UNIVERSITÄT ZU LÜBECK

From the Institute of Mathematics and Image Computing
of the University of Lübeck
Director: Prof. Dr. rer. nat. Jan Modersitzki

A Second-Order Divergence-Curl Regularization in Variational Image Registration

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Sciences and Technical Engineering

Submitted by

Saskia Neuber
from Forst in der Lausitz

Lübeck, 2025

First referee:	Prof. Dr. Jan Modersitzki
Second referee:	Prof. Dr. Jürgen Prestin
Date of oral examination:	13th January 2026
Approved for printing:	19th January 2026

Abstract

In this thesis we consider a family of second-order divergence-curl (DC) regularizers in the context of image registration. Image registration describes the process of finding geometrical correspondences between two images and can be formulated as a variational problem. Due to the ill-posedness of variational image registration problems, regularization is unavoidable.
The considered family of second-order DC regularizers consists of second-order energies based on a convex combination of the gradient of divergence and the gradient of the curl. Inspired by the physics of fluids, the second-order DC regularization allows the inclusion of a priori knowledge about the problem under consideration and also enables a coupling of the components. The second order naturally leads to smoother solutions as compared to first-order regularization and thus allows for e.g. 2D/3D registrations or corresponding landmark-based pre-registrations. The drawback of the second order is that the numerical treatment of the registration problem is significantly more complex.
In addition to the investigation of second-order DC regularization in a continuous setting, we therefore present a numerical method for solving the image registration problem under consideration. In particular, we propose a discretize-then-optimize framework based on a multigrid method, since multigrid methods are known for their high efficiency. Although the efficiency of the proposed method could be further improved, we demonstrate the suitability of multigrid methods for the considered image registration problem with second-order DC regularization.

Zusammenfassung

In dieser Arbeit betrachten wir eine Familie von Divergenz-Rotations (DC) Regularisierern zweiter Ordnung im Kontext der Bildregistrierung. Bildregistrierung beschreibt die Suche nach geometrischen Zusammenhängen zwischen zwei Bildern und kann als Variationsproblem formuliert werden. Durch die Schlechtgestelltheit von Variationsproblemen in der Bildregistrierung ist Regularisierung unverzichtbar.
Die betrachtete Familie von DC-Regularisierern zweiter Ordnung besteht aus Energien zweiter Ordnung, die auf einer Konvexkombination des Gradientens der Divergenz und des Gradientens der Rotation basieren. Inspiriert von der Physik von Flüssigkeiten, erlaubt die DC-Regularisierung zweiter Ordnung die Einbringung von a priori Wissen über das betrachtete Problem und ermöglicht auch die Kopplung der Komponenten. Die zweite Ordnung führt natürlicherweise zu glatteren Lösungen im Vergleich zur Regularisierung erster Ordnung und ermöglicht so z. B. 2D/3D-Registrierungen oder entsprechende landmarkenbasierte Vorregistrierungen. Der Nachteil der zweiten Ordnung ist, dass die numerische Handhabung des Registrierungsproblems deutlich komplexer ist.
Daher stellen wir, zusätzlich zur Untersuchung der DC-Regularisierung zweiter Ordnung in einer kontinuierlichen Umgebung, eine numerische Methode zur Lösung des betrachteten Bildregistrierungsproblems vor. Insbesondere schlagen wir eine Diskretisierungs-dann-Optimierungsmethode vor, die auf einem Mehrgitterverfahren basiert, da Mehrgitterverfahren für ihre hohe Effizienz bekannt sind. Obwohl die Effizienz der vorgeschlagenen Methode noch weiter verbessert werden könnte, zeigen wir die Eignung von Multigrid-Methoden für das betrachtete Bildregistrierungsproblem mit DC-Regularisierung zweiter Ordnung.

Acknowledgments

Zuallerst möchte ich ganz herzlich Prof. Jan Modersitzki für seine Betreuung danken. Für die Möglichkeit zu promovieren, aber auch die von Ehrlichkeit geprägte Betreuung, die neben so vielen fachlichen Rückmeldungen, detaillierten Nachfragen und Strukturierungvorschlägen auch viel Raum zum persönlichen Wachstum geboten hat.
Ein ganz besonders großes Dankeschön, geht aber auch an meine liebe Kollegin und auch Freundin Pia Schulz, die neben bereichernden inhaltlichen Diskussion und Denkanstößen auch immer ein offenes Ohr für die großen und auch kleinen Dinge des Alltags hatte und immer motivierende Worte gefunden hat.
Für ihre große emotionale Unterstützung in den Tiefs aber auch ihre Begeisterung in den Hochs (obwohl ich sicher nicht immer gut genug rübergebracht habe, was eigentlich konkret klappt) und ihr super schnelles Korrekturlesen möchte ich auch meiner lieben Freundin Laura Deecke danken.
Auch meiner Familie möchte ich für ihren festen Glauben an mich und ihre andauernde Zusprache danken.
Neben Pia, Lauri und meiner Familie gilt auch all meinen anderen lieben Freunden, ob aus Studienzeiten oder vom Ballett, sowie meinen Ballettleheren mein Dank, dafür dass sie einen Raum geschaffen haben, indem ich den Kopf von der Arbeit frei machen konnte.
Vielen lieben Dank auch an alle Kollegen des MICs/MEVIS, die die Zeit am Institut unterhaltsam gestaltet haben, aber auch hilfreiche Rückfragen z.B. in der Forschungsrunde gestellt haben.
Ebenso möchte ich meinem Zweitprüfer Prof. Jürgen Prestin besonders danken, dafür dass er mich auf die offene Stelle am MIC hingewiesen und dazu ermutigt hat mich zu bewerben und nun auch die Rolle eines Prüfers übernimmt.
Vielen Dank auch an Prof. Heinz Handels dafür, dass er der Vorsitz meinem Promotionsverfahren übernimmt.
Zuletzt bin ich auch dankbar in eine Zeit zu leben in denen Übersetzter wie DeepL das Schreiben in einer anderen Sprache als der Muttersprache erleichtern.

Contents

1 Introduction

In this chapter we introduce the main topics covered in this thesis. We study variational image registration problems with second-order divergence-curl (DC) regularization. A proper definition of the regularizer follows In Def. 2.2.1. Moreover, we present multigrid methods as a tool of numerical optimization, to solve these variational problems in combination with e.g. Gauss-Newton optimization numerically. We embed our contributions in the related literature and clarify the organization of this thesis.

1.1 Variational Image Registration with Second-Order Divergence-Curl Regularization

In this thesis we study variational image registration with second-order DC regularization.

With two given images, the aim of image registration is to determine a meaningful transformation, so that the transformed version of the first image is similar to the second image [1]. As image registration has been extensively studied, we refer to the textbooks [1, 2, 3, 4, 5, 6] and the survey articles [7, 8, 9, 10, 11, 12, 13, 14] for a general introduction to the topic. Variational methods [15, 16] are a standard approach to model image registration, see e.g. [1, 7, 10, 17], next to other formulations such as e.g. Deep Learning [18, 19] or Optical Flow [20]. Optical flow approaches are highly related to variational image registration, as they also consider a variational formulation to align images, but incorporate an additional time variable in their problem formulation; see e.g. [20]. Due to this close relation we will also consider optical flow approaches, when reviewing related literature.
The objective functional in the variational formulation of image registration, usually consists of a data-fitting term and a regularizer [1, 15, 21]. The essential role of the regularizer is to ensure the existence of a solution; see e.g. [11, 22]. The regularizer further stabilizes the problem against a potentially noise-corrupted data-fitting term, as typically regularizers are based on derivatives and thus ensure a certain smoothness. Moreover, by the concrete choice of the regularizer it is possible to model prior knowledge about the expected solution of the considered problem.

We consider a second-order DC regularization for a two- and three-dimensional setting. The family of second-order DC energies was originally introduced for two-dimensional interpolation by Amodei and Benbourhim [23, 24]. The interpolation was extended to three-dimensional settings by Dodu and Rabut [25] and Benbourhim and Bouhamidi [26]. Within the last years the second-order DC energy was extensively used for the task of regularization; see e.g [27, 28, 29].
The considered second-order DC regularizer consists of a convex combination of the gradient of the divergence and the gradient of the curl. The combination of divergence and curl is physically inspired: The Helmholtz decomposition decomposes a vector-field into a divergence free and a curl free part [24]. In [30], the physical background of the second-order DC regularizer is outlined.

1.2 Multigrid Methods in Variational Image Registration

In this thesis, we present a novel multigrid method for the numerical solution of the variational image registration problem with second-order DC regularization.

We refer to [31, 32, 33] for an introduction to multigrid methods. Originally introduced in the 1960s for the solution of discretized linear elliptic PDEs [34, 35], multigrid methods were discovered for practical applications in the 1970s [36, 37, 38, 39, 40] and further developed for other types of linear and nonlinear PDEs, eigenvalue problems, integral equations, optimization problems and many more; see [32, 41, 42] and references therein. Today, as multigrid methods are still relevant for many practical applications, theoretical considerations and new solvers are a crucial part of current research; see e.g. [42, 43, 44]. The versatility of multigrid methods led to a broad area of application, including variational image registration [45, 46, 47].
Multigrid methods are used for the numerical solution of registration problems for more than 25 years, but still are state of the art due to their high efficiency; see e.g. [48, 49, 50]. Appropriate discretization and numerical optimization are required for the numerical solution. Within the optimization of the discrete image registration problems, potentially large linear systems need to be solved; see e.g. [51]. Here, iterative numerical schemes, such as conjugate gradient [52, 53] or multigrid methods [31, 53] come into play. In this thesis, we particularly study multigrid solvers, as they are known to be optimal in the sense, that the computation of a solution has $\mathcal{O}(N)$ complexity, where N is the number of unknowns [31].
Multigrid methods can also be applied to other related problem formulations such as e.g. conformal image registration [54], optical flow [55] or deep learning approaches [56], which underlines their versatility.

Multigrid methods rely on an efficient interplay between a smoothing effect and a representation of the problem on several coarser discretizations. The concrete choice of the multigrid components is problem-specific. Thus, convergence cannot be guaranteed for a general framework. A tool to capture the theoretically expected convergence of a certain multigrid method for a specific problem is local Fourier analysis (LFA) [57]. More specifically, the h-ellipticity is a measure to ensure the existence of a point-wise smoothing procedure for a certain discretization; see [31, p. 300]. Furthermore, the smoothing factor, is a tool to verify, that an iterative procedure is a smoother for a certain choice of parameters; see [31, p. 298].

Our contribution is a novel multigrid solver for variational problems with second-order DC regularization. We further present a local Fourier analysis of the considered discretized second-order DC regularizer, to show its suitability for multigrid. Moreover, we propose an implementation of the multigrid method for variational image registration with second-order DC regularization. The implementation is validated by numerical experiments.

1.3 Overview on Literature Related to Variational Image Registration with Second-Order Divergence-Curl Regularization and Multigrid Methods

In this section, we review approaches that combine variational image registration with second-order DC regularization. Moreover, we provide an overview over multigrid methods for the solution of the image registration problems.

First, we give an overview of the variational image registration approaches in combination with second-order DC regularization.

The optical flow community was the first to discover the second-order DC energies for regularization. In particular, Suter [27] recognized the potential of the coupled energy for the regularization of two-dimensional optical flow problems. Later, Gupta and Prince [58] introduced an extension to three-dimensional optical flow problems, where not only the second-order DC regularizer but also the first-order DC regularizer was considered. The Helmholtz decomposition was used in [59] to reformulate the two-dimensional optical flow problem in the stream and velocity potential leading to a third-order problem. To reduce the order of the considered problem, [59] uses the approach with auxiliary variables of [60]. The problem formulation by [59] was further extended with a regularization on the boundary in [61].
Later on, the second-order DC energies were also discovered for classical variational image registration tasks. A two-dimensional landmark-based registration based on the splines from second-order DC interpolation was introduced by Chen and Suter [62]. Furthermore, a landmark-based followed by an intensity-based registration with the second-order DC regularizer for two-dimensional images was proposed by Sorzano, Thévenaz and Unser [28]. The intensity-based registration in [28] was parametrized with B-splines. A further extension of the approach [28] with an additional penalty term inspired from consistent registration was considered in [63]. Tzitzimpasis and co-authors [29] proposed a three-dimensional intensity-based registration approach with the second-order DC regularizer. In [29], a generalization of the DC regularization for image registration tasks was considered as different orders of the regularizer are possible. Moreover, the solution of an image registration problem in [29] is obtained by a reduction to a lower order problem based on the Helmholtz decomposition.
The recent applications of the second-order DC regularizer to variational image registration problems [64, 65, 66], emphasizes the relevance of the proposed regularizer to todays image registration problems.

The following is an overview of multigrid solvers for different regularization strategies in the context of variational image registration.
Multigrid methods were first applied to variational image registration approaches with elastic regularization. As the elastic regularizer is a coupled first-order regularizer, a coupled second-order PDE is obtained. Henn and co-authors [67] introduced a multigrid framework for the elastic registration of brain data based on a discretization with finite differences. In the follow-up paper by Henn and Witsch [68], more advanced multigrid settings, such as full multigrid, are applied and more details, as e.g. the specific discretization and the full approximation scheme, are described. Further multigrid solvers for elastic registrations are presented in [46, 69, 70]. Both [46, 69] combine the multigrid solver with a multilevel strategy. Furthermore, in [46] a new staggered grid discretization is presented and a LFA is provided. LFA is also performed in [70] for an elastic registration approach with a volume preserving penalty. Both papers determine the h-ellipticity measure to ensure the existence of a smoother. This enables a comparison to the h-ellipticity of the second-order DC regularizer, which we determine in Ch. 7.
There are also multigrid frameworks for other first-order regularizers, leading to a second-order PDE. For example, multigrid solvers for the well-known uncoupled diffusion [71], the uncoupled total variation-based [72, 48] and the coupled hyperelastic [47] regularizer exist. Also in [47], the h-ellipticity measure is determined. More recently, in [73, 74, 75] a special form of diffusion energy inspired from a large deformation diffeomorphic metric mapping approach is presented in combination with multigrid solvers.
Multigrid solvers were also used for registration problems with second-order regularization, i.e. fourth-order PDEs. Existing fourth-order schemes such as [76, 77, 78] enable

the splitting of the operator. One common example is the biharmonic problem

$$\Delta^2 u = f$$

that is split into two Poisson problems

$$\begin{pmatrix} \Delta & 0 \\ -I & \Delta \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix};$$

see [31, §8.4] for details. In [31, §8.4] it is outlined, that the solution of the two second-order problems one after another is numerically more efficient than the solution of the original fourth-order problem. Henn [76] introduced a fourth-order PDE based on the biharmonic diffusion equation with higher-order boundary conditions. Note, that the biharmonic regularizer is also known as the uncoupled curvature regularizer. Furthermore, Henn presents a special treatment of the boundary conditions. Moreover, in [77] two coupled nonlinear fourth-order PDEs rise from the image registration problem with a full curvature regularizer. The authors propose different methods to tackle the problem including a splitting of the operator, leading to multigrid solvers for second-order PDEs but also a multigrid solver for the fourth-order PDE. In particular, the method that relies on a reformulation to second-order PDEs is shown to be superior to a method based on the fourth-order formulation [77]. Within LFA, the local smoothing factor is determined numerically. Since we determine the local smoothing factor for a multigrid framework with the second-order DC regularizer in Ch. 7, this again enables a comparison. Furthermore, in [78] the uncoupled second-order curvature as well as the uncoupled first-order diffusion regularizer are used as application examples for a new smoothing procedure for image registration problems. Here, the h-ellipticity measure as well as the local smoothing factor are determined for both regularizers.
Two other fourth-order PDEs obtained from optical flow problems and solved with multigrid are outlined in the following, as they are related to our variational image registration problem with second-order DC regularization. Köstler [55] proposed a regularizer, that consists of an uncoupled first-order diffusion and an uncoupled second-order curvature-based part. Due to the uncoupled components, the PDE can be reduced to a second-order one. The h-ellipticity measure of the combined regularizer is determined. The extension of the curvature regularization to the family of second-order DC regularizers is mentioned in the outlook in [55], but, to the best of our knowledge, it was not performed so far. In [79], multigrid is used for an optical flow problem, that contains a special second-order DC regularizer. Here, the same preliminary factor is used for the gradient of the divergence and for the gradient of the curl. Hence, the approach is uncoupled in the components and the fourth-order PDE can again be split into two second-order PDEs.

1.4 Contributions and Outline

The overarching goal of this thesis is to study different aspects of second-order DC regularization in the context of variational image registration. This thesis is based on and linked to the two publications

1. S. Neuber, P. F. Schulz, S. Kuckertz and J. Modersitzki. Segmentation-inspired image registration. In BVM Workshop, pages 205– 210. Springer, 2024. [80]

2. S. Neuber and J. Modersitzki. A multigrid approach for fourth-order equations in image registration. In International Conference on Scale Space and Variational Methods in Computer Vision, pages 391–403. Springer, 2025. [81]

The contributions of the two publications are the following:

- We were the first to propose a three-dimensional variational image registration problem with second-order DC regularization in [80].
- We presented a novel combination of landmark-based with intensity-based registration in [80]. In contrast to [28], we followed a non-parametric approach for the intensity-based registration.
- We proposed a new multigrid approach for the fourth-order PDE, obtained from image registration with second-order DC regularization in [81]. More precisely, we presented a discretization with Dirichlet-0 boundary conditions on a staggered grid and verified with LFA its suitability for multigrid. In particular, we showed the h-ellipticity and determine the local smoothing factor of an ω-relaxed Jacobi iteration explicitly for the two-dimensional setting.

