Elastic Registration of Histological Serial Sections

A Finite Element Approach

by

J. T. Coppoolse

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday August 27, 2019 at 2:00 PM.

Student number:	4149815	
Project duration:	June, 2018 - August, 2019	
Thesis committee:	Dr. ir. M. B. van Gijzen,	TU Delft, Supervisor
	Prof. dr. rer. nat. J. Modersitzki,	University of Lübeck
	Dr. J. L. A. Dubbeldam,	TU Delft
	Dr. ir. R. F. Remis,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Summary

For a proper three-dimensional reconstruction of histology serial sections, adjustment of the slices is necessary for combining serial sections. Deformations occur due to the sectioning and acquisition process of the microscopic analysis of histology. By reconstructing the deformations with a transformation of the sections, a mathematical correction on the images can be applied. By using image registration, a transformation function is searched for to minimize differences between the histology slices. Due to the non-linearity of the distortions, prior knowledge is required in order to have a solvable problem. Additional information in the form of elasticity regularisation is considered. Implementing the elastic regularisation with a finite element method, provides a continuous transformation function With the continuous function, in a natural way, alignment can be monitored for folding transformations. In this work, the (bi-)linear and (bi-)quadratic elements for the finite element method are implemented and compared with the finite difference method. It is observed that for the different kinds of elements, the (bi-)linear elements yield best results with the validity of the transformation. Moreover, the computational costs for the bi-linear elements are the cheapest. Compared with the finite difference method, the differences in accuracy are not noteworthy but the computational time of the finite element method is longer. Furthermore, to steer the matching in an accurate direction, improvements are proposed by applying local stiffness of the elements or adding soft constraints on the volume of the elements. This results in significant improvements in the transformation. For these two approaches it is observed that local stiffness is more restrictive than volume-preserving. Solving the optimization problem, a Gauss-Newton method to search for descent directions is applied. A matrix-based and a matrix-free approach of elasticity regularisation is considered in the linear system of finding a descent direction. While the matrix-free approach decreases the memory usage, the computational costs are significantly increased.

Contents

1 Introduction

The project[1]: Validating MRI-based in-vivo histology, of Dr. Siawoosh Mohammadi at UKE in Hamburg, aims to develop an MRI-based method to measure microstructures of the brain. When highly sensitive and specific markers of the microstructures are obtained, disease-related changes in the structures can be tracked. Therefore, different high-resolution MRI techniques are combined. Previously, measuring microstructures was only possible on dead tissue by ex-vivo histology. For development of these models, a quantitative comparison is with ex-vivo histology.

Ex-vivo histology is an accurate model of tissue microstructures. For histology, the tissue is sliced into thin sections. After a preparation process with microscopic examination, two-dimensional information is obtained. For comparison with the three-dimensional MRI-based method, reconstruction of the sections is required. Unfortunately, due to the sectioning and the processing before the analysis, non-linear deformations occur in the sections. Therefore, a three-dimensional analysis and view of the obtained images is impossible without the reconstruction of the distortion. Therefore, solving the problem of the distortions, the technique of image registration is applied. Image registration searches for a transformation in which the differences between two images are minimal. However, for the non-linear deformations, additional information is required in order to solve the optimization problem. A commonly method is by adding elastic regularisation to the registration problem. One of the processes in the histology technique is that the tissue is embedded in paraffin before sectioning, therefore elastic deformations are reasonable to be expected. It has already been shown that elasticy equations could be used for registration of histological sections [28][2][6].

However, having a solvable minimization problem it does not necessarily result in a reasonable transformation of the image. There is no unique solution for which the images are perfectly aligned. Validity of the transformation is thus not necessarily guaranteed. Mapping with an obtained transformation function could lead to a folded transformed image. Such transformed images are physically impossible, and can not be applied. To validate the obtained transformation, it is suggested to apply the finite element method as numerical approach for the elasticity regularisation in the minimization problem. One of the benefits of the finite element method, is the property of having continuous transformation functions. Therefore, the obtained transformation function of the minimization problem can be accurately validated. Furthermore, (soft) constraints can be applied to the elements to steer the minimization in the right direction.

1.1. Research Goal

The main goal of this report is an implementation of the finite element method for elastic regularisation in registration problems such that the transformation can be accurately validated. At which the following three questions are proposed:

[1] https://gepris.dfg.de/gepris/projekt/313644856

- Does the finite element method improve the accuracy of the transformation function?
- How can the validity of the transformation function be improved using a finite element implementation of elasticity regularisation?
- How can a matrix-free approach of the finite element elasticity regularisation be achieved?

1.2. Methodology

The finite element method is implemented in the toolbox Flexible Algorithms in Image Registration (FAIR)[2]. FAIR is an academic and research-orientated tool that collects state-of-the-art implementation of the different building blocks to solve the registration problem [27]. The package is written for MATLAB [3]. The toolbox features a finite difference method for the elastic regularisation and will be used for comparison.

The different approaches are first applied to test data. These data sets are test cases of the registration toolbox FAIR. After that, the methods are applied to the histological serial section data set, from the project of Dr. Siawoosh Mohammadi. Dr. Stefan Geyer, Dr. Harald Moeller and Dr. Ricciardo Metere provided a data set from a monkey brain which consists of 555 histology sections.

1.3. Outline

This report structured as follows: In Chapter 2 an introduction is given in image registration. Implementing the elastic potential energy as a regularizer, the physical background of linear elasticity is explained in Chapter 3. After that, the mathematical framework of image registration in FAIR is discussed in Chapter 4. The finite element method and the proposed approaches of elastic regularisation are explained in Chapter 5. For the linear solution methods of the optimization problem, Chapter 6 treats different methods. In Chapter 7, the results of the finite element method and the finite difference method are compared. Further, in Chapter 8, the results are shown for the proposed improvements with the finite element method. Applying the methods to the histological serial sections, Chapter 9 discusses the approach and the results of this registration problem. At last, in Chapter 10 conclusions are made and a discussion is given.

[2] https://github.com/C4IR/FAIR.m
[3] https://nl.mathworks.com

2

Image Registration

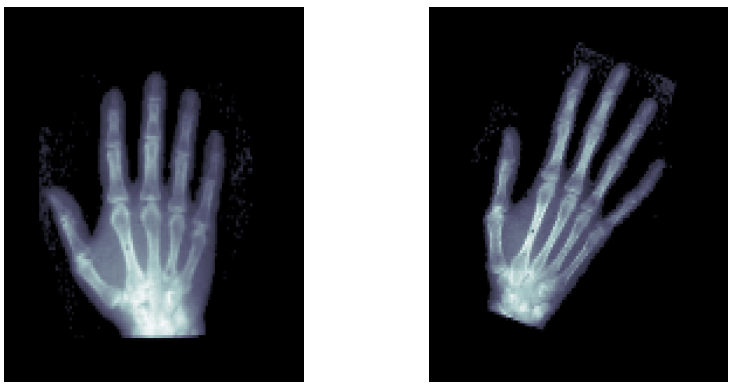
In this chapter, an introduction of Image Registration is given. At first, the general problem of Image Registration is formulated. In the second section, different registration techniques are presented. Afterwards, measures are explained for measuring the similarities between images. Distance measures are treated, and solutions are addressed for steering the solutions and solving registration problems that are not solvable without extra prior information.

2.1. General Problem

In image processing, one is often interested in combining or comparing multiple images. Unfortunately, due to the changes in image acquisition conditions, the task can be very hard. For example, with two different x-rays of a human hand, see Figure 2.1, spatial alignment is needed before it can be compared. Differences due to the acquisition conditions can appear in the geometric and/or intensity part of the images. The procedure of alignment is called Image Registration. In the procedure, a suitable transformation function is search for such that the image to transform (template image) can be matched to the target (reference image) [5][25][26].
Let $\mathcal{R}(\mathbf{x}), \mathcal{T}(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ denote the intensity values of the reference and template image at the corresponding coordinates. The mapping can be expressed as

$$\mathcal{R}(\mathbf{x}) = I(\mathcal{T}(y(\mathbf{x}))) \tag{2.1}$$

with $y(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $I : \mathbb{R} \rightarrow \mathbb{R}$. Where $\mathbf{y}$ is the spatial transformation and I the intensity transformation [5].



(a) Reference image (b) Template image

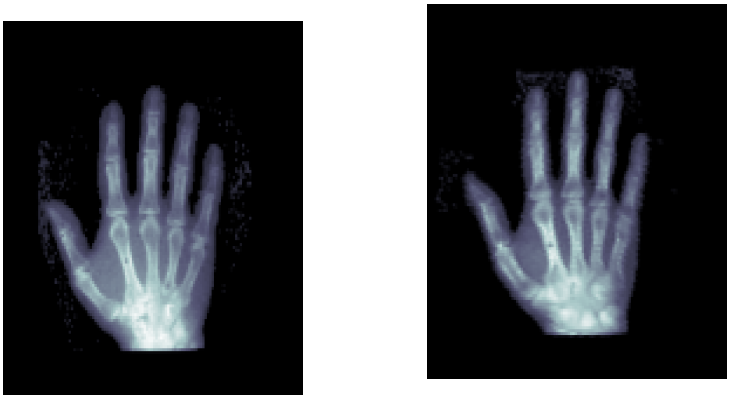
Figure 2.1: X-rays images of a human hand, taken from [1]

A perfect mapping does not only correct differences in acquisition conditions but also the variations in the data. A perfect mapping isn't the goal as we are interested in the variations of the data. To find these variations between images, geometrical mapping is seen as key in image registration [5]. It is reasonable not to consider the intensity transformation as the intensity differences are beside acquisition conditions also data related. These data related variations are caused by movements, growths, or changes in relative depths in the object of the images.

Searching for the spatial transformation function, the problem is written as a minimization of a cost function. By measuring the differences between the images, a transformation has to be found such that the differences are minimal

$$\min_{\mathbf{y}} \mathcal{J}[\mathcal{T}(\mathbf{y}(\mathbf{x})), \mathcal{R}(\mathbf{x})] \tag{2.2}$$

where $\mathcal{J} : \mathbb{R} \to \mathbb{R}$ is the measure [27]. Recall the example of the human hands, Figure 2.2 shows a suitable spatial transformation function in which both images are aligned.



(a) Reference image, $\mathcal{R}(\mathbf{x})$

(b) Transformed template image, $\mathcal{T}(\mathbf{y}(\mathbf{x}))$

Figure 2.2: Reference image and transformed template image obtained via toolbox *FAIR*

2.2. Registration Techniques

Image registration techniques are divided into two parts. An optimization where the parameters are computed and an optimization procedure where the parameters are searched for [25].

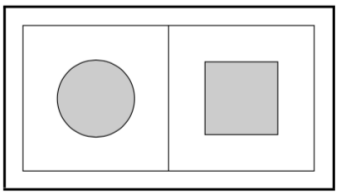
The first approach is based on a linear combination of basis functions. This technique uses a finite set of parameters to find a suitable transformation. With the transformation, the whole image can be mapped such that the template and reference image are aligned. It is an optimization procedure where the parameters are computed before the transformation. This approach is also seen in literature as parametric image registration [26].

For the second approach, the transformation is not based on any pre-defined basis function. The parameters for every location are explicitly determined from the available data using the measure. For the minimization of the measure $\mathcal{J}$, the parameters are directly searched for, and is seen as non-parametric registration [26].

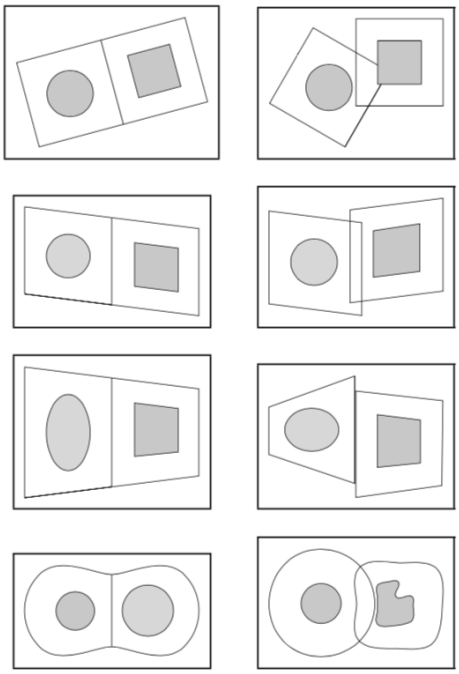
Parametric Transformation

Parametric Image Registration uses a transformation function with a finite set of parameters. The function is a linear combination of the parameters with corresponding basis functions [27]. It can be written as

$$\mathbf{y}(\mathbf{x}) = Q(\mathbf{x})\mathbf{w} \tag{2.3}$$



(a) Original



(b) Left: Global; Right: Local; Top to bottom: Rigid, Affine, Projective, curvature

Figure 2.3: Examples of Parametric transformation, taken from [25]

where $Q \in \mathbb{R}^{(d \cdot l) \times d}$ is the matrix of l basis functions multiplied by the parameter vector $\mathbf{w} \in \mathbb{R}^{d \cdot l}$[27]. Basis functions are artificial and chosen prior. Linear affine and rigid transformations are common to use and based on linear basis functions [5],[27],[31].

As stated in the Survey of The Image registration Techniques of Brown [5], the parametric transformation in literature is seen globally and locally. With the global approach, the full images are taken into account for finding the right parameters. Applying the transformations on subsections of the image is called local transformations. Local transformations are seen in two different ways. Local parameters are applied as global parameters, which saves computational time when the displacement of the image is the same on the whole domain. The other approach is by using different local transformations for every subsection, but it is rarely applied because of continuity and invertibility issues [5],[25]. In Figure 2.3, a few examples of the parametric image transformations globally and locally are shown. The downside of this kind of transformation is that the domain is transformed based on a finite set of parameters. Therefore, accuracy depends on the number of parameters and basis functions. Besides that, the basis functions are artificial and have to be justified [26].

Many registration techniques need a good initial position before taking local differences into account, a global mapping. Therefore, applying a transformation like affine or rigid as pre-registration before considering the local differences are seen [25] [27].

2.3. Measure

The parametric as the non-parametric registration needs a measure, a cost-function, to find the transformation function. In the parametric case searching for optimal parameters and in the non-parametric case for obtaining the transformation. The measure quantifies the differences in the data. The problem is written as [26].

$$\min_{\mathbf{y}} \mathcal{J} = \min_{\mathbf{y}} \mathcal{D}[\mathcal{T}(\mathbf{y}), \mathcal{R}]. \tag{2.4}$$

where $\mathcal{D} : \mathbb{R} \to \mathbb{R}$ is a distance measure and $\mathbf{y} = \mathbf{y}(\mathbf{x})$ with template and reference image, $\mathcal{T}(\mathbf{y}) = \mathcal{T}(\mathbf{y}(\mathbf{x}))$ and $\mathcal{R} = \mathcal{R}(\mathbf{x})$ respectively.

Solving the minimization can be very hard. Using only a distance measure often leads to an ill-posed problem, Definition 2.1, for which no solution exists and can heavily depend on noisy data. The problem has to be modified with additional information to make it solvable. Regularization terms with prior knowledge are added to the measure [10] [25]. Applying these terms also filters optimal transformations which are more likely. To further improve the measure, a penalty term can be added to the measure to penalize unwanted solutions.

$$\mathcal{J}[\mathbf{y}] = \mathcal{D}[\mathcal{T}(\mathbf{y}), \mathcal{R}] + \alpha\mathcal{S}[\mathbf{y}] + \beta\mathcal{P}[\mathbf{y}] \tag{2.5}$$

Where $\alpha, \beta \in \mathbb{R}$ and $\mathcal{S}, \mathcal{P} : \mathbb{R} \to \mathbb{R}$.

Definition 2.1: Hadamard

A problem is well-posed if

- It has a solution,
- The solution is unique,
- The solution depends continuously on data and parameters.

Problems which are not well-posed are called ill-posed.

Distance Measure

Different distance measures are seen in literature. Choosing the distance measure depends on different properties as computational time, accuracy, and what kind of information is used. Categorized by Maintz et al. [25], the measures are divided into extrinsic and intrinsic based measures.
For the extrinsic based measures, it relies on artificial objects (e.g., markers, frames) that are visible and accurately detectable. These methods are easy, fast, and often, the parameters can be computed explicitly and thus, no complicated optimization techniques are needed. Drawbacks are that in the pre-imaging phase, provisions must be made. Also, the registration doesn't take patient information into account, and in most cases, it is restricted to only rigid transformations.
The intrinsic based methods rely on patient-generated information. The intrinsic methods can be divided into landmarks which are based on salient points, segmentation-based using structures, and voxel-property-based methods. The methods based on landmarks take only part of the image into account, while the intensity-based(voxel-property-based)methods can use all information. The intensity-based methods nowadays form the basis for the majority of the registration approaches [33].

Landmark and intensity-based distance measures are briefly explained, based on [27] [26]. These two distance measures show the differences between less information and fast computation compared to a distance measure which uses all information but will be more complicated to solve.

Landmark-based distance measure

With the usage of salient points, landmarks, which are specified in template and reference image, a landmark-based measure can be applied. Landmarks are seen both extrinsic as intrinsic. Say, we have m landmarks corresponding to the locations $\mathbf{r}^i, \mathbf{t}^i$ for $i = 1, ...m$, a commonly used feature distance measure is

$$\mathcal{D}^{LM}[y] = \sum ||\mathbf{y}(\mathbf{r}^i) - \mathbf{t}^i||^2_{\mathbb{R}^d} \tag{2.6}$$

The benefit of the landmark-based measure is the computational cost. The downside is that landmarks need to be added or must be seen clearly. Furthermore, only a part of the information of the images is used for a suitable transformation.

Intensity-Based distance measure
By using the intensity values of the voxels, the full images are taken into account for accurate, suitable mapping and are thus expensive. In general, the intensity-based distance measures can be written as

$$\mathcal{D}[\mathcal{T}(\mathbf{y}), \mathcal{R}] = \int_{\Omega} \varphi(\mathcal{T}(\mathbf{y}), \mathcal{R}) d\Omega \tag{2.7}$$

Where $\Omega \subset \mathbb{R}^d$ and $\varphi : \mathbb{R} \to \mathbb{R}$ is the function calculating the difference between reference and template image. For the difference function, a comparison of the intensity values or intensity changes is applied.

One of the intensity-based methods is the Sum-of-Squared-Distance (SSD) measure. The measure uses the assumption that a perfect transformation results in $\mathcal{R} = \mathcal{T}(\mathbf{y})$. The SSD distance measure is given by

$$\mathcal{D}[\mathcal{T}(\mathbf{y}), \mathcal{R}] = \frac{1}{2} \int_{\Omega} \left(\mathcal{T}(\mathbf{y}) - \mathcal{R}\right)^2 d\Omega \tag{2.8}$$

and has been proven to be robust and effective for mono-modal images [10].

For multi-modality images, the intensity values for a perfect alignment doesn't correspond, and the SSD distance measure isn't suitable. Commonly used alternatives are [27],[17]: The Normalised-Cross-Correlation (NCC) distance measure, in which only linear scaling in intensity is possible. Furthermore, the Mutual-Information (MI) distance measure based on the joint density of the images is often seen. Also, the Normalised-Gradient-Field (NGF) distance measure in which it is assumed that intensity changes for both images should occur at corresponding locations is standard.

Regularisation and Penalty

For some data measures, a unique solution is impossible while having only intensity values. For example, the intensity-based distance measure for non-parametric registration is ill-posed. Only an intensity value is known for every location $\mathbf{x} \in \mathbb{R}^d$ while a vector $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^d$ is needed. Additional information in the measure resolves this issue and makes the ill-posed problem solvable.

Physical driven information can be added, but also mathematical based information like diffusive and curvature regularisation is common to apply [10] [27]. For the physically driven regularisation, the elasticity of tissue or fluid mechanics such as balances of mass and momentum are seen [26]. Using the elasticity of tissue, it measures the elastic potential energy caused by the deformation of the material [4]. For fluid mechanics, besides elasticity also viscosity characteristics caused by the deformation are measured [26]. The diffusive regularisation smoothens the solution and uses the first order derivatives of the displacement [8]. The curvature regularisation is based on the second order derivatives and aims that the registration is less dependent on the initial position of the reference and template image [9].

Having solvable problems due to the regularisation, it doesn't necessarily give suitable solutions. Solutions could be physically unlikely or don't correspond with user information about the data. Adding a penalty function to the measure discourages some optimal solutions. The function penalizes the unwanted solutions and steers to the user direction. For example, knowing that specific points should correspond, a landmark data measure for these points can be added as penalty term [21]. Furthermore, volume-preserving and local rigidity penalties are seen [10]. These are soft constraints on the volume and are for example applicable when (parts of the) tissue may not expand or shrink.

3

Linear Elasticity

Implementing an elasticity regularizer in Image Registration was first introduced by Broit [4] and is often seen in Image Registration. This chapter explains the physical background of linear elasticity. By using the stress and strain relations of the body, a minimization problem of elastic potential energy is obtained. This chapter is based on [15], [24], [26].

3.1. Stress and Strain

Image registration aims to compute the displacements of the template image to align with the reference image. By considering images as elastic bodies, we can add physically inspired prior knowledge. Without looking at the physical properties of the body, the transformation could result in folding or cracks inside the body. Adding a regularisation term that describes the change in length or shape due to the elasticity of the body, adds prior information to have a solvable problem. Furthermore, the folding and cracks of the body become less likely. The changes in the length and shape are described by stress and strain acting on the body.

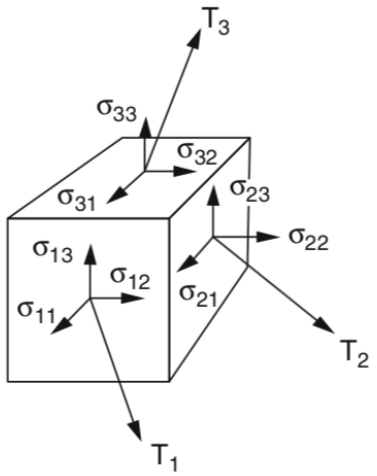


Figure 3.1: Stress tensors on a body, taken from [15]

The changes in length or shape are called strains (ϵ). These deformations lead to internal forces, *stress* (σ), due to stretching and compressing of the atomic bonds. From Newton's first law, the forces result in acceleration and therefore influences the deformation.

Strain and stress tensors in the three-dimensional case consist of nine components and can be divided

into normal and shear components. The normal components of strain and stress act perpendicular to the acting surface. The shear components are parallel to the surface. Figure 3.1 shows the stress components acting on a three-dimensional body. σ_{ii} are the normal components, and σ_{ij} for $i \neq j$ are the shear components parallel to the surface. $\mathbf{T} = \sigma \cdot \mathbf{n}$ is the traction vector with positive normal vector n.

3.2. Equations of Equilibrium

The equilibrium equations governing the stress in the elastic body are the linear and angular momentum equations. These set of equations describes the relations between the external force on the body and the stress it causes at the surface of the body. Considering only static problems, the linear momentum is defined by

$$\int_V \mathbf{f} dV + \int_A \mathbf{T} dA = 0 \tag{3.1}$$

with traction vector $\mathbf{T}$ and external force $\mathbf{f}$ on body V and body surface area A.
Using Gauss Theorem, which relates the volume integral of the divergence with the net flow across volume's boundary, it read

$$\int_V \mathbf{f} + \nabla \cdot \sigma dV = 0$$

Defining the gradient by $\nabla = [\frac{\partial}{\partial x_1}, \cdots, \frac{\partial}{\partial x_n}]^T$.
Furthermore, it has to satisfy for every sub-body and thus results in the following condition

$$\mathbf{f} = -\begin{bmatrix} \nabla \cdot \sigma_{1:} \\ \nabla \cdot \sigma_{2:} \\ \nabla \cdot \sigma_{3:} \end{bmatrix} \tag{3.2}$$

The second equilibrium equation is the angular momentum. It is the rotational equivalent of the linear momentum and read

$$\int_\Omega \mathbf{x} \times \mathbf{f} dV + \int_{\partial\Omega} \mathbf{x} \times \mathbf{T} dA = \mathbf{0} \tag{3.3}$$

where $\mathbf{x}$ is the position vector, and $\times$ is the vector cross-product. Writing out Equation 3.3, we get

$$\int_V x_i f_j - x_j f_i dV + \int_A (x_i \sigma_{j:} - x_j \sigma_{i:}) \cdot \mathbf{n} dA = 0 \qquad \text{for } (i,j) = (2,3), (3,1), (1,2);$$

Using Gauss Theorem and Integration By Parts, it results in

$$\int_V x_i f_j - x_j f_i + \nabla \cdot (x_i \sigma_{j:} - x_j \sigma_{i:}) dV = 0$$

$$\int_V x_i (f_j + \nabla \cdot \sigma_{j:}) - x_j (f_i + \nabla \cdot \sigma_{i:}) + \sigma_{ji} - \sigma_{ij} dV = 0$$

Inserting the condition of linear momentum, Equation 3.2, into the equation, we have

$$\int_V \sigma_{ji} - \sigma_{ij} dV = 0$$

It has to satisfy for every sub-body, which results in the integrand to be zero. This gives us the symmetry condition for the stress components.

$$\sigma_{ji} = \sigma_{ij} \qquad \text{for } i,j = 1,2,3 \tag{3.4}$$

3.3. Three-Dimensional Elasticity

Assuming only small strains occur in the body, the relation between stress and strain can be described by Hooke's law and gives a linear stress-strain relation

$$\sigma_{ij} = \sum_{k=1}^{3}\sum_{l=1}^{3} C_{ijkl}\epsilon_{kl} \tag{3.5}$$

with tensor C. By the symmetry condition, Equation 3.4, the linear model is reduced into six independent strain and stress components.

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{13}\\ \sigma_{23}\end{bmatrix} = \begin{bmatrix}a_{11} & \cdots & a_{16}\\ \vdots & \ddots & \vdots\\ a_{61} & \cdots & a_{66}\end{bmatrix}\begin{bmatrix}\epsilon_{11}\\ \epsilon_{22}\\ \epsilon_{33}\\ \epsilon_{12}\\ \epsilon_{13}\\ \epsilon_{23}\end{bmatrix} \tag{3.6}$$

Introducing Young's modulus, when a small force is applied in one direction, the strain in that direction is related by

$$\epsilon_{11} = \frac{1}{E}\sigma_{11} \tag{3.7}$$

where E is Young's modulus for elasticity.
Besides the strain in the direction of the stress, in general, the body also shows displacements in the other normal directions. For simplification isotropic material is assumed. With an isotropic body, the displacement is the same for every other direction. The displacement is related via Poisson's contraction ratio ν

$$\epsilon_{22} = \epsilon_{33} = -\nu\epsilon_{11} \tag{3.8}$$

Resulting in the following equations for the normal strain

$$\epsilon_{ii} = \frac{1+\nu}{E}\sigma_{ii} - \frac{\nu}{E}\sum_{j=1}^{3}\sigma_{jj} \tag{3.9}$$

Shear strain and shear stress are related by shears modulus for isotropic body and is given by

$$\sigma_{ij} = 2\mu\epsilon_{ij} \text{ , for } i \neq j. \tag{3.10}$$

The parameter μ is called the lamé second parameter and can be determined in terms of Young's modulus and Poisson's contraction ratio

$$\mu = \frac{E}{2(1+\nu)} \tag{3.11}$$

The system of stress-strain equations is

$$\begin{bmatrix}\epsilon_{11}\\ \epsilon_{22}\\ \epsilon_{33}\\ \epsilon_{12}\\ \epsilon_{13}\\ \epsilon_{23}\end{bmatrix} = \begin{bmatrix}1/E & -\nu/E & -\nu/E & 0 & 0 & 0\\ -\nu/E & 1/E & -\nu/E & 0 & 0 & 0\\ -\nu/E & -\nu/E & 1/E & 0 & 0 & 0\\ 0 & 0 & 0 & 1/2\mu & 0 & 0\\ 0 & 0 & 0 & 0 & 1/2\mu & 0\\ 0 & 0 & 0 & 0 & 0 & 1/2\mu\end{bmatrix}\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{13}\\ \sigma_{23}\end{bmatrix} \tag{3.12}$$

Introducing lamé's first parameter. The parameter λ isn't a direct physical parameter but is useful to simplify the notation.

$$\lambda := \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{3.13}$$

Calculating the inverse of the stress-strain relation, the system is given by

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{13}\\ \sigma_{23}\end{bmatrix} = \begin{bmatrix}\lambda+2\mu & \lambda & \lambda & 0 & 0 & 0\\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0\\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0\\ 0 & 0 & 0 & 2\mu & 0 & 0\\ 0 & 0 & 0 & 0 & 2\mu & 0\\ 0 & 0 & 0 & 0 & 0 & 2\mu\end{bmatrix}\begin{bmatrix}\epsilon_{11}\\ \epsilon_{22}\\ \epsilon_{33}\\ \epsilon_{12}\\ \epsilon_{13}\\ \epsilon_{23}\end{bmatrix} \tag{3.14}$$

Strains are a measure of the deformation of the body and can also be written in terms of displacements. With the assumption of small deformations, the infinitesimal strain theory is applied

$$\epsilon = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) \tag{3.15}$$

and results in the following strain matrix

$$\epsilon = \begin{bmatrix}\frac{\partial u_1}{\partial x_1} & \frac{1}{2}(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}) & \frac{1}{2}(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1})\\ \frac{1}{2}(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2})\\ \frac{1}{2}(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}) & \frac{1}{2}(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}) & \frac{\partial u_3}{\partial x_3}\end{bmatrix} \tag{3.16}$$

Rewriting the stress-strain relations of Equation 3.14 in terms of stress and the displacement function, the equations become

$$\sigma_{ij} = \mu\Big(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\Big) + \lambda(\nabla\cdot\mathbf{u})\delta_{ij} \tag{3.17}$$

with

$$\delta_{ij} = \begin{cases}1 \text{ for } i = j\\ 0 \text{ for } i \neq j\end{cases}$$

Applying the stress-displacement equations to the equilibrium condition of linear momentum, Equation 3.2, it results in

$$\nabla\cdot\sigma_{i:} + f_i = 0 \tag{3.18}$$

$$\mu\Big(\frac{\partial^2 u_i}{(\partial x_1)^2} + \frac{\partial^2 u_i}{(\partial x_3)^2} + \frac{\partial^2 u_i}{(\partial x_3)^2}\Big) + (\lambda+\mu)\frac{\partial}{\partial x_i}\Big(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}\Big) + f_i = 0 \quad \text{for } i = 1,2,3 \tag{3.19}$$

Which we can compactly write as

$$\mu\Delta\mathbf{u} + (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) + \mathbf{f} = \mathbf{0} \tag{3.20}$$

with vector Laplacian $\Delta = \nabla^2$. The equations are an equilibrium between external force and displacement and have to satisfy for every linear elastic body with small deformations. The equations are called the Navier-Lamé equations with corresponding Lamé parameters:

$$\mu = \frac{1}{2}\frac{E}{1+\nu}, \qquad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{3.21}$$

Plane strain

For a two-dimensional image, the body is considered to be a thin plate. Only strain in two dimensions occur, and the strain components of the third dimension are set to zero. Because of shears modulus, the shear stress components in the third dimension are zero.
Implementing in the Navier-lamé equations, result in the disappearance of the third equation. The Navier-Lamé equations with $\mathbf{u} = \begin{bmatrix}u_1, u_2\end{bmatrix}^T$ and lamé parameters are thus

$$\mu\Delta\mathbf{u} + (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) + \mathbf{f} = 0 \tag{3.22}$$

$$\mu = \frac{1}{2}\frac{E}{1+\nu}, \quad \lambda = \frac{E\nu}{(1+\nu)(1-\nu)} \tag{3.23}$$

3.4. Minimization Problem

The Navier-Lamé equations are partial differential equations based on conservation laws, and knowing the boundary conditions of the domain, it can also be written in terms of a minimization problem. The corresponding minimization of the Navier-Lamé equations is the elastic potential energy caused by the deformation of the body.

For the minimization problem, we consider that no external forces are exerted on the body, and no external stress occurs on the boundary. Which follows that $\mathbf{f}$ disappears, and no external stress gives traction free boundary conditions. With the stress-displacement relation of Equation 3.17, the traction free boundary conditions can be expressed as

$$\sigma_{i:} \cdot \mathbf{n} = \mu\Big(\nabla u_i + \frac{\partial \mathbf{u}}{\partial x_i}\Big) \cdot \mathbf{n} + \lambda(\nabla \cdot \mathbf{u}) n_i = 0 \qquad \mathbf{x} \in \partial\Omega \text{ for i=1,2} \tag{3.24}$$

The partial differential equation results in the following boundary value problem with domain Ω and boundary $\partial\Omega$

$$\begin{cases} -\mu\Delta\mathbf{u} - (\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) = 0, & \mathbf{x} \in \Omega \\ \mu\Big(\nabla u_i + \frac{\partial \mathbf{u}}{\partial x_i}\Big) \cdot \mathbf{n} + \lambda(\nabla \cdot \mathbf{u}) n_i = 0 & \mathbf{x} \in \partial\Omega \end{cases} \tag{3.25}$$

With corresponding minimization problem

$$\min_{\mathbf{u}\in\Sigma} \frac{1}{2} \int_\Omega \lambda(\nabla \cdot \mathbf{u})^2 + \mu\Big(2(\frac{\partial u_1}{\partial x_1})^2 + 2(\frac{\partial u_2}{\partial x_2})^2 + (\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1})^2\Big) d\Omega \tag{3.26}$$
$$\Sigma(\mathbf{u}) := \{\mathbf{u} \text{ is sufficiently smooth}\}$$

Proof is shown in appendix A.

4

Mathematical Framework of FAIR

This chapter discusses the framework of the toolbox Flexible Algorithms for Image Registration[1] (FAIR). It features a multi-level strategy in obtaining a transformation function. With a discretize-then-optimize approach, the registration problem is solved. For the discretization, a finite difference scheme is applied as numerical approach. In the first section, the multi-level approach with pre-registration is treated. Secondly, the discretization of the problem is explained. After that, the iterative optimization procedure for the non-linear problem is discussed. Lastly, issues are addressed using the finite difference scheme to motivate the usage of a finite element method in FAIR. The chapter is based on [27].

Recall from chapter 2, Equation 2.5, the measure is

$$\mathcal{J}[\mathrm{y}] = \mathcal{D}[\mathcal{T}(\mathrm{y}), \mathcal{R}] + \alpha\mathcal{S}[\mathrm{y}] + \beta\mathcal{P}[\mathrm{y}]. \tag{4.1}$$

Considering two-dimensional reference and template images of the same rectangular regions. The regions are defined by $\Omega = [\omega_1, \omega_2] \times [\omega_3, \omega_4] \subset \mathbb{R}^2$, as shown in Figure 4.1.

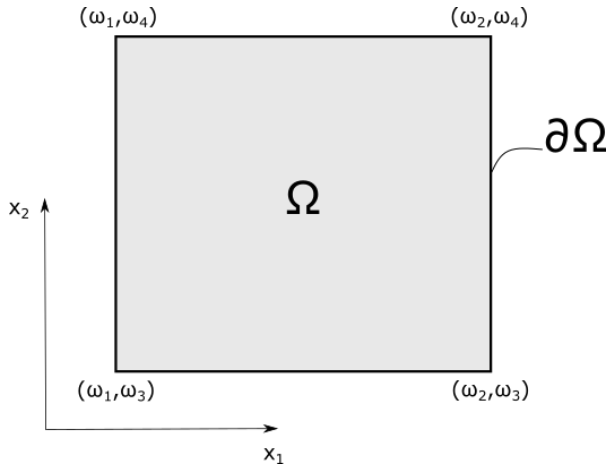


Figure 4.1: Region Ω of a two-dimensional image

4.1. Multi-level Image Registration

The Multi-Level Image Registration (MLIR) strategy uses different discretization levels to identify a suitable transformation function $\mathrm{y}(\mathrm{x})$. From a coarse grid to higher level grids, the transformation function is obtained and improved. On a coarse grid, a global correction is applied (pre-registration). In this way, priority is given to obtain first a global correction without local differences. After that, improvements of the transformation function are possible by taking the local difference into account on finer grids. A

[1] https://github.com/C4IR/FAIR.m

global correction should be in the neighborhood of the correction on the most detailed level. Therefore, fewer iterations are to be expected on the finer levels. Multiple levels can be applied before registration is executed on the finest grid. The scheme of MLIR is shown in Figure 4.2, where L is the current level, and L_{max} is the highest level of the discretization.

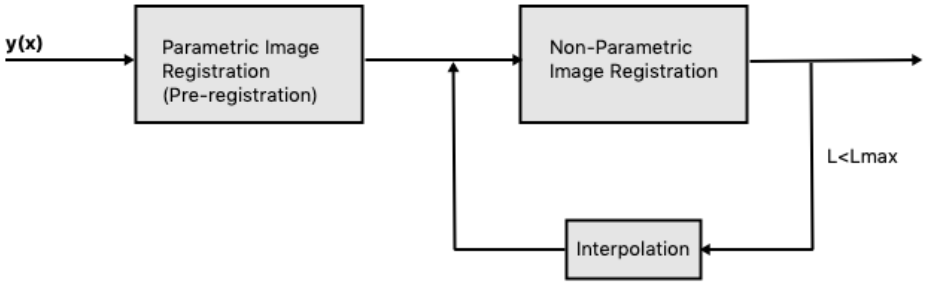


Figure 4.2: Scheme of Multi-level Image Registration

Pre-Registration

By identifying the first approximation of transformation function, only global differences are of importance. As seen in Chapter 2, Parametric Image Registration (PIR) is a cheap method for global transformations and is preferred as pre-registration. For PIR, the basis functions have to be chosen prior to the registration. Since it is not necessary to directly obtain a very accurate transformation function, only linear basis functions are considered. The transformation function is defined by

$$y(\mathbf{x}) := Q(\mathbf{x})\mathbf{f}(\mathbf{w}) \tag{4.2}$$

with parameter vector $w = [w_1, \dots, w_6]^T$ and basis function matrix $Q = \begin{bmatrix} q_1 & q_2 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_1 & q_2 & q_3 \end{bmatrix}$.
The linear basis functions are

$$\begin{aligned} q_1(\mathbf{x}) &= x_1, \\ q_2(\mathbf{x}) &= x_2, \\ q_3(\mathbf{x}) &= 1. \end{aligned}$$

Vector $\mathbf{f}(\mathbf{w}) \in \mathbb{R}^6$ depends on the kind of transformation.

- Translation, $\mathbf{f}(\mathbf{w}) = [1; 0; w_1; 0; 1; w_2]$;
- Linear affine (rotation, scaling and shearing), $\mathbf{f}(\mathbf{w}) = [w_1; w_2; w_3; w_4; w_5; w_6]$;
- Rigid (translation, rotation), $\mathbf{f}(\mathbf{w}) = [\cos(w_1); -\sin(w_1); w_2; \sin(w_1); \cos(w_1); w_3]$.

