Mixed Variational Bayesian Learning Methods and Inverse Prediction of Continuum Deformation Mappings

by

Chao Wang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Shaofan Li, Chair
Professor Khalid M. Mosalam
Associate Professor Adityanand Guntuboyina

Summer 2022

Mixed Variational Bayesian Learning Methods and Inverse Prediction of Continuum Deformation Mappings

Abstract

Mixed Variational Bayesian Learning Methods and Inverse Prediction of Continuum Deformation Mappings

by

Chao Wang

Doctor of Philosophy in Civil and Environmental Engineering

University of California, Berkeley

Professor Shaofan Li, Chair

In this dissertation, I present a computational framework to predict the inverse deformation mapping. Under this framework, two algorithms have been developed. The first one is called the material deformation finding (MDF) algorithm with a specific application in additive manufacturing. This method is designed to quantify the permanent (non-zero strain) continuum/material deformation. Different from physical-based modeling, the method developed here is based on a data-driven statistics approach, which solves the problem without needing information about the physical deformation process. The proposed method relies only on the scanned material data from the thermal distorted configuration as well as the shape of the initial design configuration. In this work, the MDF algorithm was first validated by a 2D synthetic example. We then demonstrate that the proposed MDF method can accurately find the permanent thermal distortion of a complex 3D printed structural component, and hence identify the thermal compensation design configuration. The results obtained in this work indicate that one can use this data-driven statistics approach to significantly mitigate the thermal distortion of 3D printed products in additive manufacturing.

The second algorithm developed is a mixed variational Bayesian learning finite element method (VBL-FEM), based on a Bayesian statistical continuum mechanics theory, in which elastic potential energy is used as a prior in a Bayesian regularization network, which can intelligently recover unknown continuum deformation mapping with only the information of the shapes of the deformed and undeformed continuum body without knowing actual boundary conditions, both traction as well as displacement boundary conditions and the precise material constitutive relation. Moreover, we also develop the related finite element formulation in a computational probabilistic mechanics framework. Using a data-driven likelihood function, we construct an entropic variational principle of continuum mechanics based on the maximum a posteriori (MAP) probability estimation and Bayesian regularization. By solving the probabilistic Galerkin variational problem, we also demonstrate in several exam-

ples that the proposed method is able to inversely predict continuum deformation mappings with strong discontinuity or fracture without knowing the external load conditions. This long-sought-after inverse problem solution has been a major challenge in structure failure forensic analysis in the past several decades, and the proposed method provides a machine intelligent solution for it.

To my parents.

Contents

Contents **ii**

List of Figures **iv**

List of Tables **vii**

1 Introduction **1**
1.1 Motivation and Backgrounds 1
1.2 Overview of the Dissertation 4

2 Related Work and Problem Statements **6**
2.1 Forward and Inverse Problem Statements 6
2.2 Variational Form and Weak Form in Mechanical Problems 9
2.3 Synopsis of Finite Element Method 12
2.4 Overview of Related Statistics Methods 15

3 Material Deformation Finding Algorithm **23**
3.1 Overview of the Distortion Compensation Methods in AM 23
3.2 Motivation for the MDF algorithm 24
3.3 Material Deformation Finding Algorithm 29
3.4 Accelerated Computing Implementation 33

4 A Mixed Variational Bayesian Learning Finite Element Method **37**
4.1 A Mixed Entropic Variational Principle of Probabilistic Elastic Continuum . 37
4.2 Variational Bayesian Learning Finite Element Formulation and Implementation 43

5 Numerical Examples **48**
5.1 MDF Experiments and Results 48
5.2 VBL-FEM Numerical Examples 59

6 Conclusions and Outlook **78**

Bibliography **80**

A The minimizer of the variational principle **87**

B Derivative of $\mathcal{L}_{QR}(\mathbf{V}, \sigma^2)$ with respect to $\boldsymbol{V}$ **90**

List of Figures

1.1 Schematic illustration of deformation mapping of a continuum solid. 2
1.2 An example of thermal distorted 3D printed product: Distortion on a 3D printed Inconel competent heated only from below (left) versus the same Inconel component that was made when heated from above and below (right). 3

2.1 Schematic of a boundary-value problem in solid mechanics. 8
2.2 Comparison between the boundary-value problem and its inverse problem. . . . 9
2.3 Quadrilateral element in the geometric and natural coordinate systems. 13
2.4 Gaussian mixture model example: (a) Histogram of data; (b) Gaussian mixture model fitted by maximum likelihood estimation. 17
2.5 The comparison between the non-rigid registration method and material deformation finding method: (a) Non-rigid registration problem in computer vision vs. (b) Material deformation finding problem in 3D printing and continuum mechanics 19
2.6 (a) A coherent registration between two point sets. (b) A less coherent registration between two point sets. 22

3.1 Material deformation finding for a deformed two-dimensional unit square: (I) Deformation of scanned data from the initial configuration to deformed configuration; (II) Deformation of designed mesh; (III) Topology connection between undeformed design mesh and scanned data in deformed configuration, and (IV) The exact solution for the material deformation finding problem. 27
3.2 An 1D example under uniform deformation: (a) Matching result based on the non-rigid point registration of CPD algorithm, and(b) The true underlying deformation of scanned data. 28
3.3 GMM density function: (a) non-uniform mesh with equal membership probability. (b) non-uniform mesh with specific membership probability for each member. (c) uniform mesh with equal membership probability. 29
3.4 Computation complexity comparison between unaccelerated and accelerated algorithm. 36

4.1 Schematic illustration of the inverse solution of continuum deformation mapping recovery problem. 38

4.2 Flowchart for VBL-FEM inverse recovery algorithm for continuum deformation mapping. 44

5.1 2D point sets registration results. (a) three different nonrigid deformations before registration. (b) the registration results of CPD algorithm. (c) the registration results of material deformation finding algorithm. 49
5.2 (a) Average registration errors for four corner points, and (b) Average registration errors for all data points. 50
5.3 Two lattice models: (a). simple cubic lattice; (b). body-centered cubic lattice; (c). 103032 scatter nodes of simple cubic lattice; (d). 247720 scatter points of body-centered cubic lattice. 51
5.4 The MDF method and the CPD method comparison for simple cubic lattice model: (a). MATLAB default view of design model (blue nodes) and transformed manufactured model (red nodes); (b). front view of design model and transformed manufactured model; (c). MATLAB default view of CPD method registered manufactured model (red nodes) and design model (blue nodes); (d). front view of CPD method registered manufactured model and design model; (e). MATLAB default view of MDF method registered manufactured model (red nodes) and design model (blue nodes); (f). front view of MDF method registered manufactured model and design model. 53
5.5 The MDF method and the CPD method comparison for body-centered cubic lattice model: (a). MATLAB default view of design model (blue nodes) and transformed manufactured model (red nodes); (b). front view of design model and transformed manufactured model; (c). MATLAB default view of CPD method registered manufactured model (red nodes) and design model (blue nodes); (d). front view of CPD method registered manufactured model and design model; (e). MATLAB default view of MDF method registered manufactured model (red nodes) and design model (blue nodes); (f). front view of MDF method registered manufactured model and design model. 54
5.6 Flowchart of the compensation plan with the new proposed MDF algorithm. . . 55
5.7 (a) Referential mesh of the design model in the initial or material configuration, and (b) Spatial mesh and spatial material nodes of theMDF/ scanned data. . . 56
5.8 (a) Two feature point sets before the registration. (b) The registration result of two feature point sets. 58
5.10 A summary of the thermal compensation of the double cantilever comb beam part. Note that the photos of the printed parts are real images of the printed products, and the small defects are due to the printing process fluctuation not from the thermal compensation algorithm. 59
5.12 The finite element meshes for the thin plate example: (a) Mesh in the initial configuration; (b) The same mesh in the final configuration; (c) Coarse mesh in the final configuration. 60
5.9 Redesigned model with KNN Regression. 61

5.11 The setup of the thin plate with a hole example. 61
5.13 Three different setups for the VBL-FEM model with the corresponding matching results for example 1. 62
5.14 The displacement recovery process for case 1 in example 1. 64
5.15 (a) The history of average nodal absolute error and standard deviation for case 1 in example 1; (b) The distribution of nodal absolute errors for the final result (iteration 147) for case 1 in example 1. 65
5.16 The displacement recovery process for case 2 in example 1. 66
5.17 (a) The history of average nodal absolute error and standard deviation for case 2 in example 1; (b) The distribution of nodal absolute errors for the final result (iteration 147) for case 2 in example 1. 67
5.18 The displacement recovery process for case 3 in example 1. 68
5.19 (a) The history of average nodal absolute error and standard deviation for case 3 in example 1; (b) The distribution of nodal absolute errors for the final result (iteration 147) for case 3 in example 1. 69
5.20 The setup of the ductile fracture of a thin plate. 70
5.21 The finite element meshes ductile fracture of a thin plate: (a) Mesh in the initial configuration; (b) The same mesh in the final configuration; (c) Coarse mesh in the initial configuration. 70
5.22 The displacement recovery process for the ductile fracture of a plate. 72
5.23 (a) The history of average nodal absolute error and standard deviation for the ductile fracture of a plate; (b) The distribution of nodal absolute errors for the final result (iteration 200) for the ductile fracture of a plate. 73
5.24 The setup of the thin shell cylinder. 74
5.25 The finite element meshes for the thin shell cylinders: (a) Mesh in the initial configuration; (b) A finer mesh in the initial configuration; (c) Mesh in the final configuration. 74
5.26 The displacement recovery process for the thin shell cylinder. 76
5.27 (a) The history of average nodal absolute error and standard deviation for the thin shell cylinder; (b) The distribution of nodal absolute errors for the final result. 77

List of Tables

2.1 Table of Notations in CPD 18

3.1 Table of Notations in material deformation finding algorithm 26
3.2 Comparison between the point set registration methods and the material deformation finding algorithm. 28

5.1 Values of hyperparameters for all three cases. 49
5.2 Simulation Parameters in ANSYS Additive manufacturing 52
5.3 Values of process parameters. 55
5.4 Values of design model dimensions. 55
5.5 The VBL-FEM model hyperparameters for example 1. 62
5.6 The comparison of displacement recovery accuracy between three cases. 69
5.7 The VBL-FEM model parameters for example 2. 71
5.8 The VBL-FEM model parameters the thin shell cylinder example. 75

Acknowledgments

I would like to express my sincere thanks to Professor Shaofan Li for his guidance and support. His patience and inspiration help me on every step I need to make and overcome the difficulties in research as well as in life.

I also wish to thank Professor K. Mosalam and Professor A. Guntuboyina for serving on my thesis committee. The finite element method courses that I took from Professor F. Armero and probability course that I took from Professor A. Guntuboyina give me the theoretical foundation and mathematical implementation skills for my research. I also owe my thanks to Professor F. Armero for his great courses offered which is unquestionably a large wealth in my life.

I would like to thank my colleagues Dandan Lyu, Qi Zheng, Chengyao Liang, Xuan Hu, Tiange Li and Yuxi Xie among others for their help and kindness.

I would like to thank my parents for their love and support. Without them, this day would not come. I would also like to thank my entire family for their love and supports in my life.

I also owe my thanks to Ford Motor Company and Harbin Engineering University for the grants to support my research.

Chapter 1

Introduction

1.1 Motivation and Backgrounds

In continuum solid mechanics, under external loading, a continuum object deforms from the referential configuration Ω_X to a deformed configuration Ω_x. This deformation process can be described by a continuous function $\psi : \Omega_X \to \Omega_x$, which is called as the deformation map or mapping (see Fig. 1.1). The fundamental problem of continuum mechanics is to find the deformation mapping under given boundary or loading conditions. Boundary value problem solutions in continuum mechanics have been well developed including analytical solutions as well as numerical solutions, such as finite element solutions [98].

However, in many engineering applications, we do not know the boundary conditions, and in many cases we do not even know the precise material properties [15]. This is because in many engineering applications, those information are either unknown or incomplete. For instance, after a traffic accident, one can have the image of the car wreckage configuration without knowing the details of the car collision process, and thus it is impossible to find the crash deformation map by using forward analysis or calculation. The only known information or data available is the deformed configuration of the continuum object, which may be obtained by various image processing procedures. Nevertheless, we would like to know the precise deformation field of the continuum object, so that we can find the strain and maybe even stress field of the continuum. This is the fundamental inverse problem of continuum mechanics.

Another major application for the inverse problem is in additive manufacturing (AM). AM has become an emerging technique in various engineering and technology disciplines as a result of its many technical benefits, which include unprecedented design flexibility and low lead times [38, 30]. Three-dimensional (3D) printing technology is distinguished by its ability to directly manufacture complicated structures and reduce the waste associated with traditional production [17, 9, 60]. During the production process that uses 3D printing, the level of geometric complexity has no effect on the amount of construction efficiency. Therefore, no extra work is necessary to design the molding structure, which makes 3D

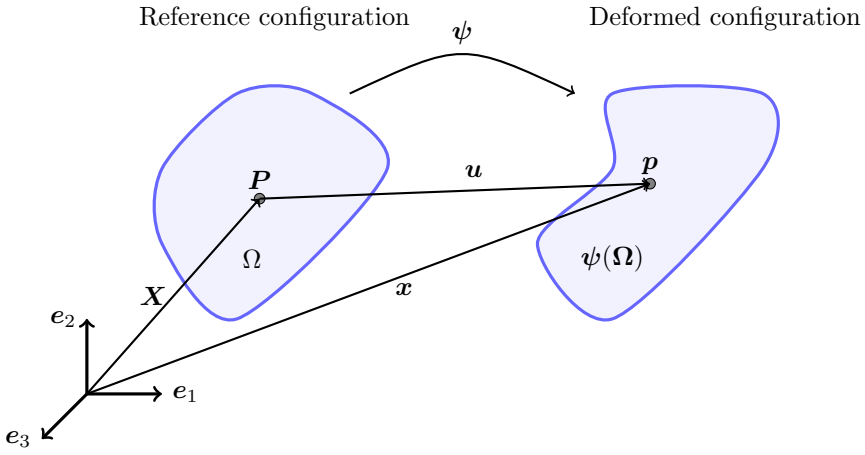


Figure 1.1: Schematic illustration of deformation mapping of a continuum solid.

printing a highly promising production approach [62, 92]. In light of the bright prospects it presents, research efforts have been stepped up across the board in the manufacturing industry, as well as in the fields of material science, computational mechanics, and artificial intelligence [54, 11, 95].

Despite these promising aspects, one of the primary problems for large-scale industrial uses of 3D printing technology is figuring out how to minimize or limit structural distortion of the items that are formed during 3D printing processes, as illustrated in Fig. 1.2. Because of this structural distortion, the benefits of accuracy and production in a single step are significantly reduced. Melting materials into a liquid or mushy condition is often the first step in the normal 3D printing process. Next, the melted material is swiftly prototyped into a variety of different specified forms. This procedure includes a significant increase or decrease in temperature, which causes the material to shrink, as well as a variety of thermally-induced permanent deformations and any residual stresses that were created during the printing process. This is not a straightforward thermal-mechanical procedure. It is a complicated interaction between and within the material layers in various stages and temperatures. Variations and oscillations in the underlying process parameters control its behavior [89, 76, 22].

The inverse problem-solving method may be used to assess and correct thermal distortion in AM [44, 5] systems. For instance, this new compensation strategy may consist of two parts. First, post-process measurements make it simple and accurate to determine the deformed shape of a printed item. Second, the inverse problem-solving technique may be used to detect and quantify the distortion field and develop a new design with compensation.

In this particular scenario, the inverse forensic analysis [20] may be carried out once it has been determined how to infer the missing information, such as the pre-crash speed and angle, by the use of machine learning (ML) techniques that are based on the inverse analysis, for example. However, in order for the ML-based inverse solution to operate, the permanent

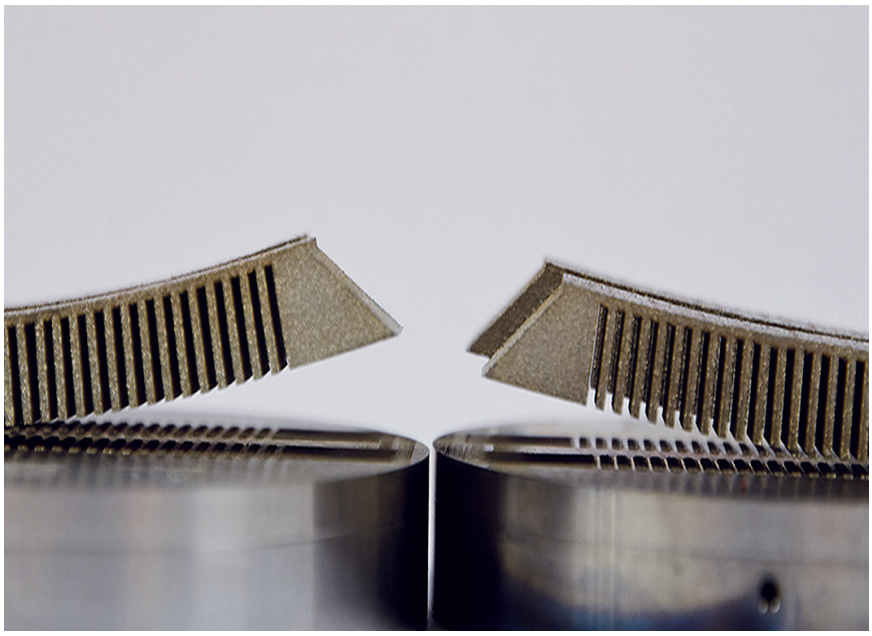


Figure 1.2: An example of thermal distorted 3D printed product: Distortion on a 3D printed Inconel competent heated only from below (left) versus the same Inconel component that was made when heated from above and below (right).

plastic deformation field must still be sent to the deep learning neural network as an input. Despite the significant progress that has been made in the forward computation of solid mechanics, the inverse prediction continues to be a difficult subject [16, 86].

Over the course of many decades, a variety of theoretical and numerical approaches to solving inverse problems have been developed. For instance, Alves et al. [2] came up with a theoretical formulation to describe the inverse scattering for elastic plane fractures, and they proposed it. Ballard et al. [6] spoke about reconstructing the stress field based on the observations of the stress on the boundary. A mathematical-physical framework was provided by Weglein et al. [90] for the inverse scattering series. Guzina et al. [35] developed a different analytical framework, which they called the inverse scattering of elastic waves.

In recent times, due to the rapid advancements made in fields like as artificial intelligence and data science, techniques that are based on machine learning have garnered a lot of attention. Deep neural networks [53], also known as DNN, have the best performance of all machine learning techniques across a wide variety of applications, including computer vision (CV) [87] and natural language processing (NLP) [39].

Besides the achievements in computer science, DNN is also applied to the inverse problem solutions in mechanics. Chen et al [19] proposed a novel deep learning inverse solution to determine the loading conditions of shell structure. After training the DNN, the loading can be identified by the information of the plastic deformation of a shell structure. Ni et

al [69] discussed the application of DNN on identifying the elastic modulus based on the measured strain field. By collecting typical representative of shear modulus, the trained DNN can learn the mapping between stress and strain with high accuracy. A deep energy approach [80, 68, 50] was also proposed to identify the displacement field by structual energy informed DNN. In the field of computational mechanics, in addition to feed forward neural networks, an image-based concurrent neural network (CNN) [33] was also included in the research. A unified solution for both forward and inverse problems was developed by Gao et al [31]. using a physics-informed graph neural Galerkin network. Piecewise polynomial basis function was used to interpolate the search space, as opposed to standard physics-informed neural networks (PINNs) that have insufficient scalability and strict boundary enforcement.

The speed of offline training is one of the primary obstacles to the implementation of deep learning algorithms in mechanics. Another obstacle is the difficulty of interpreting the physical meaning of the model. In general, DNN functions as a black box inside the framework. One might easily adapt it to a certain case. However, it is challenging to explain and use the application in a broader context. In order to address these drawbacks, approaches based on the Gaussian mixture model (GMM) [79] have recently been used to inverse analysis. The current non-rigid registration method is not designed to inversely recover the three-dimensional continuum deformation mapping in engineering applications, where many deformations may be discontinuous or have strong discontinuities. And it is limited to the the one-to-one matching process of data points on the peripheral or configuration boundary of a continuum object.

1.2 Overview of the Dissertation

The aim of this work is to construct a framework for the inverse prediction of continuum deformation mappings. In order to do so, two different methods are developed which are the material deformation finding (MDF) and variational Bayesian learning finite element method (VBL-FEM). The MDF algorithm described in this dissertation have been published by the dissertation's author previously in [88].

The organization of this thesis is given as follows.

Chapter 1 describes the motivation and background of the dissertation. The background and the significance of the inverse problem in mechanics are introduced. In particular, the application of the inverse deformation finding to additive manufacturing is discussed in detail in this chapter. Then the limitations of current models both the numerical simulation based and the machine learning based are discussed.

Chapter 2 provides a description and summary of both the mechanical concepts and statistical concepts. For the concepts in mechanics, it introduces and summarizes the inverse problem statement, the weak form and variational form of the mechanics problem, and the finite element method (FEM). For the concepts in statistics, it presents the maximum likelihood estimation (MLE), Gaussian mixture model (GMM), expectation-maximization (EM) algorithm, and coherent point drift (CPD) algorithm.

Chapter 3 introduces the MDF algorithm. This MDF algorithm is developed to quantify the compensation of the thermal distortion in additive manufacturing. The motivation and application of this algorithm are explained. Comparisons between the MDF and CPD algorithms are also discussed.

In chapter 4, we developed the VBLL-FEM based on a Bayesian statistical continuum mechanics theory, in which elastic potential energy is used as a prior in a Bayesian regularization network, which can intelligently recover unknown deformation mapping without knowing external traction loading conditions as well as the displacement boundary condition. Moreover, we have developed the related finite element formulation in a computational probabilistic mechanics framework.

In chapter 5, several numerical examples are performed and discussed. First, the MDF algorithm is validated and compared by a synthetic 2D example under non-uniform deformation. Then, it is applied to quantify the deformation fields for additive manufacturing for both the lattice structure and the double cantilever bean structure. Several numerical examples for the VBL-FEM algorithm are also discussed in this chapter. Those examples include both the elasticity and plasticity problems in both 2D and 3D.

This dissertation is summarized in chapter 6. Comments and perspectives on the MDF and VBL-FEM algorithms are also given.

Chapter 2

Related Work and Problem Statements

This chapter discusses the related work and problem statement. It first introduces the mechanical problem statement and the inverse of it. Then, the alternative forms and numerical methods of the mechanical problem are also discussed. Finally, several useful statistical methods are introduced for self-containedness.

2.1 Forward and Inverse Problem Statements

Forward Problem Statement

Let Ω and $\boldsymbol{\psi}(\Omega)$ be the reference configuration and current configuration in $\mathbb{R}^n$ respectively (shown in Fig. 1.1). Points $\boldsymbol{X} \in \Omega$ are called material points. Then a deformation mapping $\boldsymbol{\psi}$ is defined as a mapping from Ω to $\boldsymbol{\psi}(\Omega)$, which assigns a new position $\boldsymbol{x}$ to each material point $\boldsymbol{X}$ in Ω as,

$$\boldsymbol{x} = \boldsymbol{\psi}(\boldsymbol{X}). \tag{2.1}$$

Therefore, the reference and current configurations can also be denoted as Ω_X and Ω_x respectively.

Given boundary conditions and a constitutive model, for infinitesimal deformation, the conventional continuum mechanics can determine the deformation mapping or the associated displacement field by solving the following boundary-value problem (BVP) that can be summarized below:

Infinitesimal Mechanical Problem (Strong form).
Find $\boldsymbol{\psi} : \Omega \longrightarrow \boldsymbol{\psi}(\Omega) \subset \mathbb{R}^n$ and the corresponding displacement field $\mathbf{u} = \boldsymbol{\psi}(\mathbf{X}) - \mathbf{X}$, satisfying

$$\begin{aligned} &\operatorname{div}[\boldsymbol{\sigma}(\mathbf{u})] + \mathbf{b} = \mathbf{0} && \text{in} \quad \Omega_X \ , \\ &\mathbf{u} = \overline{\mathbf{u}} && \text{on} \quad \Gamma_u \ , \\ &\boldsymbol{\sigma} \cdot \mathbf{n} = \overline{\mathbf{t}} && \text{on} \quad \Gamma_t \ , \end{aligned} \tag{2.2}$$

where $\mathbf{b}$ is the body force per unit volume; $\overline{\mathbf{t}}$ is the prescribed traction on the traction boundary, and $\overline{\mathbf{u}}$ is the imposed displacement on the displacement boundary. The stress $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{u})$ is determined by the given constitutive model, and the strain field is defined as $\boldsymbol{\varepsilon}(\mathbf{u}) = \mathbf{Sym}\nabla \otimes (\mathbf{u})$.

Note that in Eq. (2.2) $\Omega \in \mathbb{R}^n$ (for $n = 1, 2, 3$) is a continuous bounded domain with essential boundary condition $\Gamma_{\boldsymbol{u}}$ and natural boundary condition $\Gamma_{\boldsymbol{t}}$. In order to have a well-posed problem, all points on the boundary Γ must have one but only one type of boundary conditions. This can be expressed by Eq. (2.3),

$$\Gamma_{\boldsymbol{u}} \cap \Gamma_{\boldsymbol{t}} = \varnothing \quad \text{and} \quad \Gamma_{\boldsymbol{u}} \cup \Gamma_{\boldsymbol{t}} = \partial\Omega \ . \tag{2.3}$$

Next let us consider the three-dimensional cases. If the stress tensor is represented by Cartesian component

$$[\boldsymbol{\sigma}] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \ , \tag{2.4}$$

then its divergence can be expressed as

$$[\operatorname{div}(\boldsymbol{\sigma})] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \sigma_{11,1} + \sigma_{12,2} + \sigma_{13,3} \\ \sigma_{21,1} + \sigma_{22,2} + \sigma_{23,3} \\ \sigma_{31,1} + \sigma_{32,2} + \sigma_{33,3} \end{bmatrix} \ , \tag{2.5}$$

where $x_j, j = 1, 2, 3$, are the Cartesian components of $\mathbf{x}$, and $\sigma_{ij,k} = \frac{\partial \sigma_{ij}}{\partial x_k}$.

By assuming a linear elastic isotropic material, the constitutive relationship between the Cauchy stress $\boldsymbol{\sigma}$ and infinitesimal strain $\boldsymbol{\varepsilon}$ can be expressed as

$$\boldsymbol{\sigma} = \frac{\partial W}{\partial \boldsymbol{\varepsilon}} = \lambda \operatorname{tr}[\boldsymbol{\varepsilon}]\mathbf{I} + 2\mu\boldsymbol{\varepsilon}, \tag{2.6}$$

where $W = W(\boldsymbol{\varepsilon})$ is the strain energy density, $\mathbf{I}$ is the identity tensor, while λ and μ are with Lamé constants.

$[\nabla \otimes \mathbf{u}]$ is the gradient of $\mathbf{u}$ expressed in Cartesian component form as

$$[\nabla \otimes \mathbf{u}] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} \end{bmatrix} = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ u_{3,1} & u_{3,2} & u_{3,3} \end{bmatrix} \ , \tag{2.7}$$

where $u_i, i = 1, 2, 3$ are the Cartesian components of the displacement $\mathbf{u}$ and $u_{i,j} = \frac{\partial u_i}{\partial x_j}$. Therefore, the strain tensor can be represented as

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} = \begin{bmatrix} u_{1,1} & \frac{1}{2}(u_{1,2} + u_{2,1}) & \frac{1}{2}(u_{1,3} + u_{3,1}) \\ \frac{1}{2}(u_{1,2} + u_{2,1}) & u_{2,2} & \frac{1}{2}(u_{2,3} + u_{3,2}) \\ \frac{1}{2}(u_{1,3} + u_{3,1}) & \frac{1}{2}(u_{2,3} + u_{3,2}) & u_{3,3} \end{bmatrix}. \tag{2.8}$$

An example of a 2D boundary value problem is shown in Fig. 2.1.

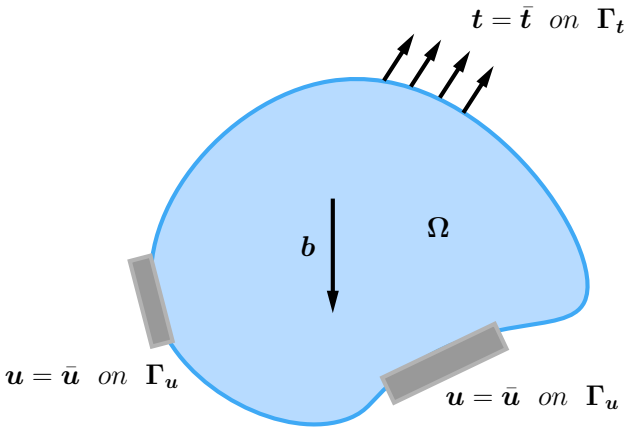


Figure 2.1: Schematic of a boundary-value problem in solid mechanics.

In the BVP of the Navier equation of linear elasticity, displacement field $\mathbf{u}(\mathbf{X})$ is the only unknown. To solve the linear elasticity boundary value problem stated in Eq. (2.2), the boundary conditions, external loads, material constants and reference configuration must be given. These conditions are the inputs to a chosen partial differential equation (PDE) solver, e.g. finite element method (FEM).

Inverse Problem Statement

The inverse problem referring to the above forward solution is that we aim to solve the above problem without knowing the boundary conditions including the precise information on Γ_u and Γ_t as well as the precise material constitutive relations. The only information that we have is the referential configuration of undeformed solid body, or the shape of the undeformed body, Ω and the the deformed configuration of, or the shape of, the solid body Ω. Mathematically, we seek to find an one-to-one deformation mapping such that

$$\mathbf{x} = \boldsymbol{\psi}(\mathbf{X}), \quad \forall \ \mathbf{x} \in \Omega_x, \text{ and } \forall \ \mathbf{X} \in \Omega_X. \tag{2.9}$$

In passing we note that $\boldsymbol{\psi}(\Omega)$ needs not to be continuous everwhere, and it may contain strong discontinuity, such as cracks. Since the displacement field determines the deformation map, we need to find the displacement field in this inverse problem setting, in which the

external loading conditions are unknown, and the displacement boundary conditions are also unknown. The only available information is the general shapes of the reference configuration and current configuration, as described in Fig. 2.2. The comparison between the linear elastic BVP and its inverse deformation map recovery problem is illustrated in Fig. 2.2. In applications, this type of problems often occurs in material and structure forensic analysis, structure failure analysis, as well as general deformation measurements in complex engineering setting such as nanoscale deformation measurement, determining plastic deformation of a crashed vehicle, determine 3D printed product shape distortion, and hazard environment deformation probe, among many others. In the above situations, we only have the image or scanned data of the deformed structure parts, but we do not know the displacement field for given material points, which is usually impossible to measure.

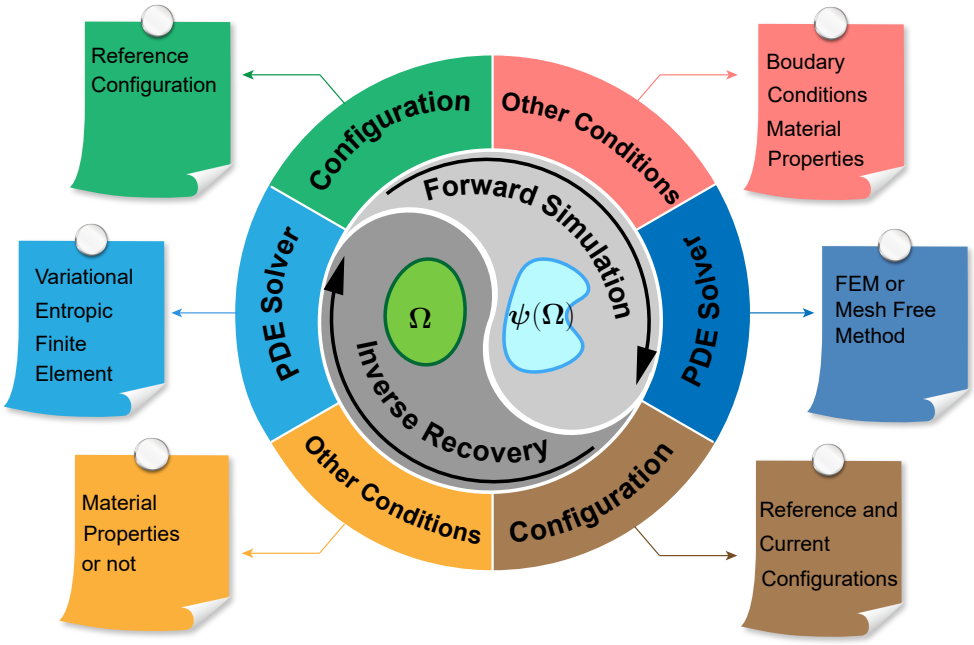


Figure 2.2: Comparison between the boundary-value problem and its inverse problem.