In addition to the publications, the contributions of this thesis are the following:

- We determine the natural boundary conditions for second-order DC regularization and explore its kernel for a parametrization with second-order polynomials.
- We propose a second-order DC regularizer with sliding boundary conditions and determine its kernel and a spectral decomposition of its Gâteaux derivative.
- We extend the LFA from our publication [81] to the three-dimensional setting.
- Numerous numerical experiments support the theoretical finding of this thesis.

The thesis has the following structure:

In Ch. 2, we give an introduction to variational image processing, in particular we present a variational image registration approach with second-order DC regularization.

In Ch. 3, we examine second-order DC regularization with natural boundary conditions. We explicitly determine the natural boundary conditions of second-order DC regularization and study the kernel for a parametrization with second-order polynomials.

In Ch. 4, we propose a second-order DC regularizer restricted to the space of displacements with sliding boundary conditions. We determine its kernel and a spectral decomposition of its Gâteaux derivative.

In Ch. 5, we propose a discretize-then-optimize approach for the considered variational image registration problem. We present a discretization of the second-order DC regularizer with sliding, Dirichlet-0 and Neumann-0 boundary conditions. We examine the consistency and convergence of the discretizations in numerical experiments.

In Ch. 6, we outline the main idea of multigrid methods and, in particular, give an introduction to LFA.

In Ch. 7, we present a multigrid framework for variational problems with second-order DC regularization. The framework is based on the proposed discretization of the second-order DC regularizer with Dirichlet-0 boundary conditions. Furthermore, we provide a LFA of the proposed multigrid framework to obtain a solid theoretical foundation for

multigrid. Numerical experiments illustrate the theoretical findings. A comparison with a multigrid framework for a variational problem with the elastic regularizer enables to classify the results of the proposed framework.

2 Variational Image Registration with Second-Order Divergence-Curl Regularization

Variational methods [15] are a standard approach for image registration, see e.g. [21, 1, 7, 10], as they are very versatile and offer the opportunity to model a priori knowledge. Since image registration is inherently ill-posed and the data is typically noise-corrupted, regularization is inevitable for the existence of a solution of the image registration problem; see e.g. [11]. In this thesis, we study a concrete regularizer, more precisely a family of second-order divergence-curl (DC) regularizers [58].

The chapter is organized as follows: First, we formulate a variational image registration problem in Sec. 2.1. Second, we present different regularization approaches and focus on a family of second-order DC regularizers in Sec. 2.2.

2.1 An Introduction to Variational Image Registration

For the formulation of the image registration problem, we briefly define its components. As the focus of this thesis is on a certain form of regularization, we only cover the components, that are relevant in the following. For a more detailed introduction to image registration and further options, we refer to e.g. [1, 7, 2, 3]. Here, we follow the notation and the ideas of [1, 5].
In this thesis, we use the following common symbols for functions spaces:

- $C^k(\mathbb{R}^p, \mathbb{R}^q)$ for the space of k times continuously differentiable functions from $\mathbb{R}^p$ to $\mathbb{R}^q$,
- $L^k(\mathbb{R}^p, \mathbb{R}^q)$ for the Lebesque space, i.e. the space of measurable functions from $\mathbb{R}^p$ to $\mathbb{R}^q$ for which the p-th power of the absolute value is Lebesgue integrable,
- $H^k(\mathbb{R}^p, \mathbb{R}^q)$ for the Sobolev space of k times weakly differentiable functions from $\mathbb{R}^p$ to $\mathbb{R}^q$.

In image registration we have given two images. In this thesis, we assume a continuous model for images, as this allows for a convenient theoretical analysis of the image registration problem. We use the definition from [1] and make some further assumptions to simplify the following image registration framework.

Definition 2.1.1 (Images, modified Modersitzki [1, Def. 3.1])**.** Let $\Omega = (0,1)^d$ with dimension $d = 2, 3$ be a domain, i.e. a bounded, connected and open subset. The boundary of the domain Ω is denoted by $\partial\Omega$. A function $\mathrm{Img} \in C^2(\mathbb{R}^d, \mathbb{R})$ is called a d-dimensional image, if

1. Img is compactly supported in Ω,
2. $0 \leq \mathrm{Img}(x) < \infty$ for all $x \in \mathbb{R}^d$,
3. $\int_{\mathbb{R}^d} (\mathrm{Img}(x))^k \mathrm{d}x$ is finite for $k > 0$.

The set of all images is denoted by

$$\mathcal{I}(d) := \{\text{Img} \in C^2(\mathbb{R}^d, \mathbb{R}) | \text{Img is } d\text{-dimensional image}\}.$$

The two given images are the so called template image $\mathcal{T} \in \mathcal{I}(d)$ and the reference image $\mathcal{R} \in \mathcal{I}(d)$. The goal is an alignment of these images. In particular, to find a reasonable transformation, such that the reference and the transformed template image are as similar as possible in a certain sense. A transformation is defined as follows.

Definition 2.1.2 (Transformations and Displacements, Modersitzki [1, p. 77])**.** Let $\Omega = (0,1)^d$ be a domain. A transformation is a vector-field $y : \Omega \to \mathbb{R}^d$. For a given transformation, the vector-field u with $u(x) = y(x) - x$ is called displacement or deformation.

The similarity between the reference image $\mathcal{R}$ and the transformed template image $\mathcal{T}[y] := \mathcal{T}(y)$ is measured by a so called data-fitting term $\mathcal{D}$. For simplicity, we fix the data-fitting term to the popular sum-of-squared-differences [82, 46, 74].

Definition 2.1.3 (Sum-of-Squared-Differences (SSD), Modersitzki [1, Def. 6.1])**.** Let $d \in \mathbb{N}$ and $\mathcal{T}, \mathcal{R} \in \mathcal{I}(d)$ be images according to Def. 2.1.1. For a transformation $y : \Omega \to \mathbb{R}^d$, see Def. 2.1.2, we define the data-fitting term sum-of-squared-differences (SSD) by

$$\mathcal{D}^{\text{SSD}}(\mathcal{T}, \mathcal{R}; y) = \frac{1}{2} \int_\Omega (\mathcal{T}[y](x) - \mathcal{R}(x))^2 \mathrm{d}x.$$

where $\mathcal{T}[y]$ is the transformed image with $\mathcal{T}[y](x) := \mathcal{T}(y(x))$ for all $x \in \Omega$.

SSD intuitively describes the similarity between the reference and the transformed template image [83]. One of the reasons for the popularity of SSD is due to its simple mathematical structure. SSD is based on an L^2-norm. Hence, within the optimization we obtain a least-square problem, which is a well understood class of problems; see e.g. [53, §10]. The disadvantage of SSD is that the gray values of the corresponding points must match [5], which might not be fulfilled e.g. for multimodal image registration. For an overview about other data-fitting terms in image registration, see for example [84, §2] or [5, §7].

A variational formulation based on the data-fitting term alone is ill-posed in the sense of Hadamard [85]; see [11]. Hence, further assumptions about the transformations are needed in the problem formulation. This allows for the formulation 'reasonable' transformation. These assumptions can be divided into parametric and non-parametric approaches; see e.g. [1].
The idea of a parametric image registration approach is to restrict the space of admissible transformations. Within parametric registration, a transformations only depends on a finite number of parameters. We refer to [1, §§4-7] for a discussion of parametric image registration.
Non-parametric image registrations assume a certain smoothness of the deformation in form of a regularizer $\mathcal{S}$, to ensure the existence of a solution of the registration problem [1, p. 78]. Moreover, the regularizer enables the inclusion of apriori knowledge and thus a preference for certain solutions, e.g. obtained from a physical model. In this thesis we study non-parametric registration approaches, in particular on registration with

second-order DC regularizers. Hence, we discuss regularization strategies in Sec. 2.2 in more detail. .

Altogether, the considered image registration problem reads as follows.

Definition 2.1.4 (Image Registration Problem)**.** For a given template and reference image $\mathcal{T}, \mathcal{R} \in \mathcal{I}(d), d = 2, 3$ the image registration problem is formulated as a minimization of the joint functional

$$\mathcal{J} : H^k(\Omega, \mathbb{R}^d) \to \mathbb{R}, \quad \mathcal{J}(y) = \mathcal{D}^{\text{SSD}}(\mathcal{T}, \mathcal{R}; y) + \alpha \mathcal{S}(y). \tag{1}$$

The joint functional $\mathcal{J}$ consists of the data-fitting term SSD $\mathcal{D}^{\text{SSD}}$ from Def. 2.1.3 and a regularizer $\mathcal{S}$. Moreover, $y \in H^k(\Omega, \mathbb{R}^d)$ is the desired transformation. The Sobolev space $H^k(\Omega, \mathbb{R}^d)$, $k \in \mathbb{N}$ is chosen, such that the regularizer $\mathcal{S}$ is defined. The regularization parameter $\alpha \in \mathbb{R}_{\geq 0}$ weights the influence of the regularizer.

2.2 Regularization Strategies

The aim of regularization is not only to ensure the existence of a solution y of the registration problem (1), but furthermore to model apriori knowledge, such as physical properties, about the transformation y. This section outlines several common regularization strategies, that are relevant in the following. For the sake of simplicity, we assume, that the transformations y are sufficiently (weakly) differentiable so that the regularizers can be defined. We present a family of second-order DC regularizers in more detail, as this form of regularization is the overarching topic of this thesis.

2.2.1 An Overview of Common Regularization Strategies

For an overview of regularization strategies in image registration, we refer to [1, §§9-12] and [14]. Tab. 1 lists all regularization strategies, that are explicitly mentioned in this thesis. In the following, we categorize and discuss the regularizers from Tab. 1.

Table 1: *An overview of common regularization strategies, which are all of the form* $\frac{1}{2}\int_\Omega I(y)\mathrm{d}x$ *with the integrand* $I(y)$ *as below. The parameters* $\alpha_j \in \mathbb{R}, j = 1, 2, 3$ *are typically non-negative and problem-specific. For meaningful choices, we refer to the related references. Similar, concrete choices for the scalar-fields* ϕ_c, ψ *of the hyperelastic regularizer are outlined in the corresponding publication.*

Regularizer	Integrand $I(y)$
Curvature [86]	$\sum_{i=1}^d \|\Delta y_i\|^2$
Diffusion [87]	$\sum_{i=1}^d \|\nabla y_i\|_2^2$
Elastic [88]	$\alpha_1 \|\operatorname{div} y\|_2^2 + \alpha_2 \sum_{i=1}^d \|\nabla y_i\|_2^2$
First-order DC [58]	$\alpha_1 \|\operatorname{div} y\|_2^2 + \alpha_2 \|\operatorname{curl} y\|_2^2$
Full-curvature [89]	$\sum_{i=1}^2 (\Delta y_i)^2 - 2(\partial_{11} y_i \partial_{22} y_i - (\partial_{12} y_i)^2)$
Hyperelastic [82]	$\alpha_1 \|\nabla(y - id)\|^2 + \alpha_2 \phi_c(\operatorname{cof} \nabla y) + \alpha_3 \psi(\det \nabla y)$
Second-order DC [27]	$\alpha_1 \|\nabla \operatorname{div} y\|_2^2 + \alpha_2 \|\nabla \operatorname{curl} y\|_2^2$
Total-variation [90]	$\sum_{i=1}^d \|\nabla y_i\|_1^2$

Typical choices of regularizers in image registration are given by a bi-linear form $a : H^m(\Omega, \mathbb{R}^d) \times H^m(\Omega, \mathbb{R}^d) \to \mathbb{R}$, i.e.

$$\mathcal{S}(y) = \frac{1}{2}a(y, y) := \frac{1}{2}\int_\Omega \langle \mathcal{B}y(x), \mathcal{B}y(x)\rangle \mathrm{d}x \tag{2}$$

with a differential operator $\mathcal{B}$ of order m; see [1, p. 78]. Thus, regularizers can be categorized by their order, i.e. the highest-order of the differential operator $\mathcal{B}$ of the bi-linear form (2).
Hence, diffusion, elastic, first-order DC, hyperelastic and total-variation regularization are so called first-order regularizers.
The remaining regularizers from from Tab. 1, i.e. curvature, full-curvature and second-order DC regularization, are of second order. Second-order regularizers lead to smoother transformations and thus enables the so called 2D/3D registration [91, 92]. Another advantage of the second-order regularizers is their corresponding interpolation schemes, which enables a landmark-based registration. In case of the curvature and full-curvature regularizers the thin-plate spline interpolation [93] corresponds to the regularization. The second-order DC regularizer corresponds to the vector-spline interpolation [23]. The relation between the regularizers and the interpolations enables a matching pre-registration [94, 28, 80] based on landmarks.

Furthermore, we categorize the regularizers into those that allow for large transformations and those that allow for small transformations. If only small displacements are possible, a pre-registration might be necessary to obtain a good alignment of the images. All mentioned second-order regularizers have affine linear transformations in their kernel, see [86, 89] for the curvature regularizers and Ch. 3 for the second-order DC regularizer. This makes large transformations within the registration possible. Also the hyperelastic regularizer allows for large displacements [82], whereas the elastic regularizer assumes small displacements [1, §9.1].

The elastic, hyperelastic, first- and second-order div-curl regularizers are physically inspired regularizers.The elastic regularizer builds on the theory of linear elasticity and and in particular on the Navier-Lamé-equations; see e.g. [1, §9]. The hyperelastic regularizer [82, 22] is an extension to a regularizer which builds on non-linear elasticity and furthermore controls the tissue compression and extension, i.e. the change of lengths, areas and volumes. The first and second-order DC regularizer are inspired by the physics of fluid dynamics [30, 29]. Even though, the regularizers are not based on equations that describe fluid motion, such as the Navier-Stokes equation, it can be observed, that the regularizer leads to a velocity field behavior, that is qualitatively similar to that of real fluids; see [30]. Note, that fluid-like bodies are expected to deform like honey [1, p.136], which is not a suitable model for the application of e.g. hands registration.
An advantage of these physically inspired models is, that they couple the components. If an unidirectional force is applied, a deformation in all dimensions is observed. This is a property, that is observable in many real-life scenarios. A problem is that the same material properties independent of location and direction are assumed by these regularizers [14]. A solution for this problem is e.g. to define new domains within the original image [29].
In contrast to the physically inspired regularizers, the diffusion, curvature, full-curvature and total variation regularizer do not couple the components. This enables fast numerical schemes. In particular, the diffusion regularizer is based on the Laplace equation, which is the standard partial differential equation (PDE) for many different fields; see

e.g. [31]. Also the curvature regularizer is related to the bi-Laplace/ biharmonic equation and is well understood; see e.g. [95].

Diffeomorphisms are desirable for many applications [22]. If the strain is small enough, it can be shown, that the elastic regularization leads to diffeomorphic deformations; see [1, Rem. 9.1]. The hyperelastic regularizer also ensures bijective deformations, but cannot ensure invertibility [22]. We show in Ch. 3, that the second-order DC regularizer has non-bijective transformations in its kernel.

The total-variation regularizer is the only regularization strategy, that allows for local discontinuities. For most applications a smooth deformation is desirable, but sliding motion can e.g. be observed on the boundary of organs; see [14]. This can be captured by the total-variation regularizer. A disadvantage is its non-differentiability [96], which leads to possible numerical instabilities, e.g. for gradient-based optimization.

2.2.2 A Family of Second-Order Divergence-Curl Regularizers

In this thesis, we focus on the family of second-order DC regularizers. As discussed above, second-order DC regularization was inspired by the physics of fluids [30]. This is also taken into account when looking at the interpretation of the meaning of the differential operators, based on [97, §3.4] in the following. We consider $d = 2, 3$ as these are the relevant cases for image registration.
The divergence

$$\operatorname{div} : H^1(\mathbb{R}^d, \mathbb{R}^d) \to H^0(\mathbb{R}^d, \mathbb{R}), \quad \operatorname{div} f = \partial_1 f_1 + \cdots + \partial_d f_d$$

turns a vector-field into a scalar field. The value of the divergence of a vector field at a particular point is a measure of the flux density of the vector field in or out of that point. Assuming that f is the velocity of a fluid. Then, $\operatorname{div} f = 0$ describes, that the rate of fluid flowing into the point is the same as the rate of fluid that is flowing out. For $\operatorname{div} f < 0$ the point is a sink, while it is a source for $\operatorname{div} f > 0$. Thus, the operator $\nabla \operatorname{div}$ measures a change in the diverging flow.
The curl

$$\operatorname{curl} : H^1(\mathbb{R}^3, \mathbb{R}^3) \to H^0(\mathbb{R}^3, \mathbb{R}^3), \quad \operatorname{curl} f = \begin{pmatrix} \partial_2 f_3 - \partial_3 f_2 \\ \partial_3 f_1 - \partial_1 f_3 \\ \partial_1 f_2 - \partial_2 f_1 \end{pmatrix}$$

turns a vector-field into another vector-field. We denote its application by $\operatorname{curl} f$ but the notation $\nabla \times f$ makes its relation to the cross-product $\times$ obvious. The curl measures the circulation of a vector field. Assuming again that f represents the velocity of a fluid, such as a the circulations of the lake. A dropped twig travels with the current in the lake. The curl measures how quickly and in what orientation the twig itself rotates as it moves. The operator $\nabla \operatorname{curl}$ measures a change in the rotation of the twig around its own axis. Its two-dimensional version

$$\operatorname{curl} : H^1(\mathbb{R}^2, \mathbb{R}^2) \to H^0(\mathbb{R}^2, \mathbb{R}), \quad \operatorname{curl} f = \partial_1 f_2 - \partial_2 f_1$$

turns a vector-field into a scalar field. An extension of the curl to higher dimensional spaces is not guaranteed. Only for the seven dimensional space there exist an extension of the curl [98].