With a small set of unknowns as for the parametric case, the distance measure results in a solvable measure. The registration problem is

$$\min_{\mathbf{y}} \mathcal{J}[\mathbf{y}] = \min_{\mathbf{y}} \mathcal{D}[\mathcal{T}(\mathbf{y}), \mathcal{R}] \tag{4.3}$$

Registration

For registration at higher levels, local differences are increasingly significant. With Non-parametric Image Registration (NPIR) no constricting prior basis functions, are applied and the local differences can be fully taken into account. The measure is ill-posed and regularisation, to guarantee a solvable problem, is added.

$$\min_{\mathbf{y}} \mathcal{J}[\mathbf{y}] = \min_{\mathbf{y}} \mathcal{D}[\mathcal{T}(\mathbf{y}), \mathcal{R}] + \alpha \mathcal{S}[\mathbf{u}] \tag{4.4}$$

Calling the transformation function of the pre-registration $\mathbf{y}^{ref}$, the non-parametric transformation function is defined by

$$\mathbf{y}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) + \mathbf{y}^{ref}(\mathbf{x}) \tag{4.5}$$

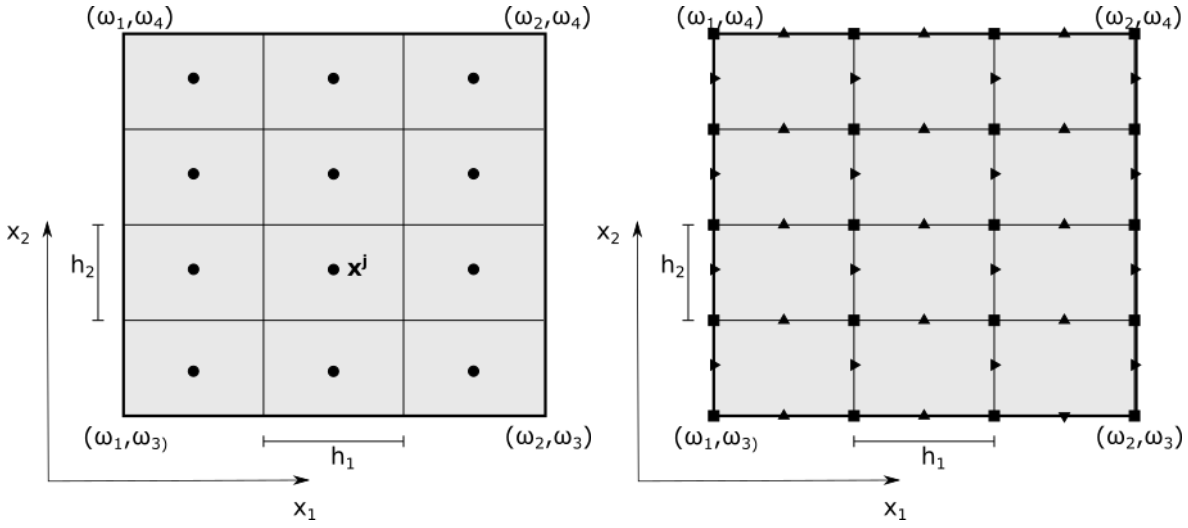
with displacement function $\mathbf{u}(\mathbf{x})$. The resulting transformation function after each level can be used as the first guess in obtaining an improved function on a higher discretization level. To apply $\mathbf{y}$ from the previous grid on a finer grid, a prolongation of $\mathbf{y}$ needs to be executed.

4.2. Discretization Grids

The information from the template and reference image are data vectors R and T, which denotes the intensity values of the pixels. Outside the domain, the intensity values are considered to be zero. Using a discretization of the domain, the cells of the discretization correspond to the pixels of the image. Therefore, it is assumed that the data points are cell-centered values. Given the region Ω, and the number of pixels in the two dimensions $\mathbf{m} = [m_1, m_2]$. The width of the cells is defined by

$$h_i = (\omega_{2i} - \omega_{2i-1})/m_i \tag{4.6}$$

Figure 4.3a shows an example of a two-dimensional cell-centered image grid.



(a) Cell-Centered image grid

(b) Discretizations of the transformation function. Staggered grid for x_1 direction (►), Staggered grid for x_2 direction (▲) and Nodal grid (■)

Figure 4.3: Discretization grids of a two-dimensional domain $\mathbf{m} = [3, 4]$. Left: Image grid; Right: Transformation grid

The cell-centered grid read

$$\begin{aligned} x_i^{j_i} &= \omega_{2i-1} + j_i h_i, \\ \mathbf{x}_c^{\mathbf{j}} &= [x_1^{j_1-0.5}, x_2^{j_2-0.5}], \qquad \text{for } j_i = 1, \dots, m_i; i = 1, 2 \\ \text{cell}^{\mathbf{j}} &= \{\mathbf{x} \in \mathbb{R}^2 | -\mathbf{h}/2 < \mathbf{x} - \mathbf{x}^{\mathbf{j}} < \mathbf{h}/2\} \end{aligned} \tag{4.7}$$

where $\mathbf{j} = [j_1, j_2]$.

For the transformation function, a different grid is used. As will be discussed for the finite difference approach, a staggered grid is needed. For the finite element method, discussed in the next chapter, a nodal grid is applied. Both grids are shown in figure 4.3b. Discretizing the transformation function $\mathbf{y} = [y_1, y_2]$ with the staggered grid, x_1 and x_2 are discretized by $\mathbf{x}_{s,1}^{\mathbf{j}}$ and $\mathbf{x}_{s,2}^{\mathbf{j}}$, respectively. The grids

are defined by

$$\begin{aligned} \mathbf{x}_{s,1}^{\mathbf{j}} &= [x_1^{j_1}, x_2^{j_2-0.5}] \quad \text{for } j_1 = 0, \dots, m_1; j_2 = 1, \dots, m_2 \\ \mathbf{x}_{s,2}^{\mathbf{j}} &= [x_1^{j_1-0.5}, x_2^{j_2}] \quad \text{for } j_1 = 1, \dots, m_1; j_2 = 0, \dots, m_2 \end{aligned} \tag{4.8}$$

For the nodal grid, the grids for y_1 and y_2 are the same and are defined by

$$\mathbf{x}_n^{\mathbf{j}} = [x_1^{j_1}, x_2^{j_2}] \qquad \text{for } j_i = 0, \dots, m_i; i = 1, 2; \tag{4.9}$$

The discrete transformation function is denoted by $y_i^{\mathbf{j}} = y_i(\mathbf{x}^{\mathbf{j}})$.

4.3. Image Interpolation

Searching for a minimum of the measure, derivatives of the data play an important role. Therefore, the image is considered to be a continuous function in which the derivatives of the image can be easily calculated. Although the data of the images are discrete, by interpolation techniques, a continuous model can be obtained. A fast and everywhere differentiable continuous model is spline interpolation. Figure 7.5 visualizes a spline interpolation for some data points.

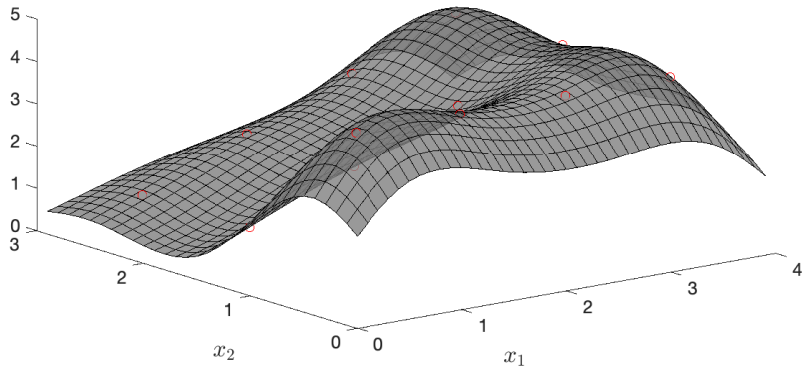


Figure 4.4: Spline interpolation of two-dimensional data points (circles)

Spline interpolation is based on the interpolation of the data by a minimization of the bending energy. Using the minimization of the bending energy, the curves of the function are limited. For the one-dimensional case, the bending energy is approximated by

$$\min_{\mathcal{T}} \mathcal{S}[\mathcal{T}] = \min_{\mathcal{T}} \int_{\Omega} \left(\frac{d^2\mathcal{T}(x)}{(dx)^2}\right)^2 d\Omega \qquad \text{subject to } \mathcal{T}(x^j) = T(j), j = 1, \dots m \tag{4.10}$$

The solution of the minimization problem are cubic splines in terms of coefficients c_j and following basis function

$$b(x) = \begin{cases} (x+2)^3, & -2 \le x < -1, \\ -x^3 - 2(x+1)^3 + 6(x+1), & -1 \le x < 0, \\ x^3 + 2(x-1)^3 - 6(x-1), & 0 \le x < 1, \\ (2-x)^3, & 1 \le x < 2, \\ 0, & \text{else.} \end{cases} \tag{4.11}$$

With the data-vector T with m data points, the spline interpolation function is written in terms of

$$\mathcal{T}^{spline}(x) = \sum_{j=1}^{m} c_j b_j(x) \tag{4.12}$$

Where $b_j(x) = b(x - j)$.

For the two-dimensional case, the coefficients in the one-dimensional case are considered splines as well. The spline interpolation function becomes

$$\mathcal{T}^{spline}(\mathbf{x}) = \sum_{j_2=1}^{m_2} \sum_{j_1=1}^{m_1} c_{j_1 j_2} b_{j_1}(x_1) b_{j_2}(x_2) \tag{4.13}$$

With the intensity values at the cell-centers, the coefficients can be calculated. The coefficients can be obtained by the following condition

$$T(j_1, j_2) = \mathcal{T}(x^{j_1, j_2}) = \sum_{k_2=1}^{m_2} \sum_{k_1=1}^{m_1} c_{k_1,k_2} b_{k_1}(x_1^{j_1}) b_{k_2}(x_2^{j_2}) \tag{4.14}$$

4.4. Discretization of the Measure

In this section, the discretization of the measure in FAIR is discussed. The continuous measure contains an integration over the domain, and the derivatives of the displacement function $\mathbf{u}$ in the regularisation plays a role. For the discrete measure, numerical approaches for the integration and derivatives need to be applied.

With an image grid at the center of the cells, the midpoint quadrature rule is a simple integration rule which evaluates the integral at the center. For the one-dimensional case, the mid-point quadrature rule is given by:

$$\int_\Omega f(\mathbf{x}) dx = h \sum_{k=1}^{j} f(x^i) + \mathcal{O}\big(h^2\big) \tag{4.15}$$

For the regularizers, it requires derivatives of the displacement function $\mathbf{u}$. For the integration approach, we want to have derivatives at the center of the cells. A central finite difference method is applied in FAIR. For the one-dimensional case it results in derivatives at the center of the cells, the method read

$$\frac{df(x^{i+.5})}{dx} = \big(f(x^{i+1}) - f(x^i)\big)/h + \mathcal{O}\big(h^2\big) \tag{4.16}$$

4.4.1. Intensity-Based Distance Measure

The FAIR toolbox features the intensity-based distance measures. Recall from Chapter 2, the measure can be written as

$$\mathcal{D}[\mathcal{T}(\mathbf{y}), \mathcal{R}] = \int_\Omega \phi\Big(\mathcal{T}(\mathbf{y}), \mathcal{R}\Big) d\Omega \tag{4.17}$$

where ϕ defines the difference between template and reference image. With the midpoint quadrature rule, the distance measure is expressed by

$$D\big(T(\mathbf{y}), R\big) = h_1 h_2 \phi\big(\mathbf{r}(\mathbf{y})\big) + \mathcal{O}\big(||\mathbf{h}||^2\big) \tag{4.18}$$

It is for ease written in outer function ϕ and inner function $\mathbf{r}(\mathbf{y})$.

4.4.2. Regularisation

In Image Registration, most regularisations are variants of the L_2-norms of the derivatives of the displacement. In general, it can be written as

$$\mathcal{S}[\mathbf{u}] = \frac{1}{2} \int_\Omega |\mathcal{B}[\mathbf{u}]|^2 d\Omega \tag{4.19}$$

where $\mathcal{B}$ is the derivative operator. For a simplified elastic regularisation, Equation 3.26, the operator is

$$\mathcal{B}^{elas} = \begin{bmatrix} \sqrt{\mu}\nabla & 0 \\ 0 & \sqrt{\mu}\nabla \\ \sqrt{\lambda+\mu}\frac{\partial}{\partial x_1} & \sqrt{\lambda+\mu}\frac{\partial}{\partial x_2} \end{bmatrix} \tag{4.20}$$

In the ideal case, the derivatives are at the center of the cells for numerical integration as for the one-dimensional case. It is impossible for two dimensions, and the central finite difference applied on a staggered grid is chosen. It results in derivatives that are partly at the centers. The other part of the derivatives is at the nodes of the cells as can be seen in the following equations.

$$\begin{aligned} \frac{\partial u_1(\mathbf{x}^{j^1-.5,j^2-.5})}{\partial x_1} &\approx (u_1^{j^1,j^2-.5} - u_1^{j^1-1,j^2-.5})/h_1, \quad \frac{\partial u_1(\mathbf{x}^{j^1,j^2})}{\partial x_2} \approx (u_1^{j^1,j^2+.5} - u_1^{j^1,j^2-.5})/h_2, \\ \frac{\partial u_2(\mathbf{x}^{j^1-.5,j^2-.5})}{\partial x_2} &\approx (u_1^{j^1-.5,j^2} - u_1^{j^1-.5,j^2-1})/h_2, \quad \frac{\partial u_2(\mathbf{x}^{j^1,j^2})}{\partial x_1} \approx (u_1^{j^1+.5,j^2} - u_1^{j^1-.5,j^2})/h_1. \end{aligned} \tag{4.21}$$

With the discretization grid of the derivatives, the midpoint quadrature rule is applied on the finite difference scheme of the integrand. The discretization of the regularisation results in

$$S(\mathbf{u}) \approx \frac{1}{2} h_1 h_2 ||B\mathbf{u}||^2 + \mathcal{O}\big(||\mathbf{h}||^2\big) \tag{4.22}$$

Where matrix B is the central finite difference approximation of $\mathcal{B}$. Details about the discretization of matrix B are discussed in [27].

4.5. Optimization Problem

The numerical optimization procedure of FAIR is a general but flexible (e.g., spatial dimensions, distance measure) automatic scheme to identify the optimal transformation. The procedure uses a Gauss-Newton approach with line-search to determine the parameters.

The discrete measure is given by

$$J(\mathbf{y}) = D(T(\mathbf{y}), R) + \alpha S(\mathbf{y} - \mathbf{y}^{ref}) \tag{4.23}$$

with $\mathbf{u} = \mathbf{y} - \mathbf{y}^{ref}$, Equation 4.5. It is a non-linear problem, and the solution can not be obtained directly. With a procedure of finding descent directions, a minimum is found in an iterative way. Therefore, the measure function is approximated by a convex function. The function J is replaced by a quadratic function $\hat{J}$ from the Taylor expansion.

$$J(\mathbf{y}+\mathbf{p}) \approx \hat{J}(\mathbf{y}+\mathbf{p}) = J + \big(\nabla J(\mathbf{y})\big)\mathbf{p}_k + \frac{1}{2}\mathbf{p}^T H(\mathbf{y})\mathbf{p} \tag{4.24}$$

where $H(\mathbf{y}) = \nabla^2 J(\mathbf{y})$ is called the Hessian matrix. When having a positive semi-definite Hessian matrix, Definition 4.1, the function $\hat{J}$ is convex and, a unique minimum exists. Withs the minimum $\mathbf{p}$ as search direction, a decrease of $J(\mathbf{y}+\mathbf{p})$ can be found. Iterate the approach, a minimum of $\mathbf{J}(\mathbf{y})$ is obtained.

Definition 4.1: Positive (semi-)definite matrix

The matrix A is called positive definite (positive semi-definite) iff

$$\forall \mathbf{u} \in \mathbb{R}^n \backslash \{\mathbf{0}\} : \mathbf{u}^T A\mathbf{u} > 0 (\mathbf{u}^T A\mathbf{u} \geq 0). \tag{4.25}$$

Gaus-Newton method

With the standard Newton-method, the search direction is obtained by solving the following linear system

$$H(\mathbf{y})\mathbf{p} = -\nabla J(\mathbf{y}) \tag{4.26}$$

Instead of using the Newton method, a modified version, the Gauss-Newton method, is considered for the data part. Having the distance measure written as $D(\mathbf{y}) = \phi(\mathbf{r}(\mathbf{y}))$, the Gauss-Newton method approximates the Hessian matrix. Higher order derivatives of the argument $\mathbf{r}(\mathbf{y})$ of $\varphi(\mathbf{r})$ are dropped. It saves computational time, and besides that, we know that the convergence rate will be the same when the remaining part is the dominant part of the Hessian matrix [29].

Applying the Gauss-Newton method to the measure, we get

$$\begin{aligned} \nabla J(\mathbf{y}) &= \nabla\phi(\mathbf{r})\nabla\mathbf{r}(\mathbf{y})P + \alpha\nabla S(\mathbf{y} - \mathbf{y}^{ref}) \\ H(\mathbf{y}) &\approx \left(\nabla\mathbf{r}(\mathbf{y})P\right)^T \nabla^2\phi(\mathbf{r})\nabla r(\mathbf{y})P + \alpha\nabla^2 S(\mathbf{y} - \mathbf{y}^{ref}) \end{aligned} \tag{4.27}$$

As can be seen, the data part also contains a matrix P. P is a grid interpolation matrix from a cell-centered grid to a staggered grid. The data part is discretized with a cell-centered grid, while the regularizer part is discretized with a staggered grid. Thus, grid interpolation is needed to have corresponding dimensions in the linear system. The interpolation matrix is discussed in [27].

The linear system can be solved by direct or iterative solution methods. For the methods, a symmetric positive definite (SPD) matrix is preferred because of the nice properties attached. The direct method, like Cholesky Decomposition, or iterative methods, Conjugate Gradient methods, are potent methods. Knowing the matrix is symmetric positive semi-definite, it will be SPD in most cases. If not, adding a small factor times the identity matrix to H will change it into a symmetric positive definite matrix.

$$H \to H + \beta I_{2n}$$

Line-Search Method

Solving the linear problem, a search direction is obtained such that with linear search methods, sufficient decreases can be ensured. The line search methods decide how far to move the function in the search direction. A popular condition is the following

$$J(\mathbf{y}_k + \alpha\mathbf{p}_k) \leq J(\mathbf{y}_k) + c_1\alpha\nabla J(\mathbf{y}_k)^T\mathbf{p}_k \tag{4.28}$$

The condition is called the Armijo condition [29]. It states that the reduction should be proportional to the step-length and the directional derivative. A standard choice of c_1 is 10^{-4}, and if a reduction is not sufficient, the condition is iterated with $\alpha = \frac{1}{2}\alpha$. Figure 4.5 visualizes the Armijo condition where the right-hand-side of the function is called $l(\alpha)$, .

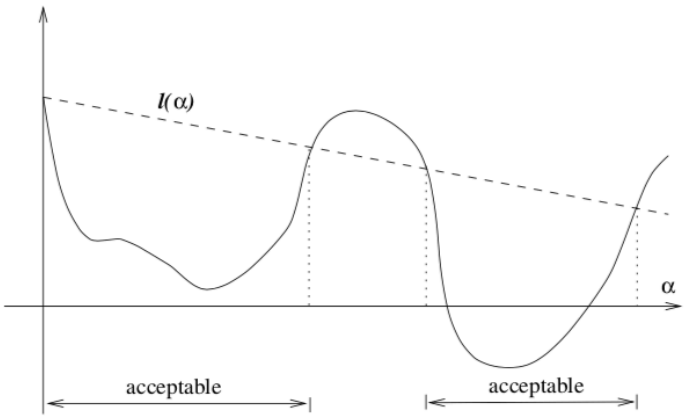


Figure 4.5: Armijo condition, taken from [29]

Stopping Criteria

As stopping criteria for the optimization problem, the following, which are common stopping criteria [13], are applied

$$|J(\mathbf{y}_{k-1}) - J(\mathbf{y}_k)| \leq \text{tol}_J(1 + |J(\mathbf{y}_{ref})|), \tag{4.29}$$

$$||\mathbf{y}_k - \mathbf{y}_{k-1}||_2 \leq \text{tol}_y(1 + ||\mathbf{y}_k||_2), \tag{4.30}$$

$$||\nabla J(\mathbf{y}_k)||_2 \leq \text{tol}_G(1 + |J(\mathbf{y}_{ref})|), \tag{4.31}$$

The first two conditions measure the relative variation after every iteration. The third condition is based on the fact that for an optimal transformation, $\mathbf{y}^*$, the transformation has to satisfy $||\nabla J(\mathbf{y}^*)||_2 = 0$. If these three conditions are fulfilled, the iterative procedure is terminated.

As safeguard, the following conditions are added as stopping criteria

$$||\nabla J(\mathbf{y}_k)||_2 \leq \epsilon, \tag{4.32}$$

$$k \leq k_{max}. \tag{4.33}$$

When an initial estimate $\mathbf{y}_0$ is close to the solution or an iterate lands very close to the solution, Equation 4.32 is satisfied and stops the iteration. To guarantee a maximum number of iterations, Equation 4.33 is implemented. As parameters, the following values are taken

Parameter	Value
tol_J	10^{-3}
tol_y	10^{-2}
tol_G	10^{-2}
ϵ	$Machine\ precision$
k_{max}	10

Table 4.1: Stopping criteria parameters

5

Finite Element Approach of Elasticity Regularisation

The finite difference method, on which FAIR is based, solves partial differential equations (PDE) using local Taylor expansions. With finite differences, the derivatives of the equations are approximated. Choosing the discretization of the domain defines the finite differences. The FDM is a fast method, but we can only apply it on structured grids. Besides that, we only know the solution at the discretization points.

The finite element method (FEM) is a numerical method for solving partial differential equations and minimization problems. FEM divides the domain in a finite set of elements. Solving the problem, the solution of every element is purely based on the information corresponding to their own element. The solution is constructed using basis functions on the elements with a certain weight. The solution is therefore defined everywhere on the domain. Because the solution is solved elements wise, it is relatively easy to implement locally different parameters and local refinement of the elements to improve the solution.

There are two different approaches to use FEM. The first approach is Ritz's method and solves a minimization problem. It expresses the solution as a linear combination of previously chosen basis functions. Using the minimization condition of the problem, it results in solving a linear system of equations. But not all PDEs can be written in terms of minimization problems. Instead, with the weak formulation, a larger class of PDEs can be solved with the Galerkin's method. This method considers the same approximation of the solution with basis functions. Replacing the test function in the weak formulation for each of the basis functions, a similar kind of system is obtained as in Ritz's method. In case there exist a minimization problem of the PDE, both methods are identical.

The chapter explains the finite element implementation of elasticity regularisation. First, Ritz's method to apply FEM is discussed. After that, the topology of the elements is treated. In this section, it elaborates the validity and distortion of the elements. The last section proposes three different methods of elastic regularisation implemented with FEM in FAIR.

5.1. Finite Element Method

This section is based on [22] and [34].

5.1.1. Ritz's Method

Consider the following two-dimensional minimization problem without any explicit boundary conditions of $\mathbf{u}$

$$\min_{\mathbf{u}\in\Sigma} \mathcal{S}[\mathbf{u}];\ \mathcal{S}(\mathbf{u}) = \int_{\Omega} \mathcal{F}[x_1, x_2, \mathbf{u}, \frac{\partial u_1}{\partial x_1}, \frac{\partial u_1}{\partial x_2}, \frac{\partial u_2}{\partial x_1}, \frac{\partial u_2}{\partial x_2}]d\Omega \tag{5.1}$$

The solution has to be found over a class of functions in the target space Σ.

$$\Sigma = \{\mathbf{u} \text{ sufficiently smooth}\}$$

Ritz's method approximates the solution of the problem by a linear combination of a finite set of known basis functions. The set of basis functions,$\{\varphi_i(\mathbf{x})\}$, is chosen in the target space. Due to the same boundary conditions for $u_1(\mathbf{x}), u_2(\mathbf{x})$, the same set of basis functions can be chosen for the approximation.

$$u_1(\mathbf{x}) \simeq \sum_{j=1}^{m} a_1^j \varphi_j(\mathbf{x}) \tag{5.2}$$

$$u_2(\mathbf{x}) \simeq \sum_{j=1}^{m} a_2^j \varphi_j(\mathbf{x}) \tag{5.3}$$

with weights $a_i^j \in \mathbb{R}$. It results in a solution which can be found in the following solution search space

$$u_i(\mathbf{x}) \in \text{span}\{\varphi_i(\mathbf{x}), i = 1, \dots, m\} \qquad (i = 1, 2) \tag{5.4}$$

The unknown functions $u_1(\mathbf{x}), u_2(\mathbf{x})$ of the minimization problem changes into a finite set of unknowns $a_1^1, a_1^2, \dots, a_1^m, a_2^1, a_2^2, \dots, a_2^m$. Substituting $u_1(\mathbf{x}), u_2(\mathbf{x})$ by Equations 5.2 and 5.3 in the minimization problem, it becomes

$$\min_{a_j^i \in \mathbb{R}^n} S(a_1^1, a_1^2, \dots, a_1^m, a_2^1, a_2^2, \dots, a_2^m) \tag{5.5}$$

Searching for a minimum, the necessary condition for the existence of a minimum read

$$\frac{\partial S(a_1^1, a_1^2, \dots, a_1^m, a_2^1, a_2^2, \dots, a_2^m)}{\partial a_j^i} = 0, \ (i = 1, 2, \dots, m; j = 1, 2) \tag{5.6}$$

This leads to $2m$ equations with $2m$ unknowns, and under certain conditions, the problem can be solved uniquely.

With the discretization of the domain, the weight is set as the solution value of the discretization points.

$$\mathbf{a} := \mathbf{u} = \begin{bmatrix} u_1^1 \\ \vdots \\ u_1^m \\ u_2^1 \\ \vdots \\ u_2^m \end{bmatrix} \tag{5.7}$$

Which follows, the basis functions have to hold the following condition at the discretization points, and it ensures that the solution at the discretization points equals the combination of basis functions.

$$\varphi_i(\mathbf{x}^j) = \delta_{ij} := \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{5.8}$$

Assuming Equation 5.6 are linear relations, it can be expressed as

$$R\mathbf{u} = \mathbf{0}. \tag{5.9}$$

The matrix $R = \nabla^2 S$ is

$$R = \begin{bmatrix} R^{1,1} & R^{1,2} \\ R^{2,1} & R^{2,2} \end{bmatrix} \text{ with } R^{k,l} = \begin{bmatrix} r_{11}^{k,l} & \cdots & r_{1m}^{k,l} \\ \vdots & \ddots & \vdots \\ r_{m1}^{k,l} & \cdots & r_{mm}^{k,l} \end{bmatrix}$$
$$\text{where } r_{ij}^{k,l} = \int_\Omega \frac{\partial^2 F(\mathbf{u})}{\partial u_k^i \partial u_l^j} d\Omega \tag{5.10}$$

5.1.2. Finite Elements

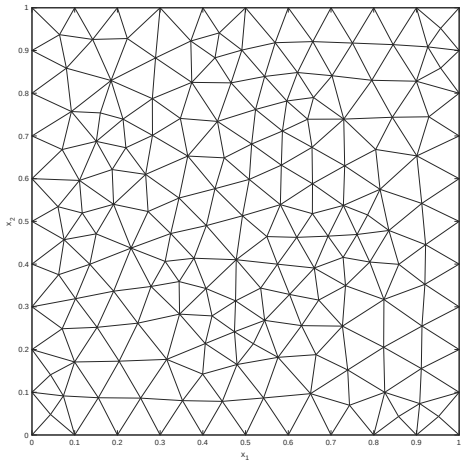


Figure 5.1: Domain of $[0,1] \times [0,1]$ divided into triangular elements

Applying a finite set of non-overlapping elements, the domain is defined as $\Omega = \bigcup_k^{n_{el}} e_k$ with $n_{el} = \#elements$, Figure 5.1. The integral inside the matrix can be written as

$$r_{ij}^{k,l} = \int_\Omega \frac{\partial^2 F(\mathbf{u})}{\partial u_k^i \partial u_l^j} d\Omega = \sum_{k=1}^{n_{el}} \int_{e_k} \frac{\partial^2 F(\mathbf{u})}{\partial u_k^i \partial u_l^j} d\Omega. \tag{5.11}$$

The matrix is a sum of all matrices applied for the different element domains. The solution inside an element is considered to be only dependent on the information of that element. Thus, only the basis functions of the corresponding nodes differ from zero. The matrix for every element reduces to

$$R^{e_k} = \begin{bmatrix} R^{1,1,e_k} & R^{1,2,e_k} \\ R^{2,1,e_k} & R^{2,2,e_k} \end{bmatrix} \quad \text{with } R^{k,l,e_k} = \begin{bmatrix} r_{11}^{l,k,e_k} & \cdots & r_{1l}^{l,ke_k} \\ \vdots & \ddots & \vdots \\ r_{l1}^{l,k,e_k} & \cdots & r_{ll}^{l,k,e_k} \end{bmatrix} \tag{5.12}$$

R^{e_k} is called the element matrix and l is the number of discretization points of the element.

The full matrix is obtained by assembling the element matrices

$$R = \sum_{k=1}^{n_{el}} Q^{e_k} R^{e_k} (Q^{e_k})^T \tag{5.13}$$

with a global mapping operator $Q^{e_k} \in \mathbb{R}^{m \times l}$. The global mapping operator maps the local nodes to the global nodes on the whole domain.

5.1.3. Shape of Elements

Dividing the domain in a finite set of non-overlapping elements, for the two-dimensional case, the triangular and quadrilateral elements are common to use. The following section elaborates on how these elements can be implemented in the numerical method.

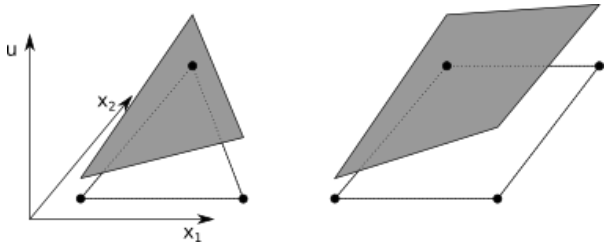


Figure 5.2: Left: linear solution of a triangular element; Right: bi-linear solution of a quadrilateral element

Choosing the number of discretization points per element, the unknown function $\mathbf{u}$ is approximated by a corresponding order of basis functions. Using the linear triangle, we have three nodal points, and the unknown function can be approximated by linear basis functions. For the bi-linear quadrilateral case, we have four discretization points and bi-linear basis functions. Examples of the solutions to these elements are visualized in Figure 5.2. The values at the nodal points are explicitly calculated, and with the basis functions the solution is known everywhere. Applying higher order elements improves the accuracy since higher order basis functions can be chosen to approximate the unknown function. Figure 5.3 shows the discretization of the triangular and quadrilateral elements for linear, quadratic, and cubic order.

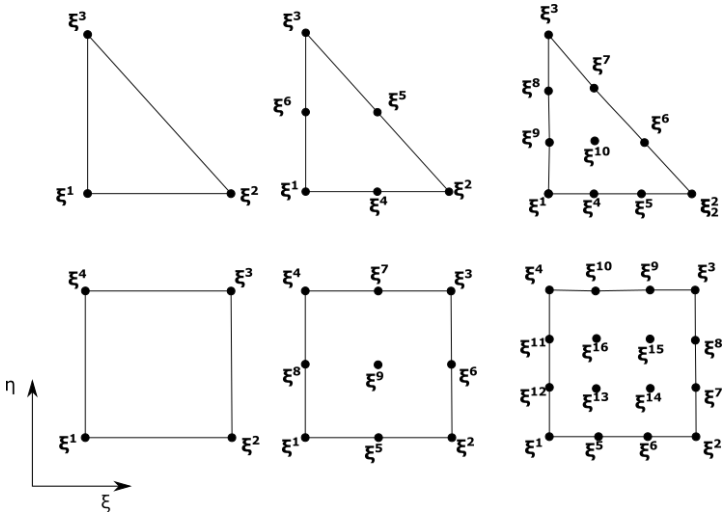


Figure 5.3: Triangular and quadrilateral element (linear, quadratic, cubic order)

Isoparametric Transformation

Every element on the domain has its pre-defined basis functions. For simplicity, a reference element can be constructed, and with Isoparemetric transformation, there is a mapping between the reference element and the element on the domain. With one reference element for all elements on the domain, only one element with basis functions has to be defined, Figure 5.4.

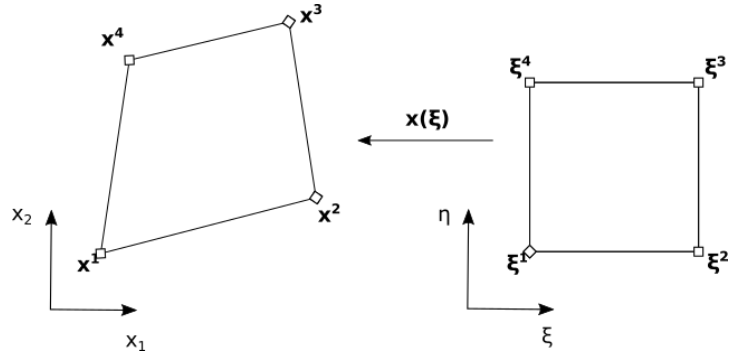


Figure 5.4: Isoparametric transformation of a bi-linear quadrilateral element

Applying an isoparametric transformation on the elements, it has to satisfy the following properties

1. The nodes $\mathrm{x}^1, \mathrm{x}^2, \ldots, \mathrm{x}^k$ are transformerd to fixed points $\xi^1, \xi^2, \ldots \xi^k$.
2. Straight sides in the original element remain straight in the reference element.
3. When basis functions in the transformed element are given by $\varphi_1(\mathrm{x}), \varphi_2(\mathrm{x}), \ldots, \varphi_k(\mathrm{x})$ then the inverse transformation $(\xi, \eta) \rightarrow (x_1, x_2)$ and interpolation is given by

$$x_i = \sum_{l=1}^{k} x_i^l \varphi_l(\xi, \eta) \qquad u_i(\mathrm{x}) = \sum_{l=1}^{k} u_i^l \varphi_l(\xi, \eta) \tag{5.14}$$

Calculating the values of the element matrices R^{e_k}, with the transformation a change of coordinates takes place for the integration

$$r_{ij}^{e_k} = \int_{e_k} \mathrm{Int}(x_1, x_2) d\Omega = \int_{e_{\xi\eta}} \mathrm{Int}(\xi, \eta) |\mathrm{Jac}_{x,\xi}| d\xi d\eta \tag{5.15}$$

Where $\mathrm{Jac}_{x,\xi}(\xi, \eta)$ is the Jacobian matrix and $|\mathrm{Jac}_{x,\xi}(\xi, \eta)|$ is the Jacobian determinant

$$\mathrm{Jac}_{x,\xi}(\xi, \eta) = \begin{bmatrix} \frac{\partial x_1}{\partial \xi} & \frac{\partial x_1}{\partial \eta} \\ \frac{\partial x_2}{\partial \xi} & \frac{\partial x_2}{\partial \eta} \end{bmatrix} \tag{5.16}$$

$$|\mathrm{Jac}_{x,\xi}(\xi, \eta)| = \frac{\partial x_1}{\partial \xi} \frac{\partial x_2}{\partial \eta} - \frac{\partial x_2}{\partial \xi} \frac{\partial x_1}{\partial \eta} \tag{5.17}$$

Basis functions

Using elements on a reference (ξ, η)-plane, the basis functions for this plane are defined. First, the linear and quadratic basis functions of the triangular elements are explained. After that, the basis functions for the quadrilateral case are discussed.

Triangular basis functions

For the triangular elements, the basis functions of the reference element, in general, can be written in terms of a polynomial. For the linear and quadratic element we have

$$\lambda_i(\boldsymbol{\xi}) = c_0^i + c_1^i \xi + c_2^i \eta \qquad \text{linear case} \tag{5.18}$$

$$\Psi_i(\boldsymbol{\xi}) = c_0^i + c_1^i \xi + c_2^i \eta + c_3^i \xi^2 + c_4^i \xi\eta + c_5^i \eta^2 \qquad \text{quadratic case} \tag{5.19}$$

Using the condition $\varphi_i(\boldsymbol{\xi}^j) = \delta_{ij}$ $(\Psi_i(\boldsymbol{\xi}^j) = \delta_{ij})$, Equation 5.8, we get a system of equations in which the parameters c_j^i are obtained.