2.2 Variational Form and Weak Form in Mechanical Problems

Introduction to the weak form

The equations shown in Eq. (2.2) are refered to as the *strong form*. This is due to those equations require *strong* regularity of the basic unknown field $\mathbf{u}(\mathbf{X})$. Note that the first equation in Eq. (2.2) applies two derivative to the unknown field $\mathbf{u}(\mathbf{X})$. Therefore, one must

require that the second derivatives of $\mathbf{u}(\mathbf{X})$ make sense (they exist and are continuous). In some realistic circumstances, these constraints might be too strict. Next, an alternative form of the same problem that does not require these strong regularity conditions is developed. It is the so-called the *weak form*.

These steps can be taken to manipulate the BVP as follows. We first introduce the following space of functions

$$\mathcal{V}_u := \{\delta\mathbf{u} : \Omega \longrightarrow \mathbb{R}^n : \delta\mathbf{u} = 0 \quad \text{on} \quad \Gamma_u\} \tag{2.10}$$

After multiplying the first equation in Eq. (2.2) by a general function $\delta\mathbf{u}(\boldsymbol{x})\,(\delta\mathbf{u} : \Omega \to \mathbb{R}^n)$ and integrating over Ω clearly we obtain

$$\int_\Omega [\operatorname{div}\boldsymbol{\sigma} + \mathbf{b}] \cdot \delta\mathbf{u} d\Omega = 0, \tag{2.11}$$

for all functions $\delta\mathbf{u}(\boldsymbol{x})$. Integrating by parts (Gauss theorem) the first term in this equality we obtain

$$\begin{aligned}
\int_\Omega \delta\mathbf{u} \cdot (\operatorname{div}(\boldsymbol{\sigma}))d\Omega &= \int_\Omega \delta u_i \sigma_{ij,j} d\Omega \\
&= \int_\Omega \left(\delta u_i \sigma_{ij}\right)_{,j} d\Omega - \int_\Omega \delta u_{i,j}\sigma_{ij} d\Omega \\
&= \int_{\partial\Omega} \delta u_i \sigma_{ij} n_j d\Gamma - \int_\Omega \delta u_{i,j}\sigma_{ij} d\Omega \\
&= \int_{\Gamma_t} \delta u_i \sigma_{ij} n_j d\Gamma - \int_\Omega \delta u_{i,j}\sigma_{ij} d\Omega,
\end{aligned} \tag{2.12}$$

given the symmetry of the stress tensor $\boldsymbol{\sigma}$. We have also used the notation $\boldsymbol{\varepsilon}(\delta\mathbf{u}) := \nabla^s\delta\mathbf{u} = \left(\nabla\delta\mathbf{u} + (\nabla\delta\mathbf{u})^T\right)/2$. With this choice, the boundary integral vanishes outside Γ_t where $\boldsymbol{\sigma}\boldsymbol{n} = \bar{\boldsymbol{t}}$ (given). Combining then this result with the original equation, we obtain

$$\int_\Omega \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\delta\mathbf{u}) d\Omega = \int_\Omega \mathbf{b} \cdot \delta\mathbf{u} d\Omega + \int_{\Gamma_t} \bar{\boldsymbol{t}} \cdot \delta\mathbf{u} d\Gamma \quad \forall\delta\mathbf{u} \in \mathcal{V}_u \tag{2.13}$$

The above development can be summarized as follow:

Infinitesimal Mechanical Problem (Weak form).
Find $\mathbf{u} : \Omega \to \mathbb{R}^n$ satisfying the essential boundary conditions $\mathbf{u} = \overline{\mathbf{u}}$ on Γ_u, and

$$\int_\Omega \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\delta\mathbf{u}) d\Omega = \int_\Omega \mathbf{b} \cdot \delta\mathbf{u} d\Omega + \int_{\Gamma_t} \overline{\boldsymbol{t}} \cdot \delta\mathbf{u} d\Gamma \quad \forall \delta\mathbf{u} \in \mathcal{V}_u \tag{2.14}$$

where the set of kinematically admissible variations $\mathcal{V}_u$ is given by

$$\mathcal{V}_u = \{\delta\mathbf{u} : \Omega \to \mathbb{R}^n : \delta\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_u\} \tag{2.15}$$

where σ is given in terms of the unknown function $\mathbf{u}$ by the constitutive and compatibility relations, that is, $\boldsymbol{\sigma} = \hat{\boldsymbol{\sigma}}(\boldsymbol{\varepsilon}(\mathbf{u}))$ with $\boldsymbol{\varepsilon}(\mathbf{u}) = \nabla^s \mathbf{u}$.

Remark 2.2.1. *The weak statement in Eq. (2.14) enforces weakly equilibrium , whereas the constitutive relation ($\boldsymbol{\sigma} = \hat{\boldsymbol{\sigma}}(\boldsymbol{\varepsilon})$), compatibility ($\boldsymbol{\varepsilon}(\boldsymbol{u}) = \nabla^s \boldsymbol{u}$) and natural boundary conditions ($\boldsymbol{u} = \overline{\boldsymbol{u}}$ on Γ_u) are enforced strongly or pointwise.*

Introduction to the variational form

The *weak form* in Eq. (2.14) can be applied to both elastic and inelastic materials. In particular, for elastic material and elastic problem, a *variational form* of the equilibrium equation can be given as below.

Firstly, a conservative system can be defined as follow:

1. The constitutive relation is given by a stored energy function $W = W(\varepsilon)$ with
$$\boldsymbol{\sigma} = \hat{\boldsymbol{\sigma}}(\boldsymbol{\varepsilon}) = \frac{\partial W}{\partial \boldsymbol{\varepsilon}}. \tag{2.16}$$
2. The external load can be represented by a functional $\Pi_{ext}(\boldsymbol{u})$, such that the (minus) external virtual work in an admissible variation $\delta\boldsymbol{u}$ is given by
$$\delta\Pi_{ext}(\boldsymbol{u}, \delta\boldsymbol{u}) := \frac{d}{d\eta}\Pi_{ext}(\boldsymbol{u} + \eta\delta\boldsymbol{u})\Big|_{\eta=0} = -\int_\Omega \boldsymbol{f} \cdot \delta\boldsymbol{u} d\Omega - \int_{\Gamma_t} \overline{\boldsymbol{t}} \cdot \delta\boldsymbol{u} d\Gamma. \tag{2.17}$$

Then, the variational form of the infinitesimal mechanical problem can be represented as

Infinitesimal Mechanical Problem (Variational Form).
Find $\mathbf{u} : \Omega \to \mathbb{R}^n$, satisfying the essential boundary conditions $\mathbf{u} = \overline{\mathbf{u}}$ on Γ_u, that makes stationary the potential energy

$$\Pi(\mathbf{u}) = \int_\Omega W(\boldsymbol{\varepsilon}(\mathbf{u})) d\Omega + \Pi_{ext}(\mathbf{u}) \tag{2.18}$$

among all admissible variations $\delta\mathbf{u} \in \mathcal{V}_u$.

It can be shown that the *variational form* statement implies the the *weak form* statement. The stationary condition for attaining the extreme of $\Pi(\mathbf{u})$ is,

$$\delta\Pi := \frac{d}{d\eta}\Pi\left(\mathbf{u}_\eta\right)\Big|_{\eta=0} = 0. \tag{2.19}$$

Follow the calculation, we arrive at the following result

$$\delta\Pi = \int_\Omega \frac{\partial W}{\partial \boldsymbol{\varepsilon}}\left(\boldsymbol{\varepsilon}\left(\mathbf{u}_{\eta=0}\right)\right) : \frac{d\boldsymbol{\varepsilon}\left(\mathbf{u}_\eta\right)}{d\eta}\Big|_{\eta=0} d\Omega + \delta\Pi_{ext}, \tag{2.20}$$

where

$$\frac{d\boldsymbol{\varepsilon}\left(\mathbf{u}_\eta\right)}{d\eta}\Big|_{\eta=0} = \delta\boldsymbol{\varepsilon} = \nabla^s(\delta\mathbf{u}) = \boldsymbol{\varepsilon}(\delta\mathbf{u}). \tag{2.21}$$

After substituting the Eq. (2.21) into Eq. (2.20), we arrive at

$$\delta\Pi = \quad = \int_\Omega \hat{\boldsymbol{\sigma}}(\boldsymbol{\varepsilon}(\mathbf{u})) : \boldsymbol{\varepsilon}(\delta\mathbf{u}) d\Omega - \int_\Omega \mathbf{b}\cdot\delta\mathbf{u} d\Omega - \int_{\Gamma_t} \overline{\boldsymbol{t}}\cdot\delta\mathbf{u} d\Gamma = 0, \tag{2.22}$$

which is the *weak form* shown in Eq. (2.14).

2.3 Synopsis of Finite Element Method

Nowadays, a lot of people employ finite element methods to numerically address structural, fluid,, and multiphysics issues [8, 77, 10]. The idea of FEM is to impose the Eq. (2.14) not for all test functions, but rather for a specific set that can be handled simply (that is, that may be described by a finite set of them). That will provide an approximate solution, enforcing in this case equilibrium approximately. For FEM, the class of functions is piecewise polynomials.

In the finite element approach, the interpolation functions must be selected in order to approximate the main field variable displacement $\mathbf{u}(\mathbf{X})$. As a result, the mathematical model's precise solution is estimated to within one finite element by

$$\mathbf{u}_{\text{exact}}\left(\mathbf{X}\right) \approx \mathbf{u}_h(\mathbf{X}) = \sum_{I=1}^{n} N_I(\mathbf{X})\ \mathbf{u}_I, \tag{2.23}$$

where N_I are the shape functions which are defined in a finite element Ω_e. Similarly, the virtual displacement $\delta\mathbf{u}$ can also be estimated as

$$\delta\mathbf{u}_{\text{exact}}\left(\mathbf{X}\right) \approx \delta\mathbf{u}_h(\mathbf{X}) = \sum_{I=1}^{n} N_I(\mathbf{X})\ \delta\mathbf{u}_I, \tag{2.24}$$

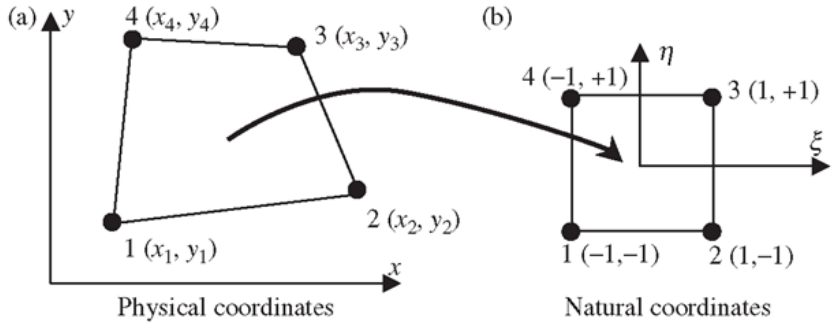


Figure 2.3: Quadrilateral element in the geometric and natural coordinate systems.

Eq. (2.23) and Eq. (2.24) can be write compactly as

$$\mathbf{u}_h = [\mathbf{N}^e]\,[\mathbf{u}^e] \quad , \quad \delta\mathbf{u}_h = [\mathbf{N}^e]\,[\delta\mathbf{u}^e] \tag{2.25}$$

The whole reference configuration can be discretized by a set of finite element:

$$\Omega \approx \Omega^h = \bigcup_{e=1}^{n_e} \Omega_e, \tag{2.26}$$

where n_e is the number of finite element in the reference configuration. The unknown nodal displacement is represented by $\mathbf{u}_I$. For three-dimensional displacement, the nodal displacement can be represented as $\mathbf{u}_I = [u_1, u_2, u_3]_I^T$.

As one can see from Eq. (2.23), the main variable $\mathbf{u}$ is interpolated by the shape functions N_I. The isoparametric notion [7] is mostly employed as an interpolation approach for numerous engineering problems due to its broad applicability. Under this concept, one use the same shape functions N_I to interpolate both the geometry and displacement. That is to say,

$$\mathbf{X}_h = \sum_{I=1}^{n} N_I(\mathbf{X})\ \mathbf{X}_I. \tag{2.27}$$

Forming the shape functions $N_I(\mathbf{X})$ directly is too burdensome. In practice, for the two dimensional case, the physical coordinates system is represented by the natural coordinate system of ξ and η as shown in Fig. (2.3). Both ξ and η are in the range -1 to 1 (refer to as $\square = [-1, 1]^2$).

For example, the shape functions for bilinear quadrilateral element is given as,

$$N_I(\xi, \eta) = \frac{1}{4}\,(1 + \xi\xi_I)\,(1 + \eta\eta_I)\,, \tag{2.28}$$

where ξ_I and η_I are the natural coordinates of the node I. Similar equation can be deduced for quadratic elements.

In practice, the stress and strain are represented by the Voigt's vector notation

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{22} & \varepsilon_{33} & 2\varepsilon_{12} & 2\varepsilon_{23} & 2\varepsilon_{31} \end{bmatrix}^T, \tag{2.29}$$

$$[\boldsymbol{\sigma}] = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{12} & \sigma_{23} & \sigma_{31} \end{bmatrix}^T. \tag{2.30}$$

Then the stress-strain relation can be written using the vector convention as

$$[\boldsymbol{\sigma}] = [\mathbf{D}][\boldsymbol{\varepsilon}], \tag{2.31}$$

For linear elastic material in three-dimentional domain, the elasticity matrix $[\mathbf{D}]$ can be represented as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{bmatrix} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \end{bmatrix}. \tag{2.32}$$

For plain strain, the above equation can be reduced to

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda+2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}. \tag{2.33}$$

One can rewrite the weak form in Eq. (2.14) using the vector forms as

$$\int_\Omega [\boldsymbol{\varepsilon}(\delta\mathbf{u})]^T[\mathbf{D}][\boldsymbol{\varepsilon}(\mathbf{u})]d\Omega = \int_\Omega [\delta\mathbf{u}]^T[\mathbf{f}]d\Omega + \int_{\Gamma_t} [\delta\mathbf{u}]^T[\overline{\mathbf{t}}]d\Gamma. \tag{2.34}$$

Note that the relationship between strain tensor and displacement vector can be expressed as

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{2.35}$$

If we define the strain-displacement matrix as

$$[\mathbf{B}_I^e] = \begin{bmatrix} \frac{\partial N_I^e}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial N_I^e}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial N_I^e}{\partial x_3} \\ \frac{\partial N_I^e}{\partial x_2} & \frac{\partial N_I^e}{\partial x_1} & 0 \\ 0 & \frac{\partial N_I^e}{\partial x_3} & \frac{\partial N_I^e}{\partial x_2} \\ \frac{\partial N_I^e}{\partial x_3} & 0 & \frac{\partial N_I^e}{\partial x_1} \end{bmatrix}. \tag{2.36}$$

Therefore, the relationship between strain and displacement can be expressed as

$$[\boldsymbol{\varepsilon}(\mathbf{u}_h)] = [\mathbf{B}^e][\mathbf{u}^e] \quad , \quad [\boldsymbol{\varepsilon}(\delta\mathbf{u}_h)] = [\mathbf{B}^e][\delta\mathbf{u}^e]. \tag{2.37}$$

The weak form in Eq. (2.34) can be applied to element e, such that

$$\int_{\Omega^e} [\varepsilon(\delta\mathbf{u}_h)]^T [\mathbf{D}][\varepsilon(\mathbf{u}_h)]\, d\Omega = \int_{\Omega^e} [\delta\mathbf{u}_h]^T [\mathbf{f}] d\Omega + \int_{\partial\Omega^e\cap\Gamma_t} [\delta\mathbf{u}_h]^T [\bar{\mathbf{t}}] d\Gamma. \tag{2.38}$$

Appealing to Eq. (2.25) and Eq. (2.37), the preceding weak form is written as

$$\begin{aligned} &\int_{\Omega^e} ([\mathbf{B}^e][\delta\mathbf{u}^e])^T [\mathbf{D}]([\mathbf{B}^e][\mathbf{u}^e])\, d\Omega \\ &= \int_{\Omega^e} ([\mathbf{N}^e][\delta\mathbf{u}^e])^T [\mathbf{f}] d\Omega + \int_{\partial\Omega^e\cap\Gamma_t} ([\mathbf{N}^e][\delta\mathbf{u}^e])^T [\bar{\mathbf{t}}] d\Gamma \end{aligned} \tag{2.39}$$

or

$$[\delta\mathbf{u}^e]^T \left[\left\{ \int_{\Omega^e} [\mathbf{B}^e]^T [\mathbf{D}][\mathbf{B}^e]\, d\Omega \right\} [\mathbf{u}^e] - \int_{\Omega^e} [\mathbf{N}^e]^T [\mathbf{f}] d\Omega - \int_{\partial\Omega^e\cap\Gamma_t} [\mathbf{N}^e]^T [\bar{\mathbf{t}}] d\Gamma \right] = 0. \tag{2.40}$$

Due to the arbitrary nature of $[\delta\mathbf{u}^e]$, Eq. (2.40) leads to the linear system

$$[\mathbf{K}^e][\mathbf{u}^e] = [\mathbf{F}^e], \tag{2.41}$$

where

$$\begin{aligned} [\mathbf{K}^e] &= \int_{\Omega^e} [\mathbf{B}^e]^T [\mathbf{D}][\mathbf{B}^e]\, d\Omega, \\ [\mathbf{F}^e] &= \int_{\Omega^e} [\mathbf{N}^e]^T [\mathbf{f}] d\Omega + \int_{\partial\Omega^e\cap\Gamma_q} [\mathbf{N}^e]^T [\bar{\mathbf{t}}] d\Gamma. \end{aligned} \tag{2.42}$$

$[\mathbf{K}^e]$ is the element stiffness matrix, $[\mathbf{F}^e]$ is the element load vector. Finally, the element local linear system can be assembled to arrive at the global system

$$[\mathbf{K}][\mathbf{u}] = [\mathbf{F}], \tag{2.43}$$

And the global nodal displacement $[\mathbf{u}]$ can be solve by

$$[\mathbf{u}] = [\mathbf{K}]^{-1}[\mathbf{F}]. \tag{2.44}$$

2.4 Overview of Related Statistics Methods

This section begins with a quick overview of various important ideas and concepts in statistics and machine learning. It is intended to guide readers through the development of the Coherent Point Drift approach, so that they may have a firm understand of the new algorithms offered in the following chapters.

Maximum likelihood estimation

Estimation theory [45] is a subfield of statistics that focuses on estimating the values of parameters based on measurable empirical data that includes a component of randomness. maximum likelihood estimation (MLE) is one of the parameter estimation methods which estimates the parameters by maximizing the likelihood function [67, 71]. The definition of the likelihood function $\mathcal{L}_N(\boldsymbol{\theta} \mid \mathbf{X})$ is given as follows,

$$\mathcal{L}_N(\boldsymbol{\theta} \mid \mathbf{X}) = \prod_{i=1}^{N} f\left(X_i \mid \boldsymbol{\theta}\right), \tag{2.45}$$

where $X_1, \ldots, X_N$ are random variables, $f\left(X \mid \theta\right)$ is the probability density function (PDF) for the random variable X with parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_k]$. MLE assumes that the random variable $X_1, \ldots, X_N$ are independent and identically distributed (IID). The estimated value of model parameters $\hat{\boldsymbol{\theta}}$ can be calculated by maximizing the likelihood function over the parameter space Θ,

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \Theta}{\arg\max}\ \mathcal{L}_N(\boldsymbol{\theta} \mid \mathbf{X})\ . \tag{2.46}$$

The log-likelihood function is defined as

$$\ell_N(\boldsymbol{\theta} \mid X) = \log \mathcal{L}_N(\boldsymbol{\theta})\ . \tag{2.47}$$

The same model parameters $\hat{\boldsymbol{\theta}}$ maximize both $\mathcal{L}_N(\boldsymbol{\theta} \mid \mathbf{X})$ and $\ell_N(\boldsymbol{\theta} \mid \mathbf{X})$.

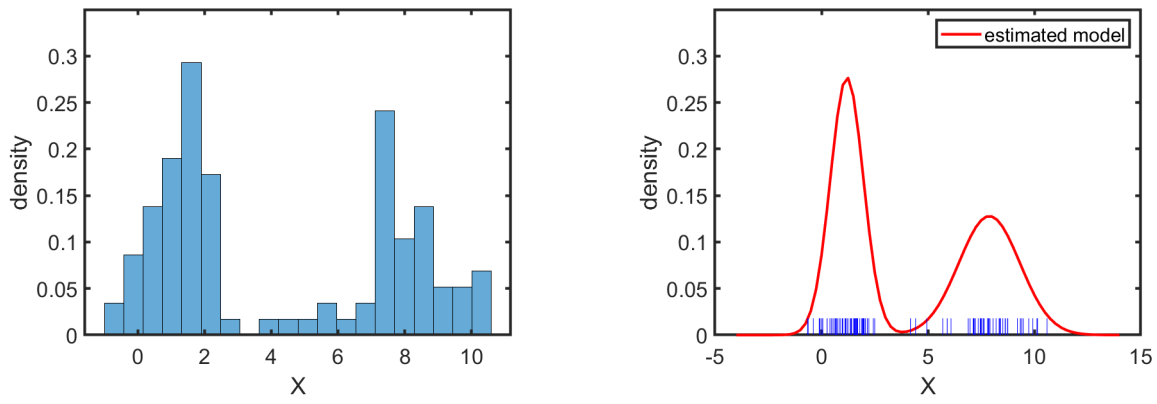
In general, it is more convenient to work with the log-likelihood, This is because the asymptotic properties of sums are easier to analyze, and the summation is more stable than product in numerical calculations. MLE possesses a number of optimal properties, such as sufficiency, consistency, and efficiency. The in-depth discussions of those properties can be found in [18, 23].

The Gaussian mixture model and the expectation-maximization algorithm

The common probability distributions, such as Gaussian distribution and the gamma distribution, are generally uni-modal. In practical applications, the data are multimodal. For instance, the fictitious data points shown in Fig. 2.4(a) are apparently bi-modal. Therefore, in order to model the multi-modal data sets, a mixture model technique needs to be applied.

One way to generate a bi-modal distribution is by using the so-called Gaussian mixture model (GMM) [79, 58]. For this example, the distribution can be modeled by a mixture of two Gaussian distributions:

$$\begin{aligned} X_1 &\sim \phi_{\mu_1,\sigma_1} := \mathcal{N}\left(\mu_1, \sigma_1^2\right), \\ X_2 &\sim \phi_{\mu_2,\sigma_2} := \mathcal{N}\left(\mu_2, \sigma_2^2\right), \\ X &= (1-\Delta)\cdot X_1 + \Delta \cdot X_2, \end{aligned} \tag{2.48}$$



(a) (b)

Figure 2.4: Gaussian mixture model example: (a) Histogram of data; (b) Gaussian mixture model fitted by maximum likelihood estimation.

where $\mathcal{N}(\mu, \sigma^2)$ is the standard notation for the normal or Gaussian distribution with mean μ and variance σ^2; X_1 and X_2 are two independent random variables that are normally distributed with different parameters. The parameter Δ may be viewed as a discrete random variable $\Delta \in \{0, 1\}$ with $\mathrm{P}(\Delta = 1) = \pi$, where $\pi \in [0, 1]$ is a parameter. Δ is also referred to as the latent variable which is not directly observed, but inferred from other observed variables. The new random variable X can model the bi-modal distribution with a two-step procedure. First, the value of Δ can be generated based on its probability value π. Then, the value of X can be calculated by the values of X_1 or X_2, and Δ.

In this example the model parameters are

$$\boldsymbol{\theta} = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2). \tag{2.49}$$

With the data in Fig. 2.4(a), the model parameter $\boldsymbol{\theta}$ can be fitted by the MLE. Let $\phi_\theta(x)$ denote the Gaussian distribution. Then the log-likelihood function of the N data points will be

$$\ell_N(\boldsymbol{\theta} \mid X) = \sum_{i=1}^{N} \log \left[(1-\pi)\phi_{\theta_1}(X_i) + \pi\phi_{\theta_2}(X_i)\right]. \tag{2.50}$$

Note that the above expression can be extended to random vector $\mathbf{X}$, i.e.

$$\ell_N(\boldsymbol{\theta} \mid \mathbf{X}) = \sum_{i=1}^{N} \log \left[(1-\pi)\phi_{\theta_1}(\mathbf{X}_i) + \pi\phi_{\theta_2}(\mathbf{X}_i)\right], \quad \mathbf{X}_i \in \mathbb{R}^D \tag{2.51}$$

where D is the dimension of the vector space.

Maximizing $\ell_N(\boldsymbol{\theta} \mid \mathbf{X})$ directly is quite challenging, due to the sum of the terms inside the logarithm. In actual computations, in order to maximize $\ell_N(\boldsymbol{\theta} \mid \mathbf{X})$, a so-called expectation-maximization (EM) algorithm [24, 13] is used, which is an iterative or alternating method

to find the maximum likelihood estimators on various incomplete data sets in statistics. Precisely, the EM algorithm can use an iterative approach to find the maximum likelihood estimation of interested parameters in the statistics model which heavily depends on missing data or the so-called latent variables.

In the EM algorithm, the first iterative step is a soft imputation of the missing data with "old" parameter values for all its possible values (E-step). For instance, the probability of Δ_j for i-th data points is given by Bayes rule,

$$p^{old}\left(\Delta_j \mid \mathbf{X}_i\right) = p(\Delta_j) p\left(\mathbf{X}_i \mid \Delta_j\right) / p\left(\mathbf{X}_i\right). \tag{2.52}$$

The second step is to update model parameters with the given imputed data (M-step) by minimizing the objective function $\mathcal{L}_Q$ with respect to the "new" parameters.

$$\mathcal{L}_Q = -\sum_{i=1}^{N}\sum_{j=1}^{M} p^{old}\left(\Delta_j \mid \mathbf{X}_i\right) \log\left(p^{new}(\Delta_j) p^{new}\left(\mathbf{X}_i \mid \Delta_j\right)\right) \tag{2.53}$$

By following the EM algorithm, the parameters of GMM in the example are fitted with the data. The fitting result is shown in Fig. 2.4(b). The above example shows that the centroids of each Gaussian distribution in GMM move to the dense areas where most data points are located. The above phenomenon is the core concept in the CPD algorithm.

The CPD algorithm

Since there is a resemblance between the *continuum/material deformation finding algorithm* and the original CPD algorithm [65, 66], for the sake of self-containedness, a brief review of the CPD method for the non-rigid point registration is provided in this subsection. The main notations in CPD are listed in Table. 2.1.

Table 2.1: Table of Notations in CPD

Symbol	Description
D	Dimension of data space (2 or 3)
N	Number of points in un-deformed configuration
m	Number of points in deformed configuration
$\mathbf{X}_{N\times D}$	The node positions in un-deformed configuration
$\mathbf{x}_{m\times D}$	The node positions in deformed configuration
σ^2	Variance of Gaussian distribution in GMM
$\chi(\mathbf{X})$	A general deformation mapping function which maps the un-deformed configuration to the deformed configuration

The CPD algorithm was first proposed and developed by Myronenko and Song [66]. An example of a registration problem, which can be solved by the CPD algorithm, is illustrated in Fig. 2.5(a). The CPD algorithm first assumes that two point sets are given: (1) $\mathbf{x}_{m\times D} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^T$ is the first set of data for spatial point positions in the deformed configuration, and (2) $\mathbf{X}_{N\times D} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)^T$ is the second set of point positions in the un-deformed configuration. In the later part of the paper, we use the shorthanded notation $\mathbf{x}$ to denote $\mathbf{x}_{m\times D} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^T$ and $\mathbf{X}$ to denote $\mathbf{X}_{N\times D} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)^T$.

After having those two point sets, the goal of this registration problem is twofold: (1) Align two point sets and identify the correspondence pairs of points $(\mathbf{X}_I, \mathbf{x}_j)$, and (2) Find the non-uniform deformation map χ.

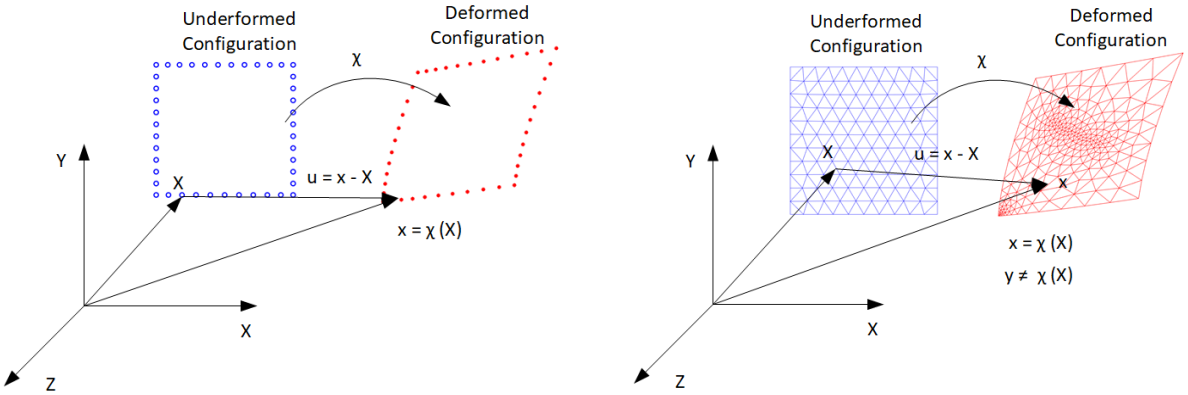


Figure 2.5: The comparison between the non-rigid registration method and material deformation finding method: (a) Non-rigid registration problem in computer vision vs. (b) Material deformation finding problem in 3D printing and continuum mechanics .

The point set registration in the CPD algorithm is formulated as a Maximum Likelihood Estimation (MLE) of a Gaussian Mixture Model (GMM). For example, one can choose $\mathbf{X}_{N\times D}$ as the data points and $\mathbf{x}_{m\times D}$ as the GMM centroids. Then the probability density function can be defined as

$$p\left(\mathbf{X}_I \mid \boldsymbol{\theta}\right) = (1-w)\sum_{j=1}^{m} p\left(\Delta_j\right) p\left(\mathbf{X}_I \mid \Delta_j\right) + wp\left(\mathbf{X}_I \mid \Delta = m+1\right), \tag{2.54}$$

where $\Delta \in \{1, 2, \cdots, m+1\}$ is the latent variable, and $\boldsymbol{\theta}$ contains all the model parameters. $\Delta = 1, 2, \cdots, m$ correspond to the GMM; $\Delta = m+1$ is an added degree of freedom for outliers, and $I = 1, 2, \cdots, N$ are indices for the data points. Here $p(\mathbf{X}_I \mid \Delta = j)$ is the conditional probability of the data point $\mathbf{X}_I$ with respect to the j-th Gaussian distribution in GMM, while $p(\Delta = j)$ is the membership probability. Here we assume that each component

probability shares the same value,

$$p(\Delta = j) = \frac{1}{m}. \tag{2.55}$$

In order to take into account noises and outliers, a uniform probability density function

$$p(\mathbf{X}_I \mid \Delta = m + 1) = \frac{1}{N} \tag{2.56}$$

is adopted for the mixture model. Then the final density function is a mixture model of the original GMM and a uniform distribution with parameter w which represents our prior belief of the level of noise in the data.

In the original CPD algorithm, the covariance matrices Σ for each Gaussian distribution are assumed to be the same, i.e.

$$\Sigma_1 = \Sigma_2 = ...\Sigma_j = \sigma^2 \cdot \mathbb{I}, \tag{2.57}$$

where $\mathbb{I}$ is the identity matrix, and all covariance matrices are controlled by the variance σ^2. The model parameters $\boldsymbol{\theta}$ contain the centroids of each Gaussian distribution and the variance

$$\boldsymbol{\theta} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m, \sigma^2). \tag{2.58}$$

Then the conditional probability density function for the I-th data point $\mathbf{X}_I$ on j-th component in GMM is given as,

$$p(\mathbf{X}_I \mid \Delta_j) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\Big(-\frac{\|\mathbf{X}_I - \mathbf{x}_j\| 2}{2\sigma^2}\Big), \tag{2.59}$$

where the space is D-dimensional. In the registration cases, D usually equals 2 or 3. Finally, the probability density function $p(\mathbf{X}_I \mid \boldsymbol{\theta})$ can be expressed as

$$p(\mathbf{X}_I \mid \boldsymbol{\theta}) = w\frac{1}{N} + \frac{(1-w)}{m} \sum_{j=1}^{m} \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\Big(-\frac{\|\mathbf{X}_I - \mathbf{x}_j\| 2}{2\sigma^2}\Big). \tag{2.60}$$

By considering a statistical MLE, in which one maximizes the probability for each data point by moving the GMM centroids, the spatial points move to align (overlap) with the second set of data points in the undeformed configuration. That is to say, at the optimum, we match the points in two sets and obtain the correspondence pairs of points $(\mathbf{X}_I, \mathbf{x}_i)$.