Thus, the family of second-order DC regularizers penalize a change in the flux density of the vector-field and a change in the 'self circulation' of an object moved by the vector-

field. These terms are weighted by a parameter $\gamma \in [0,1]$. The family of regularizers has a second-order and enables a coupling of the components of the transformation y.

Definition 2.2.1 (Family of Second-Order Divergence-Curl Regularizers)**.** The family of second-order divergence-curl (DC) regularizers is $\{\mathcal{S}(y;\gamma) : \gamma \in [0,1]\}$ where

$$\mathcal{S} : H^2(\Omega, \mathbb{R}^d) \times [0,1] \to \mathbb{R}, \quad \mathcal{S}(y;\gamma) := \frac{1}{2}\int_\Omega \gamma \|\nabla \operatorname{div} y\|_2^2 + (1-\gamma)\|\nabla \operatorname{curl} y\|^2 \mathrm{d}x, \tag{3}$$

with

$$\|\nabla \operatorname{curl} y\|^2 := \begin{cases} \|\nabla \operatorname{curl} y\|_2^2 & \text{for } d = 2, \\ \|\nabla \operatorname{curl} y\|_F^2 & \text{for } d = 3. \end{cases}$$

The parameter γ enables to incorporate apriori knowledge about the underlying deformation; see [28]. In the following, we fix the parameter γ and use the term 'the' second-order DC regularizer.
The special case $\gamma = 0.5$, leads to a decoupling of the components. More specifically, it was shown in [23], that for functions from the Beppo-Levi space of order two and an infinite domain $\Omega = \mathbb{R}^d$ the same weighting of the summand leads to the popular thin-plate-spline [93, 99, 100] energy, which corresponds to the curvature regularizer; see [86] or [1, §12.2]. Even though we consider a finite domain $\Omega = (0,1)^d$ in our registration approach, one can prove, that the decoupling remains.

Table 2: *Differential operators* $\mathcal{B}_{\text{div}}$ *and* $\mathcal{B}_{\text{curl}}$ *for* $d = 2, 3$.

	$d=2$	$d=3$
$\mathcal{B}_{\text{div}}$	$\begin{pmatrix} \partial_{11} & \partial_{12} \\ \partial_{12} & \partial_{22} \end{pmatrix}$	$\begin{pmatrix} \partial_{11} & \partial_{12} & \partial_{13} \\ \partial_{12} & \partial_{22} & \partial_{23} \\ \partial_{13} & \partial_{23} & \partial_{33} \end{pmatrix}$
$\mathcal{B}_{\text{curl}}$	$\begin{pmatrix} -\partial_{12} & \partial_{11} \\ -\partial_{22} & \partial_{12} \end{pmatrix}$	$\begin{pmatrix} 0 & -\partial_{13} & \partial_{12} \\ \partial_{13} & 0 & -\partial_{11} \\ -\partial_{12} & \partial_{11} & 0 \\ 0 & -\partial_{23} & \partial_{22} \\ \partial_{23} & 0 & -\partial_{12} \\ -\partial_{22} & \partial_{12} & 0 \\ 0 & -\partial_{33} & \partial_{23} \\ \partial_{33} & 0 & -\partial_{13} \\ -\partial_{23} & \partial_{13} & 0 \end{pmatrix}$

Remark 2.2.2 (Bilinear Form). The differential operator $\mathcal{B}$ in the bilinear form given by (2) of the second-order DC regularizer and has the form

$$\mathcal{B} := \begin{pmatrix} \sqrt{\gamma}\mathcal{B}_{\text{div}} \\ \sqrt{1-\gamma}\mathcal{B}_{\text{curl}} \end{pmatrix},$$

where the differential operators

$$\mathcal{B}_{\text{div}} : H^2(\Omega, \mathbb{R}^d) \to H^0(\Omega, \mathbb{R}^d), \quad \mathcal{B}_{\text{div}} = \nabla \operatorname{div}$$

and $\mathcal{B}_{\text{curl}} : H^2(\Omega, \mathbb{R}^2) \to H^2(\Omega, \mathbb{R}^2)$ for $d = 2$ but $\mathcal{B}_{\text{curl}} : H^2(\Omega, \mathbb{R}^3) \to H^2(\Omega, \mathbb{R}^9)$ for $d = 3$ with

$$\mathcal{B}_{\text{curl}} = \nabla \operatorname{curl}$$

are defined according to Tab. 2.

Note, that the operator $\mathcal{B}_{\text{div}}$ is the Hessian matrix.

2.2.3 Conclusion

In this section, we discussed different regularization strategies. It becomes clear, that the second-order DC regularizer has many desirable properties for image registration: It is, obviously, of second-order and L_2-norm based and hence provides smooth deformation. Furthermore, there exists a corresponding interpolation energy, which enables a matching landmark-based pre-registration. The second-order DC regularizer models transformation y as fluids, which makes an interpretation possible. Moreover, the physical motivation leads to a coupling in the components, which is an appropriate model for e.g. most medical applications. The ability to weight the divergence and curl part separately was demonstrated to be advantageous for image registration e.g. in electrophoresis [28] or in thoracic or pulmonary registration [29]. Nevertheless a decoupling is possible and leads to a form closely related to the well-understood curvature regularization.

3 Second-Order Divergence-Curl Regularization

In this chapter, we study the family of second-order divergence-curl (DC) regularizers from Def. 2.2.1. We assume a fixed parameter $\gamma \in [0, 1]$ and use the term "the" second-order DC regularizer.

The natural boundary conditions obtained from the Euler-Lagrange equation of the image registration problem from Def. 2.1.4 are implicit. Thus, they do not change the regularization, but provide additional information about suitable transformations. They are the optimality conditions for a transformation on the boundary. The knowledge of the natural boundary is in particular important for the discretization of the image registration problem. Other boundary conditions may change the space of admissible transformations, such that the real minimizer is excluded.
To the best of our knowledge, we are the first to determine the natural boundary conditions for the second-order DC regularizer. The natural boundary conditions of the second-order DC regularizer in Thm. 3.1.2/Thm. 3.1.3 are more complex compared e.g. to the elastic [1, Thm. 9.4] or the curvature regularizer [1, Thm. 12.1] due to the second order and the coupling of the components.

Transformations in the kernel of the regularizer must be captured, if they are relevant for the considered image registration, as the regularizer does not penalize them. E.g. non-bijective transformations are not desirable in the context of image registration, since a convolution of the transformation suggests that a certain area/volume disappears, which would generally not be physically plausible.
We investigate the kernel of the second-order DC regularizer under the assumption, that the considered transformations can be parameterized by second-order polynomials. We observe certain second-order polynomials in the kernel of the second-order DC regularizer with natural boundary conditions. This suggests that bijective transformations cannot be ensured by second-order DC regularization.

This chapter is structured as follows: First, we present the necessary condition for a minimizer of the considered image registration problem and derive the natural boundary conditions for this regularizer in Sec. 3.1. Second, we determine the kernel of the second-order DC regularizer for a parametrization with second-order polynomials in Sec. 3.2.

3.1 Deduction of the Natural Boundary Conditions

A necessary condition for a local minimizer y of the image registration problem from Def. 2.1.4 is a vanishing Gâteaux derivative. The resulting condition is the so called Euler-Lagrange equation, that is presented first. From this, we deduce the natural boundary conditions of the second-order DC regularizer from Def. 2.2.1, that are the intrinsic boundary conditions of the corresponding differential operators.

3.1.1 Euler-Lagrange Equation of the Considered Image Registration Functional

To determine the Euler-Lagrange equation we determine the Gâteaux derivative of the image registration functional in a first step. Due to the linearity of the Gâteaux derivative, we determine the Gâteaux derivative of the data-fitting term and the regularizer from Eq. (1) separately.

Let $\mathcal{T}, \mathcal{R} \in \mathcal{I}(d)$. The Gâteaux derivative of the data-fitting term SSD from Def. 2.1.3 is

$$\begin{aligned} d\mathcal{D}^{\mathrm{SSD}}(\mathcal{T}, \mathcal{R}; y; v) &= \lim_{h \to 0} \frac{1}{h} (\mathcal{D}^{\mathrm{SSD}}(\mathcal{T}, \mathcal{R}; y + hv) - \mathcal{D}^{\mathrm{SSD}}(\mathcal{T}, \mathcal{R}; y)) \\ &= \int_\Omega \langle (\mathcal{T}(y) - \mathcal{R}) \nabla \mathcal{T}(y), v \rangle \mathrm{d}x; \end{aligned}$$

see [1, Thm. 8.1] for details.

Lemma 3.1.1 (Gâteaux Derivative of the Second-Order DC Regularizer)**.** The Gâteaux derivative of the second-order DC regularizer from Def. 2.2.1 is

$$d\mathcal{S}(y; v) = \int_\Omega \gamma \langle \mathcal{B}_{\mathrm{div}}\, y, \mathcal{B}_{\mathrm{div}}\, v \rangle + (1 - \gamma) \langle \mathcal{B}_{\mathrm{curl}}\, y, \mathcal{B}_{\mathrm{curl}}\, v \rangle \mathrm{d}x$$

Proof. With the definition of the Gâteaux derivative and of the second-order DC regularizer, we obtain

$$\begin{aligned} d\mathcal{S}(y; v) = \lim_{h \to 0} \frac{1}{2h} \Bigg(& \gamma \int_\Omega \langle \mathcal{B}_{\mathrm{div}}\,(y + hv), \mathcal{B}_{\mathrm{div}}\,(y + hv) \rangle - \langle \mathcal{B}_{\mathrm{div}}\, y, \mathcal{B}_{\mathrm{div}}\, y \rangle \, \mathrm{d}x \\ & + (1 - \gamma) \int_\Omega \langle \mathcal{B}_{\mathrm{curl}}\,(y + hv), \mathcal{B}_{\mathrm{curl}}\,(y + hv) \rangle - \langle \mathcal{B}_{\mathrm{curl}}\, y, \mathcal{B}_{\mathrm{curl}}\, y \rangle \mathrm{d}x \Bigg), \end{aligned}$$

where the differential operators $\mathcal{B}_{\mathrm{div}}, \mathcal{B}_{\mathrm{curl}}$ are defined according to Tab. 2. Without loss of generality we set $\gamma = 1$ and concentrate on the divergence-part. With the linearity of the scalar product, we have

$$\begin{aligned} d\mathcal{S}(y; v) &= \lim_{h \to 0} \frac{1}{2h} \left(\int_\Omega \langle \mathcal{B}_{\mathrm{div}}\,(y + hv), \mathcal{B}_{\mathrm{div}}\,(y + hv) \rangle - \langle \mathcal{B}_{\mathrm{div}}\, y, \mathcal{B}_{\mathrm{div}}\, y \rangle \mathrm{d}x \right) \\ &= \lim_{h \to 0} \frac{1}{2h} \left(\int_\Omega 2h \langle \mathcal{B}_{\mathrm{div}}\, y, \mathcal{B}_{\mathrm{div}}\, v \rangle + h^2 \langle \mathcal{B}_{\mathrm{div}}\, v, \mathcal{B}_{\mathrm{div}}\, v \rangle \mathrm{d}x \right) \\ &= \int_\Omega \langle \mathcal{B}_{\mathrm{div}}\, y, \mathcal{B}_{\mathrm{div}}\, v \rangle \mathrm{d}x. \end{aligned}$$

With $\gamma = 0$ and the same arguments for the curl-part, we obtain the Gâteaux derivative. □

As the Gâteaux derivative need to vanish for a minimizer of the image registration functional, we obtain the so called Euler-Lagrange equation in its weak form

$$\begin{aligned} d\mathcal{J}(y; v) :&= \lim_{h \to 0} \frac{1}{h} (\mathcal{J}(y + hv) - \mathcal{J}(y)) \\ &= d\mathcal{D}^{\mathrm{SSD}}(\mathcal{T}, \mathcal{R}; y; v) + \alpha \, d\mathcal{S}(y; v) = 0 \quad \forall v \in H^2(\Omega, \mathbb{R}^d). \end{aligned}$$

The Euler-Lagrange equation in its strong formulation is

$$(\mathcal{T}(y) - \mathcal{R}) \nabla \mathcal{T}(y) + \alpha \mathcal{A}\, y = 0 \quad y \in H^4(\Omega, \mathbb{R}^d) \tag{4}$$

with the differential-operator $\mathcal{A} : H^4(\Omega, \mathbb{R}^d) \to H^0(\Omega, \mathbb{R}^d)$

$$\mathcal{A} := \gamma \mathcal{A}_{\text{div}} + (1-\gamma)\mathcal{A}_{\text{curl}} := \gamma(\mathcal{B}_{\text{div}})^\top \mathcal{B}_{\text{div}} + (1-\gamma)(\mathcal{B}_{\text{curl}})^\top \mathcal{B}_{\text{curl}},$$

where $\mathcal{A}_{\text{div}}, \mathcal{A}_{\text{curl}}$ can be found in Tab. 3. As the focus of this section is to deduce the natural boundary conditions and the concept is well-known for regularizers with a bilinear form, see e.g. [1, p. 78], we keep the arguments rather short and refer to e.g. [101, §17.3]. The strong formulation is obtained through partial integration and by restricting the perturbations to those that vanish on the boundary $\partial\Omega$. The natural boundary conditions that arise from this formulation are given in Thm. 3.1.2 for the two-dimensional and in Thm. 3.1.3 for the three-dimensional regularizer.

Table 3: *Differential operators* $\mathcal{A}_{\text{div}}, \mathcal{A}_{\text{curl}} : H^4(\Omega, \mathbb{R}^d) \to H^0(\Omega, \mathbb{R}^d)$ *for* $d = 2, 3$.

For $d = 2$:

$$\begin{aligned}\mathcal{A}_{\text{div}} &= \begin{pmatrix} \partial_{1111}+\partial_{1122} & \partial_{1112}+\partial_{1222} \\ \partial_{1112}+\partial_{1222} & \partial_{1122}+\partial_{2222} \end{pmatrix}, \\ \mathcal{A}_{\text{curl}} &= \begin{pmatrix} \partial_{1122}+\partial_{2222} & -(\partial_{1112}+\partial_{1222}) \\ -(\partial_{1112}+\partial_{1222}) & \partial_{1111}+\partial_{1122} \end{pmatrix}.\end{aligned}$$

For $d = 3$:

$$\begin{aligned}\mathcal{A}_{\text{div}} &= \begin{pmatrix} \partial_{1111}+\partial_{1122}+\partial_{1133} & \partial_{1112}+\partial_{1222}+\partial_{1233} & \partial_{1113}+\partial_{1223}+\partial_{1333} \\ \partial_{1112}+\partial_{1222}+\partial_{1233} & \partial_{1122}+\partial_{2222}+\partial_{2233} & \partial_{1123}+\partial_{2223}+\partial_{2333} \\ \partial_{1113}+\partial_{1223}+\partial_{1333} & \partial_{1123}+\partial_{2223}+\partial_{2333} & \partial_{1133}+\partial_{2233}+\partial_{3333} \end{pmatrix}, \\ \mathcal{A}_{\text{curl}} &= \begin{pmatrix} \partial_{1122}+\partial_{1133}+\partial_{2222}+2\partial_{2233}+\partial_{3333} & -(\partial_{1112}+\partial_{1222}+\partial_{1233}) & -(\partial_{1113}+\partial_{1223}+\partial_{1333}) \\ -(\partial_{1112}+\partial_{1222}+\partial_{1233}) & \partial_{1111}+2\partial_{1133}+\partial_{1122}+\partial_{2233}+\partial_{3333} & -(\partial_{1123}+\partial_{2223}+\partial_{2333}) \\ -(\partial_{1113}+\partial_{1223}+\partial_{1333}) & -(\partial_{1123}+\partial_{2223}+\partial_{2333}) & \partial_{1111}+2\partial_{1122}+\partial_{2222}+\partial_{1133}+\partial_{2233} \end{pmatrix}.\end{aligned}$$

3.1.2 Natural Boundary Conditions of the Second-Order Divergence-Curl Regularizer

We present the natural or also called implicit boundary conditions of the second-order DC regularizer, which result from the Euler-Lagrange equation (4). Therefore, we need the set $\partial\partial\Omega$, that is visualized in Fig. 3.1.1. Even though the curl has a completely different structure for the two- compared to the three-dimensional case, a transfer of the natural boundary conditions from the three-dimensional findings to the two-dimensional findings is possible.