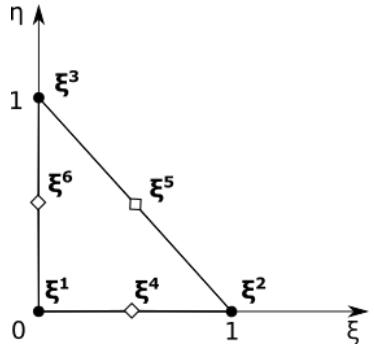


Figure 5.5: Reference Triangular element with nodes $\boldsymbol{\xi}^i$ $i = 1, 2, 3$ for linear and $i = 1, \dots, 6$ for quadratic case

Applying the equations to a reference triangle with angles at (0,0),(0,1),(1,0), Figure 5.5, the basis functions for the linear and quadratic case are

$$\lambda_1(\boldsymbol{\xi}) = 1 - \xi - \eta, \quad \lambda_2(\boldsymbol{\xi}) = \xi, \quad \lambda_3(\boldsymbol{\xi}) = \eta \qquad \text{Linear basis functions} \qquad (5.20)$$

$$\begin{cases} \Psi_i(\boldsymbol{\xi}) = \lambda_i(2\lambda_i - 1) & \text{for } i = 1, 2, 3 \\ \Psi_4(\boldsymbol{\xi}) = 4\lambda_1\lambda_2 & \Psi_5(\boldsymbol{\xi}) = 4\lambda_2\lambda_3 \quad \Psi_6(\boldsymbol{\xi}) = 4\lambda_3\lambda_1 \end{cases} \qquad \text{Quadratic basis functions} \qquad (5.21)$$

Quadrilateral basis functions

For the quadrilateral case, a rectangle is used as a reference element. Therefore, a product of two one-dimensional basis functions can be applied for the two-dimensional basis functions. The four basis functions of the bi-linear quadrilateral are thus defined as follows

$$\begin{aligned} \sigma_1(\boldsymbol{\xi}) &= \lambda_1(\xi)\lambda_1(\eta), \quad \sigma_2(\boldsymbol{\xi}) = \lambda_2(\xi)\lambda_1(\eta), \\ \sigma_3(\boldsymbol{\xi}) &= \lambda_2(\xi)\lambda_2(\eta), \quad \sigma_4(\boldsymbol{\xi}) = \lambda_1(\xi)\lambda_2(\eta) \end{aligned} \qquad (5.22)$$

For the quadratic case, the one-dimensional linear basis functions are replaced by the quadratic ones, and we get

$$\begin{aligned} \Psi_1(\boldsymbol{\xi}) &= \theta_1(\xi)\theta_1(\eta), \quad \Psi_5(\boldsymbol{\xi}) = \theta_3(\xi)\theta_1(\eta), \quad \Psi_9(\boldsymbol{\xi}) = \theta_3(\xi)\theta_3(\eta) \\ \Psi_2(\boldsymbol{\xi}) &= \theta_2(\xi)\theta_1(\eta), \quad \Psi_6(\boldsymbol{\xi}) = \theta_2(\xi)\theta_3(\eta), \\ \Psi_3(\boldsymbol{\xi}) &= \theta_2(\xi)\theta_2(\eta), \quad \Psi_7(\boldsymbol{\xi}) = \theta_3(\xi)\theta_2(\eta), \\ \Psi_4(\boldsymbol{\xi}) &= \theta_1(\xi)\theta_2(\eta), \quad \Psi_8(\boldsymbol{\xi}) = \theta_1(\xi)\theta_3(\eta), \end{aligned} \qquad (5.23)$$

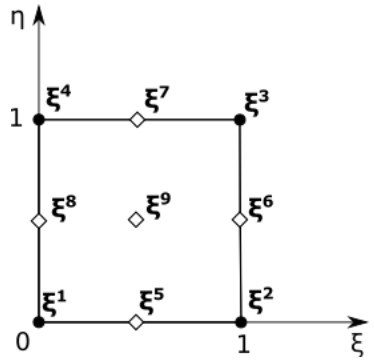


Figure 5.6: Reference quadrilateral with nodes $\boldsymbol{\xi}^i$ $i = 1, 2, 3, 4$ for linear and $i = 1, \dots, 9$ for quadratic case

Having a unit square $[0, 1] \times [0, 1]$ as reference quadrilateral, Figure 5.6, the one-dimensional basis functions on the domain $[0, 1]$ are defined by

$$\lambda_1(\xi) = 1 - \xi, \quad \lambda_2(\xi) = \xi \qquad \text{Linear basis functions} \qquad (5.24)$$

$$\theta_1(\xi) = 1 - 3\xi + 2\xi^2 \quad \theta_2(\xi) = -\xi + 2\xi^2 \quad \theta_3(\xi) = 4\xi - 4\xi^2 \qquad \text{Quadratic basis functions} \qquad (5.25)$$

5.1.4. Newton-Cotes Integration

In general, the integrals inside the matrix R are difficult to compute analytically. Using numerical integration, the elements of the matrix are approximated by l integration points with a certain weight w_k

$$\int_{e_{\xi\eta}} \text{Int}(\xi,\eta)d\xi d\eta \approx \sum_{k=1}^{l} w_k \text{Int}(\xi^k,\eta^k) \tag{5.26}$$

For an accurate solution, choosing the integration rule is of importance. Therefore, the Newton-Cotes Quadrature rule is proposed. The rule approximates the integrand with equally spaced points of the integrand and an easy to integrate interpolation function. Using the basis functions of the elements as interpolation function, the integral becomes

$$\int_{e_{\xi\eta}} \text{Int}(\xi,\eta)d\xi d\eta \approx \int_{e_{\xi\eta}} \sum_{k=1}^{l} \text{Int}(\xi^k,\eta^k)\varphi_k d\xi d\eta = \sum_{k=1}^{l} \text{Int}(\xi^k,\eta^k) \int_{e_{\xi\eta}} \varphi_k d\xi d\eta \tag{5.27}$$

Assuming we have only second order differential equations, the Newton-Cotes integration is of accuracy for the linear and quadratic elements.

5.2. Topology of Elements

This section discusses the topology of the elements. Solving the minimization problem, in case of linear elasticity, the solution is a displacement function for the elements. The displacement function could lead to physically impossible transformed elements. Therefore, the validity of the transformation for every solution is not guaranteed. Solving the minimization problem with the finite element approach. A continuous mapping exists between the initial location and the transformed location of the elements. This section explains by using the Jacobian determinant how the validity of the elements can be verified. First, the validity of elements is discussed by using the Isoparametric transformation to a reference element. After that, a measure is introduced for the distortion of the elements. Having higher order functions of the Jacobian determinant, Bezier functions are introduced. Bezier functions span the same function space as the Lagrange basis functions and have nice properties in terms of boundedness and positivity. The last part of this section discusses the Bezier functions.

5.2.1. Validity of Elements

In the registration problem, we want to find a smooth and continuous transformation function for the template image such that differences with the reference image are minimal. Searching for a solution, the topology of the image has to be preserved. Topology preserving images, are for example, stretched and/or bended images, Figure 5.7. Folded and/or cracked images doesn't preserve the topology and thus physically impossible in image registration. Having obtained a transformation, validity checking of the transformation is an essential part.

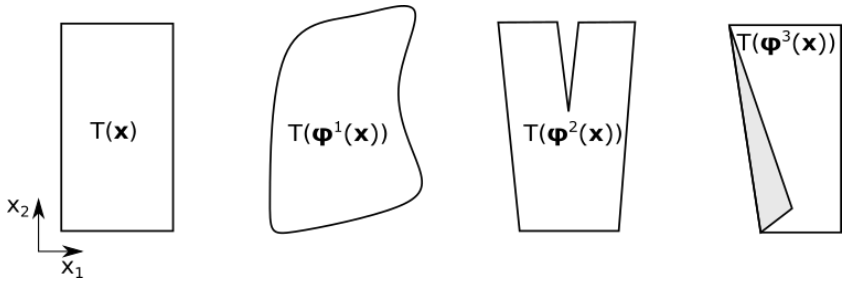


Figure 5.7: Mappings of Template image, left: original image; second left to right: stretched/bended, cracked and folded image after transformation

Using the finite element approach, valid transformations of the elements are of a specific class. The transformations of the triangular or quadrilateral elements are diffeomorphic to a simplex or square with

straight sides as a reference element, respectively [3]. Diffeomorphism functions are bijective and continuous with derivatives that are invertible (i.e., the Jacobian determinant is non-zero). Having these properties, the mapping doesn't fold or crack, and topology is preserved. Therefore, by verifying the transformed elements, local topology-preserving of the mapping can be guaranteed. Note that it says nothing about folding globally. It is still possible that the transformation causes folding between elements.

Recall from Section 5.1 the displacement function is a sum of the discretization values times the corresponding basis functions. The transformation function is

$$\mathbf{y}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x}) = \mathbf{x} + \sum_{j=1}^{m} \mathbf{u}^j \varphi_j(\mathbf{x}) \tag{5.28}$$

With Isoparametric transformation, using the mapping of Equation 5.14, the function $\mathbf{y}(\mathbf{x})$ for an element in terms a reference element is

$$\mathbf{y}(\xi,\eta) = \sum_{l=1}^{k} \mathbf{x}^l \varphi^l(\xi,\eta) + \sum_{l=1}^{k} \mathbf{u}^l \varphi^l(\xi,\eta) = \sum_{l=1}^{k} (\mathbf{x}^l + \mathbf{u}^l)\varphi^l(\xi,\eta) = \sum_{l=1}^{k} \mathbf{y}^l \varphi^l(\xi,\eta) \tag{5.29}$$

Verifying the topology of the element, the Jacobian determinant of this function needs to be non-zero. Furthermore, if the sign of the Jacobian determinant is strictly positive, also the orientation of the element hasn't changed.

The Jacobian determinant is defined as

$$|\text{Jac}_{y,\xi}(\xi,\eta)| = \frac{\partial y_1}{\partial \xi}\frac{\partial y_2}{\partial \eta} - \frac{\partial y_2}{\partial \xi}\frac{\partial y_1}{\partial \eta} \tag{5.30}$$

$$= \sum_{l=1}^{k} y_1^l \frac{\partial \varphi_l}{\partial \xi} \sum_{l=1}^{k} y_2^l \frac{\partial \varphi_l}{\partial \eta} - \sum_{l=1}^{k} y_2^l \frac{\partial \varphi_l}{\partial \xi} \sum_{l=1}^{k} y_1^l \frac{\partial \varphi_l}{\partial \eta} \tag{5.31}$$

which gives following determinants for the different kind of elements in terms of basis functions

$$|\text{Jac}_{y,\xi}(\xi,\eta)| = \begin{cases} (y_1^2 - y_1^1)(y_2^3 - y_2^1) - (y_2^2 - y_2^1)(y_1^3 - y_1^1) & \text{Linear triangular element} \\ \sum_{l=1}^{4} |\text{Jac}_{y,\xi}(\xi_l,\eta_l)|\sigma_l(\xi,\eta) & \text{Bi-linear quadrilateral element} \\ \sum_{l=1}^{6} |\text{Jac}_{y,\xi}(\xi_l,\eta_l)|\Psi_l(\xi,\eta) & \text{Quadratic triangular element} \\ \sum_{l=1}^{16} |\text{Jac}_{y,\xi}(\xi_l,\eta_l)|\mathrm{T}_l(\xi,\eta) & \text{Bi-quadratic quadrilateral element} \end{cases}$$

where σ_l are the bi-linear basis functions,Ψ_l the quadratic basis functions, and T_l bi-cubic basis functions, see appendix D.

Evaluating the Jacobian determinant at the discretization points, for first order elements it gives the lowest and the highest value of the determinant and is therefore sufficient to verify the element. For higher order elements, the Jacobian determinant function is also of higher order and the values of the determinant on the nodes doesn't provide us with bounds on the determinant. The values only indicate the changes at these nodes. Wanting to check the validity of the mapping, the equivalent quadratic and bi-cubic Bezier functions of the basis functions can be applied [20]. These functions span the same function space as the (Lagrange) basis functions and have nice properties in terms of bounding the minimum and maximum of the Jacobian determinant. The Bezier functions are explained in section 5.2.2.

Distortion

The value of the determinant can not only provide us with information about the validity of the elements but is also a measure for the distortion of the element. Comparing the Jacobian determinant of the transformation with the original element, the distortion between initial position and solution is obtained as is visualized in Figure 5.8. The measure read

$$|\mathrm{Jac}_{y,x}(\xi,\eta)| = \frac{|\mathrm{Jac}_{y,\xi}(\xi,\eta)|}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|}. \tag{5.32}$$

For $|\mathrm{Jac}_{y,x}(\xi,\eta)| = 1$ the area is equal and when $|\mathrm{Jac}_{y,x}(\xi,\eta)| < 1$ or > 1 the area is contracted or expanded, respectively.

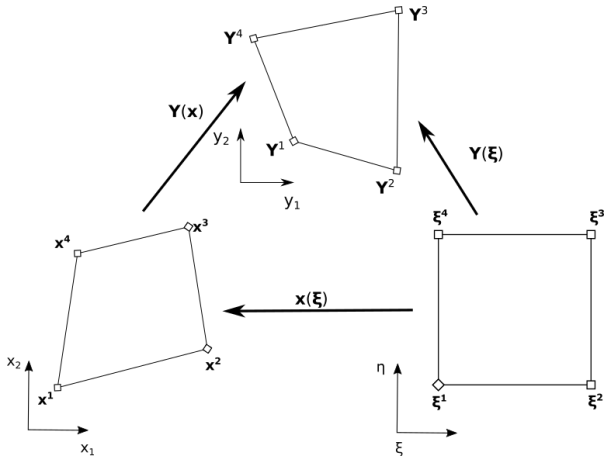


Figure 5.8: Mappings between original, solution and reference element

5.2.2. Bezier Functions

In this section, the Bezier functions are discussed. For obtaining a bound on the higher order basis functions in the Jacobian determinant, the Bezier functions are convenient. The properties of the one-dimensonial case can be extended to higher dimensions. For ease, the discussion of the properties treats only the one-dimensional functions. It is based on [7] and [20].

For the one-dimensional case, Bezier curves are expressed as Bernstein polynomials:

$$\mathcal{B}_i^{(n)}(t) = \binom{n}{i} t^i (1-t)^{n-i} \qquad (i \le n) \tag{5.33}$$

with binomial coefficient

$$\binom{n}{i} = \begin{cases} \dfrac{n!}{n!(n-i)!} & \text{if } 0 \le i \le n \\ 0 & \text{else} \end{cases}$$

Two of the properties among others are:

(a) *Positivity*: $\mathcal{B}_i^{(n)}(t) \ge 0$ for $t \in [0,1]$;

(b) *Partition of Unity*: $\sum_{j=0}^{n} \mathcal{B}_i^{(n)}(t) = 1$ for $t \in [0,1]$.

Consider a function that is based on one-dimensional (Lagrange) basis functions. The Bezier functions span the same function space and therefore

$$F(t) = \sum_{l=1}^{k} f_l \varphi_l(t) = \sum_{i=0}^{k-1} b_i \mathcal{B}_i^{(k-1)}(t) \tag{5.34}$$

Due to the properties of the Bezier function, the function $F(t)$ for $t \in [0,1]$ can be bound with the Bezier coefficients,b_i,

$$b_{min} = \sum_{i=0}^{k-1} b_{min}\mathcal{B}_i^{(k-1)}(t) \leq \sum_{i=0}^{k-1} b_i\mathcal{B}_i^{(k-1)}(t) \leq \sum_{i=0}^{k-1} b_{max}\mathcal{B}_i^{(k-1)} = b_{max} \tag{5.35}$$

where $\min_i b_i := b_{min}$, and $\max_i b_i := b_{max}$.

The Bezier function relates the Lagrange coefficients, the output of the function, and the bezier coefficients. Evaluating the relations at the nodal points, a transformation matrix is obtained, and the transformation is given by

$$\begin{gathered}\mathbf{f} = T_{\varphi\to\mathcal{B}}^{(n)}\mathbf{b} \\ \text{where } T_{\varphi\to\mathcal{B}}^{(n)} = \begin{bmatrix} \mathcal{B}_0^{(n)}(\xi_0) & \mathcal{B}_1^{(n)}(\xi_0) & \cdots & \mathcal{B}_n^{(n)}(\xi_0) \\ \mathcal{B}_0^{(n)}(\xi_1) & \mathcal{B}_1^{(n)}(\xi_1) & \cdots & \mathcal{B}_n^{(n)}(\xi_1) \\ \vdots & \ddots & \ddots & \vdots \\ \mathcal{B}_0^{(n)}(\xi_n) & \cdots & \mathcal{B}_{n-1}^{(n)}(\xi_n) & \mathcal{B}_n^{(n)}(\xi_n) \end{bmatrix}\end{gathered} \tag{5.36}$$

The inverse of the transformation matrix is the transformation from basis to Bezier coefficients, $\left(T_{\varphi\to\mathcal{B}}^{(n)}\right)^{-1} = T_{\mathcal{B}\to\varphi}^{(n)}$.

Adaptive subdivision

Furthermore, the function $F(t)$ on a sub-domain $[a,b]$ where $0 \leq a < b \leq 1$, can also be written in terms of bezier curves. It gives local bezier coefficients, $\mathbf{b}^{[a,b]}$, on the sub-domain with $t^{[a,b]} \in [0,1]$, and the function equals

$$F^{[a,b]}(t^{[a,b]}) = \sum_{i=0}^{k-1} b_i\mathcal{B}_i^{(k-1)}(t(t^{[a,b]})) = \sum_{i=0}^{k-1} b_i^{[a,b]}\mathcal{B}_i^{(k-1)}(t^{[a,b]}) \tag{5.37}$$

where $t(t^{[a,b]}) = a + (b-a)t^{[a,b]}$. With this equality, the relation between bezier coefficients on the subdomain and bezier coefficients on the whole domain is

$$T_{\mathcal{B}\to\varphi}^{(n)}\mathbf{b}^{[a,b]} = T_{\mathcal{B}\to\varphi}^{(n),[a,b]}\mathbf{b} \Longrightarrow \mathbf{b}^{[a,b]} = T_{\varphi\to\mathcal{B}}^{(n)}T_{\mathcal{B}\to\varphi}^{(n),[a,b]}\mathbf{b}$$

with transformation matrix

$$T_{\varphi\to\mathcal{B}}^{(n),[a,b]} = \begin{bmatrix} \mathcal{B}_0^{(n)}(a+(b-a)\xi_0) & \mathcal{B}_1^{(n)}(a+(b-a)\xi_0) & \cdots & \mathcal{B}_n^{(n)}(a+(b-a)\xi_0) \\ \mathcal{B}_0^{(n)}(a+(b-a)\xi_1) & \mathcal{B}_1^{(n)}(a+(b-a)\xi_1) & \cdots & \mathcal{B}_n^{(n)}(a+(b-a)\xi_1) \\ \vdots & \ddots & \ddots & \vdots \\ \mathcal{B}_0^{(n)}(a+(b-a)\xi_n) & \cdots & \mathcal{B}_{n-1}^{(n)}(a+(b-a)\xi_n) & \mathcal{B}_n^{(n)}(a+(b-a)\xi_n) \end{bmatrix}$$

Applying the division of the domain to the bound of the function $F(t)$, we get

$$\min_{i,j} b_i^{[a_j,b_j]} \leq \min F(t) \leq F(t) \leq \max F(t) \leq \max_{i,j} b_i^{[a_j,b_j]}$$

where $[0,1] = \cup_{j=1}^{k}[a_j,b_j]$ with k the total number of sub-domains.

Subdivision can be used for improvement on the bound. If the bound of the bezier coefficients on the whole domain is not sharp enough, the domain can be divided into subdomains. For example, sharper bounds are visualized in Figure 5.9 where the domain is divided into two and four subdomains with coefficients (•). As can be clearly seen, the Lagrange coefficients can't provide us with a bound while the Bezier coefficients give bounds and subdivision leads to even sharper bounds.

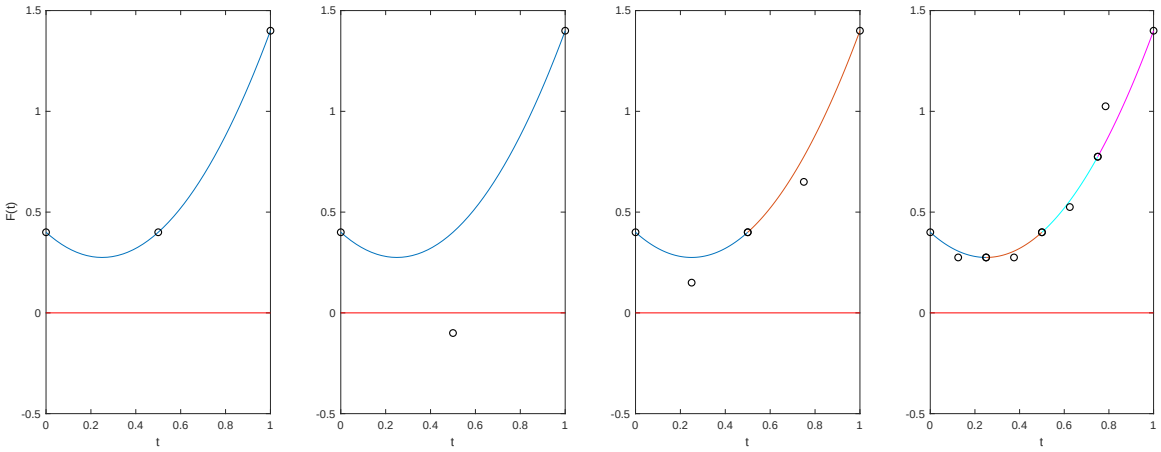


Figure 5.9: Function $F(t) = 2t^2 - t + \frac{2}{5}$ visualized in terms of Lagrange basis functions (left), Bezier curves (second left) and Bezier curves with domain divided into two and four parts (second right and right) with corresponding coefficients (o).

Bezier curves applied to the Jacobian Determinant

Rewriting our Jacobian Determinants in Bezier curves, the triangular basis functions can be written in terms of the Bezier triangular polynomials, Equation 5.38, and Bezier polynomials for quadrilateral elements are products Bernstein polynomials, Equation 5.39.

$$\mathcal{P}_{ij}^{(n)}(\xi,\eta) = \binom{n}{i}\binom{n-1}{j}\xi^i\eta^j(1-\xi-\eta)^{n-i-j} \qquad (i+j \le n) \tag{5.38}$$

$$\mathcal{Q}_{ij}^{(n)}(\xi,\eta) = \mathcal{B}_i^{(n)}(\xi)\mathcal{B}_j^{(n)}(\eta) = \binom{n}{i}\binom{n}{j}\xi^i(1-\xi)^{n-i}\eta^j(1-\eta)^{n-j} \qquad (i \le n, j \le n) \tag{5.39}$$

Applying the polynomials to the Jacobian Determinant of the (bi-)quadratic elements, it results in

$$|\mathrm{Jac}_{y,\xi}(\xi,\eta)| = \begin{cases} \sum_{l=1}^{6} |\mathrm{Jac}_{y,\xi}(\xi_l,\eta_l)|\Psi_l(\xi,\eta) = \sum_{0\le i+j\le 2} b_{ij}\mathcal{P}_{ij}^{(2)}(\xi,\eta) & \text{Quadratic element} \\ \sum_{l=1}^{16} |\mathrm{Jac}_{y,\xi}(\xi_l,\eta_l)|\mathrm{T}_l(\xi,\eta) = \sum_{i=0}^{3}\sum_{j=0}^{3} b_{ij}\mathcal{Q}_i^{(3)}(\xi,\eta) & \text{Bi-quadratic element} \end{cases} \tag{5.40}$$

5.3. Elastic Regularisation and Penalty Approaches with FEM

Implementing the finite element approach of the elastic regularisation term, three approaches are proposed. An approach is a global approach in which the same regularization parameter is applied for all elements. It is the same approach as the finite difference regularisation in FAIR. The second approach is local elastic regularization. Having our domain divided into a finite set of elements, the regularisation term is an assembly of regularisation per element. It provides us with an easy way to implement local changes and is thus proposed. The last approach is by adding a penalty term using global regularisation. The distortion of the elements can be measured using the Jacobian determinant of the elements. In that way, specific distortion values can be penalized as a soft constraint.

Global Elastic Regularisation

The simplified minimal elastic potential energy used by FAIR is

$$\mathcal{S}[\mathbf{u}] = \frac{1}{2}\int_\Omega (\mu+\lambda)(\nabla\cdot\mathbf{u})^2 + \mu(\nabla\mathbf{u}\cdot\nabla\mathbf{u})d\Omega \tag{5.41}$$

Using Ritz's method for the finite element approach and the matrix-norm, Definition 5.1, the regularisation term read

$$S(\mathbf{u}) = \frac{1}{2}||\mathbf{u}||_R^2 = \frac{1}{2}\sum_{k=1}^{n_{el}} ||\mathbf{u}||^2_{Q^{e_k}R^{e_k}(Q^{e_k})^T} \tag{5.42}$$

With element matrix R^{e_k} depending on the shape and order of the chosen element

$$R^{e_k} = \begin{bmatrix} r_{11}^{1} & \cdots & r_{1m}^{1,e_k} & r_{11}^{2,e_k} & \cdots & r_{1m}^{2,e_k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{m1}^{1,e_k} & \cdots & r_{mm}^{1,e_k} & r_{m1}^{1,e_k} & \cdots & r_{mm}^{1,e_k} \\ r_{11}^{3,e_k} & \cdots & r_{1m}^{3,e_k} & r_{11}^{4,e_k} & \cdots & r_{1m}^{4,e_k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{m1}^{3,e_k} & \cdots & r_{mm}^{3,e_k} & r_{m1}^{4,e_k} & \cdots & r_{mm}^{4,e_k} \end{bmatrix}$$

$$\begin{cases} r_{ij}^{1,e_k} = \int_{e_k} (2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + \mu\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}d\Omega \\ r_{ij}^{2,e_k} = \int_{e_k} (\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_2}d\Omega \\ r_{ij}^{3,e_k} = \int_{e_k} (\mu+\lambda)\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_2}d\Omega \\ r_{ij}^{4,e_k} = \int_{e_k} \mu\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + (2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}d\Omega \end{cases}$$

The integral is calculated by the Newton-Cotes integration rule of section 5.1.4 and is exact. For the details using different kinds of elements and the computation of the element matrix see appendix B.

Definition 5.1: Matrix-inner-product and matrix-norm

The A-inner product is defined by $(\mathbf{y},\mathbf{z})_A = \mathbf{y}^T A\mathbf{z}$ and the A-norm by

$$||\mathbf{y}||_A = \sqrt{(\mathbf{y},\mathbf{y})_A} = \sqrt{\mathbf{y}^T A\mathbf{y}} \tag{5.43}$$

Here A is a positive symmetric definite matrix

Local Elastic Regularisation

Having a finite element implementation, as can be seen in Equation 5.42, is regularisation per element. This gives possibilities to use different local regularization parameters to improve the solution.

$$S_{local}(\mathbf{u}) = \sum_{k=1}^{n_{el}} S^{e_k}(\mathbf{u}) = \sum_{k=1}^{n_{el}} \frac{\alpha_k}{2}||\mathbf{u}||^2_{Q^{e_k}kR^{e_k}(Q^{e_k})^T} \tag{5.44}$$

The different local regularization parameters can be physically interpreted in terms of local stiffness. Assuming Poisson's contraction ratio, ν, is the same everywhere, a higher regularization parameter α results in higher Young's modulus. As discussed in chapter 3, Young's modulus is the relation between stress and strain, and a higher value means stiffer material. Having higher regularization parameters gives locally stiffer material.

Global Elastic Regularisation with Penalty Term

Having a global regularisation approach, an additonal term can be added as soft constraint for heavily topology changes. Recall from chapter 2, volume-preserving penalty is a soft constraint to prevent solutions like folding and unlikely shrinking/expanding. With the finite element approach, the solution of the minimization is in terms of pre-defined basis functions. The behavior of the topology is therefore known everywhere. Having the distortion measure, Equation 5.32, a local volume-preserving method on the elements is proposed.

Applying a squared logarithmic function on the distortion measure, shrinking and expanding of the elements are equally penalized. Furthermore, it is a convex function that is useful for the optimization procedure. Unfortunately, the function is only defined for $x \in \mathbb{R}^+$ and therefore for $x \leq \epsilon > 0$ a different quadratic function is proposed where continuity and first order derivative continuity are preserved. It

results in function σ, illustrated in Figure 5.10 and defined as

$$\sigma\big(f(x)\big) = \frac{1}{2} f(x)^2. \tag{5.45}$$

with inner function

$$f(x) = \begin{cases} \log(x) & x > \epsilon \\ \log(\epsilon) + \dfrac{x-\epsilon}{\epsilon} & x \leq \epsilon \end{cases} \tag{5.46}$$

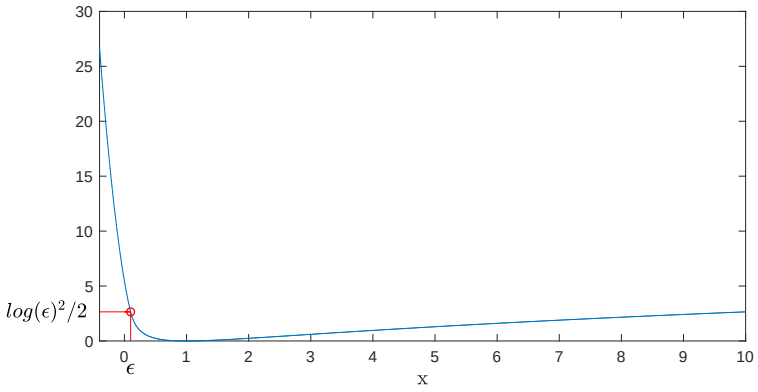


Figure 5.10: Penalty function $\sigma(f(x))$

Integrating the penalty of the distortion over the volume, the penalty term results in

$$\mathcal{P}[\mathbf{y}] = \int_{\Omega} \sigma\big(f\big(|\mathrm{Jac}_{y,x}(\mathbf{x})|\big)\big) d\Omega = \sum_{k=1}^{n_{el}} \int_{e_k} \sigma\big(f\big(|\mathrm{Jac}_{y,x}^{e_k}(\mathbf{x})|\big)\big) d\Omega \tag{5.47}$$

For ease, as for the other regularization terms, the Newton-Cotes integration rule is proposed as numerical integration. Due to the highly non-linear terms for the bi-linear and (bi-)quadratic elements, the Newton-Cotes integration isn't exact [34]. The discrete penalty function for the different elements with remainders are

$$\mathcal{P}[\mathbf{y}] = \begin{cases} \displaystyle\sum_{k=1}^{n_{el}} \frac{\Delta}{2} \sigma\big(f\big(|\mathrm{Jac}_{y,x}^{e_k}(\mathbf{x})|\big)\big) & \text{Linear element} \\ \displaystyle\sum_{k=1}^{n_{el}} \sum_{i=1}^{4} w_i \sigma\big(f\big(|\mathrm{Jac}_{y,x}^{e_k}(\mathbf{x})|\big)\big) + \mathcal{O}\big(||\mathbf{h}||^2\big) & \text{Bi-linear element} \\ \displaystyle\sum_{k=1}^{n_{el}} \sum_{i=1}^{6} w_i \sigma\big(f\big(|\mathrm{Jac}_{y,x}^{e_k}(\mathbf{x})|\big)\big) + \mathcal{O}\big(||\mathbf{h}||^3\big) & \text{Quadratic element} \\ \displaystyle\sum_{k=1}^{n_{el}} \sum_{i=1}^{9} w_i \sigma\big(f\big(|\mathrm{Jac}_{y,x}^{e_k}(\mathbf{x})|\big)\big) + \mathcal{O}\big(||\mathbf{h}||^4\big) & \text{Bi-quadratic element} \end{cases} \tag{5.48}$$

Details about the penalty term and the corresponding derivatives for the optimization procedure are worked out in Appendix C.

6

Solution Methods for Linear Problems

This chapter discusses methods to solve the linear problems with a positive definite matrix. In the first section, a weighted Least-Squares problem is explained, which corresponds with the Gauss-Newton scheme of the registration problem. After that, a fast direct method for solving the linear system is discussed. The system can be quite large when having detailed images, and it becomes practically impossible to solve the system with direct methods. The iterative Conjugate Gradient methods are treated. Iterative methods use a first guess and improves it by computing the deviation. This recursion is executed until a solution is found with sufficient precision. In the last section, a matrix-free approach of the regularisation in the solution methods is considered to reduce storage and assemble time.

6.1. Optimization Problem

From chapter 4, we've seen that the optimality criterion is a non-linear problem. By applying a Gauss-Newton type scheme, an iterative procedure is introduced in which a linear system has to be solved for obtaining a search direction. The linear system is given by, Equation 4.26 and 4.27,

$$H(\mathbf{y})\mathbf{p} = -\nabla J(\mathbf{y}) \tag{6.1}$$

with

$$\nabla J(\mathbf{y}) = \big(\nabla \mathbf{r}(\mathbf{y})P\big)^T \nabla \phi(\mathbf{r}) + \alpha \nabla S(\mathbf{y} - \mathbf{y}^{ref})$$
$$H(\mathbf{y}) \approx \big(\nabla \mathbf{r}(\mathbf{y})P\big)^T \nabla^2 \phi(\mathbf{r}) \nabla r(\mathbf{y})P + \alpha \nabla^2 S(\mathbf{y} - \mathbf{y}^{ref})$$

where H is a symmetric positive definite matrix.

Knowing the gradient of the regularisation is linear, the second derivative of $S(\mathbf{y} - \mathbf{y}^{ref})$ does not depend on $\mathbf{y} - \mathbf{y}^{ref}$ and is denoted by R. Furthermore, $\mathbf{u} = \mathbf{y} - \mathbf{y}^{ref}$, and the terms of the linear system result in

$$\nabla J(\mathbf{y}) = \big(\nabla \mathbf{r}(\mathbf{y})P\big)^T \nabla \phi(\mathbf{r}) + \alpha R\mathbf{u}$$
$$H(\mathbf{y}) \approx \big(\nabla \mathbf{r}(\mathbf{y})P\big)^T \nabla^2 \phi(\mathbf{r}) \nabla r(\mathbf{y})P + \alpha R$$

For the finite element approaches, the regularisation term is a sum of regularisation per element. For the penalty term, a Gauss-Newton approach is added to the system, and we get

$$\begin{aligned} \nabla J(\mathbf{y}) &= \big(\nabla \mathbf{r}(\mathbf{y})P\big)^T \nabla \phi(\mathbf{r}) + \alpha_{global}\Big(\sum_{k=1}^{n_{el}} \alpha_k Q^{e_k} R^{e_k} (Q^{e_k})^T\Big)\mathbf{u} + \beta\big(\nabla \mathbf{q}(\mathbf{y})\big)^T \nabla f(\mathbf{q}) \\ H(\mathbf{y}) &\approx \big(\nabla \mathbf{r}(\mathbf{y})P\big)^T \nabla^2 \phi(\mathbf{r}) \nabla r(\mathbf{y})P + \alpha_{global} \sum_{k=1}^{n_{el}} \alpha_k Q^{e_k} R^{e_k} (Q^{e_k})^T + \beta\big(\nabla \mathbf{q}(\mathbf{y})\big)^T \nabla^2 f(\mathbf{q}) \nabla \mathbf{q}(\mathbf{y}) \end{aligned} \tag{6.2}$$

Recall $\phi(\mathbf{r}), f(\mathbf{q})$ are convex, the data and penalty terms of $H(\mathbf{y}$ are thus symmetric positive semi-definite. The elastic regularisation part is symmetric positive definite and $H(\mathbf{y})$ results in a symmetric positive definite matrix.

The linear system can be seen as a special case of a (constrained) weighted least-squares problem with regularisation. Consider the following weighted least-squares problem

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{b} - A\mathbf{x}||_W^2 + \frac{1}{2}\lambda_1||\mathbf{x} + \mathbf{u}||_R^2 + \frac{1}{2}\lambda_2||\mathbf{c} - E\mathbf{x}||_V^2 \tag{6.3}$$

Taking the first derivative to $\mathbf{x}$ and set to zero, leads to the following normal equations

$$(A^T W A + \lambda_1 R + \lambda_2 E^T V E)\mathbf{x} = (A^T W)\mathbf{b} - \lambda_1 R\mathbf{u} + \lambda_2 (E^T V)\mathbf{c} \tag{6.4}$$

Alternatively, by formulating as a constrained weighted least-squares problem

$$\min_{\mathbf{d},\mathbf{x}} \frac{1}{2}||\mathbf{d}||_{W^{-1}}^2 + \frac{1}{2}\lambda_1||\mathbf{x} + \mathbf{u}||_R^2 + \lambda_2||\mathbf{c} - E\mathbf{x}||_V^2 \quad \text{with } \mathbf{d} = W(\mathbf{b} - A\mathbf{x}) \tag{6.5}$$

Applying the method of Lagrange multipliers, the equations results in

$$\mathbf{d} = W(\mathbf{b} - A\mathbf{x}), \qquad \lambda_1 R(\mathbf{x} + \mathbf{u}) + \lambda_2 (E^T V)(E\mathbf{x} - \mathbf{c}) = A^T \mathbf{d} \tag{6.6}$$

Rewriting Equation 6.1 in terms of Equation 6.3 or 6.5, we get

$$\mathbf{p} \to \mathbf{x}, -\nabla\mathbf{r}(\mathbf{y}) \to A, \nabla^2\varphi(\mathbf{r}) \to W, \nabla\varphi(\mathbf{r}) \to \mathbf{b}, \lambda_1 \to \alpha_{global},$$
$$\sum_{k=1}^{n_{el}} \alpha_k Q^{e_k} R^{e_k} (Q^{e_k})^T \to R, \mathbf{u} \to \mathbf{u}, \lambda_2 \to \beta, -\nabla\mathbf{q}(\mathbf{y}) \to E, \nabla^2 f(\mathbf{q}) \to V, \nabla f(\mathbf{q}) \to \mathbf{c};$$

For simplicity, the system is rewritten such that in the next sections the following linear system has to be solved

$$K\mathbf{w} = \mathbf{f} \qquad \text{where} \begin{cases} K = A^T W A + \lambda_1 R + \lambda_2 E^T V E, \\ \mathbf{w} = \mathbf{x}, \\ \mathbf{f} = A^T W \mathbf{b} - \lambda_1 R\mathbf{u} + \lambda_2 E^T V \mathbf{c}; \end{cases} \tag{6.7}$$

6.2. Cholesky Decomposition

This section is based on [14].

Direct methods use a form of Gaussian elimination, where the matrix is divided into two parts, an upper and lower triangular matrix. After the decomposition, with forward and backward substitution, the unknown vector is calculated. The Cholesky Decomposition is a fast direct method that uses the property of a *SPD* matrix to calculate the upper and lower triangular matrix. Consider the following system

$$K\mathbf{w} = \mathbf{f} \tag{6.8}$$

where K is a symmetric positive definite matrix. The matrix can be written as

$$K = CC^T \tag{6.9}$$

where C is a lower triangular matrix.
The decomposition of the matrix is obtained by comparing the product of the triangular matrices to the values of K. C is stored in the lower triangular part of K.