The likelihood function in the CPD algorithm is defined as

$$\mathcal{L}_N(\boldsymbol{\theta} \mid \mathbf{X}) = \prod_{I=1}^{N} p(\mathbf{X}_I \mid \boldsymbol{\theta}). \tag{2.61}$$

Thus, in essence, the model parameter set $\boldsymbol{\theta}$ is updated by maximizing the likelihood function, such that the GMM centroids can converge to the data point set $\mathbf{X}$. That is

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \log\Big(\mathcal{L}_N(\boldsymbol{\theta} \mid \mathbf{X})\Big). \tag{2.62}$$

In actual computations, the likelihood function is maximized by the EM algorithm, the objective function $\mathcal{L}_Q$ in t-th iterative step is defined as

$$\mathcal{L}_Q\left(\boldsymbol{\theta}^{(\boldsymbol{t-1})}\right) = \frac{1}{2\sigma^{(t-1)^2}} \sum_{j=1}^{m}\sum_{I=1}^{N} p^{(t-1)}\left(\Delta_j \mid \mathbf{X}_I\right) \left\|\mathbf{X}_I - \mathbf{x}_j^{(t-1)}\right\|^2 + \frac{N_p D}{2}\log\sigma^{(t-1)^2}, \tag{2.63}$$

where $\boldsymbol{\theta}^{(\boldsymbol{t-1})} = (\mathbf{x}_1^{(t-1)}, \mathbf{x}_2^{(t-1)}, \cdots, \mathbf{x}_m^{(t-1)}, \sigma^{(t-1)^2})$ is the model parameter set and $\mathbf{x}^{(t-1)}$ are the intermediate GMM centroid locations after t-1 iterations in EM algorithm;

$$N_p = \sum_{j=1}^{m}\sum_{I=1}^{N} p^{(t-1)}\left(\Delta_j \mid \mathbf{X}_I\right), \tag{2.64}$$

and

$$p^{(t-1)}\left(\Delta_j \mid \mathbf{X}_I\right) = \frac{\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{X}_I - \mathbf{x}_j^(t-1)}{\sigma^{(t-1)}}\right\|^2\right)}{\sum_{j=1}^{M}\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{X}_I - \mathbf{x}_j^(t-1)}{\sigma^{(t-1)}}\right\|^2\right) + \left(2\pi\sigma^{(t-1)^2}\right)^{D/2}\frac{w}{1-w}\frac{m}{N}}. \tag{2.65}$$

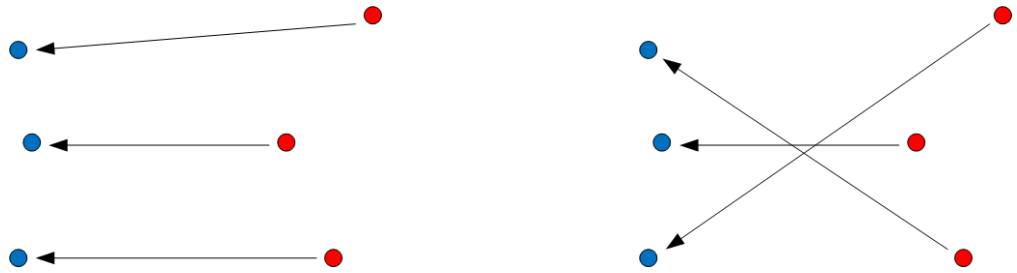
The new model parameter $\boldsymbol{\theta}^{(\boldsymbol{t})}$ is obtained by maximizing the $\mathcal{L}_Q\left(\boldsymbol{\theta}^{(\boldsymbol{t-1})}\right)$ in t-th M-step, that is

$$\boldsymbol{\theta}^{(\boldsymbol{t})} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \mathcal{L}_Q\left(\boldsymbol{\theta}^{(\boldsymbol{t-1})}\right). \tag{2.66}$$

By maximizing the likelihood function in Eq. (2.61), the GMM centroids (red) move towards the data points (blue) as shown in Fig. 2.6. However, the GMM centroids may move coherently by following the topological structure as shown in Fig. 2.6(a) or move randomly as shown in Fig. 2.6(b). Both the registration results in Fig. 2.6(a) and Fig. 2.6(b) indicate the same value of likelihood. Therefore, maximizing the likelihood function alone cannot guarantee a matching result which preserves the topological structure.

The essence of the CPD algorithm is to ensure that the topological structure of data sets is preserved during the registration process as shown in Fig. 2.6(a). In order to distinguish the differences between Fig. 2.6(a) and Fig. 2.6(b), one may need to add a prior term $\mathcal{R}$ to the objective function $\mathcal{L}_Q$ and convert it into a Maximum A Posteriori (MAP) estimation. The prior term encodes our belief of coherent movement and smoothness of the displacement field. The new objective function can be cast as

$$\mathcal{L}_{QR}(\theta) = \mathcal{L}_Q(\theta) + \lambda\mathcal{R}(\mathbf{u}), \tag{2.67}$$



(a) (b)

Figure 2.6: (a) A coherent registration between two point sets. (b) A less coherent registration between two point sets.

where λ is the trade-off parameters which defines the strength of regularization. Specifically, the regularization term used in the nonrigid CPD algorithm is expressed as

$$\mathcal{R}(\mathbf{u}) = \int_{\mathbb{R}^D} \frac{|\tilde{\mathbf{u}}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s}, \tag{2.68}$$

where $\tilde{G}(\mathbf{s})$ is the Gaussian kernel which is defined as

$$\tilde{G}(\mathbf{s}) = \exp\left(-\frac{\pi^2 \mathbf{s}^2}{\beta}\right), \tag{2.69}$$

where β is the width of the Gaussian kernel, and $\tilde{\mathbf{u}}$ is the Fourier transformation of the displacement field which is defined as $\mathbf{u} = \mathbf{x}' - \mathbf{x}$. The integration is taken in the reciprocal space with the wavenumber $\mathbf{s}$. This regularization term can first filter the high spatial frequency part of the displacement field and then add penalty on it. By penalizing the high spatial frequency energy, this regularization term forces the displacement field to put more energy on the low spatial frequency domain, hence smoothing the displacement field.

After defining the constraint on mapping the minimizer that minimizes Eq.(2.67) can be obtained as follows,

$$\mathbf{u}(\mathbf{X}) = \sum_{i=1}^{N} \mathbf{v}_i G(\boldsymbol{X} - \mathbf{X}_i) \tag{2.70}$$

where $\mathbf{v}_i$ is the unknow parameter and the detailed derivation can be found in Appendix A.

Chapter 3

Material Deformation Finding Algorithm

In this chapter, we introduce a computational probabilistic method to quantify the permanent (non-zero strain) continuum/material deformation. Different from physical-based modeling, the method developed here is based on a data-driven statistics approach, which solves the problem without needing the information of physical deformation process. The proposed method relies only on the scanned material data from the thermal distorted configuration as well as the shape of the initial design configuration. We coined this artificial intelligence based algorithm as the material deformation finding (MDF) algorithm. Before deriving this algorithm, an overview of the distortion compensation methods in AM are firstly given.

3.1 Overview of the Distortion Compensation Methods in AM

Numerous different strategies or technologies have been used or suggested for the purpose of reducing or compensating for distortion in order to solve the issue of distortion compensation in AM. The key for deformation compensation is to accurately predict or measure deformation under a given set of AM process parameters. The examination of AM distortion may be broken down into three distinct categories: data-driven approaches, physics-based models, and experimental observations.

The most direct and accurate method is the in-situ [12] measuring. Digit Image Correlation (DIC) technique has been used for in-situ measuring distortion of Additive Manufacturing. By consecutive photographing a model in printing process, geometrical deformation or shifting of the sample can be captured based on comparing a series of images [61, 70, 94]. Laser Displacement Sensor (LDS) has also been used to gather in-situ distortion data [37]. In addition to in-situ measure technologies such as DIC and LMD, the Coordinate-measuring Machine (CMM) as post-process measurements can also be used to obtain distortion pro-

files [26]. On the other hand, however, after we obtained thermal distortion data from in-situ measurements, how to quantify the 3D distortion displacement field based on the measurement data is still an open and challenging question.

Therefore, at the current stage, experimental approaches are usually coupled with numerical modelings. The most used numerical modeling approach is Finite Element Method (FEM), e.g. [26, 4, 63, 51, 83]. The residual stresses and thermal distortions in various AM settings mainly by two methods—the thermo-mechanical approach [63, 4, 81, 96] and the inherent strain method [64]. By using coupled thermal, fluid flow and structure finite element models, T. Mukherjee et al. [63] studied the evolution of stresses and strains in various 3D printed products and their dependence on major process parameters including heat input and layer thickness. The thermo-mechanical method recently adopted the dynamic adaptive mesh refinement methodology [57, 72] to speed up calculations. However, Additive Manufacturing is involved with complex physical processes such as materials phase transformation, layer-by-layer material contact and mutual penetration, and even materials chemical reactions. These complex physical processes are far beyond conventional FEM modeling capability.

No matter which approach is adopted, experimental measurement or FEM modeling or coupled approach, the ultimate goal is to find the distortion field or its mapping function $\boldsymbol{x} = \boldsymbol{\psi}(\boldsymbol{X})$ from the initial design state.

Recently the approximation theory-based machine learning algorithms have become an active research topic in many industrial applications. The related applications in Additive Manufacturing have also emerged. For example, Cheng et al. [21] proposed a geometry control method, in which the dimension deviation is modelled by the transfer learning method. Francis et al [29] inserted a Convolutional Neural Network (CNN) into the framework Industry 4.0 to model distortion in Laser-based Additive Manufacturing (LBAM) process, especially taking local heat transfer into consideration. Huang et al. [41, 40, 97, 28] established several methods to extract shape-independent derivation profiles. Several machine learning strategies were then used to generate the compensation. Other developments can also be found in [95, 47]. However, these machine learning approaches do not have generality, because they were based specific ways to obtain data from a particular machine, particular position, particular materials, etc. If any of the above situations changes, one has to take the massive measurement again, which may not be suitable for industrial-scale batch production.

3.2 Motivation for the MDF algorithm

To motivate the innovation, we first examine the thermal compensation problem in 3D printing. In 3D printing compensation, one needs to know the thermal distortion field of each material point in the design mesh in order to create a new or compensated standard triangle language (STL) file [85]. This is so that one can use the existing material and printing process parameters to print a perfect product or 3D printing parts that the original STL design file cannot do.

Now the question is: how can we find the exact thermal distortion in every material point in the design mesh? We have the original design configuration, which is essentially determined by the material nodal points, i.e. $\mathbf{X}_{N\times D} = (\mathbf{X}_1, \mathbf{X}_2, \cdots \mathbf{X}_N)$. After the tryout printing, one can scan the thermal distorted part to obtain a scanned file that contacts the deformed material spatial positions, i.e. $\mathbf{y}_{m\times D} = (\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_m)$. In general, $m >> N$, and the mesh nodal points in the designed mesh have nothing to do with the scanned mesh nodal points. In other words, even though we know the scanned spatial nodal points $\{\mathbf{y}_j\}_{j=1}^m$, we do not know their images in the design mesh. It could be true that none of the mesh nodal points in the design mesh correspond to any of the scanned spatial nodal points in the scanned mesh, i.e.

$$\mathbf{y}_j \neq \chi(\mathbf{X}_I), \quad \forall j = 1, 2, \cdots m, \quad and \quad I = 1, 2, \cdots N \tag{3.1}$$

where χ is the deformation map. Now our task is to find:

$$\mathbf{x}_I = \chi(\mathbf{X}_I), \quad I = 1, 2, \cdots N \tag{3.2}$$

with only the available information of $\{\mathbf{y}_j\}_{j=1}^m$ and $\{\mathbf{X}_I\}_{I=1}^N$. One can see that we cannot use the CPD method to solve this problem, because 3D printing is "3D", and we could have no correspondence between any pair of points from set $\{\mathbf{y}_j\}_{j=1}^m$ and set $\{\mathbf{X}_I\}_{I=1}^N$.

Comparison between the CPD and MDF algorithms

To solve the thermal compensation problem in 3D printing, we propose here a material "registration" in continuum media or a material deformation finding algorithm to match the design mesh of a 3D printed model with a scanned material data set or mesh of the final printed product.

In doing so, we always assume that two following point sets are given: (1) the data of the spatial point positions in the deformation configuration, $\mathbf{y}_{m\times D} = (\mathbf{y}_1, \ldots, \mathbf{y}_m)^T$, and (2) the data of the design mesh nodal point set in the undeformed configuration, $\mathbf{X}_{N\times D} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)^T$, as shown in Fig. 2.5(b). The main notations in the MDF algorithm are shown in Table 3.1.

It should be noted that the spatial point set is denoted by $\mathbf{y}$ instead of $\mathbf{x}$ in the MDF algorithm. This is because they could be two totally different sets of material points.

In general, the design meshes generated by computer-aided design (CAD) software are high quality and almost uniformly distributed in the undeformed configuration. Whereas, the scanned meshes generated by a scan machine are usually poor quality and non-uniformly distributed in the deformed configuration.

For example, as shown in Fig.3.1, a 2D specimen of a unit square was chosen as the shape in the referential configuration. This was first meshed with uniform triangle whose nodal points serve as the reference, or data points. The uniform mesh in the referential configuration is shown in Fig.3.1 II(a). It is also called the design mesh in the referential

Table 3.1: Table of Notations in material deformation finding algorithm

Symbol	Description
D	Dimension of data space (2 or 3)
N	Number of points in undeformed configuration
m	Number of points in deformed configuration
$\mathbf{X}_{N\times D}$	The node positions in undeformed configuration
$\mathbf{y}_{m\times D}$	The node positions in deformed configuration
σ^2	Variance of Gaussian distribution in GMM
$\chi(\mathbf{X})$	A general deformation mapping function which maps the undeformed configuration to the deformed configuration
$\mathbf{x}_{N\times D}$	The locations of nodes $\mathbf{X}$ in the deformed configuration, where $\mathbf{x} = \chi(\mathbf{X})$
$\mathbf{Y}_{m\times D}$	The locations of nodes $\mathbf{y}$ in the undeformed configuratioin, where $\mathbf{Y} = \chi^{-1}(\mathbf{y})$
$\tilde{\mathbf{W}}_{m\times 1}$	Weight vector for each Gaussian distribution
$\mathbf{G}_{m\times m}$	Gaussian kernel matrix
$\mathbf{V}_{m\times D}$	Coefficient matrix

configuration. A locally refined non-uniform mesh was then generated by the Delaunay triangulation algorithm [73], which is still in the referential configuration as shown in Fig.3.1 I(a). After that, a non-zero strain or non-rigid deformation was applied to the square specimen. It deformed into a near rhombic shape as shown in Fig.3.1 I(b). The spatial positions of the nodal points of the locally refined non-uniform mesh are assumed to be the scanned material data (see Fig.3.1 I(b)). This is usually known by scanning the 3D printed product. In this way, a simple 2D registration example was generated to demonstrate the real setting in finding the 3D printing thermal-induced displacement field. In this example, the CPD algorithm is capable of finding the displacement field in Fig.3.1 I and Fig.3.1 II. However, the actually settings in 3D printing are shown in Fig.3.1 III.

As mentioned above, the main challenge is that there is no one to one correspondence between the nodes in the two meshes. The goal of the MDF algorithm is to figure out the deformation mapping function χ between two point sets $\mathbf{X}$ and $\mathbf{y}$ without the help of correspondence.

Besides the main difference above, there are also three important differences between point set registration and material deformation finding problem. First, in the computer vision application, both feature points in the two sets are obtained by the feature extraction from images or triangulation data. This means that each feature set may contain noise. Our material deformation finding problem, however, uses the exact design mesh as one feature set. Therefore, only the scanned data set may contain noise data and the design data set is

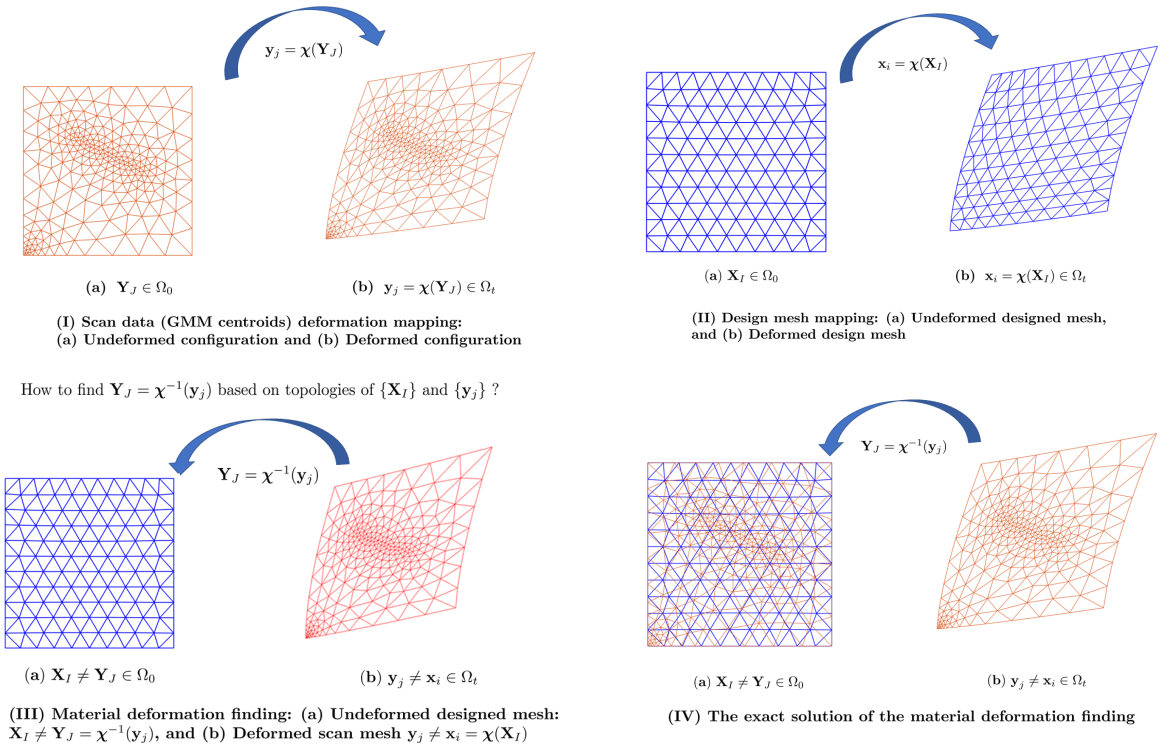


Figure 3.1: Material deformation finding for a deformed two-dimensional unit square: (I) Deformation of scanned data from the initial configuration to deformed configuration; (II) Deformation of designed mesh; (III) Topology connection between undeformed design mesh and scanned data in deformed configuration, and (IV) The exact solution for the material deformation finding problem.

noise-free.

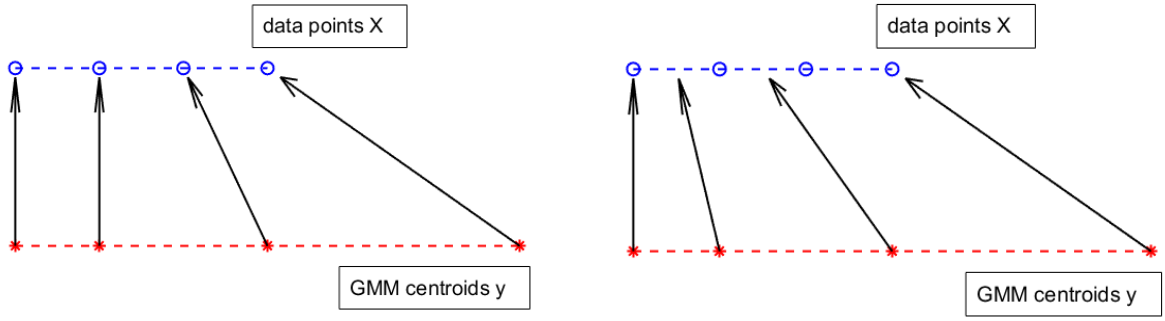
Second, in most cases, feature points in computer vision applications are only located on the boundary of the given model. On the contrary, the material deformation finding problem, as shown in Fig. 2.5(b), has feature points not only on the boundary but also in the interior domain.

Third, the objectives of the CPD algorithm are to obtain correspondence between two point sets and generate the deformation function χ. On the other hand, the objective of the MDF algorithm is to only find the deformation mapping χ. One example of this difference is shown in Fig. 3.2, which is a 1D example of a straight line. The real deformation applied to the original red line is uniform stretching and translation to the bottom. The transformed line is the blue line. However, the locations of the two middle points in the blue line are chosen arbitrarily. This means that the two middle points in the blue line are not corresponding to the two middle points in the red line. the MDF algorithm is shown in Fig. 3.2(b) where not all GMM centroid $\mathbf{y}$ converge to a data point in $\mathbf{X}$. But the deformation mapping function

Point set registration methods	**Material deformation finding method**
One to one correspondence for most feature points;	No one to one correspondence for most feature points.
Both data sets may contain noise;	Only the scanned data contains noise or outliers.
Only need feature points on the boundary of the body to define its shape;	All feature points are inside the domain of the body.
the objectives are to find correspondence and deformation mapping;	the objective is only to find the deformation mapping.

Table 3.2: Comparison between the point set registration methods and the material deformation finding algorithm.

generated by the movements of GMM centroid $\mathbf{y}$ is the real χ. This example illustrates that forcing the data points to match each other in a 3D printing setting may not be the optimal choice. A summary of the differences is shown in Table 3.2.



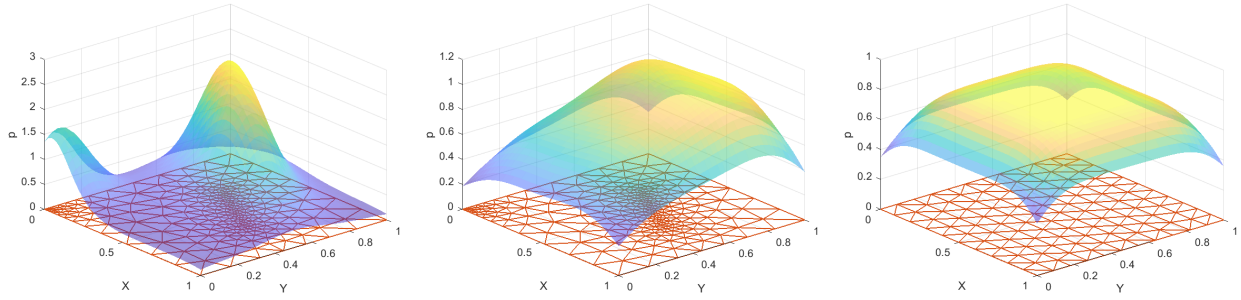
(a) (b)

Figure 3.2: An 1D example under uniform deformation: (a) Matching result based on the non-rigid point registration of CPD algorithm, and(b) The true underlying deformation of scanned data.

The comparison of different aspects of the two methods listed in Table 3.2 implies that the CPD algorithm may not be suitable to solve the material deformation finding problem. Therefore, a new method needs to be proposed. As discussed above, the original CPD

algorithm assumes equal isotropic covariances and equal membership probabilities for all GMM components. These assumptions fit well for the classical computer vision problems, where two point sets contain corresponding features extracted from either 2D images or 3D models. In most cases, there is no underlying mesh from where those feature points are selected. However, in our approach to the 3D printing thermal compensation problem, those two data points $\mathbf{X}_{N\times D}, \mathbf{y}_{m\times D}$ are vertices of the triangle mesh defined in the STL files. In fact, the mesh requirement is also not necessary, and the data points $\mathbf{X}_{N\times D}, \mathbf{y}_{m\times D}$ can serve as a meshfree representation of the given 3D printed models in different geometric configurations, i.e. spatial, as well as, referential configurations.

In meshfree particle methods e.g. [48, 52], the geometric state of a continuum is generally represented by a finite number of discrete particles. The motion of the continuum body can be tracked by the motion of these particles. In the Continuum Mechanics, by attaching a small volume, each particle represents a given fraction of the domain. The mass of each particle is proportional to the volume or area that each particle represents.



(a) (b) (c)

Figure 3.3: GMM density function: (a) non-uniform mesh with equal membership probability. (b) non-uniform mesh with specific membership probability for each member. (c) uniform mesh with equal membership probability.

3.3 Material Deformation Finding Algorithm

Inspired by meshfree particle methods e.g. [48], we assume that the membership probabilities for all GMM components and their corresponding covariances are proportional to the area or the volume that each scanned data point is assigned based on the underlying mesh. We can then define the weight $\tilde{w}$ for each membership probability in $\mathbf{y}$ and variance σ^2 as

$$\tilde{\mathbf{W}} = (\tilde{w}_1, \ldots, \tilde{w}_m)^T, \tag{3.3}$$

where $\tilde{w}_i = c \cdot area(w_i)$ and c is a constant. To compute the weight vector $\tilde{\mathbf{W}}$ and to determine the value of c, we add one constraint to it, which is

$$\frac{1}{m}\sum_{j=1}^{m}\tilde{w}_j = 1. \tag{3.4}$$

Therefore, the j-th component in GMM $p(\mathbf{X}_I \mid \Delta_j)$ and membership probability $p(\Delta_j)$ should be modified to

$$p(\mathbf{X}_I \mid \Delta_j) = \frac{1}{\left(2\pi\tilde{\sigma}_j^2\right)^{D/2}} \exp^{-\frac{\left\|\mathbf{x}_I - \mathbf{y}_j\right\|^2}{2\tilde{\sigma}_j^2}}, \; p(\Delta_j) = \frac{\tilde{w}_j}{m} \tag{3.5}$$

where $\tilde{\sigma}_j^2$ is defined as $\tilde{\sigma}_j^2 = \tilde{w}_j \cdot \sigma^2$.

Then the probability density function can be modified as

$$p(\mathbf{X}_I) = w\frac{1}{N} + (1-w)\sum_{j=1}^{m}\frac{\tilde{w}_j}{m}p(\mathbf{X}_I \mid \Delta_j) \; . \tag{3.6}$$

After recasting Eq. (2.60) into Eq. (3.6), the material deformation finding problem could be represented as an MLE and solved by the EM algorithm as in the original CPD algorithm solution. The underlying assumption of this specific membership probability is that the feature points is a meshfree representation of the model and then the shape of the model should be independent of how it is discretized. Therefore, ideally, the GMM density function generated by any arbitrary mesh should be identical to the GMM density function generated by the uniform mesh. Furthermore, the registration between a uniform mesh and an arbitrary mesh is conceptually identical to a registration problem between the two same meshes which can be solved by the CPD algorithm. In other words, the weight of several points in a dense area is equal to the weight of one single point in the same area. In numerical simulation, the weight may represent the volume or the mass. In statistics, the weight represents the probability density in that area.

For example, the GMM density function of a 2D model generated by different settings is shown in Fig. 3.3. This illustrates the uniform mesh and non-uniform mesh with specific membership probabilities for each member having similar density functions in the unit domain. However, the GMM density function of the non-uniform mesh with equal membership probability has two sharp modes where the mesh is dense, which makes it a poor representation of the true underlying density function.

By modifying the EM algorithm, we can write the objective function $\mathcal{L}_Q$ in Eq. (2.63) as,

$$\mathcal{L}_Q\left(\theta^{(t-1)}\right) = \frac{1}{2\sigma^{(t-1)^2}}\sum_{I=1}^{N}\sum_{j=1}^{m}p^{(t-1)}\left(\Delta_j \mid \mathbf{X}_I\right)\frac{\left\|\mathbf{X}_I - \mathbf{y}_j^{(t-1)}\right\|^2}{\tilde{w}_j} + \frac{N_p D}{2}\log\sigma^{(t-1)^2} \tag{3.7}$$

where $p^{(t-1)}$ is a conditional probability of each designed mesh nodal point to each member of the GMM density function by using model parameters from the late iteration step,

$$p^{(t-1)}\left(\Delta_j \mid \mathbf{X}_I\right) = \frac{\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{X}_I - \mathbf{y}_j^{(t-1)}}{\tilde{\sigma}^{(t-1)}}\right\|^2\right)}{\sum_{j=1}^{m}\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{X}_I - \mathbf{y}_j^{(t-1)}}{\tilde{\sigma}^{(t-1)}}\right\|^2\right) + c}, \tag{3.8}$$

where $c = \left(2\pi\sigma^2\right)^{D/2}\frac{w}{1-w}\frac{m}{N}$. In order to coerce GMM centroids to move in a consistent behavior, the regularization term Eq. (2.68) is also included in the objective function. Then the transformation of the GMM centroids can be defined as a generalized radial basis function which can be derived by using the representation theorem in the reproducing kernel Hilbert space (RKHS) (see [82]),

$$\chi(\mathbf{y}) = \mathbf{y}^{(t-1)} = \mathbf{y} + \mathbf{G}\mathbf{V}^{(t-1)}, \tag{3.9}$$

where $\mathbf{V}_{m\times D}^{(t-1)} = (\mathbf{v}_1^{(t-1)}, ..., \mathbf{v}_m^{(t-1)})^T$ is the coefficient matrix that determines the displacements. Then the regularization term $\mathcal{R}(\mathbf{u})$ can be reparameterized as a function of $\mathbf{V}$,

$$\mathcal{R}\left(\mathbf{V}^{(t-1)}\right) = \int_{\mathbb{R}^D}\frac{|\tilde{\mathbf{u}}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}d\mathbf{s} = \mathrm{Tr}\left(\mathbf{V}^{(t-1)^T}\mathbf{G}\mathbf{V}^{(t-1)}\right), \tag{3.10}$$

where $\mathbf{G}_{m\times m}$ is the Gaussian kernel matrix with element $g_{ij} = \exp\left(-\frac{1}{2}\left\|\frac{\mathbf{y}_i - \mathbf{y}_j}{\beta}\right\|^2\right)$. The model parameter θ can also be redefined as

$$\theta = (\mathbf{V}, \sigma^2). \tag{3.11}$$

With this regularization term, the objective function in Eq. (2.63) can be rewritten as

$$\begin{aligned}\mathcal{L}_{QR}\left(\mathbf{V}^{(t-1)}, \sigma^{(t-1)^2}\right) =& \frac{1}{2\sigma^{(t-1)^2}}\sum_{I=1}^{N}\sum_{j=1}^{m}p^{(t-1)}\left(\Delta_j \mid \mathbf{X}_I\right)\frac{\left\|\mathbf{X}_I - \mathbf{y}_j^{(t-1)}\right\|^2}{\tilde{w}_j} \\ &+ \frac{N_p D}{2}\log\sigma^{(t-1)^2} + \frac{\alpha}{2}\mathrm{Tr}\left(\mathbf{V}^{(t-1)^T}\mathbf{G}\mathbf{V}^{(t-1)}\right)\end{aligned} \tag{3.12}$$

For a general iteration, we can get rid of all the superscript (t-1). Then, the objective function in Eq. (3.12) can be rewritten in the following matrix form,

$$\begin{aligned}\mathcal{L}_{QR}\left(\mathbf{V}, \sigma^2\right) =& \frac{1}{2\sigma^2}\left\{\mathrm{Tr}\left(\mathbf{y}^T d\left(\tilde{\mathbf{P}}^T\mathbf{1}\right)\mathbf{y}\right) - 2\,\mathrm{Tr}\left(\mathbf{X}^T\tilde{\mathbf{P}}\mathbf{y}\right) - 2\,\mathrm{Tr}\left(\mathbf{V}^T\mathbf{G}\tilde{\mathbf{P}}\mathbf{y}\right) + \mathrm{Tr}\left(\mathbf{X}^T d(\tilde{\mathbf{P}}\mathbf{1})\mathbf{X}\right)\right\} \\ &+ 2\,\mathrm{Tr}\left(\mathbf{V}^T\mathbf{G}d(\tilde{\mathbf{P}}\mathbf{1})\mathbf{X}\right) + \mathrm{Tr}\left(\mathbf{V}^T\mathbf{G}d(\tilde{\mathbf{P}}\mathbf{1})\mathbf{G}\mathbf{V}\right) \\ &+ \frac{N_p D}{2}\ln\left(\sigma^2\right) + \frac{\alpha}{2}\mathrm{Tr}\left(\mathbf{V}^T\mathbf{G}\mathbf{V}\right)\end{aligned} \tag{3.13}$$

where $\mathbf{P}$ is a $m \times N$ matrix with element $P_{jI} = p^{old}(\mathbf{y}_j|\mathbf{X}_I)$, $d(\mathbf{V})$ is a diagonal matrix with elements on the diagonal equal the value of a vector $\mathbf{V}$, $\mathbf{1}$ is a column vector where all the elements equal one.

$\mathcal{L}_{QR}(\mathbf{V}, \sigma^2)$ is a quadratic function of coefficient matrix $\mathbf{V}$. With the aim of updating $\mathbf{V}$ that minimizes the objective function in Eq.(3.13), the derivative of function $\mathcal{L}_{QR}(\mathbf{V}, \sigma^2)$ is set to zero, which is

$$\frac{\partial \mathcal{L}_{QR}(\mathbf{V}, \sigma^2)}{\partial \mathbf{V}} = -\mathbf{G}\tilde{\mathbf{P}}\mathbf{y} + \mathbf{G}d(\tilde{\mathbf{P}}\mathbf{1})\mathbf{X} + \mathbf{G}d(\tilde{\mathbf{P}}\mathbf{1})\mathbf{G}\mathbf{V} + \sigma^2\alpha\mathbf{G}\mathbf{V} = 0 \ . \tag{3.14}$$

Then, the value of coefficients $\mathbf{V}$ can be solved by the equation below,

$$\left[d(\tilde{\mathbf{P}}\mathbf{1})\mathbf{G} + \sigma^2\alpha\mathbf{I}\right]\mathbf{V} = \tilde{\mathbf{P}}\mathbf{y} - d(\tilde{\mathbf{P}}\mathbf{1})\mathbf{X} \tag{3.15}$$

where the detailed derivation can be found in Appendix B.

