag{1.5}
\]
CHAPTER 1. INTRODUCTION

The coordinate transformation \( x = x(\xi, t) \) is obtained by time integration of the velocity field.

When using the optimal mass transport approach to compute the coordinate transformation, some important features such as existence and uniqueness of the transformation and no mesh crossing automatically hold from the Monge–Kantorovich theory. The proposed MKP adaptive methods can be considered as \( h - \tau \) method since it uses a fixed number of mesh points and continuously regridding the mesh.

We present a number of numerical experiments to show the effectiveness of the MKP methods for generating adaptive meshes. We apply these methods for both 2D and 3D adaptive grid generation. We also describe in detail a numerical solution of a two-dimensional physical problem. The results of the MKP methods for adaptive grid generation are to appear in [SWR08].

MKP Methods for Elastic Image Registration

Images are represented as functions \( \rho : \Omega \to \mathbb{R}^d \), \( d = 1 \) and \( d = 3 \) for gray-scale and colored (RGB) images respectively, \( \Omega \) is a bounded domain in \( \mathbb{R}^d \), \( d = 2 \) for 2D and \( d = 3 \) for 3D.

Assume that we are given two gray-scale images \( \rho_0 \) and \( \rho_1 \) defined on \( \Omega_0 \) and \( \Omega_1 \), respectively, subdomains of \( \mathbb{R}^d \). To register the two images \( \rho_0 \) and \( \rho_1 \), one needs to find a coordinate transformation \( \phi : \Omega_0 \to \Omega_1 \) that deforms the image \( \rho_0 \) so that the resulting image becomes similar to \( \rho_1 \).

We employ the MKP methods, described above for adaptive grid generation, to compute the mapping \( \phi \) that registers the two images. The \( L^2 \) Kantorovich distance is a quantitative information measure for comparing two different density functions (intensity functions for the case of images). This makes the optimal mass transport approach efficient and attractive for computing a coordinate transformation to perform image registration.

The proposed methods of registration are tested for several sample 2D images as well as for real 2D and 3D medical images. The experimental results show that the new methods are both effective and fast.
CHAPTER 1. INTRODUCTION

The results of using the MKP approach for elastic image registration will also appear in [SWRB08].

1.2.1 Outline of the Thesis

The thesis is organized as follows. In Chapter 2 we give an overview of one and two dimensional adaptive mesh methods. Chapter 2 also includes some background on the Monge–Kantorovich problem and the image registration problem. Chapter 3 presents the MKP methods for computing coordinate transformations for generating adaptive meshes suitable for solving time-dependent partial differential equations in higher dimensions. In Chapter 4 we provide several numerical experiments for 2D and 3D adaptive grid generation using the MKP methods. A detailed description of the MKP methods to compute mappings for image registration is given in Chapter 5. To show the performance of these methods, a number of experiments for 2D as well as 3D image registrations are presented in Chapter 5. Finally, we give discussion and some future directions for the MKP methods for both adaptive grid generation and elastic image registration in Chapter 6.
Chapter 2

Background and Motivation

In this chapter, we give an overview of adaptive mesh methods, the Monge-Kantorovich problem (MKP), and the image registration problem. In particular, in Section 2.1 we introduce some key ingredients of the adaptive mesh methods in the one dimensional case. In Section 2.2 we discuss some basic concepts of adaptation in higher spatial dimensions. To facilitate understanding of these basic concepts, we review two related methods of adaptive grid generation in two spatial dimensions, namely, the PMA method described in [BW06] and the GCL method of [CHR02]. A summary of the Monge-Kantorovich problem and some theoretical results on the coordinate transformation arising in the theory of solving the MKP are included in Section 2.3 and Section 2.4. We conclude this chapter by giving a brief introduction on the image registration problem in Section 2.5.

2.1 One Dimensional Adaptation

When using the standard method of lines (MOL) to solve a time-dependent partial differential equation (PDE) with suitable boundary and initial conditions, one first discretizes the PDE in space on a uniform mesh. Thus, a system of ordinary differential equations (ODEs) in time is obtained for the solution values at the spatial grid nodes. Then, a numerical solution of the system is computed using an explicit or implicit scheme for the temporal discretization. One can use the same uniform
mesh at all time levels. However, if the solution of the problem has large variations in some regions (which may change for different time levels) of the domain, then a very fine mesh must be used to resolve the solution in the regions of rapid change in the solution. This leads to a very high computational cost and becomes impractical for higher dimensional problems. One way to resolve this difficulty is to use an adaptive strategy. For example, one places sufficient nodes in regions where the solution suffers some computational difficulties and fewer nodes in regions where it is smooth. The MOL separates the spatial and temporal variables which makes it possible to use different meshes at different time levels. In what follows, we describe the adaptive mesh methods or moving mesh methods in one spatial dimension.

Consider the problem of generating an adaptive mesh that is suitable for solving a time-dependent PDE. The adaptive mesh is obtained as the image of the coordinate transformation

\[ x : \Omega_c \rightarrow \Omega, \quad x = x(\xi, t), \quad \xi \in \Omega_c, \quad x \in \Omega, \quad t \geq 0, \]

where we call \( \Omega_c \) and \( \Omega \) the computational domain and the physical domain, respectively. The mesh in the computational domain is typically assumed to be uniform. Therefore, solving the given problem numerically in terms of the computational domain variable \( \xi \) becomes easier than using the physical domain variable \( x \).

The adaptive mesh at a time \( t \) is generated as the image of the transformation \( x = x(\xi, t) \) (in the discrete form). This transformation is computed by finding a new set of partial differential equations (PDEs), known as moving mesh PDEs. The mesh PDEs can be solved either simultaneously or alternately with the original physical PDE. In one spatial dimension, only one mesh PDE is needed to solve for the transformation. The mesh PDE in this case is derived by requiring that the transformation \( x = x(\xi, t) \) to satisfy the equidistribution principle (EP). The EP was introduced by de Boor [dB74] and Dodson [Dod72] for solving boundary value problems for ordinary differential equations. In the next subsection, we explain the equidistribution principle.
2.1.1 The Equidistribution Principle

In one spatial dimension, the idea of equidistribution is to position the mesh points in such a way that some measure of the error, or the solution variation, is the same in each subinterval. In practice, the mesh is determined by approximately equidistributing an adaptation function (monitor function) $\rho(x, t)$. Mathematically, the problem is to find the set of points $x_i, i = 0, 1, \ldots, N$ such that

$$\int_{x_{i-1}(t)}^{x_{i}(t)} \rho(x, t) dx = \frac{1}{N} \int_0^1 \rho(x, t) dx = \frac{1}{N} \theta(t), \quad i = 1, 2, \ldots, N, \quad (2.1)$$

where $N + 1$ is the total number of mesh points, and $\theta(t)$ is the total integral of the adaptation function on the whole domain. Without loss of generality, we take $\Omega_c = \Omega = [0, 1]$ and $t \in [0, 1]$. Notice that, for a fixed time $t$, the right hand side of (2.1) is constant. This means that at any fixed time $t$ the integral of the adaptation function $\rho(x, t)$ is the same in each subinterval.

Equation (2.1) can be rewritten as,

$$\int_0^{x_{i}(t)} \rho(x, t) dx = \frac{i}{N} \theta(t) = \xi_i \theta(t), \quad i = 1, \ldots, N, \quad (2.2)$$

where we have used the fact that the mesh points in the computational domain $\Omega_c = [0, 1]$ are evenly spaced, i.e.,

$$\xi_i = \frac{i}{N}, \quad \text{for } \xi_i \in \Omega_c = [0, 1], \quad i = 0, 1, \ldots, N.$$

The corresponding continuous form of the equidistribution form (2.2) is

$$\int_0^x \rho(\bar{x}, t) d\bar{x} = \xi \theta(t), \quad \xi \in [0, 1]. \quad (2.3)$$

Two differential forms of the equidistribution principle follow from (2.3) by differentiation with respect to $\xi$,

$$\rho(x, t) \frac{\partial x}{\partial \xi} = \theta(t), \quad (2.4)$$

and differentiating (2.4) with respect to $\xi$,

$$\frac{\partial}{\partial \xi} \left( \rho(x, t) \frac{\partial x}{\partial \xi} \right) = 0, \quad (2.5)$$
where the transformation \( x = x(\xi, t) \) subject to the conditions, \( x(0, t) = 0 \) and \( x(1, t) = 1 \). Solving (2.4) or (2.5) gives the required equidistributed mesh.

For a fixed time \( t \) the right hand side of (2.4) is constant. This shows that the change in the physical domain variable \( x \), or the spacing between the nodes in the physical domain, is small in regions where the measure of the error or the measure of the solution variation, defined by the adaptation function \( \rho(x, t) \), is large.

In practice the solution of (2.5) is obtained as the steady-state solution of the time-dependent parabolic PDE (cf. [HRR94a])

\[
\frac{\partial x}{\partial t} = \frac{1}{\tau} \frac{\partial}{\partial \xi} \left( \rho \frac{\partial x}{\partial \xi} \right),
\]

where \( \tau \) is a positive constant used as a time smoothing parameter. The moving mesh equation (2.6) is referred to as MMPDE5.

Two other moving mesh methods called the MMPDE4 and MMPDE6 are, respectively,

\[
\frac{\partial}{\partial \xi} \left( \rho \frac{\partial x}{\partial \xi} \right) = -\frac{1}{\tau} \frac{\partial}{\partial \xi} \left( \rho \frac{\partial x}{\partial \xi} \right),
\]

and

\[
\frac{\partial^2 x}{\partial \xi^2} = -\frac{1}{\tau} \frac{\partial}{\partial \xi} \left( \rho \frac{\partial x}{\partial \xi} \right),
\]

where \( \dot{x} = \partial x / \partial t \). MMPDE4 (2.7) and MMPDE6 (2.8) follow from the differential forms of the equidistribution principle forms (2.4) and (2.5) by time differentiation (for example, see [HRR94a], [Su103] for the details).

### 2.1.2 Adaptation Function

In the previous subsection we showed that the mesh distribution in the physical domain is controlled by the adaptation function \( \rho(x, t) \). The effect of the adaptation function is to concentrate sufficient mesh points in regions where \( \rho(x, t) \) is large. Therefore, this function must be defined so that it reflects some measure of the solution variation, or the solution error. Thus, the success of the mesh adaptation depends heavily on the selection of the function \( \rho(x, t) \).
The adaptation function is very problem dependent. A good choice of the adaptation function is the one which is capable of concentrating enough mesh points to resolve the physical solution in these regions of large solution variation, or solution error. The simplest choice is just the solution gradient, i.e.,

$$ \rho = u_x, \quad (2.9) $$

where \( u_x \) is the spatial derivative with respect to the physical domain variable. Here, we assume \( u \) is a nondecreasing function, i.e., \( u_x \geq 0 \). In this case, at a fixed time \( t \), the equidistribution form (2.4) becomes

$$ x_x u_x = \text{constant}, $$

and this leads to

$$ u_x = \text{constant}. $$

In this case, the effect of the adaptation function is that the mesh points are distributed so that the same change in the physical solution \( u \) occurs over each subinterval, as illustrated in Figure 2.1. The disadvantage of this choice of an adaptation function is that it tends to make the spacing infinitely large where the solution is flat.

An alternative to the gradient adaptation function (2.9) is the arc-length function defined as

$$ \rho(x, t) = \sqrt{1 + u_x^2}. \quad (2.10) $$

An increment of arc-length, \( ds \), on the solution curve \( u(x) \) is given by

$$ ds^2 = dx^2 + du^2 = (1 + u_x^2)dx^2. \quad (2.11) $$

This leads to

$$ \rho = s_x. $$

In this case, at a fixed time \( t \), the equidistribution form (2.4) becomes

$$ s_x x_x = \text{constant}, $$
and this gives

\[ s_\xi = \text{constant}. \]

Therefore, the arc-length adaptation function distributes the mesh points in such a way that the same arc-length on the solution curve occurs over each subinterval, as shown in Figure 2.2. For the choice of arc-length adaptation function, the mesh points are evenly spaced when the solution is flat. However, the concentration of the mesh points, in the regions where \( u_x \) is large, is not as high as for the gradient adaptation function choice. This concentration can be increased by using the scaled arc-length function

\[ \rho(x, t) = \sqrt{1 + \alpha^2 u_x^2}, \quad (2.12) \]

where \( \alpha \) is a parameter to be specified. Note that, for the scaled arc-length adaptation function, the mesh points are still evenly spaced where the solution is flat.

The arc-length adaptation function has a disadvantage that the mesh points are
concentrated only in regions where the gradient is high, not near the critical points of the solution, i.e. \( u_x = 0 \). To concentrate the points near the critical points, one can choose the curvature adaptation function

\[
\rho(x, t) = 1 + \alpha^2|\kappa|,
\]

where \( \kappa \) is the solution curvature defined as

\[
\kappa = \frac{u_{xx}}{(1 + u_x^2)^{3/2}}.
\]

In order to have the effects of both the arc-length and the curvature adaptation functions (2.12) and (2.13), respectively, one chooses a combined adaptation function

\[
\rho(x, t) = (1 + \beta^2|\kappa|)\sqrt{1 + \alpha^2u_x^2}.
\]

We assume that the adaptation function \( \rho(x, t) \) is positive definite for all time \( t \in [0, 1] \), to assure that the mapping \( x = x(\xi, t) \) is well defined. Geometrically this means that mesh nodes do not cross.
2.2 Higher Dimensional Adaptation

In the previous section we discussed the basic principles of the one dimensional adaptive mesh methods. In this section, we extend these principles in higher dimensions and review two-dimensional adaptive mesh methods as described in [HS94], [CHR02] and [BW06].

Consider a coordinate transformation:

\[ x : \Omega_c \rightarrow \Omega, \quad x = x(\xi, t), \quad \Omega_c, \Omega \subset \mathbb{R}^d, d \geq 2. \quad (2.15) \]

In what follows, we use the notation \( \xi = [\xi_1, \xi_2, \cdots, \xi_d]^T \) and \( x = [x_1, x_2, \cdots, x_d]^T \) to denote the spatial coordinates on the computational domain \( \Omega_c \) and the physical domain \( \Omega \) respectively (see Figure 2.3 for the 2D case).

\[ x = x(\xi, t) \]

Figure 2.3: Representation of mesh adaptation as a coordinate transformation \( x = x(\xi, t) \) from \( \Omega_c \) to \( \Omega \) for the 2D case.

A general form of (2.11) for the arc-length increment on the solution curve \( u(x, t) \)
is written as
\[ ds^2 = dx^T dx + (du)^2, \]  \hspace{1cm} (2.16)
where in 3D
\[ dx = \begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{bmatrix}. \]
We can rewrite (2.16) as
\[ ds^2 = dx^T M dx, \]  \hspace{1cm} (2.17)
where \( M \) is a 3 x 3 matrix defined by
\[ M = \nabla u \cdot \nabla u^T + I, \]
where \( I \) is a 3 x 3 identity matrix and
\[ \nabla u = \begin{bmatrix} \frac{\partial u}{\partial x_1} \\ \frac{\partial u}{\partial x_2} \\ \frac{\partial u}{\partial x_3} \end{bmatrix}. \]
The change in the physical domain variables in terms of the computational domain is expressed as follows:
\[ dx_i = \frac{\partial x_i}{\partial \xi_1} d\xi_1 + \frac{\partial x_i}{\partial \xi_2} d\xi_2 + \frac{\partial x_i}{\partial \xi_3} d\xi_3, \quad i = 1, 2, 3. \]  \hspace{1cm} (2.18)
Thus, we can rewrite the arc-length increment (2.17) as,
\[ ds = \left[ d\xi^T \nabla x^T M \nabla x d\xi \right]^{1/2}, \]  \hspace{1cm} (2.19)
where \( \nabla x \) is the Jacobian matrix of the coordinate transformation (2.15) given by
\[ \nabla x = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_1}{\partial \xi_3} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \\ \frac{\partial x_3}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_3} \end{bmatrix}. \]  \hspace{1cm} (2.20)
From (2.19), one observes that if the solution \( u \) is required to have the same variation \( ds \) along any arc in the computational domain which has a fixed length,
i.e. \( [d\xi^T d\xi]^{1/2} \) is constant, then the right-hand side of (2.19) must be independent of \( \xi \). This leads to the conclusion that, in higher dimensions, the equivariation of the solution \( u \) requires \( \nabla x^T M \nabla x \) to be independent of \( \xi \), i.e.,

\[
[d\xi^T \nabla x^T M \nabla x d\xi]^{1/2} = [d\xi^T \tilde{M} d\xi]^{1/2},
\]

(2.21)

where \( \tilde{M} \) is a \( 3 \times 3 \) symmetric positive definite, and \( \xi \)-independent, matrix.

Notice that in 1D (2.21) reduces to

\[
\left[ \left( \frac{\partial x}{\partial \xi} \right)^2 M \right]^{1/2} = \text{constant},
\]

(2.22)

where

\[
M = 1 + \left( \frac{\partial x}{\partial \xi} \right)^2.
\]

Equation (2.22) is the 1D equidistribution principle (2.4) for \( \rho = \sqrt{M} \). Therefore, the extension of the one-dimensional adaptation function (2.10) to higher dimensions can be defined as

\[
\rho(x, t) = \sqrt{1 + |\nabla u|^2},
\]

(2.23)

where \( \nabla \) is the gradient operator with respect to the physical domain variable.

The equidistribution principle (2.21) can be used to determine the coordinate transformation \( x = x(\xi, t) \) for generating an adaptive mesh on the physical domain \( \Omega \). In principle, (2.21) can be discretized by any suitable finite difference scheme on the computation space to give a system of equations. The adaptive mesh is then obtained by solving this system with some discrete boundary conditions for the coordinate transformation (2.15).

For a fixed number of mesh points, in higher dimensions, the transformation (2.15) doesn’t satisfy (2.21) throughout the physical domain. However, an adaptive mesh can be obtained by requiring the transformation to only satisfy (2.21) locally. Huang and Sloan [HS94] describe a simple mesh adaptation method by assuming (2.21) is satisfied along the first coordinate line and then along each of the second and the third. This is basically applying separately the one-dimensional equidistribution principle along each of the coordinate lines. A drawback of this approach of mesh adaptation is that
the adaptive mesh is obtained as a result of grid movement along a certain family of fixed lines, and this can result in grid distortion. Alternative higher dimensional mesh adaptation strategies have been developed to avoid such kinds of mesh distortion. In the current thesis we present two different methods for generating adaptive mesh in higher dimensions based on solving the optimal mass transport problem. Huang and Russell [HR99] derived a two-dimensional moving mesh strategy based on a gradient flow equation. Cao, Huang and Russell [CHR02] developed a two-dimensional moving mesh method called the GCL method. This is a velocity based method which is based on the geometric conservation law. Budd and Williams [BW06] described the PMA moving mesh method in two spatial dimensions based on solving the parabolic Monge-Ampèere equation.

In order to give more understanding of the adaptive mesh techniques in higher dimensions, and because of their close relation to the adaptive methods that we present in this thesis, in the next two subsections we review the GCL and the PMA methods.

2.2.1 The GCL Adaptive Mesh Method

In this subsection we summarize the GCL method as described in [CHR02]. The key idea of the GCL method is that the mesh adaptation is achieved by specifying the Jacobian of the transformation and controlling the mesh speed by using the geometric conservation law.

The geometric conservation law was first proposed by Thomas and Lombard [TL79] in the context of fluid dynamics. In two spatial dimensions the GCL integral form can be written as

$$\frac{d}{dt} \int_{S(t)} dx dy = \int_{\partial S(t)} \mathbf{x}_t \cdot \mathbf{n} \, dx dy, \quad \forall S(t) \subset \Omega,$$  \hspace{1cm} (2.24)

where \( \mathbf{n} \) is the outer normal to the boundary \( \partial S(t) \) and \( \mathbf{x}_t \) is the mesh velocity. The GCL form (2.24) defines the rate of change in surface \( S(t) \) as the total flux through its boundary \( \partial S(t) \).

By a change of variables, the left-hand side of (2.24) can be written as

$$\frac{d}{dt} \int_{S(t)} dx dy = \frac{d}{dt} \int_{S} J(\xi, \eta, t) \, d\xi d\eta = \int_{S} \frac{D}{Dt} J(\xi, \eta, t) \, d\xi d\eta,$$  \hspace{1cm} (2.25)
where \( J(\xi, \eta, t) \) is the determinant of the Jacobian of the transformation, \( S_c \) is the cell in \( \Omega_c \) corresponding to the cell \( S(t) \) and

\[
\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{x}_i \cdot \nabla,
\]

for which \( \nabla \) is the spatial gradient operator with respect to the physical domain variables.