Algorithm 1 Cholesky factorization step

for $k = 1 \rightarrow n$ **do**
  $K(k,k) \leftarrow C(k,k) = \sqrt{K(k,k) - \sum_{j=1}^{k-1} C(k,j)^2}$
  for $i = k+1 \rightarrow n$ **do**
    $K(i,k) \leftarrow C(i,k) = \frac{1}{C(k,k)}\left(K(i,k) - \sum_{j=1}^{k-1} C(i,j)C(k,j)\right)$
  end for
end for

After the decomposition, with forward and back substitution, w is calculated.

$$C\mathbf{y} = \mathbf{f} \tag{6.10}$$
$$C^T\mathbf{w} = \mathbf{y} \tag{6.11}$$

Algorithm 2 Forward substitution

for $i = 1 \rightarrow n$ **do**
  $y(i) \leftarrow \big(f(i) - K(i, 1:i-1) \cdot f(1:i-1)\big)/K(i,i)$
end for

Algorithm 3 Backward substitution

for $i = 1 \rightarrow n$ **do**
  $w(i) \leftarrow \big(y(i) - K(i, 1:i-1) \cdot y(1:i-1)\big)/K^T(i,i)$
end for

6.3. Krylov Subspace Methods

Iterative approach to improve computational costs are the Krylov Subspace methods. Basic iterative methods use the following recursion to approximate vector $\mathbf{w}$

$$\mathbf{w}_k = (I - M^{-1}K)\mathbf{w}_{k-1} + M^{-1}\mathbf{f} = \mathbf{w}_{k-1} + M^{-1}\mathbf{s}_{k-1} \tag{6.12}$$

Writing out the first steps, we see that

$$\begin{aligned}\mathbf{w}_1 &= \mathbf{w}_0 + M^{-1}\mathbf{s}_0\\ \mathbf{w}_2 &= \mathbf{w}_1 + M^{-1}\mathbf{s}_1 = \mathbf{w}_0 + 2M^{-1}\mathbf{s}_0 + M^{-1}KM^{-1}\mathbf{s}_0\\ &\vdots\end{aligned}$$

which implies that the vector $\mathbf{w}_k$ will be

$$\mathbf{w}_k \in \mathbf{w}_0 + \text{span}\left\{M^{-1}\mathbf{s}_0, M^{-1}K(M^{-1}\mathbf{s}_0), \dots, (M^{-1}K)^{k-1}(M^{-1}\mathbf{s}_0)\right\} \tag{6.13}$$

Calling

$$\mathcal{K}^k(M^{-1}K, M^{-1}\mathbf{s}_0) := \text{span}\{M^{-1}\mathbf{s}_0, M^{-1}KM^{-1}\mathbf{s}_0, \dots, (M^{-1}K)^{k-1}M^{-1}\mathbf{s}_0\}$$

It spans the product of matrix $(M^{-1}K)^{j-1}$ for $j = 1 \dots, i$ with the residual of the first guess multiplied by a preconditioning matrix M^{-1}. It is called the Krylov-space of dimension k with matrix $M^{-1}K$ and initial vector $M^{-1}\mathbf{s}_0$.

The Conjugate Gradient method is a powerful method, with $M^{-1} = I$, which minimizes the K-norm of the error in the Krylov subspace. This section is based on [11] and[30].

Conjugate Gradient Method

The Conjugate Gradient method (CG) is an iterative method that uses the properties of the SPD matrix on the Krylov-subspace. The method is introduced by Hesteness and Steifel [18]. The method approximates, like the basic iterative methods, $\mathbf{w}$ by improving a first guess $\mathbf{w}_0$ with new estimates $\mathbf{w}_k$. With the usage of a search direction $\mathbf{p_k}$ and the residual $\mathbf{s}_k$ a new estimate is obtained. The K-norm, is set as a minimality condition and leads to the Conjugate Gradient Method.

$$||\mathbf{w} - \mathbf{w}_i||_K = \min_{\mathbf{y} \in \mathcal{K}^i(K;\mathbf{s}^0)} ||\mathbf{w} - \mathbf{y}||_K \tag{6.14}$$

The solution of the minimality condition results in a recursion

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{p}_k \tag{6.15}$$

$$\mathbf{s}_{k+1} = \mathbf{s}_k - \alpha_k K \mathbf{p}_k \tag{6.16}$$

$$\mathbf{p}_{k+1} = \mathbf{s}_{k+1} + \beta_k \mathbf{p}_k \tag{6.17}$$

where the search and residual vector are perpendicular $\mathbf{s}_{k+1} \perp \mathbf{s}_k$ and $\mathbf{p}_{k+1} \perp K\mathbf{p}_k$. This gives us following relation for α_k and β_k

$$\alpha_k = \frac{\mathbf{s}_k^T \mathbf{s}_k}{\mathbf{p}_k^T K \mathbf{p}_k}, \qquad \beta_k = \frac{\mathbf{s}_{k+1}^T \mathbf{s}_{k+1}}{\mathbf{s}_k^T \mathbf{s}_k}$$

The complete algorithm read

Algorithm 4 Standard CG

Require: $K \in \mathbb{R}^{n \times n}, \mathbf{w}_0, \mathbf{f} \in \mathbb{R}^n$;
Ensure: Approximate solution $\mathbf{w}_k$ such that $\|\mathbf{f} - K\mathbf{w}_k\| \leqslant TOL$.

$\mathbf{s}_0 = \mathbf{f} - K\mathbf{w}_0$; $\mathbf{p}_0 = \mathbf{s}_0$; $\gamma_0 = \mathbf{s}_0^T \mathbf{s}_0$;
while $\sqrt{\gamma_k} > TOL$ **and** $k < k_{max}$ **do**
  $\xi_k = \mathbf{p}_k^T K \mathbf{p}_k$
  $\alpha_k = \frac{\gamma_k}{\xi_k}$
  $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{p}_k$
  $\mathbf{s}_{k+1} = \mathbf{s}_k - \alpha_k K \mathbf{p}_k$
  $\gamma_{k+1} = \mathbf{s}_{k+1}^T \mathbf{s}_{k+1}$
  $\beta_k = \frac{\gamma_{k+1}}{\gamma_k}$
  $\mathbf{p}_{k+1} = \mathbf{s}_{k+1} + \beta_k \mathbf{p}_k$
  $k = k + 1$
end while

The rate of convergence of the CG method can be written in terms of an upper-bound.

$$||\mathbf{w} - \mathbf{w}_k||_K \leq 2\left(\frac{\sqrt{\kappa_2(K)} - 1}{\sqrt{\kappa_2(K)} + 1}\right)^k ||\mathbf{w} - \mathbf{w}_0||_K, \tag{6.18}$$

where $\kappa_2(K)$ is the condition-number and in case of a SPD matrix can be expressed as

$$\kappa_2(K) := ||K||_2 ||K^{-1}||_2 = \frac{\lambda_{max}(K)}{\lambda_{min}(K)}.$$

$\lambda_{max}(K)$ is the largest and $\lambda_{min}(K)$ the smallest eigenvalue of matrix K. Having eigenvalues clustered around one will result in fast convergence of the method.

Conjugate Gradient Least Squares Method

The *Conjugate Gradient for Least Squares* (CGLS) method is obtained by applying the *CG* method on the normal equations $A^T A\mathbf{x} = A^T \mathbf{b}$. The minimality condition, Equation 6.14, leads to the following

$$||\mathbf{x} - \mathbf{x}_i||_{A^T A} = \sqrt{(\mathbf{x} - \mathbf{x}_i)^T A^T A (\mathbf{x} - \mathbf{x}_i)} = \sqrt{(A(\mathbf{x} - \mathbf{x_i}))^T (A(\mathbf{x} - \mathbf{x_i}))} = \sqrt{(\mathbf{r}_i - \mathbf{r})^T (\mathbf{r_i} - \mathbf{r})}, \quad \mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i$$

It can be proven that in general, the CGLS not only minimizes $||\mathbf{r}_i - \mathbf{r}||_2^2$, but it minimizes the norm of the residual $\mathbf{r}_i$.
Applying this to the normal equation with regularisation, Equation 6.4, the linear system of the CG-method is replaced by $K \rightarrow A^T WA + \lambda_1 R + \lambda_2 E^T VE$, $\mathbf{w}_k \rightarrow \mathbf{x_k}, \mathbf{f} \rightarrow A^T\mathbf{b} - \lambda R\mathbf{u} + \lambda_2 E^T V\mathbf{c}$. Modifying it slightly by defining the constraint

$$\mathbf{r}_k = W(\mathbf{b} - A\mathbf{x}_k) \tag{6.19}$$

This gives us the following recursion of the residual

$$\mathbf{s}_k = A^T\mathbf{r}_k - \lambda_1 R(\mathbf{u} + \mathbf{x}_k) - \lambda_2(E^T V)(E\mathbf{x}_k - \mathbf{c}) \tag{6.20}$$

Introducing $\mathbf{m} := \lambda_1 R\mathbf{u} - \lambda_2 E^T V\mathbf{c}$ and $Q := E^T VE$, the recursion becomes

$$\mathbf{s}_k = A^T\mathbf{r}_k - \lambda_1 R\mathbf{x}_k - \lambda_2 Q\mathbf{x}_k - \mathbf{m} \tag{6.21}$$

Calling $\mathbf{q}_k := A\mathbf{p}_k$, the complete *CGLS* algorithm reads

Algorithm 5 CGLS

Require: $A, E \in \mathbb{R}^{m\times n}; W, V \in \mathbb{R}^{m\times m}; R \in \mathbb{R}^{n\times n}; \mathbf{x}_0 \in \mathbb{R}^n; \mathbf{b} \in \mathbb{R}^m; \mathbf{u}, \mathbf{c} \in \mathbb{R}^n; \lambda_1, \lambda_2 \in \mathbb{R} \geq 0;$
Ensure: Approximate solution $\mathbf{x}_k$ such that $\|A^T\mathbf{r}_k - \lambda_1 R\mathbf{x}_k - \lambda_2 Q\mathbf{x}_k - \mathbf{m}\| \leqslant TOL.$

$\mathbf{r}_0 = W(\mathbf{b} - A\mathbf{x}_0)$;$\mathbf{m} = \lambda_1 R\mathbf{u} - \lambda_2 E^T V\mathbf{c}$; $Q = E^T VE$; $\mathbf{s}_0 = A^T\mathbf{r}_0 - \lambda_1 R\mathbf{x}_0 - \lambda_2 Q\mathbf{x}_0 - \mathbf{m}$; $\mathbf{p}_0 = \mathbf{s}_0$; $\mathbf{q_0} = A\mathbf{p_0}$; $\gamma_0 = \mathbf{s}_0^T\mathbf{s}_0$;$k = 0$;
while $\sqrt{\gamma_k} > TOL$ **and** $k < k_{max}$ **do**
  $\xi_k = \mathbf{q}_k^T W\mathbf{q}_k + \lambda_1\mathbf{p}_k^T R\mathbf{p}_k + \lambda_2\mathbf{p}_k^T Q\mathbf{p}_k$
  $\alpha_k = \frac{\gamma_k}{\xi_k}$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$; $R\mathbf{x}_{k+1} = R\mathbf{x}_k + \alpha_k R\mathbf{p}_k$; $Q\mathbf{x}_{k+1} = Q\mathbf{x}_k + \alpha_k Q\mathbf{p}_k$
  $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k W\mathbf{q}_k$
  $\mathbf{s}_{k+1} = A^T\mathbf{r}_{k+1} - \lambda_1 R\mathbf{x}_{k+1} - \lambda_2 Q\mathbf{x}_{k+1} - \mathbf{m}$
  $\gamma_{k+1} = \mathbf{s}_{k+1}^T\mathbf{s}_{k+1}$
  $\beta_k = \frac{\gamma_{k+1}}{\gamma_k}$
  $\mathbf{p}_{k+1} = \mathbf{s}_{k+1} + \beta_k\mathbf{p}_k$
  $\mathbf{q}_{k+1} = A\mathbf{p}_{k+1}$
  $k = k + 1$
end while

Preconditioning

The rate of convergence of the Krylov subspace methods depends on the eigenvalues of the matrix. In the case of the weighted least squares problem with regularisation, the matrix is $K = A^T WA + \lambda_1 R + \lambda_2 E^T VE$. Having extreme eigenvalues, the condition number is high and results in slow convergence. Applying a left preconditioner P on the normal equations, Equation 6.22, the rate of convergence can be improved. Having the eigenvalues of $P^{-1}(A^T WA + \lambda_1 R + \lambda_2 E^T VE)$ clustered around 1, the rate of convergence will be strong. Because an extra step is required in the method, the operator has to be 'cheap'.

$$P^{-1}(A^T WA + \lambda_1 R + \lambda_2 E^T VE)\mathbf{x} = P^{-1}(A^T\mathbf{r} - \lambda_1 R\mathbf{u} + \lambda_2 E^T V\mathbf{c}) \tag{6.22}$$

The preconditioner has to be chosen somewhere between the identity matrix and the matrix of the system K . $P = I$ will give the same method as the original and is therefore not an improvement. Choosing $P = A^T WA + \lambda_1 R + \lambda_2 E^T VE$, the solution is obtained in one iteration, but then the problem is shifted to calculating the operator P^{-1}.

Applying the preconditioner to the *CGLS* method, it results in Algorithm 6.

Diagonal Scaling

The preconditioned system can be written as

$$\tilde{K}\mathbf{w} = \tilde{\mathbf{f}} \tag{6.23}$$

Algorithm 6 Preconditioned CGLS

Require: $A, E \in \mathbb{R}^{m\times n}; W, V, P \in \mathbb{R}^{m\times m}; R \in \mathbb{R}^{n\times n}; \mathbf{x}_0 \in \mathbb{R}^n; \mathbf{b} \in \mathbb{R}^m; \mathbf{u}, \mathbf{c} \in \mathbb{R}^n; \lambda_1, \lambda_2 \in \mathbb{R} \geq 0;$
Ensure: Approximate solution $\mathbf{x}_k$ such that $\|A^T\mathbf{r}_k - \lambda_1 R\mathbf{x}_k - \lambda_2 Q\mathbf{x_k} - \mathbf{m}\|_{P^{-1}} \leqslant TOL.$

$\mathbf{r}_0 = W(\mathbf{b} - A\mathbf{x}_0)$; $\mathbf{m} = \lambda_1 R\mathbf{u} - \lambda_2 E^T V\mathbf{c}$; $Q = E^T VE$; $\mathbf{s}_0 = A^T\mathbf{r}_0 - \lambda R\mathbf{x}_0 - \lambda_2 Q\mathbf{x}_0 - \mathbf{m}$; $\mathbf{z}_0 = P^{-1}\mathbf{s}_0$;$\mathbf{p}_0 = \mathbf{z}_0$; $\mathbf{q_0} = A\mathbf{p_0}$;$\gamma_0 = \mathbf{s}_0^T\mathbf{z}_0$;$k = 0$;
while $\sqrt{\gamma_k} > TOL$ **and** $k < k_{max}$ **do**
  $\xi_k = \mathbf{q}_k^T W\mathbf{q}_k + \lambda_1\mathbf{p}_k^T R\mathbf{p}_k + \lambda_2\mathbf{p}_k^T Q\mathbf{p}_k$
  $\alpha_k = \frac{\gamma_k}{\xi_k}$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k$; $R\mathbf{x}_{k+1} = R\mathbf{x}_k + \alpha_k R\mathbf{p}_k$; $Q\mathbf{x}_{k+1} = Q\mathbf{x}_k + \alpha_k Q\mathbf{p}_k$
  $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k W\mathbf{q}_k$
  $\mathbf{s}_{k+1} = A^T\mathbf{r}_{k+1} - \lambda_1 R\mathbf{x}_{k+1} - \lambda_2 Q\mathbf{x}_{k+1} - \mathbf{m}$
  $\mathbf{z}_{k+1} = P^{-1}\mathbf{s}_{k+1}$
  $\gamma_{k+1} = \mathbf{s}_{k+1}^T\mathbf{z}_{k+1}$
  $\beta_k = \frac{\gamma_{k+1}}{\gamma_k}$
  $\mathbf{p}_{k+1} = \mathbf{z}_{k+1} + \beta_k\mathbf{p}_k$
  $\mathbf{q}_{k+1} = A\mathbf{p}_{k+1}$
  $k = k + 1$
end while

with $\tilde{K} = P^{-1}(A^T WA + \lambda_1 R + \lambda_2 E^T VE), \tilde{\mathbf{f}} = P^{-1}\mathbf{f}$.
Using the diagonal matrix as a preconditioner and defining it by $p_{ii} = a_{ii}w_{ii}a_{ii} + \lambda_1 r_{ii} + \lambda_2 e_{ii}v_{ii}e_{ii}$. The matrix $\tilde{K}$ is easily calculated, and the diagonal of $\tilde{K}$ is one, which saves n multiplications in a matrix-vector product. Van der Sluis [32] showed that with the diagonal matrix as a preconditioner, it almost minimizes the condition number of matrix $\tilde{K}$. Applying the diagonal preconditioner is also called the Jacobi preconditioning.

6.4. Matrix-free approach of Regularisation

When using a Finite Element Method on the regularisation, the matrix R is assembled by the element matrices. With only matrix-vector multiplications of the matrix in the solution methods, a different approach can be applied. Instead of assembling the whole matrix R, the following approach for the matrix-vector product in the iterative methods is attractive

$$R\mathbf{v} = \Big(\sum_{k=1}^{n_{el}} Q^{e_k} R^{e_k} (Q^{e_k})^T\Big)\mathbf{v} = \sum_{k=1}^{n_{el}} Q^{e_k} R^{e_k} (Q^{e_k})^T \mathbf{v}^{e_k} \tag{6.24}$$

where n_{el} are the number of elements and v^{e_k} is the vector for the local nodes of the element.

The advantages of the elementwise approach are the assembly time and storage of the matrix. A drawback of the approach is that it takes more computations per (global) product [12].

7

Comparison of FEM and FDM

In this chapter, different experiments of the proposed finite element approach are discussed. A comparison is made with the existing finite difference method of elasticity regularisation in FAIR.

Three different sets of data are introduced, as shown in Figure 7.1. The first set of images contains two X-ray images of a human hand. The registration of these images is a simple medical example where the differences in data are clearly visible. The second data set is an artificial disc- and a c-shaped image, the transformation from disc- to the c-shaped image could lead to large deformations. Finding a valid transformation is challenging and therefore, interesting for the validity of the finite element approaches. The last set of images is a sample from the data of the histology images. Solving this problem gives besides the performance for this sample, insight into the parameters which can be used for the whole dataset.

The images of the first two data sets are 128 by 128 pixels. The original size of the histology images differ. However, the images are edited and downscaled to 320 by 256 pixels, which will be discussed in Chapter 9.

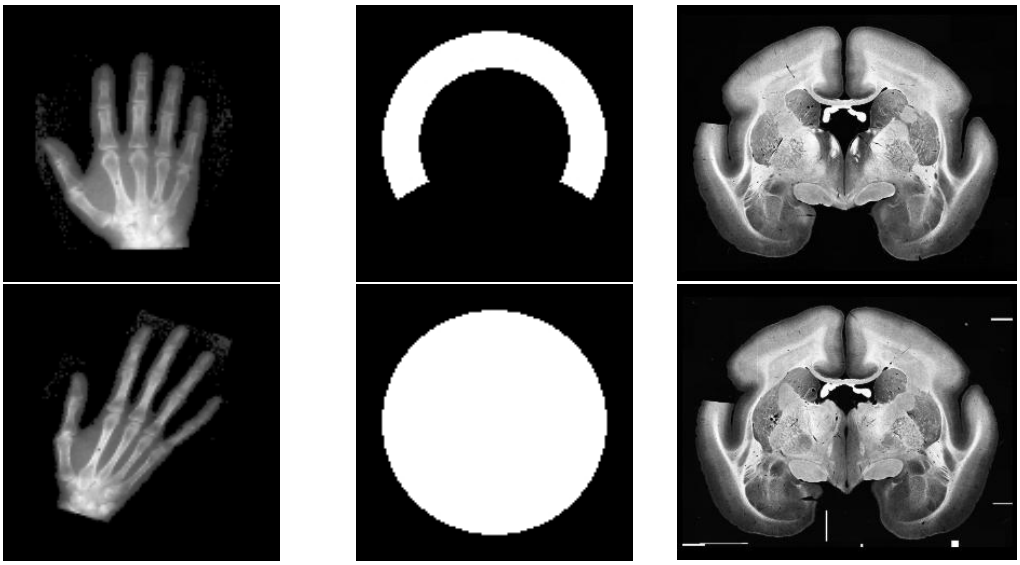


Figure 7.1: Left to right: X-ray images of human hand; disc and c-shaped image; Sample of histological data set; Top: reference images; Bottom: Template images;

This chapter is structured as follows: First, the methods and the parameters for the registration procedure are set. After that, the results are shown for the different kinds of elements compared to the finite difference method. Next, the validity of the obtained transformation functions is investigated. In the last

section, the transformations and reductions of the measure for different regularisation parameters are discussed.

7.1. Registration Approach and Parameters

For the multi-level approach, first, a global matching will be applied. A linear affine pre-registration is chosen, which allows for rotation, scaling, and translation. Improving the transformation function by going to finer grids, the following levels are defined

Data	level	pixels
Hand	$2, \dots, 7$	$2^{level} \times 2^{level}$
Disc	$2, \dots, 7$	$2^{level} \times 2^{level}$
Histology	$3, \dots, 8$	$5 \cdot 2^{level-2} \times 2^{level}$

The maximum number of iterations for the line-search and the non-linear iterations are both set to ten iterations.

The distance measure proposed is the Sum-of-Squared (SSD) distance measure, Equation 2.8. With similar acquisition conditions for template and reference images, intensity values should correspond when an accurate transformation is obtained, and the SSD distance measure is thus a robust distance measure.

For the non-parametric registration, the elastic regularisation is added to the measure, and recall, the optimization problem is

$$\min_{\mathbf{y}(\mathbf{x})} J\big(\mathbf{y}(\mathbf{x})\big); \qquad J\big(\mathbf{y}(\mathbf{x})\big) = D^{SSD}\big(T\big(\mathbf{y}(\mathbf{x})\big), R(\mathbf{x})\big) + \alpha S\big(\mathbf{y}(\mathbf{x}) - \mathbf{y}^{ref}(\mathbf{x})\big). \tag{7.1}$$

To compare the accuracy of the method, the reduction of the measure is defined by

$$\frac{J}{J_0} := \frac{J\big(\mathbf{y}(\mathbf{x})\big)}{J(\mathbf{x})} \tag{7.2}$$

The same holds for the the data D/D_0 and regularisation S/S_0 part.

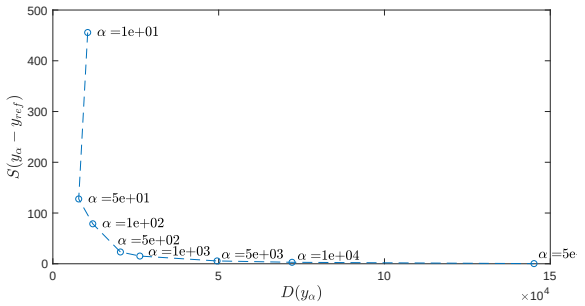
7.1.1. Regularisation and Elasticity Parameters

For the elasticity part of the function, different parameters are unknown. In the first place, we don't know the characteristics of the deformation and the Lamé parameters for the elasticity equations are thus not known. Secondly, it is unknown how much regularisation has to be added to the problem.

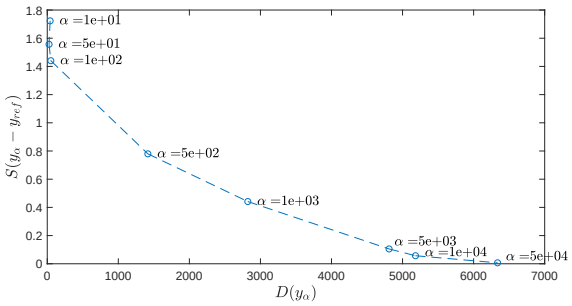
For the first Lamé parameter, the value of $\lambda = 0$ is chosen. The value is a common choice for elastic regularisation [4][26]. The second Lamé parameter has a default value of one ($\mu = 1$), because the regularisation parameter can be seen as part of the second lamé parameter. A good value of μ is combined with finding the optimal regularisation parameter. Moreover, these values are chosen because finding the correct parameter is not of importance as we are more interested in a meaningful transformation than a correct physical model.

The regularisation parameter is of choice by hand. Using a high value for the regularisation parameter, registration will result in mainly optimizing the regularization part without optimizing the data. Applying a small regularisation part, only the data part will be optimized. The solution will then be less smooth and become irregular. The irregular solutions can result in physically unlikely or impossible transformations. By registrating with different parameters, an optimal value is determined. The curves of reduction with different regularisation parameters for the bi-linear quadrilateral element are shown. For all data sets, it results in an L-shaped curve, Figure 7.2. With the L-curve approach, a suitable value can be chosen near the corner of the curve [16]. This gives a trade-off between a high reduction in data and a high reduction in regularisation. However, the regularisation ensures smooth transformations, so a

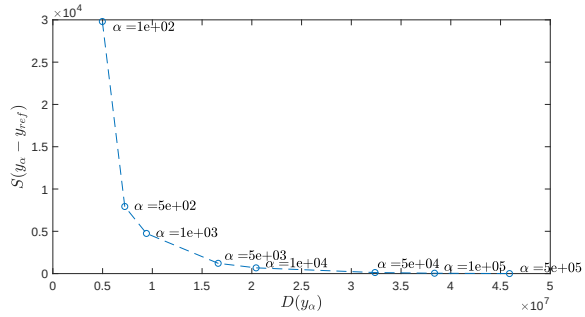
significant regularisation reduction is required. The chosen values of parameters are shown in the table of Figure 7.2.



(a) X-rays human hand



(b) Disc and C-shaped image



(c) Histology images

Data	Regularisation parameter (α)
Hand	1000
Disc	500
Histology	5000

(d) Chosen regularisation parameters

Figure 7.2: Elasticity-Data reduction curve for bi-linear quadrilateral elements

7.2. Minimization

The four different elements, linear triangular (FEM-TRI-1), bi-linear quadrilateral (FEM-QUAD-1), quadratic triangular (FEM-TRI-2), and bi-quadratic quadrilateral (FEM-QUAD-2) are compared with the finite difference method (FDM). The reduction results of the different approaches are visualized in Table 7.1. One can see that the variations in the reduction for the hand and the histology images do not differ significantly. The transformation functions of the finite difference method yield the highest reduction, but differences are not noteworthy. For the c-shaped data set, the finite element approach results in a slight improvement of the minimization, and with the linear element, it performs best.

Data	Approach	J/J_0	D/D_0
Hand	FDM	0,0455	0,0316
	FEM-TRI-1	0,0462	0,0327
	FEM-QUAD-1	0,0463	0,0328
	FEM-TRI-2	0,0458	0,0325
	FEM-QUAD-2	0,0458	0,0325
Disc	FDM	0,1881	0,1618
	FEM-TRI-1	0,1743	0,1447
	FEM-QUAD-1	0,1842	0,1568
	FEM-TRI-2	0,1739	0,1448
	FEM-QUAD-2	0,1843	0,1571
Histology	FDM	0,4447	0,3279
	FEM-TRI-1	0,4483	0,3332
	FEM-QUAD-1	0,4482	0,3331
	FEM-TRI-2	0,4427	0,3306
	FEM-QUAD-2	0,4432	0,3307

Table 7.1: Minimization reduction of the data using different FEM approaches

As we are interested in the transformed images, the images for the linear elements are visualized in Figure 7.3. For the hand and the histology problems, a nice transformation is obtained. Only for the transformed disc-shaped image, differences are clearly visible.

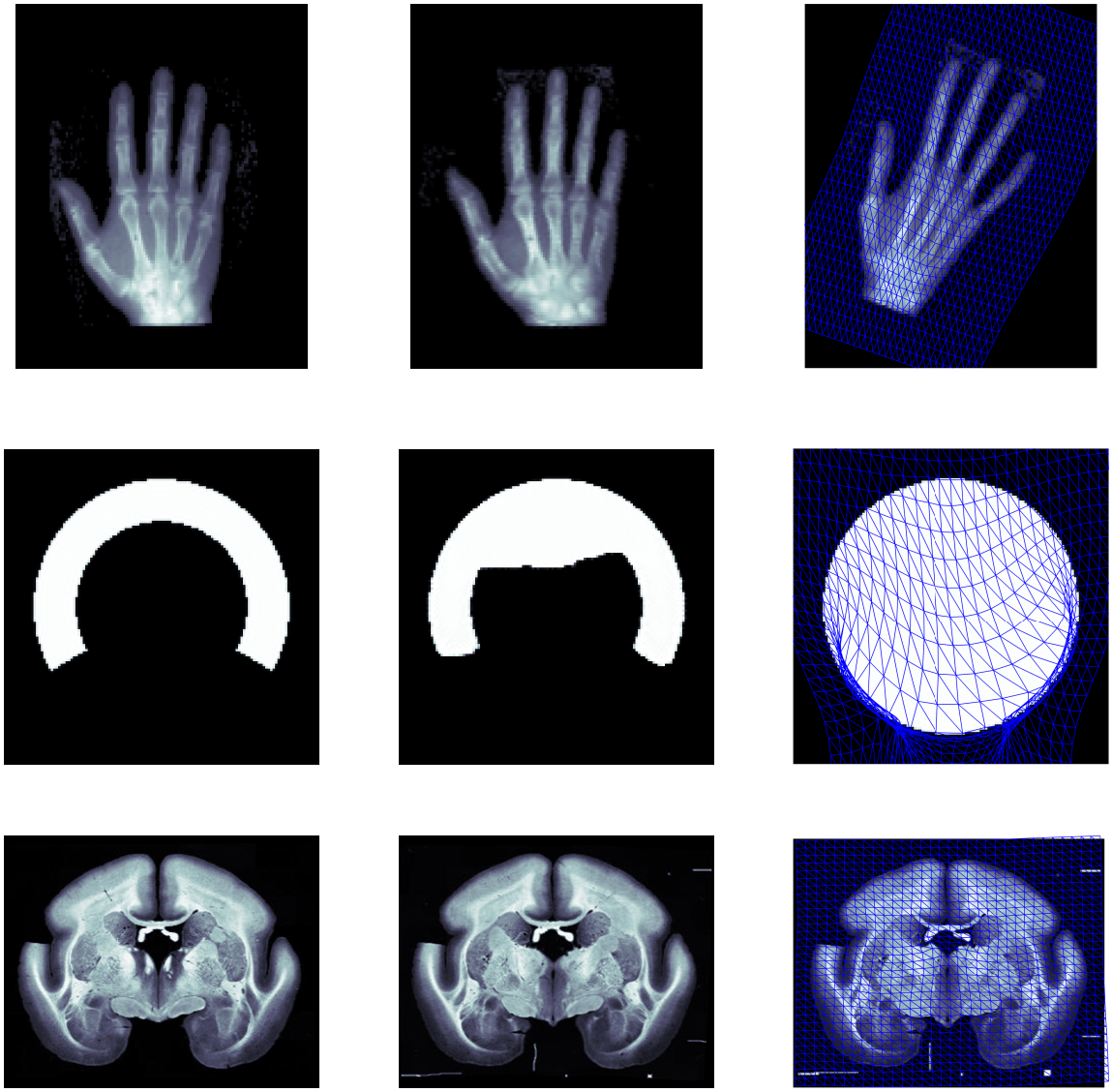
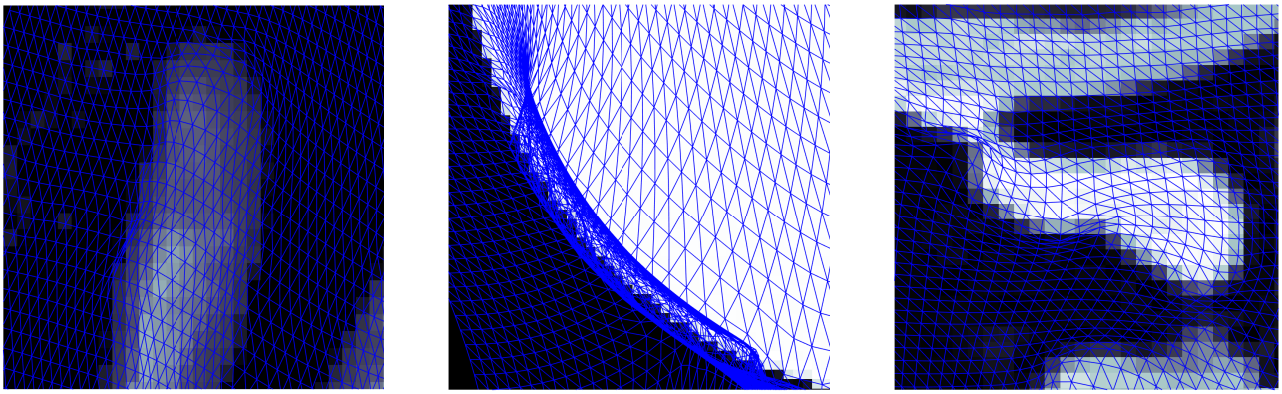


Figure 7.3: Left to right: X-ray images of human hand; disc- and c-shaped image; Sample of histological data set

7.3. Distortion

In this section, the distortion and the validity of the obtained transformation functions from the previous section are discussed. As illustrated in Figure 7.4, for all data sets, significant deformations are present with the linear triangular elements. By using the distortion measure of Chapter 5.2, Equation 5.32, the distortion at the discretization points of the elements are calculated.



(a) Hand (b) Disc (c) Histology

Figure 7.4: Zoomed in parts of the transformed grid with linear triangular element with large deformation

For the finite difference method, the solution is only known in the staggered grid points, and a distortion measure can't be constructed. Therefore, without any interpolation, measuring the distortion of the transformation is impossible. So, it is assumed that the solution in the grid points can be interpolated to linear cells. An averaging operator is applied, which results in values in the nodes and the centers of the linear cells, Figure 7.5. The averaging operator is discussed in [27]. The cells are divided into four triangles, and by using the same approach as for the linear triangular element, the distortion for the finite difference method is possible.

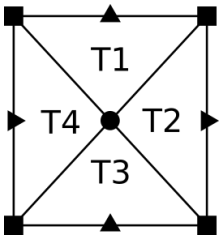


Figure 7.5: Staggered grid (►, ▲) interpolated to a linear cell with a cell-centered (•) and nodal (■) grid and divided into four linear triangles

Data	Approach	Distortion at discretization points
Hand	FDM	[0,0702 , 2,36]
	FEM-TRI-1	[0,120 , 2,44]
	FEM-QUAD-1	[0,152 , 2,41]
	FEM-TRI-2	[0,0363 , 2,70]
	FEM-QUAD-2	[0,0376 , 2,76]
Disc	FDM	[-3,44 , 8,80]
	FEM-TRI-1	[-3,15 , 11,7]
	FEM-QUAD-1	[-3,56 , 9,29]
	FEM-TRI-2	[-5,65 , 16,3]
	FEM-QUAD-2	[-4,47 , 10,1]
Histology	FDM	[-0,000366 , 4,94]
	FEM-TRI-1	[-0,225 , 5,21]
	FEM-QUAD-1	[-0,247 , 5,17]
	FEM-TRI-2	[-0,636 , 5,98]
	FEM-QUAD-2	[-0,689 , 5,91]

Table 7.2: Distortion of the transformation

The results of the distortion measure are given in Table 7.2. As observed, the transformation of the finite difference method yields the best minimum and maximum distortions. The distortions are closest to one, whereas the (bi-)linear elements show a slightly worse result. The (bi-)quadratic elements have significant poorer distortion. This could be explained by the usage of the number of data points of the corresponding elements. The higher order elements depend on more data points, compared to the (bi-)linear elements and thus has less freedom in the optimization procedure.

For the data sets, it is of importance that the transformation is physically possible. With the set of the human hand, elements at the discretization points are contracted by a factor six. As visualized in figure 7.6, the contraction of the elements are inside the hand. When only parts of the hand shrink with that factor, it is unlikely that the transformation is physically possible. The distortion of the histology data set is even more significant. The transformation shows large distortions at the defects of the images. Besides that, also contraction inside the slice is present. Furthermore, negative values of the distortion are observed, which means folding of the image and is physically impossible. As expected, the transformation to the c-shaped image causes heavy distortions and folding is unsurprising.

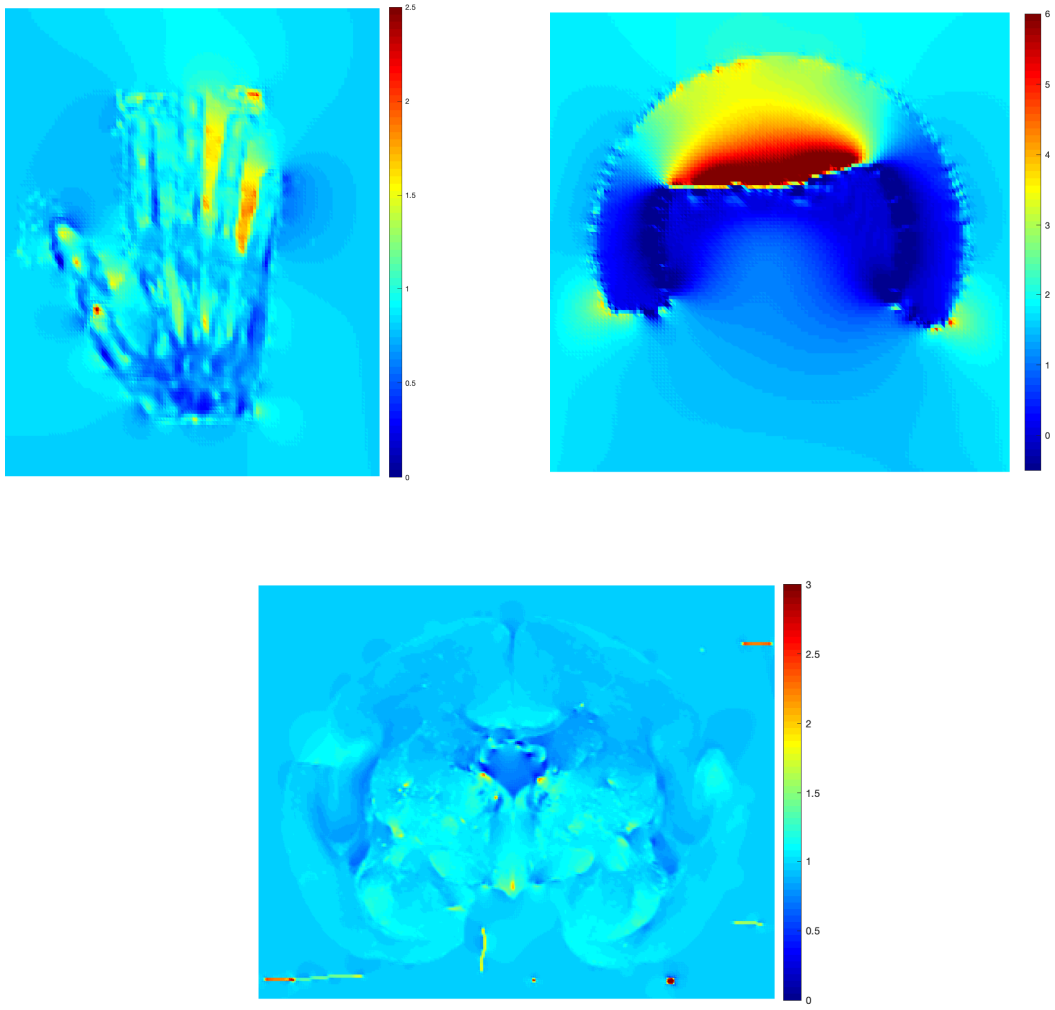


Figure 7.6: Distortion of transformation using linear triangular elements

7.4. Validity

As discussed in Section 5.2, validity can by using the Jacobian determinant of the transformation for. When the determinant is strictly positive, the transformation of an element is valid. Otherwise, the topology of the transformation isn't preserved and transformation results in folding elements. As the distortion measure is based on the Jacobian determinant, the distortion measure at discretizaton points already indicates if folding is present. For the bi-linear or linear element, the Jacobian determinant is a bi-linear function or constant, respectively, and therefore, the distortion values at discretizations points are sufficient.
For the quadratic elements, the Jacobian determinant are of higher order and checking the distortion measure at discretization points is insufficient. To apply a fast and robust method for validation, the distortion measure is written into bezier functions. Using the properties of the bezier functions, the validity can be easily determined.
If all bezier coefficients of the bezier functions are positive or negative, the element is valid or invalid, respectively, otherwise the element is undetermined. The undetermined elements are divided into subelements and the new bezier coefficients of the subelements are computed for validation. This recursion is applied till the validity of the whole element is determined, which means one of the subelements is invalid or all subelements are valid. An example of a recursion is shown in Figure 7.7.