Natural Boundary Conditions of the Two-Dimensional Regularizer

We first consider the two-dimensional second-order DC regularizer and its natural boundary conditions and outline the idea of the proof in detail. An extension to the three-dimensional case with a less detailed proof can be found in the following paragraph.

Theorem 3.1.2. Let $n = (n_1, n_2)^\top \in \mathbb{R}^2$ be the outer normal vector on $\partial\Omega$. The natural boundary conditions, that arise from the Euler-Lagrange equation (4) with

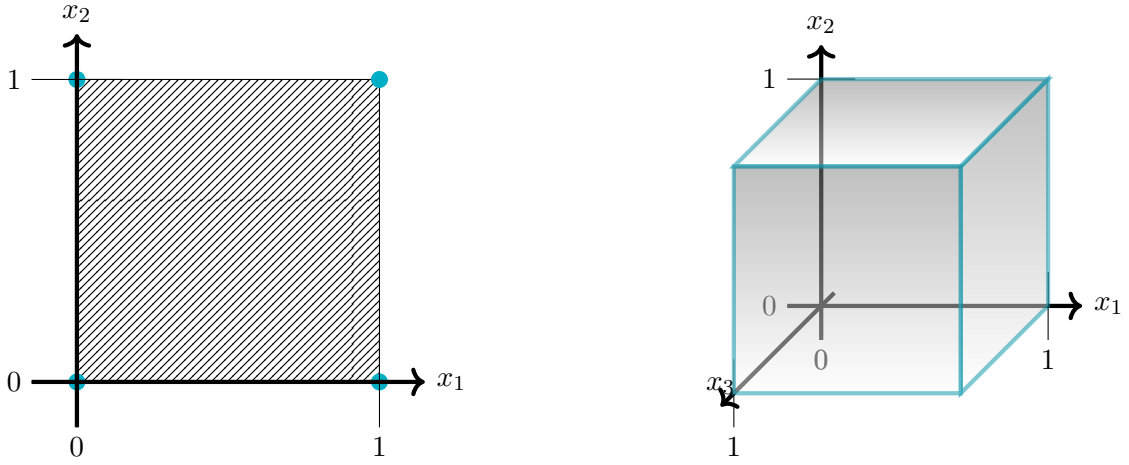

Figure 3.1.1: *Visualization of $\partial\partial\Omega$ for $d = 2$ (left) and $d = 3$ (right).*

$\mathcal{A}$ according to Tab. 3 and $y \in C^4(\overline{\Omega}, \mathbb{R}^d)$ are

$$\begin{aligned}
\Delta\left(\gamma \operatorname{div}(y)\begin{pmatrix} n_1 \\ n_2 \end{pmatrix} + (1-\gamma)\operatorname{curl}(y)\begin{pmatrix} -n_2 \\ n_1 \end{pmatrix}\right) &= 0 \quad \text{on } \partial\Omega, \\
\partial_n\left(\gamma \begin{pmatrix} \partial_1 \\ \partial_2 \end{pmatrix}\operatorname{div}(y) + (1-\gamma)\begin{pmatrix} -\partial_2 \\ \partial_1 \end{pmatrix}\operatorname{curl}(y)\right) &= 0 \quad \text{on } \partial\Omega, \\
\partial_n\left(\gamma \operatorname{div}(y)\begin{pmatrix} n_1 \\ n_2 \end{pmatrix} + (1-\gamma)\operatorname{curl}(y)\begin{pmatrix} -n_2 \\ n_1 \end{pmatrix}\right) &= 0 \quad \text{on } \partial\partial\Omega.
\end{aligned}$$

Proof. The proof is based on the following scheme: We assume $y \in C^4(\overline{\Omega}, \mathbb{R}^d)$ to be a minimizer of the minimization problem from Def. 2.1.4. Thus, the Euler-Lagrange equation (4) needs to hold. We determine the first variation of the problem (1), perform partial integration, cancel out the summand from the Euler-Lagrange equations (4) and deduce the natural boundary conditions.
The Gâteaux derivative of the joint functional (1) with second-order DC regularization is

$$\begin{aligned}
0 &= \lim_{h\to 0} \mathcal{J}(y + hv) - \mathcal{J}(y) \\
&= \int_\Omega \langle (\mathcal{T}(y(x)) - \mathcal{R}(x))\nabla\mathcal{T}(y(x)), v(x)\rangle \mathrm{d}x \\
&\quad + \alpha\left(\gamma \int_\Omega \langle \nabla \operatorname{div} y, \nabla \operatorname{div} v\rangle \mathrm{d}x + (1-\gamma)\int_\Omega \langle \nabla \operatorname{curl} y, \nabla \operatorname{curl} v\rangle \mathrm{d}x\right).
\end{aligned} \tag{5}$$

This follows directly from the bilinear form of the regularizer; see Lemma 3.1.1.
For the ease of presentation we first focus on the curl-part, i.e. we set $\gamma = 0$. We use the representation

$$\langle \nabla \operatorname{curl} y, \nabla \operatorname{curl} v\rangle = \sum_{k=1}^{2} \sum_{i,j=1,2} (-1)^{(i+j)} \partial_{m_{3-i}k} y_{m_i} \partial_{m_{3-j}k} v_{m_j}$$

with

$$m := (m_1, m_2) := (1, 2).$$

This representation may seem unnecessarily complicated, but allows a direct extension to the three-dimensional case; see Thm. 3.1.3. Without loss of generality we consider one arbitrary summand. For the ease of presentation we use the substitutions

$$a := m_{3-i}, \quad \alpha := m_i, \quad b := m_{3-j} \quad \text{and } \beta := m_j.$$

Partial integration yields

$$\begin{aligned}
&\int_\Omega \partial_a \partial_k y_\alpha \partial_b \partial_k v_\beta \mathrm{d}x \\
&= \int_{\partial\Omega} \partial_a \partial_k y_\alpha n_b \partial_k v_\beta \mathrm{d}S(x) - \int_\Omega \partial_a \partial_b \partial_k y_\alpha \partial_k v_\beta \mathrm{d}x \\
&= \int_{\partial\Omega} \partial_a \partial_k y_\alpha n_b \partial_k v_\beta \mathrm{d}S(x) - \int_{\partial\Omega} \partial_a \partial_b \partial_k y_\alpha n_k v_\beta \mathrm{d}S(x) + \int_\Omega \partial_a \partial_b \partial_k \partial_k y_\alpha v_\beta \mathrm{d}x, \qquad (6)
\end{aligned}$$

where $n = (n_1, n_2)^\top$ denotes the outer normal vector on $\partial\Omega$. As we have the domain $\Omega = (0, 1)^2$, the outer normal vector n is a unit vector $\pm e_j, j = 1, 2$. Furthermore, we obtain

$$\begin{aligned}
&\int_\Omega \partial_a \partial_k y_\alpha \partial_b \partial_k v_\beta \mathrm{d}x \\
&= \int_{\partial\Omega} \partial_a \partial_k y_\alpha n_k \partial_b v_\beta \mathrm{d}S(x) - \int_\Omega \partial_b \partial_k \partial_k y_\alpha \partial_b v_\beta \mathrm{d}x \\
&= \int_{\partial\Omega} \partial_a \partial_k y_\alpha n_k \partial_b v_\beta \mathrm{d}S(x) - \int_{\partial\Omega} \partial_b \partial_k \partial_k y_\alpha n_b v_\beta \mathrm{d}S(x) + \int_\Omega \partial_a \partial_b \partial_k \partial_k y_\alpha v_\beta \mathrm{d}x. \qquad (7)
\end{aligned}$$

Thus, for the integral over Ω we use

$$\int_\Omega \sum_{k=1}^{2} \sum_{i,j=1,2} (-1)^{(i+j)} \partial_{m_{3-i}} \partial_{m_{3-j}} \partial_k \partial_k y_{m_i} v_{m_{3-j}} \mathrm{d}x = \int_\Omega \langle \mathcal{A}_{\mathrm{curl}}\, y, v \rangle \mathrm{d}x$$

and we can reformulate Eq. (5) to

$$0 = \underbrace{\int_\Omega \langle (\mathcal{T}(y(x)) - \mathcal{R}(x)) \nabla \mathcal{T}(y(x)), v(x) \rangle \mathrm{d}x \; + \alpha \int_\Omega \langle \mathcal{A}_{\mathrm{curl}}\, y, v \rangle \mathrm{d}x}_{=0 \text{ due to the Euler-Lagrange equation (4)}} + \alpha \int_{\partial\Omega} R(x) \mathrm{d}S(x) \qquad (8)$$

for some $R(x)$ as discussed above in Eq. (6) and Eq. (7). We take a closer look, at $R(x)$ on $\partial\Omega$, as this yields the natural boundary conditions.
First, we consider perturbations v with a derivative that vanishes on the boundary, i.e. the first summand of Eq. (6) and Eq. (8) vanishes. We re-substitute and deduce from Eq. (6)

$$\int_{\partial\Omega} \sum_{k=1}^{2} \sum_{i,j=1,2} (-1)^{(i+j)} \partial_{m_{3-i}} \partial_{m_{3-j}} \partial_k y_{m_i} n_k v_{m_j} \mathrm{d}S(x) = 0.$$

From this, the conditions

$$0 = \partial_n \begin{pmatrix} -\partial_2 \\ \partial_1 \end{pmatrix} \operatorname{curl} y \quad \text{on } \partial\Omega$$

follow. With similar arguments we obtain from Eq. (7)

$$\int_{\partial\Omega} \sum_{k=1}^{2} \sum_{i,j=1,2} (-1)^{(i+j)} \partial_{m_{3-j}} \partial_k \partial_k y_{m_i} n_{m_{3-j}} v_{m_j} \mathrm{d}S(x) = 0 \tag{9}$$

and thus the conditions

$$0 = \Delta \operatorname{curl}(y) \begin{pmatrix} -n_2 \\ n_1 \end{pmatrix} \quad \text{on } \partial\Omega.$$

The remaining summand from Eq. (6) is

$$\int_{\partial\Omega} \partial_a \partial_k y_\alpha n_b \partial_k v_\beta \mathrm{d}S(x)$$

and the remaining summand from Eq. (7) is

$$\int_{\partial\Omega} \partial_a \partial_k y_\alpha n_k \partial_b v_\beta \mathrm{d}S(x).$$

We further extract the boundary conditions over $\partial\partial\Omega$ from these summands as they need to vanish too. For a visualization of $\partial\partial\Omega$ see Fig. 3.1.1. We define $\Sigma := [0, 1]$ and consider the boundary $\partial\Omega$. The boundary consists of four lines and without loss of generality we consider $\Gamma = \{0\} \times \Sigma$ with the outer normal vector $n = (-1, 0)^\top$. From [102, p. 357] we know, that it holds

$$\int_\Gamma \partial_a \partial_k y_\alpha n_b \partial_k v_\beta \mathrm{d}S(x) = \int_\Sigma \partial_a \partial_k y_\alpha(0, x_2) n_b \partial_k v_\beta(0, x_2) \mathrm{d}x_2.$$

Then we apply partial integration and have

$$\begin{aligned} \int_\Sigma \partial_a \partial_k y_\alpha(0, x_2) n_b \partial_k v_\beta(0, x_2) \mathrm{d}x_2 &= \int_{\partial\Sigma} \partial_a \partial_k y_\alpha(0, x_2) n_b n_k v_\beta(0, x_2) \mathrm{d}S(x_2) \\ &\quad - \underbrace{\int_\Sigma \partial_b \partial_k \partial_k y_\alpha(0, x_2) n_b v_\beta(0, x_2) \mathrm{d}x_2}_{=0 \text{ due to Eq. (9)}} - \int_\Sigma \partial_a \partial_k y_\alpha(0, x_2) \partial_k n_b v_\beta(0, x_2) \mathrm{d}x_2. \end{aligned}$$

With a perturbation v which is zero on $\partial\Sigma$, the last summand needs to vanish. We have

$$\begin{aligned} \int_\Sigma \partial_a \partial_k y_\alpha(0, x_2) \partial_k n_b v_\beta(0, x_2) \mathrm{d}x_2 &= \int_{\partial\Sigma} \partial_a \partial_k y_\alpha(0, x_2) n_k n_b v_\beta(0, x_2) \mathrm{d}S(x_2) \\ &\quad - \underbrace{\int_\Sigma \partial_b \partial_k \partial_k y_\alpha(0, x_2) n_b v_\beta(0, x_2) \mathrm{d}x_2}_{=0 \text{ due to Eq. (9)}} - \int_\Sigma \partial_a \partial_k y_\alpha(0, x_2) n_b \partial_k v_\beta(0, x_2) \mathrm{d}x_2. \end{aligned}$$

Within the last term the last summand needs to vanish and furthermore we obtain

$$\sum_{k=1}^{2} \sum_{i,j=1,2} (-1)^{(i+j)} \int_{\partial\Sigma} \partial_{m_{3-i}} \partial_k y_{m_i}(0, x_2) n_k n_{m_{3-j}} v_{m_j}(0, x_2) \mathrm{d}S(x_2) = 0.$$

The same arguments are repeated for all other lines of $\partial\Omega$. Thus, we get the conditions

$$0 = \partial_n \operatorname{curl}(y) \begin{pmatrix} -n_2 \\ n_1 \end{pmatrix} \text{ on } \partial\partial\Omega.$$

The divergence-part is obtained through the same argumentation with $\gamma = 1$ and with the representation

$$\langle \nabla \operatorname{div} y, \nabla \operatorname{div} v \rangle = \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \partial_{ij} y_j \partial_{ik} v_k.$$

From this, we get the conditions

$$\begin{aligned} \mathcal{A}_{\operatorname{div}} y &= 0 \text{ on } \Omega, \\ \Delta \operatorname{div}(y) n &= 0 \text{ on } \partial\Omega, \\ \partial_n \nabla \operatorname{div}(y) &= 0 \text{ on } \partial\Omega \text{ and} \\ \partial_n \operatorname{div}(y) n_k &= 0 \text{ on } \partial\partial\Omega. \end{aligned}$$

□

Due to the second order of the regularizer, we obtain natural boundary conditions that require rather smooth transformations on the boundary $\partial\Omega$ and $\partial\partial\Omega$. This makes the boundary conditions more complicated compared to the natural boundary conditions of the elastic regularizer, that are only formulated on $\partial\Omega$ and consist of first-order derivatives of the deformation. Moreover, the coupling in the components of the second-order DC regularizer is also reflected in the coupling of the natural boundary conditions.

Natural Boundary Conditions of the Three-Dimensional Regularizer

We extend the results of the previous paragraph to the three-dimensional second-order DC regularizer.

Theorem 3.1.3. Let $n \in \mathbb{R}^3$ be the outer normal vector on $\partial\Omega$ and $\times$ be the cross product. The natural boundary conditions, that arise from the Euler-Lagrange equation (4) with $\mathcal{A}$ according to Tab. 3 and $y \in C^4(\overline{\Omega}, \mathbb{R}^d)$ are

$$\begin{aligned} \Delta(\gamma \operatorname{div}(y) n + (1-\gamma)(\operatorname{curl} y \times n)) &= 0 \quad \text{on } \partial\Omega, \\ \partial_n(\gamma \nabla \operatorname{div}(y) + (1-\gamma) \operatorname{curl}(\operatorname{curl}(y))) &= 0 \quad \text{on } \partial\Omega, \\ \partial_n(\gamma \operatorname{div}(y) n + (1-\gamma)(\operatorname{curl} y \times n))) &= 0 \quad \text{on } \partial\partial\Omega. \end{aligned}$$

Proof. The proof follows the same scheme as for $d = 2$; see Thm. 3.1.2. For the three-

dimensional setting, we use the representation

$$\langle \nabla \operatorname{curl} y, \nabla \operatorname{curl} v \rangle = \sum_{k=1}^{3} \sum_{m \in M} \sum_{i,j=1,2} (-1)^{(i+j)} \partial_{m_{3-i}k} y_{m_i} \partial_{m_{3-j}k} v_{m_j}$$

of the curl-part with

$$M = \{(2,3),(3,1),(1,2)\}.$$

The representation of the divergence-part

$$\langle \nabla \operatorname{div} y, \nabla \operatorname{div} v \rangle = \sum_{i=1}^{3} \sum_{j=1}^{3} \sum_{k=1}^{3} \partial_{ij} y_j \partial_{ik} v_k.$$

is a direct extension of the two-dimensional representation.
To obtain the boundary conditions over the set $\partial\partial\Omega$, we consider the six faces of $\partial\Omega$.

□

3.2 Exploration of the Kernel of the Regularizer for a Parametrization with Second-Order Polynomials

We use a parametrization with polynomials of second-order for the transformation. From this, we observe, that some second-order polynomials are in the kernel of the second-order DC regularizer from Def. 2.2.1. Thus, the regularizer allows for non-bijective transformations, since second-order polynomials are in general not bijective.