Using the divergence theorem, we write the right-hand side of (2.24) as

\[
\int_{\partial S(t)} \mathbf{x}_i \cdot \mathbf{n} \, dx dy = \int_{S(t)} \nabla \cdot \mathbf{x}_i dx dy = \int_{S_c} (\nabla \cdot \mathbf{x}_i) J(\xi, \eta, t) d\xi d\eta.
\]

Substituting (2.25) and (2.26) into (2.24) gives

\[
\int_{S_c} \frac{D}{Dt} J(\xi, \eta, t) d\xi d\eta = \int_{S_c} (\nabla \cdot \mathbf{x}_i) J(\xi, \eta, t) d\xi d\eta.
\]

Since (2.27) holds for all arbitrary \( S_c \), then one obtains the differential form of the

GCL

\[
\nabla \cdot \mathbf{x}_i = \frac{1}{J} \frac{D}{Dt}.
\]

When generating an adaptive mesh as an image of a coordinate transformation, one wants to cluster the mesh nodes in regions where a measure of the error indicator or the solution variation (i.e., the adaptation function) is large. In these regions the Jacobian of the transformation must be small. This means that the Jacobian and the adaptation function are inversely proportional. Thus, the Jacobian of the transformation can be chosen as

\[
J(\xi, \eta, t) = \frac{c(\xi, \eta)}{\rho(x(\xi, \eta, t), t)},
\]

where \( c(\xi, \eta) \) is a time-independent function which is determined from the initial coordinate transformation and \( \rho(x, y, t) > 0 \) is the user defined adaptation function. Note that, in one spatial dimension with \( c(\xi, \eta) = 1 \), (2.29) reduces to the equidistribution principle discussed in Subsection 2.1.1. This suggests that we can view (2.29) as a generalization of the one-dimensional equidistribution principle for problems in higher dimensions.
In order to derive the moving mesh PDEs for the GCL method we substitute (2.29) into (2.28), and simple calculations lead to

\[ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{x}_i) = 0, \]  
(2.30)

where \( \nabla \) is the gradient operator with respect to the physical domain variables \( \mathbf{x} = (x, y) \).

Equation (2.30) or equivalently (2.29) alone is insufficient to uniquely determine the mesh velocity field for 2D or higher spatial dimensions problems since it is one equation in more than one unknown to be determined. Therefore, additional equations are required to uniquely specify the mesh velocity. Various choices of additional conditions lead to deriving different methods of mesh adaptation. For example, Cao, Huang and Russell in [CHR02] have used the Helmholtz’s decomposition theorem to obtain an additional equation for the 2D case. This theorem is stated as follows:

**Theorem 2.1.** (Helmholtz’s decomposition) Any sufficiently smooth vector field \( \mathbf{V} \) can be resolved into irrotational (curl-free) and solenoidal (divergence-free) component vector fields, i.e.,

\[ \mathbf{V} = \mathbf{U} + \nabla P, \]  
(2.31)

for a smooth potential \( P \) and a vector field \( \mathbf{U} \) such that \( \nabla \cdot \mathbf{U} = 0 \).

\[ \nabla \times w(\mathbf{v} - \mathbf{u}) = 0 \quad \text{in} \quad \Omega, \]  
(2.32)

where \( w \) and \( \mathbf{u} \) are, respectively, a weight function and a background velocity field to be specified, and \( \mathbf{v} \) is the mesh velocity field which was previously denoted by \( \mathbf{x}_i \).

Equation (2.32) gives

\[ \mathbf{v} = \mathbf{u} + \frac{1}{w} \nabla \phi, \]  
(2.33)

where \( \phi \) is a smooth potential function.
CHAPTER 2. BACKGROUND AND MOTIVATION

In summary, with the assumption that no mesh points move in or out of the physical domain \( \Omega \), one can generate an adaptive mesh using the GCL method by solving the following elliptic system:

\[
\nabla \cdot \left( \frac{\rho}{w} \nabla \phi \right) = -\frac{\partial \rho}{\partial t} - \nabla \cdot (\rho \mathbf{u}) \quad \text{in } \Omega, \tag{2.34a}
\]

\[
\frac{\partial \phi}{\partial n} = -w \mathbf{u} \cdot \mathbf{n} \quad \text{on } \partial \Omega. \tag{2.34b}
\]

This system is solved for the potential \( \phi \), and then the mesh velocity is obtained by taking the spatial gradient of \( \phi \) with respect to the physical domain variable. Then, the transformation \( \mathbf{x} = \mathbf{x}(\xi, t) \) can be obtained by time integration of

\[
\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t) + \frac{1}{w(\mathbf{x}, t)} \nabla \phi(\mathbf{x}, t). \tag{2.35}
\]

starting from \( \mathbf{x}(\xi, 0) \).

Advantages of the GCL method include a theoretical assurance of existence and nonsingularity of the coordinate transformation. Another nice feature of this method is that the rate of change in cell volumes is shown to be proportional to the rate of change in the adaptation function \( \rho(\mathbf{x}, t) \). Therefore, the mesh adaptation can be achieved by controlling the cell volumes. A drawback of the GCL approach is that, in general, skew and nonsmooth adaptive meshes may not be avoided due to its close relation with the Lagrangian method. Also note that the elliptic system (2.34) is quite expensive to solve. Thus, an alternative approach is needed to overcome these challenges.

2.2.2 The PMA Adaptive Mesh Method

In this subsection we review the parabolic Monge-Ampère (PMA) moving mesh method as described in [BW06] for the case of two spatial dimensions.

Let \( \mathbf{x} = \mathbf{x}(\xi, \eta, t), t > 0 \) be a two-dimensional coordinate transformation from a computational domain \( \Omega_e \) to a physical domain \( \Omega \).

To generate an adaptive mesh with the PMA method, the transformation \( \mathbf{x} = \mathbf{x}(\xi, \eta, t) \) is computed as the gradient of some potential function \( Q \) which is determined as a solution of a parabolic Monge-Ampère equation. Precisely speaking, the
transformation can be uniquely determined by specifying two conditions. The first one is to satisfy the equidistribution principle

\[ \rho(x, y, t) \det \nabla x = \theta(t), \quad \theta(t) = \int_{\Omega} \rho(x, y, t) \, dx \, dy, \]  \hspace{1cm} (2.36)

where \( \det \nabla x \) is the determinant of the Jacobian matrix of the transformation \( x(\xi, \eta, t) \) and \( \rho(x, y, t) \) is the adaptation function. The second condition is to require the transformation \( x = x(\xi, \eta, t) \) to be a Legendre transformation. This means that there exists a potential \( P(\xi, \eta, t) \) such that

\[ x(\xi, \eta, t) = \nabla P(\xi, \eta, t), \]  \hspace{1cm} (2.37)

where \( \nabla \) is the gradient operator with respect to the computational domain variables \( (\xi, \eta) \). The result (2.37) also follows directly from the Monge-Kantorovich theory which will be discussed in more detail in Section 2.3 below.

Substituting (2.37) into (2.36) leads to the Monge-Ampère equation (MAE)

\[ \rho(\nabla P, t) \det D^2 P(\xi, \eta, t) = \theta(t), \]  \hspace{1cm} (2.38)

where \( \det D^2 P(\xi, \eta, t) \) is the determinant of the Hessian matrix of \( P(\xi, \eta, t) \) which is defined as

\[ D^2 P = \begin{bmatrix} P_{\xi\xi} & P_{\xi\eta} \\ P_{\eta\xi} & P_{\eta\eta} \end{bmatrix}. \]

The MAE (2.38) is fully nonlinear and its numerical solution is quite challenging. Therefore, a relaxed form of (2.38) is used to obtain the adaptive mesh. The relaxed form is written as (cf. [BW08])

\[ \epsilon(I - \nu \Delta) Q_t = \left( \rho(\nabla Q) \det D^2 Q(\xi, \eta, t) \right)^{\frac{1}{2}}, \]  \hspace{1cm} (2.39)

where the \( \Delta \) is the spatial Laplacian with respect to the computational variables and \( \epsilon \) and \( \nu \) are parameters which control temporal and spatial smoothing, respectively.

The PMA method is computationally simpler than the GCL method. For example, one can use simple forward Euler scheme to integrate (2.39) for the scalar potential function \( Q \). It also has some nice features such as that the transformation obtained
by PMA method is curl-free. Disadvantages of the PMA method are the lack of controlling the skewness of the mesh near the boundaries, so the mesh points are forced to be orthogonal to the boundaries, and the loss of the equidistribution property over time.

The PMA and the GCL methods can be related since they both use the equidistribution relation (2.29) or (2.36) in their formulations. However, the two methods are different in defining the additional closure condition to uniquely specifying the two-dimensional adaptive mesh. The PMA method is a location based method because it determines directly the coordinate transformation defined from the computational domain to the physical domain. On the other hand the GCL method is a velocity based method since the mesh adaptation is achieved by determining the mesh velocity.

2.3 The Monge–Kantorovich Problem

In this section we review the Monge–Kantorovich problem and discuss some theoretical results related to this problem.

The original Monge problem is stated as follows: Given two positive density functions of equal masses of a given material, \( \rho_0 \) and \( \rho_1 \) defined in \( \mathbb{R}^d, d \geq 1 \), find a transport map \( x = \phi(\xi), \xi, x \in \mathbb{R}^d \), that transfers the density \( \rho_0 \) to \( \rho_1 \) and minimizes the transport cost

\[
C(\phi) = \int_{\mathbb{R}^d} |\phi(\xi) - \xi| \rho_0(\xi) d\xi.
\]

Mathematically, the map \( \phi \) realizes the transfer of the density \( \rho_0 \) to \( \rho_1 \) if

\[
\int_{\phi^{-1}(\Omega)} \rho_0(\xi) d\xi = \int_{\Omega} \rho_1(x) dx, \quad \forall \Omega \subset \mathbb{R}^d,
\]

where \( | \cdot | \) is the standard Euclidean norm defined on the space \( \mathbb{R}^d \).

If \( x = \phi(\xi) \) is a smooth one-to-one map then, by change of variables, (2.41) leads to the Jacobian equation

\[
\rho_1(\phi(\xi)) |\nabla \phi(\xi)| = \rho_0(\xi),
\]

where \( |\nabla \phi(\xi)| \) is the determinant of the Jacobian matrix for the transformation \( \phi \).
Observe that (2.42) is very similar to the equidistribution equation (2.29) which was introduced in the GCL method formulation for the mesh adaptation. In fact they are identical if at a fixed time $t$ we think of the density function $\rho_1(\phi(\xi))$ as the adaptation function $\rho(x(\xi), t)$ and $\rho_0(\xi) = c(\xi)$. This shows that the coordinate transformation obtained as a result of solving the Monge–Kantorovich problem can be used for adaptive grid generation in higher dimensions. This will be described in more detail in Chapter 3 of this thesis.

More generally, the cost functional (2.40) can be written as

$$
C(\phi) = \int_\Omega |\phi(\xi) - \xi| p \rho_0(\xi) d\xi, \quad p \geq 1.
$$

(2.43)

The integrand in (2.43) gives the traveled distance weighted by the amount of the transported mass. We can view the cost (2.43) as the total amount of effort required or work done to move one pile of mass from one location to another.

**Definition 2.1.** The $L^p$ Kantorovich–Wasserstein distance between two density functions $\rho_0$ and $\rho_1$ is defined as,

$$
d(\rho_0, \rho_1) = \inf_{\phi} \left( \int_{\Omega_0} |\phi(\xi) - \xi|^p \rho_0(\xi) d\xi \right)^{\frac{1}{p}}, \quad p \geq 1,
$$

(2.44)

where $| \cdot |$ denotes the Euclidean norm in $\mathbb{R}^d$ and the infimum is taken over all maps $\phi$ transporting $\rho_0$ to $\rho_1$.

A one-to-one mapping $x = \phi(\xi)$ that satisfies (2.42) and minimizes the cost (2.43) is then called the *optimal solution* to the $L^p$ Monge–Kantorovich problem ($L^p$ MKP).

This thesis will restrict consideration to the case of the $L^2$ cost (i.e. $p = 2$) because of the simplified mathematical formulation, the theory available for this case, and the broad range of applications in various areas of applied science. The $L^2$ cost has been extensively used in many different application areas, such as mathematical finance, functional analysis, atmospheric science, shape recognition in image processing, computer vision, and signal processing (for example see the book [Vil03] and the references therein).
The existence and the characterization of the solution for the Monge-Kantorovich problem is addressed in the following theorem for the $L^2$ cost (see [KS84], [Bre91a]).

**Theorem 2.2.** There is a unique optimal transport map $\hat{x} = \hat{\phi}(\xi)$ that transfers $\rho_0(\xi)$ to $\rho_1(\mathbf{x})$. Moreover, this map can be written as the gradient of a convex potential function $\Psi(\xi)$, i.e.,

$$\hat{x} = \hat{\phi}(\xi) = \nabla \Psi(\xi).$$

The above theorem is a direct result of the polar factorization of vector fields which is a generalization of the Helmholtz's decomposition (for example, see [Bre91b, BB94, Gan94]). Recall that Helmholtz's decomposition has been used in the GCL method to determine the mesh velocity.

Substituting (2.45) into (2.42) shows that $\Psi$ is a solution of the Monge-Ampère equation (MAE)

$$\rho_1(\nabla \Psi(\xi)) \det D^2 \Psi(\xi) = \rho_0(\xi),$$

where $D^2 \Psi$ is the Hessian matrix of the potential $\Psi$.

This result concludes that the solution of the $L^2$ Monge-Kantorovich mass transfer problem reduces to solving the Monge-Ampère equation (2.46). However, this equation is fully nonlinear. A direct numerical solution of (2.46) raises many numerical challenges, some of which we address in Chapter 3. Avoiding solving (2.46) directly, Benamou and Brenier introduce a fluid dynamics framework for the Monge-Kantorovich problem [BB00]. In the next section, we briefly summarize the fluid dynamics approach for solving the $L^2$ Monge-Kantorovich problem ($L^2$ MKP).

### 2.4 The Fluid Dynamics Framework of the MKP

In this section we summarize the fluid dynamics formulation of the $L^2$ Monge-Kantorovich problem as described in [BB00].

Consider a time interval $[0,T]$ which for simplicity we take to be $[0, 1]$, and consider all possible sufficiently smooth time-dependent density and velocity fields $\rho(\mathbf{x}, t) > 0$, $\mathbf{v}(\mathbf{x}, t)$.
\( v(x, t), x \in \mathbb{R}^d \), which satisfy the continuity equation
\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0
\]  \hspace{1cm} (2.47)
and the initial and final conditions
\[
\rho(x, 0) = \rho_0(x), \quad \rho(x, 1) = \rho_1(x).
\]  \hspace{1cm} (2.48)

The relationship between the \( L^2 \) Monge-Kantorovich problem and the fluid dynamics is established by the following proposition.

**Proposition 2.1** (Benamou & Brenier [BB00]). The square of the \( L^2 \) Kantorovich Wasserstein distance is equal to the infimum of the cost functional
\[
C(\rho, v) = \int_0^1 \int_{\mathbb{R}^d} \rho(x, t) |v(x, t)|^2 \, dx \, dt,
\]  \hspace{1cm} (2.49)
over all the density and velocity fields \( \rho(x, t) \) and \( v(x, t) \) that satisfy the conditions (2.47) and (2.48).

This proposition shows that the \( L^2 \) Monge-Kantorovich problem can be reformulated as finding the velocity vector field \( v(x, t) \) by minimizing the functional (2.49) under the constraints (2.47) and (2.48).

The optimality conditions of the minimization problem (2.49) under the constraints (2.47) and (2.48) can be written as follows:
\[
v(x, t) = \nabla \Phi(x, t),
\]  \hspace{1cm} (2.50)
where the potential \( \Phi \) is the Lagrange multiplier of the constraints (2.47) and (2.48), and satisfies the Hamilton-Jacobi equation
\[
\partial_t \Phi + \frac{1}{2} |\nabla \Phi|^2 = 0.
\]  \hspace{1cm} (2.51)
In the fluid dynamics context this says the optimal solution is given by a potential flow which is pressure free.

Notice that the velocity field (2.50) is of the same form as the velocity field (2.33), with \( w = 1 \) and \( u = 0 \), used for mesh adaptation in the GCL method.


2.5 Image Registration

Image registration is the process of finding a suitable spatial transformation so that each point in one image can be mapped to a point in the second image. The mapping should "optimally" match the two images in some sense, for example, minimizing the $L^2$ norm of the difference between the two images. The optimality here depends on what needs to be matched. Registration is important in finding changes in images taken from the same object at different times, from different imaging devices or from different viewpoints. In this section, we give a brief introduction to the image registration problem.

Definition 2.2. Let $d \in \mathbb{N}$. A $d$-dimensional image is defined as a mapping,

$$
\rho : \mathbb{R}^d \rightarrow \mathbb{R}^+, \text{ such that } 0 \leq \rho(x) < \infty, \forall x \in \mathbb{R}^d,
$$

where $\rho(x)$ gives the intensity of the image at a spatial position $x \in \mathbb{R}^d$ and satisfies the condition

$$
\int_{\mathbb{R}^d} \rho(x)^k dx \text{ is finite, for } k > 0.
$$

The set of all images is denoted by

$$
\text{Img}(d) := \{ \rho : \mathbb{R}^d \rightarrow \mathbb{R} \mid \rho \text{ is a } d \text{ - dimensional image} \}.
$$

Image registration means establishing a suitable spatial transformation such that the transformed (deformed) image becomes similar to another one. Typically, one of the images is viewed as a template $\rho_0$ and the other one as a target $\rho_1$. In this case, we seek a coordinate transformation $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$, such that the target $\rho_1$ and the deformed template image are similar (see Figure 2.5 for the case $d = 2$). Mathematically, an appropriate similarity measure needs to be defined. Different similarity measures $\mathcal{D}$ leads to different methods of image registration. The general registration problem is stated as follows.
**Definition 2.3.** Given a distance measure $D : \text{Img}(d) \rightarrow \mathbb{R}$ and two images, $\rho_0$ and $\rho_1 \in \text{Img}(d)$ defined on $\Omega \subseteq \mathbb{R}^d$, $d = 2$ or $3$ (see Figure 2.4 for the 2D case), find a transformation,
\[ \phi : \Omega \longrightarrow \Omega, \]
such that $D(\rho_1, \rho_0 \circ \phi^{-1})$ is minimum.

Figure 2.4: Transformation of two squares, $\phi$ transforms a template image $\rho_0$ (left) to a target image $\rho_1$ (right).

In order to illustrate how to compute the optimal transformation $\phi$ for image registration, in the next two subsections we briefly summarize two image registration methods. The first method is introduced by Angenent, Haker and Tannenbaum [HT01, AHT03]. Their method is based on deriving a partial differential equation to minimize the $L^2$ Kantorovich-Wasserstein distance under a mass preservation constraint. For future reference call this the AHT method. The second method is of Beg et al. (cf. [BMTY05]) which is based on minimizing an energy functional over
CHAPTER 2. BACKGROUND AND MOTIVATION

2.5.1 AHT Image Registration Method

Given two images $\rho_0$ and $\rho_1$ defined on $\Omega_0$ and $\Omega_1$, respectively, and consider the set of all diffeomorphisms $\phi$ that satisfy

$$
\rho_0(\xi) = \rho_1(\phi(\xi)) \det \nabla \phi(\xi), \xi \in \Omega_0,
$$

where $\det \nabla \phi(\xi)$ is the determinant of the Jacobian matrix of the diffeomorphism $\phi$.

The similarity between the two images $\rho_0$ and $\rho_1$ is measured by their $L^2$ Kantorovich-Wasserstein distance defined as

$$
\ell_2(\rho_0, \rho_1)^2 = \inf_{\phi} \int_{\Omega_0} |\phi(\xi) - \xi|^2 \rho_0(\xi) d\xi,
$$

where the infimum is taken among all $\phi$ satisfying (2.52).

Figure 2.5: Three images of a human brain given as a template image $\rho_0$ (left), a reference (target) image $\rho_1$ (middle) and a deformed template (right). The image was extracted from the web in the year 2007 during the development of the thesis.
CHAPTER 2. BACKGROUND AND MOTIVATION

Given an initial mapping \( \phi_0 : \Omega_0 \longrightarrow \Omega_1 \) with the mass preserving property (2.52), the minimizer in (2.53) is shown to be the steady-state solution of the evolution equation (see [HT01, AHT03, HT03a, HT03b, HZTA04])

\[
\frac{\partial \phi(\xi)}{\partial t} = -\frac{1}{\rho_0(\xi)} \left( \phi(\xi) - \nabla \Delta^{-1} \text{div}(\phi(\xi)) \right) \cdot \nabla \phi(\xi).
\] (2.54)

In [HZTA04] it is proposed to compute the mapping \( \phi_0 \) by solving a family of one-dimensional mass transport problems. First, transport mass along lines parallel to the x-axis, and then afterward transport mass along lines parallel to the y-axis.