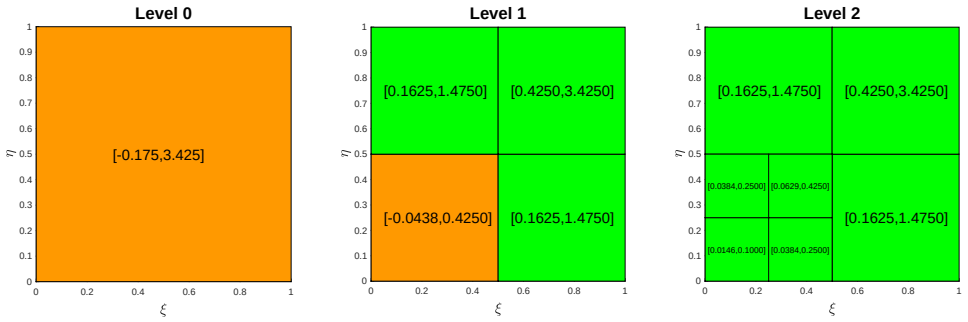


Figure 7.7: Example of validity check of elements using Bezier Coefficients

The results of the validation are shown in Table 7.3. The disc and histology data set has for all the different approaches invalid elements. Furthermore, differences are observed in the kinds of elements. The ratio of invalid to the total elements is for the linear and bi-linear elements the same, while for the quadratic elements, the ratio is higher. As discussed in the previous section, the distortion is larger for the quadratic elements due to the number of grid points per element. Thus, this also results in more invalid elements for the quadratic elements.

Data	Approach	Invalid elements	Total elements
Hand	FDM	0	65536
	FEM-TRI-1	0	32768
	FEM-QUAD-1	0	16384
	FEM-TRI-2	0	8192
	FEM-QUAD-2	0	4096
c-shaped	FDM	3991	65536
	FEM-TRI-1	2273	32768
	FEM-QUAD-1	1283	16384
	FEM-TRI-2	875	8192
	FEM-QUAD-2	414	4096
Histology	FDM	1	327680
	FEM-TRI-1	2	163840
	FEM-QUAD-1	2	81920
	FEM-TRI-2	4	40960
	FEM-QUAD-2	6	20480

Table 7.3: Distortion and Validity of the transformation

For the quadratic elements, the results of subdivision are shown in Table 7.4. Recall, the Jacobian determinant is written in terms of Lagrange basis functions. First, level 0 (L), the coefficients of the Lagrange basis functions are checked. If one of these coefficients is negative and the element is invalid, the coefficients of the element do not need to be written to the bezier coefficients. If all coefficients are positive, the element is undetermined, and coefficients are transformed to bezier coefficients. As one can see with the three test data, only for one case, the quadratic triangular element for the c-shaped data set, subdivision is required.

Data	Approach	Level	Undetermined	Invalid	Valid
Hand	FEM-TRI-2	0 (L)	8192	0	0
		0 (B)	0	0	8192
	FEM-QUAD-2	0 (L)	4096	0	0
		0 (B)	0	0	4096
c-shaped	FEM-TRI-2	0 (L)	7317	875	0
		0 (B)	1	875	7316
		1 (B)	1	875	7316
		2 (B)	0	875	7317
	FEM-QUAD-2	0 (L)	3682	414	0
		0 (B)	0	414	3682
Histology	FEM-TRI-2	0 (L)	40956	4	0
		0 (B)	0	4	40956
	FEM-QUAD-2	0 (L)	20474	6	0
		0 (B)	0	6	20474

Table 7.4: Validity check of the quadratic elements using bezier functions

7.5. Time and Memory

Besides the reduction and the validity, also time and memory are of importance for the performance of the methods. Depending on the linear solution methods, improvements can be made.

For all data sets the measure is built similarly, an SSD distance measure with elastic regularisation. Thus, for all data sets the same kind of linear system needs to be solved. For that reason, a comparison is only applied for the histological data set, and the behavior of the computational time and memory is expected to be similar for the other data sets. The time is measured with the cputime function of Matlab, which calculates the elapsed CPU time of the registration problem. It is executed on a 2.9 Ghz Intel Core i7 computer using 8Gb of RAM with Matlab version R2018A.

As discussed in Chapter 6, the Cholesky decomposition, and the iterative Krylov subspace methods, CG and the (preconditioned) CGLS method, are proposed for the finite element method. For the finite difference method, FAIR features the (preconditioned) CG method and is applied, both matrix-based (mb) as matrix-free (mf). For the iterative methods, a tolerance level for the residual is set to $\epsilon = 1e-3$. This results in similar steps for the non-linear iterations. In that way, we can apply a proper evaluation of the multi-level approach of image registration with the different solution methods and assembly of the elastic regularisation matrix.

Two different approaches of assembling the elastic matrix are considered, resulting in different impacts on the usage of memory. A matrix-based approach (mb), where the elastic matrix is fully assembled on each level. The second approach is the matrix-free method, where only the products of the matrix-vectors are stored in memory. The matrix-vector product changes for every iteration, and therefore, when many iterations are needed, the computational time will be significant.

Histology	Method	J/J_0	D/D_0	CPU time (s)
FDM	CD	0,4447	0,3279	8,8
	mbCG			9,5
	J-mbCG			10,2
	mfCG			36,6
	J-mfCG			33,0
FEM-TRI-1	CD	0,4483	0,3332	14,4
	mbCG			15,2
	mbCGLS			16,3
	J-mbCGLS			17,2
	mfCGLS			52,4
	J-mfCGLS			54,3
FEM-QUAD-1	CD	0,4482	0,3331	14,2
	mbCG			14,1
	mbCGLS			14,5
	J-mbCGLS			15,6
	mfCGLS			33,3
	J-mfCGLS			34,3
FEM-TRI-2	CD	0,4427	0,3305	15,9
	mbCG			19,5
	mbCGLS			20,7
	J-mbCGLS			22,3
	mfCGLS			40,0
	J-mfCGLS			43
FEM-QUAD-2	CD	0,4432	0,3307	15,3
	mbCG			19,4
	mbCGLS			21,2
	J-mbCGLS			19,3
	mfCGLS			31,6
	J-mfCGLS			34,4

Table 7.5: Results of the computational time using different linear solution methods

As shown in Table 7.5, the finite difference method implementation yields best for all different implementations. One of the benefits with the finite difference method is the assembling time, and it is, therefore, to be expected that computational costs are the lowest.
Using the CG or (preconditioned) CGLS method, no optimal method can be observed. Differences are small for all elements. The matrix of the linear system appears not to be diagonal dominant, as the diagonal preconditioning doesn't improve results.
The matrix-free methods increase the computational time and are related to the number of elements. Recall from chapter 6, the matrix-free approaches are a sum of element vectors. Table 7.5 shows results as expected, for linear elements, the most significant increase of computational time by a factor 3,2. For the bi-quadratic elements, the increase is a factor 1,5, and thus, it results that the bi-quadratic elements perform best for all elements.
For the matrix-free approach, the vector size is the number of discretization points in x_1 and x_2 direction. For the matrix-based approach, the matrix is sparse, and only non-zero elements are stored. The non-zero elements are based on the discrete points of the element, and thus, memory usages depend on the kind of element. However, for all elements, it holds that the matrix-free approach gives a significant reduction of memory, Table 7.6.

Approach	Matrix-based	Matrix-free
FDM	1475132	164416
FEM-TRI-1	1970688	164994
FEM-QUAD-1	1482634	
FEM-TRI-2	2381440	
FEM-QUAD-2	2632970	

Table 7.6: Number of values stored for matrix-based and matrix-free linear solution methods

7.6. Results using Different Regularisation Parameters

In the previous sections, the minimization of the measure is analyzed, using different kinds of elements, compared with the finite difference method, and is applied for one regularisation parameter. In this section, for multiple regularisation parameters and the different kinds of elements, registration is executed and compared.

For the different data sets, the reduction, data reduction, and ratio of invalid elements are plotted against the regularisation parameter, visualized in Figure 7.8,7.9 and 7.10. As one can see, as previously observed, only for the c-shaped data set, the differences in the reduction and data reduction between methods are present. However, due to the high ratio of invalid elements, no meaningful conclusion can be drawn for the transformation. Looking at the other data sets, using different kinds of elements doesn't result in significant differences in terms of reduction.
For the ratio of invalid elements, the interpolated finite difference transformation performs best. For the finite element methods, the (bi-)linear element results in better ratios compared to the (bi-)quadratic elements.

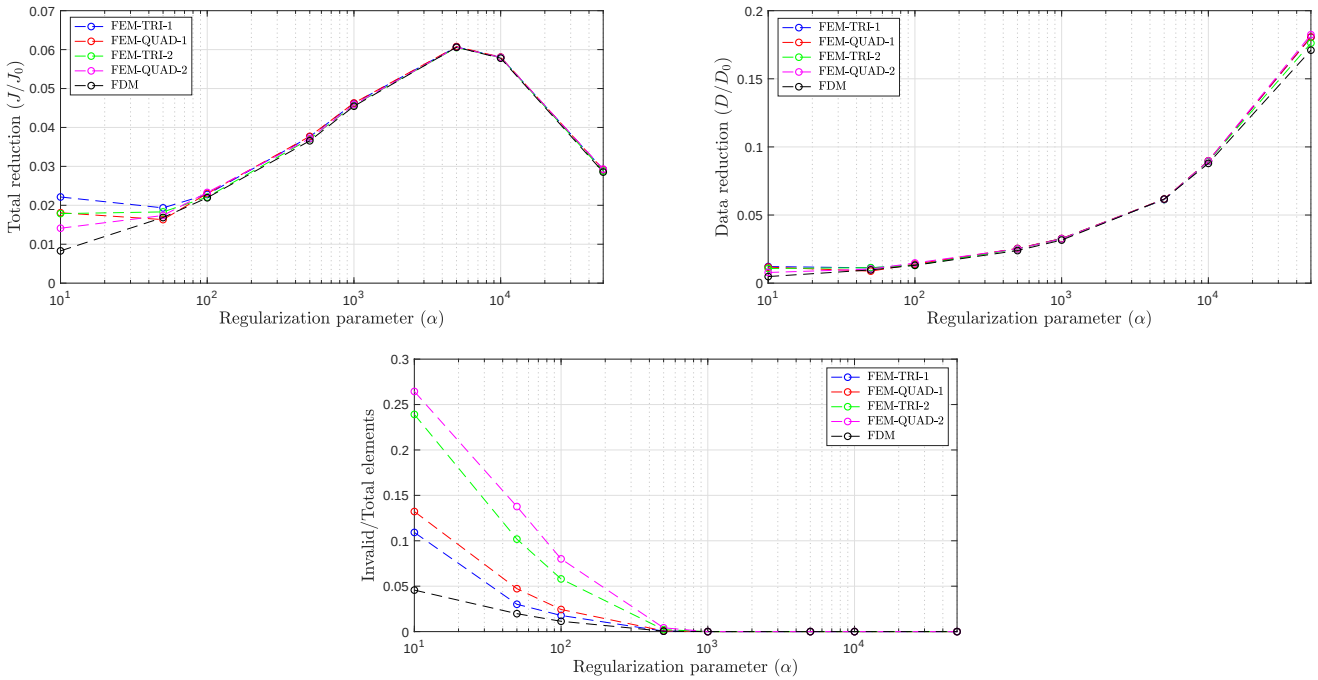


Figure 7.8: Total, Data reduction and ratio invalid elements plotted against regularisation parameter for hand data set

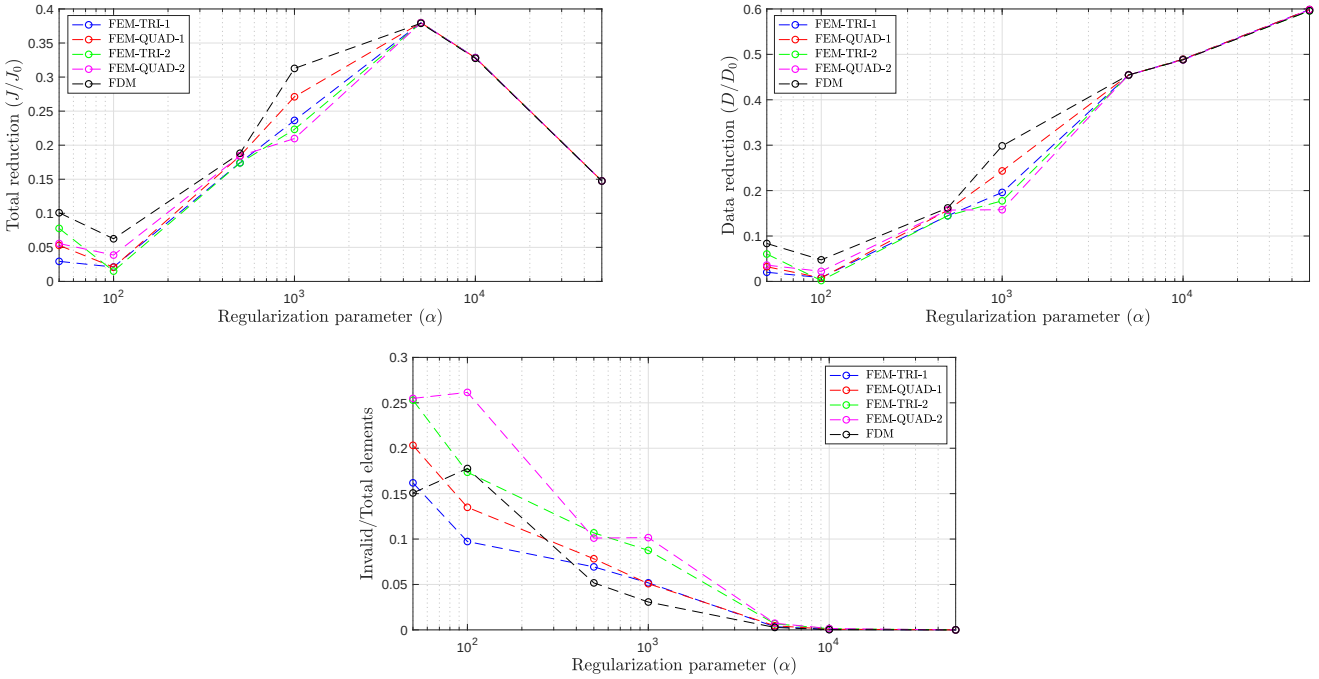


Figure 7.9: Total, Data reduction and ratio invalid elements plotted against regularisation parameter for c-shaped data set

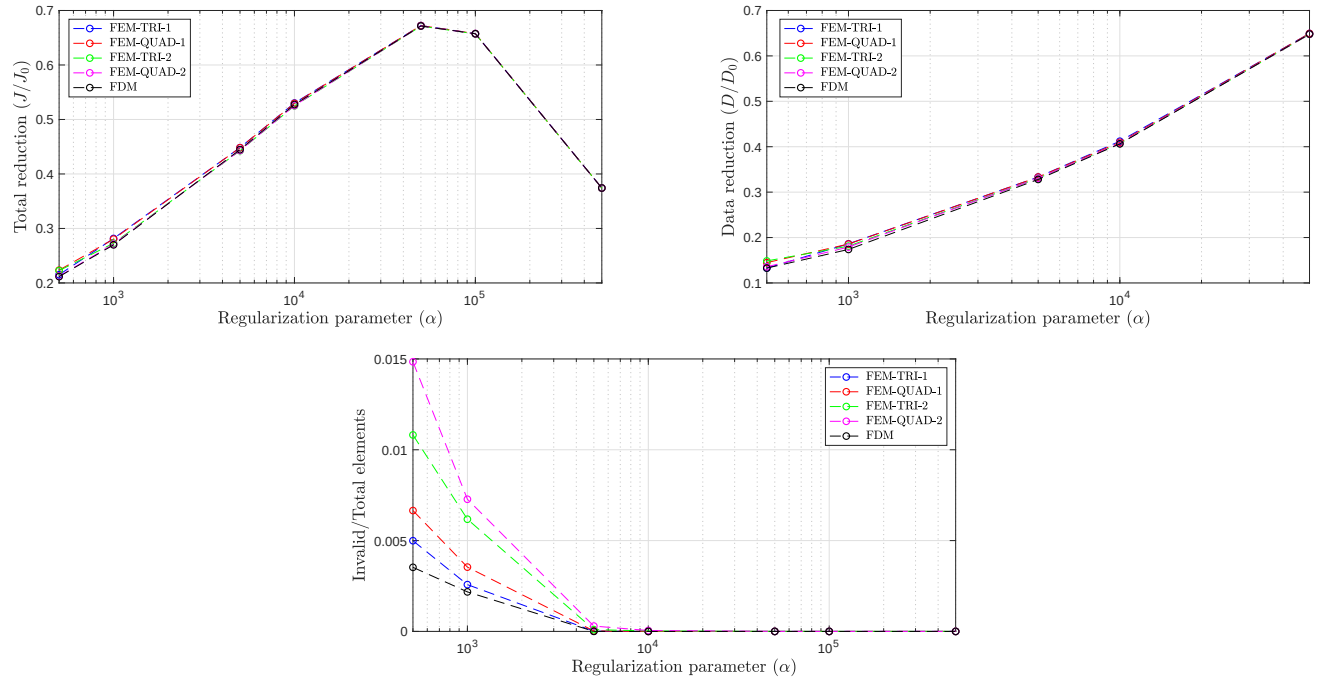


Figure 7.10: Total, Data reduction and ratio invalid elements plotted against regularisation parameter for histology data set

8

Results of Different Regularisation Approaches using FEM

In the previous chapter, a comparison is made with FEM and FDM. In order to have a high reduction in data and regularisation, it leads to significant distortions and/or invalid transformations. Increasing the regularisation parameter resolves the issue of invalid elements, but it worsens the reduction significantly. In this chapter, the two other approaches of Section 5.3 are investigated to obtain a meaningful transformation with high reduction. The first approach is local regularisation by local stiffness, and secondly, a penalty approach on the volume of the elements. In this chapter, these two approaches are investigated and compared.

All three data sets from Chapter 7 are used in this chapter.

8.1. Results of Local regularisation

With a local regularisation parameter, regularisation is based on the local stiffness of the elements. The elements which tend to cause large deformations are made stiffer. Not knowing beforehand how the elements behave during the minimization process, it is not clear what to choose. Therefore, an implementation is chosen in which the elements are evaluated per iteration. First, a global regularization parameter is determined for which elasticity reduction and data reduction are combined significant, figure 7.2. After that, the local regularization parameters are increased based on the distortion. It is defined by the following measure

$$\begin{aligned} J\big(\mathbf{y}(\mathbf{x})\big) &= D^{SSD}\big(\mathbf{y}(\mathbf{x})\big) + \alpha_{global} \sum_{k=1}^{n_{el}} (\alpha_k + 1) S^{elas,e_k}\big(\mathbf{y}(\mathbf{x}) - \mathbf{y}^{ref}(\mathbf{x})\big) \\ \text{with } \alpha_k &= \max_{\mathbf{x}} \Big(\alpha_{local}\big(|\mathrm{Jac}^{e_k}_{\mathbf{y},\mathbf{x}}(\mathbf{x})|\big)\Big) \end{aligned} \tag{8.1}$$

where α_{local} is a penalty function.

The function $\alpha_{local}(x) = wf(x)^2$ is chosen. With function $f(x)$, defined in Equation 5.45, and a weight w. The penalty function ($\alpha_{local}(x)$), penalizes contraction and expansion equally and is visualized in Figure 8.1.

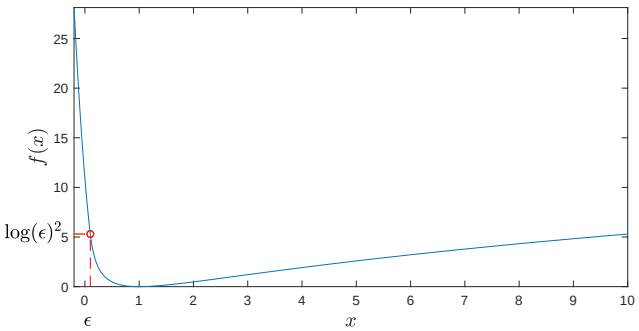


Figure 8.1: Penalty function

To determine the balance between local and global regularisation, different weights are considered to penalize the distorted elements. In the following paragraphs, the results applied to the different data sets are treated.

Hand

In Table 8.1, the results for the triangular elements are shown. For comparison, the results with the global regularisation parameter are included. Adding a local regularisation, it results in all cases in decreased deformations of the elements. The data is thereby less reduced, but compared to global regularization with the same maximum distortion, the reduction is improved.

Figure 8.2 visualizes the results applied to the hand data set for the (bi-)linear and (bi-)quadratic elements. The linear triangular yields the best reduction. For the distortion, the minimum and maximum of the distortion values at the discretization points are visualized. For all kind of elements, the distortion is significantly reduced. As one can see, for a weight of one, the distortion is already close to the neighborhood of contraction or expansion of factor two.

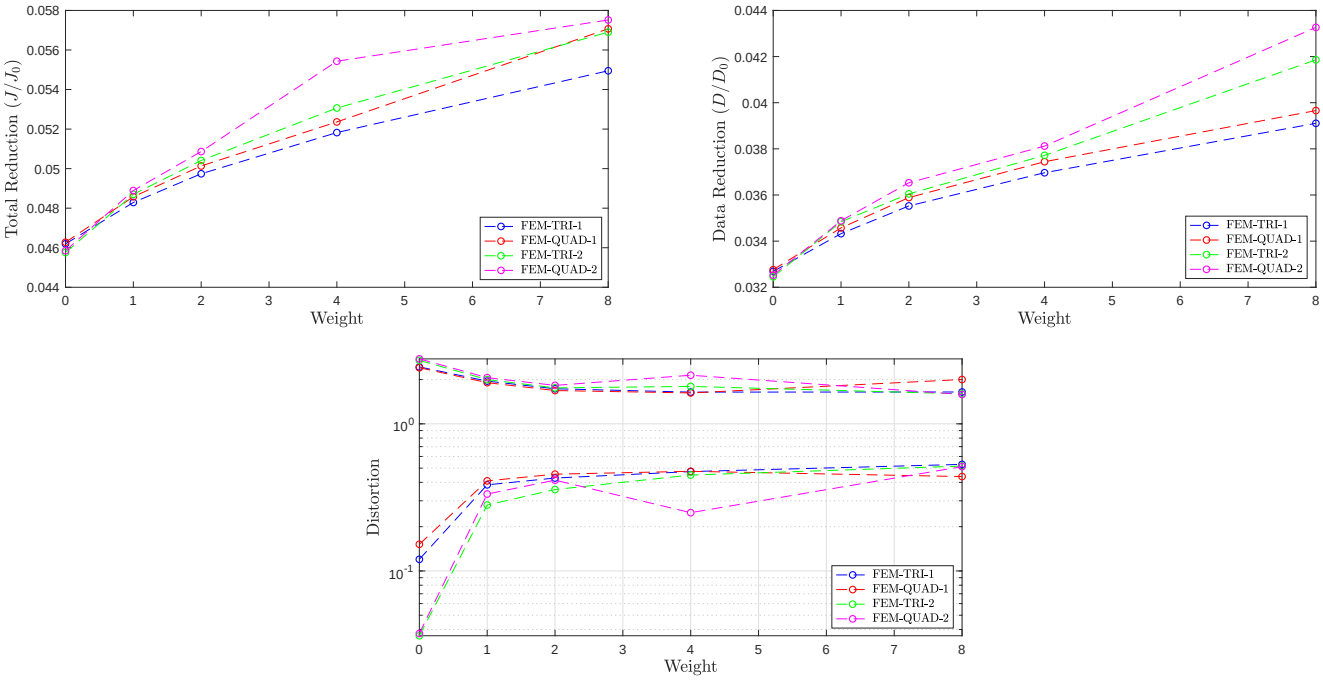


Figure 8.2: Results of local regularisation for the human hand. Top: Total reduction (left) and data reduction (Right); Bottom: Minimum and maximum distortion at the discretization points;

α_{global}	w	J/J_0	D/D_0	Distortion	Invalid elements
1000	1	0,0462	0,0327	[0,120 , 2,44]	0
	1	0,0483	0,0343	[0,385 , 1,95]	0
	2	0,0497	0,0355	[0,428 , 1,73]	0
	4	0,0518	0,0370	[0,473 , 1,64]	0
	8	0,0549	0,0391	[0,530 , 1,65]	0
2000	1	0,0544	0,0421	[0,394 , 1,69]	0
3000	1	0,0582	0,0491	[0,492 , 1,53]	0

Table 8.1: Local regularisation applied to the hand data set using linear triangular elements

C-shaped data set

For the c-shaped data set, large deformations are to be expected in order to have similar images between reference and template image. Therefore, making the element stiffer results in worse data reduction as can be seen in Figure 8.3 and Table 8.2. The distortion graph doesn't show a lower bound everywhere, which means elements are still folding. Local regularisation leads to fewer invalid elements compared to the global approach, but due to the reduction, making the elements stiffer isn't a well-suited approach for this data set.

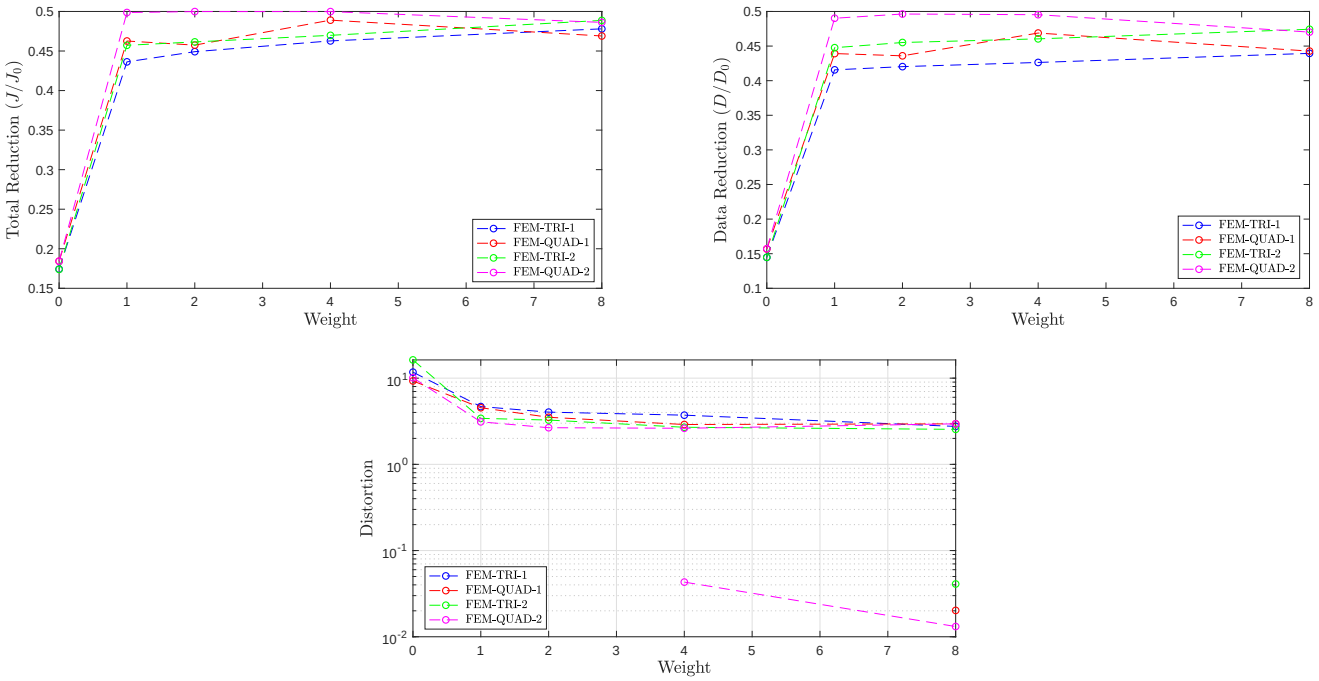


Figure 8.3: Results of local regularisation for disc- and c-shaped image. Top: Total reduction (left) and data reduction (Right); Bottom: Minimum and maximum distortion at the discretization points;

α_{local}	$\alpha_{global}(x)$	J/J_0	D/D_0	Distortion	Invalid elements
500	1	0,1743	0,1447	[-3,15 , 11,7]	2273
	$f(x)$	0,4363	0,4156	[-0,436 , 4,68]	261
	$2f(x)$	0,4492	0,4203	[-0,291 , 4,03]	199
	$4f(x)$	0,4628	0,4263	[-0,184, 3,72]	114
	$8f(x)$	0,4780	0,4395	[-0,0513, 2,76]	23
1000	1	0,2364	0,1960	[-4,10 , 8,23]	1697
1500	1	0,3168	0,2937	[-3,49 , 5,64]	1159

Table 8.2: Local regularization applied on the c-shaped data set using linear triangular elements

Histology

Figure 8.4 and Table 8.3 shows the results for the histology data set. One can see that the differences between local regularization and global regularization with the same maximum amount of distortion differ substantially. The distortion is reduced with local regularization from folding elements and expansion of factor five to contraction and expansion in the neighborhood of factor two. For a maximum distortion around factor two, the reduction for the linear triangular element between global and local approach is for total and data measure, $57,88\%$ to $47,67\%$ and $46,84\%$ to $36,22\%$, respectively. Furthermore, using the same global regularisation without and with local regularization, the reduction between the transformed template and reference image has only worsened from $33,32\%$ to $36,22\%$. Also, for this data set, the linear elements yield the best result.

α_{global}	$\alpha_{local}(x)$	J/J_0	D/D_0	Distortion	Invalid elements
5000	1	0,4483	0,3332	[-0,225 , 5,21]	2
	$f(x)$	0,4618	0,3476	[0,298 , 3,32]	0
	$2f(x)$	0,4666	0,3520	[0,314 , 2,52]	0
	$4f(x)$	0,4767	0,3622	[0,444 , 2,26]	0
	$8f(x)$	0,4876	0,3739	[0,458, 1,89]	0
10000	1	0,5302	0,4121	[0,206 , 2,97]	0
15000	1	0,5788	0,4684	[0,489 , 2,36]	0

Table 8.3: Local regularization applied on histology data set using linear triangular elements

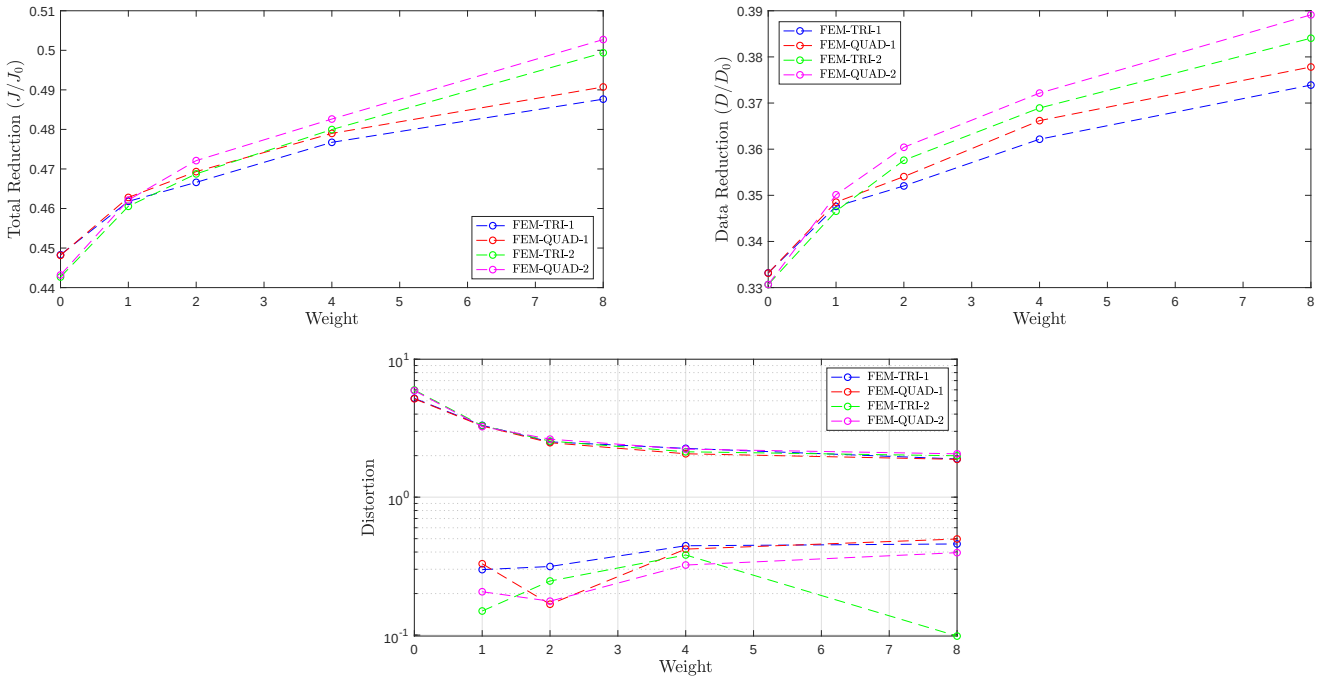


Figure 8.4: Results of local regularisation for histology data set. Top: Total reduction (left) and data reduction (Right); Bottom: Minimum and maximum distortion at discretization points;

8.2. Results of Penalty Approach

The second approach is to improve the solution of the transformation using a dedicated penalty term for volume-preserving. Adding the soft constrained, it gives the following measure

$$J(\mathbf{y}) = D(\mathbf{y}) + \alpha S(\mathbf{y} - \mathbf{y}^{ref}) + \beta P(\mathbf{y}) \tag{8.2}$$

Recall, the penalty term penalizes the contraction and expansion equally, Equation 5.47.

Searching for an optimal penalty parameter (β), different values around the optimal value of the regularization parameters (α) are considered. This section investigates the penalty parameter for the three data sets with the four different kinds of elements. As for the local regularisation, the results for the linear triangle are shown with the results of global regularisation. Furthermore, the differences between the different types of elements are visualized.

Hand

For the human hand, the penalty term decreases the distortion as can be seen in Table 8.4 and Figure 8.7. Increasing the penalty parameter results in a more important soft constraint and thus leads to less volume change. Compared to increasing the regularisation parameter, it follows that the data reduction has improved. Looking at the different kind of elements, the behavior of the quadratic quadrilateral differ from the rest. The worsening of reduction by increasing the penalty term is more significant compared to the other three elements. Also, the minimum distortion doesn't increase like the others, whereas the other three elements behave more or less the same.

α	β	J/J_0	D/D_0	Distortion	Invalid elements
1000	0	0,0462	0,0327	[0,120 , 2,44]	0
	500	0,0526	0,0349	[0,400 , 1,98]	0
	1000	0,0570	0,0367	[0,458 , 1,86]	0
	2000	0,0635	0,0395	[0,527 , 1,68]	0
	5000	0,0758	0,0468	[0,626 , 1,43]	0
2000	0	0,0544	0,0421	[0,394, 1,69]	0
3000	0	5,82%	4,92%	[0,492 , 1,53]	0

Table 8.4: Measure with the penalty term on the hand data set using linear triangular elements

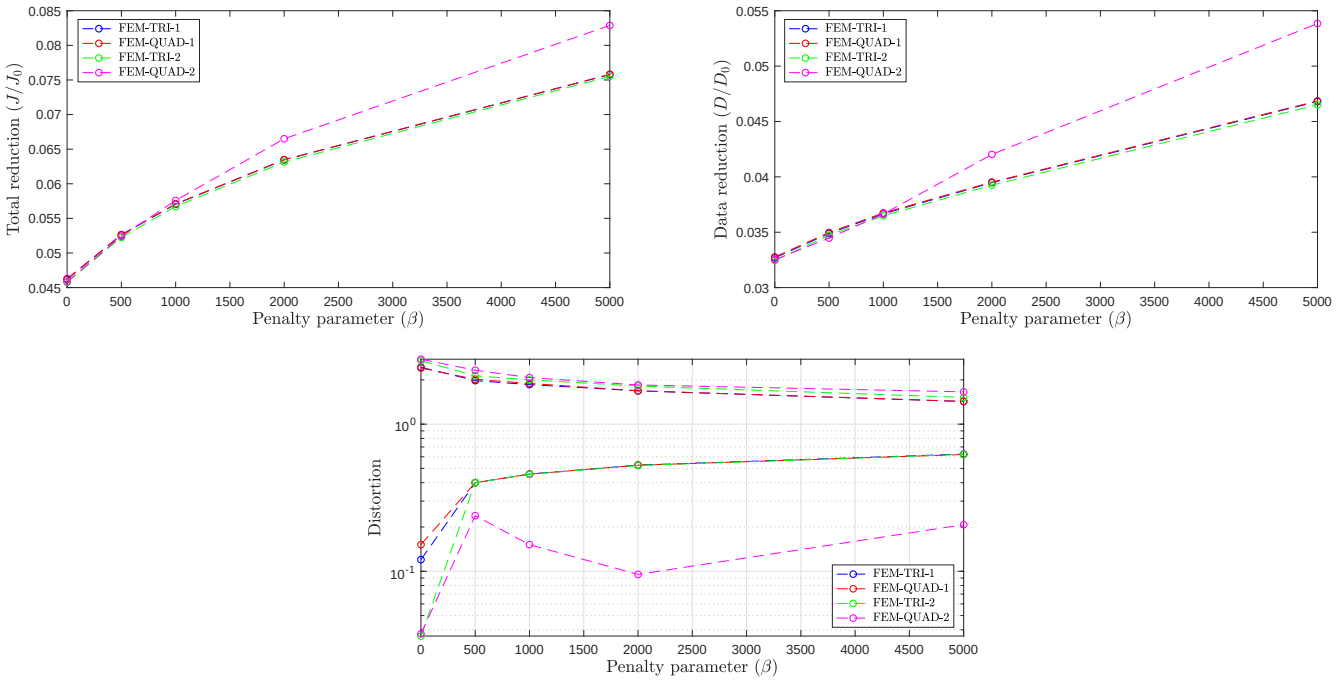


Figure 8.5: Results of the penalty approach for the hand data set. Top: Total reduction (left) and data reduction (Right); Bottom: Minimum and maximum distortion at the discretization points;

C-shaped

For the c-shaped data set, the behavior in reduction differs a lot per parameter as visualized in Figure 8.6. The minimization problem is highly sensitive for changes in the penalty parameter, and it also depends on the kind of element of how it tends to behave during the process. However, the power of

the penalty function is visible in the number of invalid elements and the reduction of the data term, Table 8.5. For the linear elements, the soft constraint results in valid transformations with a still significant reduction. With the (bi-)linear quadrilateral elements, valid transformation can be obtained, but it has poor reduction. The (bi-)quadratic elements lead to folding transformations.