Exploration of the Kernel of the Two-Dimensional Regularizer

We start to explore the kernel of the two-dimensional second-order DC regularizer. An extension to $d = 3$ is provided in the following paragraph.
We consider the polynomials of second-order

$$\Pi_2^2(\mathbb{R}^2) = \left\{ y : \mathbb{R}^2 \to \mathbb{R}^2 \middle| y(x) = \sum_{i=1}^{2} \sum_{j \in J} a_{i,j} x_1^{j_1} \cdot x_2^{j_2} e_i,\ a_{i,j} \in \mathbb{R} \right\},$$

where $J = \{j \in \mathbb{N}_0^2 : \sum_{l=1}^{2} j_l \leq 2\}$ and $e_i \in \mathbb{R}^2$ is an unitary vector. It holds

$$\dim(\Pi_2^2(\mathbb{R}^2)) = 2 \cdot \#J = 12.$$

Clearly, due to the second-order of the differential operator $\mathcal{B}$ all first-order polynomials $\Pi_1^2(\mathbb{R}^2)$ are in the kernel of the regularizer. This also ensures, that affine linear transformations are in the kernel, which is a desired property for a global alignment of the images.

Theorem 3.2.1. For a parametrization of the transformation with second-order poly-

Table 4: *Vector-fields $b_k \in \Pi_2^d(\mathbb{R}^d)\backslash\Pi_1^d(\mathbb{R}^d)$ that are in the kernel of the second-order DC regularizer for $d = 2$ (top) and $d = 3$ (bottom).*

k	1	2
$b^k(x)$	$\begin{pmatrix} x_1^2-x_2^2 \\ -2x_1x_2 \end{pmatrix}$	$\begin{pmatrix} 2x_1x_2 \\ x_1^2-x_2^2 \end{pmatrix}$

k	1	2	3	4	5	6	7
$b^k(x)$	$\begin{pmatrix} x_1^2-x_2^2 \\ -2x_1x_2 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 2x_1x_2 \\ x_1^2-x_2^2 \\ 0 \end{pmatrix}$	$\begin{pmatrix} x_1^2-x_3^2 \\ 0 \\ -2x_1x_3 \end{pmatrix}$	$\begin{pmatrix} 2x_1x_3 \\ 0 \\ x_1^2-x_3^2 \end{pmatrix}$	$\begin{pmatrix} 2x_1x_3 \\ -2x_2x_3 \\ x_1^2-x_2^2 \end{pmatrix}$	$\begin{pmatrix} 2x_1x_2 \\ x_1^2-x_3^2 \\ -2x_2x_3 \end{pmatrix}$	$\begin{pmatrix} x_2x_3 \\ x_1x_3 \\ x_1x_2 \end{pmatrix}$

Table 5: *Conditions for transformations the kernel of the differential operator $\mathcal{B}$ from Tab. 2.*

$$
\begin{array}{lll}
d=2: & \partial_{ii}y_i + \partial_{ij}y_j & = 0 \quad \text{and} \\
 & \partial_{ii}y_j - \partial_{ij}y_i & = 0 \quad \text{with } i,j = 1,2,\ j \neq i. \\
d=3: & \partial_{1j}y_1 + \partial_{2j}y_2 + \partial_{3j}y_3 & = 0, \\
 & \partial_{2j}y_3 - \partial_{2j}y_2 & = 0, \\
 & \partial_{3j}y_1 - \partial_{1j}y_3 & = 0 \quad \text{and} \\
 & \partial_{1j}y_2 - \partial_{2j}y_1 & = 0 \quad \text{with } j = 1,2,3.
\end{array}
$$

nomials $\Pi_2^2(\mathbb{R}^2)$, the kernel of the second-order DC regularizer from Def. 2.2.1 is

$$\ker(\mathcal{S}) = \left\{ \sum_{i=1}^{6} \alpha_i p^i + \sum_{k=1}^{2} \beta_k b^k : \alpha_i, \beta_k \in \mathbb{R} \right\},$$

where $(p^i, i = 1, \ldots, 6)$ is a polynomial basis of the first-order polynomials $\Pi_1^2(\mathbb{R}^2)$ and $b^k \in \Pi_2^2(\mathbb{R}^2)\backslash\Pi_1^2(\mathbb{R}^2)$ are the vector-fields from the top of Tab. 4.

Proof. If $y \in \Pi_2^2(\mathbb{R}^2)$ is supposed to be in the kernel of the second-order DC regularizer, the conditions from Tab. 5 need to be fulfilled. The number of conditions for the kernel is

$$\mathrm{C}(\mathcal{S}) = 4.$$

Obviously, all polynomials of degree one are in the kernel of the differential operator, i.e. $\Pi_1^2(\mathbb{R}^2) \subset \ker(\mathcal{B})$. With $(p^i, i = 1, \ldots, 6)$ we denote a basis functions of $\Pi_1^2(\mathbb{R}^2)$. Moreover, from the conditions in Tab. 5, we deduce a homogeneous system of linear equations for the coefficients $a_{i,l}$ of the second-order polynomials with $l \in L = \{l \in \mathbb{N}_0^2 | \sum_{i=1}^2 l_i = 2\}$. We obtain the linear system

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} a_{1,(2,0)} \\ a_{1,(1,1)} \\ a_{1,(0,2)} \\ a_{2,(2,0)} \\ a_{2,(1,1)} \\ a_{2,(0,2)} \end{pmatrix} = 0.$$

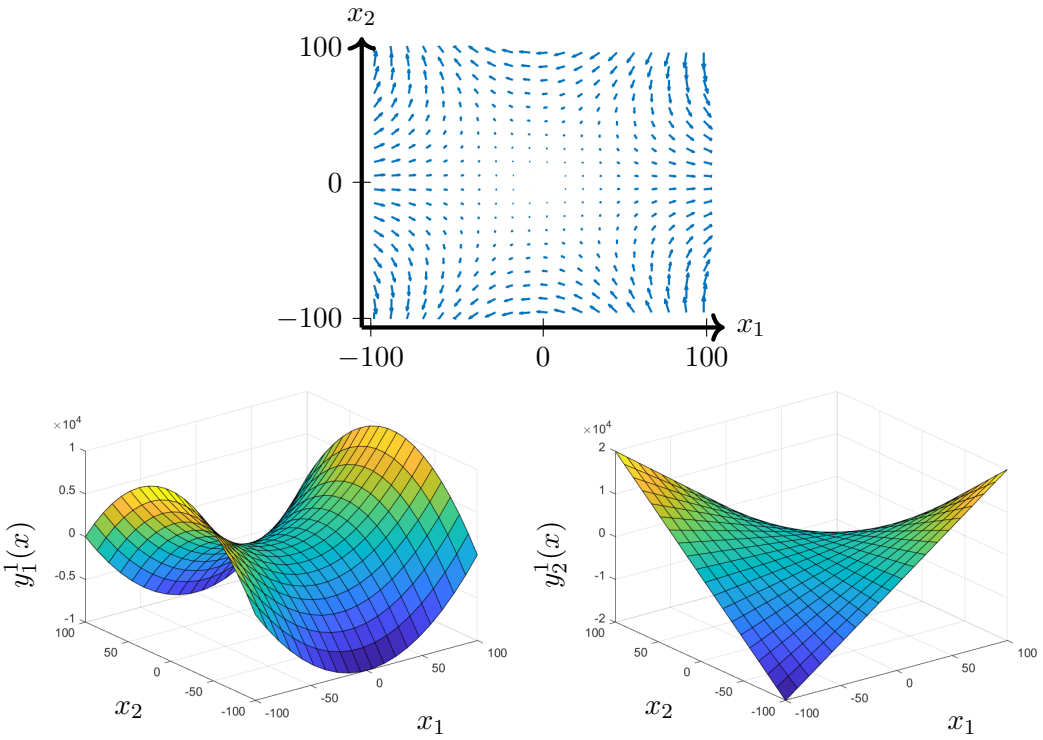

Figure 3.2.1: *Visualization of the two-dimensional vector field* $\begin{pmatrix} x_1^2 - x_2^2 \\ -2x_1x_2 \end{pmatrix} \in \Pi_2^2(\mathbb{R}^2)$*, that is in the kernel of the two-dimensional second-order DC regularizer. Plot of the vector field (top) and of the first (bottom, left) and the second (bottom, right) components of the vector field separately.*

Solving the linear equations yields the vector-fields $b^k \in \Pi_2^2(\mathbb{R}^2)$ from Tab. 4, that are in the kernel of the differential operator $\mathcal{B}$. Exemplarily, visualizations of the two-dimensional vector-field $b^1 \in \Pi_2^2(\mathbb{R}^2)$ are given in Fig. 3.2.1. It can be shown that

$$(p^1, \dots, p^6, b^1, b^2)$$

is linear independent. The completeness for a parametrization with second-order polynomials follows from

$$\dim(\ker(\mathcal{S})) = \dim(\Pi_2^2(\mathbb{R}^2)) - \mathrm{C}(\mathcal{S}) = 8.$$

□

The second-order polynomials b^k from Tab. 4 fulfill $\operatorname{div} b^k = 0$ and $\operatorname{curl} b^k = 0$ and thus also the natural boundary conditions from Thm. 3.1.2. Moreover, they demonstrate, that the regularizer does not guarantee the bijectivity of the transformations.

For the special case $\gamma = 0.5$, we obtain the uncoupled regularizer. As shown in [23], the uncoupled regularizer is similar to the curvature regularizer for functions of the Beppo-Levi space of order two. The kernel of the curvature regularizer is known [89] to include all harmonic functions and thus is infinite dimensional. This indicates, that the kernel of the second-order DC regularizer might be infinite dimensional. A rigorous analysis is left to future work.

Exploration of the Kernel of the Three-Dimensional Regularizer

We extend the results of the last paragraph to the three-dimensional setting.

In the three-dimensional setting the set of indices $J = \{j \in \mathbb{N}_0^3 : \sum_{l=1}^d j_l \leq 2\}$ has ten elements and thus it holds

$$\dim(\Pi_2^3(\mathbb{R}^3)) = 3 \cdot \#J = 30.$$

The extension of Thm. 3.2.1 is as follows.

Theorem 3.2.2. For a parametrization of the transformation with second-order polynomials $\Pi_2^3(\mathbb{R}^3)$, the kernel of the second-order DC regularizer from Def. 2.2.1 is

$$\ker(\mathcal{S}) = \left\{ \sum_{i=1}^{12} \alpha_i p^i + \sum_{k=1}^{7} \beta_k b^k : \alpha_i, \beta_k \in \mathbb{R} \right\},$$

where $(p^i, i = 1, \ldots, 12)$ is a polynomial basis of the first-order polynomials $\Pi_1^3(\mathbb{R}^3)$ and $b^k \in \Pi_2^3(\mathbb{R}^3) \backslash \Pi_1^3(\mathbb{R}^3)$ are the vector-fields from the bottom of Tab. 4.

Proof. The proof is obtained through the same arguments as in Thm. 3.2.1. Note, that we formulated twelve conditions for $d = 3$ in Tab. 5, but $\partial_{13} y_2 = \partial_{23} y_1$ is redundant to the conditions

$$\partial_{12} y_3 = \partial_{13} y_2 \quad \text{and} \quad \partial_{23} y_1 = \partial_{12} y_3.$$

Hence, the number of conditions for the kernel is

$$\mathrm{C}(\mathcal{S}) = 11.$$

□

3.3 Conclusion

In this chapter, we examined the plain second-order DC regularizer with its natural boundary conditions. In a first step we deduced the natural boundary conditions from the Euler-Lagrange equations. Due to the coupling of the components in the regularizer, there also is a coupling in the components of the boundary conditions. The second-order of the regularizer request for very smooth transformations. This makes the interpretation rather complex. Due to the complexity of the natural boundary conditions, it would furthermore be challenging to discretize them explicitly. For their discretization we would rather suggest an implicit discretization e.g. with the ghost-cell method [103]. Within the ghost-cell method, extra grid points are added outside the domain to enforce a closed linear system. Thus, the optimization incorporates the optimization over the boundary conditions. In this thesis, we consider boundary conditions, that can explicitly be discretized, for an easier implementation of the grid changes within the multigrid framework in Ch. 7.

Another disadvantage of the plain second-order DC regularizer with natural boundary conditions is, that bijective transformation cannot be guaranteed by the regularization.

Specifically, we observed that certain second-order polynomials are in the kernel of the second-order DC regularizer.

Yet, another strategy to get the kernel under control, is to impose further boundary conditions. From the curvature regularizer [89], for example, it is known that further boundary conditions reduce the kernel and lead to a finite-dimensional kernel. Hence, we present in the following chapter 4 a second-order DC regularizer with additional boundary conditions.

4 Second-Order Divergence-Curl Regularization For Displacements with Sliding Boundary Conditions

In this chapter, we further study the second-order divergence-curl (DC) regularizers from Def. 2.2.1, but restrict the analysis to displacements with sliding boundary conditions. In particular, we show, that the regularizer has a trivial kernel for displacements with sliding boundary conditions in Thm. 4.2.1. We further derive a spectral decomposition of the Gâteaux derivative for the second-order DC regularizer restricted to displacements with sliding boundary conditions.

Sliding boundary or also called slipping boundary can e.g. capture the movement of the lung within the rib cage; see [104, 105]. These boundary conditions fix the boundary in the sense that the boundary is mapped onto the boundary, but the boundary points can slide along the boundary [1, p. 95]. The boundary conditions are formulated for the displacement and not the transformation. Thus, sliding boundary conditions are additional assumptions towards the considered space of displacements. They were proposed together with other regularizers as e.g. the elastic regularizer [106, 107]. To the best of our knowledge combining the second-order DC regularizer with displacements restricted to sliding boundary conditions is a novel approach.

A spectral decomposition can be found for numerous differential operators, such as the elastic operator in [1, 106, 107], or the operator $\nabla \operatorname{div}$ in [108], but it was missing for the operator corresponding to the second-order DC regularizer, so far. The spectral decomposition from Thm. 4.3.1 enables a structuring of the frequencies, i.e. the eigenfunctions, of the operator [107]. This interpretation is particularly helpful for understanding multigrid methods (see Ch. 6 and Ch. 7), as these divide the frequency space into two parts: the low and the high frequencies.

The chapter is organized as follows: In Sec. 4.1, we present sliding boundary conditions for the displacement. In Sec. 4.2, we show a trivial kernel for the second-order DC regularizer restricted to the space of displacements with sliding boundary conditions. Furthermore, in Sec. 4.3 we provide a spectral decomposition of the differential operator $\mathcal{A}$ from Tab. 3 for displacements with sliding boundary conditions.

4.1 A Space of Displacements with Sliding Boundary Conditions

Since sliding boundary conditions are formulated in the displacement u and not in the transformation y, see Def. 2.1.2, we reformulate the whole image registration problem (1) with respect to the displacement. In particular, we consider the functional

$$\mathcal{J} : H^4(\Omega, \mathbb{R}^d) \to \mathbb{R}, \quad \mathcal{J}(u) = \mathcal{D}^{\text{SSD}}(\mathcal{T}, \mathcal{R}; \text{id} + u) + \alpha \mathcal{S}(u) \tag{10}$$

in this chapter. Compared to (1), we apply the regularizer to the displacement and not to the overall transformation. It is easily possible to incorporate a preregistration, by substitution id with the obtained transformation y^{pre}.
In this section, we present a generating set for the considered space of smooth displacements, that fulfill sliding boundary conditions. We rely on the notation used in [1, §9.2.1].

For the ease of presentation, we start with the definitions for the two-dimensional setting and continue with the three-dimensional setting afterwards.

Two-Dimensional Displacements with Sliding Boundary Conditions

We consider a two-dimensional displacement $u : \Omega \to \mathbb{R}^2$ according to Def. 2.1.2 and use the same definition of sliding boundary conditions as suggested in [1, p. 96].

Definition 4.1.1 (Sliding Boundary Conditions, Modersitzki [1, p. 96])**.** Let $\Omega = (0,1)^2$ be a domain. For a displacement $u : \Omega \to \mathbb{R}^2$ according to Def. 2.1.2 we define sliding boundary conditions with $z = 0, 1$ and $x_1, x_2 \in [0,1]$ by

$$\begin{aligned} u_1(z, x_2) = u_2(x_1, z) &= 0, \\ \partial_2 u_1(x_1, z) = \partial_1 u_2(z, x_2) &= 0. \end{aligned}$$

Remark 4.1.2 (Relation to Dirichlet-0 and Neumann-0 Boundary Conditions). Sliding boundary conditions are a combination of

- Dirichlet-0 boundary conditions [109, p. 294], i.e. $u = 0$ on $\partial\Omega$, and
- Neumann-0 boundary conditions [109, p. 345], i.e. $\partial_n u = 0$, on $\partial\Omega$ for a normal vector n on the boundary $\partial\Omega$,

for the different spatial dimension of a component $u_j, j = 1, 2$ of a displacement u.

We need the following definition for the construction of a generating set of smooth functions, that fulfill sliding boundary conditions. It is based on a product of sine and cosine functions to ensure the sliding boundary. Furthermore, a scaling matrix enters into play. The scaling factor, i.e. the amplitude, is chosen in dependence of the frequency of the considered function to simplify the proof of Thm. 4.3.1.