The AHT registration method, described above, has some nice features that are directly obtained from the Kantorovich theory. For example, the optimal transformation automatically adjusts changes in density that result from changes in area or volume. This method has some disadvantages such as the need of computing an initial mapping with the mass preserving property (this is not a straightforward procedure), and it also requires inverting the Laplacian during each iteration. This method generally needs a significant amount of time for the convergence of (2.54) to the steady-state solution. In this thesis we investigate elastic registration based on solving the optimal mass transport problem with some simpler and efficient methods.

2.5.2 LDDMM Image Registration Method

In this subsection we review the large deformation diffeomorphic metric mapping (LDDMM) registration method as described in [BMTY05].

Consider a family of diffeomorphisms (smooth invertible transformations with smooth inverse)

\[
\phi(\cdot, t) : \Omega \rightarrow \Omega, \quad \Omega \subset \mathbb{R}^d, \quad d = 2 \text{ or } 3, \quad 0 \leq t \leq 1.
\]

The goal is to find a particular \( \phi \) that registers the given two images \( \rho_0 \) and \( \rho_1 \).

The diffeomorphisms are governed by the flow equation

\[
\frac{\partial \phi}{\partial t} = \mathbf{v}, \quad \phi(\cdot, 0) = \text{id},
\] (2.55)

where \( \mathbf{v} = \mathbf{v}(\phi(\cdot, t), t), \quad t \in [0, 1] \) is the associated velocity field and id is the identity transformation.
We seek a time-dependent velocity vector field \( v \) that gives the optimal matching diffeomorphism. The associated velocity vector field \( v \) is computed as the solution of the variational problem (see [BMTY05], [MTY02])

\[
\arg\min_v \left( \int_0^1 \|v\|^2 \, dt + \frac{1}{\sigma^2} \|\rho_0 \circ \phi_t^{-1} - \rho_1\|^2 \right),
\]

where \( \arg\min \) stands for the argument of the minimum, \( V \) is the space of all smooth velocity vector fields defined on the domain \( \Omega \), the norm \( \| \cdot \|_V \) is defined as

\[
\|f\|_V = \|L f\|_2,
\]

where \( \| \cdot \|_2 \) is the \( L^2 \) norm for square integrable functions \( \|f\|_2^2 = \int_\Omega |f(x)|^2 \, dx \), and \( L \) is a differential operator defined as

\[
L = (-\alpha \Delta + \nu) I_{d \times d}
\]

where \( I_{d \times d} \) is a \( d \times d \) identity matrix. The coefficient \( \alpha \) is a smoothing parameter and the coefficient \( \nu \) is chosen to be positive so that the operator \( L \) is non-singular.

The LDDMM method provides a large deformation coordinate system transformation. The diffeomorphic flow may be considered to induce a distance on the space of diffeomorphisms. The transformations are enforced to be one to one, smooth and invertible. If the velocity field is not smooth then the time step in solving (2.55) must be infinitesimally small to assure diffeomorphism and the endpoint condition will never be met. Therefore, the computational cost of the LDDMM method is considerably high.
Chapter 3

The MKP Adaptive Mesh Methods

In this chapter we present two adaptive mesh methods based on solving the $L^2$ Monge-Kantorovich problem ($L^2$ MKP). In the first method, the adaptive mesh is obtained as the image of the coordinate transformation defined as the optimal mapping that minimizes the $L^2$ Kantorovich distance. The optimal mapping is computed as the steady-state solution of a parabolic Monge-Ampère equation. In the second method, the mesh redistribution is controlled by the velocity field computed as the solution of the $L^2$ MKP in a fluid dynamics framework. From now on, throughout this thesis, we refer to these two methods as the PMKP method and the FDMKP method, respectively, and we call them the MKP methods. In the next two sections we describe the details of these two methods.

3.1 The PMKP Adaptive Mesh Method

Assume that we are given a real positive adaptation function $\rho(x,t)$, $x \in \Omega \subset \mathbb{R}^d$, $t \geq 0$. The goal is to generate an adaptive mesh that is suitable for solving a time-dependent physical partial differential equation at some time $t$. To this end, let $x = x(\xi, t)$ be a coordinate transformation defined from the computational domain $\Omega_c \subset \mathbb{R}^d$ to the physical domain $\Omega \subset \mathbb{R}^d$, i.e.,

$$x : \Omega_c \rightarrow \Omega, \quad x = x(\xi, t).$$
The adaptive mesh at this fixed time is then obtained as the image of the coordinate transformation \( \mathbf{x} = \mathbf{x}(\xi, t) \).

We compute the coordinate transformation at a fixed but arbitrary time \( t \) as a solution of the \( L^2 \) Monge–Kantorovich problem constructed by first defining the initial and final densities \( \rho_0 \) and \( \rho_1 \), respectively, as

\[
\rho_0(\xi) := c_0, \quad \forall \xi \in \Omega_c, \quad (3.1)
\]
and

\[
\rho_1(\mathbf{x}) := \rho(\mathbf{x}, t), \quad \forall \mathbf{x} \in \Omega, \quad (3.2)
\]

where \( c_0 \) is some positive constant.

We seek a one to one coordinate transformation \( \phi \) defined from \( \Omega_c \) to \( \Omega \) which transports \( \rho_0 \) to \( \rho_1 \), i.e.,

\[
\int_{\phi^{-1}(A)} \rho_0(\xi) \, d\xi = \int_A \rho_1(\mathbf{x}) \, d\mathbf{x}, \quad \forall A \subset \Omega. \quad (3.3)
\]

The condition (3.3) and a change of variables lead to

\[
\rho_1(\phi(\xi)) \det(\nabla \phi(\xi)) = \rho_0(\xi), \quad (3.4)
\]

where \( \det(\nabla \phi) \) is the determinant of the Jacobian matrix of the transformation \( \phi \).

Therefore, the \( L^2 \) Monge–Kantorovich problem is the following: for the positive density functions \( \rho_0 \) and \( \rho_1 \) defined as in (3.1) and (3.2) respectively, determine the mapping \( \mathbf{x} = \phi(\xi) \) that minimizes the cost functional

\[
C(\phi) = \int_{\Omega_c} |\phi(\xi) - \xi|^2 \rho_0(\xi) \, d\xi, \quad (3.5)
\]

over all functions \( \phi \) satisfying the condition (3.3) or equivalently (3.4), i.e., transferring the density \( \rho_0(\xi), \xi \in \Omega_c \) to the density \( \rho_1(\mathbf{x}), \mathbf{x} = \phi(\xi) \in \Omega \).

The integrand in (3.5) is the Euclidean distance squared weighted by the density \( \rho_0(\xi) \). Thus, the minimization problem (3.5) gives the closest mapping to the identity in this weighted space. This feature intends to prevent mesh crossing, a necessity in adaptive grid generation.
CHAPTER 3. THE MKP ADAPTIVE MESH METHODS

In Section 2.3, we have seen that the optimal solution of the $L^2$ Monge-Kantorovich problem is unique and is characterized as the gradient of a convex function $\Psi(\xi)$, i.e.,

$$\phi(\xi) = \nabla \Psi(\xi). \quad (3.6)$$

From (3.4) and (3.6) we conclude that the potential $\Psi$ is a solution of the Monge-Ampère equation

$$\det D^2 \Psi(\xi) = \frac{\rho_0(\xi)}{\rho_1(\nabla \Psi(\xi))}, \quad (3.7)$$

where $\det D^2 \Psi$ is the determinant of the Hessian matrix of $\Psi(\xi)$ and $\nabla$ is the gradient operator with respect to $\xi$.

Therefore, the problem of computing the coordinate transformation for the mesh adaptation in higher dimensions simplifies to the problem of solving the Monge-Ampère equation (3.7) (one equation in one scalar potential) with some suitable boundary conditions. However, the Monge-Ampère equation (3.7) is fully non-linear, so the problem of developing a fast and robust numerical method is quite challenging. Oberman [Obe08] developed a wide stencil finite difference schemes to solve (3.7). The approach of [Obe08] has showed some preliminary success but its efficiency and practical implementations are still under study.

Our alternative approach for computing an approximate numerical solution to (3.7) is based on finding a steady-state solution of the logarithmic parabolic Monge-Ampère equation

$$\frac{\partial \Psi}{\partial \tau} = \log \left( \frac{\rho_1(\nabla \Psi(\xi)) \det D^2 \Psi}{\rho_0(\xi)} \right), \quad (3.8)$$

with some suitable temporal initial and spatial boundary conditions. The logarithmic term in (3.8) is introduced in order to preserve the convexity of the solution. A general form of (3.8) has been used by many authors to solve the Dirichlet problem for the Monge-Ampère equation (cf. [Tso90, Tso92, Tso91]).

If we denote by $\Psi^\infty$ the solution of (3.8) as $\tau \to \infty$, then the optimal mapping $\hat{\phi}$ is found by taking the spatial gradient of the potential $\Psi^\infty$, i.e.,

$$\hat{\phi}(\xi) = \nabla \Psi^\infty(\xi), \quad \forall \xi \in \Omega, \quad (3.9)$$

where $\nabla \Psi^\infty$ is the gradient of $\Psi^\infty$ with respect to $\xi$. Therefore, the adaptive mesh $x$ is obtained as the image of the optimal solution (3.9), i.e., $x = \hat{\phi}(\xi)$. 
3.1.1 Boundary and Initial conditions

In this subsection we discuss the boundary and initial conditions that we use in solving for the steady-state solution to (3.8).

Without loss of generality, we assume that both the computational and the physical domains $\Omega_c$ and $\Omega$ are squares (in 2D) or cubes (in 3D). In this case, we introduce the following Neumann boundary condition

$$\nabla \Psi \cdot n = \xi \cdot n, \quad \text{for } \xi \in \partial \Omega_c,$$

(3.10)

where $\partial \Omega_c$ is the boundary of $\Omega_c$ and $n$ is the outward normal to $\partial \Omega_c$. The condition (3.10) enforces the boundary points of $\Omega_c$ to be mapped to the boundary points of $\Omega$.

We solve (3.8) in conjunction with the Neumann boundary condition (3.10) and the initial condition

$$\Psi(\xi, 0) = \Psi_0(\xi) = \frac{1}{2} \xi \cdot \xi^T.$$

(3.11)

This initial condition assures that we always start from a uniform mesh.

3.1.2 The Stopping Criterion

In this subsection we describe the stopping criterion for solving the parabolic Monge-Ampère equation (3.8) to the steady-state solution.

We derive the stopping criterion based on the observation that the gradient of the solution of the Monge-Ampère equation (3.7) is unique. Therefore, if the steady-state solution of (3.8) exists then it is unique up to a constant, i.e., $\nabla \Psi^\infty$, is the unique minimizer of the cost functional (3.5). This concludes that the functional

$$C(\Psi(\xi, \tau)) = \int_{\Omega_c} |\nabla \Psi(\xi, \tau) - \xi|^2 p_0(\xi) d\xi$$

(3.12)

is decreasing in time and its minimum is achieved as $\tau \to \infty$. Thus, $\|\nabla \Psi(\xi, \tau) - \xi\|_2, \tau > 0$, changes slowly close to the steady-state solution. This means that

$$\nabla \Psi_\tau \to 0 \text{ as } \tau \to \infty,$$

(3.13)
where $\Psi_\tau$ is the partial derivative of $\Psi(\xi, \tau)$ with respect to $\tau$. Therefore, we use the following convergence criterion

$$\text{res}^n = \|\nabla \Psi^n\|_2 = \left( \int_{\Omega} |\nabla \Psi_\tau(\xi, \tau^n)|^2 \, d\xi \right)^{\frac{1}{2}} \leq \text{TOL} \quad (3.14)$$

where TOL is the user defined tolerance.

### 3.1.3 Existence and Uniqueness Results

We now discuss the global existence, uniqueness and convergence of the solution of the initial boundary value problem (3.8), (3.10), (3.11) to the solution of the Monge-Ampère equation (3.7).

Equation (3.7) is a particular case of the following general form of the Monge-Ampère equation

$$\det D^2 \Psi(\xi) = g(\xi, \Psi, \nabla \Psi(\xi)). \quad (3.15)$$

The authors in [LTU86] study the problem (3.15) for a bounded uniformly convex domain $\Omega \subset \mathbb{R}^d$ supplied with the Neumann boundary condition

$$\nabla \Psi \cdot n = f(\xi, \Psi), \quad \text{on} \quad \partial \Omega, \quad (3.16)$$

where $n$ denotes the unit outward normal vector field to $\partial \Omega$.

Existence of globally smooth convex solutions of the boundary value problem (3.15), (3.16) can be shown under suitable regularity and structure on $\Omega$, $g$, and $f$ (cf. [SS03, SS04]). In what follows we describe these results.

Consider the initial value problem

$$\begin{cases}
\frac{\partial \Psi}{\partial \tau} = \log(\det D^2 \Psi) - \log g(\xi, \Psi, \nabla \Psi) & \text{in } \Omega \times [0, T), \\
\nabla \Psi(\Omega, 0) = \Omega^* & \text{in } \Omega, \\
\Psi(\cdot, 0) = \Psi_0 & \text{in } \Omega,
\end{cases} \quad (3.17)$$

on a time interval $[0, T)$, $T > 0$, where $\Omega, \Omega^* \subset \mathbb{R}^d$, $d \geq 2$ are smooth strictly convex domains. Let $\Psi_0 : \overline{\Omega} \to \mathbb{R}$ be a smooth strictly convex function satisfying

$$\nabla \Psi_0(\Omega) = \Omega^*. \quad (3.18)$$
Moreover, assume
\[ g : \overline{\Omega} \times \mathbb{R} \times \Omega^* \rightarrow \mathbb{R} \]
to be a smooth positive function such that
\[ g_z > 0, \quad (3.19) \]
where \( z \) is the argument used for the solution \( \Psi \) and \( g_z \) is the derivative of the function \( g \) with respect to \( z \). The condition (3.18) is equivalent to \( h (\nabla \Psi) = 0 \) on \( \partial \Omega \), for a smooth strictly convex function \( \Psi \), where \( h : \mathbb{R}^d \rightarrow \mathbb{R} \) is a smooth strictly concave function such that \( h|_{\partial \Omega} = 0 \) and \( |\nabla h| = 1 \) on \( \partial \Omega^* \). Here, \textit{strictly convex function} means a function whose Hessian matrix is positive definite, and \textit{strictly convex domain} means a domain for which all principal curvatures (minimum and maximum values of the curvature) of the boundary are positive.

The existence, uniqueness, and convergence of (3.17) follow directly from the following theorem (cf. [SS03]):

**Theorem 3.1.** Assume that \( \Omega, \Omega^*, g \) and \( \Psi_0 \) are as defined above. Furthermore, assume that either
\[ \frac{g_z}{g} \geq C_g > 0, \quad (3.20) \]
or
\[ \begin{cases} \det D^2 \Psi_0 - \log g (\xi, \Psi_0, \nabla \Psi_0) \geq 0, & \text{in } \Omega, \\ \frac{d}{d\tau} h (\nabla \Psi_0)|_{\tau=0} = 0, & \text{on } \partial \Omega. \end{cases} \quad (3.21) \]
Then there exists a smooth strictly convex function, \( \Psi : \overline{\Omega} \times (0, \infty) \rightarrow \mathbb{R} \), that solves (3.17) for all times, i.e., \( T = \infty \), and \( \Psi \) converges smoothly to a solution \( \Psi^\infty \) of
\[ \begin{cases} \det D^2 \Psi^\infty (\xi) - g (\xi, \Psi^\infty, \nabla \Psi^\infty (\xi)), & \text{in } \Omega \\ \nabla \Psi^\infty (\Omega) = \Omega^* \end{cases} \quad (3.22) \]
as \( \tau \rightarrow \infty \). Furthermore, if (3.20) is satisfied then \( \Psi \) is smooth on \([0, T]\) and converges exponentially to \( \Psi^\infty \) for \( \tau \in [0, \infty) \).
In what follows we show that the solution of (3.8), defined on a square domain (for simplicity, $\Omega = [0,1]^d \subset \mathbb{R}^d, d = 2,3,\ldots$), exists and converges weakly to the unique solution of the Monge-Ampère equation (3.7).

Let $\Omega_{\epsilon}$ to be a smooth strictly convex domain obtained by modifying the boundaries of the square $\Omega$ where (see Figure 3.1)

$$\epsilon = \sup_{\xi_1,\xi_2} \{|\xi_1 - \xi_2| : \xi_1 \in \partial \Omega, \xi_2 \in \partial \Omega_{\epsilon}\},$$

where $|\cdot|$ is the standard Euclidean norm defined in $\mathbb{R}^d$.

Now define

$$\Psi^\epsilon : \mathbb{R}^d \to \mathbb{R}$$

such that

$$\nabla \Psi^\epsilon : \Omega_{\epsilon} \to \Lambda_{\epsilon}, \text{ i.e., } \nabla \Psi^\epsilon(\Omega_{\epsilon}) = \Lambda_{\epsilon},$$

where $\Lambda_{\epsilon}$ is a smooth strictly convex subset of $\mathbb{R}^d$.

**Remark 3.1.** Originally $\Psi^\epsilon$ was defined on $\Omega_{\epsilon}$; we can extend it to $\mathbb{R}^d$ by using the convention

$$\Psi^\epsilon(\xi) = \sup_{x \in \Lambda_{\epsilon}} \{\xi \cdot x - \Phi^\epsilon(x)\}, \xi \in \mathbb{R}^d,$$

where $\Phi^\epsilon$ is defined as

$$\Phi^\epsilon(x) = \sup_{\xi \in \Omega_{\epsilon}} \{\xi \cdot x - \Psi^\epsilon(\xi)\}, x \in \mathbb{R}^d.$$

Let $\Psi^\epsilon$ be a solution to the initial-boundary value problem (3.8), (3.10), (3.11) for all time $\tau$ and suppose that the conditions (3.20) and (3.21) are satisfied for the function $g$ defined as the right hand side of (3.7) for the smooth strictly convex domains $\Omega_{\epsilon}$ and $\Lambda_{\epsilon}$. Therefore, from Theorem 3.1 $\Psi^\epsilon$ converges smoothly to the unique solution of

$$\begin{cases}
\rho_1'(\nabla \Psi^\epsilon(\xi)) \det D^2 \Psi^\epsilon = \rho_0'(\xi), & \text{in } \Omega,
\nabla \Psi^\epsilon(\Omega_{\epsilon}) = \Lambda_{\epsilon}.
\end{cases} \quad (3.23)$$
Figure 3.1: Sketch of a smooth strictly convex domain $\Omega_\epsilon$ and a square domain $\Omega$ for the 2D case.

**Theorem 3.2.** Suppose that $\rho'_\epsilon \rightarrow \rho_1$, $\rho'_0 \rightarrow \rho_0$ weakly in $L^1(\mathbb{R}^d)$ and the support of $\rho'_1, \rho'_0 \subseteq B_R$, a subset of a disk of radius $R$, and let $\Psi^\epsilon$ and $\Psi$ be as defined in (3.23) and (3.7). If $0 \in \Omega, 0 \in \Omega_\epsilon, \forall \epsilon$, then as $\epsilon \rightarrow 0$,

$$\nabla \Psi^\epsilon \rightharpoonup \nabla \Psi$$

uniquely in $W^{1,p}_{\text{loc}}(\mathbb{R}^d)$, $\Psi'(0) = 0 = \Psi(0)$, \hspace{1cm} (3.24)

for $1 \leq p < \infty$.

The proof of Theorem 3.2 is a direct consequence of the following facts.

**Theorem 3.3.** If $\Psi^\epsilon : \mathbb{R}^d \rightarrow \mathbb{R}$ are convex and for each compact set $K$,

$$\int_K |\Psi^\epsilon| \leq C_K$$

where $C_K$ is a constant depending only on $K$, then there exist $\Psi : \mathbb{R}^d \rightarrow \mathbb{R}$ convex such that $|\nabla \Psi| \leq M_K$ on each $K$ and $\Psi^\epsilon \rightarrow \Psi$ in $W^{1,p}_{\text{loc}}(\mathbb{R}^d)$ for a subsequence $\{\Psi^\epsilon_n\}$. 

\qed
Proof: To prove this result, it suffices to bound
\[ \int_K |\nabla \Psi^\epsilon| \, dx \quad \text{and} \quad \int_K |\nabla^2 \Psi^\epsilon| \, dx \]

The first bound is given by theorem 1 (pages 144-145) of the book by Evans-Gariepy [EG92]. In fact, that theorem gives the stronger result:
\[ |\Psi'|, |\nabla \Psi^\epsilon| \leq C_K. \quad (3.25) \]

By (3.25) and the Ascoli-Arzela theorem (e.g. see [Dsh01]), up to a subsequence, \( \Psi^\epsilon \) converges uniformly to a convex function \( \Psi \) and \( \nabla \Psi^\epsilon \) converges weakly to \( \nabla \Psi \).