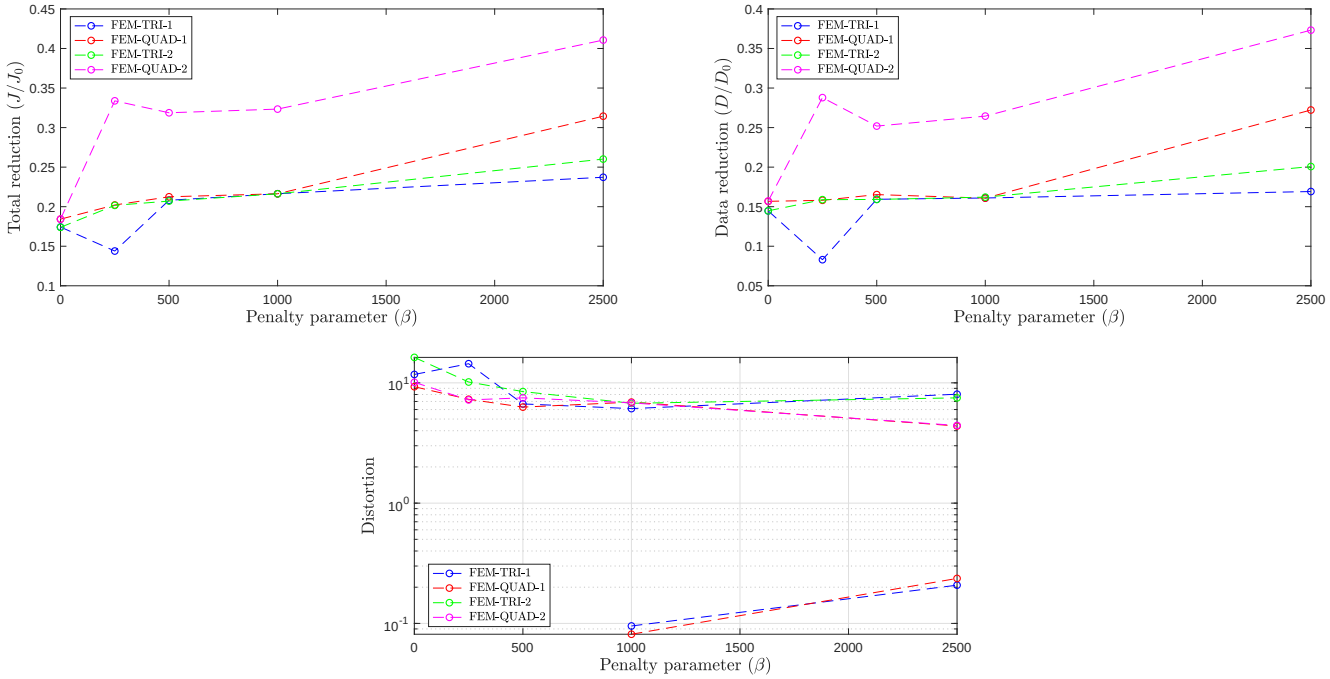


Figure 8.6: Results of the penalty approach for the c-shaped data set. Top: Total reduction (left) and data reduction (Right); Bottom: Minimum and maximum distortion at the discretization points;

α	β	J/J_0	D/D_0	Distortion	Invalid elements
500	0	0,1743	0,1447	[-3,15 , 11,7]	2273
	250	0,1447	0,0830	[-0,183 , 14,4]	79
	500	0,2081	0,1593	[-0,0754 , 6,68]	9
	1000	0,2164	0,1610	[0,0952, 6,11]	0
	2500	0,2372	0,1692	[0,208 , 8,04]	0
1000	0	0,2364	0,1960	[-4,10 , 8,23]	1697
1500	0	0,3168	0,2937	[-3,49 , 5,64]	1159

Table 8.5: Measure with the penalty term on the c-shaped data set using linear triangular elements

Histology

For the histology data set, the impact of the penalty term results in elements close to a maximum contraction of factor two, but the expansion is still significant. Compared to increasing the global regularisation parameter, the results in reduction are better for the same maximum contraction or expansion. For the bi-quadratic element, it has poor behavior in reduction. For the other elements, the differences are small.

α	β	J/J_0	D/D_0	Distortion	Invalid elements
5000	0	0,4483	0,3332	[-0,225 , 5,21]	2
	2500	0,4633	0,3441	[0,314 , 5,36]	0
	5000	0,4765	0,3545	[0,402 , 5,44]	0
	10000	0,4978	0,3730	[0,501 , 4,96]	0
	25000	0,5391	0,4148	[0,656 , 3,99]	0
10000	0	0,5302	0,4121	[0,206 , 2,97]	0
15000	0	0,5788	0,4684	[0,489 , 2,36]	0

Table 8.6: Measure with penalty term on histology data set using linear triangular elements

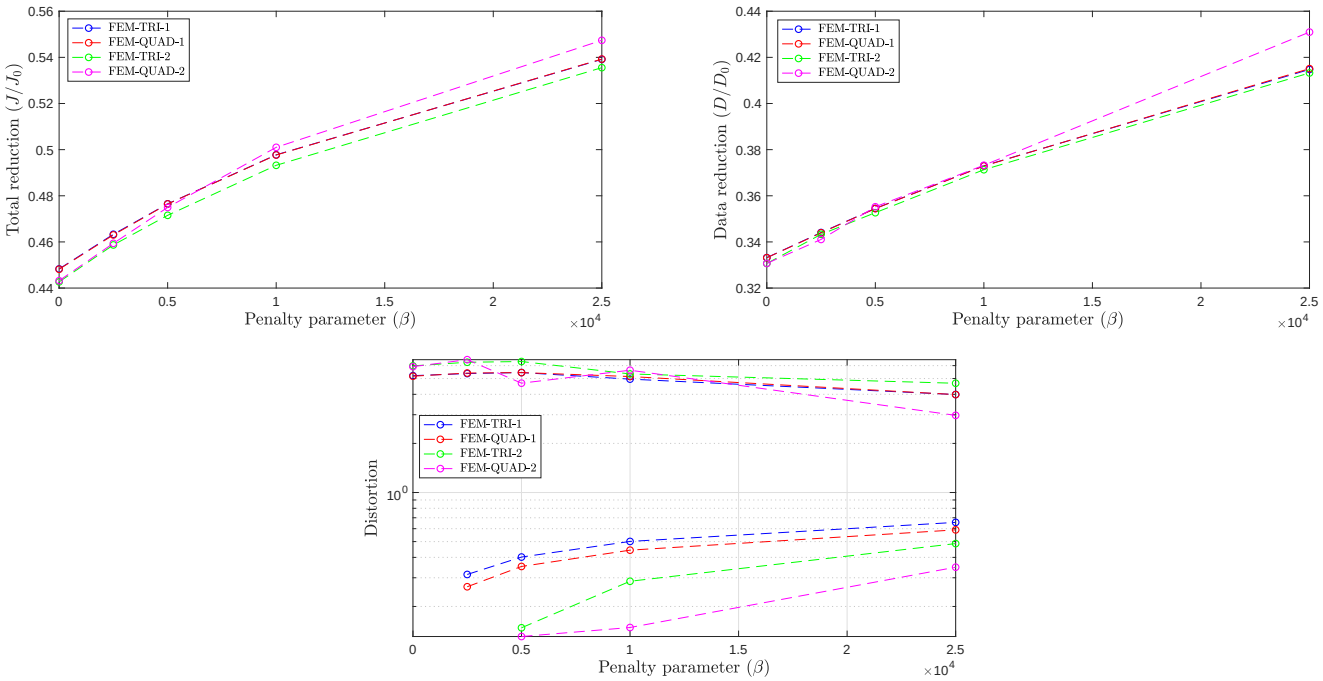


Figure 8.7: Results of penalty approach for histology data set. Top: Total reduction (left) and data reduction (Right); Bottom: Minimum and maximum distortion at discretization points;

8.3. Comparison between Global, Local and Penalty approach

This section compares the best results for the different approaches. Knowing the values of the reduction and distortion, we still don't know which approach is more suitable. The locations of the large distortions are, for example, more important than the maximum or minimum distortion value of all transformed elements. Furthermore, how similar are the transformed template and the reference image? Besides that, what is the performance in terms of time and memory with the local and penalty approach? In the first part of this section, the transformed images and distortions of the images, are discussed. In the second part of this section the results of time and memory costs are discussed.

Transformed mages and distortion

In Figure 8.8, 8.9 and 8.10 the results of the transformed images and distortion of the three approaches are shown for the linear triangular element. Table 8.7, gives the results with the optimal parameters for different approaches, which is a trade-off between distortion and reduction.

Data	Approach	J/J_0	D/D_0	Distortion	Invalid Elements
Hand	Global $(\alpha = 1000)$	0,0462	0,0327	[0,120 , 2,44]	0
	Local $(\alpha_{global} = 1000, \alpha_{local} = 4f(x))$	0,0518	0,0370	[0,473 , 1,64]	0
	Penalty $(\alpha = 1000, \beta = 1000)$	0,0570	0,0367	[0,458 , 1,86]	0
C-shaped	Global $(\alpha = 500)$	0,1743	0,1447	[-3,15 , 11,7]	2273
	Local $(\alpha_{global} = 500, \alpha_{local} = 8f(x))$	0,4780	0,4395	[-0,0513 , 2,76]	23
	Penalty $(\alpha = 500, \beta = 1000)$	0,2164	0,1610	[0,0952 , 6,11]	0
Histology	Global $(\alpha = 5000)$	0,4483	0,3332	[-0,225 , 5,21]	2
	Local $(\alpha_{global} = 5000, \alpha_{local} = 4f(x))$	0,4767	0,3622	[0,444 , 2,26]	0
	Penalty $(\alpha = 5000, \beta = 10000)$	0,4978	0,3730	[0,501 , 4,96]	0

Table 8.7: Optimal parameters for the triangular element

For the hand data set, Figure 8.8, the differences in transformation are not visible. For the distortions of the elements, large deformations are clearly visible inside the hand and around the fingers for the global approach. Whereas for the local and penalty approach, these deformations are much smaller. The differences between the local and penalty approach are minor.

The different approaches of the c-shaped data contain many differences in the transformed images, Figure 8.9. First of all, it is hard to obtain a similar transformed image as can be seen. For this data set, it has significant differences beforehand in the data and questions the use of elastic regularisation as it only holds for small deformations. The global regularisation approach leads to folding elements, and the local approach restricts the deformation too much. For the penalty term, the volume is a soft constrained, and the element can deform in every direction without volume change. For the disc image, this approach performs best.

The histology data set contains errors in the data. These defects are clearly visible in the template image. As can be seen in 8.10, the distortion in all approaches at the defects, has large deformations, and is not of importance. Besides the deformations at the defects, there is also large contraction and expansion inside the slice. The global approach leads to invalid elements, whereas for the local and penalty approach, valid transformations are obtained. The differences between local and penalty aren't significant.

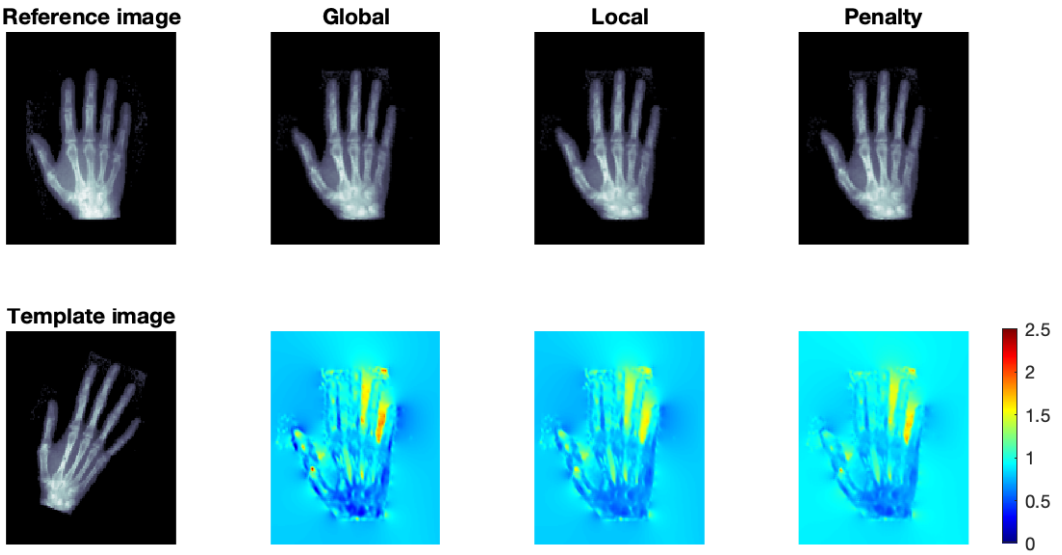


Figure 8.8: Results for hand data set; Top: Reference and transformed template images Bottom: Template image and distortion images;

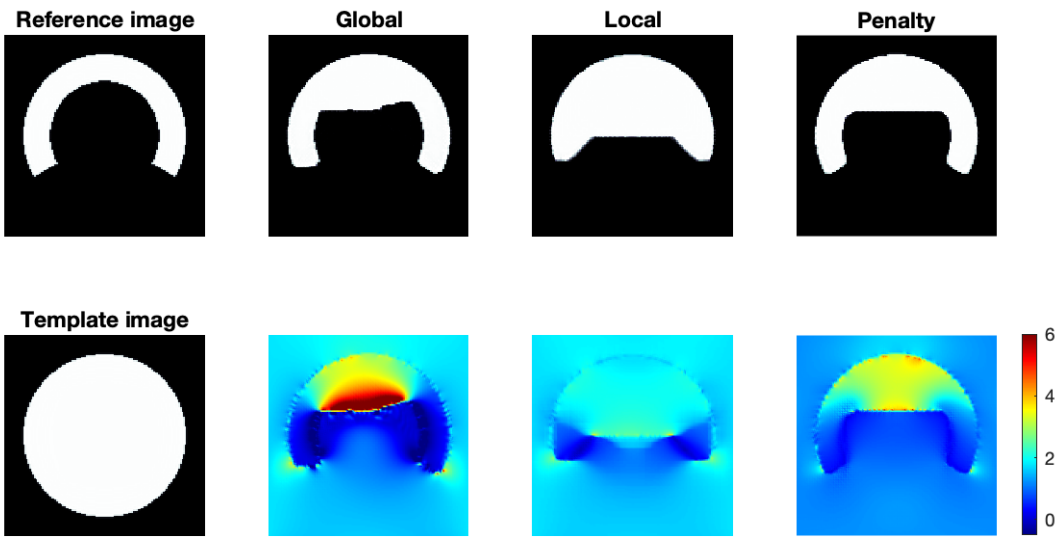


Figure 8.9: Results for disc data set; Top: Reference and transformed template images Bottom: Template image and distortion images;

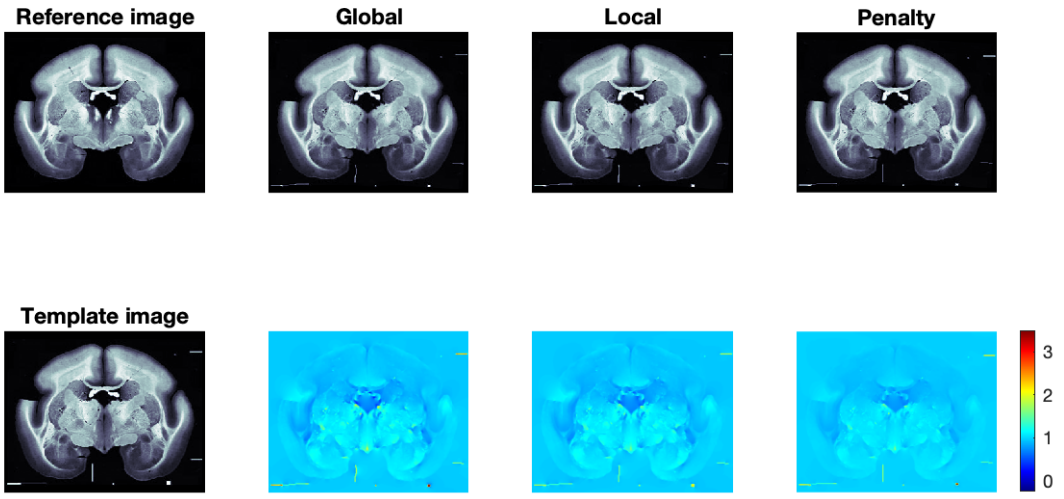


Figure 8.10: Results for histology data set; Top: Reference and transformed template images Bottom: Template image and distortion images;

Time and Memory

Time and memory are compared between the different regularisation approaches. The same solution methods are considered as in the previous chapter, which uses the benefits of a symmetric positive definite matrix.

Unfortunately, for the penalty term, it is not reasonable to apply a matrix-free approach. The penalty function is non-linear, and therefore, for every element, it results in a different element matrix. A matrix-free approach for the penalty function is thus very time-consuming. For every element, a different element matrix has to be calculated in each iteration.

The results are shown in Table 8.8. As one can see, for the global approach, the matrix-based approaches have the lowest computational cost. Due to the costs of computing the local parameters and building the penalty term, the other approaches have a significant increase in computational time. For the matrix-free methods of local regularisation, with diagonal preconditioning, a reduction of computational time is observed. For the quadrilateral elements, the reduction results in computational time in the neighborhood of the matrix-based approach. Furthermore, the differences between a global and local approach for these elements do not differ significantly.

Looking at the memory for the different regularisation approaches, Table 8.9, for the local approach, it has the same usage of memory as for the global approach. For the penalty approach, usage of memory is increased due to the density of the element matrices caused by the non-linear terms.

Histology		Global		Local		Penalty	
	Method	Reduction	time (s)	Reduction	time (s)	Reduction	time (s)
FEM-TRI-1	CD		14,4		31,0		35,3
	mbCG	J/J_0=0,4483	15,2	J/J_0=0,4767	34,7	J/J_0=0,4978	31,3
	mbCGLS		16,3		36,0		34,0
	J-mbCGLS		17,2		33,0		33,7
	mfCGLS	D/D_0=0,3332	52,4	D/D_0=0,3622	84,4	D/D_0=0,3730	x
	J-mfCGLS		54,3		58,8		x
FEM-QUAD-1	CD		14,2		44,7		42,6
	mbCG	J/J_0=0,4482	14,1	J/J_0=0,4790	36,4	J/J_0=0,4977	39,3
	mbCGLS		14,5		38,1		43,5
	J-mbCGLS		15,6		34,4		42,2
	mfCGLS	D/D_0=0,3331	33,3	D/D_0=0,3662	56,2	D/D_0=0,3729	x
	J-mfCGLS		34,3		40,4		x
FEM-TRI-2	CD		15,9		42,3		38,3
	mbCG	J/J_0=0,4427	19,5	J/J_0=0,4800	52,5	J/J_0=0,4932	41,2
	mbCGLS		20,7		54,6		46,7
	J-mbCGLS		22,3		48,5		45,8
	mfCGLS	D/D_0=0,3305	40,0	D/D_0=0,3689	76,4	D/D_0=0,3713	x
	J-mfCGLS		43,0		56,4		x
FEM-QUAD-2	CD		15,3		49,5		64,1
	mbCG	J/J_0=0,4432	19,4	J/J_0=0,4826	47,0	J/J_0=0,5010	57,8
	mbCGLS		21,2		49,5		61,8
	J-mbCGLS		19,3		41,1		58,7
	mfCGLS	D/D_0=0,3307	31,6	D/D_0=0,3722	54,5	D/D_0=0,3732	x
	J-mfCGLS		34,4		35,6		x

Table 8.8: Elapsed CPU time of different solution methods for the local and penalty approach

Histology	Global/Local		Penalty	
	Matrix-based	Matrix-free	Matrix-based	Matrix-free
FEM-TRI-1	1970688	164994	2300676	x
FEM-QUAD-1	1482634		2956036	x
FEM-TRI-2	2381440		3777540	x
FEM-QUAD-2	2632970		5252100	x

Table 8.9: Number of non-zeros stored for the matrix-based approach and length of vector stored for matrix-free approach

9

3D Reconstruction of Histological Serial Sections

In the previous chapters, two-dimensional problems are considered. As discussed in the introduction, the data is from a three-dimensional object which is sectioned into slices. With microscopic analysis of these sections, it provides us with two-dimensional images of microstructures. Due to the process of obtaining these images, the slices are deformed. Unfortunately, without Image Registration, the images can't be combined for the three-dimensional reconstruction, see Figure 9.1. Applying registration, an iteration over the slices is introduced. This method is discussed in the first section. After that, the results of the registration with the finite element approaches and finite difference method are discussed.

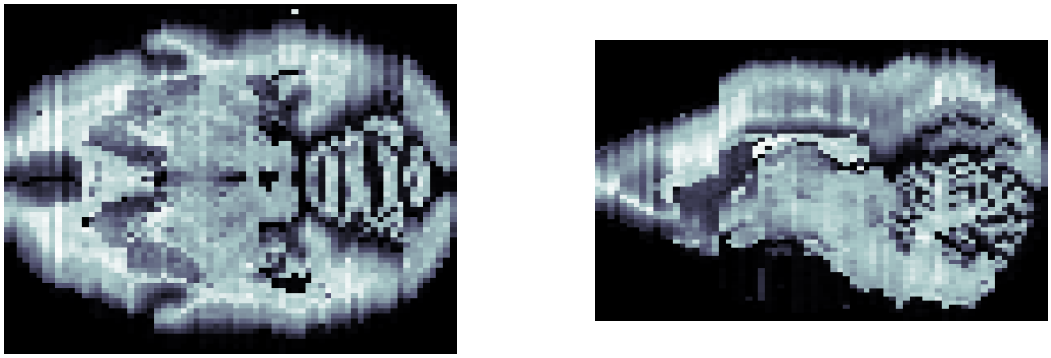


Figure 9.1: Axial and Sagittal view of the histological serial sections before registration

9.1. Reference-free Gauss-Seidel Iteration Model

The registration of histological serial sections is based on [28].

For the n serial sections, the sequence of images is denoted by $V^1, \dots V^n$. The joint measure is defined as the sum of all individual measures for the two-dimensional case and the sum of regularization and penalty over all transformation functions. It is given by

$$J(\mathbf{y}^1, \dots, \mathbf{y}^n) := \sum_{j=2}^{n} D\big(V^j(\mathbf{y}^j), V^{j-1}(\mathbf{y}^{j-1})\big) + \sum_{i=1}^{n} \Big(\alpha S(\mathbf{u}^j) + \beta P(\mathbf{y}^j)\Big) \tag{9.1}$$

Using an iterative approach over the sections, a reference-free Gauss-Seidel method is applied. The Gauss-Seidel approach uses the transformation functions $\mathbf{y}_{k+1}^1, \dots, \mathbf{y}_{k+1}^{j-1}$ which are already computed

for the next iteration of $\mathbf{y}^j$. For the other transformation functions, the transformations of the previous iteration $\mathbf{y}_k^{j+1}, \dots, \mathbf{y}_k^n$ are applied. The measure reads

$$\hat{J}(\mathbf{y}^j) := J(\mathbf{y}_{k+1}^1, \dots, \mathbf{y}_{k+1}^{j-1}, \mathbf{y}^j, \mathbf{y}_k^{j+1}, \dots, \mathbf{y}_k^n) \tag{9.2}$$

where k is the iteration number for $j = 1, \dots, n$.

From the distance measures, we know that only neighboring sections of the sequence take part in optimizing $\mathbf{y}^j$. Therefore, for the Sum-of-Squared distance measure with notation $V_k^{j+1} = V^{j+1}(\mathbf{y}_k^j)$, we have

$$\begin{aligned}\hat{D}(\mathbf{y}^j) =& D\big(V_k^{j+1}, V^j(\mathbf{y}^j)\big) + D\big(V^j(\mathbf{y}^j), V_{k+1}^{j-1}\big) + \sum_{i=1}^{j-1} D\big(V_{k+1}^i, V_{k+1}^{i-1}\big)\big) + \sum_{i=j+2}^{n} D\big(V_k^i, V_k^{i-1}\big)\big) \\ =& D\big(V_k^{j+1}, V^j(\mathbf{y}^j)\big) + D\big(V^j(\mathbf{y}^j), V_{k+1}^{j-1}\big) + r_{j_1} \\ =& 2D\big(V^j(\mathbf{y}^j), \frac{1}{2}(V_k^{j+1} + V_{k+1}^{j-1})\big) + r_{j_2}\end{aligned} \tag{9.3}$$

for $1 < j < n$.
For the measure of the first section and last section it results in

$$\hat{D}(\mathbf{y}^1) = D\big(V_k^2, V^1(\mathbf{y}^1)\big) + r_1 \tag{9.4}$$

$$\hat{D}(\mathbf{y}^n) = D\big(V^n(\mathbf{y}^1), V_{k+1}^{n-1}\big) + r_n \tag{9.5}$$

Searching for a minimizer, the remainder part can be removed from the distance measure. A new measure is introduced to find a minimizer of the function $\hat{J}(\mathbf{y}^j)$ and is defined as follows

$$J_2(\mathbf{y}^j) := D(T^j(\mathbf{y}^j), R^j) + \alpha S(\mathbf{u}^j) + \beta P(\mathbf{y}^j) \tag{9.6}$$

where

$$T^j(\mathbf{y}^j) := V^j(\mathbf{y}^j), \qquad R^j := \begin{cases} \frac{1}{2}(V_{k+1}^{j-1} + V_k^{j+1}) & \text{for } 1 < j < n \\ V_k^2 & \text{for } j = 1 \\ V_{k+1}^{n-1} & \text{for } j = n \end{cases}$$

9.2. Results of 3D reconstruction

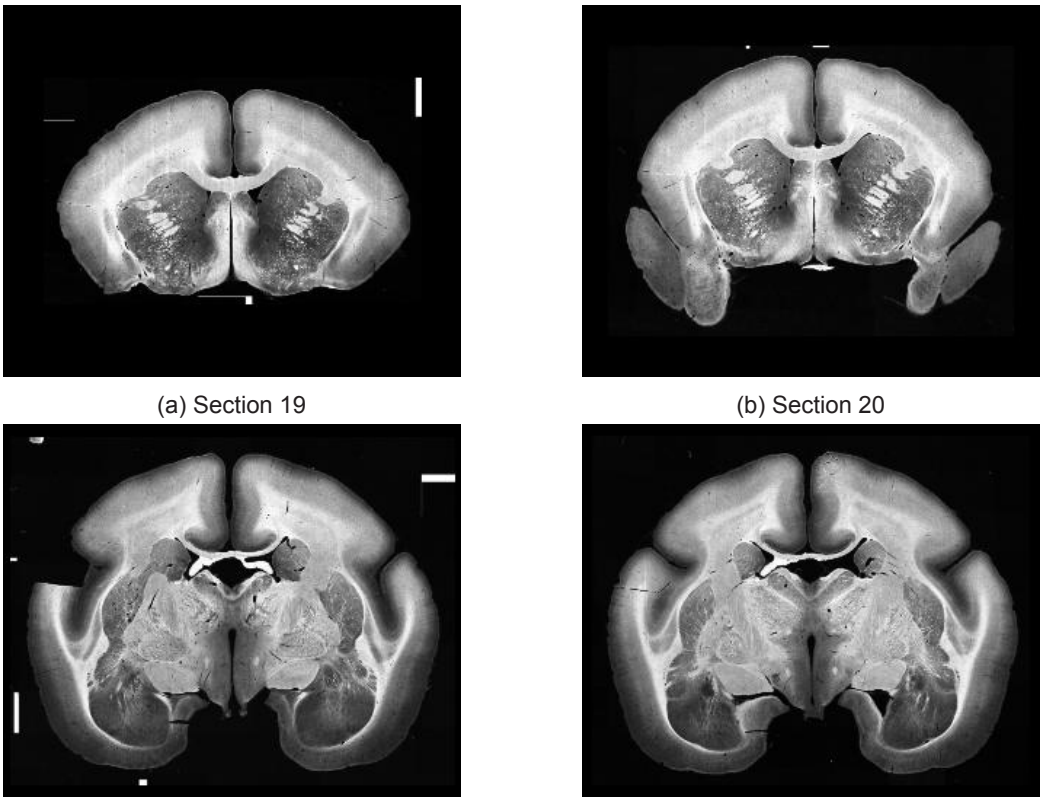
Depending on the size of the sections, the images are between 9 million and 75 million pixels. The intensities are defined in an RGB colorspace. Image registration compares single-valued intensities, and therefore, the colors are converted to the gray colorspace. Also, the amount of unknowns is too large for the methods, and the images are downscaled to 320 by 256 pixels. In the downscaling, the size of the sections in the images is preserved. Furthermore, the data consists of 555 sections, and as for the resolution of the images, it is downscaled. 69 Sections are taken into account and are equally divided from the whole data set.

Registrating the 69 sections, differences due to the deformations aren't the only discrepancies in the sections. With the downscaling of the sections and defects, the neighboring images can vary largely. The downscaling can result in large differences between the data of the adjacent sections, visualized in Figure 9.2 with sections 19 and 20. Also, defects in imaging and data occur. Errors in the background of the images, white blocks, are present. Furthermore, it contains defects such as gaps and cracks from the histology technique. The defects could lead to bad registration at those locations. An example is shown in Figure 9.2 with sections 27 and 28, where gaps and imaging errors are clearly visible.

The optimal parameters for the sample of the data in the previous chapter are applied. The parameters are shown in Table 8.7. Solving the linear system, the matrix-free diagonal preconditioned conjugate gradient method is used. A maximum of five iterations for the number of descent directions per level

is considered. For the pre-registration, a rigid transformation is chosen instead of affine in previous chapters. The size of the sections can differ significantly, as shown with the example of sections number 19 and 20. Affine transformation considers scaling of the images and therefore could lead to a bad starting point after pre-registration. Also, we know that the sections are subjected to the same imaging conditions, and the sizes of the slices are preserved in the images. Therefore, for pre-registration only rigid transformations are proposed.

This paragraph discusses the results of the three-dimensional reconstruction of the histological serial sections. First, as for the test data, a comparison is made between the different kinds of elements and the finite difference method. After that, the different proposed approaches of regularisation with FEM are treated.



(a) Section 19 (b) Section 20

(c) Section 27 (d) Section 28

Figure 9.2: Top: Example of large differences between sections; Bottom: Example of ‘gaps’ in sections;

9.2.1. Comparison of FEM and FDM

The results are visualized in Figure 9.3 and Table 9.1. The results of the minimization in the table are computed by Equation 9.1, whereas in the figure, the measure of Equation 9.6 is visualized. The large peaks before registration occur mostly due to large differences in the size, as shown with the example of sections number 19 and 20. Registrating the sections, the variations using different kinds of elements for the measure aren’t significant. For all elements, the measure is reduced to around 47% of its original value and the data term to 41% respectively for all methods.

Looking at the validity of the transformation in Figure 9.3, the finite difference method gives the least number of invalid cells. For the distortion, the maximum distortion of the elements at the discretization points does not vary significantly for the different methods. Maximum distortion is between a factor six and eight. For the lower bounds, larger differences are observed between the elements. When having large shrinkage with the finite difference approach, for the finite element approach, the (bi-)linear elements results in larger contraction of elements. For the (bi-)quadratic elements, it often leads to folding elements. As shown in the previous chapter with the sample, the large deformations of the elements

not only occur at the defects but also in the slices. Therefore, improvements in the transformation functions are needed using a higher regularisation parameter or a different approach.

Method	J/J_0	D/D_0
FDM	0,4674	0,4040
FEM-TRI-1	0,4705	0,4079
FEM-QUAD-1	0,4705	0,4080
FEM-TRI-2	0,4677	0,4060
FEM-QUAD-2	0,4683	0,4063

Table 9.1: Minimization results of measure and data term of histological serial sections

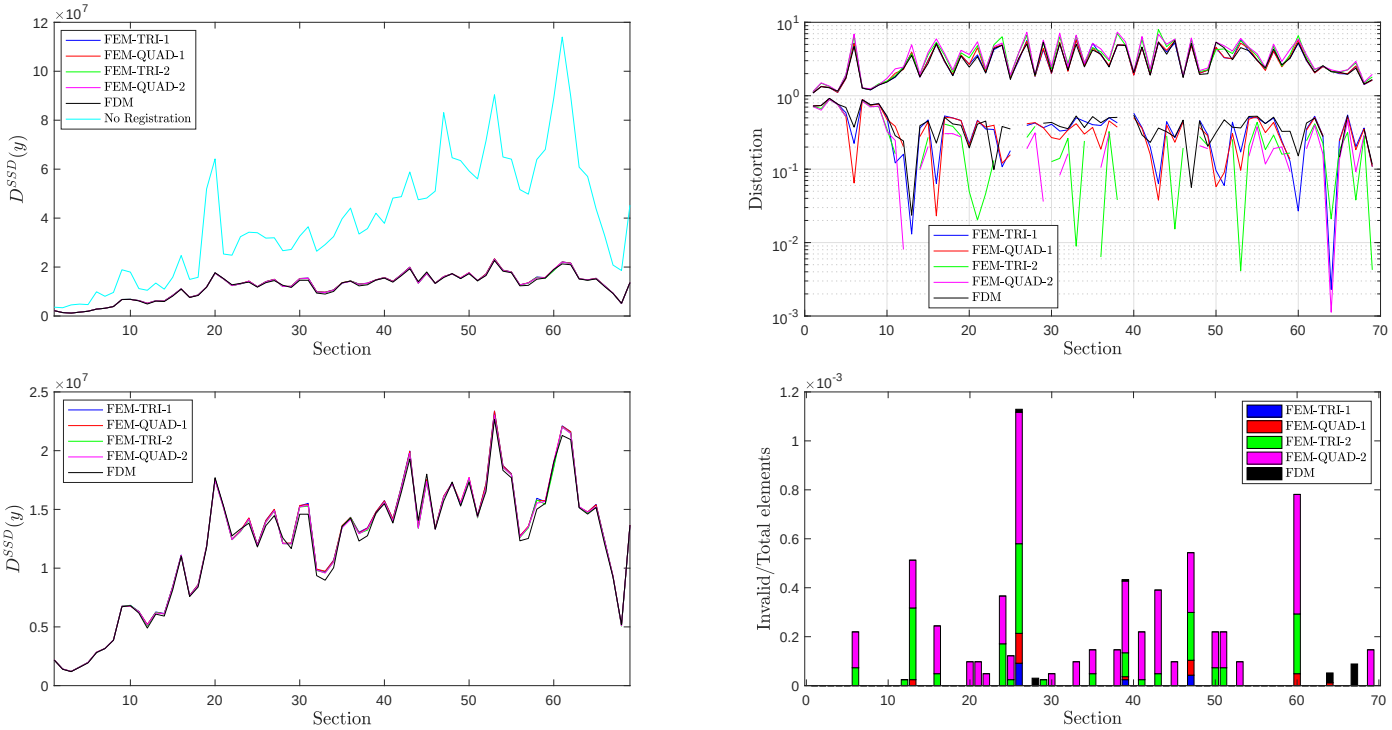


Figure 9.3: Left: Results of (zoomed in) data measure; Right: Top: Minimum and maximum distortion; Bottom: validity of elements

9.2.2. Regularisaton approaches with FEM

The different finite element proposed approaches are applied to the data. Table 9.2 gives the results of the minimization of the measure and data term. Adding a penalty or applying local approach results as expected in a worsened reduction. Looking at the validity of the elements, which is visualized in Figures 9.5, 9.6, 9.7 and 9.8, the transformations are improved. Only the penalty approach for the bi-quadratic elements contains invalid elements. For all elements, using a local or penalty approach results in better reduction and validity, then increasing the global parameter. The local approach still has some large shrinkage for some elements, but in all cases it is valid. The maximum expansion for the local approach is reduced to a factor between two and three. For the penalty approach, it is the other way around. The shrinkage is reduced to a maximum around factor two, whereas the expansion is still large. When visualizing the transformed sections in an axial and sagittal view, the reduction in differences is clearly visible, see Figure 9.4.