The σ^2 can be obtained by setting the derivative of $\mathcal{L}_{QR}(\mathbf{V}, \sigma^2)$ with respect to σ^2 to zero,

$$\sigma^2 = \frac{1}{N_P D}\left(\mathrm{tr}\left(\mathbf{y}^T \mathrm{d}\left(\tilde{\mathbf{P}}^T\mathbf{1}\right)\mathbf{y}\right) - 2\,\mathrm{tr}\left((\tilde{\mathbf{P}}\mathbf{X})^T\mathbf{T}\right) + \mathrm{tr}\left(\mathbf{T}^T\mathrm{d}(\tilde{\mathbf{P}}\mathbf{1})\mathbf{T}\right)\right) \ . \tag{3.16}$$

To implement the above computational formulation, a pseudo-code of the material deformation finding algorithm is given as follows.

Pseudo-code for the material deformation finding algorithm

- Initialization: $\mathbf{V} = 0, \sigma^2 = \frac{1}{DNm}\sum_{j=1}^{m}\sum_{I=1}^{N}\|\mathbf{y}_j - \mathbf{X}_I\|^2$
- Initialize model parameters: $w, \beta, \lambda, \tilde{\mathbf{W}}$
- calculate $\mathbf{G}$: $g_{ij} = \exp\left(-\frac{1}{2\beta^2}\|\mathbf{y}_i - \mathbf{y}_j\|^2\right)$
- Repeat the following E-step and M-step until convergence is reached:
  - E-step: Compute $\mathbf{P}$ and $\tilde{\mathbf{P}}$:
    - $p_{jI} = \dfrac{\tilde{w}_m^{1-D/2}\cdot\exp\left(-\frac{1}{2\tilde{\sigma}_m^2}\|\mathbf{X}_I - \mathbf{y}_j + G(I,\cdot)\mathbf{V}\|^2\right)}{\sum_{k=1}^{m}\tilde{w}_m^{1-D/2}\cdot\exp\left(-\frac{1}{2\tilde{\sigma}_m^2}\|\mathbf{X}_I - \mathbf{y}_k + G(K,\cdot)\mathbf{V}\|^2 + \frac{w}{1-w}\frac{(2\pi\sigma^2)^{D/2}m}{N}\right)}$
    - $\tilde{p}_{jI} = \frac{p_{jI}}{\tilde{w}_m}$
  - M -step:
    - Solve $\mathbf{V}$ based on $\left(\mathbf{G} + \lambda\sigma^2 d(\tilde{\mathbf{P}}\mathbf{1})^{-1}\right)\mathbf{V} = d(\tilde{\mathbf{P}}\mathbf{1})^{-1}\tilde{\mathbf{P}}\mathbf{y} - \mathbf{X}$
    - $N_{\tilde{\mathbf{P}}} = \mathbf{1}^T\tilde{\mathbf{P}}\mathbf{1}, \mathbf{T} = \mathbf{X} + \mathbf{G}\mathbf{V}$
    - $\sigma^2 = \frac{1}{N_P D}\left(\mathrm{tr}\left(\mathbf{y}^T\mathrm{d}\left(\tilde{\mathbf{P}}^T\mathbf{1}\right)\mathbf{y}\right) - 2\,\mathrm{tr}\left((\tilde{\mathbf{P}}\mathbf{y})^T\mathbf{T}\right) + \mathrm{tr}\left(\mathbf{T}^T\mathrm{d}(\tilde{\mathbf{P}}\mathbf{1})\mathbf{T}\right)\right)$

Remark 3.3.1. *In the proposed material deformation finding algorithm, there are three free parameters: w, α, β. Parameter w indicates our prior belief on the level of noise in the data set, and both α and β determine the level of motion coherence that one wants to enforce. We can tune all three free parameters until finding the best matching result.*

3.4 Accelerated Computing Implementation

In engineering application, the numbers of design points and scan points are usually very different. The number of design points are usually more than one hundred thousand, i.e. $N > 100,000$, whereas the number of the scan points can exceed 1 million, i.e. $M > 1,000,000$. The two most time consuming steps in the application of the proposed MDF algorithm are: To compute $\tilde{\boldsymbol{P}}\mathbf{1}$, $\tilde{\boldsymbol{P}}^T\mathbf{1}$ and $\tilde{\boldsymbol{P}}\boldsymbol{X}$, and to solve Eq.(3.15).

Fast Gauss transform

In computing matrix-vector multiplications, it is found that the sum equations having such form as

$$f(y_j) = \sum_{i=1}^{N} q_i \exp\left(-\left\|y_j - x_i\right\|^2 / h_i^2\right), j = 1, 2, \cdots, M \tag{3.17}$$

always appear in $\tilde{\boldsymbol{P}}\mathbf{1}$, $\tilde{\boldsymbol{P}}^T\mathbf{1}$ and $\tilde{\boldsymbol{P}}\boldsymbol{X}$ so that the *Fast Gauss Transform (FGT)* method is introduced to convert $\mathcal{O}(MN)$ operations to $\mathcal{O}(M+N)$.

Remark 3.4.1. *Due to different covariances $\tilde{\sigma}_m^2$ in various components, the bandwidth h_i^2 in the exponential sum above are various. It is different from exponential sum in the CPD method whose bandwidth h^2 is constant.*

Using the improved fast gauss transform algorithm in high dimension [78], we can deal with variable bandwidth Gaussian kernels, and the exponential sum can be expanded as,

$$\begin{aligned} f(y_j) &= \sum_{i=1}^{N} q_i \exp\left(-\frac{\left\|y_j - x_i\right\|^2}{h_i^2}\right) \\ &= \sum_{i=1}^{N} q_i \exp\left(-\frac{\left\|x_i - x_*\right\|^2}{h_i^2}\right) \times \exp\left(-\frac{\left\|y_j - x_*\right\|^2}{h_i^2}\right) \times \exp\left(\frac{2\left(y_j - x_*\right)\left(x_i - x_*\right)}{h_i^2}\right) \end{aligned} \tag{3.18}$$

in which,

$$\exp\left(-\frac{\left\|y_j - x_*\right\|^2}{h_i^2}\right) \approx \sum_{m=0}^{p_1^i - 1} \frac{(-1)^m}{m!} \left(\frac{y_j - x_*}{h_i}\right)^{2m}, \tag{3.19}$$

by using Taylor's expansion in multi-dimension.

Define the multi-index notation $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_d)$ with the following rules,

$$\begin{aligned} |\alpha| &= \alpha_1 + \alpha_2 + \cdots + \alpha_d, \\ \alpha! &= \alpha_1!\alpha_2! \cdots \alpha_d! \\ \boldsymbol{x}^\alpha &= x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d} \text{ if } \boldsymbol{x} \in \mathbb{R}^d \end{aligned} \tag{3.20}$$

We can then write

$$\begin{aligned} \exp\left(\frac{2\left(y_j - x_*\right)\left(x_i - x_*\right)}{h_i^2}\right) &\approx \sum_{n=1}^{p_2^i - 1} \frac{2^n}{n!} \left[\frac{\left(y_j - x_*\right)\left(x_i - x_*\right)}{h_i^2}\right]^n \\ &= \sum_{|\alpha| \le p_2^i - 1} \frac{2^\alpha}{\alpha!} \left(\frac{1}{h_i^2}\right)^2 \left(y_j - x_*\right)^\alpha \left(x_i - x_*\right)^\alpha \end{aligned} \tag{3.21}$$

In summary, the approximated exponential sum $\hat{f}(y_j)$ can be written as,

$$\begin{aligned}\hat{f}(y_j) &= \sum_{i=1}^{N} q_i \exp\left(-\frac{\|x_i - x_*\|^2}{h_i^2}\right) \times \exp\left(-\frac{\|y_j - x_*\|^2}{h_i^2}\right) \cdot \exp\left(\frac{2(y_j - x_*)(x_i - x_*)}{h_i^2}\right) \\ &= \sum_{i=1}^{N} q_i \exp\left(-\frac{\|x_i - x_*\|^2}{h_i^2}\right) \times \left[\sum_{m=0}^{p_1^i - 1} \frac{(-1)^m}{m!}\left(\frac{y_j - x_*}{h_i}\right)^{2m}\right] \\ &\quad \times \left[\sum_{|\alpha| \le p_2^i - 1} \frac{2^\alpha}{\alpha!}\frac{1}{h_i^{2\alpha}}(y_j - x_*)^\alpha (x_i - x_*)^\alpha\right] . \end{aligned} \tag{3.22}$$

After regrouping, we can put it into a succinct form,

$$\begin{aligned}\hat{f}(y_j) &= \sum_{m=0}^{p_1^i - 1} \sum_{|\alpha| \le p_2^i - 1} \frac{(-1)^m 2^\alpha}{m!\alpha!} \sum_{i=1}^{N} q_i \exp\left(-\frac{\|x_i - x_*\|^2}{h_i^2}\right) \\ &\quad \times \left(\frac{1}{h_i^2}\right)^{m+|\alpha|} (x_i - x_*)^\alpha \|y_j - x_*\|^{2m} (y_j - x_*)^\alpha . \end{aligned} \tag{3.23}$$

Low-rank matrix approximation

The $M \times M$ matrix $\boldsymbol{G}$ with element $g_{ij} = \exp\left(-\frac{1}{2}\frac{\|\boldsymbol{y}_{0i} - \boldsymbol{y}_{0j}\|^2}{\beta^2}\right)$ is a dense one. Once the number of scan data points increases to be very large, it requires lots of storage resources so that the MDF algorithm is NOT practical in some applications. Besides, solving Eq.(3.15) needs $\mathcal{O}(M^3)$ operations, which has a very large computational cost. Therefore, the low-rank matrix approximation of matrix $\boldsymbol{G}$ is used,

$$\boldsymbol{G} \approx \tilde{\boldsymbol{G}} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^T \tag{3.24}$$

in which, the diagonal matrix Λ stores the K largest eigenvalues and the matrix $\boldsymbol{Q}$ stores corresponding eigenvectors.

By the Woodbury identity [91], the matrix inversion in Eq.(3.15) can be shown as,

$$\begin{aligned}\left[\boldsymbol{Q}\Lambda\boldsymbol{Q}^T + 2\sigma^2\lambda diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right)^{-1}\right]^{-1} =& \frac{1}{2\lambda\sigma^2} diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right) - \frac{1}{(2\lambda\sigma^2)^2} diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right)\boldsymbol{Q} \\ &\times \left[\Lambda^{-1} + \frac{1}{2\lambda\sigma^2}\boldsymbol{Q}^T diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right)\boldsymbol{Q}\right]^{-1} \boldsymbol{Q} diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right) \end{aligned} \tag{3.25}$$

the computation complexity can be decreased from $\mathcal{O}(M^3)$ to $\mathcal{O}(K^3)$.

The complexity comparison test is conducted in Intel Core i5-6440HQ CPU with serial processing. So that one is to have time costs comparison as shown in Fig.3.4. It is found

that when product $M \times N$ of two data sets exceeds 10^6 the effect of computing acceleration becomes obvious. Currently the acceleration can save about 75% time, when $M \times N = \mathcal{O}(10^9)$ and it is believed that the computation efficiency can increase further when more data are included.

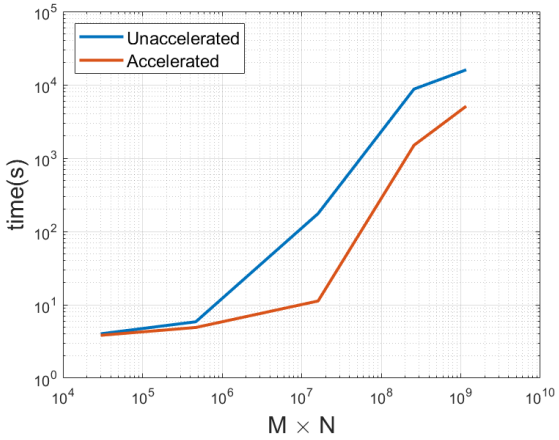


Figure 3.4: Computation complexity comparison between unaccelerated and accelerated algorithm.

Chapter 4

A Mixed Variational Bayesian Learning Finite Element Method

In this chapter, we develop a mixed variational Bayesian learning finite element method (VBL-FEM), based on a Bayesian statistical continuum mechanics theory, in which elastic potential energy is used as a prior in a Bayesian regularization network, which can intelligently recover unknown continuum deformation mapping with only the information of the shapes of the deformed and undeformed continuum body without knowing actual boundary conditions, both traction as well as displacement boundary conditions and the precise material constitutive relation. Moreover, we develop the related finite element formulation in a computational probabilistic mechanics framework.

4.1 A Mixed Entropic Variational Principle of Probabilistic Elastic Continuum

In continuum mechanics, almost all finite element method formulations are based on energetic variational principles. In this work, we developed finite element method based on a probabilistic entropic variational principle. to solve the inverse problem of continuum mechanics. To do so, we first consider a generative model [36] to describe the formulation of $\mathbf{x}$ from $\mathbf{X}$. As a starting point, we consider that the reference configuration and current configuration are represented by two different meshes or two different sets of points that have no one-to-one correspondence as shown in Fig. 4.1. Using the terminology in computer vision, we may call the points $\mathbf{X}$ in the reference configuration as the centroids of a GMM and the points $\mathbf{x}$ in current configuration as the data points, which are usually the scan data or image data. Under this circumstance, we can move the centroid points $\mathbf{X}$ in the reference configuration towards the data points $\mathbf{x}$ in current configuration to recover the deformation map. Similar with the updated Lagrangian method, the recovery displacement $\mathbf{u}_n(\mathbf{X})$ in the

deformation recovery process is defined as

$$\mathbf{u}_n(\mathbf{X}) = \mathbf{X}_n - \mathbf{X}_0 \quad \rightarrow \quad \boldsymbol{\psi}_n(\mathbf{x}) = \mathbf{X}_0 + \mathbf{u}_n \tag{4.1}$$

where $\mathbf{X}_0 = \mathbf{X}$ is the initial position for the material point $\mathbf{X}$ in the reference configuration, and $\mathbf{X}_n$ is the updated position for the same material point after nth iteration. In computation, all the quantities are computed in the moving mesh in the updated configuration.

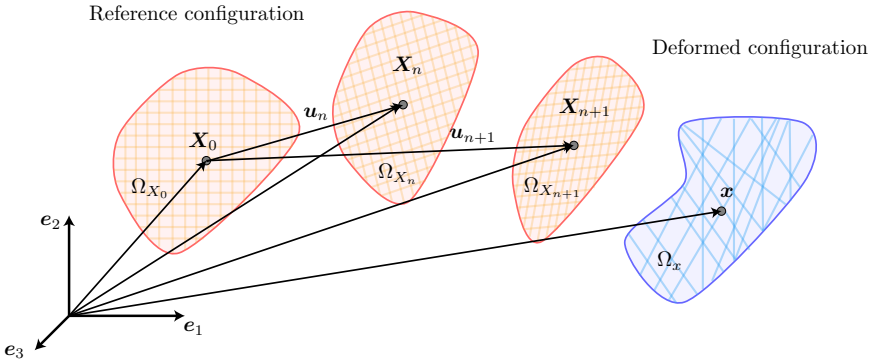


Figure 4.1: Schematic illustration of the inverse solution of continuum deformation mapping recovery problem.

It may be noted that the determination of which configuration to move and which to unmove is totally arbitrary. Therefore, it is feasible to move the current configuration towards to the reference configuration, Ω_x, to recover the inverse of deformation mapping. We can let $\mathbf{x}_0 = \mathbf{x}$ and the recovered displacement field is defined as

$$\mathbf{u}_n(\mathbf{x}) = \mathbf{x}_0 - \mathbf{x}_n \quad \rightarrow \quad \boldsymbol{\psi}_n^{-1}(\mathbf{x}) = \mathbf{x}_0 - \mathbf{u}_n(\mathbf{x}) \ . \tag{4.2}$$

This inverse recovery setting would be further discussed in section 5.2. Again, it is worthy emphasizing that $\mathbf{x}_0 = \mathbf{x}$ and $\mathbf{X}_0 = \mathbf{X}$ are not the same material point.

Nevertheless, in the following presentation, we assume that the material point $\mathbf{X} \in \Omega_X$ move towards the current configuration. Then for a specific data point $\mathbf{x} \in \Omega_x$, the generative model can be established in a two step procedure. First, a latent variable $\mathbf{X}_n \in \Omega_{X_n}$ is generated, and then depending on the value of the probability distribution parameter Δ, a data point $\mathbf{x} \in \Omega_x$ can be related with the variable $\mathbf{X}_n$ by the selected Gaussian distribution centered at $\mathbf{X}_n$ as follows,

$$p(\mathbf{x}, \mathbf{X}_n) = p(\mathbf{X}_n)p(\mathbf{x}|\mathbf{X}_n) \tag{4.3}$$

Then, by law of total probability, the total probability of the specific data point $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \int_{\Omega_X} p(\mathbf{X}_n)p(\mathbf{x}|\mathbf{X}_n)d\Omega_X, \tag{4.4}$$

where $p(\mathbf{X}_n)$ is the prior distribution of the material point $\mathbf{X}$. After applying the corresponding displacement $\mathbf{u}$, the Gaussian distribution $p(\mathbf{x}|\mathbf{X}_n)$ is given as

$$p(\mathbf{x}|\mathbf{X}_n) = \frac{1}{(2\pi\Sigma_n^2)^{D/2}} \exp\Big(-\frac{\|\mathbf{X}+\mathbf{u}_n-\mathbf{x}\|^2}{2\Sigma_n^2}\Big). \tag{4.5}$$

In order to distinguish the Cauchy stress with the Gaussian distribution variance, the variance is denoted by capital letter Σ. Then the log-likelihood of the probability of all data points $\mathbf{x}$ can be expressed as

$$\int_{\Omega_x} \log(p(\mathbf{x}))d\Omega_x = \int_{\Omega_x} \log\Big(\int_{\Omega_{X_n}} p(\mathbf{X}_n)p(\mathbf{x}|\mathbf{X}_n)d\Omega_X\Big)d\Omega_x. \tag{4.6}$$

We propose the following Bayesian probabilistic elastic entropy potential,

$$\Pi(\mathbf{u},\Sigma) := -\int_{\Omega_x} \log\Big(\int_{\Omega_X} p(\mathbf{X})p(\mathbf{x}|\mathbf{X})d\Omega_X\Big)d\Omega_x + \gamma\int_{\Omega_X}\phi(\mathbf{u})d\Omega_X, \tag{4.7}$$

where $\phi(\mathbf{u})$ is a regularization function, λ is a constant. We choose the probability of the strain energy state as the prior probability and the probability of the L_2 error norm between the current positions of material points and their measured position as the likelihood function, i.e.

$$p(\mathbf{X}) \;=\; \exp(-\beta W(\mathbf{u}))p_0(\mathbf{X}), \tag{4.8}$$

where $p_0(\mathbf{X})$ is the original material probability distribution in the referential configuration. While the regularization function is a stabilized term, which may be chosen as

$$\phi(\mathbf{u}) = \frac{1}{2}\mathbf{u}^2. \tag{4.9}$$

Remark 4.1.1. *One can see that the above probabilistic elastic potential entropy has two sources: (1) The material-based probabilistic entropy, and (2) the deformation-based probabilistic entropy. In this work, we choose the material-based probability distribution as a uniform probability distribution, which means that we do not consider the effects of material heterogeneity, which will be studies in the late sequence of this work.*

The essence of the VBL-FEM is to minimize the elastic potential entropy by using finite element method. The mixed functional in Eq. (4.7) can be minimized by using a staggered EM algorithm [24, 13]. In EM algorithm, we first utilize Bayes' theorem to calculate the previous step posterior probability distribution $p(\mathbf{X}_n \mid \mathbf{x})$ based on the previous displacement field $\mathbf{u}_n$. This is the expectation or the E-step in the algorithm. Then the unknown

displacement $\mathbf{u}_n$ can be updated by minimizing the expectation of the complete negative log-likelihood function,

$$\begin{aligned} Q(\mathbf{u}_n, \Sigma_n) &= -\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}_n \mid \mathbf{x}\right) \log\Big(p(\mathbf{X}+\mathbf{u}_n)p(\mathbf{x}|\mathbf{X}+\mathbf{u}_n)\Big) d\Omega_X d\Omega_x \\ &\quad +\gamma\int_{\Omega_X}\phi(\mathbf{u}_n)d\Omega_X, \end{aligned} \tag{4.10}$$

which is the so-called maximization or the M-step. The posterior probability distribution $p\left(\mathbf{X}_n \mid \mathbf{x}\right)$ can be expressed as

$$\begin{aligned} p\left(\mathbf{X}_n \mid \mathbf{x}\right) &= p\left(\mathbf{X}+\mathbf{u}_n \mid \mathbf{x}\right) \\ &= \frac{p\left(\mathbf{x} \mid \mathbf{X}+\mathbf{u}_n\right) p\left(\mathbf{X}+\mathbf{u}_n\right)}{p\left(\mathbf{x}\right)} \\ &= \frac{\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{X}+\mathbf{u}_n-\mathbf{x}}{\Sigma_n}\right\|^2 - \beta W(\mathbf{u}_n)\right)}{\displaystyle\int_{\Omega_X}\exp\left(-\frac{1}{2}\left\|\frac{\mathbf{X}+\mathbf{u}_n-\mathbf{x}}{\Sigma_n}\right\|^2 - \beta W(\mathbf{u}_n)\right) d\Omega_X} \end{aligned} \tag{4.11}$$

The EM algorithm is an iterative approach which is proceed by applying E-step and M-step alternatively until convergence, which is guaranteed based on the fact that the probabilistic entropy potential is monotonous decreasing.

To show that the proposed Bayesian probabilistic entropy potential in a deformable continuum is a monotonous decreasing sequence, we first express Eq. (4.7) as

$$\begin{aligned} \Pi(\mathbf{u}_n, \Sigma_n) &= -\int_{\Omega_x}\log p(\mathbf{x})d\Omega_x + \gamma\int_{\Omega_X}\phi(\mathbf{u}_n)d\Omega_X \\ &= -\int_{\Omega_x}\log p(\mathbf{x}, \mathbf{X}_n)d\Omega_x + \int_{\Omega_x}\log p(\mathbf{X}_n|\mathbf{x})d\Omega_x + \gamma\int_{\Omega_X}\phi(\mathbf{u}_n)d\Omega_X. \end{aligned} \tag{4.12}$$

Then taking the expectation of Eq. (4.12) over all possible values of the latent variable $\mathbf{X}$ under the current parameter estimate $\mathbf{u}_n, \Sigma_n$ by multiplying both sides by $p\left(\mathbf{X}_n \mid \mathbf{x}\right)$ and integrate over Ω_X. We have,

$$\begin{aligned} \Pi(\mathbf{u}_n, \Sigma_n) &= -\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}_n \mid \mathbf{x}\right) \log\Big(p(\mathbf{X}_n)p(\mathbf{x}|\mathbf{X}_n)\Big) d\Omega_X d\Omega_X + \gamma\int_{\Omega_x}\phi(\mathbf{u}_n)d\Omega_X \\ &\quad + \int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}_n \mid \mathbf{x}\right) \log p(\mathbf{X}_n|\mathbf{x}) d\Omega_X d\Omega_x \\ &= Q(\mathbf{u}_n, \Sigma_n) + H(\mathbf{u}_n, \Sigma_n), \end{aligned} \tag{4.13}$$

where $H(\mathbf{u}_n, \Sigma_n)$ is defined as

$$H(\mathbf{u}_n, \Sigma_n) = \int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}_n \mid \mathbf{x}\right) \log p(\mathbf{X}_n|\mathbf{x}) d\Omega_X d\Omega_x. \tag{4.14}$$

By Gibbs' inequality, we have

$$H(\mathbf{u}_n, \Sigma_n) \geq H(\mathbf{u}, \Sigma), \qquad \forall \mathbf{u}, \Sigma. \tag{4.15}$$

Hence we can calculate the difference between two probabilistic elastic potentials corresponding to two adjacent deformation configurations,

$$\begin{aligned} \Pi(\mathbf{u}_{n+1}) - \Pi(\mathbf{u}_n) &= Q(\mathbf{u}_{n+1}, \Sigma_{n+1}) - Q(\mathbf{u}_n, \Sigma_n) + H(\mathbf{u}_{n+1}, \Sigma_{n+1}) - H(\mathbf{u}_n, \Sigma_n) \\ &\leq Q(\mathbf{u}_{n+1}, \Sigma_{n+1}) - Q(\mathbf{u}_n, \Sigma_n) \\ &\leq 0. \end{aligned} \tag{4.16}$$

Hence during the EM iteration the probabilistic entropic potential is monotonously decreasing.

After showing the monotonous decreasing property of the probabilistic elastic potential, we focus on the M-step in EM algorithm which may be formulated as the following optimization problem.

Infinitesimal Deformation Recovery Inverse Problem (Variational Form).
Find $\mathbf{u} : \Omega \longrightarrow \mathbb{R}^D$ and variance Σ^2, such that it minimizes the probabilistic potential energy

$$(\mathbf{u}^\star, \Sigma^\star) = \underset{\mathbf{u}, \Sigma \in \mathcal{V} \oplus \mathbb{R}^+}{\arg\min}\ Q(\mathbf{u}, \Sigma) \tag{4.17}$$

where the probabilistic potential energy is given as

$$\begin{aligned} Q(\mathbf{u}, \Sigma) \quad = \quad & \int_{\Omega_x} \int_{\Omega_X} p\left(\mathbf{X} + \mathbf{u} \mid \mathbf{x}\right) \cdot \left(\frac{\left\| \mathbf{X} + \mathbf{u} - \mathbf{x} \right\|^2}{2\Sigma^2} + D \log \Sigma + \beta W(\mathbf{u}) \right) d\Omega_X d\Omega_x \\ & + \gamma \int_{\Omega_X} \phi(\mathbf{u}) d\Omega_X \end{aligned} \tag{4.18}$$

where D is the space dimension, and $D = 2, 3$.

Theorem 4.1.1. *For $\mathbf{u} \in \mathcal{V}$ and $\Sigma \in \mathcal{R}^+$, there exists a unique solution $(\mathbf{u}^\star, \Sigma^\star)$, such that*

$$Q(\mathbf{u}^\star, \Sigma^\star) = \inf_{\mathbf{u}, \Sigma \in \mathcal{V} \oplus \mathcal{R}^+} Q(\mathbf{u}, \Sigma)$$

Proof. The stationary condition for attaining the extreme of $Q(\mathbf{u}, \Sigma)$ is

$$\delta Q(\mathbf{u}, \Sigma) = \delta_\Sigma Q(\mathbf{u}, \Sigma) + \delta_\mathbf{u} Q(\mathbf{u}, \Sigma) = 0, \quad \forall (\delta\mathbf{u}, \delta\Sigma) \in \mathcal{V}_0 \oplus \mathcal{R}^+ \tag{4.19}$$

where

$$\delta_\Sigma Q(\mathbf{u}, \Sigma) = \int_{\Omega_x} \int_{\Omega_X} p\left(\mathbf{X} + \mathbf{u} \mid \mathbf{x}\right) \cdot \left(-\frac{\left\| \mathbf{X} + \mathbf{u} - \mathbf{x} \right\|^2}{\Sigma^3} + \frac{D}{\Sigma} \right) \cdot \delta\Sigma d\Omega_X d\Omega_x \tag{4.20}$$

and

$$\begin{aligned}\delta_{\mathbf{u}}Q(\mathbf{u},\Sigma) &= \beta\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\cdot\boldsymbol{\Sigma}(\boldsymbol{\varepsilon}(\mathbf{u})):\delta\boldsymbol{\varepsilon}d\Omega_X d\Omega_x \\ &\quad+\frac{1}{\Sigma^2}\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\cdot(\mathbf{X}+\mathbf{u}-\mathbf{x})\delta\mathbf{u}d\Omega_X d\Omega_x+\gamma\int_{\Omega_x}\delta\phi(\mathbf{u})d\Omega_x \end{aligned} \tag{4.21}$$

Since $\delta\Sigma$ and $\delta\mathbf{u}$ are independent, we have

$$\delta Q(\mathbf{u},\Sigma)=0\ \rightarrow\delta_{\mathbf{u}}Q(\mathbf{u},\Sigma)=0\ \text{ and }\ \delta_{\Sigma}Q(\mathbf{u},\Sigma)=0\ .$$

From $\delta_{\Sigma}Q(\mathbf{u},\Sigma)=0$ and using Eq. (4.20), we can obtain

$$\Sigma^2=\frac{1}{D}\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\left\|\mathbf{X}+\mathbf{u}-\mathbf{x}\right\|^2 d\Omega_X d\Omega_x \tag{4.22}$$

The sufficient condition for $Q(\mathbf{u},\Sigma^2)$ having an infimum is:

$$\delta^2Q(\mathbf{u},\Sigma^2)=\delta^2_{\mathbf{uu}}Q(\mathbf{u},\Sigma)+2\delta^2_{\mathbf{u}\Sigma}Q(\mathbf{u},\Sigma)+\delta^2_{\Sigma\Sigma}Q(\mathbf{u},\Sigma)>0\ , \tag{4.23}$$

where

$$\delta^2_{\mathbf{uu}}Q=\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\cdot\Big(\delta\boldsymbol{\varepsilon}:\gamma\frac{\partial^2 W}{\partial\varepsilon\partial\varepsilon}:\delta\boldsymbol{\varepsilon}+\frac{1}{\Sigma^2}\delta\mathbf{u}\cdot\delta\mathbf{u}\Big)d\Omega_X d\Omega_x+\gamma\delta\mathbf{u}\cdot\frac{\partial^2\phi}{\partial\mathbf{u}\partial\mathbf{u}}\cdot\delta\mathbf{u} \tag{4.24}$$

$$\delta^2_{\Sigma\Sigma}Q = \int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\cdot\left(\frac{3\left\|\mathbf{X}+\mathbf{u}-\mathbf{x}\right\|^2}{\Sigma^4}-\frac{D}{\Sigma^2}\right)\cdot(\delta\Sigma)^2 d\Omega_X d\Omega_x \tag{4.25}$$

$$\delta^2_{\Sigma u}Q = -\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\cdot\left(\frac{(\mathbf{X}+\mathbf{u}-\mathbf{x})}{\Sigma^3}\right)\cdot\delta\mathbf{u}\delta\Sigma d\Omega_X d\Omega_x \tag{4.26}$$

We can rewrite $\delta^2Q(\mathbf{u},\Sigma)$ as

$$\begin{aligned}\delta^2Q(\mathbf{u},\Sigma)=&\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\Big(\delta\boldsymbol{\varepsilon}:\gamma\frac{\partial^2 W}{\partial\boldsymbol{\varepsilon}\partial\boldsymbol{\varepsilon}}:\delta\boldsymbol{\varepsilon}\Big)d\Omega_X d\Omega_x+\gamma\int_{\Omega_x}\delta\mathbf{u}\cdot\frac{\partial^2\phi}{\partial\mathbf{u}\partial\mathbf{u}}\cdot\delta\mathbf{u}d\Omega_x \\ &+\frac{1}{\Sigma^2}\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\Big(\delta\mathbf{u}-\frac{\mathbf{X}+\mathbf{u}-\mathbf{x}}{\Sigma^2}\delta\Sigma\Big)^2 d\Omega_X d\Omega_x \\ &+\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\left(\frac{2\left\|\mathbf{X}+\mathbf{u}-\mathbf{x}\right\|^2}{\Sigma^4}-\frac{D}{\Sigma^2}\right)\cdot(\delta\Sigma)^2 d\Omega_X d\Omega_x\ .\end{aligned} \tag{4.27}$$

Because the regularization term is positive definite, the first term in Eq. (4.27) is positive (definite), if the strain energy is convex. The second term of Eq. (4.27) is always positive.

Considering Eq. (4.22), we can rewrite the third term of Eq. (4.27) as

$$\begin{aligned}
&\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}\mid\mathbf{x}\right)\left(\frac{2\left\|\mathbf{X}+\mathbf{u}-\mathbf{x}\right\|^2}{\Sigma^4}-\frac{D}{\Sigma^2}\right)\cdot(\delta\Sigma)^2 d\Omega_X d\Omega_x \\
&=\Big(\frac{2\Sigma^2 D}{\Sigma^4}-\frac{D}{\Sigma^2}\Big)(\delta\Sigma)^2 \\
&=D\frac{(\delta\Sigma)^2}{\Sigma^2}>0\ .
\end{aligned} \tag{4.28}$$

Thus, we conclude that

$$\delta^2 Q(\mathbf{u},\Sigma)>0\ . \tag{4.29}$$

That is: the mixed probabilistic variational principle (Eq.(4.19) is a minimum there is a solution $(\mathbf{u}^\star,\Sigma^\star)\in\mathcal{V}\oplus\mathcal{R}^+$ such that it minimizes $Q(\mathbf{u},\Sigma)$, i.e.

$$Q(\mathbf{u}^\star,\Sigma^\star)=\inf_{(\mathbf{u},\Sigma)\in\mathcal{V}\oplus\mathbb{R}^+} Q(\mathbf{u},\Sigma)\ . \tag{4.30}$$

□

Remark 4.1.2. *As shown in Eq. (4.18), the functional $Q(\mathbf{u},\Sigma)$ is a Kullback-Leibler divergence, or it is a relative entropy. This is the foundational origin of the maximum principle stated in Theorem 2.1.*

4.2 Variational Bayesian Learning Finite Element Formulation and Implementation

We now present the VBL-FEM formulation for general elastic continua. Note that the elastic continuum itself is not necessarily statistical or stochastic. it is lacking of information, or epistemic uncertainty, that makes VEL-FEM formulation probabilistic.