Next, let \( a_{\epsilon,1} \cdots a_{\epsilon,d} \) be the eigenvalues of \( D^2 \Psi^\epsilon \). Then
\[ |D^2 \Psi^\epsilon|^2 = a_{\epsilon,1}^2 + \cdots + a_{\epsilon,d}^2 \]
and
\[ \Delta \Psi^\epsilon = a_{\epsilon,1} + \cdots + a_{\epsilon,d}, \]
where \( D^2 \) is the Hessian matrix and \( \Delta \) is the Laplacian operator. Because \( \Psi^\epsilon \) is convex, the eigenvalues are nonnegative. Thus,
\[ |D^2 \Psi^\epsilon| \leq \sqrt{d} \Delta \Psi^\epsilon. \quad (3.26) \]

But if \( B \) is a ball then by the divergence theorem
\[ \int_B \Delta \Psi^\epsilon = \int_{\partial B} \nabla \Psi^\epsilon \cdot n \, d\sigma \leq \int_{\partial B} |\nabla \Psi^\epsilon| \, d\sigma \leq C_K \sigma(\partial B). \quad (3.27) \]
To obtain the last inequality, we have used (3.25). Here \( n \) is the unit outward normal to the boundary \( \partial B \). By (3.26) and (3.27), we have
\[ \int_K |D^2 \Psi^\epsilon| \leq \sqrt{d} C_K \sigma(\partial B). \quad (3.28) \]
Using (3.25) and (3.28), we have proven that \( \nabla \Psi^\epsilon \) is bounded in \( BV(K) \) (functions of bounded variation) for each compact set. By theorem 4 (page 176) of the book by Evans-Gariepy [EG92], up to a subsequence, we have that \( \nabla \Psi^\epsilon \) converges in \( L^1(K) \) to \( \nabla \Psi \). Because by (3.25) \( \nabla \Psi^\epsilon \) is bounded in \( L^\infty(K) \) we conclude that up to that subsequence, \( \nabla \Psi^\epsilon \) converges in \( L^p(K) \) to \( \nabla \Psi \).
CHAPTER 3. THE MKP ADAPTIVE MESH METHODS

Remark 3.2. The uniqueness of the solution of the Monge-Ampère equation (3.7) implies that the whole sequence \( \{ \Psi^\epsilon \} \) converges to \( \Psi \).

Proof of the remark: Assume the converse, i.e., suppose that there exists some positive number \( \epsilon_0 \) such that

\[
\forall \epsilon \leq \epsilon_0, \| \Psi^\epsilon - \Psi \|_{W^{1,p}_loc(Q)} \geq \epsilon_0, \quad Q \subset \mathbb{R}^d.
\]

This contradicts the fact that there exists a subsequence \( \{ \Psi^{\epsilon_n} \} \) that converges to the unique solution \( \Psi \).

\[\Box\]

3.1.4 Implementation of the PMKP Method

In this subsection we describe the implementation of the PMKP method and the computational details of solving the initial boundary value problem (3.8), (3.10) and (3.11). We describe these details for 2D; the 3D case follows similarly.

For the 2D case, for simplicity let \( \Omega_c = \Omega = [0,1]^2 \), we consider the Neumann boundary condition

\[
\frac{\partial \Psi}{\partial \xi} = 0, \quad \text{on} \ \xi = 0, 1, \quad \frac{\partial \Psi}{\partial \eta} = 1, \quad \text{on} \ \eta = 0, 1, \quad (3.29)
\]

and the initial condition

\[
\Psi(\xi, 0) = \frac{1}{2} (\xi^2 + \eta^2). \quad (3.30)
\]

The boundary conditions (3.29) follow directly from (3.10).

We discretize (3.8) on the unit square domain \( \Omega_c \) with a uniform mesh of size \( M \times N \). To this end, define

\[
\Delta \xi = \frac{1}{M - 1}, \quad \Delta \eta = \frac{1}{N - 1}, \quad i = 1, 2, \ldots, \ M, \quad j = 1, 2, \ldots, \ N, \quad (3.31)
\]

\[
\xi_i = (i - 1) \Delta \xi, \quad \eta_j = (j - 1) \Delta \eta, \quad (3.32)
\]
CHAPTER 3. THE MKP ADAPTIVE MESH METHODS

and
\[ \tau^n = n \Delta \tau, \quad \Delta \tau > 0, \quad \Psi^n_{ij} = \Psi(\xi_i, \eta_j, \tau^n), \quad x^n_{ij} = \nabla \Psi^n_{ij}, \]

where the variable \( \tau \) denotes the artificial time used during the computation of the steady-state solution, and \( \nabla \) is the gradient operator with respect to the computational domain variable at \((\xi_i, \eta_j)\).

Then, for the spatial discretization, we use standard centered finite difference approximations for the second-order derivatives in (3.8) on a nine point stencil (see Figure 3.2). After the spatial discretization, we integrate the resulting system of ordinary differential equations using an explicit scheme, for example, first order forward-Euler or a higher-order Runge-Kutta scheme.

Let the numerical solution \( U^q \) of the physical problem and the adaptive mesh \( \hat{x}^q_{ij} \) at time \( t = t^q, \quad q \in \mathbb{N} \), be given. Here, \( t \) denotes the real time variable for the physical problem. We can then use the PMKP method to compute the adaptive mesh for the next time level \( \hat{x}^{q+1}_{ij} \) (at time \( t = t^{q+1} \)) by first computing the adaptation function, for instance,

\[ \rho(\hat{x}^q_{ij}, t^q) = \sqrt{1 + |\nabla U^q_{ij}|^2}, \quad \text{(3.33)} \]
where $\nabla$ is the gradient operator with respect to the physical domain variables.

Define the density function $\rho_1$ as

$$\rho_1(x) = \rho(x, t^0), \quad \forall x \in \Omega,$$  \hspace{1cm} (3.34)

and define the initial density as

$$\rho_0(\xi_{ij}) = c_0, \quad c_0 > 0, \quad \forall \xi_{ij} \in \Omega_c.$$  \hspace{1cm} (3.35)

We then compute the optimal transformation that transports $\rho_0$ to $\rho_1$ as the spatial gradient of the steady-state solution of the initial boundary value problem (3.8), (3.29), (3.30).

The procedure we employ to solve for the steady-state solution and then compute the adaptive mesh $\bar{x}^{q+1}$ is summarized in the following steps:

1. Given $\Psi_{ij}^n$ and $x_{ij}^n$
2. Compute $\rho_1^n(x_{ij}^n)$ using (3.33)
3. Compute $\det D^2\Psi_{ij}^n$
4. Set $\rho_1 = \sqrt{1 + |\nabla U_{ij}^n|^2}$, $\rho_0 = c_0$
5. Set $F_{ij}^n = \log \left( \frac{\rho_1(x_{ij}^n) \det D^2\Psi_{ij}^n}{\rho_0(\xi_{ij})} \right)$
6. Update $\Psi_{ij}^{n+1}$ with the forward Euler scheme $\Psi_{ij}^{n+1} = \Psi_{ij}^n + \Delta \tau F_{ij}^n$
7. Compute $x_{ij}^{n+1} = \nabla \Psi_{ij}^{n+1}$
8. If $||\nabla \partial_\tau \Psi_{ij}^{n+1}||_2 \leq \text{TOL}$ go to 9, else set $n = n + 1$ and go to 2
9. Set $\bar{x}_{ij}^{q+1} = x_{ij}^{n+1}$ and stop.

Here $\partial_\tau$ denotes the time derivative with respect to $\tau$, TOL is the user supplied tolerance, $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$. 
3.2 The FDMKP Adaptive Mesh Method

In the previous section we computed the adaptive mesh by solving directly for the coordinate transformation. In this section, we describe the FDMKP method to compute the adaptive mesh. For the FDMKP method, the mesh adaptation is achieved by employing a velocity field to control the mesh movement. The velocity field is computed by solving the $L^2$ Monge–Kantorovich problem in a fluid dynamics framework. Thus, the FDMKP method is a velocity based method, and it is closely related to the GCL method of [CHR02] discussed in Subsection 2.2.1.

3.2.1 Implementation of FDMKP Method

In this subsection, we explain the procedure for computing adaptive mesh by the FDMK method.

Let $\Omega$, $\Omega_c \subset \mathbb{R}^d, d = 2$ or $3$, be the physical and the computation domains, respectively. Consider the problem of generating an adaptive mesh on the physical domain $\Omega$ for a given adaptation function.

Given the physical solution at time $t = t^p$, $p \in \mathbb{N}$, we would like to determine the adaptive mesh $x^{p+1}$ at the next time level $t = t^{p+1}$. To this end, we first compute $\rho_0$ and $\rho_1$ using (3.35) and (3.34), respectively. Then, fix a time interval $E [0, T], T > 0$; for simplicity we take $T = 1$. Then, consider all possible smooth enough time dependent velocity and density fields $v(\xi, \tau)$ and $\rho(\xi, \tau) > 0$, for $\xi \in \Omega_c$ and $\tau \in [0, 1]$, that satisfy

$$\frac{\partial \rho}{\partial \tau} + \nabla \cdot (\rho v) = 0, \quad (3.36)$$

and

$$\rho(\cdot, 0) = \rho_0, \quad \rho(\cdot, 1) = \rho_1. \quad (3.37)$$

In this formulation, the transport of $\rho_0$ to $\rho_1$ is controlled by the velocity field $v(\xi, \tau), \tau \in [0, 1]$. If we were to follow the motion of a point $\xi \in \Omega_c$, then it now follows a time-dependent trajectory $\phi(\xi, \tau), \tau \in [0, 1]$, and the final position of this point is $x = \phi(\xi, 1)$ in the physical domain $\Omega$. 
The goal is to compute the velocity and density fields $\dot{v}(\xi, \tau)$ and $\dot{\rho}(\xi, \tau)$, respectively, which solve the minimization problem

$$\inf_{\dot{v}, \dot{\rho}} \int_0^1 \int_\Omega \rho(\xi, \tau) |v(\xi, \tau)|^2 \, d\xi \, d\tau,$$

(3.38)

over all $v(\xi, \tau)$ and $\rho(\xi, \tau)$ that satisfy the conditions (3.36) and (3.37).

The adaptive mesh $x^{p+1}$ is obtained by defining the flow $\phi(\xi, \tau)$ which is computed by time integration of the evolution equation

$$\frac{d\phi}{d\tau} = \dot{v}(\xi, \tau), \quad \phi(\xi, 0) = \xi, \quad \xi \in \Omega_c$$

(3.39)

from $\tau = 0$ to $\tau = 1$.

We summarize the steps of computing the adaptive mesh at the time level $t = t^{p+1}$ using the FDMKP method as follows:

1. Given the physical solution $U^p$, compute $\rho_0$ and $\rho_1$ using (3.35) and (3.34).
2. Compute the optimal velocity field $\dot{v}(\xi, \tau)$ for $\xi \in \Omega_c, \tau \in [0, 1]$ by solving the minimization problem (3.38).
3. Compute $\phi(\xi, \tau)$ by solving the initial value problem (3.39) for $\tau = 0$ to $\tau = 1$.
4. Compute the adaptive mesh by setting

$$x^{p+1} = \phi(\xi, 1).$$

### 3.2.2 The Saddle Point Problem Formulation

In Section 2.3, we have seen that the optimality conditions for the minimization problem (3.38) can be written as

$$v(\xi, \tau) = \nabla \Phi(\xi, \tau),$$

(3.40)

where the potential $\Phi$ solves the Hamilton-Jacobi equation

$$\partial_\tau \Phi + \frac{1}{2} |\nabla \Phi|^2 = 0.$$
We use the augmented Lagrangian method (cf. [BB00]) to compute the numerical solution of the minimization problem (3.38) with the constraints (3.36) and (3.37).

The Lagrangian for the minimization problem (3.38) with the constraints (3.36) and (3.37) can be written as

\[
L(\Phi, \rho, m) = \int_0^1 \int_\Omega \left( \frac{|m|^2}{2\rho} - \frac{\partial_x \Phi}{\rho} - \nabla \Phi \cdot m \right) d\xi d\tau - \int_\Omega (\Phi(\cdot, 0) \rho_0 - \Phi(\cdot, 1) \rho_1) d\xi,
\]

where \( \rho = \rho(\xi, \tau), m = \rho(\xi, \tau) v(\xi, \tau) \), \( \nabla \) is the gradient with respect to the spatial variables \( \xi \) and \( \Phi(\xi, \tau) \) is space-time dependent Lagrange multiplier for the constraints (3.36) and (3.37).

Therefore, solving the \( L^2 \) Monge–Kantorovich problem in the fluid dynamics framework is equivalent to solving the saddle-point problem

\[
\inf_{\rho, m} \sup_{\Phi} L(\Phi, \rho, m).
\]  

3.2.3 The Saddle Point Problem Solution Procedure

In this subsection we briefly discuss the numerical details and the solution procedure for the saddle-point problem (3.43).

Introduce the notation (adopted from [BB00])

\[
\mu = \{\rho, m\}, \quad q = \{a, b\}, \quad \langle \mu, q \rangle = \int_0^1 \int_\Omega \mu \cdot q,
\]

\[
F(q) = \begin{cases} 
0 & \text{if } q \in K, \\
+\infty & \text{else,}
\end{cases} \quad \text{and} \quad G(\Phi) = \int_\Omega (\Phi(\cdot, 0) \rho_0 - \Phi(\cdot, 1) \rho_1) d\xi d\tau,
\]

where

\[
K = \left\{ \{a, b\} : \mathbb{R} \times \mathbb{R}^d, \text{ s.t. } a + \frac{|b|^2}{2} \leq 0 \right\}.
\]

The problem (3.43) is equivalent to the new saddle-point problem

\[
\sup_{\mu} \inf_{\Phi, q} L_r(\Phi, q, \mu),
\]
where $L_r$ is the augmented Lagrangian defined as,

$$L_r(\Phi, q, \mu) = F(q) + G(\Phi) + \langle \mu, \nabla_{\xi, \tau} \Phi - q \rangle + \frac{r}{2} \langle \nabla_{\xi, \tau} \Phi - q, \nabla_{\xi, \tau} - q \rangle,$$

(3.48)

where $r$ is a positive parameter and $\nabla_{\xi, \tau}$ is the space-time gradient operator defined as

$$\nabla_{\xi, \tau} = \partial_r + \nabla,$$

in which $\partial_r$ is the temporal derivative operator and $\nabla$ is the spatial gradient operator with respect to $\xi$.

The theoretical existence of the saddle-point for the Lagrangian (3.43) or (3.47) is described in [Gui03].

Recasting (3.43) as (3.47) allows it to be solved by the ALG2 algorithm of [FG83]. This algorithm is based on a relaxation of the Uzawa algorithm.

**ALG2 Algorithm:**

1. Given $\Phi^{n-1}$, $q^{n-1}$, $\mu^n$.

2. Find $\Phi^n$ such that

$$L_r(\Phi^n, q^{n-1}, \mu^n) \leq L_r(\Phi, q^{n-1}, \mu^n), \quad \forall \Phi$$

(3.49)

3. Find $q^n$ such that

$$L_r(\Phi^n, q^n, \mu^n) \leq L_r(\Phi^n, q, \mu^n), \quad \forall q$$

(3.50)

4. Update $\mu$ using

$$\mu^{n+1} = \mu^n + r(\nabla_{\xi, \tau} \Phi^n - q^n),$$

(3.51)

(where $r > 0$ is a parameter of the augmented Lagrangian).

5. Go to step 2.
To find $\Phi^n$ in step 2 differentiate $L_r$ with respect to $\Phi$ and then integrate by parts in space and time to obtain

$$-\tau \triangle \xi_\tau \Phi^n = \nabla \xi_\tau \cdot (\mu^n - r \eta^{n-1}),$$

(3.52)

with periodic boundary conditions in space and Neumann boundary conditions in time, i.e.,

$$r \partial_\tau \Phi^n(\cdot, 0) = \rho_0 - \rho^n(\cdot, 0) + \tau a^{n-1}(\cdot, 0),$$

(3.53)

$$r \partial_\tau \Phi^n(\cdot, 1) = \rho_1 - \rho^n(\cdot, 1) + \tau a^{n-1}(\cdot, 1).$$

(3.54)

Finding $\eta^n$ in step 3 is equivalent to solving for $(a, b)$ as

$$\inf \left\{ (a - a^n(\xi, \tau))^2 + |b - b^n(\xi, \tau)|^2, a + \frac{|b|^2}{2} \leq 0 \right\},$$

(3.55)

where

$$a^n(\xi, \tau) = \frac{\partial^2 \Phi^n}{\partial \tau^2} + \frac{\rho^n}{r}, \quad b^n(\xi, \tau) = \nabla \Phi^n + \frac{m}{r}.$$

The convergence of the ALG2 algorithm described above is shown in [Gui03].

The convergence criterion can be obtained using the optimality conditions (3.40) and (3.41). Precisely speaking, we define the convergence criterion in terms of the residual of the Hamilton-Jacobi equation (3.41)

$$\text{res}^n = \partial_\tau \Phi^n + \frac{1}{2} |\nabla \Phi^n|^2.$$

Notice that the residual $\text{res}^n$ converges towards 0 as we approach the optimal solution of the Monge–Kantorovich problem. In practice, to stop the computations we use the normalized convergence criterion

$$\text{err}^n = \sqrt{\frac{\int_0^1 \int_\Omega \rho^n |\text{res}^n|}{\int_0^1 \int_\Omega \rho^n |\nabla \Phi^n|^2}} < \text{TOL},$$

(3.56)

where TOL is the user defined convergence tolerance.

For a discretization with $N$ total points in the space-time domain the cost of computations involved in steps 3 and 4 in the ALG2 are of order $O(N)$. The Poisson’s
equation (3.52) derived in step 2 can be solved at a cost of $O(N \log N)$. Therefore, the total cost of this method is $O(nN \log N)$; $n$ is the number of iterations required for convergence.

Typically, in two spatial dimensions, if the spatial mesh is of size $n_\xi \times n_\eta$ and the number of points in time is $n_\tau$, then total number of points in the space-time lattice (for simplicity we take it to be defined on the unit cube $[0,1]^3$) is $N = n_\xi \cdot n_\eta \cdot n_\tau$. The space-time Poisson's equation (3.52) can be approximated using the five-point approximation to the Laplacian

$$
\nabla^2_{\xi,\tau} \Phi \approx \nabla_{ijk} \Phi = \frac{\Phi_{i+1,j,k} - 2\Phi_{i,j,k} + \Phi_{i-1,j,k}}{\Delta \xi^2} + \frac{\Phi_{i,j+1,k} - 2\Phi_{i,j,k} + \Phi_{i,j,k-1}}{\Delta \eta^2} + \frac{\Phi_{i,j,k+1} - 2\Phi_{i,j,k} + \Phi_{i,j,k-1}}{\Delta \tau^2},
$$

(3.57)

where

$$
\Delta \tau = 1/(n_\tau - 1), \Delta \xi = 1/(n_\xi - 1), \Delta \eta = 1/(n_\eta - 1)
$$

and

$$
\Phi_{i,j,k} = \Phi(\xi_i, \eta_j, \tau_k), \xi_i = (i - 1)\Delta \xi, \eta_j = (j - 1)\delta_j, \tau_k = (k - 1)\Delta \tau_k,
$$

for $i = 1, 2, \ldots, n_\xi$, $j = 1, 2, \ldots, n_\eta$ and $k = 1, 2, \ldots, n_\tau$.

Taylor-expansion of $\nabla_{ijk} \Phi$ about $(\xi_i, \eta_j, \tau_k)$ gives

$$
\nabla^2_{\xi,\tau} \Phi = \nabla_{ijk} \Phi - \frac{1}{12} \left( \Delta \xi^2 \frac{\partial^4 \Phi}{\partial \xi^4} + \Delta \eta^2 \frac{\partial^4 \Phi}{\partial \eta^4} + \Delta \tau^2 \frac{\partial^4 \Phi}{\partial \tau^4} \right) + \text{h.o.t.}
$$

(3.58)

Therefore, the error in solving (3.52) is of $O(\Delta \xi^2 + \Delta \eta^2 + \Delta \tau^2)$. In our computations, we choose $n_\tau = n_\xi = n_\eta$.

### 3.3 Solution Procedure for the Physical Problem

In this section we explain in detail the steps to compute a numerical solution of a two-dimensional time dependent partial differential equation using the MKP adaptive mesh methods.