	FEM-TRI-1		FEM-QUAD-1		FEM-TRI-2		FEM-QUAD-2	
Approach	J/J_0	D/D_0	J/J_0	D/D_0	J/J_0	D/D_0	J/J_0	D/D_0
Global	0,4705	0,4079	0,4705	0,4080	0,4677	0,4060	0,4683	0,4063
Local	0,5000	0,4394	0,5019	0,4416	0,5043	0,4455	0,5076	0,4490
Penalty	0,5174	0,4595	0,5174	0,4595	0,5154	0,4579	0,5321	0,4743
2·Global	0,5267	0,4948	0,5267	0,4948	0,5252	0,4935	0,5255	0,4937

Table 9.2: Minimization results of the different regularisation approaches

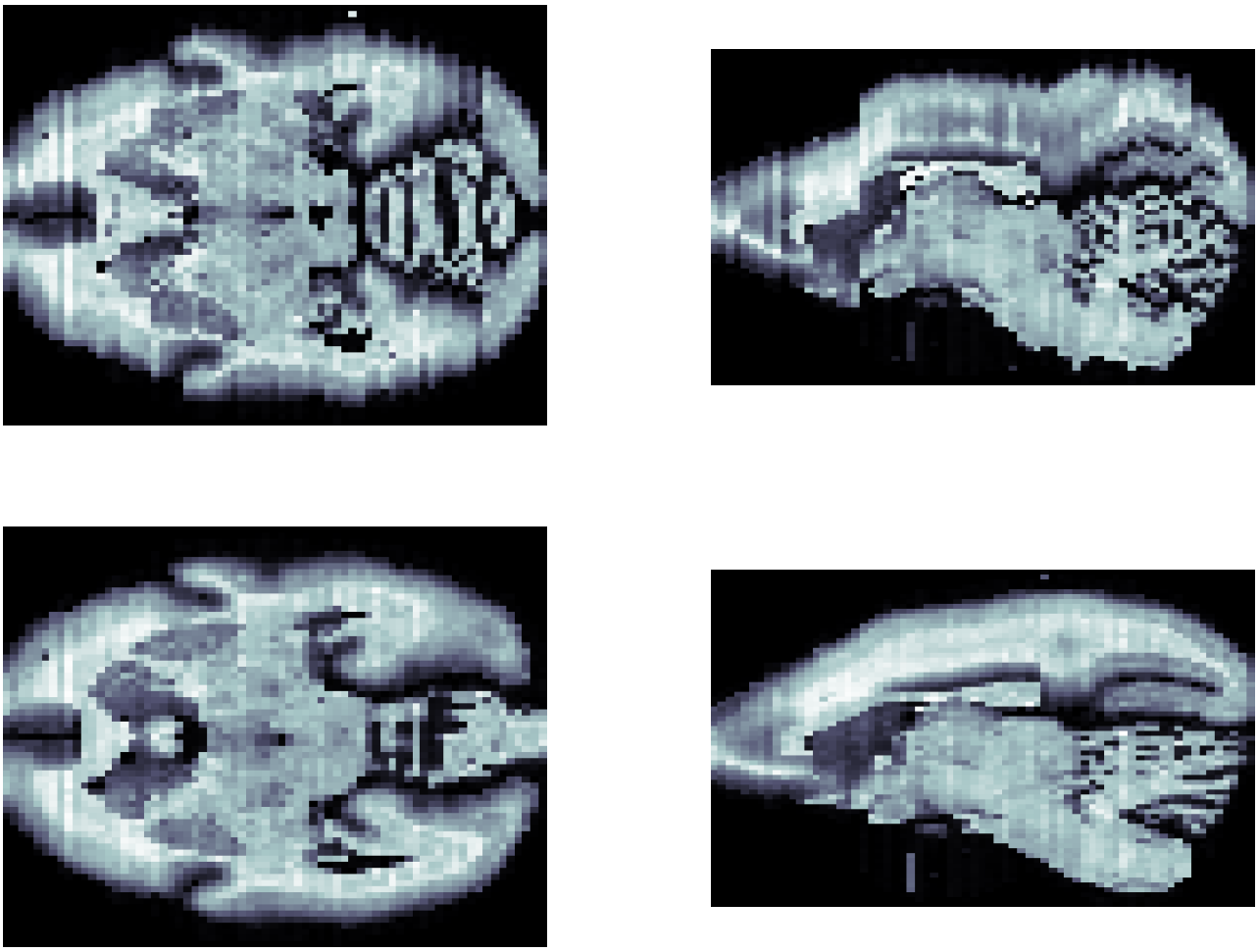


Figure 9.4: 3D reconstruction of histological serial sections; Top: before registration, Bottom: Registration with local regularisation using linear triangular elements;

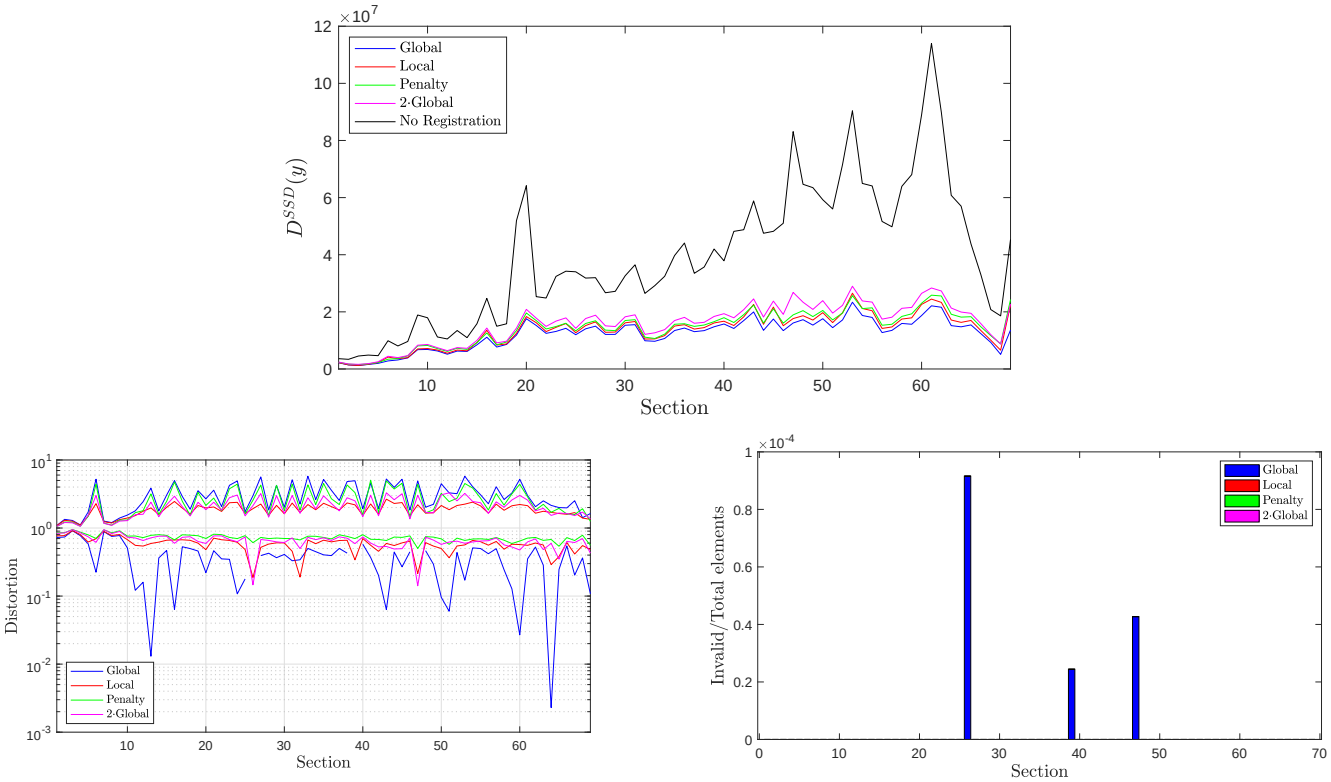


Figure 9.5: Results with linear triangular elements. Top: reduction of data measure; Bottom: Left: minimum and maximum distortion Right: validity of the elements

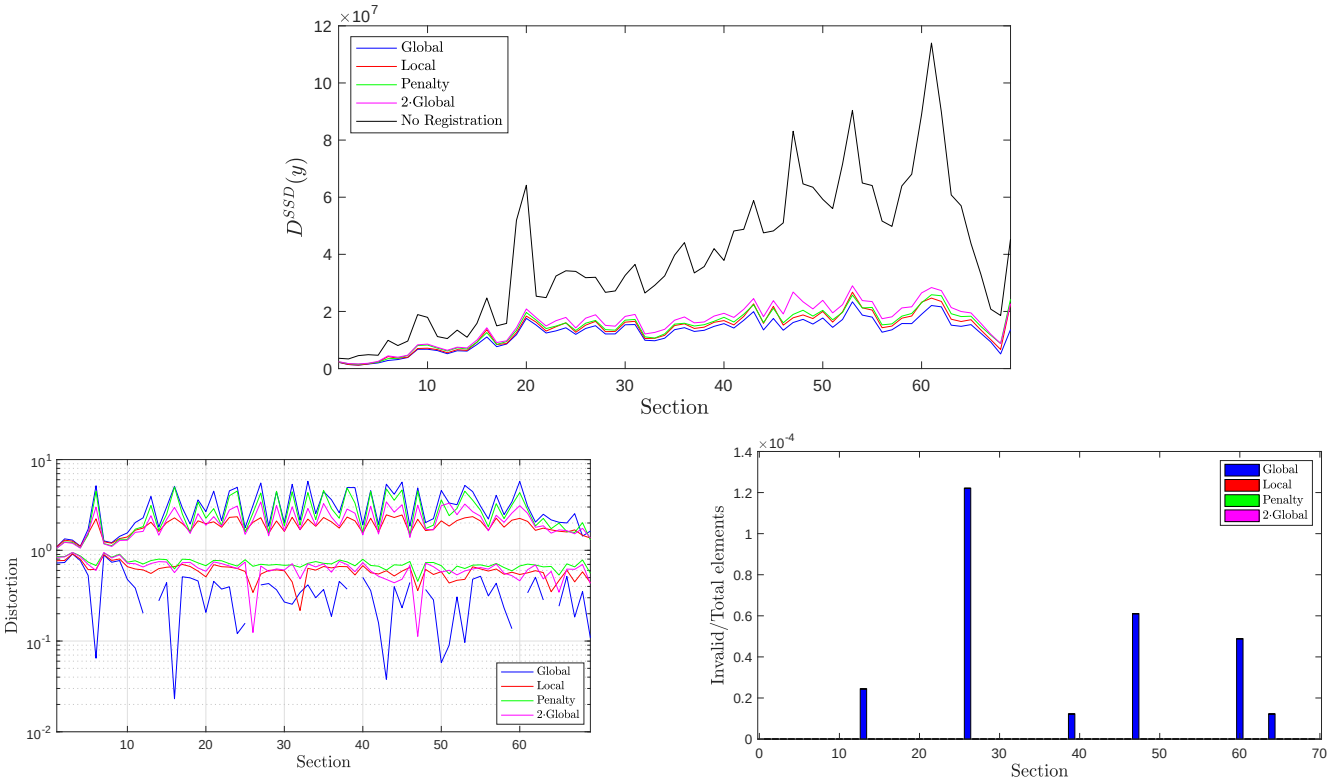


Figure 9.6: Results with bi-linear quadrilateral elements. Top: reduction of data measure; Bottom: Left: minimum and maximum distortion Right: validity of the elements

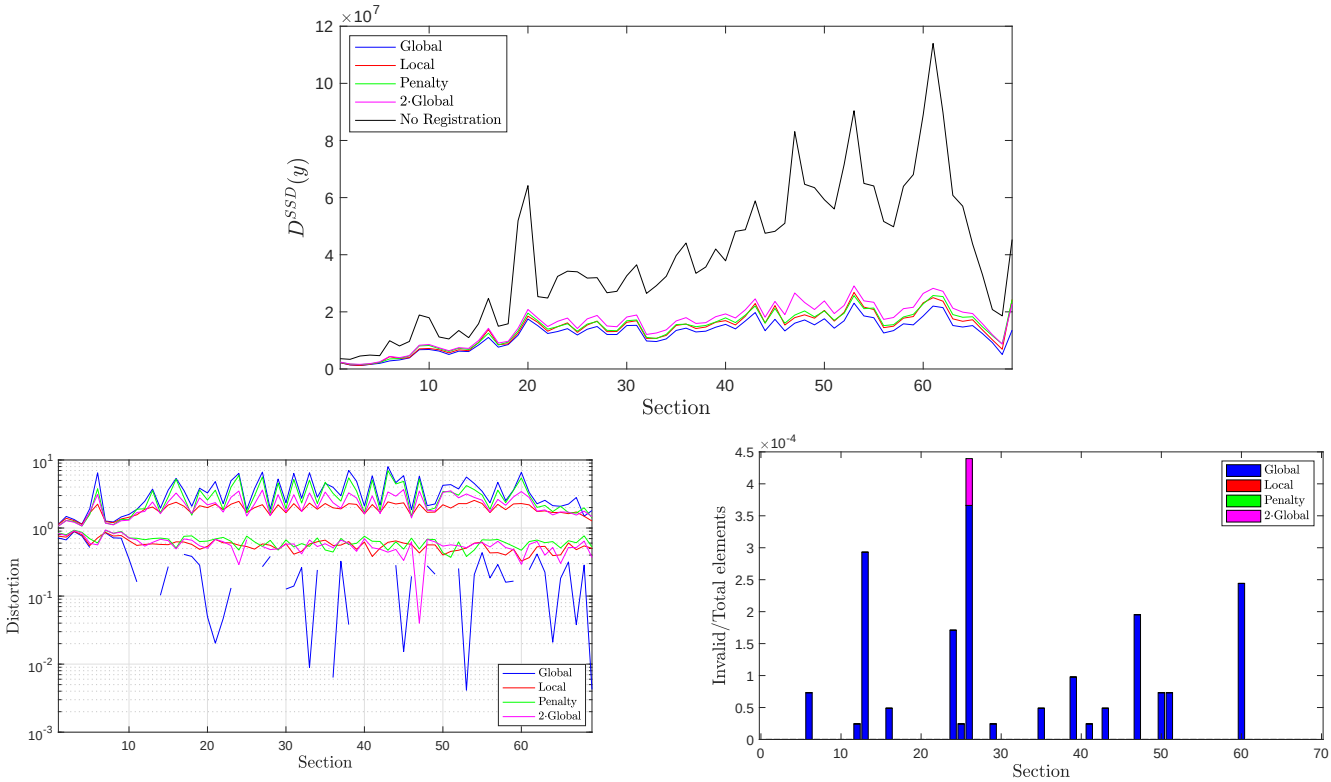


Figure 9.7: Results with quadratic triangular elements. Top: reduction of data measure; Bottom: Left: minimum and maximum distortion Right: validity of the elements

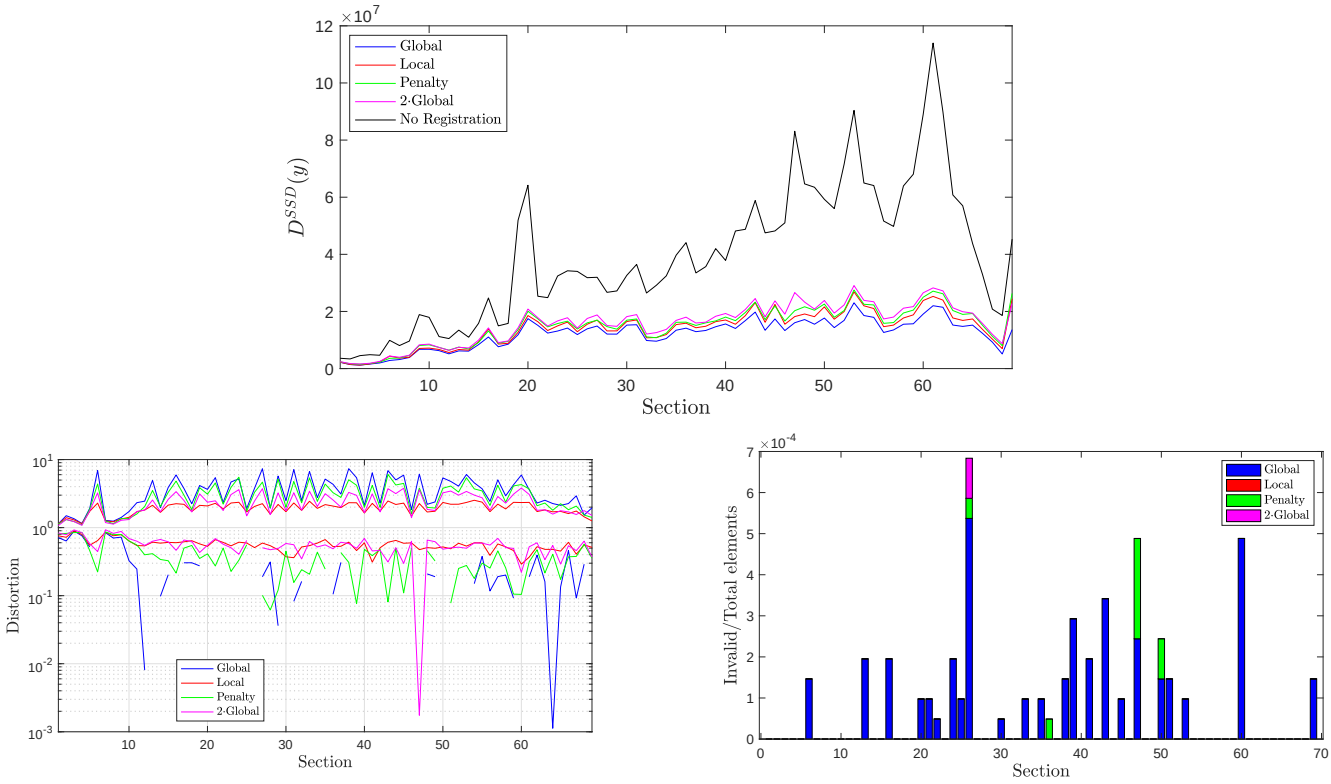


Figure 9.8: Results with bi-quadratic quadrilateral elements. Top: reduction of data measure; Bottom: Left: minimum and maximum distortion Right: validity of the elements

10

Conclusion & Discussion

In this thesis, the finite element approach for elastic regularisation is investigated. Four different element types are evaluated, improvements are made on the regularisation, and a matrix-free approach is constructed. The methods are applied to different test cases and the histological serial sections. This chapter summarizes and discusses the outcomes of the research.

The four different kinds of elements are implemented and evaluated to determine which element yields the best result. Therefore, the reduction of the differences in the images, the validity of the transformation function is compared for three different data sets. For the non-artificial data sets, no significant differences are observed in the reduction among the different kinds of elements, and the results are similar to the finite difference method. For the artificial data set, the finite element method shows improvements, but it results in a substantially high ratio of folding elements. So, with this data set, elastic regularisation does not seem to be well-suited. The validity of the transformation function is determined by checking the elements for folding. Overall, for the finite element method, the linear elements outperforms the bi-quadratic elements for validity of the elements. When comparing it to the interpolated transformation of the finite difference method, the finite difference method yields the lowest ratio. For the computational costs of the different methods, the finite difference method is the fastest. This is mainly due to the assembly of the elastic regularisation matrix. The finite difference method is a very efficient method for structured grids. For the different kinds of elements, registration with the (bi-)linear elements has the best performance, because they yield the sparsest matrix.

Using the benefits of a finite element implementation, two different approaches with elastic regularisation are considered. In both approaches, the goal is to penalize folding elements, which are physically unlikely, in the transformations. The first approach is by increasing the stiffness of the elements during the iteration process when elements tend to be heavily deformed. The second approach is adding a soft constraint on the contraction and expansion of the elements. This method is not restricted to only elastic bodies, as can be seen in the results of the data sets. Both methods show improvements in the distortion of the elements and reducing the number of folded elements significantly to validly transformed elements. Moreover, it results in reasonable transformation functions for which the images are accurately aligned. The computational costs for both approaches are large, as it results in longer computational time between a factor two and four depending on the element.

After that, a matrix-free approach of the elastic regularisation is implemented. Using iterative methods to solve linear systems, assembling of the elastic regularisation matrix is not required. A memory-saving method is not to store the matrix, but the matrix-vector product per iteration. Due to limitations of storage, for large systems, it is very attractive. For global regularisation and local regularisation, the matrix-free method is implemented in the (preconditioned-)CGLS method. A significant reduction in memory is obtained, but the computational time is increased as per iteration an elements-wise matrix-vector needs to be calculated. The increase depends on the elements, as for the high order elements, fewer elements-wise , matrix-vector products are computed. With diagonal preconditioning for the local regularisation, improvements are made in computational time. For the matrix-free approach of the

quadrilateral elements, the computing time is reduced to similar computational costs as for the matrix-based approach. Besides that, the computational costs for the quadrilateral elements are similar to the global regularisation approach.

The four kinds of elements and the two approaches to penalize are applied to the histological data. As for the test data, the linear elements yield the best results, but the differences are small compared to the other kinds of elements. Without the penalization of the distortion, a high regularisation parameter need to be chosen for valid transformations. With the suggested approaches, significant higher reduction of the differences between the images can be obtained while the distortion of the elements is limited.

In conclusion, one the one hand, the finite element method does not improve the optimization problem, as it takes longer to assemble and to solve the system compared to the finite difference method. However, by implementing the finite element method, there is a robust way of checking the validity of the transformation function and no assumptions need to be made for the finite difference method. On the other hand, having a finite element implementation, parts of the domain can be easily penalized or restricted. Two powerful approaches are implemented to steer the elements in a non-folding, less distorted transformation. The reduction of the differences between the images for these approaches is significantly better compared to increasing the regularization parameter. Furthermore, a memory-efficient way of solving the linear system is implemented. Comparing the different kinds of elements, the (bi-)linear elements yields the best results in reduction and validity. Moreover, in terms of computational costs, the bi-linear element is preferred.

10.1. Discussion

From the discretized measure of the differences between the images, it follows that the discretization of the distance measure is based on a cell-centered image grid. So, interpolation is required to the nodal grid of the finite element approach. Therefore, using the finite element discretization in the distance measure could improve the accuracy of the matrices in the optimization procedure.

For the iterative linear solution methods, Krylov subspace methods are considered. The matrix-free approach causes a significant increase in computational time. Improvements in the reduction of computational costs by using preconditioning are, therefore, of importance. By having a finite element approach of the regularisation matrix, the approximation of the regularisation by Hughes is recommended as preconditioning. Hughes proposed an Element-by-Element approach to approximate the matrix [19].

A multi-level approach is applied to obtain and improve the transformation function by going from a coarse to a fine grid. Therefore, multigrid is an attractive approach to solve the linear system. For the bi-linear elements, a multigrid solution method is previously investigated [23], which yields a fast linear solution method. Therefore, using multigrid or multigrid preconditioned Krylov-subspace methods is recommended to implement for the finite elements. Moreover, the proposed Element-by-Element approach of Hughes could be used as smoother for the multigrid method.

With the multi-level approach, the refinement of the grid is applied to the whole domain. The slices are a (small) part of the whole image. Refinement on the whole grid is not therefore necessary to measure the deformations. One of the benefits of the finite element approach is freedom in the discretization grid. To reduce memory usage and computational time, it is worthwhile to apply refinement on only specific subdomains.

A

BVP to Minimization

Theorem A.1 *Let L be a linear, symmetric, positive differential operator defined over a space* Σ *and let*

$$L(u) = \mathbf{g}. \tag{A.1}$$

Then the solution u minimizes the functional

$$I(u) = \int_{\Omega} \{\frac{1}{2} uL(u) - uf\} d\Omega, \text{ over the space } \Sigma. \tag{A.2}$$

On the other hand if u minimizes (A.2) then u satisfies (A.1).

Considering a two dimensional domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$. The boundary value problem of linear elasticity is

$$\begin{cases} -\mu\Delta\mathbf{u} - (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) = 0, & \mathbf{x} \in \Omega \\ \sigma_{i:} \cdot \mathbf{n} = \mu\Big(\nabla u_i + \dfrac{\partial \mathbf{u}}{\partial x_i}\Big) \cdot \mathbf{n} + \lambda(\nabla\cdot\mathbf{u}) n_i = 0 & \mathbf{x} \in \partial\Omega \end{cases} \tag{A.3}$$

The solution must be found in the space $\Sigma(\mathbf{u}) = \{\mathbf{u} \text{ is sufficiently smooth}\}$. Say $\mathbf{g} = 0$ and $L(\mathbf{u}) = -\mu\Delta\mathbf{u} - (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u})$. Using theorem A.1, L has to be linear, symmetric and positive. The is appendix verifies the conditions for the Navier-Lamé equations.

Linear

The linear condition is

$$L(\alpha\mathbf{u} + \beta\mathbf{v}) = \alpha L(\mathbf{u}) + \beta L(\mathbf{v}) \tag{A.4}$$

where α, β are constants and $\mathbf{u}, \mathbf{v} \in \Sigma$.

$$\begin{aligned} L(\alpha\mathbf{u} + \beta\mathbf{v}) &= -\mu\Delta(\alpha\mathbf{u} + \beta\mathbf{v}) + (\lambda+\mu)\nabla(\nabla\cdot(\alpha\mathbf{u} + \beta\mathbf{v}) \\ &= -\alpha\mu\Delta\mathbf{u} - \beta\mu\Delta\mathbf{v} + \alpha(\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) + \beta(\lambda+\mu)\nabla(\nabla\cdot\mathbf{v}) \\ &= \alpha\big(-\mu\Delta\mathbf{u} + (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u})\big) + \beta\big(-\mu\Delta\mathbf{v} + (\lambda+\mu)\nabla(\nabla\cdot\mathbf{v})\big) \\ &= \alpha L(\mathbf{u}) + \beta L(\mathbf{v}) \end{aligned}$$

Symmetric

For the symmetric property, it has to satisfy

$$\int_{\Omega} \mathbf{u} L(\mathbf{v}) d\Omega = \int_{\Omega} \mathbf{v} L(\mathbf{u}) d\Omega \qquad \forall \mathbf{u}, \mathbf{v} \in \Sigma \tag{A.5}$$

Assume $\mathbf{u}, \mathbf{v} \in \Sigma$, then

$$\begin{aligned}\int_\Omega \mathbf{u}L(\mathbf{v})d\Omega &= \int_\Omega -\mathbf{u}\cdot(\mu\Delta\mathbf{v} - (\mu+\lambda)(\nabla(\nabla\cdot\mathbf{v})))d\Omega \\ &= \int_\Omega -\mu\mathbf{u}\cdot(\Delta\mathbf{v} + \nabla(\nabla\cdot\mathbf{v}) - \lambda\mathbf{u}\cdot(\nabla(\nabla\cdot\mathbf{v}))d\Omega\end{aligned}$$

Usage of Integration By Parts and Theorem of Gauss

$$\begin{aligned}\int_\Omega u_k\frac{\partial}{\partial x_k}(\nabla\cdot\mathbf{v}d\Omega &= \int_\Omega \frac{\partial}{\partial x_k}(u_k\nabla\cdot\mathbf{v})d\Omega - \int_\Omega \frac{\partial u_k}{\partial x_k}(\nabla\cdot\mathbf{v})d\Omega \\ &= \int_{\partial\Omega}(u_k\nabla\cdot\mathbf{v})n_k d\Gamma - \int_\Omega \frac{\partial u_k}{\partial x_k}(\nabla\cdot\mathbf{v})d\Omega\end{aligned}$$

$$\begin{aligned}\int_\Omega u_k\Big(\Delta v_k + \frac{\partial}{\partial x_k}(\nabla\cdot\mathbf{v})\Big)d\Omega &= \int_\Omega \nabla\cdot\Big(u_k(\nabla v_k + \frac{\partial\mathbf{v}}{\partial x_k})\Big)d\Omega - \int_\Omega \nabla u_k\cdot\Big(\nabla v_k + \frac{\partial\mathbf{v}}{\partial x_k}\Big)d\Omega \\ &= \int_{\partial\Omega}\Big(u_k(\nabla v_k + \frac{\partial\mathbf{v}}{\partial x_k})\Big)\cdot\mathbf{n}d\Gamma - \int_\Omega \nabla u_k\cdot\Big(\nabla v_k + \frac{\partial\mathbf{v}}{\partial x_k}\Big)d\Omega\end{aligned}$$

$$\begin{aligned}\int_\Omega \mathbf{u}L(\mathbf{v})d\Omega = \int_{\partial\Omega}\mu\sum_{k=1}^{2}\Big(u_k(\nabla v_k + \frac{\partial\mathbf{v}}{\partial x_k})\Big)\cdot\mathbf{n} + \lambda(\nabla\cdot\mathbf{v})(\mathbf{u}\cdot\mathbf{n})d\Gamma + \\ \int_\Omega \mu\Big(\nabla u_1\cdot\nabla v_1 + \nabla u_2\cdot\nabla v_2 + \nabla u_1\frac{\partial\mathbf{v}}{\partial x_1} + \nabla u_2\frac{\partial\mathbf{v}}{\partial x_2}\Big) + \lambda(\nabla\cdot\mathbf{u})(\nabla\cdot\mathbf{v})d\Omega\end{aligned}$$

The boundary integral can be rewritten as

$$\begin{aligned}\int_{\partial\Omega}\mu\sum_{k=1}^{2}\Big(u_k(\nabla v_k + \frac{\partial\mathbf{v}}{\partial x_k})\Big)\cdot\mathbf{n} + \lambda(\nabla\cdot\mathbf{v})(\mathbf{u}\cdot\mathbf{n})d\Gamma &= \int_{\partial\Omega}u_1\Big(\mu\Big(\nabla v_1 + \frac{\partial\mathbf{v}}{\partial x_1}\Big)\cdot\mathbf{n} + \lambda(\nabla\cdot\mathbf{v})n_1\Big) + \\ &\quad u_2\Big(\mu\Big(\nabla v_2 + \frac{\partial\mathbf{v}}{\partial x_2}\Big)\cdot\mathbf{n} + \lambda(\nabla\cdot\mathbf{v})n_2\Big)d\Gamma \\ &= \int_{\partial\Omega}u_1(\sigma_{1:}\cdot\mathbf{n}) + u_2(\sigma_{2:}\cdot\mathbf{n})d\Gamma = 0\end{aligned}$$

This results in

$$\begin{aligned}\int_\Omega \mathbf{u}L(\mathbf{v})d\Omega &= \int_\Omega \mu\Big(\nabla u_1\cdot\nabla v_1 + \nabla u_2\cdot\nabla v_2 + \nabla u_1\frac{\partial\mathbf{v}}{\partial x_1} + \nabla u_2\frac{\partial\mathbf{v}}{\partial x_2}\Big) + \lambda(\nabla\cdot\mathbf{u})(\nabla\cdot\mathbf{v})d\Omega \\ &= \int_\Omega -\mathbf{v}\cdot(\mu\Delta\mathbf{u} - (\mu+\lambda)(\nabla(\nabla\cdot\mathbf{u})))d\Omega + \\ &\quad \int_{\partial\Omega}v_1(\sigma_{1:}\cdot\mathbf{n}) + v_2(\sigma_{2:}\cdot\mathbf{n})d\Gamma \\ &= \int_\Omega \mathbf{v}L(\mathbf{u})d\Omega\end{aligned}$$

and thus satisfies the symmetric property.

Positiveness

For positiveness, the operator L has to satisfy the following equation

$$\int_\Omega \mathbf{u}L(\mathbf{u})d\Omega \geq 0 \tag{A.6}$$

Assume $\mathbf{u} \in \Sigma$

$$\int_\Omega \mathbf{u}L(\mathbf{v})d\Omega = \int_\Omega -\mathbf{u} \cdot (\mu\Delta\mathbf{u} - (\mu + \lambda)(\nabla(\nabla \cdot \mathbf{u})))d\Omega$$

Applying Integration by Parts, Theorem of Gauss and the boundary conditions as showed for the symmetric property, it results in

$$\begin{aligned}\int_\Omega \mathbf{u}L(\mathbf{u})d\Omega &= \int_\Omega \mu\Big(\nabla u_1 \cdot \nabla u_1 + \nabla u_2 \cdot \nabla u_2 + \nabla u_1 \frac{\partial \mathbf{u}}{\partial x_1} + \nabla u_2 \frac{\partial \mathbf{u}}{\partial x_2}\Big) + \lambda(\nabla \cdot \mathbf{u})(\nabla \cdot \mathbf{u})d\Omega \\ &= \int_\Omega \mu\Big(2(\frac{\partial u_1}{\partial x_1})^2 + 2(\frac{\partial u_2}{\partial x_2})^2 + (\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1})^2\Big) + \lambda(\nabla \cdot \mathbf{u})^2 d\Omega \geq 0\end{aligned}$$

Knowing the Navier-Lamé parameters $\mu, \lambda \geq 0$, the integrand consist only of positive quadratic parts and therefore, the integral is greater or equal then zero.

Minimization

All conditions are satisfied and according to theorem A.1, the corresponding minimization problem of the boundary value problem is

$$\int_\Omega \frac{1}{2}\mathbf{u}L(\mathbf{u})d\Omega = \frac{1}{2}\int_\Omega \mu\Big(2(\frac{\partial u_1}{\partial x_1})^2 + 2(\frac{\partial u_2}{\partial x_2})^2 + (\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1})^2\Big) + \lambda(\nabla \cdot \mathbf{u})^2 d\Omega \quad \text{over the space } \Sigma \tag{A.7}$$

B

Element Matrices of Elastic Potential Energy

Minimal Elastic Potential Energy

Applying the Finite Element Approach to the linear elasticity equations. Recall from chapter 3, the minimal elastic potential energy is given by

$$\min_{\mathbf{u}\in\Sigma} \frac{1}{2}\int_{\Omega} \lambda(\nabla\cdot\mathbf{u})^2 + \mu\Big(2(\frac{\partial u_1}{\partial x_1})^2 + 2(\frac{\partial u_2}{\partial x_2})^2 + (\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1})^2\Big)d\Omega \tag{B.1}$$
$$\Sigma(\mathbf{u}) := \{\mathbf{u} \text{ is sufficiently smooth}\}$$

As for the elastic regularisation in FAIR, the elastic potential energy is approximated by

$$\min_{\mathbf{u}\in\Sigma} \frac{1}{2}\int_{\Omega} (\mu+\lambda)(\nabla\cdot\mathbf{u})^2 + \mu(\nabla\mathbf{u}\cdot\nabla\mathbf{u})d\Omega \tag{B.2}$$
$$\Sigma(\mathbf{u}) := \{\mathbf{u} \text{ is sufficiently smooth}\}$$

Approximating the solution by a finite set of known basis functions in the function space Σ, the solution is written as

$$u_1(\mathbf{x}) \simeq \sum_{j=1}^{m} a_1^j \varphi_j(\mathbf{x}) \tag{B.3}$$

$$u_2(\mathbf{x}) \simeq \sum_{j=1}^{m} a_2^j \varphi_j(\mathbf{x}) \tag{B.4}$$

Applying Ritz's method to the minimal elastic potential energy, the term read

$$\begin{aligned}\mathcal{S}[a_1^1,\dots,a_1^m,a_2^1,,\dots,a_2^m] = \frac{1}{2}\int_{\Omega} (2\mu+\lambda)\left[\Big(\frac{\partial(\sum a_1^j\varphi_j)}{\partial x_1}\Big)^2 + \Big(\frac{\partial(\sum a_2^j\varphi_j)}{\partial x_2}\Big)^2\right] + 2\lambda\frac{\partial(\sum a_1^j\varphi_j)}{\partial x_1}\frac{\partial(\sum a_2^j\varphi_j)}{\partial x_2} \\ + \mu\left[\Big(\frac{\partial(\sum a_1^j\varphi_j)}{\partial x_2}\Big)^2 + 2\frac{\partial(\sum a_1^j\varphi_j)}{\partial x_1}\frac{\partial(\sum a_2^j\varphi_j)}{\partial x_2} + \Big(\frac{\partial(\sum a_2^j\varphi_j)}{\partial x_1}\Big)^2\right]d\Omega\end{aligned} \tag{B.5}$$

Taking the first derivatives to $a_1^1, \dots, a_1^m, a_2^1, \dots, a_2^m$

$$\begin{cases} \sum_{j=1}^{m} a_1^j \int_\Omega (2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + \mu\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}d\Omega + \sum_{j=1}^{m} a_2^j \int_\Omega (\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_2}d\Omega = 0 \\ \qquad\qquad \text{for } i = 1, 2, \dots, m \\ \sum_{j=1}^{m} a_1^j \int_\Omega (\mu+\lambda)\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_2}d\Omega + \sum_{j=1}^{m} a_2^j \int_\Omega \mu\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + (2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}d\Omega = 0 \\ \qquad\qquad \text{for } i = 1, 2, \dots, m \end{cases} \tag{B.6}$$

Which are linear equations and can be written as

$$\nabla S = R\mathbf{u}, \quad \mathbf{u} = \begin{bmatrix} u_1^1 \\ \vdots \\ u_1^m \\ u_2^1 \\ \vdots \\ u_2^m \end{bmatrix} \tag{B.7}$$

Dividing the domain in a finite set of non-overlapping elements, such that $\Omega = \bigcup_k^{n_{el}} e_k, n_{el} = \#elements$.

$$r_{ij} = \sum_{k=1}^{n_{el}} r_{ij}^{e_k} \tag{B.8}$$

The element matrix is given by

$$R^{e_k} = \begin{bmatrix} r_{11}^{1,e_k} & \cdots & r_{1l}^{1,e_k} & r_{11}^{2,e_k} & \cdots & r_{1l}^{2,e_k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{l1}^{1,e_k} & \cdots & r_{ll}^{1,e_k} & r_{l1}^{1,e_k} & \cdots & r_{ll}^{1,e_k} \\ r_{11}^{3,e_k} & \cdots & r_{1l}^{3,e_k} & r_{11}^{4,e_k} & \cdots & r_{1l}^{4,e_k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{l1}^{3,e_k} & \cdots & r_{ll}^{3,e_k} & r_{l1}^{4,e_k} & \cdots & r_{ll}^{4,e_k} \end{bmatrix} \tag{B.9}$$

$$\begin{cases} r_{ij}^{1,e_k} = \int_{e_k} (2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + \mu\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}d\Omega \\ r_{ij}^{2,e_k} = \int_{e_k} (\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_2}d\Omega \\ r_{ij}^{3,e_k} = \int_{e_k} (\mu+\lambda)\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_2}d\Omega \\ r_{ij}^{4,e_k} = \int_{e_k} \mu\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + (2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}d\Omega \end{cases} \tag{B.10}$$

where l is the number of discretization points of the element.

Example of the bi-linear quadrilateral element matrix

Using the example of the bi-linear quadrilateral element, the element matrix $R_{ij}^{e_k}$ is a 8 by 8 matrix. First Isoparametric transformation is applied in which the elements of the matrix becomes

$$r_{ij}^{1,e_k} = \int_0^1 \int_0^1 \left((2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + \mu\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}\right)|\text{Jac}(\xi,\eta)_{x,\xi}|d\xi d\eta \tag{B.11}$$

$$r_{ij}^{2,e_k} = \int_0^1 \int_0^1 (\mu+\lambda)\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_2}|\text{Jac}(\xi,\eta)_{x,\xi}|d\xi d\eta \tag{B.12}$$

$$r_{ij}^{3,e_k} = \int_0^1 \int_0^1 (\mu+\lambda)\frac{\partial\varphi_j}{\partial x_1}\frac{\partial\varphi_i}{\partial x_2}|\text{Jac}(\xi,\eta)_{x,\xi}|d\xi d\eta \tag{B.13}$$

$$r_{ij}^{4,e_k} = \int_0^1 \int_0^1 \left(\mu\frac{\partial\varphi_i}{\partial x_1}\frac{\partial\varphi_j}{\partial x_1} + (2\mu+\lambda)\frac{\partial\varphi_i}{\partial x_2}\frac{\partial\varphi_j}{\partial x_2}\right)|\text{Jac}_(\xi,\eta)x,\xi|d\xi d\eta \tag{B.14}$$

Where $|Jac_{x,\xi}(\xi,\eta)|$ is the determinant of the Jacobian

$$|\text{Jac}_{x,\xi}(\xi,\eta)| = (x_2^1 - x_1^1 + A_{x^1}\eta)(x_4^2 - y_1^2 + A_{x^2}\xi) - (x_4^1 - x_1^1 + A_{x^1}\xi)(x_2^2 - x_1^2 + A_{x^2}\eta) \tag{B.15}$$

with $A_{x^1} = x_1^1 - x_2^1 + x_3^1 - x_4^1$ and $A_{x^2} = x_1^2 - x_2^2 + x_3^2 - x_4^2$.