Assume that in an elastic continuum, the stress-strain state is $(\boldsymbol{\sigma}(\mathbf{x}),\boldsymbol{\epsilon}(\mathbf{x})),\forall\mathbf{x}\in\Omega$. Based on Theorem 2.1, to maximize the Kullback-Leibler divergence or the relative entropy, we first consider the following variational formulation:

$$\begin{aligned}
\delta_{\mathbf{u}}Q(\mathbf{u}_n,\Sigma_n) \;=\;& \beta\int_{\Omega_X}\int_{\Omega_x} p\left(\mathbf{X}+\mathbf{u}_n\mid\mathbf{x}\right)\cdot\boldsymbol{\sigma}(\mathbf{X}):\delta\boldsymbol{\varepsilon}\, d\Omega_X d\Omega_x \\
&+\frac{1}{\Sigma^2}\int_{\Omega_X}\int_{\Omega_x} p\left(\mathbf{X}+\mathbf{u}_n\mid\mathbf{x}\right)\cdot(\mathbf{X}+\mathbf{u}_n-\mathbf{x})\delta\mathbf{u}\, d\Omega_X d\Omega_x \\
&+\gamma\int_{\Omega_X}\mathbf{u}_n\cdot\delta\mathbf{u}\, d\Omega_X=0.
\end{aligned} \tag{4.31}$$

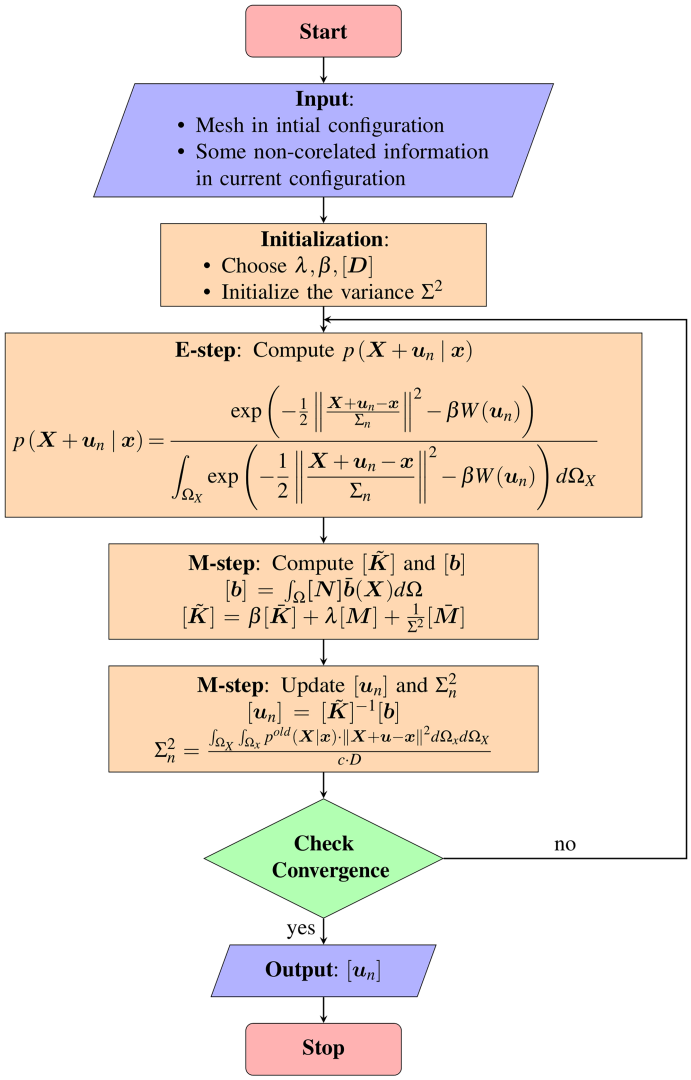


Figure 4.2: Flowchart for VBL-FEM inverse recovery algorithm for continuum deformation mapping.

The equivalent stress $\bar{\boldsymbol{\sigma}}(\mathbf{X})$ and the equivalent body force $\bar{\mathbf{b}}(\mathbf{X})$ that share similar properties as their finite element counterparts can be defined as

$$\bar{\boldsymbol{\sigma}}(\mathbf{X}) = \int_{\Omega_x} p\left(\mathbf{X}+\mathbf{u}_n \mid \mathbf{x}\right) \cdot \boldsymbol{\sigma}(\mathbf{X}+\mathbf{u}_n) d\Omega_x, \tag{4.32}$$

and

$$\bar{\mathbf{b}}(\mathbf{X}) = \frac{1}{\Sigma^2} \int_{\Omega_x} p\left(\mathbf{X}+\mathbf{u}_n \mid \mathbf{x}\right) \cdot (\mathbf{x}-\mathbf{X}-\mathbf{u}_n) d\Omega_x. \tag{4.33}$$

Furthermore, to simplify Eq. (4.31), we define an equivalent probability $\bar{p}(\mathbf{X})$ as

$$\bar{p}(\mathbf{X}) = \int_{\Omega_x} p\left(\mathbf{X} + \mathbf{u}_n \mid \mathbf{x}\right) d\Omega_x. \tag{4.34}$$

Hence, Eq. (4.31) can be rewritten as follow:

$$\beta \int_{\Omega_X} \bar{\boldsymbol{\sigma}} : \delta\boldsymbol{\epsilon} d\Omega_X + \gamma \int_{\Omega_X} \mathbf{u} : \delta\mathbf{u} d\Omega_X + \frac{1}{\Sigma^2} \int_{\Omega_X} \bar{p}\mathbf{u} : \delta\mathbf{u} d\Omega_X - \int_{\Omega_X} \bar{\mathbf{b}} : \delta\mathbf{u} d\Omega_X = \mathbf{0}, \tag{4.35}$$

which is the Galerkin weak formulation or the so-called principle of virtual work (PVW) for the VBL-FEM. Considering finite element interpolation e.g. [42],

$$\mathbf{u}^h = [\mathbf{N}][\mathbf{d}], \quad \delta\mathbf{u}^h = [\mathbf{N}][\delta\mathbf{d}], \quad \text{and} \quad \boldsymbol{\epsilon}^h = [\mathbf{B}][\mathbf{d}] \tag{4.36}$$

the entropic Galerkin weak formulation can be discretized as:

$$\int_{\Omega_X} [\mathbf{B}]^T \bar{\boldsymbol{\sigma}} d\Omega_X + \gamma \int_{\Omega_X} [\mathbf{N}]\mathbf{u} d\Omega_X + \frac{1}{\Sigma^2} \int_{\Omega_X} [\mathbf{N}]\bar{p}\mathbf{u} d\Omega_X - \int_{\Omega_X} [\mathbf{N}]\bar{\mathbf{b}} d\Omega_X = 0, \tag{4.37}$$

where $[\mathbf{N}]$ is the global shape function matrix, $[\mathbf{B}]$ is the global strain-displacement matrix, and $[\mathbf{d}]$ is the finite element nodal displacement. Then the equivalent stress $\bar{\boldsymbol{\sigma}}$ can be defined as

$$\bar{\boldsymbol{\sigma}}(\mathbf{X}) = \bar{p}(\mathbf{X})[\mathbf{D}][\mathbf{B}][\mathbf{d}], \tag{4.38}$$

where $[\mathbf{D}]$ is the the elastic modulus matrix and $[\mathbf{u}]$ is the global nodal displacement vector. Let the internal force vector $\mathbf{f}^{int}$ and the external force vector $\mathbf{f}^{ext}$ be expressed as

$$\mathbf{f}^{int} = \int_{\Omega_X} [\mathbf{B}]^T \bar{\boldsymbol{\sigma}} d\Omega_X + \gamma \int_{\Omega_X} [\mathbf{N}]\mathbf{u} d\Omega_X + \frac{1}{\Sigma^2} \int_{\Omega_X} [\mathbf{N}]\bar{p}\mathbf{u} d\Omega_X \tag{4.39}$$

and

$$\mathbf{f}^{ext} = \int_{\Omega_X} [\mathbf{N}]\bar{\mathbf{b}} d\Omega_X. \tag{4.40}$$

Then, Eq. (4.37) can also be expressed as,

$$\mathbf{f}^{int} - \mathbf{f}^{ext} = 0, \tag{4.41}$$

which is the discrete version of equilibrium equation in continuum mechanics.

For each element in the reference configuration, the element nodal force is defined as

$$\begin{aligned} \mathbf{f}_e^{int} =& \int_{\Omega_e} [\mathbf{B}]^T \bar{\boldsymbol{\sigma}} d\Omega_e + \gamma \int_{\Omega_e} [\mathbf{N}]\mathbf{u} d\Omega_e + \frac{1}{\Sigma^2} \int_{\Omega_e} [\mathbf{N}]\bar{p}\mathbf{u} d\Omega_e \\ =& \beta[\bar{\mathbf{K}}_e][\mathbf{d}_e] + \gamma[\mathbf{M}_e][\mathbf{d}_e] + \frac{1}{\Sigma^2}[\bar{\mathbf{M}}_e][\mathbf{d}_e], \end{aligned} \tag{4.42}$$

where $[\mathbf{d}]_e$ is element nodal displacement vector, and $[\bar{\mathbf{K}}_e]$, $[\mathbf{M}_e]$ and $[\bar{\mathbf{M}}_e]$ are defined as

$$[\bar{\mathbf{K}}_e] = \int_{\Omega_e} \bar{p}(\mathbf{X})[\mathbf{B}]^T[\mathbf{D}][\mathbf{B}]d\Omega_e, \tag{4.43}$$

$$[\mathbf{M}_e] = \int_{\Omega_e} [\mathbf{N}]^T[\mathbf{N}]d\Omega_e, \quad \text{and} \quad [\bar{\mathbf{M}}_e] = \int_{\Omega_e} \bar{p}(\mathbf{X})[\mathbf{N}]^T[\mathbf{N}]d\Omega_e. \tag{4.44}$$

Correspondingly, the equivalent element nodal force $[\mathbf{b}_e]$ can be defined as follow:

$$[\mathbf{b}_e] = \mathbf{f}_e^{ext} = \int_{\Omega_e} [\mathbf{N}]\bar{\mathbf{b}}(\mathbf{X})d\Omega_e \ . \tag{4.45}$$

After obtaining the element equivalent stiffness matrix $[\bar{\mathbf{K}}_e]$, the global equivalent stiffness matrix $[\bar{\mathbf{K}}]$ can be connected by the connectivity matrix $[\mathbf{L}_e]$, which is

$$[\bar{\mathbf{K}}] = \mathop{\mathbf{A}}_{e=1}^{N_e} [\mathbf{L}_e]^T[\bar{\mathbf{K}}_e], \tag{4.46}$$

where N_e is the total number of elements, and $\mathbf{A}_{e=1}^{N_e}$ element assembly operator [42]. Similarly, the global matrices $[\mathbf{M}]$, $[\bar{\mathbf{M}}]$ and $[\mathbf{b}]$ can also be assembled as

$$[\mathbf{M}] = \mathop{\mathbf{A}}_{e=1}^{N_e} [\mathbf{L}_e]^T[\mathbf{M}_e], \quad [\bar{\mathbf{M}}] = \sum_e [\mathbf{L}_e]^T[\bar{\mathbf{M}}_e], \quad \text{and} \quad [\mathbf{b}] = \sum_e [\mathbf{L}_e]^T[\mathbf{b}_e]. \tag{4.47}$$

Therefore, substituting Eq. (4.46) and Eq. (4.47) into Eq. (4.37), one obtains

$$\mathbf{f}^{int} - \mathbf{f}^{ext} = \beta[\bar{\mathbf{K}}][\mathbf{d}] + \gamma[\mathbf{M}][\mathbf{d}] + \frac{1}{\Sigma^2}[\bar{\mathbf{M}}][\mathbf{d}] - [\mathbf{b}] = 0. \tag{4.48}$$

If we define the total stiffness matrix $[\tilde{\mathbf{K}}]$ as

$$[\tilde{\mathbf{K}}] = \beta[\bar{\mathbf{K}}] + \gamma[\mathbf{M}] + \frac{1}{\Sigma^2}[\bar{\mathbf{M}}], \tag{4.49}$$

by substituting Eq. (4.49) into Eq. (4.48), the discrete FEM governing equation becomes,

$$[\tilde{\mathbf{K}}][\mathbf{d}] = [\mathbf{b}], \tag{4.50}$$

which has the same form as that in the static finite element analysis. Finally, the nodal displacement vector $[\mathbf{d}]$ can be solved by the same methods used in classical finite element, such as

$$[\mathbf{d}] = [\tilde{\mathbf{K}}]^{-1}[\mathbf{b}]. \tag{4.51}$$

Next, we consider the variational formulation for the statistical parameter:

$$\begin{aligned}\delta_\Sigma Q(\mathbf{u}_n, \Sigma_n) = &\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X} + \mathbf{u}_n \mid \mathbf{x}\right) \cdot \left(-\frac{\|\mathbf{X} + \mathbf{u}_n - \mathbf{x}\|^2}{\Sigma_n^3} + \frac{D}{\Sigma_n}\right) \cdot \delta\Sigma d\Omega_X d\Omega_x \\ &= 0, \ \forall \delta\Sigma \in \mathbb{R}^+\end{aligned} \tag{4.52}$$

Note that the above equation is valid for any arbitrary $\delta\Sigma \in \mathbb{R}^+$. As a result, the Eq. (4.52) implies that

$$\int_{\Omega_x}\int_{\Omega_X} p\left(\mathbf{X}+\mathbf{u}_n \mid \mathbf{x}\right)\cdot\left(-\frac{\left\|\mathbf{X}+\mathbf{u}_n-\mathbf{x}\right\|^2}{\Sigma_n^3}+\frac{D}{\Sigma_n}\right)\cdot d\Omega_X d\Omega_x = 0. \tag{4.53}$$

After some algebraic manipulations, Eq. (4.53) can be expressed as

$$\begin{aligned}\Sigma^2 =& \frac{\int_{\Omega_X}\int_{\Omega_x} p\left(\mathbf{X}+\mathbf{u}_n \mid \mathbf{x}\right)\cdot\left\|\mathbf{X}+\mathbf{u}_n-\mathbf{x}\right\|^2 d\Omega_x d\Omega_X}{D\int_{\Omega_X}\int_{\Omega_x} p\left(\mathbf{X}+\mathbf{u}_n \mid \mathbf{x}\right) d\Omega_x d\Omega_X}\\ =& \frac{\int_{\Omega_X}\int_{\Omega_x} p\left(\mathbf{X}+\mathbf{u}_n \mid \mathbf{x}\right)\cdot\left\|\mathbf{X}+\mathbf{u}_n-\mathbf{x}\right\|^2 d\Omega_x d\Omega_X}{c\cdot D},\end{aligned} \tag{4.54}$$

where constant c represents the area of the mesh in the current configuration. After solve the nodal displacement vector $[\mathbf{u}]$ in Eq. (4.51), all the variables on the right hand side of the Eq. (4.54) are given. Therefore, the Gaussian distribution variance Σ^2 can be obtained by applying Gauss quadrature to integrate the numerator in Eq. (4.54). In summary, the procedure of VBL-FEM is shown as Fig. 4.2.

Chapter 5

Numerical Examples

In this chapter we present several numerical examples for both MDF and VBL-FEM. The MDF algorithm is validated by a synthetic 2D problem and applied to compensate distortion in AM. Then we demonstrate in several examples that the VBL-FEM is able to inversely predict continuum deformation mappings with strong discontinuity or fracture without knowing the external load conditions.

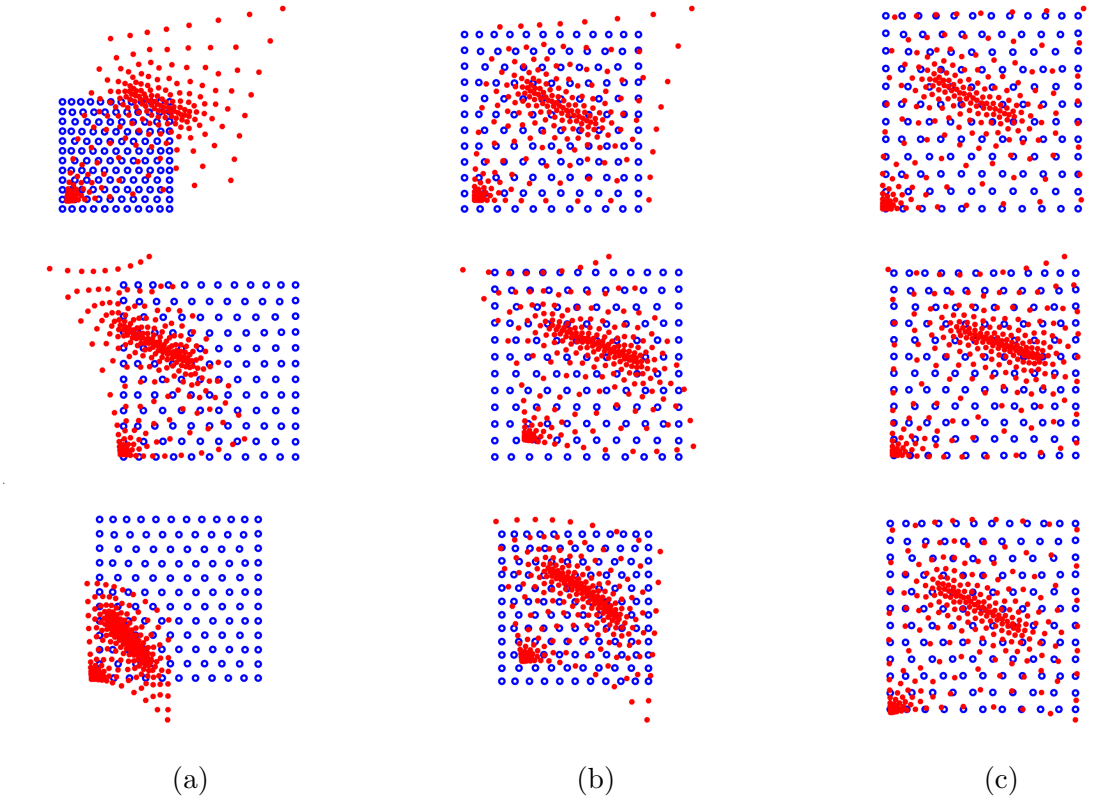
5.1 MDF Experiments and Results

In this section, three examples are presented to demonstrate the accuracy, robustness, and improvement of the material deformation finding algorithm over the original CPD algorithm. The first example is a synthetic 2D problem. The last two experiments are a real application in 3D printing. In this example, the material deformation finding algorithm was used to find the displacement field of a distorted printed part.

Registration between uniform mesh and locally refined non-uniform mesh

As stated previously, the most difficult aspect of determining the distortion displacement field for a 3D-printed structure component is that we only know the spatial positions of material points in the current configuration and not the corresponding material coordinates in the referential configuration. To calculate the displacement field, one must determine the material coordinates of all spatial material locations using material registration. In this example, the suggested MDF method was used to recover the material coordinates for the scanned material points shown in Fig. 3.1 I(b) based on a set of reference or created nodal points indicated in Fig. 3.1 II(a).

To begin, the open source general purpose mesh generator was used to build the triangular meshes shown in Fig. 3.1 I(a) and Fig. 3.1 II(a). The polynomial displacement field with random coefficients was then used to construct three distinct arbitrary non-uniform



(a) (b) (c)

Figure 5.1: 2D point sets registration results. (a) three different nonrigid deformations before registration. (b) the registration results of CPD algorithm. (c) the registration results of material deformation finding algorithm.

deformations. Random coefficients with varied amounts of non-uniform deformation were selected for each of the three sets. After that, the locally refined mesh was subjected to the non-uniform movements or deformations selected(see Fig. 5.1 (a)).

In this numerical experiment, Table 5.1 lists the hyperparameter values for both the MDF and CPD algorithms for all three situations. These hyperparameter settings were selected such that both the MDF and CPD algorithms yielded optimal matching results.

Hyperparameters	α	β	w
Case 1	3	20	0.05
Case 2	3	3	0.05
Case 3	3	2	0.05

Table 5.1: Values of hyperparameters for all three cases.

The results of the registration are shown in Fig. 5.1 for three distinct prescribed deformation scenarios. The experimental settings for the three distinct experiments are shown in the left column. The CPD algorithm's registration results are shown in the center column of the table. The registration results obtained via the use of the MDF method are shown in the right column.

This figure indicates that the non-uniform deformation in all circumstances can be successfully represented by the suggested MDF approach, particularly for the border points that determine the shape of the domain, while the original CPD method fails to register the two data sets.

In addition, as seen in Figure 5.2, in order to quantitatively evaluate the outcomes of the CPD and MDF algorithms, the average registration errors of the four corner points and all the data points in the three scenarios are compared using Euclidean distance.

Note that the average registration error for all data points is the weighted mean with the weight vector $\tilde{\mathbf{W}}$ in Eq. (3.3), whereas the average registration error for the four corner points is simply the arithmetic mean. The performance of the suggested MDF algorithm is consistently and noticeably superior than that of the CPD algorithm, regardless of the extent of non-uniform deformation.

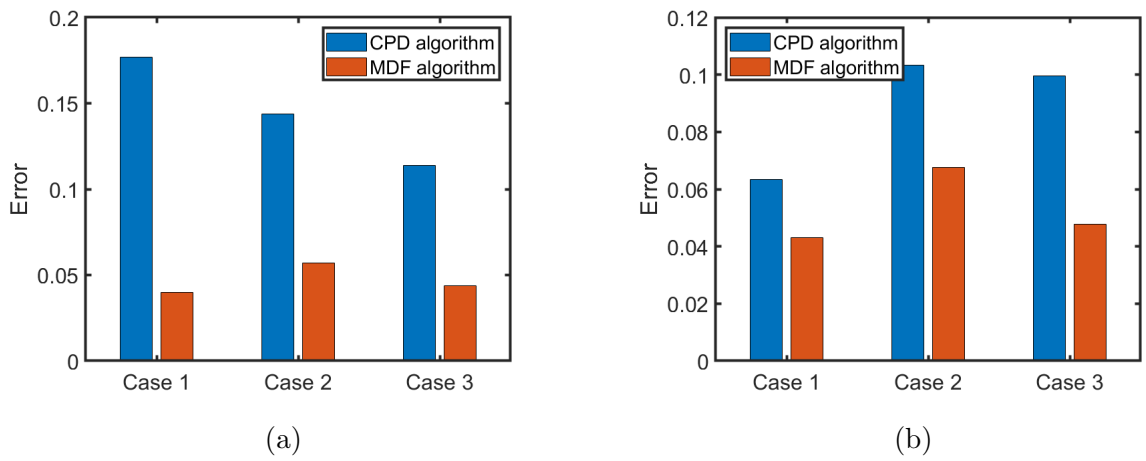


Figure 5.2: (a) Average registration errors for four corner points, and (b) Average registration errors for all data points.

Lattice structure registration

Lattice structures that have a large number of holes or voids in their models are attracting a growing amount of interest and attention. This is due to the fact that models including these interior gaps allow for lighter components while yet retaining outstanding mechanical performance. However, traditional subtractive manufacturing techniques like mold manu-

facturing may not be able to produce such lattices with intricate interior structures. As a result, the use of additive manufacturing is very close to being required in order to create complicated lattice models.

The unexpected distortion brought on by thermal variation, edge swelling brought on by residual stress existing in free surfaces or thermal concentration, and the staircase effect in cutting layers are some of the most common geometric defects that occur in lattice models while they are being manufactured using additive manufacturing.

The lattice models are created by Rhinoceros [59]. It is a three-dimensional model builder with a very powerful extension thanks to Grasshopper, a plugin that builds models using visual programming. Crystallon [74] is utilized to create complicated lattice-shaped structures in our two three-dimensional models(see Fig.5.3).

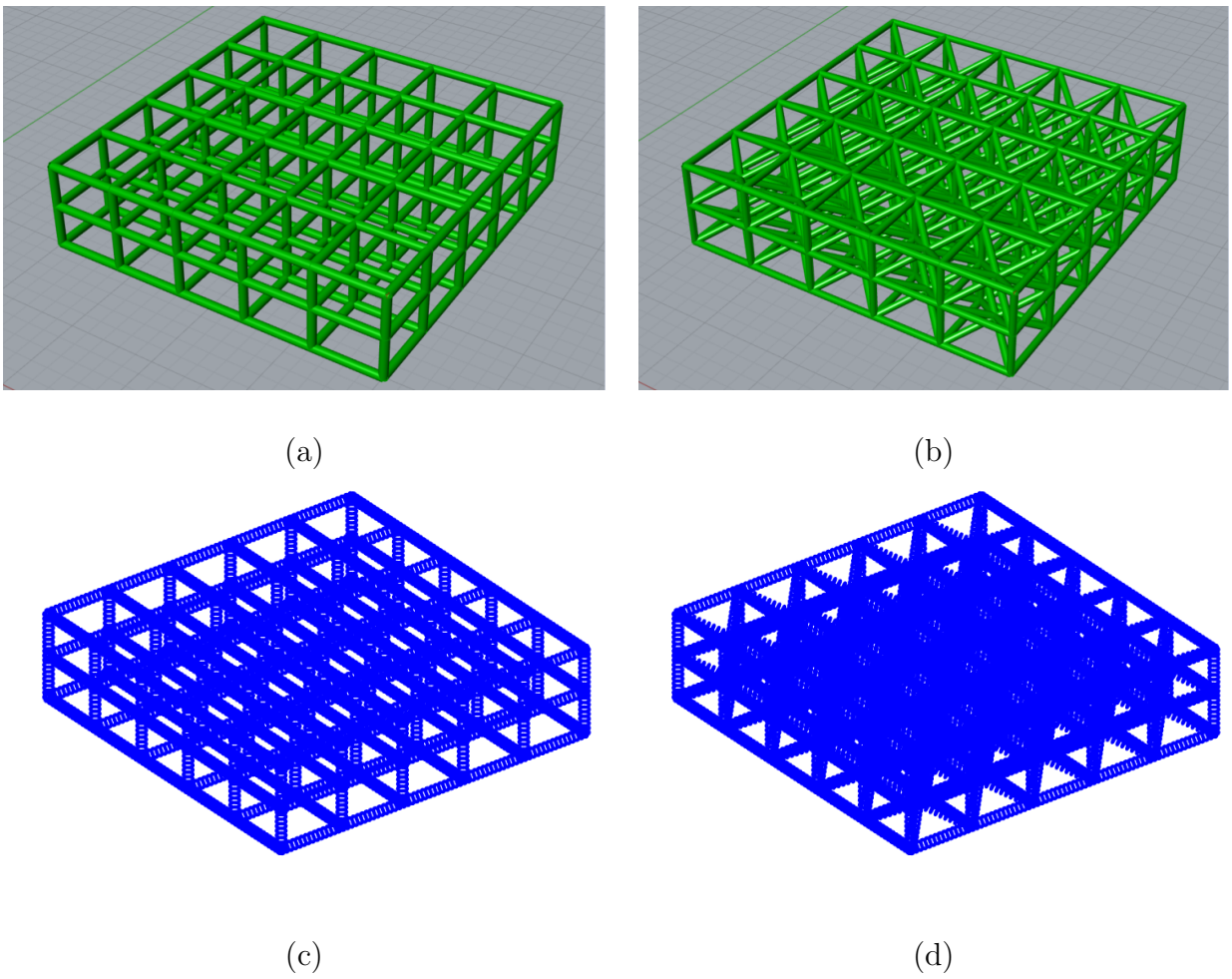


Figure 5.3: Two lattice models: (a). simple cubic lattice; (b). body-centered cubic lattice; (c). 103032 scatter nodes of simple cubic lattice; (d). 247720 scatter points of body-centered cubic lattice.

Table 5.2: Simulation Parameters in ANSYS Additive manufacturing

Material	Layer thickness	Layer power	Scanning speed	Scan width	Voxel size
IN718	$50\mu m$	195 W	1000 mm/s	$100\mu m$	$0.3\mu m$

To simulate geometric faults in lattice models, we utilized the ANSYS Additive Manufacturing program [55]. The ANSYS Additive Production program can simulate metal powder bed manufacturing using an STL file. The program provides settings for controlling the manufacturing process, and the appropriate parameters selected in our simulation are listed in the table 5.2.

Because ANSYS Additive Production software automatically compensates for geometric flaws throughout the manufacturing process, the difference between the design model and the created model is small. The non-rigid transformation is done to points of the produced model in order to have evident geometric flaws:

$$\boldsymbol{y} = \boldsymbol{RSx} + \boldsymbol{t}$$

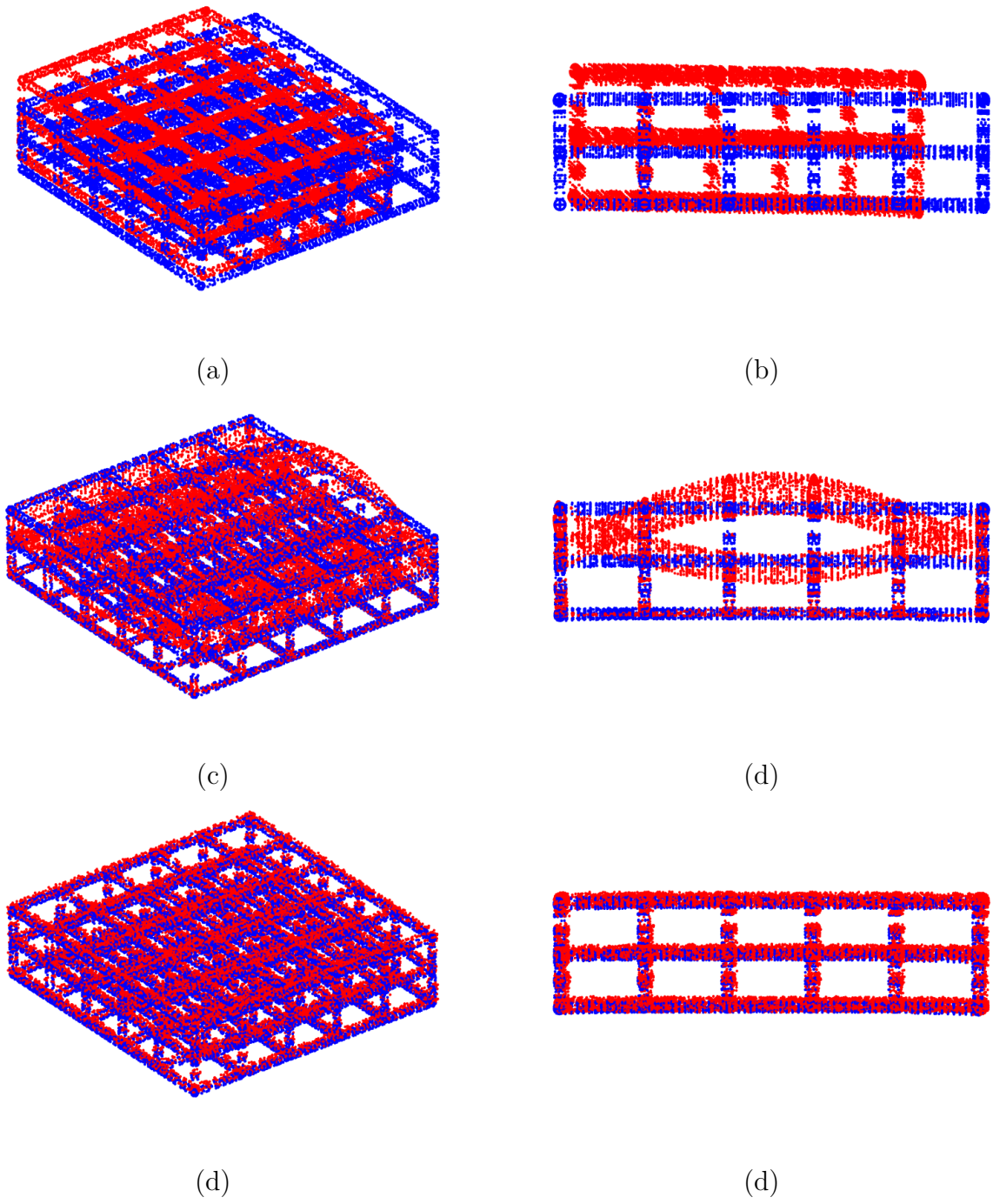
in which, $\boldsymbol{y}$ is transformed point coordinates, $\boldsymbol{x}$ is manufactured model point coordinates, $\boldsymbol{R}$ is the three-dimensional rotation matrix, $\boldsymbol{S}$ is scaling matrix and $\boldsymbol{t}$ is a translation vector.