Consider the time dependent physical PDE

$$
\frac{\partial u}{\partial t} = f(t, x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}), \quad (x, y) \in \Omega, \quad t > 0
$$

(3.59)
with the initial condition

\[ u(x, y, 0) = u_0(x, y) \]

and some appropriate boundary conditions in space.

Let \( \Omega \) denote the domain of the physical variables \((x, y)\), and let \( \Omega_c \) denote the domain of the computational domain variables \((\xi, \eta)\).

In what follows, we describe the procedure used to solve the time dependent PDE (3.59) using the PMKP or the FDMKP method.

To apply the MKP methods to solve (3.59), we first rewrite the PDE (3.59) in terms of the computational domain variables by first rewriting (3.59) in the form

\[ \dot{u} - u_x \dot{x} - u_y \dot{y} = f(t, x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}), \quad (3.60) \]

and then write the derivatives \( u_x, u_y, u_{xx}, u_{xy}, u_{yy} \) in terms of the computational domain variables \( \xi, \eta \) (see [HR99] for more details) as follows:

\[
\begin{align*}
    u_x &= \frac{1}{J} \left[ + \left( y_\eta u_\xi - (y_\xi u_\eta) \right) \xi - \left( y_\xi y_\eta u_\eta \right) \xi / J + \left( y_\xi y_\xi u_\eta \right) \eta / J \right], \\
    u_y &= \frac{1}{J} \left[ - \left( x_\eta u_\xi + (x_\xi u_\eta) \right) \xi + \left( x_\xi x_\eta u_\eta \right) \xi / J - \left( x_\xi x_\xi u_\eta \right) \eta / J \right], \\
    u_{xx} &= \frac{1}{J} \left[ + \left( y_\eta^2 u_\xi \xi / J + \left( y_\xi y_\eta^2 u_\eta \right) \xi / J \right) \xi - \left( y_\xi y_\eta u_\eta \right) \xi / J + \left( y_\xi^2 u_\eta \right) \eta / J \right], \\
    u_{xy} &= \frac{1}{J} \left[ - \left( x_\eta y_\eta u_\xi \xi / J + \left( x_\xi y_\eta^2 u_\eta \right) \xi / J \right) \xi + \left( x_\xi y_\xi u_\eta \right) \xi / J - \left( x_\xi y_\xi y_\eta \right) \eta / J \right], \\
    u_{yy} &= \frac{1}{J} \left[ + \left( x_\eta^2 u_\xi \xi / J + \left( x_\xi x_\eta u_\eta \right) \xi / J \right) \xi - \left( x_\xi x_\eta u_\eta \right) \xi / J + \left( x_\xi^2 u_\eta \right) \eta / J \right], 
\end{align*}
\]

where \( J \) is the determinant of the Jacobian of the coordinate transformation defined as

\[ J = x_\xi y_\eta - x_\eta y_\xi. \]

### 3.3.1 Finite Difference Scheme

In this subsection, we illustrate the details of the spatial discretization of the physical problem (3.60) for the first and second order derivative terms as described in [HR99].
For convenience, let the computational domain $\Omega_c$ to be the unit square. Let

$$x_{ij} = x(\xi_i, \eta_j), \quad y_{ij} = y(\xi_i, \eta_j), \quad u_{ij} = u(\xi_i, \eta_j),$$

(3.66)

where $\xi_i, \eta_j$ are as given in (3.31), $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$ for some positive integers $M$ and $N$.

Using a nine point stencil (see figure 3.2), the forms (3.61), (3.62), (3.63), (3.64) and (3.65) for the derivatives $u_x$, $u_y$, $u_{xx}$, $u_{xy}$ and $u_{yy}$, respectively, are approximated with finite differences in the computational domain $\Omega_c$. These terms are approximated at the point $(\xi_i, \eta_j)$ by the central finite differences

$$\frac{\partial}{\partial \xi} [y_\eta u_\xi] \approx y_{\eta,i+\frac{1}{2},j} u_{i+\frac{1}{2},j} - y_{\eta,i-\frac{1}{2},j} u_{i-\frac{1}{2},j},$$

(3.67)

where

$$u_{i\pm\frac{1}{2},j} = \frac{1}{2} (u_{i\pm1,j} + u_{i,j}),$$

$$y_{\eta,i\pm\frac{1}{2},j} = \frac{1}{4\Delta \xi \Delta \eta} (y_{\eta,i+1,j+1} - y_{\eta,i+1,j-1} + y_{\eta,i,j+1} - y_{\eta,i,j-1}),$$

(3.68)

where

$$u_{i,j\pm\frac{1}{2}} = \frac{1}{2} (u_{i\pm1,j} + u_{i,j}),$$

$$y_{\xi,i,j\pm\frac{1}{2}} = \frac{1}{4\Delta \xi \Delta \eta} (y_{\xi,i+1,j\pm1} - y_{\xi,i-1,j\pm1} + y_{\xi,i+1,j} - y_{\xi,i-1,j}).$$

The terms $x_\eta u_\xi$ and $x_\xi u_\eta$ are similarly discretized. The terms in the right hand side of (3.63), used to compute $u_{xx}$, are discretized as follows:

$$\frac{\partial}{\partial \xi} \left[ \frac{y_\eta^2 u_\xi}{J} \right] \bigg|_{(\xi_i, \eta_j)} = \frac{1}{\Delta \xi} \left[ \frac{y_{\eta,i+\frac{1}{2},j} u_{\xi,i+\frac{1}{2},j}}{J_{i+\frac{1}{2},j}} - \frac{y_{\eta,i-\frac{1}{2},j} u_{\xi,i-\frac{1}{2},j}}{J_{i-\frac{1}{2},j}} \right],$$

(3.69)

$$\frac{\partial}{\partial \xi} \left[ \frac{y_\xi y_\eta u_\eta}{J} \right] \bigg|_{(\xi_i, \eta_j)} = \frac{1}{\Delta \xi} \left[ \frac{y_{\xi,i+\frac{1}{2},j} y_{\eta,i+\frac{1}{2},j} u_{\eta,i+\frac{1}{2},j}}{J_{i+\frac{1}{2},j}} - \frac{y_{\xi,i-\frac{1}{2},j} y_{\eta,i-\frac{1}{2},j} u_{\eta,i-\frac{1}{2},j}}{J_{i-\frac{1}{2},j}} \right].$$

(3.70)
CHAPTER 3. THE MKP ADAPTIVE MESH METHODS

\[
\frac{\partial}{\partial \eta} \left[ \frac{y_\xi y_\eta u_\xi}{J} \right]_{(\xi_i, \eta_j)} = \frac{1}{\Delta \eta} \left[ \frac{y_{\xi,i,j+\frac{1}{2}} y_{\eta,i,j+\frac{1}{2}} u_{\xi,i,j+\frac{1}{2}}}{J_{i,j+\frac{1}{2}}} - \frac{y_{\xi,i,j}\ y_{\eta,i,j}}{J_{i,j-\frac{1}{2}}} \right], 
\]

\[
\frac{\partial}{\partial \eta} \left[ \frac{y_\eta^2 u_\eta}{J} \right]_{(\xi_i, \eta_j)} = \frac{1}{\Delta \eta} \left[ \frac{y_{\eta,i,j+\frac{1}{2}} u_{\eta,i,j+\frac{1}{2}}}{J_{i,j+\frac{1}{2}}} - \frac{y_{\eta,i,j-\frac{1}{2}} u_{\eta,i,j-\frac{1}{2}}}{J_{i,j-\frac{1}{2}}} \right],
\]

where

\[
x_{\xi,i+\frac{1}{2},j} = \frac{x_{i+1,j} - x_{i,j}}{\Delta \xi}, \quad x_{\xi,i-\frac{1}{2},j} = \frac{x_{i,j} - x_{i-1,j}}{\Delta \xi},
\]

\[
y_{\eta,i,j+\frac{1}{2}} = \frac{y_{i,j+1} - y_{i,j}}{\Delta \eta}, \quad y_{\eta,i,j-\frac{1}{2}} = \frac{y_{i,j} - y_{i,j-1}}{\Delta \eta},
\]

\[
u_{\xi,i+\frac{1}{2},j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta \xi}, \quad u_{\xi,i-\frac{1}{2},j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta \xi},
\]

\[
u_{\eta,i,j+\frac{1}{2}} = \frac{u_{i,j+1} - u_{i,j}}{\Delta \eta}, \quad u_{\eta,i,j-\frac{1}{2}} = \frac{u_{i,j} - u_{i,j-1}}{\Delta \eta},
\]

\[
J_{i+\frac{1}{2},j} = \frac{1}{4\Delta \xi \Delta \eta} \left[ \left( x_{i+1,j} - x_{i,j} \right) \left( y_{i,j+1} - y_{i,j-1} + y_{i+1,j+1} - y_{i+1,j-1} \right) 
- \left( y_{i+1,j} - y_{i,j} \right) \left( x_{i,j+1} - x_{i,j-1} + x_{i+1,j+1} - x_{i+1,j-1} \right) \right],
\]

\[
J_{i-\frac{1}{2},j} = \frac{1}{4\Delta \xi \Delta \eta} \left[ \left( x_{i,j} - x_{i-1,j} \right) \left( y_{i,j+1} - y_{i,j-1} + y_{i-1,j+1} - y_{i-1,j-1} \right) 
- \left( y_{i,j} - y_{i-1,j} \right) \left( x_{i,j+1} - x_{i,j-1} + x_{i-1,j+1} - x_{i-1,j-1} \right) \right],
\]

\[
J_{i+\frac{1}{2},j} = \frac{1}{4\Delta \xi \Delta \eta} \left[ \left( x_{i+1,j} - x_{i-1,j} \right) \left( x_{i,j+1} + x_{i+1,j+1} - x_{i-1,j+1} \right) 
- \left( y_{i+1,j} - y_{i-1,j} \right) \left( x_{i,j+1} - x_{i,j} \right) \right],
\]

\[
J_{i-\frac{1}{2},j} = \frac{1}{4\Delta \xi \Delta \eta} \left[ \left( x_{i,j} - x_{i-1,j} \right) \left( x_{i,j+1} + x_{i+1,j+1} - x_{i-1,j+1} \right) 
- \left( y_{i+1,j} - y_{i-1,j} \right) \left( x_{i,j+1} - x_{i,j} \right) \right] .
\]
CHAPTER 3. THE MKP ADAPTIVE MESH METHODS

We notice that the above approximations for the derivatives with respect to \(\xi\) and \(\eta\) are of the forms

\[
\frac{\partial F}{\partial \xi} = \frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta \xi}
\]

and

\[
\frac{\partial F}{\partial \eta} = \frac{F_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}}{\Delta \eta}
\]

In what follows we derive the truncation error for the expression (3.73).

Using Taylor-expansion about \((\xi, \eta)\) we can write the terms in the right hand side of (3.73) as follows:

\[
F_{i+\frac{1}{2},j} = F_{i,j} + \frac{\Delta \xi}{2} F_{\xi,i,j} + \frac{\Delta \xi^2}{8} F_{\xi\xi,i,j} + \frac{\Delta \xi^3}{48} F_{\xi\xi\xi,i,j} + \cdots,
\]

\[
F_{i-\frac{1}{2},j} = F_{i,j} - \frac{\Delta \xi}{2} F_{\xi,i,j} + \frac{\Delta \xi^2}{8} F_{\xi\xi,i,j} - \frac{\Delta \xi^3}{48} F_{\xi\xi\xi,i,j} + \cdots,
\]

which leads to

\[
\frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta \xi} = F_{\xi,i,j} + \frac{1}{24} F_{\xi\xi\xi,i,j} \Delta \xi^2 + \text{h.o.t.}
\]

Therefore, the truncation error for the expression (3.73) is \(O(\Delta \xi^2)\). Similarly the truncation error for the expression (3.74) can be derived.

We use the method of lines for the numerical solution of the physical PDE. First, discretize the system (3.60) in space using central finite differences and then integrate the resulting ODE system of equations for one time level.

For the mesh adaptation we use the arc-length adaptation function

\[
\rho(x, y) = \sqrt{1 + |\nabla x u|^2},
\]

and the adaptation function

\[
\rho(\xi, \eta) = \sqrt{1 + \alpha u^2 + \beta |\nabla \xi u|^2},
\]

where \(\alpha, \beta\) are scalars, \(\nabla x = (\partial_x, \partial_y)\), and \(\nabla \xi = (\partial_{\xi}, \partial_{\eta})\). Notice that the adaptation function (3.79), unlike (3.78), is defined in term of the computational domain variables. The adaptation function (3.79) was used by Ceniceros and Hou [CH01] to generate adaptive mesh for potentially singular solutions.
CHAPTER 3. THE MKP ADAPTIVE MESH METHODS

In standard moving mesh methods one solves the physical problem and the mesh PDEs simultaneously or alternately for the physical solution and the mesh. In this thesis we solve these equations alternately. Specifically, if we are given the mesh $x^n$ and the physical solution $u^n$ at time level $n$, then to obtain the solution $u^{n+1}$ at time level $n + 1$ we proceed as follows:

1. Compute the adaptation function $\rho$ at time step $n$ using (3.78) or (3.79)
2. Compute the adaptive mesh using the PMKP or FDMKP method
3. Compute $u^{n+1}$ by integrating the ODE system obtained from the spatial discretization of (3.60)
4. Set $n = n+1$ and go to 1.
Chapter 4

Numerical Experiments

In this chapter we present two and three dimensional numerical results to demonstrate the performance and various features of the PMKP and the FDMKP methods. More precisely, we give several numerical examples to illustrate the properties and the quality measure of the adaptive meshes generated using these methods. Some examples are included to compare the MKP methods with the GCL method and the PMA method. We consider both cases where the adaptation function is given as an analytic function and for some real applications. Here, we apply the MKP methods to find a numerical solution for the two-dimensional Burgers' equation.

In two spatial dimension, we use the function

\[ E_{adp}(x, y, t) = \rho(x, y, t) \det D^2 \Psi(\xi, \eta, t) |\Omega_c| \]  \hspace{1cm} (4.1)

as a measure of how close the distribution of the obtained mesh comes to equidistributing the adaptation function \( \rho(x, y, t) \). Here, \( |\Omega_c| \) is the area of the computational domain \( \Omega_c \). A good adaptive mesh is obtained as the surface of the function \( E_{adp}(x, y, t) \) is close to 1. Roughly speaking, we find that a well equidistributed mesh is obtained for the surface \( E_{adp}(x, y, t) \) that is in between 0.8 and 1.2. The function \( E_{adp}(x, y, t) \) has been used in [Hua05] and [CHR02] to measure the mesh quality.

All the computations for the MKP methods were done in a double precision Fortran 77. The computations for the PMA method were done in Matlab 7.1. The computations for the GCL method were done in C++. All the computations in this
chapter were done on a computer with a 2.4 GHz AMD Opteron processor.

Example 4.1. To show the effectiveness of the MKP methods for generating adaptive meshes, we use the time dependent analytic function

\[ \rho(x, t) = 1 + 5\exp\left(-50 \left| (x - 0.5 - t)^2 + (y - 0.5)^2 - 0.09 \right|\right), \quad t > 0, \quad (4.2) \]

as an adaptation function. This adaptation function has been used to test the methods of [CHR02] for adaptive grid generation.

The function (4.2) has its maximum values on a circle of radius 0.3 moving horizontally to the right. The adaptive mesh is expected to cluster around this moving circle.

We apply both the PMKP and the FDMKP methods to generate adaptive meshes for the given adaptation function (4.2). The adaptive meshes are computed in the time interval \([0, 1]\) at the discrete time levels \(t^n = (n - 1)\Delta t, n = 1, 2, \ldots 101\), for \(\Delta t = 0.01\). The total time used to compute the meshes at these time levels is of order two minutes for each method for a mesh of size \(41 \times 41\).

The generated adaptive meshes together with the surfaces of the equidistribution measure function \(E_{\text{adp}}(x, y, t)\) at some selected times \(t = 0, 0.25, 0.5\) and 0.75, are shown in Figure 4.1 and Figure 4.2 for the case of using the PMKP the FDMKP methods, respectively. Observe that in both figures the mesh points are, as expected, concentrated around a circle moving horizontally to the right and leaving the domain while maintaining its shape. Table 4.1 shows the values of the function \(E_{\text{adp}}(x, y, t)\) computed by the PMKP and the FDMKP methods. Notice that the maximum and minimum values of the function \(E_{\text{adp}}(x, y, t)\) for the FDMKP method case are not as good as for the PMKP method case. This is due to some reasons related to the numerical relaxation method used in solving the saddle point problem for the optimal mesh velocity. Also the time integration of the mesh velocity contributes extra error in the computation of the adaptive mesh.
Figure 4.1: Example 4.1. The plots of the adaptive meshes (left) and the equidistribution measure function $E_{\text{adp}}(x, y, t)$ (right) computed with the PMKP method for the adaptation function (4.2) at $t = 0, 0.25, 0.5, 0.75$. The surface of $E_{\text{adp}}(x, y, t)$ shows a very accurate equidistribution of the mesh.
Figure 4.2: Example 4.1. The plots of the adaptive meshes (left) and the surfaces of $E_{\text{adv}}(x, y, t)$ (right) computed using the FDMKP method for the adaptation function \((4.2)\) at $t = 0, 0.25, 0.5, 0.75$. Notice that the mesh points are properly concentrated in the right regions.
Table 4.1: Example 4.1. The table shows the maximum and minimum values of the mesh quality measure function $E(x,y,t)$ with respect to the adaptation function (4.2) for the PMKP and the FDMKP methods.

<table>
<thead>
<tr>
<th>$t = 0$</th>
<th>$t = 0.25$</th>
<th>$t = 0.5$</th>
<th>$t = 0.75$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{min}}$</td>
<td>0.9441</td>
<td>0.9589</td>
<td>0.9599</td>
</tr>
<tr>
<td>$E_{\text{max}}$</td>
<td>1.0426</td>
<td>1.0782</td>
<td>1.0919</td>
</tr>
</tbody>
</table>

Example 4.2. In this example, we apply the FDMKP and the PMKP methods to generate two-dimensional adaptive meshes for a given analytical solution of a physical problem. Let the physical solution to be defined as

$$u(x,y) = 5.0 - \sum_{i=0}^{4} \tanh \left( 30 \left( 16(x-x_i)^2 + 16(y-y_i)^2 - 0.125 \right) \right), \quad (4.3)$$

where $(x,y) \in \Omega = (0,1) \times (0,1)$ and $x_i = 0.5, 3/8, 3/8, 5/8, 5/8$ and $y_i = 0.5, 3/8, 5/8, 5/8, 3/8$, for $i = 0, 1, 2, 3, 4$, respectively.

For the solution (4.3), its large variations occur around five circles with centers at $(x_i, y_i), i = 0, 1, 2, 3, 4$, as seen in Figure 4.3. Thus, the adaptive mesh is expected to concentrate the grid points around these five circles.

We apply both the PMKP and the FDMKP methods with the arc-length adaptation function (3.79) to generate the adaptive mesh corresponding to the solution (4.3). Figure 4.4 shows the adaptive meshes computed with the PMKP and the FDMKP methods. Clearly, the mesh points are well concentrated in the correct regions as expected. To show the mesh quality, we also plot the equidistribution measure function $E_{\text{adp}}(x,y)$ in Figure 4.4.

In this example we study the stopping criterion (3.14) for the PMKP method (see Subsection 3.1.2). Here, we plot the res = $\|\nabla \Psi_r\|_2$ as a function of the number of iterations of the first order Euler scheme used for the time integration of the parabolic Monge-Ampère equation (3.8). We run the scheme for a very large number of iterations and then plot the residual to see how it decreases in time. This gives an
estimate of when to stop the time integration. The plot of the residual is shown in Figure 4.5. Observe that the $L^2$ norm of the residual is decreasing as time evolves (i.e. the number of iterations increases). In this example, we only need about 12000 iterations to obtain a very good approximation for the solution. In this case, the $L^2$ norm of the residual is about $10^{-2}$. 

Figure 4.3: Example 4.2. The plot of the physical solution (4.3) defined on the physical domain $\Omega$ of size $41 \times 41$ grid points.
Figure 4.4: Example 4.2. A typical adaptive mesh of size $41 \times 41$ (first row), a cutaway of the adaptive mesh in the region close to where the solution has large variation (second row) and the surface of the function $E_{adp}(x, y)$ (third row) computed using the PMKP method (left) and the FDMKP method (right).
CHAPTER 4. NUMERICAL EXPERIMENTS

Figure 4.5: Example 4.2. The loglog plot of the $L^2$ norm of the residual obtained in the time integration of the parabolic Monge-Ampère equation.

Example 4.3. In this example, we demonstrate the capability of the FDMKP and the PMKP methods for generating adaptive meshes by considering the adaptation function defined as the analytic function

$$
\rho(x, y, t) = 1 + 5.0 \exp\left(-100 (y - 0.5 - 0.25 \sin(2\pi x) \sin(2\pi t))^2\right),
$$

which simulates the motion of a periodic sine wave.