Definition 4.1.3 (Construction of Scaling Vectors, modified Modersitzki [107, Def. 2.3])**.** For $x \in \mathbb{R}^2$, we define $\mathrm{nz}(x)$ as the number of nonzero components of x:

$$\mathrm{nz}(x) := \#\{j : x_j \neq 0\} = \sum_{j=1}^{2} \delta_{x_j \neq 0}.$$

For $x \neq 0$, we define scaling vectors $\tau^j(x)$,

$$\begin{aligned} \tau^1(x) &:= \frac{x}{\|x\|} && \text{for } \mathrm{nz} \geq 1, \\ \tau^2(x) &:= (x_2, -x_1)^\top / \|(x_2, -x_1)\| && \text{for } \mathrm{nz} = 2, \end{aligned}$$

Example 4.1.4 (Construction of Scaling Vectors)**.** We construct scaling vectors according to Def. 4.1.3 for a given vector $x \in \mathbb{R}^2$

1. Let $x = (1,0)^\top$. As $\mathrm{nz}(x) = 1$, we only obtain the vector $\tau^1(x) = x$.
2. Let $x = (1,1)^\top$ and thus $\mathrm{nz}(x) = 2$. We obtain

$$\tau^1(x) = \frac{1}{\sqrt{2}}(1,1)^\top, \quad \tau^2(x) = \frac{1}{\sqrt{2}}(1,-1)^\top.$$

Lemma 4.1.5 (Day and Romero [106]). Let $r = (r_1, r_2)^\top \neq 0 \in \mathbb{Z}^2$, $C_j, S_j : [0,1] \to [0,1]$, $C_j(\xi) := \cos(r_j \pi \xi)$ and $S_j(\xi) := \sin(r_j \pi \xi)$ and further $w_r^j : \Omega \to \mathbb{R}, j = 1, 2$ with $w_r = (w_r^1, w_r^2,) : \Omega \to [0,1]^2$ and

$$w_r^1(x) := C_2(x_2) S_1(x_1) = \cos(r_2 \pi x_2) \sin(r_1 \pi x_1),$$
$$w_r^2(x) := S_2(x_2) C_1(x_1) = \sin(r_2 \pi x_2) \cos(r_1 \pi x_1).$$

Let $t^j(r) = (t_1^j, t_2^j)^\top \in \mathbb{R}^2$ be a scaling vector according to Def. 4.1.3 and $D_{t^j} := \operatorname{diag}(t^j) \in \mathbb{R}^{2,2}$ be a scaling matrix. Then any function $u \in C^\infty(\Omega, \mathbb{R}^2)$ that satisfies sliding boundary conditions can be expanded in terms of the functions

$$v^{r;j} := D_{t^j} w_r.$$

The proof is given for the three-dimensional setting in [106, §5]. The two-dimensional case follows directly by restricting the domain to $\Omega = [0,1] \times [0,1] \times \{0\}$.

Three-Dimensional Displacements with Sliding Boundary Conditions

Next, we present the definitions for a three-dimensional displacement u.

Definition 4.1.6 (Sliding Boundary Conditions, Modersitzki [1, p. 96]). Let $\Omega = (0,1)^3$ be a domain. For a displacement $u : \Omega \to \mathbb{R}^3$ according to Def. 2.1.2 we define sliding boundary conditions with $z = 0, 1$ and $x_1, x_2, x_3 \in [0,1]$ by

$$u_1(z, x_2, x_3) = u_2(x_1, z, x_3) = u_3(x_1, x_2, z) = 0,$$
$$\partial_2 u_1(x_1, z, x_3) = \partial_3 u_2(x_1, x_2, z) = \partial_1 u_3(z, x_2, x_3) = 0,$$
$$\partial_3 u_1(x_1, x_2, z) = \partial_1 u_2(z, x_2, x_3) = \partial_2 u_3(x_1, z, x_3) = 0.$$

For the construction of the scaling vector we have three cases.

Definition 4.1.7 (Construction of Scaling Vectors, modified Modersitzki [107, Def. 2.3]). For $x \in \mathbb{R}^3$, we define $\operatorname{nz}(x)$ as the number of nonzero components of x:

$$\operatorname{nz}(x) := \#\{j : x_j \neq 0\} = \sum_{j=1}^{3} \delta_{x_j \neq 0}.$$

For $x \neq 0$, we define scaling vectors $\tau^j(x)$,

$$\begin{aligned}
\tau^1(x) &:= \frac{x}{\|x\|} && \text{for } \operatorname{nz} \geq 1,\\
\tau^2(x) &:= \tau^1(x) \times \delta_{x=0} && \text{for } \operatorname{nz} = 2,\\
\tau^2(x) &:= (x_2, -x_1, 0)^\top / \|(x_2, -x_1, 0)\| && \text{for } \operatorname{nz} = 3,\\
\tau^3(x) &:= \tau^1(x) \times \tau^2(x) && \text{for } \operatorname{nz} = 3,
\end{aligned}$$

Example 4.1.8 (Construction of Scaling Vectors). We construct scaling vectors according to Def. 4.1.7 for a given vector $x \in \mathbb{R}^3$

1. Let $x = (1,0,0)^\top$. As $\mathrm{nz}(x) = 1$, we only obtain the vector $\tau^1(x) = x$.
2. Let $x = (1,1,0)^\top$ and thus $\mathrm{nz}(x) = 2$. We obtain

$$\tau^1(x) = \frac{1}{\sqrt{2}}(1,1,0)^\top, \quad \tau^2(x) = \frac{1}{\sqrt{2}}(1,-1,0)^\top.$$

3. Let $x = (1,1,1)$, hence $\mathrm{nz}(x) = 3$. Thus, we have

$$\tau^1(x) = \frac{1}{\sqrt{3}}(1,1,1)^\top, \quad \tau^2(x) = \frac{1}{\sqrt{2}}(1,-1,0)^\top, \quad \tau^3(x) = \frac{1}{\sqrt{6}}(1,1,-2)^\top.$$

Due to the Neumann-0 boundary condition in the additional spatial dimension of the component of the displacement $u_j, j = 1,2,3$, we need a multiplication with another cosine when building on the two-dimensional setting.

Lemma 4.1.9 (Day and Romero [106]). Let $r = (r_1, r_2, r_3)^\top \neq 0 \in \mathbb{Z}^3$, $C_j, S_j : [0,1] \to [0,1]$, $C_j(\xi) := \cos(r_j \pi \xi)$ and $S_j(\xi) := \sin(r_j \pi \xi)$ and further $w_r^j : \Omega \to \mathbb{R}, j = 1,2,3$ with $w_r = (w_r^1, w_r^2, w_r^3) : \Omega \to [0,1]^3$ and

$$\begin{aligned}
w_r^1(x) &:= C_3(x_3)C_2(x_2)S_1(x_1) = \cos(r_3\pi x_3)\cos(r_2\pi x_2)\sin(r_1\pi x_1),\\
w_r^2(x) &:= C_3(x_3)S_2(x_2)C_1(x_1) = \cos(r_3\pi x_3)\sin(r_2\pi x_2)\cos(r_1\pi x_1),\\
w_r^3(x) &:= S_3(x_3)C_2(x_2)C_1(x_1) = \sin(r_3\pi x_3)\cos(r_2\pi x_2)\cos(r_1\pi x_1).
\end{aligned}$$

Let $t^j(r) = (t_1^j, t_2^j, t_3^j)^\top \in \mathbb{R}^3$ be a scaling vector according to Def. 4.1.7 and $D_{t^j} := \mathrm{diag}\,(t^j) \in \mathbb{R}^{3,3}$ be a scaling matrix. Then any function $u \in C^\infty(\Omega, \mathbb{R}^3)$ that satisfies sliding boundary conditions can be expanded in terms of the functions

$$v^{r;j} := D_{t^j} w_r.$$

The proof of Lemma 4.1.9 is given in [106, §5]. The main idea of the proof is, that the divergence operator is applied to a function satisfying sliding boundary conditions. The divergence of that function can then be expressed by the complete set of eigenfunctions of the Laplace operator with Neumann-0 boundary conditions.

4.2 Kernel of the Regularizer For Displacements with Sliding Boundary Conditions

We show, that the second-order DC regularizer has a trivial kernel, when restricted to the space of smooth displacements with sliding boundary conditions. In particular, we analyze the kernel of the differential operator $\mathcal{B}$ from Tab. 2.

Theorem 4.2.1. Let $\gamma \in (0,1)$. The kernel of the differential operator $\mathcal{B}$ from Tab. 2 restricted to $C^\infty(\mathbb{R}^d, \mathbb{R}^d)$ functions, that fulfill sliding boundary conditions is trivial.

The proof is provided for the two- and three-dimensional setting separately in the paragraphs at the end of this section.

Remark 4.2.2 (Special Cases). The kernel of the differential operator restricted to the $\mathbb{C}^\infty(\mathbb{R}^d, \mathbb{R}^d)$ functions with sliding boundary conditions is not trivial for the special cases $\gamma = 0$ or $\gamma = 1$. For example for $d = 3$ and $\gamma = 1$ all functions $v^{r,j}$ with scaling vectrors $t^j \perp r$ are in the kernel.

Due to the restriction to sliding boundary conditions the polynomials of second order from Thm. 4.2.1 are eliminated form the kernel, but so are the affine linear transformations. Therefore, it is expected that a pre-registration y^{pre} to capture the affine linear parts will improve the results of image registration.

Proof of Theorem 4.2.1 for the Two-Dimensional Regularizer

We provide the proof of Thm. 4.2.1 for the two-dimensional setting.

Proof. From Lem. 4.1.5, we know, that any smooth displacement, that fulfills the sliding boundary conditions, can be expanded in terms of the functions $v^{r,j}$.
The application of the differential operator $\mathcal{B}$ from Tab. 2 to a function $v^{r,j}$, leads to

$$\mathcal{B}_{\text{div}}\, v^{r;j} = \begin{pmatrix} -(r_1^2 t_1^j + r_1 r_2 t_2^j)\pi^2 w_r^1 \\ -(r_2 r_1 t_1^j + r_2^2 t_2^j)\pi^2 w_r^2 \end{pmatrix} = -(\pi)^2 D_{(r^\top t^j)r} w_r,$$

and

$$\mathcal{B}_{\text{curl}}\, v^{r;j} = \begin{pmatrix} -(r_1 r_2 t_1^j - r_1^2 t_2^j)\pi^2 w_r^2 \\ -(r_2^2 t_1^j - r_1 r_2 t_2^j)\pi^2 w_r^1 \end{pmatrix} = -(\pi)^2 D_{(r^\top r)t^j - (r^\top t^j)r} \begin{pmatrix} -w_r^2 \\ w_r^1 \end{pmatrix}.$$

Thus, for $v^{r,j}$ to be in the kernel of $\mathcal{B}_{\text{div}}$ and $\mathcal{B}_{\text{curl}}$ we need

$$(r^\top t^j)r = 0 \quad \text{and} \tag{11}$$
$$(r^\top r)t^j - (r^\top t^j)r = 0. \tag{12}$$

For $r \neq 0$, we need $t^j \perp r$ from Eq. (11) and thus it follows $t^j = 0$ from Eq. (12). Hence, we end up with the trivial kernel. □

Proof of Theorem 4.2.1 for the Three-Dimensional Regularizer

We provide the proof of Thm. 4.2.1 for the three-dimensional setting.

Proof. The application of the differential operator $\mathcal{B}$ from Tab. 2 to a function $v^{r,j}$ from Lem. 4.1.9, leads to

$$\mathcal{B}_{\text{div}}\, v^{r;j} = \begin{pmatrix} -(r_1^2 t_1^j + r_1 r_2 t_2^j + r_1 r_3 t_3^j)\pi^2 w_r^1 \\ -(r_2 r_1 t_1^j + r_2^2 t_2^j + r_2 r_3 t_3^j)\pi^2 w_r^2 \\ -(r_3 r_1 t_1^j + r_3 r_2 t_2^j + r_3^2 t_3^j)\pi^2 w_r^3 \end{pmatrix} = -(\pi)^2 D_{(r^\top t^j)r} w_r,$$

and

$$\mathcal{B}_{\text{curl}}\, v^{r;j}(x) = \begin{pmatrix} (-r_1r_3t_2^j + r_1r_2t_3^j)\pi^2 S_1(x_1)S_2(x_2)S_3(x_3) \\ (-r_1r_3t_1^j + r_1^2t_3^j)\pi^2 w_r^3(x) \\ (r_1r_2t_1^j - r_1^2t_2^j)\pi^2 w_r^2(x) \\ (r_2r_3t_2^j - r_2^2t_3^j)\pi^2 w_r^3(x) \\ (r_2r_3t_1^j - r_1r_2t_3^j)\pi^2 S_1(x_1)S_2(x_2)S_3(x_3) \\ (r_2^2t_1^j - r_1r_2t_2^j)\pi^2 w_r^1(x) \\ (r_3^2t_2^j - r_2r_3t_3^j)\pi^2 w_r^2(x) \\ (-r_3^2t^jr_1 + r_1r_3t_3^j)\pi^2 w_r^1(x) \\ (-r_2r_3t_1^j + r_1r_3t_2^j)\pi^2 S_1(x_1)S_2(x_2)S_3(x_3) \end{pmatrix}.$$

Analogously to the two-dimensional proof, we need $r \perp t^j$ for $r \neq 0$ for $v^{r,j}$ to be in the kernel of $\mathcal{B}_{\text{div}}$. Furthermore, for $v^{r,j}$ to be in the kernel of $\mathcal{B}_{\text{curl}}$, we need $r_k(r \times t^j) = 0$ for all $k = 1, 2, 3$ i.e. $r \times t^j = 0$ for $r \neq 0$. Hence, we end up with $t^j = 0$ for $v^{r,j}$ to be in the kernel of $\mathcal{B}$ and thus have a trivial kernel. □

4.3 Spectral Decomposition of the Corresponding Fourth-Order Differential Operator For Displacements with Sliding Boundary Conditions

This section provides a spectral decomposition of the differential operator $\mathcal{A}$ from Tab. 3 for smooth displacements with sliding boundary conditions.

Theorem 4.3.1. Let the functions

$$v^{r;j} := D_{t^j} w_r$$

be defined as in Lem. 4.1.5 for $d = 2$ and according to Lem. 4.1.9 for $d = 3$. The functions $v^{r;j}$ are a complete set of eigenfunction of the partial differential operator $\mathcal{A}$ (see Tab. 3) with respect to the sliding boundary conditions. The corresponding eigenvalues are

$$\lambda_{r;j} = \|r\|^4\pi^4 \begin{cases} \gamma & \text{for } j = 1, \\ (1-\gamma) & \text{for } j = 2, 3. \end{cases}$$

The proof is provided for the two- and three-dimensional setting seperately in the paragraphs at the end of this section together with simple examples.

The spectral decomposition enables a natural structuring of the space of displacements with sliding boundary conditions via the eigenfunction [107]. The eigenfunctions are also called frequencies; see [31, p.16]. This concept is used for the multigrid approach in Ch. 7. Moreover, it verifies the results for the kernel, as stated in the following remark.

Remark 4.3.2. Let $\gamma \in (0, 1)$. Then the spectral decomposition from Thm. 4.3.1 verifies the trivial kernel $\ker(\mathcal{B}) = \{0\}$ for all $\mathbb{C}^\infty(\mathbb{R}^d, \mathbb{R}^d)$ functions, that fulfill sliding boundary conditions as stated in Thm. 4.2.1.

Spectral Decomposition for the Two-Dimensional Setting

We consider the two-dimensional differential operator $\mathcal{A}$ in the following and provide the proof for Thm. 4.3.1.