In the non-rigid transformation, the 3D rotation is $\theta = [1°, 1°, 1°]$, the scaling factor $\boldsymbol{S} = [0.8, 1.0, 1.2]$ and translation vector $\boldsymbol{t} = [1, 1, 0]$. The points set registration comparison between the MDF method and the CPD method are shown in Fig.5.4 $\sim$ Fig.5.5. It has been discovered that the CPD method may generate obvious mismatches between the design model points and the manufactured model points, whereas the MDF method generates very excellent points set registration results, especially for the points in interior surfaces. This is especially true for the points in interior surfaces.

An additive manufacturing compensation method based on the MDF algorithm

In this experiment, the MDF method that was presented was used to solve an issue with the thermal distortion adjustment for 3D printing. Fig. 5.6 presents a flowchart of the compensation plan that incorporates the new MDF algorithm that has been developed.

This compensation plan begins with a design model including the intended geometry of a 3D-printed component. In spite of this, the 3D-printed item has undesirable distortions related to material shrinkage and different thermal-induced permanent deformations. The MDF method then registers both the design data and scanned data. By interpolating the registration result, the distortion field is formed. Finally, a new design model is reprinted with preset reverse distortion. The preset reverse distortion field mitigates or compensates for the initial distortion. Consequently, this redesigned model with reverse correction may reach a high degree of geometric precision.

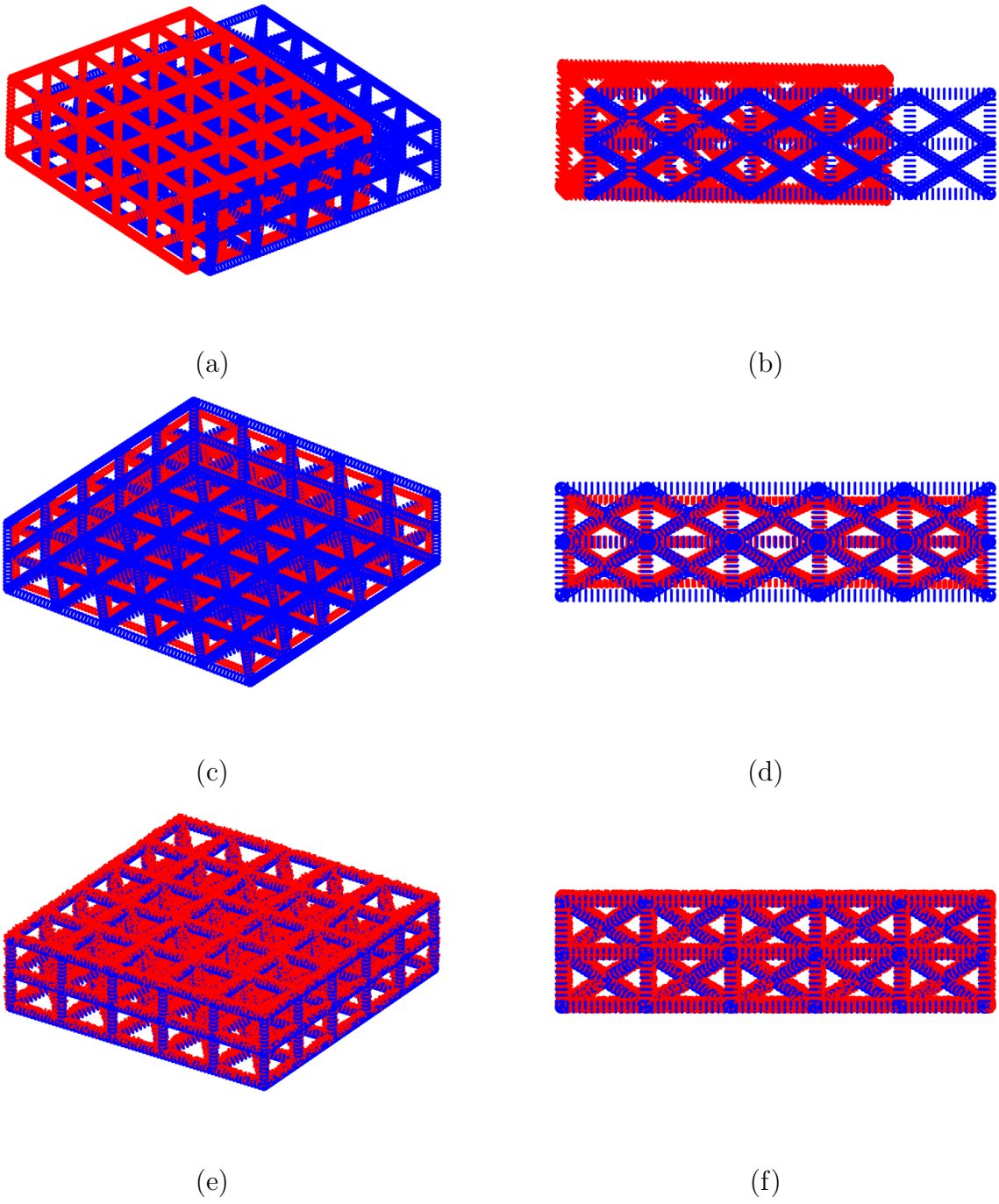


(a) (b)

(c) (d)

(d) (d)

Figure 5.4: The MDF method and the CPD method comparison for simple cubic lattice model: (a). MATLAB default view of design model (blue nodes) and transformed manufactured model (red nodes); (b). front view of design model and transformed manufactured model; (c). MATLAB default view of CPD method registered manufactured model (red nodes) and design model (blue nodes); (d). front view of CPD method registered manufactured model and design model; (e). MATLAB default view of MDF method registered manufactured model (red nodes) and design model (blue nodes); (f). front view of MDF method registered manufactured model and design model.



(a) (b)

(c) (d)

(e) (f)

Figure 5.5: The MDF method and the CPD method comparison for body-centered cubic lattice model: (a). MATLAB default view of design model (blue nodes) and transformed manufactured model (red nodes); (b). front view of design model and transformed manufactured model; (c). MATLAB default view of CPD method registered manufactured model (red nodes) and design model (blue nodes); (d). front view of CPD method registered manufactured model and design model; (e). MATLAB default view of MDF method registered manufactured model (red nodes) and design model (blue nodes); (f). front view of MDF method registered manufactured model and design model.

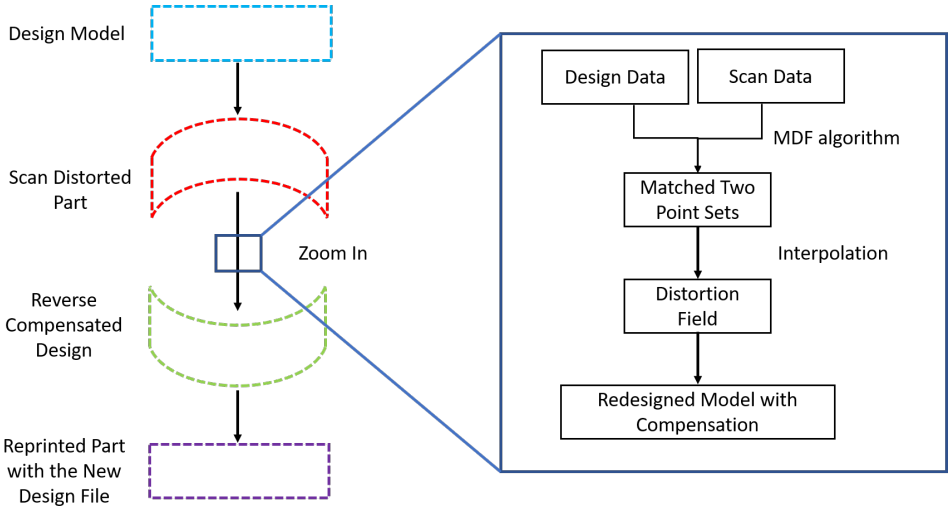


Figure 5.6: Flowchart of the compensation plan with the new proposed MDF algorithm.

In this experiment, a double cantilever beam with a sawtooth support structure was used to verify the suggested compensation strategy. Fig. 5.7(a) illustrates the geometry of the design model. The AM process parameters and design model dimensions are given in Tables 5.3 and 5.4, respectively. With these process parameters, the Ultimaker 3 manufactures a double cantilever beam. Fig. 5.7(b) depicts the deformed shape of the printed output.Note that both the original design model and the modified model with compensation were produced using the same process settings.

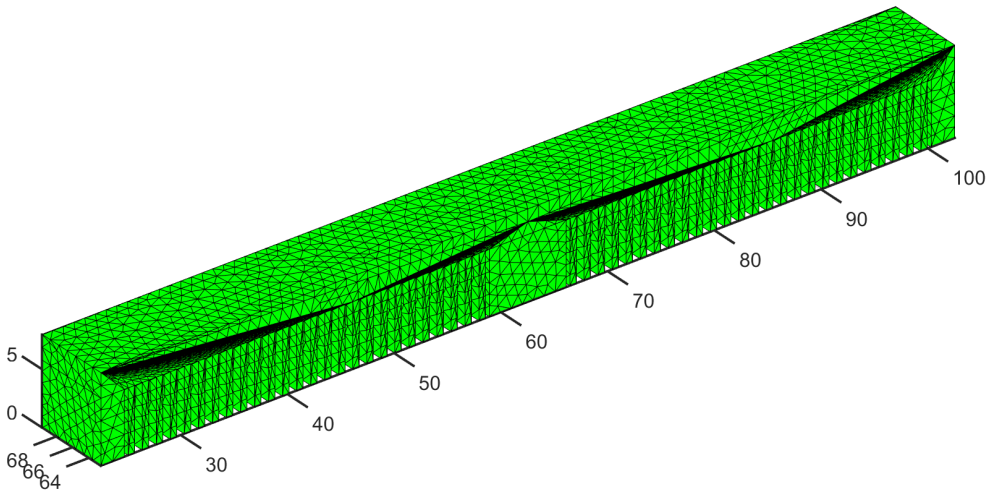
Material	Nozzle Size	Layer Height	Line Width	Print Speed	Print Temperature
Blue ABS	0.4mm	0.1mm	0.35mm	30mm/s	230 ℃

Table 5.3: Values of process parameters.

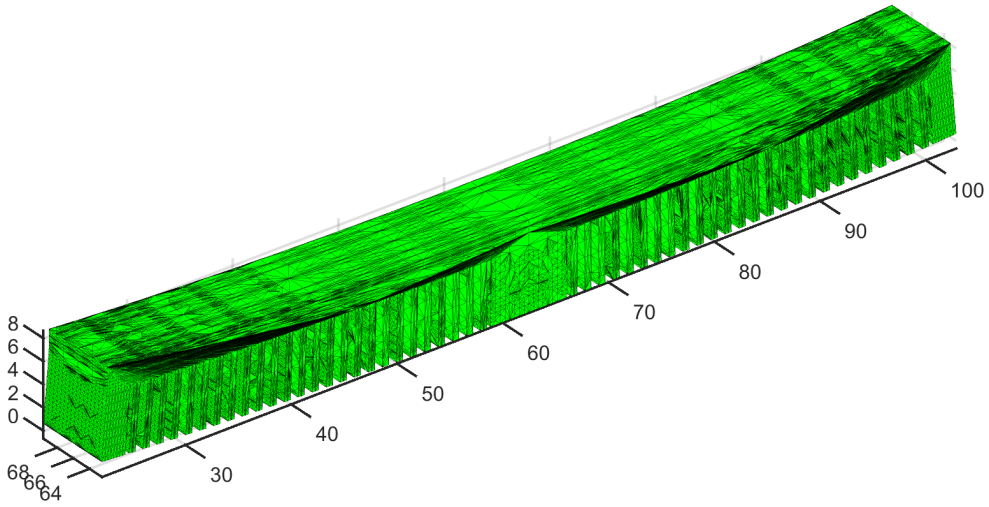
Part Length	Part Width	Part Height
80mm	7.2mm	8mm

Table 5.4: Values of design model dimensions.

After obtaining the deformed model, a 100-micron-resolution ATOS 3D scanner was used to scan it. In this instance, there are 103,784 nodes in the scanned mesh and 11,261 nodes in the design mesh, indicating that matching all the original nodes directly is not a time-efficient process.



(a)



(b)

Figure 5.7: (a) Referential mesh of the design model in the initial or material configuration, and (b) Spatial mesh and spatial material nodes of theMDF/ scanned data.

In order to overcome the aforementioned issue, a box grid filter was used to downsample the nodes in both data sets. In order to enforce the requirement of a single point inside a $1mm^3$ cube, the size of the box grid filter was set at $1mm$ in each direction for both meshes. In the case that a $1mm3$ cube has more than one node, a point was selected at random to serve as a representation for all of the points included in that cube, and the weights of the points that were removed were added to the weight of the point that was chosen. Following the use of the downsampling technique, the design data include 4,369 nodes whereas the

scanned data have 4,722 nodes. The feature points in each data set are made up of that relatively small amount of data points.

The material deformation finding algorithm then recorded two downsampled feature point sets in the subsequent phase of the process. Fig. 5.8(a) illustrates the nonrigid deformation that takes place between two feature point sets. In the simulation scenario described in this study, the following hyperparameter values were selected: $\beta = 3, \alpha = 2$, and $w = 0.1$. In this instance, the scan feature points were regarded as GMM centroids, whereas the design feature points were considered data points. As a consequence of this, the scan feature points move in a coherent way to match the design feature points, and the displacement field was interpolated by the displacements of the GMM centroids. The majority of the feature points move in a way that is consistent with the appropriate directions, as can be seen in the illustration that can be found at Fig. 5.8(b).

After matching the two point sets, the displacement field $\mathbf{u}$ was calculated by the GMM centroid displacements

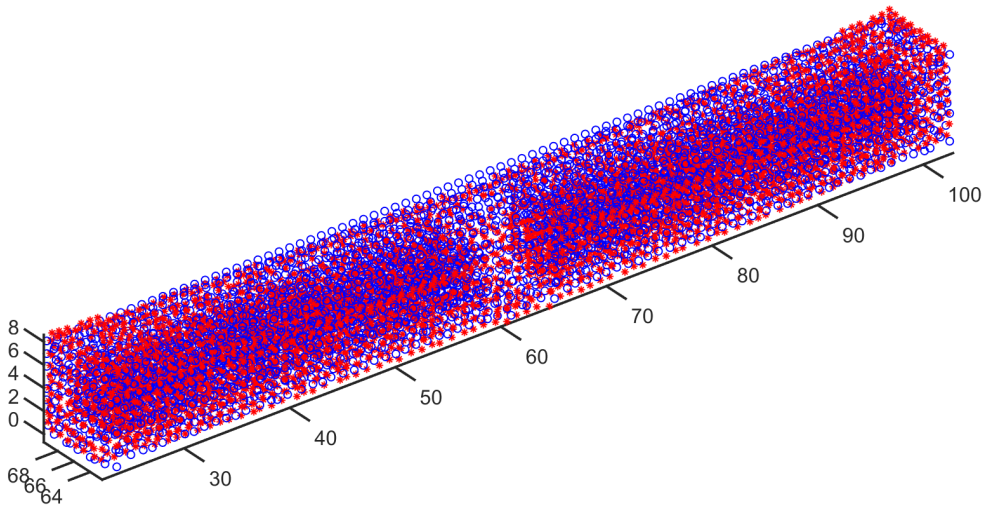
$$\mathbf{u} = \mathbf{Y}' - \mathbf{y} \tag{5.1}$$

where $\mathbf{y}$ is the spatial location of the scanned data nodes, $\mathbf{Y}'$ is the location of the nodes of the scanned data after material deformation finding. The displacement values at additional locations were then interpolated using k-nearest neighbors (KNN) regression [1]. KNN regression is a non-parametric technique that interpolates the displacement field in a new configuration using the k training data nearest to it in the feature space. Let $N_k(\mathbf{y})$ denote the k values of $\mathbf{y}_1, \ldots, \mathbf{y}_k$ that are closest to a new location $\mathbf{y}$ where need to interpolate. Then the estimator is the mean over the corresponding displacements $\mathbf{u}_i$,

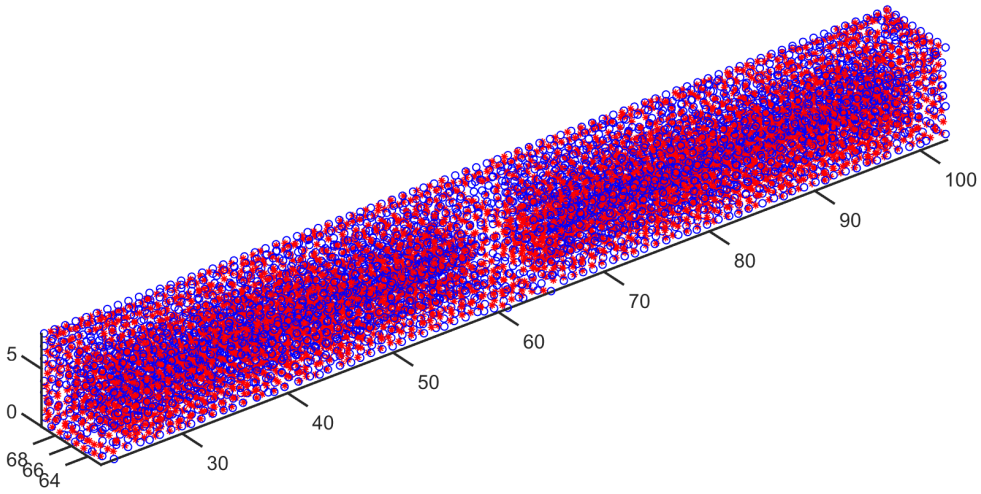
$$\mathbf{u}(\mathbf{y}) = \frac{1}{k} \sum_{i:\mathbf{y}_i \in N_k(\mathbf{y})} \mathbf{u}_i \ . \tag{5.2}$$

The motivation for this estimator is that nearby observations are representative for the samples around a particular point $\mathbf{y}$ [25]. Not usually true, it is still a good approximation, if $\mathbf{u}$ is smooth enough, and the neighborhood support is small.

After identifying the heat distortion displacement field, a 85% inverse displacement field was added to the original design mesh. The redesigned result is illustrated in Fig. 5.9. Finally, this modified model was printed using the same AM settings with additional support structure by the Ultimaker 3. Fig. 5.10 displays an overview of the suggested compensating strategy together with the printed outcome of a model that has been altered. Regardless of the quality of the support structure in the corrected model, the newly introduced inverse displacement field counteracted or minimized the thermal distortion to a significant degree. The highest distortion value was reduced from $3mm$ to $0.6mm$.



(a)



(b)

Figure 5.8: (a) Two feature point sets before the registration. (b) The registration result of two feature point sets.

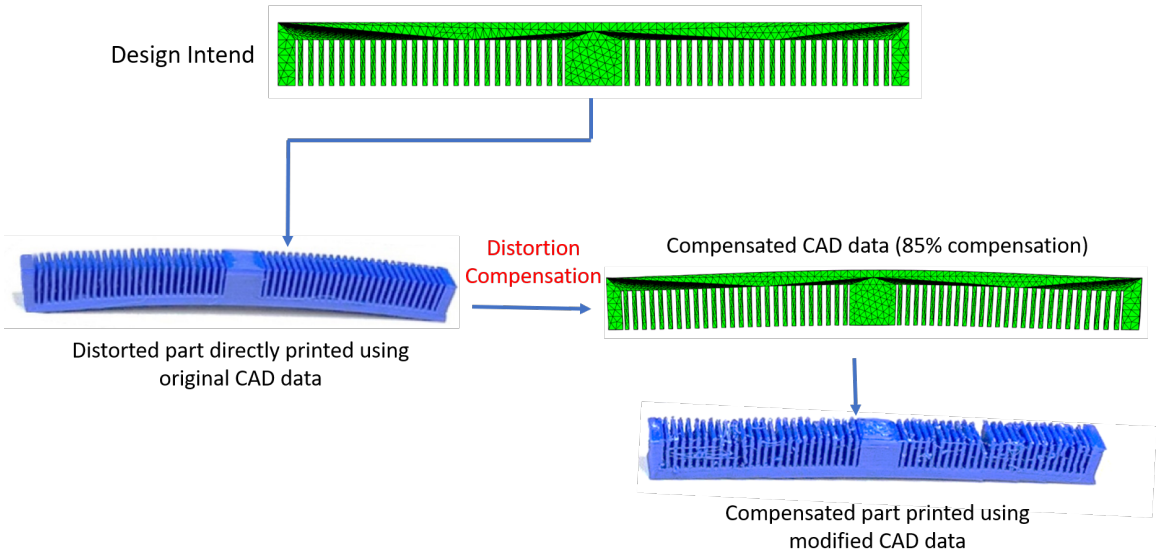


Figure 5.10: A summary of the thermal compensation of the double cantilever comb beam part. Note that the photos of the printed parts are real images of the printed products, and the small defects are due to the printing process fluctuation not from the thermal compensation algorithm.

5.2 VBL-FEM Numerical Examples

In this section, we demonstrate how the proposed VBL-FEM can be used to recover the deformation mapping. The first subsection discusses a two-dimensional thin plate with a hole problem on linear elasticity. Later, two cases with plasticity deformation and fracture propagation are studied. The recovery results for all three cases are then compared with the available numerical solutions. The implementation of VBL-FEM has been carried out using the MFEM package [3].

Linear elasticity problem of a thin plate with a hole

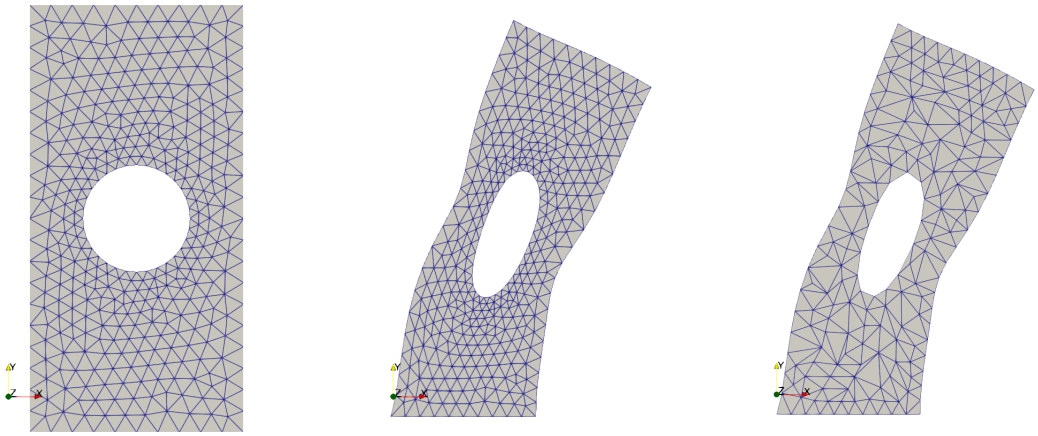
To verify the effectiveness of the proposed VBL-FEM, a thin plate with a hole under oblique traction is conducted. The dimensions of the thin plate are shown in Fig. 5.11, where $L = 20\ cm$, $l = 10\ cm$, and the thickness of the plate is $1\ cm$. A circular hole with a diameter of $d = 5\ cm$ is positioned in the center of the thin plate.

This thin plate is fixed at the bottom and an oblique uniformly distributed load $\boldsymbol{p}$ is applied at the top of the plate which reads:

$$p_x = 0.01\ MPa, \quad p_y = 0.1\ MPa \tag{5.3}$$

where p_x and p_y are the x and y components of the distributed load $\boldsymbol{p}$ respectively. The constitutive model of this plate is chosen as isotropic linear elasticity with Lamé constants $\lambda = 0.58\ MPa$ and $\mu = 0.38\ MPa$ (corresponding to Young Modulus $E = 1.0\ MPa$ and poisson ratio $v = 0.3$).

Then the thin plate is meshed by 712 linear triangular elements in Gmsh [32]. This mesh is generated uniformly on the boundary with mesh size equals 0.2 mm and local refinement around the circle hole is performed to capture the high stress and strain (see Fig. 5.12 (a)). Finally, a finite element analysis is conducted by using a open-source FEM package called MFEM [3]. The FEA deformation result is shown in Fig. 5.12 (b).



(a) (b) (c)

Figure 5.12: The finite element meshes for the thin plate example: (a) Mesh in the initial configuration; (b) The same mesh in the final configuration; (c) Coarse mesh in the final configuration.

The proposed VBL-FEM is then utilized to recover the deformation mapping. The initial and final finite element configurations, as illustrated in Fig. 5.12, are given as the inputs to the VBL-FEM. And the constitutive model of VBL-FEM is also chosen as isotropic linear elasticity. The mesh in the initial configuration is chosen as the Gaussian distributions and the mesh in the final configuration is selected as the data points. Therefore, the initial mesh moves towards the final mesh to recover the deformation.

In order to better illustrate the capacity of VBL-FEM, three different cases are conducted with increasing complexity. In the first case, the same Lamé constants, $\lambda = 0.58\ MPa$ and $\mu = 0.38\ MPa$, and the same mesh utilized in FEA are given as the inputs to the VBL-FEM model. For the second case, in order to increase the deformation recovery complexity, the Lamé constants are selected as $\lambda = 0.001\ MPa$ and $\mu = 0.001\ MPa$ which are not the real material parameters. For the last case, not only the Lamé constants are chosen as $\lambda = 0.001\ MPa$ and $\mu = 0.001\ MPa$, but also the mesh in the final configuration is

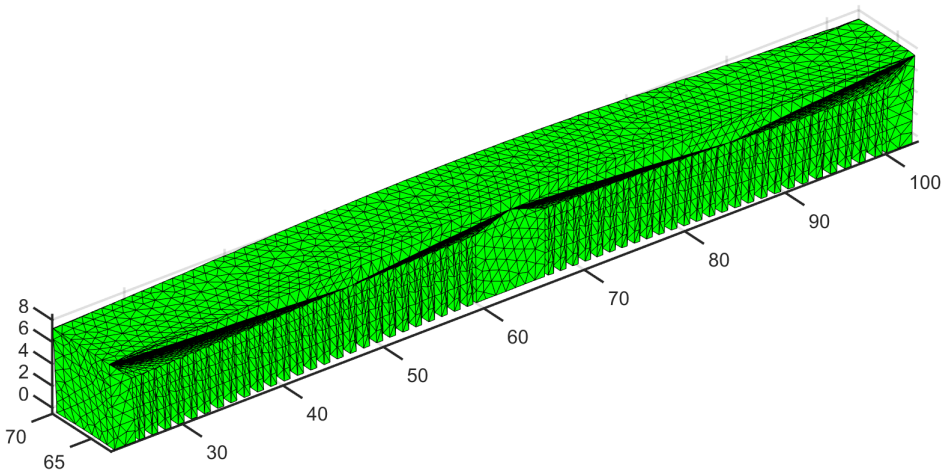


Figure 5.9: Redesigned model with KNN Regression.

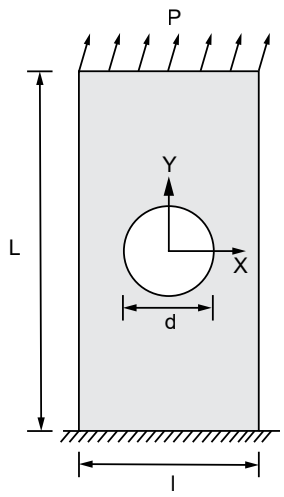


Figure 5.11: The setup of the thin plate with a hole example.

regenerated to a coarse one with only 317 linear triangular elements (see Fig. 5.12 (c)). The setups for all three case are shown in Fig. 5.13. Other model parameters for all three cases are shown in Table 5.5. Note that the values of hyperparameters β and λ are chosen to achieve the best accuracy for each case respectively.

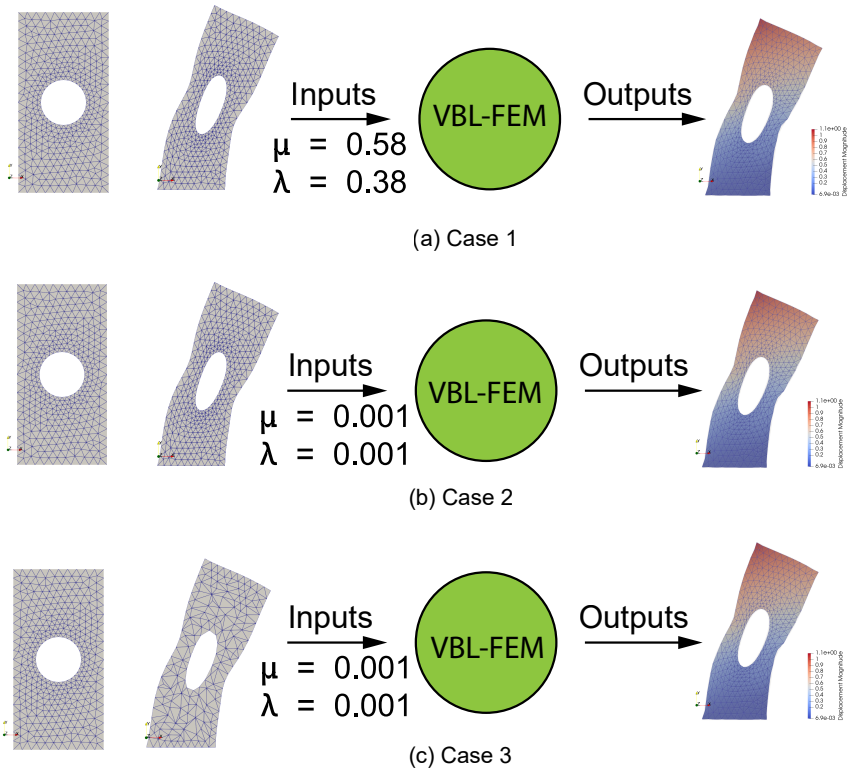


Figure 5.13: Three different setups for the VBL-FEM model with the corresponding matching results for example 1.

Table 5.5: The VBL-FEM model hyperparameters for example 1.

Model parameters	**C**ase 1	**C**ase 2	**C**ase 3
Hyperparameter β	6×10^{-7}	8×10^{-4}	8×10^{-4}
Hyperparameter λ	1×10^{-7}	1×10^{-4}	1×10^{-5}
Max number of iterations	200	200	450

Fig. 5.14 presents the displacement recovery process for case 1. The final configuration is shown in gray as the true underlying deformation mapping. The initial configuration which is color-coded by the magnitude of displacement moves along the matching process. As it can

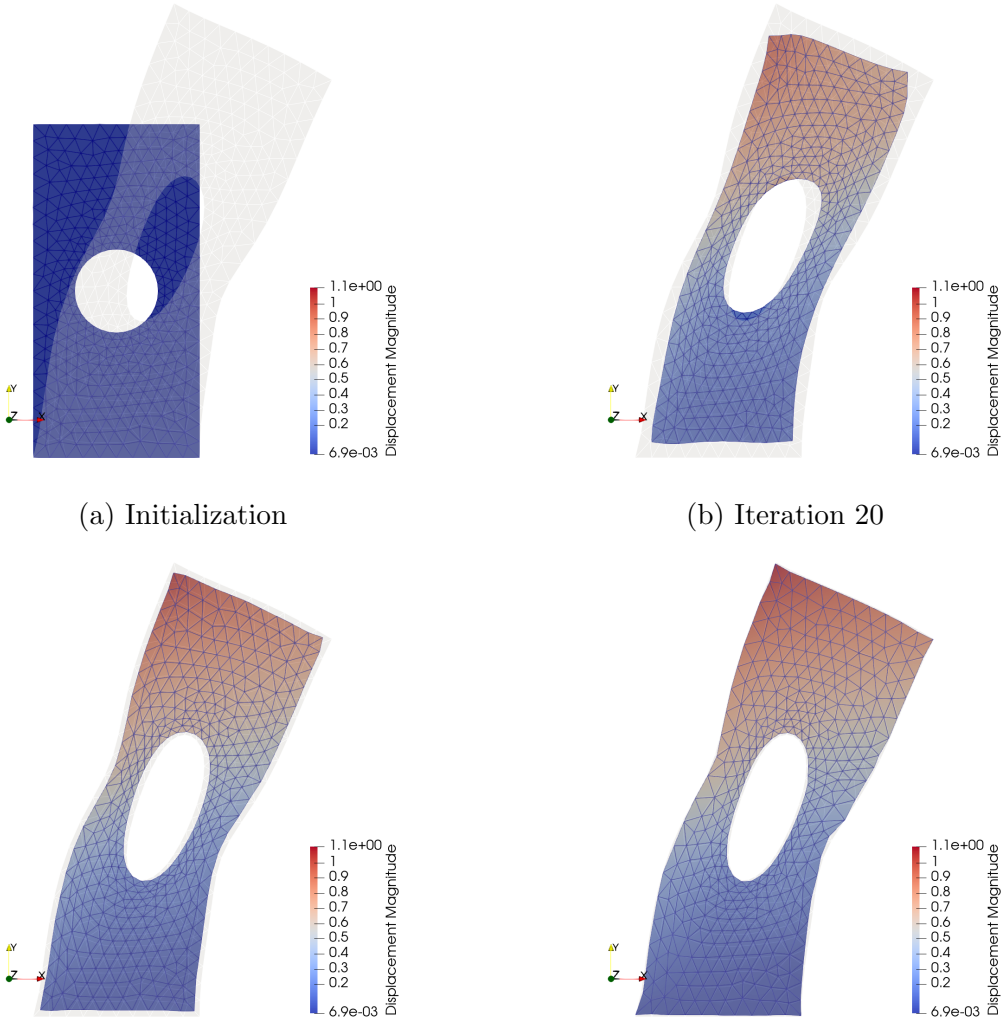
be seen from Fig. 5.14, the iteration process stops at 147 steps of iterations and recovers the majority of displacement. In order to better exemplify the accuracy of deformation recovery, the average nodal deformation error $\bar{e}$ is evaluated, which is given as

$$\bar{e} = \frac{1}{n}\sum_{i=1}^{n} e_i = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{x}_i^{FEA} - \boldsymbol{x}_i^{recovery}||_2 \tag{5.4}$$

where e_i is the nodal displacement error for the ith node, n is the number of nodes in the given mesh, $\boldsymbol{x}_i^{FEA}$ is the final position for ith node which is calculated by FEA, $\boldsymbol{x}_i^{recovery}$ is the recovered final position for ith node by the VBL-FEM model and $||\cdot||_2$ represents the Euclidean norm. The history of the average error $\bar{e}$ and the standard deviation Σ of the Gaussian distribution are shown in Fig. 5.15 (a).