In Figure 4.6 and Figure 4.7, we show the adaptive meshes together with the mesh quality measure function $E(x, y, t)$ at times $t = 0, 0.25,$ and $0.75$ obtained with the PMKP and the FDMKP methods, respectively. Notice that the mesh concentration accurately captures the moving sine wave. The function $E(x, y, t)$ is approximately equal 1 over the whole physical domain. Also notice that, for this example, at time $t = 1$ we exactly recover the adaptive mesh obtained at $t = 0$ due to the fact that the two meshes are obtained as a result of solving the same $L^2$ Monge–Kantorovich problem because the adaptation function has the same value at $t = 0$ and $t = 1.$
Figure 4.6: Example 4.3. Plot of adaptive meshes of size $41 \times 41$ (left) and the function $E_{adp}(x, y)$ (right) obtained with the PMKP method. Note that the mesh is properly concentrated along the moving sine wave.
Figure 4.7: Example 4.3. Plot of adaptive meshes of size $41 \times 41$ (left) and the function $E_{\text{adp}}(x, y)$ (right) obtained with the FDMKP method. Note that the mesh is properly concentrated along the moving sine wave.
Example 4.4. In this example, we compare the quality of the adaptive meshes generated with the PMKP and the FDMKP methods with the adaptive meshes generated by the GCL [CHR02] and the PMA [BW06] methods. To this end, we study two different cases of adaptation functions. In the first case, we consider a time dependent adaptation function defined as the analytic function

\[ \rho(x, y, t) = 1 + 10 \exp \left( -50 \left( (x - 0.5 - 0.25 \cos(2\pi t))^2 + (y - 0.5 - 0.25 \sin(2\pi t))^2 - 0.01 \right) \right) . \]  

(4.5)

In the second case we use the adaptation function

\[ \rho(x, y, t) = 1 + 5 \text{sech} \left( -50 \left( x + y - 1.5t - 0.25 \right) \right) + 5 \text{sech} \left( -50 \left( x + y + 1.5t - 1.75 \right) \right) . \]  

(4.6)

The function (4.5) simulates the periodic motion of a circle defined on a unit square domain, and its large values occur around this moving circle. The function (4.6) simulates the motion of two linear fronts traveling towards each other.

We use the adaptation function (4.5) for generating the adaptive mesh. The mesh quality measure function \( E(x, y, t) \) in (4.1) is used to compare the quality of the adaptive meshes obtained by the PMKP, FDMKP, PMA and GCL methods.

In Table 4.2 and Table 4.3 we show the minimum and the maximum values of the function \( E(x, y, t) \) for the two cases of using the adaptation functions (4.5) and (4.6), respectively. In Figure 4.8, Figure 4.9, Figure 4.10 and Figure 4.11 we show the adaptive meshes and the corresponding surfaces of the function \( E(x, y, t) \) computed using the four adaptive mesh methods at some selected times \( t = 0.0, 0.25, 0.5, \) and 0.75. Figure 4.12, Figure 4.13, Figure 4.14 and Figure 4.15 present the adaptive meshes and the surface of the function \( E(x, y, t) \) for the linear wave adaptation function (4.6) at \( t = 0.0, 0.25, 0.35, \) and 0.5. Observe that, the surfaces of \( E_{\text{adp}}(x, y, t) \) stay close to 1 for all times \( t, \) and the adaptive meshes are smoother and more accurately concentrated in the proper regions for the PMKP method compare to the FDMKP, PMA and GCL methods. The GCL method breaks down for the case of the two linear front waves adaptation function, and that is mainly due to the fact that the mesh obtained by the GCL method follows the moving features of the adaptation function. The
FDMKP method doesn’t have this problem although it is closely related to the GCL method. The reason for this is because the mesh obtained by the FDMKP method at the current time level is independent from that obtained in the previous time level.

Notice that, for the (4.6), when using the MKP methods, the adaptive meshes obtained at times $t > 0.5$ (after the collision of the two shocks) are exactly the same as the ones obtained at times $t < 0.5$ (before the collision of the two shocks). This is due to the fact that the adaptation function has the same value before and after the collision of the two shocks and as a result the same $L^2$ Monge–Kantorovich problem is solved to obtain the adaptive mesh before and after the collision time.

\[
\begin{array}{|c|cc|c|cc|c|cc|c|cc|}
\hline
& \text{PMKP} & & \text{FDMKP} & & \text{PMA} & & \text{GCL} & \\
& E_{\text{min}} & E_{\text{max}} & E_{\text{min}} & E_{\text{max}} & E_{\text{min}} & E_{\text{max}} & E_{\text{min}} & E_{\text{max}} \\
\hline
 t = 0 & 0.9918 & 1.0187 & 0.8939 & 1.1977 & 1.1095 & 1.2105 & 0.9134 & 1.1102 \\
t = 0.25 & 0.9892 & 1.0160 & 0.8403 & 1.2816 & 0.9110 & 1.6676 & 0.7976 & 1.1821 \\
t = 0.5 & 0.9924 & 1.0193 & 0.8178 & 1.4245 & 0.9486 & 1.7231 & 0.2885 & 4.0641 \\
t = 0.75 & 0.9899 & 1.0168 & 0.6679 & 1.2599 & 0.9619 & 1.7416 & 0.2103 & 5.6454 \\
\hline
\end{array}
\]

Table 4.2: Example 4.4. The table shows the maximum and minimum values of the mesh quality measure function $E(x, y, t)$ with respect to the adaptation function (4.5) for the methods PMKP, FDMKP, PMA and GCL.

\[
\begin{array}{|c|cc|c|cc|c|cc|c|cc|}
\hline
& \text{PMKP} & & \text{FDMKP} & & \text{PMA} & & \text{GCL} & \\
& E_{\text{min}} & E_{\text{max}} & E_{\text{min}} & E_{\text{max}} & E_{\text{min}} & E_{\text{max}} & E_{\text{min}} & E_{\text{max}} \\
\hline
 t = 0 & 0.9852 & 1.0238 & 0.5678 & 1.4697 & 1.0073 & 1.1034 & 0.8060 & 1.2540 \\
t = 0.25 & 0.9894 & 1.0359 & 0.6970 & 1.7618 & 0.8675 & 1.5633 & 0.7068 & 2.8510 \\
t = 0.35 & 0.9946 & 1.0399 & 0.6922 & 1.9866 & 0.8866 & 1.5412 & 0.6165 & 5.3956 \\
t = 0.45 & 0.9947 & 1.0365 & 0.6467 & 2.257 & 0.9162 & 1.5974 & 0.2539 & 23.8259 \\
\hline
\end{array}
\]

Table 4.3: Example 4.4. The table shows the maximum and minimum values of the mesh quality measure function $E(x, y, t)$ with respect to the adaptation function (4.6) for the methods PMKP, FDMKP, PMA and GCL.
Figure 4.8: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{adp}(x, y, t)$ (right) computed at $t = 0, 0.25, 0.5$ and $0.75$ with the PMKP method for the adaptation function (4.5). Note that $E_{adp}(x, y, t)$ remains close to 1 as the time evolves.
Figure 4.9: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{adp}(x, y)$ (right) computed at $t = 0, 0.25, 0.5$ and $0.75$ with the FDMKP method for the adaptation function (4.5). Notice that $E_{adp}(x, y, t)$ has the same profile as the time evolves.
Figure 4.10: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{\text{adp}}(x, y, t)$, $t = 0, 0.25, 0.5$ and 0.75, (right) generated with the PMA method for the adaptation function (4.5). Observe that the change of the surface of $E_{\text{adp}}(x, y, t)$ as time evolves.
Figure 4.11: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{adp}(x,y,t), \ t = 0, 0.25, 0.5$ and $0.75$, (right) for the adaptation function (4.5) with the GCL method. Note that the surface $E_{adp}(x,y,t)$ diverges from 1 as time evolves.
Figure 4.12: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{adp}(x, y, t)$ (right) computed with the PMKP method at $t = 0, 0.25, 3.5$ and 0.5 for the adaptation function (4.6). Notice that $E_{adp}(x, y, t)$ is close to 1 for all times $t$. 
Figure 4.13: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{odp}(x, y, t)$ (right) computed at $t = 0, 0.25, 0.35$ and $0.5$ with the FDMKP method for the adaptation function (4.6).
Figure 4.14: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{\text{adp}}(x, y, t)$ (right) computed at $t = 0, 0.25, 3.5$ and $0.5$ with the PMA method for the adaptation function (4.6).
Figure 4.15: Example 4.4. Adaptive meshes of size $41 \times 41$ (left) and the function $E_{adp}(x, y, t)$ (right) computed at $t = 0, 0.25, 3.5$ and $0.5$ with the GCL method for the adaptation function (4.6).
Example 4.5. We consider generating an adaptive mesh that is suitable for solving a blow-up type problem using the MKP methods. To this end, we consider the extreme case of a given analytical solution

\[ u(x, y) = c \exp \left( -c^2 (x^2 + y^2) \right), \quad c = 100. \]  

(4.7)

The solution \( u(x, y) \) is very sharp near the point \((x, y) = (0, 0)\). The solution (4.7) has been used by Ceniceros and Hou [CHO1] to show the efficiency of their mesh generation method.

To show the effect of the adaptation function on concentrating the generated mesh, we use both the adaptation functions

\[ \rho(x, y) = \sqrt{1 + |\nabla x u|^2}, \]  

(4.8)

and

\[ \rho(\xi, \eta) = \sqrt{1 + \alpha u^2 + \beta |\nabla \xi u|^2} \]  

(4.9)

to generate the adaptive mesh on the physical domain.

Notice that the first adaptation is defined in terms of the physical domain variables \( x = (x, y) \), where the second is defined in terms of the computational domain variables \( \xi = (\xi, \eta) \). For this example we take \( \alpha = \beta = 1 \).

A mesh of size 129x129 is used to compute the adaptive mesh for the given solution (4.7). The adaptive mesh on the physical domain and a cutaway close to the peak region are shown in Figure 4.16 and Figure 4.18 for using the adaptation functions (4.9) and (4.8), respectively. We plot the solution \( u(x, y) \) on both the physical and computational domains in Figure 4.17 and Figure 4.18 with respect to the adaptation functions (4.9) and (4.8). The figures show that the solution has a wider support on the computational domain and that the mesh points are well concentrated in the blow-up region. The concentration factor, the ratio of the smallest cell size of the adaptive mesh and the uniform mesh, is equal to 97 and 87 for the adaptation function (4.8) and (4.9), respectively.
Figure 4.16: Example 4.5. Adaptive mesh of size $129 \times 129$ generated on the physical domain (top) and a cutaway near the blow-up region (bottom) for the solution $u = ce^{-c(x^2+y^2)}, c = 100$, using the adaptation function (4.9).
Figure 4.17: Example 4.5. The solution $u = ce^{-c^2(x^2+y^2)}, c = 100$ plotted on the physical domain (top) and the computational domain (bottom) corresponding to the adaptation function (4.9).
Figure 4.18: Example 4.5. Adaptive mesh of size $129 \times 129$ generated on the physical domain (top) and a cutaway near the blow-up region (bottom) for the solution $u = ce^{-c^2(x^2+y^2)}$, $c = 100$, using the adaptation function (4.8). Clearly, the grid points are well concentrated around the peak region.
Figure 4.19: Example 4.5. The solution \( u = ce^{-c^2(x^2+y^2)} \), \( c = 100 \) plotted on the physical domain (top) and the computational domain (bottom) corresponding to the adaptation function (4.8). Notice that the solution \( u \) has a wider support on the computational domain.
Example 4.6. In this example, we consider generating adaptive meshes with the MKP methods for some practical applications. Precisely speaking, we employ the PMKP and the FDMKP methods to compute the numerical solution of the initial boundary value problem for the two-dimensional viscous Burgers' equation

\[ u_t = \epsilon \Delta u - uu_x - uu_y, \quad (x, y) \in (0, 1)^2, \quad 0.25 \leq t \leq 1.25, \quad \epsilon = 5 \times 10^{-3}, \quad (4.10) \]

with initial and Dirichlet boundary conditions chosen such that the exact solution to the problem is

\[ u(x, y, t) = 1/(1 + \exp((x + y - t)/2\epsilon)). \quad (4.11) \]

The solution \( u(x, y, t) \) has a steep interior planar layer of thickness \( O(\epsilon) \), and it is a straight line wave propagating along the diagonal of the domain with constant velocity.

We follow the alternate procedure described in Section 3.3 for the numerical solution of (4.10) and the adaptive mesh. The adaptive mesh is computed by applying the MKP methods. For the physical solution, we first rewrite (4.10) as

\[ \dot{u} - u_x \dot{x} - u_y \dot{y} = \epsilon \Delta u - uu_x - uu_y. \quad (4.12) \]

After transforming (4.12) into the computational domain \( \Omega_c \), using the formulas (3.61), (3.62), (3.63), (3.64) and (3.65), the resulting PDE is discretized on a uniform mesh of size 41 x 41. The first order derivatives are approximated by upwind finite differences, where the standard centered finite differences are used to approximate the second order derivatives. One can use an implicit or explicit ODE solver to solve the resulting ODE system for the physical solution. Here, we use DASSL, a stiff differential-algebraic system solver that uses backward differentiation schemes, to solve the ODE system. Both of the adaptation functions (4.9) and (4.8) are used to illustrate various properties of the physical solution and the adaptive meshes generated corresponding to different adaptation functions.

Figure 4.20 and Figure 4.21 show the numerical results of computing the solution of problem (4.10) on the time interval \([0, 1.25]\) when using the adaptation functions (4.8) and (4.9), respectively. The front lies along the line joining the points \((1,0)\) and
(0,1) at time $t = 1.0$. At this time, the concentration of the mesh points switches from one edge to another, and this leads to some additional numerical challenges. Figure 4.20 shows that the mesh points are well clustered along the front which leads to a very good resolution of the front with no observable oscillations in the solution. This is a good example test which demonstrates the capability of the MKP methods for generating adaptive meshes suitable for solving time-dependent partial differential equation.

The error of the solution is computed using the $L^2$ norm defined in the physical domain as

$$
\text{error} = \left( \int_{\Omega} |u_e(x) - u_c(x)|^2 \, dx \right)^{\frac{1}{2}} = \left( \int_{\Omega_{\xi}} |u_e(\xi) - u_c(\xi)|^2 |J(\xi)| \, d\xi \right)^{\frac{1}{2}},
$$

where $u_e$ is the exact solution, $u_c$ is the computed solution and $J(\xi)$ is the Jacobian of the transformation $x(\xi)$. For this example, the exact solution is given by (4.11).

The effect of the adaptation functions in generating the adaptive mesh can be shown by comparing the adaptive mesh and the $L^2$ norm of the error of the physical solution computed on the adaptive mesh obtained using the adaptation functions (4.9) and (4.8). In Figure 4.22 (a), (b) and (c) we show the plots of the $L^2$ norm of the errors of the numerical solutions computed on the adaptive mesh and the uniform mesh. Figure 4.20 shows that the mesh obtained with the adaptation function (4.8) is concentrated more accurately on the traveling front. For the case of the adaptation function (4.9), Figure 4.21 shows that the mesh is less concentrated along the front compare to that obtained using the adaptation function (4.8). This results in not perfectly resolving the solution at the front which cause an increase in the $L^2$ norm of the error of the numerical solution for the physical problem as illustrated in Figure 4.22 (a), (b). Therefore, we obtain a more accurate solution when using the adaptation function (4.8) defined in terms of the physical domain variables. This raises an interesting question of defining the adaptation function in terms of the physical domain variables versus the computational domain variables. In this particular example, the adaptation function (4.8) is preferable over (4.9).

In what follows, we carry a numerical convergence study to show how the error changes with the grid spacing and the number of the grid points. To this end, fix the
time and consider the following three cases:

1. the change of the error with the number of the grid points $N$.

2. the change of the error with the maximum of the grid spacing $\Delta x$.

3. the change of the error with the minimum of $\Delta x$.

Table 4.4 shows the change of the $L^2$ norm of the error with the maximum and minimum of $\Delta x$ at a fixed time $t = 0.75$. In Table 4.5, we show the change of the $L^2$ norm of the error with the number of the grid points, $N$, at some selected fixed times $t = 0.5, 0.75, 1, 1.25$. Figure 4.23 (left) shows the plot of the $L^2$ norm of the error as a function of $N$. The change of the $L^2$ norm of the error with the maximum and minimum grid spacing $\Delta x$ at a fixed time $t = 0.75$ is shown in Figure 4.23 (right). The three cases give straight lines with slopes of absolute values 1.2002, 1.0547 and 1.2388, respectively. Thus, we conclude that the convergence is of order one.

To compare the PMKP and PMA methods, in Figure 4.24 we plot the $L^2$ norm error obtained in computing the numerical solution of the 2D Burgers' equation. For the purpose of the comparison we write the code for both methods in Fortran. The computational time for the PMKP method is 8.3 minutes and for the PMA method is 6.2833 minutes. Figure 4.24 shows that we obtain more accurate numerical solution when we use the PMKP method, observe that at time $t = 1.25$ the error for the PMA method is about twice the error for the PMKP method.
Figure 4.20: Example 4.6: Snapshots of the numerical solution of the 2D Burgers' equation (4.10) (left) and the adaptive meshes (right) computed using the adaptation function (4.8) at $t = 0.25, 0.5, 1.0$ and 1.25.
Figure 4.21: Example 4.6. Snapshots of the numerical solution of the 2D Burgers’ equation (4.10) (left) and the adaptive meshes (right) computed with the adaptation function (4.9) at times $t = 0.25, 0.5, 1.0$ and $1.25$. 
CHAPTER 4. NUMERICAL EXPERIMENTS

Figure 4.22: Example 4.6. The $L^2$ norm of the errors in the numerical solution of the 2D Burgers’ equation (4.10) computed on the adaptive grid of size $41 \times 41$ using the adaptation functions (a) (4.8) and (b) (4.9). Plotted in (c) is the error corresponding to the uniform mesh case. Note that the solution is more accurate for the case of using the adaptation function (4.8) defined in terms of the physical domain variables.

Figure 4.23: Example 4.6. The numerical convergence study results for solving the 2D Burgers' equation. The plot of the $L^2$ norm of the error as a function of the number of the grid points (left) and the maximum and minimum of the grid spacing $\Delta x$ (right) at fixed time $t = 0.75$. The slope is 1.2388 for the first plot (left) and is 1.2002 for maximum $\Delta x$ and 1.0547 for minimum $\Delta x$ in the second plot (right).
CHAPTER 4. NUMERICAL EXPERIMENTS

Table 4.4: Example 4.6. The numerical convergence study results of solving the 2D Burgers' equation. The table shows the change of the $L^2$ norm of the error with the grid spacing $\Delta x$ at fixed time $t = 0.75$.

<table>
<thead>
<tr>
<th>N</th>
<th>$\Delta x_{min}$</th>
<th>$\Delta x_{max}$</th>
<th>$L^2$ error</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0.0186</td>
<td>0.1861</td>
<td>0.0457</td>
</tr>
<tr>
<td>21</td>
<td>0.0066</td>
<td>0.0981</td>
<td>0.0193</td>
</tr>
<tr>
<td>41</td>
<td>0.029</td>
<td>0.0492</td>
<td>0.0084</td>
</tr>
<tr>
<td>81</td>
<td>0.0014</td>
<td>0.0252</td>
<td>0.0039</td>
</tr>
</tbody>
</table>

Table 4.5: Example 4.6. The numerical convergence study results for solving the 2D Burgers' equation. The table shows the change of the $L^2$ norm of the error with the number of the grid points given at times $t = 0.5, 0.75, 1.0, 1.25$.

<table>
<thead>
<tr>
<th>N</th>
<th>$t = 0.5$</th>
<th>$t = 0.75$</th>
<th>$t = 1.$</th>
<th>$t = 1.25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0.0234</td>
<td>0.0457</td>
<td>0.0754</td>
<td>0.1096</td>
</tr>
<tr>
<td>21</td>
<td>0.009</td>
<td>0.0193</td>
<td>0.0345</td>
<td>0.0539</td>
</tr>
<tr>
<td>41</td>
<td>0.0045</td>
<td>0.0084</td>
<td>0.016</td>
<td>0.0274</td>
</tr>
<tr>
<td>81</td>
<td>0.0034</td>
<td>0.0039</td>
<td>0.0066</td>
<td>0.0127</td>
</tr>
</tbody>
</table>
Figure 4.24: Example 4.6. The $L^2$ norm of the errors in the numerical solution of the 2D Burgers' equation obtained for the PMKP (blue) and PMA (red) methods. The computational time for the PMKP method is 8.3 minutes and for the PMA method is 6.2833 minutes. Notice that at $t = 1.25$ the error of the PMA method is about twice the error of the PMKP method.