Using Newton-Cotes integration over the unit square

$$\int_0^1\int_0^1 \text{Int}(\xi,\eta)d\xi d\eta \approx \int_0^1\int_0^1 \sum_{k=1}^{l} \text{Int}(\xi^k,\eta^k)\varphi_k d\xi d\eta = \sum_{k=1}^{l}\text{Int}(\xi^k,\eta^k)\int_0^1\int_0^1 \varphi_k d\xi d\eta = \frac{1}{4}\sum_{k=1}^{l}\text{Int}(\xi^k,\eta^k) \tag{B.16}$$

Therefore, at first, the derivatives of the basis functions are calculated

$$B(\xi^k,\eta^k) = \begin{bmatrix} \frac{\partial\varphi_1}{\partial x_1} & \frac{\partial\varphi_1}{\partial x_2} \\ \frac{\partial\varphi_2}{\partial x_1} & \frac{\partial\varphi_2}{\partial x_2} \\ \frac{\partial\varphi_3}{\partial x_1} & \frac{\partial\varphi_3}{\partial x_2} \\ \frac{\partial\varphi_4}{\partial x_1} & \frac{\partial\varphi_4}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial\varphi_1}{\partial\xi^k} & \frac{\partial\varphi_1}{\partial\eta^k} \\ \frac{\partial\varphi_2}{\partial\xi^k} & \frac{\partial\varphi_2}{\partial\eta^k} \\ \frac{\partial\varphi_3}{\partial\xi^k} & \frac{\partial\varphi_3}{\partial\eta^k} \\ \frac{\partial\varphi_4}{\partial\xi^k} & \frac{\partial\varphi_4}{\partial\eta^k} \end{bmatrix} \begin{bmatrix} \frac{\partial\xi^k}{\partial x^1} & \frac{\partial\xi^k}{\partial x^2} \\ \frac{\partial\eta^k}{\partial x^1} & \frac{\partial\eta^k}{\partial x^2} \end{bmatrix} = \begin{bmatrix} \eta^k - 1 & \xi^k - 1 \\ 1-\eta^k & -\xi^k \\ \eta^k & \xi^k \\ -\eta^k & 1-\xi^k \end{bmatrix} J(\eta^k,\xi^k)^{-1} \tag{B.17}$$

Then, the submatrices of the element matrix for every discretization point are written as

$$R_{u^1u^1}^{e_k}(\xi^k,\eta^k) = (2\mu+\lambda)b_{:,1}(\xi^k,\eta^k)b_{:,1}^T(\xi^k,\eta^k) + \mu b_{:,2}(\xi^k,\eta^k)b_{:,2}^T(\xi^k,\eta^k) \tag{B.18}$$

$$R_{u^1u^2}^{e_k}(\xi^k,\eta^k) = \mu b_{:,2}(\xi^k,\eta^k)b_{:,1}^T(\xi^k,\eta^k) + \lambda b_{:,1}(\xi^k,\eta^k)b_{:,2}^T(\xi^k,\eta^k) \tag{B.19}$$

$$R_{u^2u^1}^{e_k}(\xi^k,\eta^k) = \mu b_{:,1}(\xi^k,\eta^k)b_{:,2}^T(\xi^k,\eta^k) + \lambda b_{:,2}(\xi^k,\eta^k)b_{:,1}^T(\xi^k,\eta^k) \tag{B.20}$$

$$R_{u^2u^2}^{e_k}(\xi^k,\eta^k) = \mu b_{:,1}(\xi^k,\eta^k)b_{:,1}^T(\xi^k,\eta^k) + (2\mu+\lambda)b_{:,2}(\xi^k,\eta^k)b_{:,2}^T(\xi^k,\eta^k) \tag{B.21}$$

With Newton-Cotes, the element matrix is

$$R^{e_k} = \frac{1}{4}\sum_{k=1}^{4}|J(\xi_k,\eta_k)|R^{e_k}(\xi^k,\eta^k), \qquad R^{e_k}(\xi,\eta) = \begin{bmatrix} R_{u^1u^1}^{e_k}(\xi^k,\eta^k) & R_{u^1u^2}^{e_k}(\xi^k,\eta^k) \\ R_{u^2u^1}^{e_k}(\xi^k,\eta^k) & R_{u^2u^2}^{e_k}(\xi^k,\eta^k) \end{bmatrix} \tag{B.22}$$

C

FEM approach of the Penalty Term

The penalty term is

$$\mathcal{P}[\mathrm{y}] = \int_{\Omega} \sigma\big(f\big(|\mathrm{Jac}_{y,x}(\mathrm{x})|\big)\big)d\Omega = \sum_{k=1}^{n_{el}} \int_{e_k} \sigma\big(f\big(|\mathrm{Jac}_{y,x}(\mathrm{x}^{e_k})|\big)\big)d\Omega \tag{C.1}$$

Calling the element penalty term $P^{e_k}(\mathrm{y}^{e_k}) := \int_{e_k} \sigma\big(f\big(|\mathrm{Jac}_{y,x}(\mathrm{x}^{e_k})|\big)\big)d\Omega$ and using isoparametric transformation, it reads

$$P^{e_k}(\mathrm{y}^{e_k}) = \int_{e_k} \sigma\big(f\big(|\mathrm{Jac}^{e_k}_{y,x}(\mathrm{x})|\big)\big)d\Omega \tag{C.2}$$

$$= \int_{e_{\xi\eta}} \sigma\left(f\left(\frac{|\mathrm{Jac}_{y,\xi}(\xi,\eta)|}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|}\right)\right)|\mathrm{Jac}_{x,\xi}(\xi,\eta)|d\xi d\eta \tag{C.3}$$

Writing the Jacobian Determinant of the transformed grid y out, we get

$$P^{e_k}(\mathrm{y}^{e_k}) = \int_{e_{\xi\eta}} \sigma((f(g(\xi,\eta)))|\mathrm{Jac}_{x,\xi}(\xi,\eta)|d\xi d\eta \tag{C.4}$$

$$\text{with } g(\xi,\eta) = \Big(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|}\sum_{n=1}^{l} y_1^{n,e_k}\frac{\partial\varphi_n}{\partial\xi}\sum_{n=1}^{n} y_2^{n}\frac{\partial\varphi_n}{\partial\eta} - \sum_{n=1}^{l} y_1^{n}\frac{\partial\varphi_n}{\partial\eta}\sum_{n=1}^{l} y_2^{n}\frac{\partial\varphi_n}{\partial\xi})\Big) \tag{C.5}$$

With first derivatives

$$\begin{cases} \dfrac{\partial P^{e_k}(\mathrm{y}^{e_k})}{\partial y_1^{i,e_k}} = \displaystyle\int_{e_{\xi\eta}} \left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|}\sum_{n=1}^{l} y_2^{n,e_k}\Big(\frac{\partial\varphi_i}{\partial\xi}\frac{\partial\varphi_n}{\partial\eta} - \frac{\partial\varphi_i}{\partial\eta}\frac{\partial\varphi_n}{\partial\xi}\Big)\right)\frac{\partial f(g)}{\partial g}f(g)|\mathrm{Jac}_{x,\xi}(\xi,\eta)|d\xi d\eta \\ \dfrac{\partial P^{e_k}(\mathrm{y}^{e_k})}{\partial y_2^{i,e_k}} = \displaystyle\int_{e_{\xi\eta}} \left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|}\sum_{n=1}^{l} y_1^{n,e_k}\Big(\frac{\partial\varphi_i}{\partial\eta}\frac{\partial\varphi_n}{\partial\xi} - \frac{\partial\varphi_i}{\partial\xi}\frac{\partial\varphi_n}{\partial\eta}\Big)\right)\frac{\partial f(g)}{\partial g}f(g)|\mathrm{Jac}_{x,\xi}(\xi,\eta)|d\xi d\eta \end{cases} \tag{C.6}$$

Due to the computational time in the optimization problem, the higher order derivatives of the inner function of f are neglected. The second order derivatives are

$$
\begin{cases}
\dfrac{\partial^2 \mathcal{P}^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{i,e_k} \partial y_1^{j,e_k}} \approx \displaystyle\int_{e_{\xi\eta}} \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_2^{n,e_k} \left(\frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} - \frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} \right) \right) \frac{\partial f(g)}{\partial g} \right) \cdot \\
\qquad \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_2^{n,e_k} \left(\frac{\partial \varphi_j}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} - \frac{\partial \varphi_j}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} \right) \right) \frac{\partial f(g)}{\partial g} |\mathrm{Jac}_{x,\xi}(\xi,\eta)| \right) d\xi d\eta \\
\dfrac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{i,e_k} \partial y_2^{j,e_k}} \approx \displaystyle\int_{e_{\xi\eta}} \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_2^{n,e_k} \left(\frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} - \frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} \right) \right) \frac{\partial f(g)}{\partial g} \right) \cdot \\
\qquad \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_1^{n,e_k} \left(\frac{\partial \varphi_j}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} - \frac{\partial \varphi_j}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} \right) \right) \frac{\partial f(g)}{\partial g} |\mathrm{Jac}_{x,\xi}(\xi,\eta)| \right) d\xi d\eta \\
\dfrac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{i,e_k} \partial y_1^{j,e_k}} \approx \displaystyle\int_{e_{\xi\eta}} \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_1^{n,e_k} \left(\frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} - \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} \right) \right) \frac{\partial f(g)}{\partial g} \right) \cdot \\
\qquad \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_2^{n,e_k} \left(\frac{\partial \varphi_j}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} - \frac{\partial \varphi_j}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} \right) \right) \frac{\partial f(g)}{\partial g} |\mathrm{Jac}_{x,\xi}(\xi,\eta)| \right) d\xi d\eta \\
\dfrac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{i,e_k} \partial y_2^{j,e_k}} \approx \displaystyle\int_{e_{\xi\eta}} \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_1^{n,e_k} \left(\frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} - \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} \right) \right) \frac{\partial f(g)}{\partial g} \right) \cdot \\
\qquad \left(\left(\frac{1}{|\mathrm{Jac}_{x,\xi}(\xi,\eta)|} \sum_{n=1}^{l} y_1^{n,e_k} \left(\frac{\partial \varphi_i}{\partial \eta} \frac{\partial \varphi_n}{\partial \xi} - \frac{\partial \varphi_i}{\partial \xi} \frac{\partial \varphi_n}{\partial \eta} \right) \right) \frac{\partial f(g)}{\partial g} |\mathrm{Jac}_{x,\xi}(\xi,\eta)| \right) d\xi d\eta
\end{cases}
$$

Assembling the derivatives, the global first derivative of the penalty term is

$$
\nabla \mathcal{P}(\mathbf{y}) = \sum_{k=1}^{n_{el}} Q^{e_k} \begin{bmatrix} \frac{\partial P^{e_k}(\mathbf{y}^{e_k}}{\partial y_1^{1,e_k}} \\ \vdots \\ \frac{\partial P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{l,e_k}} \\ \frac{\partial P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{1,e_k}} \\ \vdots \\ \frac{\partial P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{l,e_k}} \end{bmatrix} \tag{C.7}
$$

And the assembled second order derivative matrix is

$$
\nabla^2 P = \sum_{k=1}^{n_{el}} Q^{e_k} \nabla^2 P^{e_k}(\mathbf{y}^{e_k}) (Q^{e_k})^T \tag{C.8}
$$

with

$$
\nabla^2 P^{e_k}(\mathbf{y}^{e_k}) = \begin{bmatrix}
\frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{1,e_k} \partial y_1^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{1,e_k} \partial y_1^{l,e_k}} & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{1,e_k} \partial y_2^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{1,e_k} \partial y_2^{l,e_k}} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{l,e_k} \partial y_1^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{l,e_k} \partial y_1^{l,e_k}} & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{l,e_k} \partial y_2^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_1^{l,e_k} \partial y_2^{l,e_k}} \\
\frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{1,e_k} \partial y_1^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{1,e_k} \partial y_1^{l,e_k}} & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{1,e_k} \partial y_2^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{1,e_k} \partial y_2^{l,e_k}} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{l,e_k} \partial y_1^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{l,e_k} \partial y_1^{l,e_k}} & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{l,e_k} \partial y_2^{1,e_k}} & \cdots & \frac{\partial^2 P^{e_k}(\mathbf{y}^{e_k})}{\partial y_2^{l,e_k} \partial y_2^{l,e_k}}
\end{bmatrix} \tag{C.9}
$$

D

Jacobian Determinant at Reference Elements

This chapter explains how the Jacobian determinant for the quadrilateral and the quadratic triangular case can be simplified by writing it in terms of the basis functions times the Jacobian determinant values at the discretization points. Recall, the jacobian determinant at a reference element is

$$|J_{u,\xi}(\xi,\eta)| = \frac{\partial u_1}{\partial \xi}\frac{\partial u_2}{\partial \eta} - \frac{\partial u_1}{\partial \eta}\frac{\partial u_2}{\partial \xi} \tag{D.1}$$

where $\mathbf{u}$ is a function in terms of l basis functions

$$u_1 = \sum_{i=1}^{l} u_1^i \varphi_i(\xi,\eta) \tag{D.2}$$

$$u_2 = \sum_{i=1}^{l} u_2^i \varphi_i(\xi,\eta) \tag{D.3}$$

Bi-Linear Quadrilateral Element

The basis functions for the bi-linear quadrilateral element are

$$\sigma_1(\boldsymbol{\xi}) = \lambda_1(\xi)\lambda_1(\eta), \quad \sigma_2(\boldsymbol{\xi}) = \lambda_2(\xi)\lambda_1(\eta),$$
$$\sigma_3(\boldsymbol{\xi}) = \lambda_2(\xi)\lambda_2(\eta), \quad \sigma_4(\boldsymbol{\xi}) = \lambda_1(\xi)\lambda_2(\eta)$$
$$\text{with } \lambda_1(\xi) = 1-\xi, \lambda_2(\xi) = \xi$$

inserting the basis functions into equation D.1, we get

$$\begin{aligned}
|J_{u,\xi}(\xi,\eta)| =& \Big(-u_1^1\lambda_1(\eta) + u_1^2\lambda_1(\eta) + u_1^3\lambda_2(\eta) - u_1^4\lambda_2(\eta)\Big)\Big(-u_2^1\lambda_1(\xi) - u_2^2\lambda_2(\xi) + u_2^3\lambda_2(\xi) + u_2^4\lambda_1(\xi)\Big) \\
&- \Big(-u_2^1\lambda_1(\eta) + u_2^2\lambda_1(\eta) + u_2^3\lambda_2(\eta) - u_2^4\lambda_2(\eta)\Big)\Big(-u_1^1\lambda_1(\xi) - u_1^2\lambda_2(\xi) + u_1^3\lambda_2(\xi) + u_1^4\lambda_1(\xi)\Big) \\
=& \Big((u_1^2-u_1^1)\lambda_1(\eta) + (u_1^3-u_1^4)\lambda_2(\eta)\Big)\Big((u_2^4-u_2^1)\lambda_1(\xi) + (u_2^3-u_2^2)\lambda_2(\xi)\Big) \\
&- \Big((u_2^2-u_2^1)\lambda_1(\eta) + (u_2^3-u_2^4)\lambda_2(\eta)\Big)\Big((u_1^4-u_1^1)\lambda_1(\xi) + (u_1^3-u_1^2)\lambda_2(\xi)\Big)
\end{aligned}$$

Which can be written as

$$\begin{aligned}
|J_{u,\xi}(\xi,\eta)| =& |J_{u,\xi}(0,0)|\lambda_1(\xi)\lambda_1(\eta) + |J_{u,\xi}(1,0)|\lambda_2(\xi)\lambda_1(\eta) \\
&+ |J_{u,\xi}(1,1)|\lambda_2(\xi)\lambda_2(\eta) + |J_{u,\xi}(0,1)|\lambda_1(\xi)\lambda_2(\eta) \\
=& \sum_{l=1}^{4} |J_{u,\xi}(\xi_l,\eta_l)|\sigma_l(\xi,\eta)
\end{aligned} \tag{D.4}$$

Quadratic Triangular Element

The basis functions for the reference quadratic triangle are

$$\begin{cases}\Psi_i(\boldsymbol{\xi}) = \lambda_i(2\lambda_i - 1) & \text{for } i = 1,2,3\\ \Psi_4(\boldsymbol{\xi}) = 4\lambda_1\lambda_2 & \Psi_5(\boldsymbol{\xi}) = 4\lambda_2\lambda_3 \quad \Psi_6(\boldsymbol{\xi}) = 4\lambda_3\lambda_1\\ \lambda_1(\boldsymbol{\xi}) = 1 - \xi - \eta, & \lambda_2(\boldsymbol{\xi}) = \xi, \lambda_3(\boldsymbol{\xi}) = \eta\end{cases}$$

This gives the following Jacobian determinant

$$\begin{aligned}|J_{u,\xi}(\xi,\eta)| =& \Big(\Big(u_1^1(1-4\lambda_1) + u_1^2(4\lambda_2 - 1) + u_1^4(4\lambda_1 - \lambda_2) + u_1^5(4\lambda_3) + u_1^6(-4\lambda_3)\Big)\cdot\\ &\Big(u_2^1(1-4\lambda_1) + u_2^3(4\lambda_3 - 1) + u_2^4(-4\lambda_2) + u_2^5(4\lambda_2) + u_2^6(4\lambda_1 - \lambda_3)\Big)\\ &- \Big(\Big(u_2^1(1-4\lambda_1) + u_2^2(4\lambda_2 - 1) + u_2^4(4\lambda_1 - \lambda_2) + u_2^5(4\lambda_3) + u_2^6(-4\lambda_3)\Big)\cdot\\ &\Big(u_1^1(1-4\lambda_1) + u_1^3(4\lambda_3 - 1) + u_1^4(-4\lambda_2) + u_1^5(4\lambda_2) + u_1^6(4\lambda_1 - \lambda_3)\Big)\\ |J_{u,\xi}(\xi,\eta)| =& \Big(\Big((u_1^1 - u_1^2) + (-4u_1^1 + 4u_1^4)\lambda_1 + (4u_1^2 - u_1^4)\lambda_2 + (4u_1^5 - 4u_1^6)\lambda_3\Big)\cdot\\ &\Big((u_2^1 - u_2^3) + (-4u_2^1 + 4u_2^6)\lambda_1 + (-4u_2^4 + 4u_2^5)\lambda_2 + (4u_2^3 - u_2^6)\lambda_3\Big)\Big)\\ &- \Big(\Big((u_2^1 - u_2^2) + (-4u_2^1 + 4u_2^4)\lambda_1 + (4u_2^2 - u_2^4)\lambda_2 + (4u_2^5 - 4u_2^6)\lambda_3\Big)\cdot\\ &\Big((u_1^1 - u_1^3) + (-4u_1^1 + 4u_1^6)\lambda_1 + (-4u_1^4 + 4u_1^5)\lambda_2 + (4u_1^3 - u_1^6)\lambda_3\Big)\Big)\end{aligned}$$

Using the condition $\lambda_1 + \lambda_2 + \lambda_3 = 1$, the Jacobian is rewritten and results in

$$\begin{aligned}|J_{u,\xi}(\xi,\eta)| =& \Big(A_1^1\lambda_1 + A_1^2\lambda_2 + A_1^3\lambda_3\Big)\Big(B_2^1\lambda_1 + B_2^2\lambda_2 + B_2^3\lambda_3\Big)\\ &- \Big(A_2^1\lambda_1 + A_2^2\lambda_2 + A_2^3\lambda_3\Big)\Big(B_1^1\lambda_1 + B_1^2\lambda_2 + B_1^3\lambda_3\Big)\end{aligned}$$

$$\text{with } \begin{cases}A_i^1 = -3u_i^1 - u_i^2 + 4u_i^4\\ A_i^2 = u_i^1 + 3u_i^2 - u_1^4\\ A_i^3 = u_i^1 - u_i^2 + 4u_i^5 - 4u_i^6\\ B_i^1 = -3u_i^1 - u_i^3 + 4u_i^6\\ B_i^2 = u_i^1 - u_i^3 - 4u_i^4 + 4u_i^5\\ B_i^3 = u_i^1 + 3u_i^3 - u_i^6\end{cases}$$

Which gives

$$\begin{aligned}|J_{u,\xi}(\xi,\eta)| =& \Big(A_1^1B_2^1(\lambda_1)^2 + A_1^2B_2^2(\lambda_2)^2 + A_1^3B_2^3(\lambda_3)^2 + (A_1^1B_2^2 + A_1^2B_2^1)\lambda_1\lambda_2\\ &+ (A_1^2B_2^3 + A_1^3B_2^2)\lambda_2\lambda_3 + (A_1^1B_2^3 + A_1^3B_2^1)\lambda_1\lambda_3\Big)\\ &- \Big(A_2^1B_1^1(\lambda_1)^2 + A_2^2B_1^2(\lambda_2)^2 + A_2^3B_1^3(\lambda_3)^2 + (A_2^1B_1^2 + A_2^2B_1^1)\lambda_1\lambda_2\\ &+ (A_2^2B_1^3 + A_2^3B_1^2)\lambda_2\lambda_3 + (A_2^1B_1^3 + A_2^3B_1^1)\lambda_1\lambda_3\Big)\end{aligned}$$

Using again the condition $\lambda_1+\lambda_2+\lambda_3 = 1$, then the Jacobian determinant in terms of quadratic triangular basis functions is

$$\begin{aligned}|J_{u,\xi}(\xi,\eta)| =&(A_1^1B_2^1 - A_2^1B_1^1)\lambda_1(2\lambda_1 - 1) - (A_1^2B_2^2 - A_2^2B_1^2)\lambda_2(2\lambda_2 - 1) - (A_1^3B_2^3 - A_2^3B_1^3)\lambda_3(2\lambda_3 - 1)\\ &+ \frac{1}{4}\Big((A_1^1 + A_1^2)(B_2^1 + B_2^2) - (A_2^1 + A_2^2)(B_1^1 + B_1^2)\Big)4\lambda_1\lambda_2\\ &+ \frac{1}{4}\Big((A_1^2 + A_1^3)(B_2^2 + B_2^3) - (A_2^2 + A_2^3)(B_1^2 + B_1^3)\Big)4\lambda_2\lambda_3\\ &+ \frac{1}{4}\Big((A_1^1 + A_1^3)(B_2^1 + B_2^3) - (A_2^1 + A_2^3)(B_1^1 + B_1^3)\Big)4\lambda_1\lambda_3\end{aligned}$$

Which is exactly the Jacobian determinant at the discretization points times the basis function of the corresponding point

$$|J_{u,\xi}(\xi,\eta)| = \sum_{l=1}^{6} |J_{u,\xi}(\xi_l,\eta_l)|\Psi_l(\xi,\eta)$$

Bi-Quadratic Quadrilateral Element

$$\begin{cases}\Psi_1(\boldsymbol{\xi}) = \theta_1(\xi)\theta_1(\eta), & \Psi_5(\boldsymbol{\xi}) = \theta_3(\xi)\theta_1(\eta), & \Psi_9(\boldsymbol{\xi}) = \theta_3(\xi)\theta_3(\eta)\\ \Psi_2(\boldsymbol{\xi}) = \theta_2(\xi)\theta_1(\eta), & \Psi_6(\boldsymbol{\xi}) = \theta_2(\xi)\theta_3(\eta),\\ \Psi_3(\boldsymbol{\xi}) = \theta_2(\xi)\theta_2(\eta), & \Psi_7(\boldsymbol{\xi}) = \theta_3(\xi)\theta_2(\eta),\\ \Psi_4(\boldsymbol{\xi}) = \theta_1(\xi)\theta_2(\eta), & \Psi_8(\boldsymbol{\xi}) = \theta_1(\xi)\theta_3(\eta),\end{cases} \tag{D.5}$$

$$\text{with } \theta_1(\xi) = 1 - 3\xi + 2\xi^2 \quad \theta_2(\xi) = -\xi + 2\xi^2 \quad \theta_3(\xi) = 4\xi - 4\xi^2 \tag{D.6}$$

Applying the basis functions for the bi-quadratic quadrilateral element to the Jacobian determinant, we get

$$\begin{aligned}|J_{u,\xi}(\xi,\eta)| =&\Bigg(\Big(u_1^1(-3+4\xi)\theta_1(\eta) + u_1^2(-1+4\xi)\theta_1(\eta) + u_1^3(-1+4\xi)\theta_2(\eta) + u_1^4(-3+4\xi)\theta_2(\eta) + u_1^5(4-8\xi)\theta_1(\eta)\\ &+ u_1^6(-1+4\xi)\theta_3(\eta) + u_1^7(4-8\xi)\theta_2(\eta) + u_1^8(-3+4\xi)\theta_3(\eta) + u_1^9(4-8\xi)\theta_3(\eta)\Big)\cdot\\ &\Big(u_2^1(-3+4\eta)\theta_1(\xi) + u_2^2(-3+4\eta)\theta_2(\xi) + u_2^3(-1+4\eta)\theta_2(\xi) + u_2^4(-1+4\eta)\theta_1(\xi) + u_2^5(-3+4\eta)\theta_3(\xi)\\ &+ u_2^6(4-8\eta)\theta_2(\xi) + u_2^7(-1+4\eta)\theta_3(\xi) + u_2^8(4-8\eta)\theta_1(\xi) + u_2^9(4-8\eta)\theta_3(\xi)\Big)\Bigg)\\ &-\Bigg(\Big(u_2^1(-3+4\xi)\theta_1(\eta) + u_2^2(-1+4\xi)\theta_1(\eta) + u_2^3(-1+4\xi)\theta_2(\eta) + u_2^4(-3+4\xi)\theta_2(\eta) + u_2^5(4-8\xi)\theta_1(\eta)\\ &+ u_2^6(-1+4\xi)\theta_3(\eta) + u_2^7(4-8\xi)\theta_2(\eta) + u_2^8(-3+4\xi)\theta_3(\eta) + u_2^9(4-8\xi)\theta_3(\eta)\Big)\cdot\\ &\Big(u_1^1(-3+4\eta)\theta_1(\xi) + u_1^2(-3+4\eta)\theta_2(\xi) + u_1^3(-1+4\eta)\theta_2(\xi) + u_1^4(-1+4\eta)\theta_1(\xi) + u_1^5(-3+4\eta)\theta_3(\xi)\\ &+ u_1^6(4-8\eta)\theta_2(\xi) + u_1^7(-1+4\eta)\theta_3(\xi) + u_1^8(4-8\eta)\theta_1(\xi) + u_1^9(4-8\eta)\theta_3(\xi)\Big)\Bigg)\end{aligned}$$

Using the linear basis functions $\lambda_1(\xi) = 1 - \xi, \lambda_2(\xi) = \xi$, it reads

$$\begin{aligned}|J_{u,\xi}(\xi,\eta)| =&\Big(\big((A_1^{1,1}\lambda_1(\xi) + A_1^{1,2}\lambda_2(\xi))\theta_1(\eta)) + (A_1^{2,1}\lambda_1(\xi) + A_1^{2,2}\lambda_2(\xi))\theta_2(\eta)) + (A_1^{3,1}\lambda_1(\xi) + A_1^{3,2}\lambda_2(\xi))\theta_3(\eta)\big)\\ &\big((B_2^{1,1}\lambda_1(\eta) + B_2^{1,2}\lambda_2(\eta))\theta_1(\xi)) + (B_2^{2,1}\lambda_1(\eta) + B_2^{2,2}\lambda_2(\eta))\theta_2(\xi)) + (B_2^{3,1}\lambda_2(\eta) + B_2^{3,2}\lambda_2(\eta))\theta_3(\xi)\big)\\ -&\Big(\big((A_2^{1,1}\lambda_1(\xi) + A_2^{1,2}\lambda_2(\xi))\theta_1(\eta)) + (A_2^{2,1}\lambda_1(\xi) + A_2^{2,2}\lambda_2(\xi))\theta_2(\eta)) + (A_2^{3,1}\lambda_1(\xi) + A_2^{3,2}\lambda_2(\xi))\theta_3(\eta)\big)\\ &\big((B_1^{1,1}\lambda_1(\eta) + B_1^{1,2}\lambda_2(\eta))\theta_1(\xi)) + (B_1^{2,1}\lambda_1(\eta) + B_1^{2,2}\lambda_2(\eta))\theta_2(\xi)) + (B_1^{3,1}\lambda_2(\eta) + B_1^{3,2}\lambda_2(\eta))\theta_3(\xi)\big)\end{aligned}$$

The coefficients are

$$A_i^{1,1} = -3u_i^1 - u_i^2 + 4u_i^5, \qquad B_i^{1,1} = -3u_i^1 - u_i^4 + 4u_i^8,$$
$$A_i^{1,2} = u_i^1 + 3u_i^2 - 4u_i^5, \qquad B_i^{1,2} = u_i^1 + 3u_i^4 - 4u_i^8,$$
$$A_i^{2,1} = -u_i^3 - 3u_i^4 + u_i^7, \qquad B_i^{2,1} = -3u_i^2 - u_i^3 + 4u_i^6,$$
$$A_i^{2,2} = 3u_i^3 + u_i^4 - 4u_i^7, \qquad B_i^{2,2} = u_i^2 + 3u_i^3 - 4u_i^6,$$
$$A_i^{3,1} = -u_i^6 - 3u_i^8 + 4u_i^9, \qquad B_i^{3,1} = -3u_i^5 - u_i^7 + 4u_i^9,$$
$$A_i^{3,2} = 3u_i^6 + u_i^8 - 4u_i^9, \qquad B_i^{3,2} = u_i^5 + 3u_i^7 - 4u_i^9.$$

Results in

$$|J_{u,\xi}(\xi,\eta)| = \sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{2}\sum_{l=1}^{3} C_{ijkl}\lambda_i(\xi)\theta_j(\xi)\lambda_k(\eta)\theta_l(\eta)$$
$$\text{with } C_{ijkl} = (A_1^{l,i}B_2^{j,k} - A_2^{l,i}B_1^{j,k})$$

The linear times quadratic basis functions can be written in terms of cubic basis functions

$$\lambda_1(\xi)\theta_1(\xi) = \tau_1(\xi) + \frac{1}{27}\big(4\tau_3(\xi) - \tau_4(\xi)\big), \qquad \lambda_2(\xi)\theta_2(\xi) = \tau_2(\xi) + \frac{1}{27}\big(-\tau_3(\xi) + 4\tau_4(\xi)\big),$$
$$\lambda_2(\xi)\theta_1(\xi) = \frac{1}{27}\big(2\tau_3(\xi) - 2\tau_4(\xi)\big), \qquad \lambda_1(\xi)\theta_3(\xi) = \frac{1}{27}\big(16\tau_3(\xi) + 8\tau_4(\xi)\big),$$
$$\lambda_1(\xi)\theta_2(\xi) = \frac{1}{27}\big(-2\tau_3(\xi) + 2\tau_4(\xi)\big), \qquad \lambda_2(\xi)\theta_2(\xi) = \frac{1}{27}\big(8\tau_3(\xi) + 16\tau_4(\xi)\big)$$

Where the cubic basis functions are

$$\tau_i(\xi) = \begin{cases} \frac{1}{2}\big(3\lambda_i(\xi) - 1\big)\big(3\lambda_i(\xi) - 2\big)\lambda_i(\xi) & \text{for } i = 1,2 \\ \frac{9}{2}\lambda_1(\xi)\lambda_2(\xi)\big(3\lambda_1(\xi) - 1\big) & \text{for } i = 3 \\ \frac{9}{2}\lambda_1(\xi)\lambda_2(\xi)\big(3\lambda_2(\xi) - 1\big) & \text{for } i = 4 \end{cases}$$

Which results in the following Jacobian determinant in terms of cubic basis functions

$$|J_{u,\xi}(\xi,\eta)| = \sum_{i=1}^{4}\sum_{j=1}^{4} |J_{u,\xi}(\xi_i,\eta_j)|\tau_i(\xi)\tau_j(\eta) \tag{D.7}$$
$$= \sum_{l=1}^{16} |J_{u,\xi}(\xi_l,\eta_l)|\mathrm{T}_l(\xi,\eta) \tag{D.8}$$

with

$$\mathrm{T}_1(\xi,\eta) = \tau_1(\xi)\tau_1(\eta), \quad \mathrm{T}_5(\xi,\eta) = \tau_3(\xi)\tau_1(\eta), \quad \mathrm{T}_9(\xi,\eta) = \tau_4(\xi)\tau_2(\eta), \quad \mathrm{T}_{13}(\xi,\eta) = \tau_3(\xi)\tau_3(\eta),$$
$$\mathrm{T}_2(\xi,\eta) = \tau_2(\xi)\tau_1(\eta), \quad \mathrm{T}_6(\xi,\eta) = \tau_4(\xi)\tau_1(\eta), \quad \mathrm{T}_{10}(\xi,\eta) = \tau_3(\xi)\tau_2(\eta), \quad \mathrm{T}_{14}(\xi,\eta) = \tau_4(\xi)\tau_3(\eta),$$
$$\mathrm{T}_3(\xi,\eta) = \tau_2(\xi)\tau_2(\eta), \quad \mathrm{T}_7(\xi,\eta) = \tau_2(\xi)\tau_3(\eta), \quad \mathrm{T}_{11}(\xi,\eta) = \tau_1(\xi)\tau_4(\eta), \quad \mathrm{T}_{15}(\xi,\eta) = \tau_4(\xi)\tau_4(\eta),$$
$$\mathrm{T}_4(\xi,\eta) = \tau_1(\xi)\tau_2(\eta), \quad \mathrm{T}_8(\xi,\eta) = \tau_2(\xi)\tau_4(\eta), \quad \mathrm{T}_{12}(\xi,\eta) = \tau_1(\xi)\tau_3(\eta), \quad \mathrm{T}_{16}(\xi,\eta) = \tau_3(\xi)\tau_4(\eta).$$

Bibliography

[1] Y. Amit. A nonlinear variational problem for image matching. *SIAM Journal on Scientific Computing*, 15(1):207–224, 1994.

[2] I. Arganda-Carreras, C. Sorzano, R. Marabini, J. M. Carazo, C. Ortiz-de Solorzano, and J. Kybic. Consistent and elastic registration of histological sections using vector-spline regularization. In *International Workshop on Computer Vision Approaches to Medical Image Analysis*, pages 85–95. Springer, 2006.

[3] C. Ayala Bravo and V. Ayala. Mathematical analysis of the geometric mapping in the finite element method. *Journal of Mathematics and Computer Science*, 16:273–281, 01 2016.

[4] C. Broit. *Optimal registration of deformed images*. PhD thesis, University of Pensylvania, USA, 1981.

[5] L.G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, December 1992. ISSN 0360-0300.

[6] J. Dauguet, T. Delzescaux, F. Condé, J-F. Mangin, N. Ayache, P. Hantraye, and V. Frouin. Three-dimensional reconstruction of stained histological slices and 3d non-linear registration with in-vivo mri for whole baboon brain. *Journal of neuroscience methods*, 164(1):191–204, 2007.

[7] G. E. Farin and G. Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.

[8] B. Fischer and J. Modersitzki. Fast diffusion registration. *Contemporary Mathematics*, 313:117–128, 2002.

[9] B. Fischer and J. Modersitzki. Curvature based image registration. *Journal of Mathematical Imaging and Vision*, 18(1):81–85, 2003.

[10] B. Fischer and J. Modersitzki. Ill-posed medicine—an introduction to image registration. *Inverse Problems*, 24(3):034008, 2008.

[11] M. B. van Gijzen. CG methods for weighted and regularised least squares problems, notes.

[12] M. B. van Gijzen. *Iterative solution methods for linear equations in finite element computations*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 1994.

[13] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic Press, London, UK, 1981.

[14] G. H. Golub and C. F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, Baltimore, USA, 2013.

[15] P. L. Gould. *Introduction to linear elasticity*. Springer, New York, USA, 2013.

[16] P. C. Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM review*, 34 (4):561–580, 1992.

[17] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras. Variational methods for multimodal image matching. *International Journal of Computer Vision*, 50(3):329–343, 2002.

[18] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[19] T.J.R. Hughes, I. Levit, and J. Winget. An element-by-element solution algorithm for problems of structural and solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 36(2): 241 – 254, 1983.

[20] A. Johnen, J.-F. Remacle, and C. Geuzaine. Geometrical validity of curvilinear finite elements. *Journal of Computational Physics*, 233:359 – 372, 2013.

[21] H. J. Johnson and G. E. Christensen. Consistent landmark and intensity-based image registration. *IEEE transactions on medical imaging*, 21(5):450–461, 2002.

[22] J. van Kan, A. Segal, and F. Vermolen. *Numerical methods in scientific computing*. VSSD, Delft, The Netherlands, 2005.

[23] A. Lange. Multigrid method for FEM-based elastic registration of histological serial slices. Master's thesis, Universität zu Lübeck, Germany, 2018.

[24] R. J. LeVeque. *Finite volume methods for hyperbolic problems.* Cambridge University Press, Cambridge, UK, 2005.

[25] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1 – 36, 1998. ISSN 1361-8415.

[26] J. Modersitzki. *Numerical methods for Image Registration*. Oxford University Press, Oxford, UK, 2004.

[27] J. Modersitzki. *FAIR: Flexible Algorithms for Image Registration*. SIAM, Philadelphia, USA, 2009.

[28] J. Modersitzki, O. Schmitt, and S. Wirtz. Registration of histological serial sectionings. In *Mathematical Models for Registration and Applications to Medical Imaging*, pages 63–80. Springer, 2006.

[29] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, New York, USA, 2006.

[30] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, USA, 2003.

[31] P. A. van den Elsen, E. J. D. Pol, and M. A. Viergever. Medical image matching-a review with classification. *IEEE Engineering in Medicine and Biology Magazine*, 12(1):26–39, March 1993. ISSN 0739-5175.

[32] A Van der Sluis. Condition, equilibration and pivoting in linear algebraic systems. *Numerische Mathematik*, 15(1):74–86, 1970.

[33] M. Viergever, J.B. Maintz, S. Klein, K. Murphy, M. Staring, and J.P.W. Pluim. A survey of medical image registration - under review. *Medical Image Analysis*, 33:140 – 144, 2016. ISSN 1361-8415. 20th anniversary of the Medical Image Analysis journal (MedIA).

[34] O. C. Zienkiewicz, Robert L. Taylor, and J. Z. Zhu. *Finite element method : its basis and fundamentals*. Butterworth-Heinemann, Oxford, UK, 2013.