It can be seen that both the average of nodal error and the standard deviation Σ decrease exponentially along the iteration process. The decrease of the standard deviation Σ indicate the increase of alignment between the two meshes. While, the average error decreases from 41 cm to 2.2 cm which indicates a recovery of 94.6% of the magnitude of the nodal displacement. Meanwhile, as displayed in Fig. 5.15 (b), the majority of nodal absolute errors are located in the $[0, 0.04]$ interval and none of the nodal absolute error is larger than 0.18 cm. It clearly shows the recovery accuracy for all nodal points in this thin plate example.

Similarly, the displacement recovery process for case 2 and case 3 are shown in Fig. 5.17 and Fig. 5.19, respectively. The error histories and final error histograms for both cases are illustruted in Fig. 5.16 and Fig. 5.18 respectively. Moreover, the comparison of the displacement recovery accuracy among all three cases is shown in Table 5.6. As one can see that the displacements are recovery with high accuracies in all three cases, even though different meshes are used to represent the reference and current configurations and different material models are chosen in the recovery process.



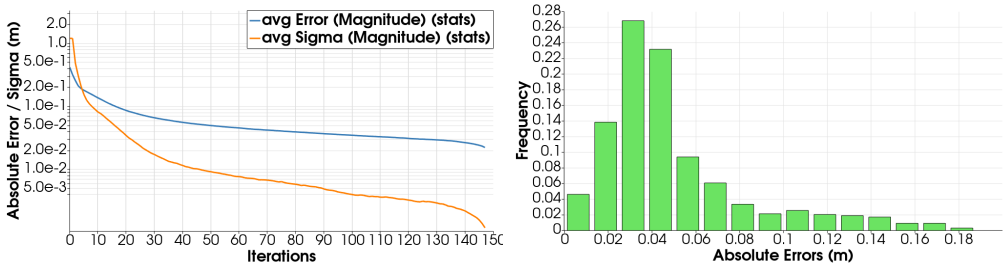
(a) Initialization

(b) Iteration 20

(c) Iteration 80

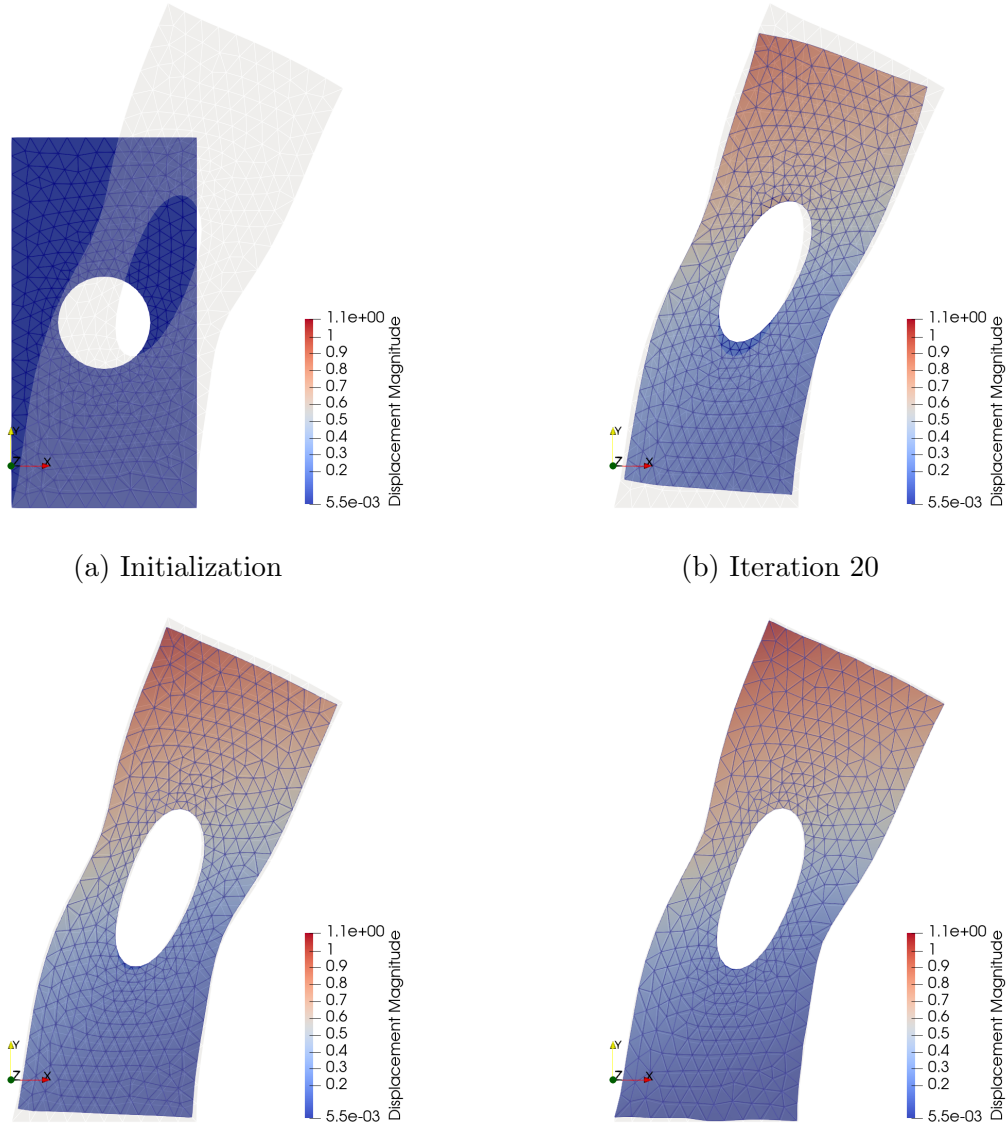
(d) Final result (iteration 147)

Figure 5.14: The displacement recovery process for case 1 in example 1.



(a) (b)

Figure 5.15: (a) The history of average nodal absolute error and standard deviation for case 1 in example 1; (b) The distribution of nodal absolute errors for the final result (iteration 147) for case 1 in example 1.



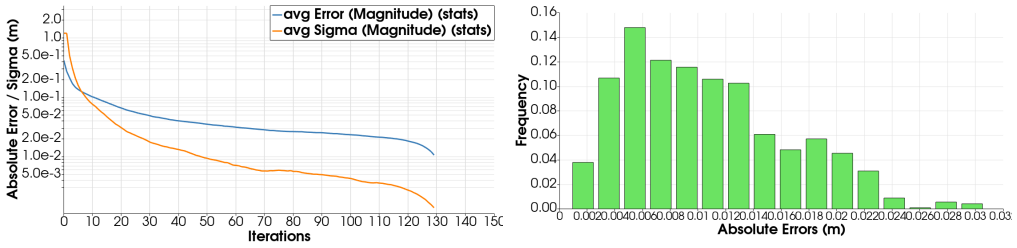
(a) Initialization

(b) Iteration 20

(c) Iteration 80

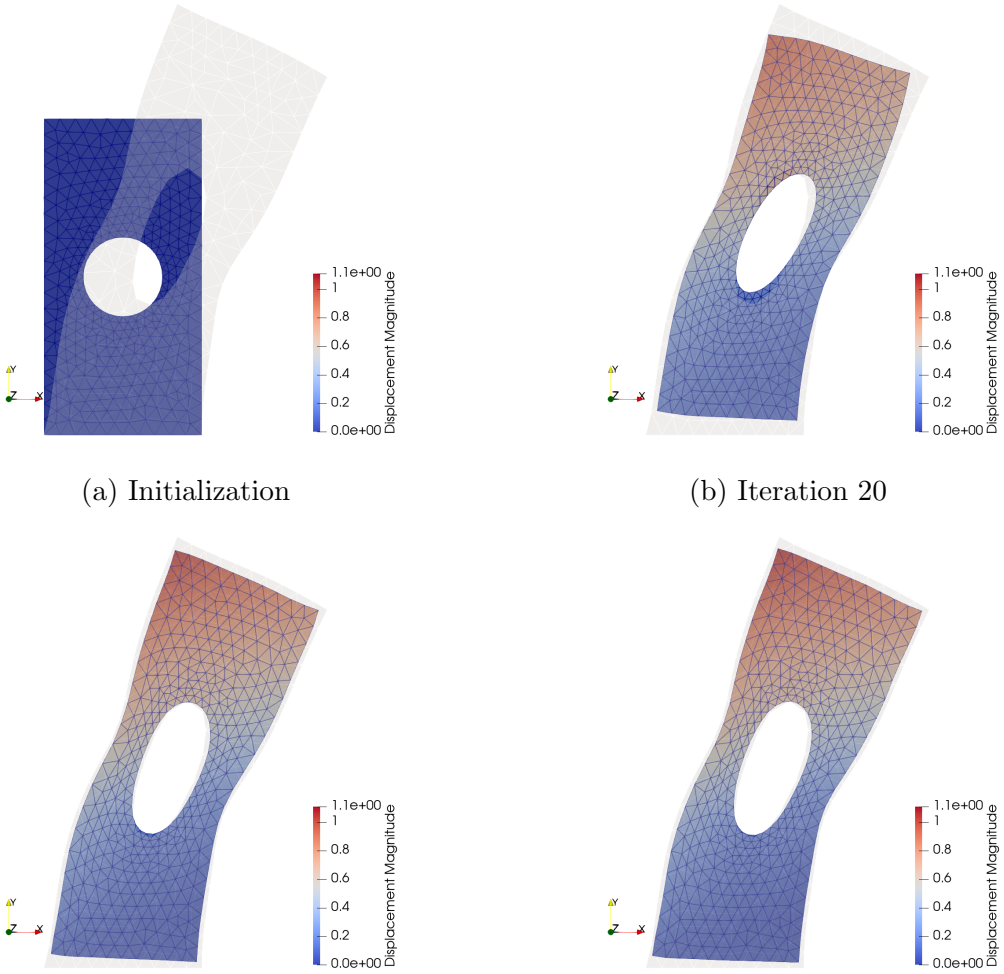
(d) Final result (iteration 129)

Figure 5.16: The displacement recovery process for case 2 in example 1.



(a) (b)

Figure 5.17: (a) The history of average nodal absolute error and standard deviation for case 2 in example 1; (b) The distribution of nodal absolute errors for the final result (iteration 147) for case 2 in example 1.



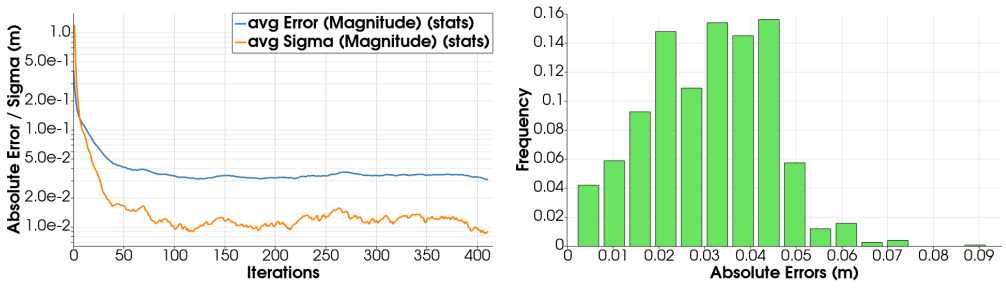
(a) Initialization

(b) Iteration 20

(c) Iteration 80

(d) Final result (iteration 410)

Figure 5.18: The displacement recovery process for case 3 in example 1.



(a) (b)

Figure 5.19: (a) The history of average nodal absolute error and standard deviation for case 3 in example 1; (b) The distribution of nodal absolute errors for the final result (iteration 147) for case 3 in example 1.

Table 5.6: The comparison of displacement recovery accuracy between three cases.

Recovery accuracy	**C**ase 1	**C**ase 2	**C**ase 3
Average nodal absolute error [m]	0.022	0.011	0.031
Displacement recovery rate [%]	94.6	97.3	92.4

Ductile fracture of a thin plate

To further investigate the properties of the entropic variational theory, we consider an example of ductile fracture of a thin plate which was utilized in [84, 49]. The edge lengths of the plate are $L = 4\ mm$ and $l = 2\ mm$. The initial crack is in the middle of the thin plate with size $d = 0.5\ mm$ (see Fig. 5.20). This thin plate is under uniaxial tension condition with velocity boundary conditions, $V = 10\ m/s$, prescribed at both ends of the plate.

For this example, the Gurson-Tvergaard-Needleman(GTN) constitutive model [34] is taken into account with parameters:

$$\begin{aligned} E &= 211\text{GPa} \\ \rho &= 7800\ \text{kg m}^3 \\ v &= 0.3 \\ \sigma_Y &= 469\text{MPa} \\ q_1 &= 1.5, \quad q_2 = 1.0, \quad q_3 = 2.25, \quad f_0 = 0.0025, \\ f_N &= 0.02, \quad f_c = 0.15, \quad f_f = 0.25; \\ \epsilon_N &= 0.3, \quad s_N = 0.1. \end{aligned} \tag{5.5}$$

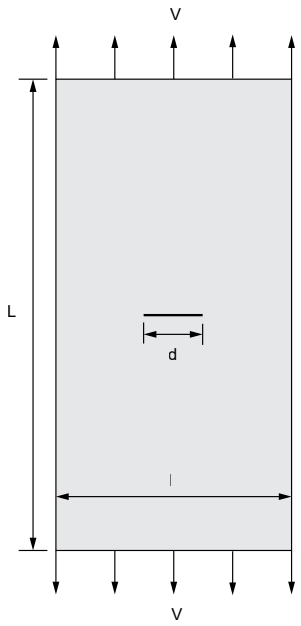
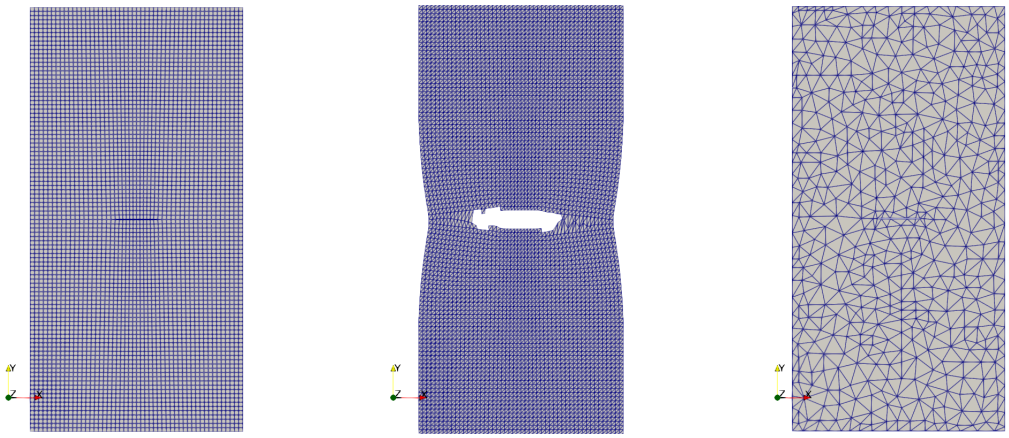


Figure 5.20: The setup of the ductile fracture of a thin plate.



(a) (b) (c)

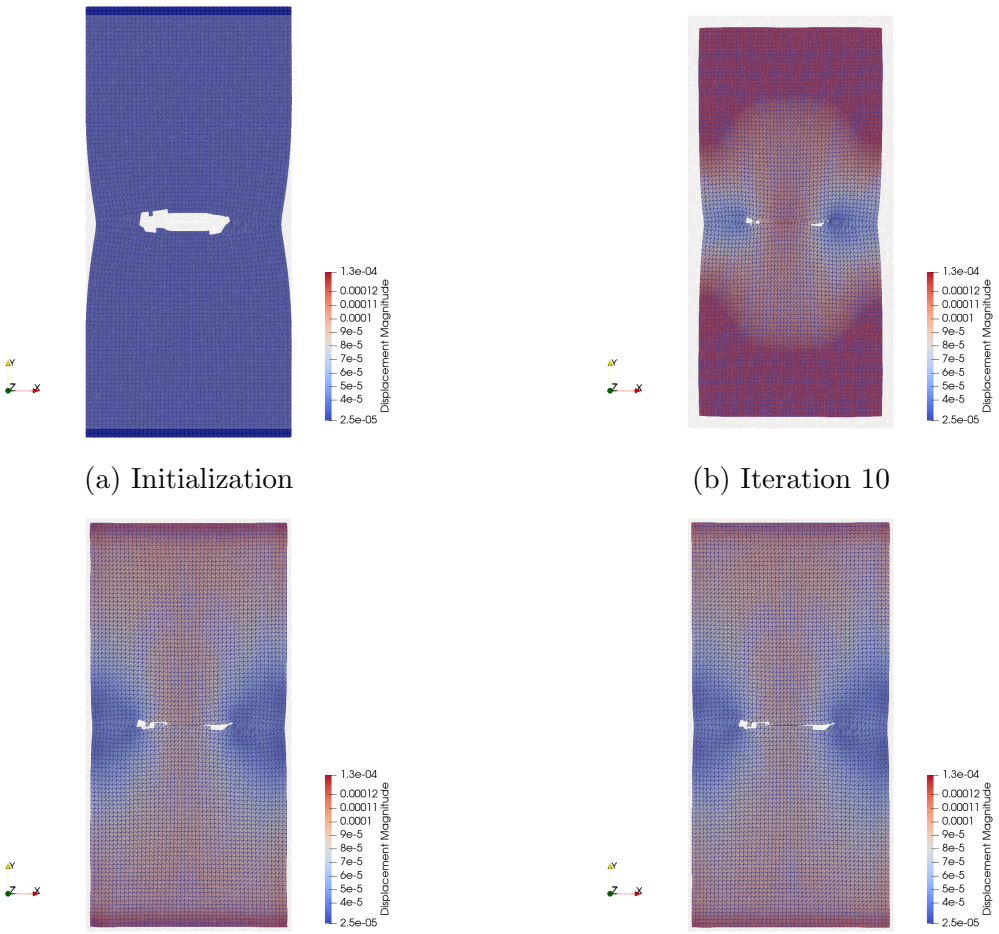
Figure 5.21: The finite element meshes ductile fracture of a thin plate: (a) Mesh in the initial configuration; (b) The same mesh in the final configuration; (c) Coarse mesh in the initial configuration.

The mesh used in the numerical simulation is shown in Fig. 5.21(a) which contains $5,000$

quadrilateral elements. The result of the simulation with crack propagation is shown in Fig. 5.21(a). Unlike the example in section 5.2, for this two-dimensional ductile fracture problem, the mesh in current configuration is set as the GMM centroids. And a coarse version of the mesh in initial configuration with 999 linear triangular elements (see Fig. 5.21(c)) is chosen as the data points. Therefore, during the recovery process, the deformed mesh shown in Fig. 5.21(b) moves to align the coarse initial mesh in Fig. 5.21(c). Furthermore, in contrast to the numerical simulation which use the GTN model, a linear elastic model is taken into account in the VBL-FEM. The values of Lamé contants and other model parameters are shown in Table 5.7.

Table 5.7: The VBL-FEM model parameters for example 2.

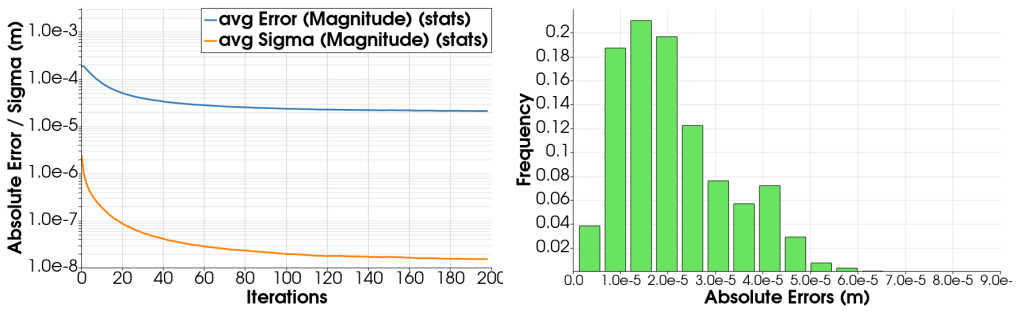
Model parameters	**V**alue
Lamé constant λ [MPa]	0.001
Lamé constant μ [MPa]	0.001
Hyperparameter β	4×10^{-4}
Hyperparameter λ	1×10^{-5}
Max number of iterations	200



(a) Initialization (b) Iteration 10

(c) Iteration 100 (d) Final result (iteration 200)

Figure 5.22: The displacement recovery process for the ductile fracture of a plate.



(a) (b)

Figure 5.23: (a) The history of average nodal absolute error and standard deviation for the ductile fracture of a plate; (b) The distribution of nodal absolute errors for the final result (iteration 200) for the ductile fracture of a plate.

Fig. 5.22 shows the displacement recovery process for the ductile fracture of a plate. Similarly, The coarse mesh in the reference configuration is shown in gray as the true inverse deformation mapping. The current configuration which is color-coded by the magnitude of displacement moves back to the reference configuration along the matching process. As it can be seen from Fig. 5.23, the iteration process stops at 200 steps of iterations. The majority of inverse displacement mapping is recovered at $60th$ iteration. And the error is mainly introduced by the alignment of the crack tip. At the end of the recovery process, the average nodal inverse displacement error which is calculated by Eq. (5.4) is decreased from 0.18 mm to 0.019 mm. That is to say, 89.4% of the total inverse displacement is recovered by the VBL-FEM. It can be clearly seen that the proposed VBL-FEM is capable of the recovery of both the forward and inverse deformation mapping and can be applied to both elastic and plastic problem with fracture propagation in two dimension.

Thin shell cylinder subjected to internal pressure

In this numerical example, the VBL-FEM is applied to recover the displacement field of a thin shell cylinder subjected to internal pressure. The dimension of this example is shown in Fig. 5.24. The GTN constitutive mode is chosen. This thin shell cylinder is subjected to internal pressure which is 10 MPa. Further details of numerical simulation can be found in [75].

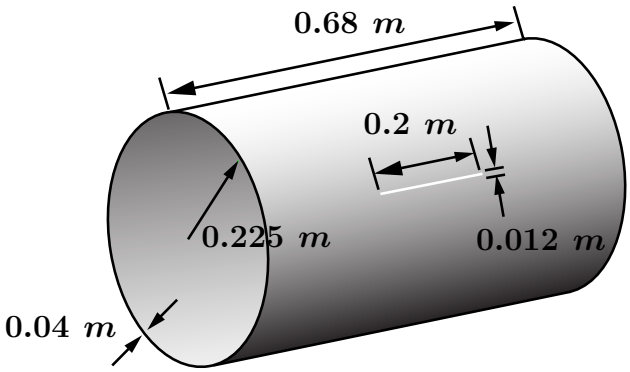
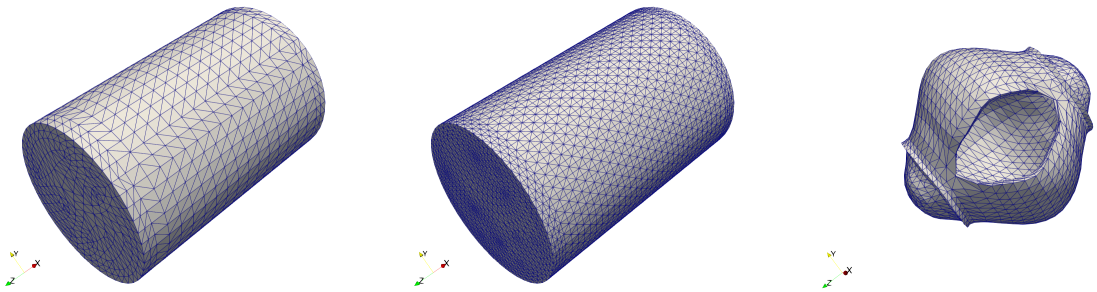


Figure 5.24: The setup of the thin shell cylinder.

The FE model for this thin shell cylinder is shown in Fig. 5.25(a) with 13,848 tetrahedron elements. Then a dense mesh is generated on the initial configuration with 110,784 tetrahedron elements which is shown in 5.25(b). And the result of the simulation with crack propagation is shown in Fig. 5.25(c). For this example, the mesh in current configuration is set as the GMM centroids. And a coarse version of the dense mesh in initial configuration is chosen as the data points. Therefore, during the recovery process, the deformed mesh shown moves to align the dense undeformed mesh.



(a) (b) (c)

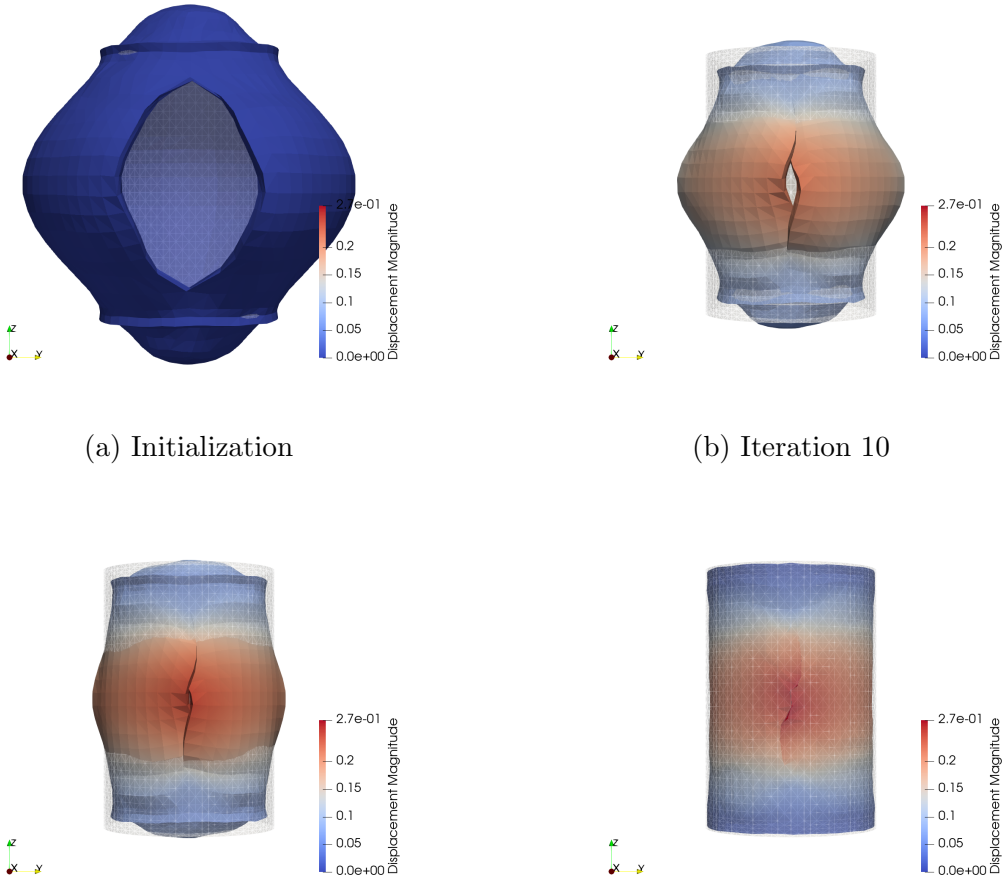
Figure 5.25: The finite element meshes for the thin shell cylinders: (a) Mesh in the initial configuration; (b) A finer mesh in the initial configuration; (c) Mesh in the final configuration.

Furthermore, in contrast to the numerical simulation which use the GTN model, a linear elastic model is still taken into account in the VBL-FEM. The values of Lamé contants and other model parameters are shown in Table 5.8.

Table 5.8: The VBL-FEM model parameters the thin shell cylinder example.

Model parameters	**V**alue
Lamé constant λ [MPa]	0.001
Lamé constant μ [MPa]	0.001
Hyperparameter β	4×10^{-4}
Hyperparameter λ	1×10^{-5}
Max number of iterations	50

Fig. 5.26 shows the displacement recovery process for the thin shell cylinder. Similarly, The dense mesh in the reference configuration is shown in gray as the true inverse deformation mapping. The current configuration which is color-coded by the magnitude of displacement moves back to the reference configuration along the matching process. As it can be seen from Fig. 5.27, the iteration process stops at 30 steps of iterations. At the end of the recovery process, the average nodal inverse displacement error which is calculated by Eq. (5.4) is decreased from 0.11 m to 0.0058 m. That is to say, 94.7% of the total inverse displacement is recovered by the VBL-FEM. It can be clearly seen that the proposed VBL-FEM is capable of the recovery of three dimensional problem with fracture propagation.



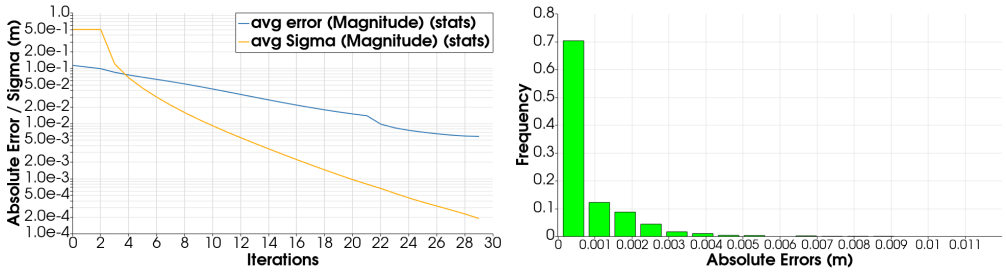
(a) Initialization

(b) Iteration 10

(c) Iteration 15

(d) Final result (iteration 30)

Figure 5.26: The displacement recovery process for the thin shell cylinder.



(a) (b)

Figure 5.27: (a) The history of average nodal absolute error and standard deviation for the thin shell cylinder; (b) The distribution of nodal absolute errors for the final result.

Chapter 6

Conclusions and Outlook

In this dissertation, the MDF and VBL-FEM algorithms are proposed. We first develop an MDF algorithm and apply this MDF algorithm to calculate the thermal compensation allowance in 3D printing product design. A double cantilever beam is used in the verification process for the MDF algorithm. In this data-driven methodology, the only information that is required is the spatial locations of the material points, also known as the scan positions in the current configuration. Whereas the information about the matching material coordinates in the reference configuration is not required. By completely exploiting the information included in the STL file, a unique weight is assigned to each feature point based on the areas of triangle components, causing the spatial material points to converge back to their original or material coordinates through maximum probability optimization. Finally, it is possible to determine the displacements as well as the deformation of each and every material point that is included in the scanned data.

We have proved via quantitative analyses that the MDF method presented here is accurate and robust. This innovative AI-based continuous manifold registration solution might pave the way for the elimination of heat distortion in additive manufacturing and 3D printing.

Moreover, a mixed variational Bayesian learning finite element method has been developed, which provides a machine intelligence numerical solution for the inverse problem of recovering continuum mappings only based on the initial undeformed and final deformed configurations of continuum solids and structures that can contain defects and strong discontinuities. Comparing to the non-rigid registration algorithms in computer vision, the proposed VBL-FEM approach is an inverse solution for three-dimensional solids, whereas the current GMM-based registration algorithms are only for matching the surface or domain boundary points. Moreover, the VBL-FEM inverse solution does not require the data point sets in the initial and final configuration have one-to-one correspondence.

It is demonstrated that the VBL-FEM can provide practically useful inverse solutions with good accuracy, and this inverse finite element solution procedure can be applied to a host of engineering problems, such as structural forensic analysis, digital image correlation, predictive modeling of geometric deviations in 3D printing, AI-aided diagnosis of medical imaged, and among many others.

Of course, there are a few aspects that need further investigation. The first one is the finite deformation version of VBL-FEM. One can easily extend the proposed VBL-FEM formulation to the nonlinear FEM [93] which has the capacity to model large deformation accurately and is robust to rotation. By incorporating finite deformation theory, one can have better results for the distortion field with large deformation and rotation. In addition, the proposed VBL-FEM can be extended for image registration. Several statistical and FEM-based models [56, 43, 27, 14, 46] are proposed recently. Following the same approach, the image can be represented by control points and then further modeled by a unity finite element mesh. By applying the VBL-FEM, the pixels in the target image shall move along the unity finite element mesh to match the pixels in template one.