Example 4.7. In this example, we apply the PMKP method to generate a three-dimensional adaptive mesh for the analytic solution

$$u(x, y, z) = \tanh \left(100 \left((x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2\right)\right)$$ \hspace{1cm} (4.13)

defined in the unit cube. The maximum values of the given function (4.13) occur on a sphere of radius $1/4$ centered at $(0.5, 0.5, 0.5)$. In this case, the adaptive mesh is expected to be concentrated on this sphere.

For adaptive grid generation in three spatial dimension, we use the arc-length adaptation function

$$\rho(x, y, z) = \sqrt{1 + u_x^2 + u_y^2 + u_z^2},$$ \hspace{1cm} (4.14)

where $u_x, u_y, u_z$ are the partial derivatives with respect to the physical variables $x, y, z$, respectively.

The mesh quality measure function in 3D is defined as

$$E_{\text{adv}}(x, y, z) = \rho(x, y, z) \left| \det D^2 \Psi(\xi, \eta, \zeta) \right|_{\Omega_c}.$$ 


A mesh of size 65 x 65 x 65 is used to generate the adaptive mesh on the physical domain. To facilitate the visualization of the mesh, in Figure 4.25 we show the generated adaptive meshes on three slices parallel to the xy, xz and yz planes. As expected, the adaptive meshes are concentrated around a circle on each slice. The minimum and maximum values of the mesh quality measure function $E_{adp}(x, y, z)$ are 1.1173 and 1.1249 respectively.

![Figure 4.25: Example 4.7. A typical 3D adaptive mesh of size 65 x 65 x 65 computed with the PMKP method using the adaptation function (4.14) for the analytic function (4.13). Plane slices parallel to xy-plane (left), xz-plane (middle) and yz-plane (right). Observe that the mesh points are well concentrated in the proper regions.](image)

**Example 4.8.** In this example, the PMKP method is applied to generate a 3D adaptive mesh for the analytic solution

$$u(x, y, z) = \sum_{p=0}^{8} \tanh \left(30 \left((x - x_0(p))^2 + (y - y_0(p))^2 + (z - z_0(p))^2 - 0.1875\right)\right)$$

defined in the cube $(-2, 2)^3$, where, $x_0(p) = 0, +0.5, -0.5, -0.5, +0.5, +0.5, -0.5, -0.5, y_0(p) = 0, +0.5, +0.5, -0.5, -0.5, -0.5, -0.5, -0.5, z_0(p) = 0, +0.5, -0.5, +0.5, -0.5, +0.5, -0.5, +0.5, -0.5$, for $p = 0, 1, \ldots 8$. The solution $u(x, y, z)$ given in (4.15) has its large variations around nine spheres centered at the points $(x_0(p), y_0(p), z_0(p)), p =$
0, 1, 2, \cdots, 8, each is of radius $\sqrt{0.1875}$. Therefore, the adaptive mesh is expected to be clustered on these nine spheres.

We use the arc-length adaptation function (4.14) to generate the adaptive mesh of size $41 \times 41 \times 41$ on the physical domain. In Figure 4.26 we show the cutaway plot of three slices parallel to the xy, xz and yz planes for the generated 3D adaptive meshes. Clearly, the adaptive mesh is properly concentrated around the nine spheres. The mesh quality measure function $E_{\text{adp}}(x, y, z)$ has minimum and maximum values of 1.2661 and 1.2669, respectively.
Chapter 5

The MKP Elastic Registration Methods

In Chapter 3, we discussed an optimal transport approach for computing a coordinate transformation for adaptive mesh generation. In this chapter, we use this approach to describe methods for computing coordinate transformations for elastic image registration in two or three spatial dimensions. The elastic model of image registration, was first studied by Broit (1981) [Bro81]. In this model, the images (the template and the target) are viewed as two different states of an elastic body, one before and one after a deformation. The template image is deformed by applying external forces. The forces are often derived to minimize some cost function measuring the similarity between the template image and the target image.

5.1 MKP Registration Framework

In this section we describe the Monge–Kantorovich or the optimal mass transport approach for image registration.

Let $\Omega_0$ and $\Omega_1$ be bounded domains in $\mathbb{R}^d$, $d = 2$ or $d = 3$. Given two images, a template $\rho_0$ and a target $\rho_1$, defined on $\Omega_0$ and $\Omega_1$ respectively. The two images may be obtained from the same object at two different times. The images $\rho_0$ and $\rho_1$ are assumed to be gray-scale images, more precisely, real bounded positive functions.
The registration problem is to determine a coordinate transformation, \( \phi : \Omega_0 \rightarrow \Omega_1 \), such that the deformed template image becomes similar to the target image \( \rho_1 \). Different choices of the similarity measures lead to different registration methods; for example, one can use the regular \( L^2 \) distance

\[
\| \rho_0 \circ \phi^{-1} - \rho_1 \|_2^2
\]

as a similarity measure, where \( \| \cdot \|_2 \) is the standard \( L^2 \) norm and \( \rho_0 \circ \phi^{-1}(x) = \rho_0(\phi^{-1}(x)), x \in \Omega_1 \).

If one thinks of \( \rho_0 \) and \( \rho_1 \) as density functions, then the MKP approach for image registration seeks to find the mapping \( \phi : \Omega_0 \rightarrow \Omega_1 \) that moves a point \( \xi \in \Omega_0 \) to location \( x = \phi(\xi) \in \Omega_1 \) with displacement \( u(\xi) = \phi(\xi) - \xi \) such that the density of material \( \rho_0 \) in domain \( \Omega_0 \) is transferred to become the density \( \rho_1 \) given in the domain \( \Omega_1 \). This is the data matching constraint of the problem. Mathematically, this means that the mass of material in any bounded subset \( \Omega \subset \Omega_1 \) must then equal the mass of material coming from the pre-image in the subset \( \phi^{-1}(\Omega) \subset \Omega_0 \), i.e.,

\[
\int_{\phi^{-1}(\Omega)} \rho_0(\xi) d\xi = \int_{\Omega} \rho_1(x) dx
\]

for all bounded subsets \( \Omega \) of \( \Omega_1 \).

For matching anatomical images, the coordinate transformation for image registration is ideally desired to be bijective (one-to-one and onto) so that the neighborhood structure is maintained. This transformation, if possible, is required to be continuous with a continuous inverse, so that tears are not created by such a transformation. The optimal mass approach for image registration assures these features, therefore makes this approach to be more appropriate and attractive for medical applications.

If \( \phi \) is a smooth one-to-one map then by a change of variables, (5.2) leads to the Jacobian equation

\[
\rho_1(\phi(\xi))\det \nabla \phi(\xi) = \rho_0(\xi),
\]

where \( \det \nabla \phi(\xi) \) is the determinant of the Jacobian matrix that describes the distortions of the area or volume element induced by the transformation \( \phi \).

The solution to (5.3) is not unique as it is only one equation involving \( d \) unknowns. Thus, from all the different transformations \( \phi \) that could yield the same quality of
registration, an optimal one must be specified. The optimal mass transport approach for image registration is based on taking the $L^2$ Kantorovich-Wasserstein distance

$$d(\rho_0, \rho_1) = \inf_{\phi} \left( \int_{\Omega_0} |\phi(\xi) - \xi|^2 \rho_0(\xi) d\xi \right)^{\frac{1}{2}}$$

as a similarity measure, where $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^d$ and the infimum is taken among all maps $\phi$ transporting $\rho_0$ to $\rho_1$.

The registration problem in terms of the $L^2$ Monge-Kantorovich problem is summarized as follows. Given two images $(\Omega_0, \rho_0)$ and $(\Omega_1, \rho_1)$, find the transformation $\phi$ which takes the image $\rho_0$ to the image $\rho_1$ and minimizes the cost

$$C(\phi) = \int_{\Omega_0} |\phi(\xi) - \xi|^2 \rho_0(\xi) d\xi.$$  

Here, we seek mappings $\phi$ which deform one image to another while minimizing the cost of moving the pixels (5.5). From the Monge-Kantorovich theory, we know that the minimizer in (5.4) has a number of nice features including uniqueness and its characterization as the gradient of some convex potential. Furthermore, the problem is symmetric, i.e., if $\phi$ is the optimal transformation from image A to image B then $\phi^{-1}$ is the optimal transformation from image B to image A (see [Vil03]). These features make the optimal mass transport approach well suited for the registration problem.

We emphasize that the idea of using the optimal mass transport approach for image registration is not new. For example, Haker et al. [HT01, AHT03, HT03a, HZTA04] have described an image registration based upon finding the transformation $\phi$ as a steady-state solution of a gradient flow equation (cf. Subsection 2.5.1). However, in this thesis we use different and simpler methods to solve for the optimal transformation. More precisely, we use the PMKP and the FDMKP methods, described in Chapter 3 for adaptive grid generation, to compute the optimal transformation. In the next two sections, we describe these two methods for computing transformations for elastic image registration.
5.2 The PMKP Registration Method

In this section, we discuss the details of using the PMKP method to compute the optimal mapping for the image registration problem.

Consider the gray-scale images \((\Omega_0, \rho_0)\) and \((\Omega_1, \rho_1)\). The mapping \(\phi\) that transfers \(\rho_0\) to \(\rho_1\) and minimizes the cost (5.5) is unique and is characterized as (cf. Section 2.3)

\[
x = \phi(\xi) = \nabla \Psi(\xi), \quad \xi \in \Omega_0, \quad x \in \Omega_1.
\] (5.6)

The equations (5.3) and (5.6) lead to the conclusion that the solution of the registration problem reduces to solving the Monge-Ampère equation (MAE)

\[
\rho_1 (\nabla \Psi(\xi)) \det (D^2 \Psi(\xi)) = \rho_0(\xi),
\] (5.7)

where \(\det (D^2 \Psi)\) is the determinant of the Hessian matrix of the potential \(\Psi\).

The PMKP method, as described in Section 3.1, is based on solving (5.7) by finding the steady-state solution of

\[
\frac{\partial \Psi}{\partial t} = \log \left( \frac{\rho_1 (\nabla \Psi(\xi)) \det (D^2 \Psi(\xi))}{\rho_0(\xi)} \right)
\] (5.8)

with some appropriate temporal initial and spatial boundary conditions.

If \(\Psi^\infty\) is the limiting solution as \(t \to \infty\), then the optimal matching map \(\hat{\phi}\) is obtained by taking the gradient of \(\Psi^\infty\), i.e., \(\hat{\phi} = \nabla \Psi^\infty\).

The corresponding flow \(\phi(\xi; t)\) for the image warping (deformation) is defined as

\[
\phi(\xi, t) = \xi + t(\nabla \Psi(\xi) - \xi), \quad \xi \in \Omega_0, \quad 0 \leq t \leq 1.
\] (5.9)

The formula (5.9) shows that at time \(t = 0\), \(\phi\) is the identity map and at \(t = 1\), it is the optimal solution of the \(L^2\) Monge–Kantorovich problem. Thus, the formula (5.9) can be used to define continuous warping map \(\phi\) between the initial density \(\rho_0\) and the final density \(\rho_1\).

5.2.1 Computational Details

In this subsection, we discuss the computational details of solving (5.8) for the solution \(\Psi\) and its gradient to find the optimal matching coordinate transformation for image registration.
We use a standard centered finite difference scheme for the spatial discretization of (5.8) and an explicit scheme such as forward Euler or a higher-order Runge-Kutta scheme in time. Given two images $\rho_0$ and $\rho_1$, then the simplest algorithm to compute the potential $\Psi$ is the following:

1. Given $\Psi^n$
2. Compute $D^2\Psi^n$ and $\nabla\Psi^n$
3. Interpolate $\rho_1$ onto $\nabla\Psi^n$
4. Set
   \[ F^n = \log \left( \frac{\rho_1(\nabla\Psi^n)\det(D^2\Psi^n)}{\rho_0(\xi)} \right) \]
5. Update $\Psi^n$: $\Psi^{n+1} = \Psi^n + \Delta t F^n$
6. If $\|\nabla F^n\|_2 < TOL$ then STOP, else set $n = n + 1$ and go to 2.

where $mbox{TOL}$ is a user defined tolerance, $\|\nabla F^n\|_2$ is equivalent to $\|\nabla \Psi_i^{n+1}\|_2$.

The PMKP method is applied to 2D or 3D images defined on a unit square or unit cube domain, respectively. For this application, the active region of the domain can be defined away from the boundaries. Therefore, it is natural to use the Neumann conditions

\[ \frac{\partial \Psi}{\partial \xi_i} = 0, \quad \text{on } \xi_i = 0, \quad \frac{\partial \Psi}{\partial \xi_i} = 1, \quad \text{on } \xi_i = 1, \quad i = 1, 2 \text{ or } 3, \quad (5.10) \]

to solve (5.8) with the initial condition

\[ \Psi(\xi,0) = \frac{1}{2} \xi^T \cdot \xi, \quad \xi \in \Omega_0. \quad (5.11) \]

### 5.3 FDMKP Image Registration Method

In the previous section, the coordinate transformation for image registration is determined as the gradient of the solution of the Monge-Ampère equation (5.7). In this section, we present an elastic registration method based on solving the $L^2$ Monge–Kantorovich problem in a fluid dynamics framework.
Given the two images \((\Omega_0, \rho_0)\) and \((\Omega_1, \rho_1)\), fix a time interval, which for simplicity is taken to be \([0, 1]\). If we were to follow the motion of a point \(\xi \in \Omega_0\), then it should follow a time-dependent trajectory \(x = \phi(\xi, t), \ t \in [0, 1]\) with initial position \(x_0 = \phi(\xi, 0) = \xi\) and final position at the destination \(x = \phi(\xi, 1)\).

Consider all possible sufficiently smooth time-dependent density and velocity fields, \(\rho(x, t) > 0, \ v(x, t) \in \mathbb{R}^d\), which satisfy the continuity equation
\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0, \tag{5.12}
\]
and the conditions
\[
\rho(\cdot, 0) = \rho_0, \ \rho(\cdot, 1) = \rho_1, \tag{5.13}
\]
where \(\nabla\) is the gradient operator with respect to the variables \(\xi \in \Omega_0\).

The optimal trajectory \(x = \phi(\xi, t), \ t \in [0, 1]\), that takes \(\rho_0\) to \(\rho_1\), is then determined by time integration of the velocity field computed as the solution of the following minimization problem: Under the constraints (5.12) and (5.13), among all the density and velocity fields, \(\rho(\xi, t)\) and \(v(\xi, t)\), respectively, choose the ones that minimize the cost
\[
C(\rho, v) = \int_0^1 \int_{\Omega_0} \rho(\xi, t) |v(\xi, t)|^2 \, d\xi \, dt. \tag{5.14}
\]

The augmented Lagrangian method described in Section 3.2 is used to obtain the velocity and the density fields that minimize (5.14) with the constraints (5.12) and (5.13). The optimal matching transformation is then obtained by time integration of the velocity \(v(\xi, t)\) from \(t = 0\) to \(t = 1\) in conjunction with the initial condition (5.11).

The fluid dynamics formulation of the \(L^2\) Monge–Kantorovich problem provides a natural time interpolant \(\rho(x, t)\) from the data \(\rho_0\) to \(\rho_1\) and a velocity field \(v(x, t)\) which transports \(\rho_0\) to \(\rho_1\), a feature that is useful to show the intermediate stages of the image warping.

Note that minimizing (5.14) under the constraints (5.12) and (5.13) is formally similar to the minimization problem (2.56) defined for the LDDMM method [BMTY05]. In fact, the second term in (2.56) simply enforces the matching of the images to be exact at \(t = 1\), whereas for our approach this matching is achieved by the constraint (5.13).
5.4 Numerical Experiments

In this section we present numerous examples to illustrate the properties of our new methods for image registration. The examples have been chosen to give the reader some insight into how the method works for both simple problems and practical real world examples. For completeness, we present the calculations for computing the exact optimal solution for the toy problem in example 5.1.

5.4.1 Numerical Examples

Example 5.1. We begin with the simple problem of computing the optimal transformation for mapping one circle to another concentric one using the FDMKP method. For radially symmetric densities, the solution of the Monge-Ampère equation (3.79) can be found exactly by direct integration. Specifically, defining the polar coordinates

\[\xi = r \cos(\theta), \quad \eta = r \sin(\theta), \quad \xi, \eta \in \Omega_0,\]

and

\[r = \sqrt{\xi^2 + \eta^2}, \quad \theta = \tan^{-1}\left(\frac{\eta}{\xi}\right),\]

and using (5.15), (5.16), taking a change of variables we obtain

\[\frac{\partial}{\partial \xi} = \cos(\theta) \frac{\partial}{\partial r} - \frac{\sin(\theta)}{r} \frac{\partial}{\partial \theta},\]

and

\[\frac{\partial}{\partial \eta} = \sin(\theta) \frac{\partial}{\partial r} + \frac{\cos(\theta)}{r} \frac{\partial}{\partial \theta}.\]

Using (5.17) and (5.18), we obtain the second order derivatives for \(\Psi\) as

\[\frac{\partial^2 \Psi}{\partial \xi^2} = \cos^2(\theta) \frac{\partial^2 \Psi}{\partial r^2} - \frac{2 \cos(\theta) \sin(\theta)}{r} \frac{\partial^2 \Psi}{\partial r \partial \theta} + \frac{\sin^2(\theta)}{r} \frac{\partial \Psi}{\partial r} + \frac{\sin^2(\theta)}{r^2} \frac{\partial^2 \Psi}{\partial \theta^2},\]

(5.19)

\[\frac{\partial^2 \Psi}{\partial \eta^2} = \sin^2(\theta) \frac{\partial^2 \Psi}{\partial r^2} + \frac{2 \cos(\theta) \sin(\theta)}{r} \frac{\partial^2 \Psi}{\partial r \partial \theta} + \frac{\cos^2(\theta)}{r} \frac{\partial \Psi}{\partial r} + \frac{\cos^2(\theta)}{r^2} \frac{\partial^2 \Psi}{\partial \theta^2},\]

(5.20)

and

\[\frac{\partial^2 \Psi}{\partial \xi \partial \eta} = \frac{\sin(2\theta)}{2r} \frac{\partial^2 \Psi}{\partial r^2} + \frac{\cos(2\theta)}{r} \frac{\partial^2 \Psi}{\partial r \partial \theta} - \frac{\cos(2\theta)}{r^2} \frac{\partial \Psi}{\partial \theta} - \frac{\sin(2\theta)}{2r} \frac{\partial \Psi}{\partial r} - \frac{\sin(2\theta)}{2r^2} \frac{\partial^2 \Psi}{\partial \theta^2},\]

(5.21)
From the symmetry, the densities $\rho_0$ and $\rho_1$ are independent of $\theta$; therefore, the solution $\Psi$ of the Monge-Ampère equation (5.7) is also independent of $\theta$. Thus, the determinant of the Hessian matrix of $\Psi$ reduces to $\Psi_r \Psi_{rr}/r$ and (5.7) simplifies to

$$\frac{1}{r} \Psi_r \Psi_{rr} = \frac{\rho_0(r)}{\rho_1(\Psi_r(r))},$$

(5.22)

where $\Psi_r$ and $\Psi_{rr}$ are the first and second derivatives with respect to the radial variable $r$.

Here, the densities $\rho_0$ and $\rho_1$ are piecewise constant, so the right hand side of (5.22) is constant, say $a$. Integrating (5.22) once gives

$$\Psi_r = \sqrt{a} \sqrt{r^2 + \alpha},$$

where $a$ is the ratio $\rho_0/\rho_1$ and $\alpha$ is the integration constant.

Note that our primary goal is to compute the optimal mapping that transfers $\rho_0$ to $\rho_1$ and not the potential $\Psi(r)$ in its entirety. Therefore, from (5.6), we obtain the optimal mapping in terms of $\Psi_r$ as

$$x = \frac{\xi}{r} \Psi_r, \quad y = \frac{\eta}{r} \Psi_r.$$  

(5.23)

The exact solution (5.23) is shown in Figure 5.2.