Proof. With $C_j, S_j : [0,1] \to [0,1]$ defined as in Lem. 4.1.5, it holds

$$\partial\partial C_j(\xi) = -\pi^2 r_j^2 C_j(\xi) \quad \text{and} \quad \partial\partial S_j(\xi) = -\pi^2 r_j^2 S_j(\xi)$$

it follows

$$\Delta w_r^j(x) = -\pi^2(r_1^2 + r_2^2) w_r^j(x) = -\pi^2 \|r\|_2^2 w_r^j(x).$$

Further, it holds

$$\begin{aligned}\partial_{jk}\Delta w_r^j &= \pi^4 r_j r_k \|r\|_2^2 w_r^k \\ \partial_{kk}\Delta w_r^j &= \pi^4 r_k r_k \|r\|_2^2 w_r^j\end{aligned}$$

for $j,k = 1,2$. Let $t \in \mathbb{R}^2$ and $D_t = \operatorname{diag}(t) \in \mathbb{R}^{2,2}$ be a scaling matrix. Thus, it holds

$$\begin{aligned}\mathcal{A}_{\text{div}}(D_t w_r) &= \begin{pmatrix} t_1\partial_{11}\Delta w_r^1 + t_2\partial_{12}\Delta w_r^2 \\ t_1\partial_{12}\Delta w_r^1 + t_2\partial_{22}\Delta w_r^2 \end{pmatrix} \\ &= \begin{pmatrix} \pi^4\|r\|_2^2 (r^\top t) r_1 w_r^1 \\ \pi^4\|r\|_2^2 (r^\top t) r_2 w_r^2 \end{pmatrix} \\ &= \pi^4 \|r\|_2^2 D_{(r^\top t)r} w_r,\end{aligned}$$

and

$$\begin{aligned}\mathcal{A}_{\text{curl}}(D_t w_r) &= \begin{pmatrix} t_1\partial_{22}\Delta w_r^1 - t_2\partial_{12}\Delta w_r^2 \\ -t_1\partial_{12}\Delta w_r^1 + t_2\partial_{11}\Delta w_r^2 \end{pmatrix} \\ &= \begin{pmatrix} \pi^4\|r\|_2^2 r_2 (t_1 r_2 - t_2 r_1) w_r^1 \\ \pi^4\|r\|_2^2 r_1 (-t_1 r_2 + t_2 r_1) w_r^2 \end{pmatrix} \\ &= \|r\|_2^4 \pi^4 (D_{(r^\top r)t - (r^\top t)r} w_r).\end{aligned}$$

Henceforth, we have

$$\mathcal{A}(D_t w_r) = \gamma\pi^4\|r\|_2^2 D_{(r^\top t)r} w_r + (1-\gamma)\pi^4\|r\|_2^2 D_{(r^\top r)t - (r^\top t)r} w_r. \tag{13}$$

For $D_t w_r$ to be an eigenfunction, we have $\mathcal{A}(D_t w_r) = \lambda D_t w_r$ for some $\lambda \in \mathbb{R}$, i.e. we need t to fulfill

$$t = \alpha_1 (r^\top t) r + \alpha_2 ((r^\top r) t - (r^\top t) r) \tag{14}$$

for some constants α_1, α_2. Thus, for $r \neq 0$, we either need $t \perp r$ or $t \in \operatorname{span}(r)$. Let $t^j \in \{\tau^j(r), j = 1, \dots, \operatorname{nz}(r)\}$ be chosen according to Def. 4.1.3. Then $t^1 := \frac{r}{\|r\|}$, and $t^2 \perp r$ by construction. Hence, Eq. (14) is fulfilled for t^1 and t^2, since we have

$$\begin{aligned}(r^\top t^1) r &= \frac{\|r\|^2}{\|r\|} r = \|r\|^2 t^1 \\ (r^\top t^2) r &= 0\end{aligned}$$

and

$$(r^\top r)t^1 - (r^\top t^1)r = (\|r\|^2 - \|r\|^2)t^1 = 0$$
$$(r^\top r)t^2 - (r^\top t^2)r = \|r\|^2 t^2.$$

Henceworth, Eq. (13) yields

$$\mathcal{A}\left(D_{t^j} w_r\right) = \pi^4 \|r\|_2^4 \rho_j D_{t^j} w_r \quad \text{with} \quad \rho_j = \begin{cases} \gamma & \text{for } j = 1, \\ (1-\gamma) & \text{for } j = 2. \end{cases}$$

From Lem. 4.1.5 the completeness follows immediately. □

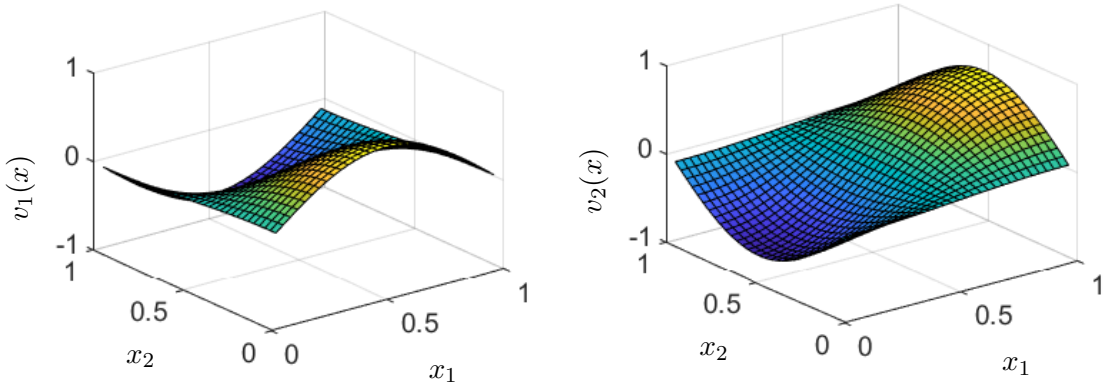

Figure 4.3.1: *First (left) and second (right) components of the eigenfunction* $v(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} \sin(\pi x_1)\cos(\pi x_2) \\ -\cos(\pi x_1)\sin(\pi x_2) \end{pmatrix}$ *for the differential operator* $\mathcal{A}$ *from Tab. 3. Details are provided in Ex. 4.3.3.*

Example 4.3.3 (Spectral Decomposition for $d = 2$)**.** For the partial differential operator $\mathcal{A}$ from Tab. 3 for $d = 2$ with respect to the sliding boundary conditions we obtain the eigenfunctions

$$v^{(1,1)^\top;\,1}(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} \sin(\pi x_1)\cos(\pi x_2) \\ \cos(\pi x_1)\sin(\pi x_2) \end{pmatrix}, \quad v^{(1,1)^\top;\,2}(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} \sin(\pi x_1)\cos(\pi x_2) \\ -\cos(\pi x_1)\sin(\pi x_2). \end{pmatrix}$$

The eigenfunction $v^{(1,1)^\top;\,2}$ is visualized in Fig. 4.3.1. Here we choose $r = (1,1)^\top$. Hence, with Def. 4.1.3 we obtain the scaling vectors $t^1 = \frac{1}{\sqrt{2}}(1,1)^\top$ and $t^2 = \frac{1}{\sqrt{2}}(1,-1)^\top$; see

Ex. 4.1.4. We can easily see that for $v^i, i = 1, 2$ with $v^i = (v_1^i, v_2^i)^\top$

$$\begin{aligned}
\mathcal{A}v^i &= \gamma \nabla\Delta \operatorname{div} v^i + (1-\gamma)\nabla\Delta \operatorname{curl} v^i \\
&= \gamma \begin{pmatrix} \partial_{11}\Delta & \partial_{12}\Delta \\ \partial_{12}\Delta & \partial_{22}\Delta \end{pmatrix} v^i + (1-\gamma) \begin{pmatrix} \partial_{22}\Delta & -(\partial_{12}\Delta) \\ -(\partial_{12}\Delta) & \partial_{11}\Delta \end{pmatrix} v^i \\
&= \begin{pmatrix} [\gamma\partial_{11}\Delta + (1-\gamma)\partial_{22}\Delta]v_1^i + [(2\gamma-1)\partial_{12}\Delta]v_2^i \\ [(2\gamma-1)\partial_{12}\Delta]v_1^i + [\gamma\partial_{22}\Delta + (1-\gamma)\partial_{11}\Delta]v_2^i \end{pmatrix} \\
&= \begin{pmatrix} [\gamma 2\pi^4 + (1-\gamma)2\pi^4]v_1^i + (-1)^{(i+1)}[(2\gamma-1)2\pi^4]v_1^i \\ (-1)^{(i+1)}[(2\gamma-1)2\pi^4]v_2^i + [\gamma 2(\pi)^4 + (1-\gamma)2\pi^4]v_2^i \end{pmatrix} \\
&= v^i \cdot \begin{cases} \gamma 4(\pi)^4 & \text{for } i = 1, \\ (1-\gamma)4(\pi)^4 & \text{for } i = 2. \end{cases}
\end{aligned}$$

Thus, we have the eigenvalues

$$\lambda_{(1,1)^\top;\,1} = \gamma 4\pi^4, \quad \lambda_{(1,1)^\top;\,2} = (1-\gamma)4\pi^4.$$

Spectral Decomposition for the Three-Dimensional Setting

We outline the proof of Thm. 4.3.1 for the three-dimensional setting.

Proof. The proof follows the same scheme as in the two-dimensional setting. The application of the three-dimensional differential operators to $D_t w_r$ leads to

$$\begin{aligned}
\mathcal{A}_{\text{div}}\left(D_t w_r\right) &= \begin{pmatrix} t_1\partial_{11}\Delta w_r^1 + t_2\partial_{12}\Delta w_r^2 + t_3\partial_{13}\Delta w_r^3 \\ t_1\partial_{12}\Delta w_r^1 + t_2\partial_{22}\Delta w_r^2 + t_3\partial_{23}\Delta w_r^3 \\ t_1\partial_{13}\Delta w_r^1 + t_2\partial_{23}\Delta w_r^3 + t_3\partial_{33}\Delta w_r^3 \end{pmatrix} \\
&= \begin{pmatrix} \pi^4\|r\|_2^2(r^\top t)r_1 w_r^1 \\ \pi^4\|r\|_2^2(r^\top t)r_2 w_r^2 \\ \pi^4\|r\|_2^2(r^\top t)r_3 w_r^3 \end{pmatrix} \\
&= \pi^4\|r\|_2^2 D_{(r^\top t)r} w_r,
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{A}_{\text{curl}}\left(D_t w_r\right) &= \begin{pmatrix} t_1(\partial_{22}+\partial_{33})\Delta w_r^1 - t_2\partial_{12}\Delta w_r^2 - t_3\partial_{13}\Delta w_r^3 \\ -t_1\partial_{12}\Delta w_r^1 + t_2(\partial_{11}+\partial_{33})\Delta w_r^2 - t_3\partial_{23}\Delta w_r^3 \\ -t_1\partial_{13}\Delta w_r^1 - t_2\partial_{23}\Delta w_r^3 + t_3(\partial_{11}+\partial_{22})\Delta w_r^3 \end{pmatrix} \\
&= \begin{pmatrix} \pi^4\|r\|_2^2(r_2(t_1r_2 - t_2r_1) - r_3(t_3r_1 - t_1r_3))w_r^1 \\ \pi^4\|r\|_2^2(r_3(t_2r_3 - t_3r_2) - r_1(t_1r_2 - t_2r_1))w_r^2 \\ \pi^4\|r\|_2^2(r_1(t_3r_1 - t_1r_3) - r_2(t_2r_3 - t_3r_2))w_r^3 \end{pmatrix} \\
&= \pi^4\|r\|_2^2 D_{r\times(t\times r)} w_r.
\end{aligned}$$

With the Graßmann-identity, i.e.

$$r \times (t \times r) = (r^\top r)t - (r^\top t)r,$$

we can rewrite the application of $\mathcal{A}_{\text{curl}}$ to

$$\mathcal{A}_{\text{curl}}\left(D_t w_r\right) = \pi^4\|r\|_2^2 D_{(r^\top r)t-(r^\top t)r} w_r.$$

Thus, we obtain the same condition for t as in the two-dimensional setting. From Def. 4.1.7, we know that $t^1 := \frac{r}{\|r\|}$, and $t^2, t^3 \perp r$ by construction. And thus we obtain

$$\mathcal{A}\left(D_{t^j} w_r\right) = \pi^4 \|r\|_2^4 \rho_j D_{t^j} w_r \quad \text{with} \quad \rho_j = \begin{cases} \gamma & \text{for } j = 1, \\ (1-\gamma) & \text{for } j = 2, 3. \end{cases}$$

From Lem. 4.1.9 the completeness follows immediately. □

Example 4.3.4 (Spectral Decomposition for $d = 3$)**.** For the three-dimensional partial differential operator $\mathcal{A}$ from Tab. 3 with respect to the sliding boundary conditions we determine the eigenfunctions

$$v^{(1,1,1)^\top;\, j} = D_{t^j} \begin{pmatrix} \sin(\pi x_1)\cos(\pi x_2)\cos(\pi x_3) \\ \cos(\pi x_1)\sin(\pi x_2)\cos(\pi x_3) \\ \cos(\pi x_1)\cos(\pi x_2)\sin(\pi x_1) \end{pmatrix}, \quad j = 1, 2, 3.$$

Here we built on Ex. 4.1.8 to determine the scaling vectors corresponding to $r = (1,1,1)^\top$ and have $t^1 = \frac{1}{\sqrt{3}}(1,1,1)^\top$, $t^2 = \frac{1}{\sqrt{2}}(1,-1,0)^\top$ and $t^3 = \frac{1}{\sqrt{6}}(1,1,-2)^\top$. The corresponding eigenvalues are

$$\lambda_{(1,1,1)^\top;\,1} = \gamma 9\pi^4, \quad \lambda_{(1,1,1)^\top;\,2} = (1-\gamma)9\pi^4 = \lambda_{(1,1,1)^\top;\,3}.$$

4.4 Conclusion

In this chapter we examined the kernel and the spectral decomposition of the second-order DC regularizer which was restricted to displacements with sliding boundary conditions.
In particular, we showed, that the regularizer has a trivial kernel and thus solves the problem of the non-bijective mappings from the previous chapter 3. At the same time a trivial kernel makes a preregistration necessary, which is already assumed in the approach as it is formulated in the displacement.
In addition, we determined a spectral decomposition of the Gâteaux derivative of the regularizer. This again supports our findings of a trivial kernel and furthermore allows for new numerical solutions, but also gives a first understanding of the frequencies for the multigrid approach, which follows in Ch. 7.

A more detailed analysis of how the sliding boundary conditions change the image registration problem compared to natural boundary conditions is left to future work. More specifically, the similarity of the solution of the image registration problem with natural compared to sliding boundary conditions is an open question.

A main advantage of the sliding boundary conditions is, that they can be discretized explicitly, which we will present in the following Ch. 5.

5 Discretize-Then-Optimize Approaches for Image Registration with Second-Order Divergence-Curl Regularization

In general, image registration problems, such as (1) or (10), cannot be solved analytically, which makes an appropriate discretization and numerical schemes inevitable. In this chapter, we outline well-known discretization tools and optimization schemes for variational image registration, as can e.g. be found in [1]. Furthermore, we propose discretizations based on finite differences for the second-order divergence-curl (DC) regularizer with three different boundary conditions: Dirichlet-0, Neumann-0 [109] and sliding [1] boundary conditions. Furthermore, we provide numerical examples with regard to consistency and convergence of the proposed discretizations.

In a discretize-then-optimize approach [110, 111], that we follow here, the image registration problem from Def. 2.1.4 is discretized first and solved in the finite dimensional space afterwards; see e.g. [112, 46]. An advantage of the discretize-then-optimize approach is, that standard optimization methods can be used [46]. Another option are optimize-then-discretize approaches, where the continuous Euler-Lagrange equation is discretized and then solved numerically; see e.g. [86, 69]. Here, well-known schemes to solve PDEs can be used; see e.g. [1, §8.2].

The discretization of the differential operators of the second-order DC regularizer are based on finite difference schemes [113, 114, 115]. Finite differences are a standard approach due to their simplicity.
As staggered grid discretizations are known to be able to capture the divergence and the curl operator, see e.g. [112], we use staggered grids for the discretization of the differential operators of the second-order DC regularizer.

The chapter is organized as follows: We present a discretization of the proposed image registration problem (10) in Sec. 5.1. Especially, we propose discretizations for the second-order DC regularization with different boundary conditions. Furthermore, we outline standard methods [53] for the optimization of the discrete problem in Sec. 5.2.

5.1 Discretization of the Considered Image Registration Problem

Since we follow a discretize-then-optimize approach [110, 111], we define a discrete analogue of the image registration problem (10) in a first step. The discretizations in this thesis build on standard approaches, as e.g. described in [1]. The discretization of the domain is based on regular grids with equidistant points within the domain Ω. Integrals are discretized by quadrature rules and differential operators are discretized with finite difference operators.

5.1.1 Discretization of the Domain

A grid Ω_h is an uniform division of an interval $\Omega = [0,1]^d \subset \mathbb{R}^d$ into $m \in \mathbb{N}^d$ cells, where m is the so called grid size. Grid points are a representation of these cells, whereby the

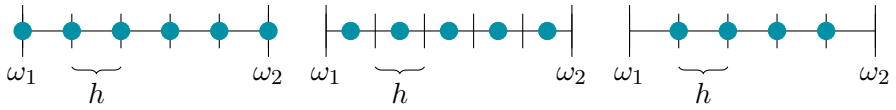

Figure 5.1.1: *A one-dimensional grid with grid size $m = 5$ with grid points according to the grid types from left to right: nodal, centered and inner-nodal.*

Table 6: *Grid point positions for a one-dimensional grid with grid size $m \in \mathbb{N}$ and grid width $h = \frac{\omega_2 - \omega_1}{m}$ on a domain $\Omega = [\omega_1, \omega_2]$.*

nodal grid:	$\mathfrak{N}_j =$	$\omega_1 + h(j-1),$	$j = 1, \dots, m+1$
centered:	$\mathfrak{C}_j =$	$\omega_1 + h(j-\frac{1}{2}),$	$j = 1, 2 \dots m,$

grid type defines the specific position of the points in relation to the cells. Hence, a grid Ω_h is defined by a domain Ω, a grid size m and the set of grid points X. For the ease of presentation we consider a grid Ω_h to be represented by the vector of grid points x_h in the following. Furthermore, we first define a grid for $d = 1$ as an extension to $d = 2, 3$ can easily be realized.