Bibliography

[1] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3 (1992), pp. 175–185.

[2] CJS Alves and T. Ha-Duong. "Inverse scattering for elastic plane cracks". In: *Inverse Problems* 15.1 (1999), p. 91.

[3] R. Anderson et al. "MFEM: A Modular Finite Element Methods Library". In: *Computers & Mathematics with Applications* 81 (2021), pp. 42–74.

[4] PJ Arrazola et al. "Recent advances in modelling of metal machining processes". In: *CIRP Annals* 62.2 (2013), pp. 695–718.

[5] Xingwang Bai, Haiou Zhang, and Guilan Wang. "Improving prediction accuracy of thermal analysis for weld-based additive manufacturing by calibrating input parameters using IR imaging". In: *The International Journal of Advanced Manufacturing Technology* 69.5 (2013), pp. 1087–1095.

[6] P. Ballard and A. Constantinescu. "On the inversion of subsurface residual stresses from surface stress measurements". In: *Journal of the Mechanics and Physics of Solids* 42.11 (1994), pp. 1767–1787.

[7] Roshdy S Barsoum. "On the use of isoparametric finite elements in linear fracture mechanics". In: *International journal for numerical methods in engineering* 10.1 (1976), pp. 25–37.

[8] Klaus-Jürgen Bathe. *Computational fluid and solid mechanics*. Elsevier, 2001.

[9] Joseph Beaman, Dave Bourell, and Darrell Wallace. "Additive manufacturing (AM) and 3D printing". In: *Journal of Manufacturing Science and Engineering* 136.6 (2014).

[10] Ted Belytschko et al. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2014.

[11] M.A. Bessa et al. "A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality". In: *Computer Methods in Applied Mechanics and Engineering* 320 (2017), pp. 633–667.

[12] M Biegler, B Graf, and Michael Rethmeier. "In-situ distortions in LMD additive manufacturing walls can be measured with digital image correlation and predicted using numerical simulations". In: *Additive Manufacturing* 20 (2018), pp. 101–110.

[13] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[14] Alexandre Bône et al. "Deformetrica 4: an open-source software for statistical shape analysis". In: *International Workshop on Shape in Medical Imaging*. Springer. 2018, pp. 3–13.

[15] Marc Bonnet and Andrei Constantinescu. "Inverse problems in elasticity". In: *Inverse problems* 21.2 (2005), R1.

[16] Huy Duong Bui. *Fracture mechanics: inverse problems and solutions*. Vol. 139. Springer Science & Business Media, 2007.

[17] Thomas Campbell et al. "Could 3D printing change the world". In: *Technologies, Potential, and Implications of Additive Manufacturing, Atlantic Council, Washington, DC* 3 (2011).

[18] George Casella and Roger L Berger. *Statistical inference*. Vol. 2. Duxbury Pacific Grove, CA, 2002.

[19] G. Chen et al. "Application of deep learning neural network to identify collision load conditions based on permanent plastic deformation of shell structures". In: *Computational Mechanics* 64.2 (2019), pp. 435–449.

[20] Q. Chen et al. "A deep neural network inverse solution to recover pre-crash impact data of car collisions". In: *Transportation research part C: emerging technologies* 126 (2021), p. 103009.

[21] L. Cheng, F. Tsung, and A. Wang. "A statistical transfer learning perspective for modeling shape deviations in additive manufacturing". In: *IEEE Robotics and Automation Letters* 2.4 (2017), pp. 1988–1993.

[22] KG Jaya Christiyan, U Chandrasekhar, and K Venkateswarlu. "A study on the influence of process parameters on the Mechanical Properties of 3D printed ABS composite". In: *IOP conference series: materials science and engineering*. Vol. 114. 1. IOP Publishing. 2016, p. 012109.

[23] Morris H DeGroot. *Probability and statistics*. 04; QA273, D4 1986. 1986.

[24] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[25] Sahibsingh A Dudani. "The distance-weighted k-nearest-neighbor rule". In: *IEEE Transactions on Systems, Man, and Cybernetics* 4 (1976), pp. 325–327.

[26] Alexander J Dunbar et al. "Experimental validation of finite element modeling for laser powder bed fusion deformation". In: *Additive Manufacturing* 12 (2016), pp. 108–120.

[27] Stanley Durrleman et al. "Morphometry of anatomical shape complexes with dense deformations and sparse parameters". In: *NeuroImage* 101 (2014), pp. 35–49.

[28] Raquel de Souza Borges Ferreira, Arman Sabbaghi, and Qiang Huang. "Automated geometric shape deviation modeling for additive manufacturing systems via Bayesian neural networks". In: *IEEE Transactions on Automation Science and Engineering* 17.2 (2019), pp. 584–598.

[29] Jack Francis and Linkan Bian. "Deep learning for distortion prediction in laser-based additive manufacturing using big data". In: *Manufacturing Letters* 20 (2019), pp. 10–14.

[30] William E Frazier. "Metal additive manufacturing: a review". In: *Journal of Materials Engineering and performance* 23.6 (2014), pp. 1917–1928.

[31] H. Gao, M. J. Zahr, and J.-X. Wang. "Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems". In: *Computer Methods in Applied Mechanics and Engineering* 390 (2022), p. 114502.

[32] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities". In: *International journal for numerical methods in engineering* 79.11 (2009), pp. 1309–1331.

[33] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[34] Arthur L Gurson. *Continuum theory of ductile rupture by void nucleation and growth: Part I Yield criteria and flow rules for porous ductile media*. Tech. rep. Technical Report No. 39. Brown University, 1977.

[35] B. B. Guzina and M. Bonnet. "Topological derivative for the inverse scattering of elastic waves". In: *Quarterly Journal of Mechanics and Applied Mathematics* 57.2 (2004), pp. 161–179.

[36] T. Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[37] JC Heigel, P Michaleris, and TA Palmer. "In situ monitoring and characterization of distortion during laser cladding of Inconel® 625". In: *Journal of Materials Processing Technology* 220 (2015), pp. 135–145.

[38] Dirk Herzog et al. "Additive manufacturing of metals". In: *Acta Materialia* 117 (2016), pp. 371–392.

[39] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[40] Qiang Huang. "An analytical foundation for optimal compensation of three-dimensional shape deformation in additive manufacturing". In: *Journal of Manufacturing Science and Engineering* 138.6 (2016).

[41] Qiang Huang et al. "Optimal offline compensation of shape shrinkage for three-dimensional printing processes". In: *Iie transactions* 47.5 (2015), pp. 431–441.

[42] TJR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[43] Yue Jia, Yongjie Zhang, and Timon Rabczuk. "A novel dynamic multilevel technique for image registration". In: *Computers & Mathematics with Applications* 69.9 (2015), pp. 909–925.

[44] Chandrika Kamath, Juliette Franzman, and Ravi Ponmalai. "Data mining for faster, interpretable solutions to inverse problems: A case study using additive manufacturing". In: *Machine Learning with Applications* 6 (2021), p. 100122.

[45] Steven M Kay. *Modern spectral estimation: theory and application*. Pearson Education India, 1988.

[46] Juelin Leng, Guoliang Xu, and Yongjie Zhang. "Medical image interpolation based on multi-resolution registration". In: *Computers & Mathematics with Applications* 66.1 (2013), pp. 1–18.

[47] Jingran Li, Ran Jin, and Z Yu Hang. "Integration of physically-based and data-driven approaches for thermal field prediction in additive manufacturing". In: *Materials & Design* 139 (2018), pp. 473–485.

[48] Shaofan Li and Wing Kam Liu. "Meshfree and particle methods and their applications". In: *Applied Mechanics Reviews* 55.1 (2002), pp. 1–34.

[49] Shaofan Li and Cerup B Simonsen. "Meshfree simulations of ductile crack propagations". In: *International journal for computational methods in engineering science and mechanics* 6.1 (2005), pp. 1–19.

[50] W. Li, M. Z. Bazant, and J. Zhu. "A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches". In: *Computer Methods in Applied Mechanics and Engineering* 383 (2021), p. 113933.

[51] Xuan Liang et al. "A modified method for estimating inherent strains from detailed process simulation for fast residual distortion prediction of single-walled structures fabricated by directed energy deposition". In: *Additive Manufacturing* 23 (2018), pp. 471–486.

[52] Gui-Rong Liu and Moubin B Liu. *Smoothed particle hydrodynamics: a meshfree particle method*. World scientific, 2003.

[53] Weibo Liu et al. "A survey of deep neural network architectures and their applications". In: *Neurocomputing* 234 (2017), pp. 11–26.

[54] Z. Liu, M.A. Bessa, and W.K. Liu. "Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials". In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 319–341.

[55] Erdogan Madenci and Ibrahim Guven. *The finite element method and applications in engineering using ANSYS®*. Springer, 2015.

[56] Sherif Makram-Ebeid and Oudom Somphone. "Non-rigid image registration using a hierarchical partition of unity finite element method". In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007, pp. 1–8.

[57] Richard Martukanitz et al. "Toward an integrated computational system for describing the additive manufacturing process for metallic materials". In: *Additive Manufacturing* 1 (2014), pp. 52–63.

[58] Geoffrey J McLachlan and Suren Rathnayake. "On the number of components in a Gaussian mixture model". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4.5 (2014), pp. 341–355.

[59] R. McNeel. *Rhino-Grasshopper, 2020*. `http://www.rhino3d.com/`.

[60] Ferry PW Melchels, Jan Feijen, and Dirk W Grijpma. "A review on stereolithography and its applications in biomedical engineering". In: *Biomaterials* 31.24 (2010), pp. 6121–6130.

[61] O. A. Mohamed, S. H. Masood, and J. L. Bhowmik. "Optimization of fused deposition modeling process parameters: a review of current research and future prospects". In: *Advances in Manufacturing* 3.1 (2015), pp. 42–53.

[62] Bernhard Mueller. "Additive manufacturing technologies–Rapid prototyping to direct digital manufacturing". In: *Assembly Automation* (2012).

[63] T Mukherjee, W Zhang, and Tarasankar DebRoy. "An improved prediction of residual stresses and distortion in additive manufacturing". In: *Computational Materials Science* 126 (2017), pp. 360–372.

[64] Hidekazu Murakawa, Yu Luo, and Yukio Ueda. "Prediction of welding deformation and residual stress by elastic FEM based on inherent strain". In: *Journal of the society of Naval Architects of Japan* 1996.180 (1996), pp. 739–751.

[65] A. Myronenko, X. Song, and M.A. Carreira-Perpinán. "Non-rigid point set registration: Coherent point drift". In: *Advances in neural information processing systems*. 2007, pp. 1009–1016.

[66] Andriy Myronenko and Xubo Song. "Point set registration: Coherent point drift". In: *IEEE transactions on pattern analysis and machine intelligence* 32.12 (2010), pp. 2262–2275.

[67] In Jae Myung. "Tutorial on maximum likelihood estimation". In: *Journal of mathematical Psychology* 47.1 (2003), pp. 90–100.

[68] V. M. Nguyen-Thanh et al. "Parametric deep energy approach for elasticity accounting for strain gradient effects". In: *Computer Methods in Applied Mechanics and Engineering* 386 (2021), p. 114096.

[69] Bo Ni and Huajian Gao. "A deep learning approach to the inverse problem of modulus identification in elasticity". In: *MRS Bulletin* 46.1 (2021), pp. 19–25.

[70] V. Ocelík, J. Bosgra, and J. T. M. de Hosson. "In-situ strain observation in high power laser cladding". In: *Surface and coatings technology* 203.20-21 (2009), pp. 3189–3196.

[71] Jian-Xin Pan and Kai-Tai Fang. "Maximum likelihood estimation". In: *Growth curve models and statistical diagnostics*. Springer, 2002, pp. 77–158.

[72] Nachiket Patil et al. "A generalized feed forward dynamic adaptive mesh refinement and derefinement finite element framework for metal laser sintering—part I: formulation and algorithm development". In: *Journal of Manufacturing Science and Engineering* 137.4 (2015).

[73] Per-Olof Persson and Gilbert Strang. "A simple mesh generator in MATLAB". In: *SIAM review* 46.2 (2004), pp. 329–345.

[74] A. Porterfield. *Crystallon, 2019*. `http://www.food4rhino.com/app/crystallon`.

[75] Dong Qian et al. "Meshfree simulation of failure modes in thin cylinders subjected to combined loads of internal pressure and localized heat". In: *International journal for numerical methods in engineering* 76.8 (2008), pp. 1159–1184.

[76] M Ramesh, L Rajeshkumar, and D Balaji. "Influence of process parameters on the properties of additively manufactured fiber-reinforced polymer composite materials: a review". In: *Journal of Materials Engineering and Performance* 30.7 (2021), pp. 4792–4807.

[77] Singiresu S Rao. *The finite element method in engineering*. Butterworth-heinemann, 2017.

[78] V. C. RAYKAR and R. DURAISWAMI. "Improved fast Gauss transform with variable source scales". In: ().

[79] Douglas A Reynolds. "Gaussian Mixture Models." In: *Encyclopedia of biometrics* 741 (2009).

[80] E. Samaniego et al. "An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications". In: *Computer Methods in Applied Mechanics and Engineering* 362 (2020), p. 112790.

[81] Babis Schoinochoritis, Dimitrios Chantzis, and Konstantinos Salonitis. "Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review". In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 231.1 (2017), pp. 96–117.

[82] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. "A generalized representer theorem". In: *International conference on computational learning theory*. Springer. 2001, pp. 416–426.

[83] Iñaki Setien et al. "Empirical methodology to determine inherent strains in additive manufacturing". In: *Computers & Mathematics with Applications* 78.7 (2019), pp. 2282–2295.

[84] DC Simkins and S Li. "Meshfree simulations of thermo-mechanical ductile fracture". In: *Computational Mechanics* 38.3 (2006), pp. 235–249.

[85] Márta Szilvśi-Nagy and GY Matyasi. "Analysis of STL files". In: *Mathematical and computer modelling* 38.7-9 (2003), pp. 945–960.

[86] Emilio Turco. "Tools for the numerical solution of inverse problems in structural mechanics: review and research perspectives". In: *European Journal of Environmental and Civil Engineering* 21.5 (2017), pp. 509–554.

[87] Athanasios Voulodimos et al. "Deep learning for computer vision: A brief review". In: *Computational intelligence and neuroscience* 2018 (2018).

[88] Chao Wang et al. "Quantification and compensation of thermal distortion in additive manufacturing: A computational statistics approach". In: *Computer Methods in Applied Mechanics and Engineering* 375 (2021), p. 113611.

[89] WL Wang et al. "Influence of process parameters on stereolithography part shrinkage". In: *Materials & Design* 17.4 (1996), pp. 205–213.

[90] A. B. Weglein et al. "Inverse scattering series and seismic exploration". In: *Inverse problems* 19.6 (2003), R27.

[91] Wikipedia contributors. *Woodbury matrix identity — Wikipedia, The Free Encyclopedia*. [Online; accessed 18-May-2021]. 2021. URL: `https://en.wikipedia.org/w/index.php?title=Woodbury_matrix_identity&oldid=1021909740`.

[92] Kaufui V Wong and Aldo Hernandez. "A review of additive manufacturing". In: *International scholarly research notices* 2012 (2012).

[93] Peter Wriggers. *Nonlinear finite element methods*. Springer Science & Business Media, 2008.

[94] A. S. Wu et al. "An experimental investigation into additive manufacturing-induced residual stresses in 316L stainless steel". In: *Metallurgical and Materials Transactions A* 45.13 (2014), pp. 6260–6270.

[95] W. Yan et al. "Data-driven multi-scale multi-physics models to derive process–structure–property relationships for additive manufacturing". In: *Computational Mechanics* 61(5) (2018), pp. 521–541.

[96] Jie Yin et al. "Simulation of temperature distribution in single metallic powder layer for laser micro-sintering". In: *Computational Materials Science* 53.1 (2012), pp. 333–339.

[97] Zuowei Zhu et al. "Machine learning in tolerancing for additive manufacturing". In: *CIRP Annals* 67.1 (2018), pp. 157–160.

[98] Olek C Zienkiewicz, Robert L Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

Appendix A

The minimizer of the variational principle

The variational formulation in the approximation problem is

$$\mathcal{H}[\mathbf{u}] = \sum_{i=1}^{N} \|\mathbf{u}(\mathbf{X}_i) - \mathbf{X}_i\|^2 + \lambda \mathcal{R}[\mathbf{u}] \tag{A.1}$$

in which, the smoothness function $\mathcal{R}[\mathbf{u}]$ is defined as,

$$\mathcal{R}[\mathbf{u}] = \int_{\mathbb{R}^d} \frac{\|\tilde{\mathbf{u}}(\boldsymbol{s})\|^2}{\tilde{G}(\boldsymbol{s})} d\boldsymbol{s} \tag{A.2}$$

Based on Fourier transform, one is to have,

$$\mathbf{u}(\mathbf{X}) = \int_{\mathbb{R}^d} \tilde{\mathbf{u}}(\boldsymbol{s}) e^{i\mathbf{X}\cdot\boldsymbol{s}} d\boldsymbol{s}$$

substitute back to the variational formulation, one is to have,

$$\mathcal{H}[\tilde{\mathbf{u}}] = \sum_{i=1}^{N} \left\| \mathbf{X}_i - \int_{\mathbb{R}^d} \tilde{\mathbf{u}}(\boldsymbol{s}) \exp(i\mathbf{X}_i \cdot \boldsymbol{s}) d\boldsymbol{s} \right\|^2 + \lambda \int_{\mathbb{R}^d} \frac{\tilde{\mathbf{u}}(-\boldsymbol{s}) \cdot \tilde{\mathbf{u}}(\boldsymbol{s})}{\tilde{G}(\boldsymbol{s})} d\boldsymbol{s} \tag{A.3}$$

where $\tilde{\mathbf{u}}(\boldsymbol{s}) = \tilde{\mathbf{u}}(-\boldsymbol{s})$, because $\mathbf{u}$ is real.

To find the desired minimizer, we take the Gateaux derivative of the expanded functional and set it to zero,

$$\begin{aligned} \delta\mathcal{H} &= \lim_{\tau\to 0} \frac{1}{\tau}\Big(\mathcal{H}(\tilde{\mathbf{u}} + \tau\delta\tilde{\mathbf{u}}) - \mathcal{H}(\tilde{\mathbf{u}})\Big) = \mathcal{H}'(\tilde{\mathbf{u}})\delta\tilde{\mathbf{u}} = 0, \\ &\rightarrow \quad \frac{\delta\mathcal{H}[\tilde{\mathbf{u}}]}{\delta\tilde{\mathbf{u}}(\boldsymbol{t})} := \mathcal{H}'(\tilde{\mathbf{u}}) = \mathbf{0}, \quad \forall \boldsymbol{t} \in \mathbb{R}^d \ . \end{aligned}$$

Thus we have,

$$\frac{\delta \mathcal{H}\left[\tilde{\mathbf{u}}\right]}{\delta \tilde{\mathbf{u}}(\boldsymbol{t})} = \frac{\delta}{\delta \tilde{\mathbf{u}}(\boldsymbol{t})} \sum_{i=1}^{N} \left\| \mathbf{X}_i - \int_{\mathbb{R}^d} \tilde{\mathbf{u}}(\boldsymbol{s}) e^{i\mathbf{X}_i \cdot \boldsymbol{s}} d\boldsymbol{s} \right\|^2 + \frac{\delta}{\delta \tilde{\mathbf{u}}(\boldsymbol{t})} \int_{\mathbb{R}^d} \frac{\tilde{\mathbf{u}}(-\boldsymbol{s}) \cdot \tilde{\mathbf{u}}(\boldsymbol{s})}{\tilde{G}(\boldsymbol{s})} d\boldsymbol{s} \ .$$

After some mathematical manipulations, the first term of the above becomes

$$\begin{aligned}
& \frac{\delta}{\delta \tilde{\mathbf{u}}(\boldsymbol{t})} \sum_{i=1}^{N} \left\| \mathbf{X}_i - \int_{\mathbb{R}^d} \tilde{\mathbf{u}}(\boldsymbol{s}) e^{i\mathbf{X}_i \cdot \boldsymbol{s}} d\boldsymbol{s} \right\|^2 \\
= \ & 2 \sum_{i=1}^{N} \left(\mathbf{X}_i - \int_{\mathbb{R}^d} \tilde{\mathbf{u}}(\boldsymbol{s}) e^{i\mathbf{X}_i \cdot \boldsymbol{s}} d\boldsymbol{s} \right) \cdot \int_{\mathbb{R}^d} \frac{\delta \tilde{\mathbf{u}}(\boldsymbol{s})}{\delta \tilde{\mathbf{u}}(\boldsymbol{t})} e^{i\mathbf{X}_i \cdot \boldsymbol{s}} d\boldsymbol{s} \\
= \ & 2 \sum_{i=1}^{N} \left(\mathbf{X}_i - \mathbf{u}(\mathbf{X}_i) \right) \int_{\mathbb{R}^d} \boldsymbol{\Delta}(\boldsymbol{s} - \boldsymbol{t}) e^{i\mathbf{X}_i \cdot \boldsymbol{s}} d\boldsymbol{s} \\
= \ & 2 \sum_{i=1}^{N} \left(\mathbf{X}_i - \mathbf{u}(\mathbf{X}_i) \right) \cdot e^{i\mathbf{X}_i \cdot \boldsymbol{t}}
\end{aligned}$$

and the second term becomes

$$\begin{aligned}
\frac{\delta}{\delta \tilde{\mathbf{u}}(\boldsymbol{t})} \int_{\mathbb{R}^d} \frac{\tilde{\mathbf{u}}(-\boldsymbol{s}) \cdot \tilde{\mathbf{u}}(\boldsymbol{s})}{\tilde{G}(\boldsymbol{s})} d\boldsymbol{s} &= 2 \int_{\mathbb{R}^d} \frac{\tilde{\mathbf{u}}(-\boldsymbol{s})}{\tilde{G}(\boldsymbol{s})} \frac{\delta \tilde{\mathbf{u}}(\boldsymbol{s})}{\delta \tilde{\mathbf{u}}(\boldsymbol{t})} d\boldsymbol{s} \\
&= 2 \int_{\mathbb{R}^d} \frac{\tilde{\mathbf{u}}(-\boldsymbol{s})}{\tilde{G}(\boldsymbol{s})} \boldsymbol{\Delta}(\boldsymbol{s} - \boldsymbol{t}) d\boldsymbol{s} \\
&= 2 \frac{\tilde{\mathbf{u}}(-\boldsymbol{t})}{\tilde{G}(\boldsymbol{t})}
\end{aligned}$$

where

$$\boldsymbol{\Delta}(\boldsymbol{s} - \boldsymbol{t}) = \begin{bmatrix} \delta_1(\boldsymbol{s} - \boldsymbol{t}) & \cdots & \cdots & \cdots \\ \cdots & \delta_2(\boldsymbol{s} - \boldsymbol{t}) & \cdots & \cdots \\ \cdots & \cdots & \ddots & \cdots \\ \cdots & \cdots & \cdots & \delta_d(\boldsymbol{s} - \boldsymbol{t}) \end{bmatrix}_{d \times d}$$

and

$$\delta_i(\boldsymbol{s} - \boldsymbol{t}) = \frac{\delta \tilde{\Psi}_i(\boldsymbol{s})}{\delta \tilde{\Psi}_i(\boldsymbol{t})} = \begin{cases} 1, & \boldsymbol{s} = \boldsymbol{t} \\ 0, & \boldsymbol{s} \neq \boldsymbol{t} \end{cases}$$

are the Kronecker deltas.

Therefore, substituting the two expansions back, we can have,

$$\sum_{i=1}^{N} \left(\mathbf{X}_i - \mathbf{u}(\mathbf{X}_i) \right) e^{i\mathbf{X}_i \cdot \boldsymbol{t}} + \lambda \frac{\tilde{\mathbf{u}}(-\boldsymbol{t})}{\tilde{G}(\boldsymbol{t})} = \mathbf{0} \tag{A.4}$$

after exchanging $\boldsymbol{t} \leftrightarrow -\boldsymbol{t}$,

$$\Rightarrow \sum_{i=1}^{N} \left(\mathbf{X}_i - \mathbf{u}(\mathbf{X}_i)\right) e^{i\mathbf{X}_i\cdot(-\boldsymbol{t})} + \lambda \frac{\tilde{\mathbf{u}}(\boldsymbol{t})}{\tilde{G}(-\boldsymbol{t})} = \mathbf{0}$$

and multiplying $\tilde{G}(-\boldsymbol{t})$,

$$\Rightarrow \tilde{G}(-\boldsymbol{t}) \sum_{i=1}^{N} \left(\mathbf{X}_i - \mathbf{u}(\mathbf{X}_i)\right) e^{i\mathbf{X}_i\cdot(-\boldsymbol{t})} + \lambda \tilde{\mathbf{u}}(\boldsymbol{t}) = \mathbf{0}$$

finally

$$\tilde{\mathbf{u}}(\boldsymbol{t}) = \tilde{G}(-\boldsymbol{t}) \sum_{i=1}^{N} \frac{\mathbf{X}_i - \mathbf{u}(\mathbf{X}_i)}{\lambda} e^{i\mathbf{X}_i\cdot(-\boldsymbol{t})} = \tilde{G}(-\boldsymbol{t}) \sum_{i=1}^{N} \boldsymbol{v}_i e^{i\mathbf{X}_i\cdot(-\boldsymbol{t})}$$

where $\boldsymbol{v}_i = \frac{\mathbf{X}_i - \mathbf{u}(\mathbf{X}_i)}{\lambda}$

Taking the inverse Fourier transformation, we find the final minimizer of the variational problem,

$$\begin{aligned}
\mathbf{u}(\mathbf{X}) &= \int_{\mathbb{R}^d} \tilde{G}(-\boldsymbol{t}) \sum_{i=1}^{N} \boldsymbol{v}_i e^{i\mathbf{X}_i\cdot(-\boldsymbol{t})} \cdot e^{i\boldsymbol{t}\cdot\mathbf{X}} d\boldsymbol{t} \\
&= \sum_{i=1}^{N} \boldsymbol{v}_i \int_{\mathbb{R}^d} \tilde{G}(-\boldsymbol{t}) e^{i(\mathbf{X}-\mathbf{X}_i)\cdot\boldsymbol{t}} d\boldsymbol{t} \\
&= \sum_{i=1}^{N} \boldsymbol{v}_i \int_{\mathbb{R}^d} \tilde{G}(\boldsymbol{t}) e^{i(\mathbf{X}-\mathbf{X}_i)\cdot\boldsymbol{t}} d\boldsymbol{t} \\
&= \sum_{i=1}^{N} \boldsymbol{v}_i G(\mathbf{X} - \mathbf{X}_i) \qquad \text{(A.5)}
\end{aligned}$$

where $\tilde{G}(-\boldsymbol{t}) = \tilde{G}(\boldsymbol{t})$ due to symmetry.

Appendix B

Derivative of $\mathcal{L}_{QR}\left(\mathbf{V}, \sigma^2\right)$ with respect to V

Taking derivative of $\mathcal{L}_{QR}\left(\mathbf{V}, \sigma^2\right)$ with respect to the matrix $\boldsymbol{C}$, one has,

$$\frac{\partial \mathcal{L}_{QR}\left(\mathbf{V}, \sigma^2\right)}{\partial \boldsymbol{V}} = \frac{\partial}{\partial \boldsymbol{C}} \sum_{n=1}^{N} \sum_{m=1}^{M} p^{(t-1)}\left(\Delta_m | \boldsymbol{X}_n\right) \frac{\left\| \boldsymbol{X}_n - \sum_{k=1}^{M} \boldsymbol{v}_k G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0k}\right) \right\|^2}{2\sigma_m^2} + \frac{\lambda}{2} \frac{\partial}{\partial \boldsymbol{V}} \mathrm{Tr}\left(\boldsymbol{V}^T \boldsymbol{G} \boldsymbol{V}\right)$$

It can be split into two terms, one is to consider the derivative with respect to $\boldsymbol{v}_l$ first, i.e.

$$\begin{aligned}
& \frac{\partial}{\partial \boldsymbol{v}_l} \sum_{n=1}^{N} \sum_{m=1}^{M} p^{(t-1)}\left(\Delta_m | \boldsymbol{X}_n\right) \frac{\left\| \boldsymbol{X}_n - \sum_{k=1}^{M} \boldsymbol{v}_k G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0k}\right) \right\|^2}{2\sigma_m^2} \\
= & \sum_{n=1}^{N} \sum_{m=1}^{M} p^{(t-1)}\left(\Delta_m | \boldsymbol{X}_n\right) \frac{2\left(\boldsymbol{X}_n - \sum_{k=1}^{M} \boldsymbol{v}_k G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0k}\right)\right)}{2\sigma_m^2} \cdot \left(-G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0l}\right)\right) \\
= & \sum_{n=1}^{N} \sum_{m=1}^{M} p^{(t-1)}\left(\Delta_m | \boldsymbol{X}_n\right) \frac{-\boldsymbol{X}_n + \sum_{k=1}^{M} \boldsymbol{v}_k G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0k}\right)}{\sigma_m^2} G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0l}\right) \\
= & \frac{1}{\sigma^2} \sum_{m=1}^{M} \left(\sum_{n=1}^{N} \frac{p^{(t-1)}\left(\Delta_m | \boldsymbol{X}_n\right)}{\tilde{w}_m} \right) \sum_{k=1}^{M} \boldsymbol{v}_k G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0k}\right) \cdot G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0l}\right) \\
& - \frac{1}{\sigma^2} \sum_{m=1}^{M} \frac{\sum_{n=1}^{n} p^{(t-1)}\left(\Delta_m | \boldsymbol{X}_n\right) \boldsymbol{X}_n}{\tilde{w}_m} G\left(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0l}\right) \\
= & \frac{1}{\sigma^2} \sum_{m=1}^{M} G\left(\boldsymbol{y}_{0l} - \boldsymbol{y}_{0m}\right) diag\left(\tilde{\boldsymbol{P}} \mathbf{1}\right) \boldsymbol{G}\left(m, :\right) \boldsymbol{V} - \frac{1}{\sigma^2} \sum_{m=1}^{M} G\left(\boldsymbol{y}_{0l} - \boldsymbol{y}_{0m}\right) \left(\tilde{\boldsymbol{P}} \boldsymbol{X}\right) \\
= & \frac{1}{\sigma^2} \boldsymbol{G}\left(l, :\right) diag\left(\tilde{\boldsymbol{P}} \mathbf{1}\right) \boldsymbol{G} \boldsymbol{V} - \frac{1}{\sigma^2} \boldsymbol{G}\left(l, :\right) \left(\tilde{\boldsymbol{P}} \boldsymbol{X}\right)
\end{aligned}$$

so that the matrix form of the first term is obtained,

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{V}} \sum_{n=1}^{N} \sum_{m=1}^{M} p^{(t-1)}(\Delta_m | \boldsymbol{X}_n) \frac{\left\| \boldsymbol{X}_n - \sum_{k=1}^{M} \boldsymbol{v}_k G(\boldsymbol{y}_{0m} - \boldsymbol{y}_{0k}) \right\|^2}{2\sigma_m^2} \\
&= \frac{1}{\sigma^2} \left[\boldsymbol{G} diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right) \boldsymbol{G}\boldsymbol{V} - \boldsymbol{G}\left(\tilde{\boldsymbol{P}}\boldsymbol{X}\right) \right] \qquad \text{(B.1)}
\end{aligned}$$

The second term is to consider indices form,

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{V}} \mathrm{Tr}\left(\boldsymbol{V}^T \boldsymbol{G}\boldsymbol{V}\right) &= \frac{\partial}{\partial V_{pq}} (V_{ij} G_{ik} V_{kj}) = \delta_{ip}\delta_{jq} G_{ik} V_{kj} + V_{ij} G_{ik} \delta_{kp}\delta_{jq} \\
&= G_{pk} V_{kq} + V_{iq} G_{ip} = \boldsymbol{G}\boldsymbol{V} + \boldsymbol{V}^T \boldsymbol{G} = 2\boldsymbol{G}\boldsymbol{V} \ . \qquad \text{(B.2)}
\end{aligned}$$

Summarily, we can have

$$\frac{\partial \mathcal{L}_{QR}(\mathbf{V}, \sigma^2)}{\partial \boldsymbol{V}} = \boldsymbol{G} diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right) \boldsymbol{G}\boldsymbol{V} - \boldsymbol{G}\left(\tilde{\boldsymbol{P}}\boldsymbol{X}\right) + \sigma^2 \lambda \boldsymbol{G}\boldsymbol{V} = \mathbf{0}$$

so that the system for $\boldsymbol{V}$ will be

$$\left[diag\left(\tilde{\boldsymbol{P}}\mathbf{1}\right) \boldsymbol{G} + \sigma^2 \lambda \mathbf{1} \right] \boldsymbol{V} = \tilde{\boldsymbol{P}}\boldsymbol{X} \ . \qquad \text{(B.3)}$$