We use the FDMKP method to compute the numerical solution for the optimal transformation. The numerical and the exact solutions are plotted in Figure 5.2 on a grid of size $33 \times 33$. Notice that the numerical solution is smoother than the exact solution because of the regularization in the numerical discretization. The $L^2$ norm of the error in the numerical solution of for the optimal mapping is $0.0054$. In this example, we use the optimal transformation to deform a circle (template image) to a smaller concentric circle (target image); therefore, the grid points are expected to be displaced radially towards the center of the circles. As expected, the displacement vector field (see Figure 5.2(a)) shows that the grid points have been moved radially towards the center. In Figure 5.1(a), (b) and (d) we show the template, the target and the deformed images, respectively. We show the absolute error between the deformed image and the target image in Figure 5.1(c).
Figure 5.1: Example 5.1. Registration of two concentric circles defined on a $33 \times 33$ grid. Shown here are the (a) template image, (b) reference image, (c) absolute error between the deformed template and the reference images, and (d) deformed template image obtained using the FDMKP method.
Figure 5.2: Example 5.1. Image matching of two concentric circles. Shown here are (a) displacement vector field, (b) numerical solution for the optimal mapping computed using the FDMKP method on a grid of size $33 \times 33$, and (c) exact solution for the optimal mapping. Observe that the numerical solution is smoother than the exact solution due to the regularization used in the numerical discretization.

Example 5.2. In this example we apply the PMKP method to perform an elastic registration of a circle (template) image to a C-shape (target) image (see Figure 5.3 (a) and (b)). The template and the target images are shown in Figure 5.3 and are defined on a $33 \times 33$ grid.

The computed optimal matching mapping and the displacement vector field are given in Figure 5.3 (e) and (d), respectively. The displacement vector field shows that, as expected, the grid points have been moved radially towards the boundary of the circle. The deformed template image and the absolute error between the target and the deformed template are shown in Figure 5.3 (c) and (f).

The interpolation formula (5.9) is used to compute a sequence of deformations of the template image to the targets $\det (\nabla \phi^{-1}) \rho_0 \circ \phi^{-1}$ at $t = 0, 0.25, 0.45, 0.65, 0.85, 1$. Figure 5.4 shows the sequence of the deformations.
Figure 5.3: Example 5.2. Registration of a circle (template) to a C-shape (target) using the PMKP method. Shown are the (a) template image, (b) reference image, (c) deformed template, (d) displacement vector field, (e) computed optimal transformation, and (f) absolute error between the deformed template and the target.
Figure 5.4: Example 5.2. PMKP registration of a circle (template) to a C-shape (target). The pictures here represent the deformations of the template image to the target image at selected intermediate times $t = 0, 0.25, 0.45, 0.65, 0.85, 1$. 
Example 5.3. This is a practical example from medical imaging applications. The template and the target images, shown in Figure 5.5 (a) and (b), are of the Macaque cortex (monkey brain images).

The FDMKP method is applied to register the two images with $79 \times 79$ grid points. Figure 5.5 (d) and (e) presents the displacement vector field and the computed optimal matching transformation. In Figure 5.5 (c) and (f) we show the deformed template image and the absolute error between the target and the deformed images. A sequence of deformations of the template image to the target image at some selected times between $t = 0$ and $t = 1$ are shown in Figure 5.6.
Figure 5.6: Example 5.3. FDMKP registration of Macaque cortex. Shown are a sequence of deformations of the template image to the target image at some selected times between $t = 0$ to $t = 1$. 
Example 5.4. In this example, we apply the our optimal mass transport approach for image registration to register 3D medical images of two heart images. The images are defined on a $240 \times 240 \times 80$ grid. The template and the target images are shown in Figure 5.7 (1) and (2).

We apply the PMKP method to compute the optimal matching transformation that registers the two 3D heart images. In Figure 5.7 (3) we show the deformed template image. In Figure 5.8 we show three slices on the xy, yz, and xz planes of the computed optimal transformation.

Figure 5.7: Example 5.4. Registration of two 3D heart images of size $240 \times 240 \times 80$. Shown are the (1) heart template, (2) target, and (3) deformed template images obtained with the PMKP method.
Example 5.5. In this example, we use the MKP approach to register a 3D heart image (template) to six other 3D heart images (targets). The size of each image is $128 \times 128 \times 128$.

We apply the PMKP method to compute the optimal transformations for the registration of the template image to the six target images. The results of the registrations are shown in Figure 5.9. The template is shown on the first row. The first and the third columns show the six target images, with the second and fourth columns showing the corresponding six deformed images.
Figure 5.9: Example 5.5. 3D registration of a heart (template) to six other hearts (targets) on a 128 x 128 x 128 grid. The template is in the first row. The six pictures on the first and third columns show the targets. The corresponding six deformed templates are shown in the second and fourth columns.
Example 5.6. In this example, we use the MKP registration approach for a 3D registration of a left hippocampus images, a template to eight different targets. The images are defined on a grid of size $40 \times 64 \times 64$.

The PMKP method is applied to compute the optimal transformations for the registration of the template image to the eight target images. Figure 5.10 shows the results of registration of the (top) template to the eight different target images. The first and third columns show the eight target images, and the second and fourth columns show the corresponding eight deformed images.

Example 5.7. Here the PMKP approach is used to register 3D right hippocampus images, a template to eight different targets. The images are defined on a grid of size $40 \times 64 \times 64$.

The optimal transformations are computed with the PMKP method to register the template image to the eight different target images. The results of the registrations are shown in Figure 5.11. Again, the first and third columns show the eight target images, and the second and fourth columns show the corresponding eight deformed images.
Figure 5.10: Example 5.6. 3D registration of a left hippocampus. Shown are template (first row), eight targets (first and third columns), and eight deformed templates (second and fourth columns).
Figure 5.11: Example 5.7. 3D registration of a right hippocampus. Shown are template (first row), eight targets (first and third columns), and eight deformed templates (second and fourth columns).
Example 5.8. To illustrate the effectiveness of our MKP registration approach, we use this approach for the registration of gray-scale images. Specifically, we apply the method to register two gray-scale 3D brain MRI images. The images are defined on a grid of size $80 \times 56 \times 80$.

We compute the optimal transformation using the PMKP method. The results of the registration are shown in Figure 5.12, where the first row represents the registration of the midcoronal slices, the second row corresponds to the registration of the midsagittal slices, and the third row is for the registration of the midhorizontal slices. The first column shows the template image, the second column shows the target image and third column shows the deformed template.

5.4.2 Summary of the Numerical Results

The computational speed of the FDMKP and PMKP methods depends upon the size of the images and the respective number of iterations required to obtain a good approximation. In what follows, all stated run times are for the PMKP, and all computer experiments are run in Fortran 77 on a laptop with a single 1GHz processor and 1Gb of memory. For 2D images of size $32 \times 32$, it takes only 5 to 10 seconds to obtain a very good approximate solution. For 2D images of size $80 \times 80$, it takes from 1 to 3 minutes, and for 3D images of size $128 \times 128 \times 128$, it takes 2 to 3 hours. Several promising methods for speeding this up are currently being investigated as well. Table 5.1 shows average run times and results for a typical 3D dataset. Several metric measures for determining the accuracy of numerical registration have been used in the literature. In this thesis, five of these measures are considered. The overlay error is defined as

$$E_{\text{overlay}} = \frac{1}{|\Omega|} \| \det \left( \nabla \phi^{-1} \right) \rho_0 \circ \phi^{-1} - \rho_1 \|_{L^1},$$

where $\|f\|_{L^1} = \int_{\Omega} |f(x)| dx$, $|\Omega|$ is the area or volume of $\Omega$ (in 2D or 3D, respectively), and $\rho_0$, $\rho_1$ and $\det \left( \nabla \phi^{-1} \right) \rho \circ \phi^{-1}$ are the template, the target and the deformed template images.
Figure 5.12: Example 5.8. 3D registration of gray-scale brain MRI image. Shown are (1), (2) and (3) the midcoronal slices, (4), (5) and (6) the midsagittal slices and (7), (8) and (9) the midhorizontal slices. The template image, the target image, and the deformed template image are shown in the first, second, and third columns, respectively.
The relative error is defined as

$$E_{\text{relative}} = \| \det (\nabla \phi^{-1}) \rho_0 \circ \phi^{-1} - \rho_1 \|_{L^2} / \| \rho_0 - \rho_1 \|_{L^2}. \quad (5.25)$$

A new metric measure is introduced in this thesis is the $L^2$-Kantorovich distance

$$E_K = \inf_{\phi} \left( \int_{\Omega} |\phi(\xi) - \xi|^2 \rho_0(\xi) d\xi \right)^{\frac{1}{2}}. \quad (5.26)$$

In Table 5.2 we show the relative error, the overlay error, and the $L^2$-Kantorovich distance for the examples in the previous subsection.

To examine the quality of the registration images, we also use a similarity measure defined as follows. We first compute the optimal transformation that registers an MRI template image to an MRI target image. We then use this transformation to map a given manually segmented template to generate a segmented image in the target domain — this is referred to as an automated segmentation. In Figure 5.13 and Figure 5.14 we show coronal slices of the manual and automated segmentations outlines overlaid on top of the gray-scale image for left and right hippocampus respectively.

To assess the performance of the MKP methods for image registration, we use manually segmented regions in both the template and the target images which can be found in the Internet Brain Segmentation Repository (IBSR). IBRS is an open online database with 18 MR brain 3D images and their associated segmentations.

To evaluate the accuracy of our registration methods we compute the Kappa overlap metric between the regions segmented using our algorithm and that of the IBRS. The Kappa overlap metric ($\kappa$-metric) [ZDP94] is a similarity measure defined as

$$\kappa(I, J) = \frac{2n(I \cap J)}{n(I) + n(J)}. \quad (5.27)$$

where $n(A)$ is the number of voxels in image $A$. The overlap metric measures the similarity of two sets and ranges from 0 for sets that are disjoint and 1 for sets that are identical. The Kappa values of 0.7 or higher are considered as an excellent agreement (cf.[ZDP94]).

\(^{1}\text{http://www.cma.mgh.harvard.edu/ibsr/}\)
In Tables 5.3 and 5.4 we show the measurements of the Kappa metric of manual and automated segmentations of the IBSR datasets for the MRI left hippocampus of size $42 \times 64 \times 64$ and the MRI right hippocampus of size $44 \times 80 \times 64$ experiments. The average kappa value is 0.75523 with standard deviation 0.037459 for the left hippocampus and is 0.77311 with standard deviation 0.035865 for the right hippocampus. The accuracy is comparable to the results reported in [KWB08, LCSC07, JSTL07], for example in [KWB08] the average kappa value is 0.7115 with standard deviation 0.0577.

The Kappa overlap metric (5.27) is sensitive to both differences in location and size. Therefore we use the volume error, defined for two images $I$ and $J$ with volumes $V_I$ and $V_J$ as

$$E_{\text{volume}} = \frac{|V_I - V_J|}{V_I} \quad (5.28)$$

to measure the difference in size. In Tables 5.3 and 5.4 we show the volume error for 3D datasets for the left and right hippocampus experiments.
Figure 5.13: Three coronal slices of left hippocampus: Segmentation outlines overlaid on top of the gray-scale image. (1), (2), and (3) manual segmentations and (4), (5), (6) automated segmentations.
Figure 5.14: Three coronal slices of right hippocampus: Segmentation outlines overlaid on top of the gray-scale image. (1), (2), and (3) manual segmentations and (4), (5), (6) automated segmentations.
Table 5.1: Typical average run-time results for some 3D data sets for PMKP.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size of dataset</th>
<th>Run time in minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hippocampus</td>
<td>40 × 64 × 64</td>
<td>14.58</td>
</tr>
<tr>
<td>Brain MRI</td>
<td>80 × 56 × 80</td>
<td>31.75</td>
</tr>
<tr>
<td>Heart</td>
<td>128 × 128 × 128</td>
<td>126.73</td>
</tr>
</tbody>
</table>

Table 5.2: Overlay error, relative error, and $L^2$-Kantorovich distance for some sample 3D datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Overlay Error</th>
<th>Relative error</th>
<th>$L^2$-Kantorovich distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>two concentric circles</td>
<td>0.0010</td>
<td>0.24%</td>
<td>0.0305</td>
</tr>
<tr>
<td>circle to C-shape</td>
<td>0.0003</td>
<td>0.06%</td>
<td>0.01031</td>
</tr>
<tr>
<td>macaque</td>
<td>8.8718 × 10^{-9}</td>
<td>2.6691 × 10^{-8}</td>
<td>0.0018± 0.0003</td>
</tr>
<tr>
<td>left hippocampus</td>
<td>0.0008 ± 0.0005</td>
<td>1.37 % ± 0.80%</td>
<td>0.0018± 0.0003</td>
</tr>
<tr>
<td>right hippocampus</td>
<td>0.0010 ± 0.0006</td>
<td>1.77 % ± 0.95%</td>
<td>0.0020± 0.0003</td>
</tr>
<tr>
<td>heart</td>
<td>0.0060 ± 0.0018</td>
<td>5.46% ± 1.23%</td>
<td>0.0193 ± 0.0041</td>
</tr>
<tr>
<td>brain (mri)</td>
<td>0.0113</td>
<td>7.11%</td>
<td>0.0088</td>
</tr>
</tbody>
</table>

Table 5.3: Left hippocampus: Kappa overlap metric and volume error. The mean value for Kappa overlap is 0.75523 with standard deviation 0.037459. The mean value for volume error is 0.21362 with standard deviation 0.17166.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Kappa</th>
<th>Volume Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBSR01/ir</td>
<td>0.76</td>
<td>0.10</td>
</tr>
<tr>
<td>IBSR02/ir</td>
<td>0.75</td>
<td>0.02</td>
</tr>
<tr>
<td>IBSR03/ir</td>
<td>0.68</td>
<td>0.03</td>
</tr>
<tr>
<td>IBSR04/ir</td>
<td>0.81</td>
<td>0.04</td>
</tr>
<tr>
<td>IBSR05/ir</td>
<td>0.75</td>
<td>0.01</td>
</tr>
<tr>
<td>IBSR06/ir</td>
<td>0.76</td>
<td>0.25</td>
</tr>
<tr>
<td>IBSR07/ir</td>
<td>0.76</td>
<td>0.05</td>
</tr>
<tr>
<td>IBSR08/ir</td>
<td>0.72</td>
<td>0.48</td>
</tr>
<tr>
<td>IBSR09/ir</td>
<td>0.75</td>
<td>0.35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Kappa</th>
<th>Volume Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBSR11/ir</td>
<td>0.81</td>
<td>0.11</td>
</tr>
<tr>
<td>IBSR12/ir</td>
<td>0.78</td>
<td>0.07</td>
</tr>
<tr>
<td>IBSR13/ir</td>
<td>0.78</td>
<td>0.24</td>
</tr>
<tr>
<td>IBSR14/ir</td>
<td>0.79</td>
<td>0.3</td>
</tr>
<tr>
<td>IBSR15/ir</td>
<td>0.77</td>
<td>0.3</td>
</tr>
<tr>
<td>IBSR16/ir</td>
<td>0.69</td>
<td>0.51</td>
</tr>
<tr>
<td>IBSR17/ir</td>
<td>0.77</td>
<td>0.30</td>
</tr>
<tr>
<td>IBSR18/ir</td>
<td>0.70</td>
<td>0.45</td>
</tr>
</tbody>
</table>
Table 5.4: Right Hippocampus: Shown here are Kappa overlap metric and the volume error. The mean value for Kappa overlap is 0.77311 with standard deviation 0.03565. The mean value for the volume error is 0.19449 with standard deviation 0.14413.
Chapter 6

Concluding Remarks and Future Directions

In this thesis we have presented two adaptive mesh methods, the PMKP and the FDMKP methods, based on solving the optimal mass transport problem, more precisely the $L^2$ Monge-Kantorovich problem. For the PMKP method, the coordinate transformation for the mesh adaptation is computed as the steady-state solution of the parabolic Monge-Ampère equation. In the FDMKP method, the mesh movement is controlled by the velocity field obtained as the minimizer of a cost functional defined for the $L^2$ Monge-Kantorovich problem in a fluid dynamics framework.

We have also presented some theoretical results for the existence and convergence of the process of solving the parabolic Monge-Ampère equation to steady-state. A derivation of a good stopping criterion for the convergence of the solution has been studied.

To illustrate the performance of the PMKP and the FDMKP methods, we ran several 2D and 3D computations to generate adaptive meshes. We also showed some numerical experiments to compare the two methods with the GCL method of Cao, Huang and Russell [CHR02] and the PMA method of Budd and Williams [BW06]. We also provide a detailed description to compute a numerical solution of an initial-boundary value problem for Burgers’ partial differential equation in two spatial dimension. The results show that the proposed methods are effective for accurately
concentrating the mesh points in the proper regions.

We have described two methods of elastic image registration based on the approaches used for computing the coordinate transformation for mesh adaptation. The performance of these methods is demonstrated with various numerical experiments for 2D and 3D elastic image registration. A number of numerical examples for registering 3D medical images are also included.

From the known theory for the Monge-Kantorovich problem, existence and uniqueness of the optimal transformation are guaranteed, helping to assure that our methods are both robust and reliable for both applications, adaptive grid generation and image registration.

There is a major difference between the MKP adaptive mesh methods presented in this thesis and the standard adaptive moving mesh methods in their approach for computing the adaptive mesh suitable for solving a time-dependent partial differential equation. The difference is that, for the MKP methods, at each time step the adaptive mesh is generated independently of the adaptive mesh generated in the previous time step due to the fact the two meshes are obtained as a result of solving two different $L^2$ Monge-Kantorovich problems.

The FDMKP method for both adaptive grid generation and image registration, described Section 3.2 and Section 5.3, can in theory be applied in any dimension. However, in this thesis we only successfully applied the FDMKP method for the 2D problems. The limitation of applying the method for 3D problems is due to the fact that the ALG2 algorithm introduced to solve the saddle point problem for the FDMKP method involves solving a 4-dimensional Poisson equation (three spatial variables and the time variable) which introduces substantial computational difficulties. In contrast, we have successfully implemented the PMKP method for both 2D and 3D adaptive grid generation and image registration.

Our approach for solving equation (5.8) bears a close relationship to that of [AHT03], for which a gradient descent method is introduced to solve for the optimal mapping. However, the method in [AHT03] requires finding an initial mapping satisfying (5.3) and solving a Poisson equation at each step time, which would make the method very computationally expensive for large 3D registration problems. In
contrast, our PMKP approach requires no construction of an initial mapping (always using a uniform initial mesh) and does not involve solving a Poisson equation at each stage.

The results for the PMKP and FDMKP methods, for both adaptive grid generation and image registration, are promising. However, a number of practical issues are yet to be addressed. For example, it is necessary to define a suitable adaptation function to control the mesh adaptation near the boundaries. Investigation is needed for finding other more efficient stopping criteria for the convergence of the parabolic Monge-Ampère equation solution. A more sensible interpolation formula is needed for finding the in between images when performing image registration using the PMKP method.

In the future, we intend to consider other techniques for solving the original Monge-Ampère equation (3.79) in order to develop the most suitable implementations for real world physical problems and medical applications. A joint work with Oberman on computing the coordinate transformation for mesh adaptation and image registration, from directly solving the Jacobian equation with some constraints on the curl of the transformation, is currently under consideration. An interesting direction is to develop techniques of computing the coordinate transformation in three spatial dimension using the FDMKP method. Finally, there is certainly a greater need of investigating various techniques to improve the performance of the proposed methods for both adaptive grid generation and image registration applications so that they can be efficiently used for a wider range of challenging problems.
Bibliography


BIBLIOGRAPHY


Index

Adaptation function, 2
Adaptation function
  arc-length, 12
  curvature, 14
  gradient, 12
Adaptive mesh
  FDMKP method, 44
  PMA method, 21
  PMKP method, 32, 41
Adaptive mesh
  GCL method, 18
Adaptive methods, 1
ALG2 algorithm, 47
augmented Lagrangian, 47
augmented Lagrangian
  method, 46
Brenier theorem, 25
Burgers’ equation, 80
convex
  potential, 25
Equidistribution
  in higher dimensions, 17
Equidistribution principle, 9
Equidistribution principle
  in higher dimensions, 20
  in one dimension, 10
fluid dynamics formulation, 26
Geometric conservation law
  integral form, 18
Hamilton-Jacobi equation, 45
Helmholtz’s decomposition theorem, 20
Hessian matrix, 22
Image registration, 3, 27
Image registration
  AHT method, 28
  LDDMM method, 30
Kantorovich–Wasserstein
  \( L^2 \) distance, 29
  \( L^p \) distance, 24
Kantorovich–Wasserstein
  \( L^2 \) distance, 5
Kappa overlap metric, 113
Legendre transformation, 22
mesh quality measure, 55
method of lines, 53
Monge–Kantorovich
  \( L^2 \) problem, 26
  \( L^p \) problem, 24
  problem, 4
Monge–Kantorovich
  \( L^2 \) problem, 5
Monge-Ampère equation, 25, 34
parabolic Monge-Ampère equation, 22
parabolic Monge-Ampère equation, 5
physical PDE, 49
PMKP Registration Method, 93
INDEX

registration problem, 28
saddle-point problem, 46
similarity measure, 112
similarity measure, 91
spatial discretization, 50
stopping criterion, 35