Definition 5.1.1 (Gridpoints for $d = 1$, modified Modersitzki [5, p. 20])**.** A domain $\Omega = [\omega_1, \omega_2]$ is divided into $m \in \mathbb{N}$ congruent cells, where a cell c^j is an interval of the form

$$c^j = \left\{ c \in \mathbb{R} : -\frac{h}{2} < c_i - \left(\omega_1 + (j - \frac{1}{2})h \right) < \frac{h}{2} \right\},$$

where $h = \frac{\omega_2 - \omega_1}{m}$ is the grid width. A grid point is called

- centered, if it is in the center of a cell.
- nodal, if it is on the vertex of a cell.

Hence, with the grid point position from Tab. 6, a grid with the set of grid points X

- has nodal grid type, if all grid points are nodal , i.e.

$$X = \{x_j | x_j = \mathfrak{N}_j \text{ for } j = 1, \dots, m+1\}.$$

- has centered grid type, if all grid points are centered , i.e.

$$X = \{x_j | x_j = \mathfrak{C}_j \text{ for } j = 1, \dots, m\}.$$

- has inner-nodal grid type, if all grid points within the domain Ω are nodal , i.e.

$$X = \{x_j | x_j = \mathfrak{N}_{j+1} \text{ for } j = 1, \dots, m-1\}.$$

A visualization of the three one-dimensional grid types is given in Fig. 5.1.1. For a given

one-dimensional grid Ω_h with grid size m, the number of grid points is

$$N := |X| = \begin{cases} m & \text{for a centered grid type,} \\ m+1 & \text{for a nodal grid type,} \\ m-1 & \text{for an inner-nodal grid type.} \end{cases}$$

We outline the extension to d-dimensional domains. For a d-dimensional domain $\Omega = \bigtimes_{i=1}^{d}[\omega_1^i, \omega_2^i] \subset \mathbb{R}^d$ a cell c^j is a d-dimensional interval of the form

$$c^j = \left\{ c \in \mathbb{R}^d : -\frac{h_i}{2} < c_i - \left(\omega_1^i + (j_i - \frac{1}{2})h_i\right) < \frac{h_i}{2} \right\}$$

with grid width $h \in \mathbb{R}^d$ with $h_i := \frac{\omega_2^i - \omega_1^i}{m_i}, i = 1, \ldots, d$, a multi-index $j = (j_1, \cdots j_d) \in \mathbb{N}^d$ with $j_i = 1, \cdots, m_i$ and grid size $m \in \mathbb{N}^d$; see [5, p. 20]. An element of the grid points in the d-dimensional space consists of d components, where each component is an element of $\overline{\Omega}$, i.e. for $x \in X : x = (x^1, \ldots, x^d)$ with $x^i = (x_1^i, \ldots, x_d^i) \in \overline{\Omega}, i = 1, \ldots, d$.

Table 7: *Set of grid points for a two-dimensional grid with grid size $m \in \mathbb{N}^2$ and grid width $h \in \mathbb{N}^2$ on a domain $\Omega = [\omega_1^1, \omega_2^1] \times [\omega_1^2, \omega_2^2]$ with the grid type : nodal ($X^{\mathfrak{N}}$), inner-nodal ($X^{\mathfrak{IN}}$), centered ($X^{\mathfrak{C}}$), staggered ($X^{\mathfrak{S}}$) and inner-staggered ($X^{\mathfrak{IS}}$).*

$$\begin{aligned}
X^{\mathfrak{N}} = \quad & \{(x_{j_1}^1, x_{j_2}^2)^\top \quad | x_{j_i}^i = (\mathfrak{N}_{j_i^1}, \mathfrak{N}_{j_i^2}) \text{ for } j_i^k = 1, \ldots, m_k + 1\} \\
X^{\mathfrak{IN}} = \quad & \{(x_{j_1}^1, x_{j_2}^2)^\top \quad | x_{j_i}^i = (\mathfrak{N}_{j_i^1+1}, \mathfrak{N}_{j_i^2+1}) \text{ for } j_i^k = 1, \ldots, m_k - 1\} \\
X^{\mathfrak{C}} = \quad & \{(x_{j_1}^1, x_{j_2}^2)^\top \quad | x_{j_i}^i = (\mathfrak{C}_{j_i^1}, \mathfrak{C}_{j_i^2}) \text{ for } j_i^k = 1, \ldots, m_k\} \\
X^{\mathfrak{S}} = \quad & \{(x_{j_1}^1, x_{j_2}^2)^\top \quad | x_{j_1}^1 = (\mathfrak{N}_{j_1^1}, \mathfrak{C}_{j_1^2}) \wedge x_{j_2}^2 = (\mathfrak{C}_{j_2^1}, \mathfrak{N}_{j_2^2}) \\
& \qquad\qquad \text{for } j_i^i = 1, \ldots, m_i + 1,\ j_i^k = 1, \ldots, m_k \text{ for } i \neq k\} \\
X^{\mathfrak{IS}} = \quad & \{(x_{j_1}^1, x_{j_2}^2)^\top \quad | x_{j_1}^1 = (\mathfrak{N}_{j_1^1+1}, \mathfrak{C}_{j_1^2}) \wedge x_{j_2}^2 = (\mathfrak{C}_{j_2^1}, \mathfrak{N}_{j_2^2+1}) \\
& \qquad\qquad \text{for } j_i^k = 1, \ldots, m_k\}
\end{aligned}$$

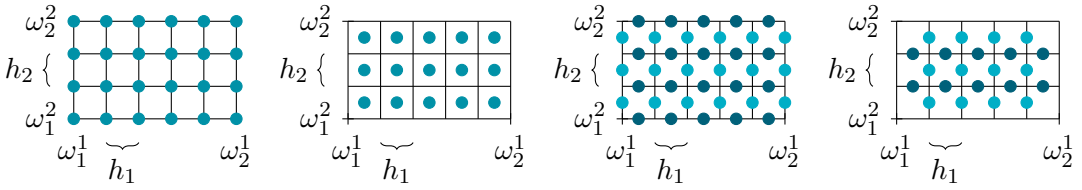

Figure 5.1.2: *A two-dimensional grid with grid size $m = (5, 3)$ with grid points according to the grid types from left to right: nodal, centered, staggered and inner-staggered.*

Definition 5.1.2 (Grid points for $d = 2$)**.** For $d = 2$ a grid point x consists of two components x^1, x^2 where each component has two coordinates, i.e. $x^i \in \mathbb{R}^2, i = 1, 2$. A grid point is called

- centered, if the components are the center of a cell.
- nodal, if the components are on the vertex of a cell.

- staggered, if the components have one nodal coordinate and one centered coordinate, where nodal and centered is a one-dimensional concepts as in Def. 5.1.1.

See, Tab. 7 for details about the concrete positions of the grid point coordinates of the five different grid types.

A visualization for the different grid point positions for $d = 2$, as described in Tab. 7, is shown in Fig. 5.1.2.

Table 8: *Set of grid points for a three-dimensional grid with grid size $m \in \mathbb{N}^3$ and grid width $h \in \mathbb{N}^3$ on a domain $\Omega = [\omega_1^1, \omega_2^1] \times [\omega_1^2, \omega_2^2] \times [\omega_1^2, \omega_2^2]$ with the grid type : nodal ($X^{\mathfrak{N}}$), inner-nodal ($X^{\mathfrak{I}\mathfrak{N}}$), centered ($X^{\mathfrak{C}}$), face-staggered ($X^{\mathfrak{F}\mathfrak{S}}$), inner-face-staggered ($X^{\mathfrak{I}\mathfrak{F}\mathfrak{S}}$), edge-staggered ($X^{\mathfrak{E}\mathfrak{S}}$) and inner-edge-staggered ($X^{\mathfrak{I}\mathfrak{E}\mathfrak{S}}$).*

$$
\begin{aligned}
X^{\mathfrak{N}} = \quad & \{(x^1_{j_1}, x^2_{j_2}, x^3_{j_3})^\top \mid x^i_{j_i} = (\mathfrak{N}_{j_i^1}, \mathfrak{N}_{j_i^2}, \mathfrak{N}_{j_i^3}),\ j_i^k = 1, \dots, m_k + 1\} \\
X^{\mathfrak{I}\mathfrak{N}} = \quad & \{(x^1_{j_1}, x^2_{j_2}, x^3_{j_3})^\top \mid x^i_{j_i} = (\mathfrak{N}_{j_i^1+2}, \mathfrak{N}_{j_i^2+1}, \mathfrak{N}_{j_i^3+1}),\ j_i^k = 1, \dots, m_k - 1\} \\
X^{\mathfrak{C}} = \quad & \{(x^1_{j_1}, x^2_{j_2}, x^3_{j_3})^\top \mid x^i_{j_i} = (\mathfrak{C}_{j_i^1}, \mathfrak{C}_{j_i^2}, \mathfrak{C}_{j_i^3}),\ j_i^k = 1, \dots, m_k\} \\
X^{\mathfrak{F}\mathfrak{S}} = \quad & \{(x^1_{j_1}, x^2_{j_2}, x^3_{j_3})^\top \mid x^1_{j_1} = (\mathfrak{N}_{j_1^1}, \mathfrak{C}_{j_1^2}, \mathfrak{C}_{j_1^3}) \wedge x^2_{j_2} = (\mathfrak{C}_{j_2^1}, \mathfrak{N}_{j_2^2}, \mathfrak{C}_{j_2^3}) \\
& \quad \wedge x^3_{j_3} = (\mathfrak{C}_{j_3^1}, \mathfrak{C}_{j_3^2}, \mathfrak{N}_{j_3^3}),\ j_i^i = 1, \dots, m_i + 1,\ j_i^k = 1, \dots, m_k,\ i \neq k\} \\
X^{\mathfrak{I}\mathfrak{F}\mathfrak{S}} = \quad & \{(x^1_{j_1}, x^2_{j_2}, x^3_{j_3})^\top \mid x^1_{j_1} = (\mathfrak{N}_{j_1^1+1}, \mathfrak{C}_{j_1^2}, \mathfrak{C}_{j_1^3}) \wedge x^2_{j_2} = (\mathfrak{C}_{j_2^1}, \mathfrak{N}_{j_2^2+1}, \mathfrak{C}_{j_2^3}) \\
& \quad \wedge x^3_{j_3} = (\mathfrak{C}_{j_3^1}, \mathfrak{C}_{j_3^2}, \mathfrak{N}_{j_3^3+1}),\ j_i^k = 1, \dots, m_k\} \\
X^{\mathfrak{E}\mathfrak{S}} = \quad & \{(x^1_{j_1}, x^2_{j_2}, x^3_{j_3})^\top \mid x^1_{j_1} = (\mathfrak{C}_{j_1^1}, \mathfrak{N}_{j_1^2}, \mathfrak{N}_{j_1^3}) \wedge x^2_{j_2} = (\mathfrak{N}_{j_2^1}, \mathfrak{C}_{j_2^2}, \mathfrak{N}_{j_2^3}) \\
& \quad \wedge x^3_{j_3} = (\mathfrak{N}_{j_3^1}, \mathfrak{N}_{j_3^2}, \mathfrak{C}_{j_3^3}),\ j_i^i = 1, \dots, m_i,\ j_i^k = 1, \dots, m_k + 1, i \neq k\} \\
X^{\mathfrak{I}\mathfrak{E}\mathfrak{S}} = \quad & \{(x^1_{j_1}, x^2_{j_2}, x^3_{j_3})^\top \mid x^1_{j_1} = (\mathfrak{C}_{j_1^1}, \mathfrak{N}_{j_1^2+1}, \mathfrak{N}_{j_1^3+1}) \wedge x^2_{j_2} = (\mathfrak{N}_{j_2^1+1}, \mathfrak{C}_{j_2^2}, \mathfrak{N}_{j_2^3+1}) \\
& \quad \wedge x^3_{j_3} = (\mathfrak{N}_{j_3^1+1}, \mathfrak{N}_{j_3^2+1}, \mathfrak{C}_{j_3^3}),\ j_i^k = 1, \dots, m_k\}
\end{aligned}
$$

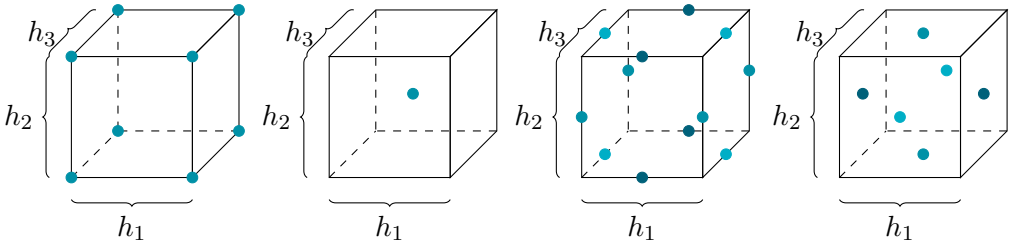

Figure 5.1.3: *Grid types from left to right: nodal, centered, edge-staggered and face-staggered for $d = 3$ with their grid points visualized on a single cuboid cell. Note that for an inner-edge-staggered or inner-face-staggered grid type the grid points are the same as for the edge-staggered or face-staggered grid type for an inner-cell c_j, i.e. $j_i \neq 1$ and $j_i \neq m_i$.*

Definition 5.1.3 (Grid points for $d = 3$)**.** For $d = 3$ a grid point x consists of three components x^1, x^2, x^3 where each component has two coordinates, i.e. $x^i \in \mathbb{R}^3, i = 1, 2, 3$. A grid point is called

- centered, if the components are the center of a cell.

- nodal, if the components are on the vertex of a cell.
- face-staggered, if the components have one nodal coordinate and two centered coordinates, where nodal and centered are one-dimensional concepts as in Def. 5.1.1.
- edge-staggered, if the components have two nodal coordinates and one centered coordinate, where nodal and centered are one-dimensional concepts as in Def. 5.1.1.

See, Tab. 8 for details about the concrete positions of the grid point coordinates of the seven different grid types.

A visualization of the grid point positions for $d = 3$ is given in Fig. 5.1.3 on a single cell.

Table 9: *The number of grid points per coordinate x^i.*

$$N_i = \begin{cases} \prod_{k=1}^{d} m_k, & \text{for a centered grid type,} \\ \prod_{k=1}^{d} (m_k + 1), & \text{for a nodal grid type,} \\ \prod_{k=1}^{d} (m_k - 1), & \text{for an inner-nodal grid type,} \\ (m_i + 1) m_k, & \text{for a staggered grid type with } k \neq i \text{ ,} \\ (m_i - 1) m_k, & \text{for an inner-staggered grid type with } k \neq i \text{ ,} \\ (m_i + 1) \prod_{\substack{k=1 \\ k \neq i}}^{3} m_k, & \text{for a face-staggered grid type,} \\ (m_i - 1) \prod_{\substack{k=1 \\ k \neq i}}^{3} m_k, & \text{for an inner- face-staggered grid type,} \\ m_i \prod_{\substack{k=1 \\ k \neq i}}^{3} (m_k + 1), & \text{for an edge-staggered grid type,} \\ m_i \prod_{\substack{k=1 \\ k \neq i}}^{3} (m_k - 1), & \text{for an inner-edge-staggered grid type.} \end{cases}$$

The number of grid points of a d-dimensional grid is $N = |X| = \sum_{i=1}^{d} N_i$ where N_i is the number of grid points per coordinate x^i according to Tab. 9. For a nodal, inner-nodal or centered grid type it holds $N_1 = N_2 (= N_3)$.

In the following, we rely on the notation utilized in [5] and use a vector $x_h \in \mathbb{R}^N$ that contains all grid points from the set of grid points X. Therefore, a vectorization of the components of the grid points $x_h^i \in \mathbb{R}^{N_i}, i = 1, \ldots, d$ is assumed. These components are stored in a vector $x_h = (x_h^1, \ldots, x_h^d)^\top \in \mathbb{R}^N$.

5.1.2 Discretizations of the Second-Order Divergence-Curl Regularizer

This section describes discretizations of the second-order DC regularizer from Def. 2.2.1 with different boundary conditions on a staggered grid.

We use a rectangular or midpoint quadrature rule [116, §2.1] for the discretization of the integrals, due to its simplicity. Let $f \in C^2(\Omega, \mathbb{R})$ and let Ω_h be a grid with a vector of grid points x_h and grid width h. Then it holds

$$\int_\Omega f(x) \mathrm{d}x = h \sum_{i=1}^{N} f(x_h^i) + \mathcal{O}(h^2), \tag{15}$$

where N is the number of grid points; see [5, Eq. (6.2)]. The $\mathcal{O}(h^2)$ relation follows directly from the approximation error [116, p. 54].

For the discretization of the differential operators $\mathcal{B}$ from Tab. 2 we use short central finite differences. We refer to e.g. [113, 114, 115] for detailed descriptions of finite differences. We prefer the short differences over long differences, as long differences are blind for highly oscillatory input [5, p. 125]. To have these high frequency components in the kernel of the discretized differential operator would be a major drawback for the usage of multigrid methods in Ch. 7; see e.g. [31, p.124].
We explicitly incorporate boundary conditions in the differential operator. We propose a discretization which fulfills sliding boundary conditions; see Def. 4.1.1/Def. 4.1.6. From Rem. 4.1.2 we know the relation to Dirichlet-0 and Neumann-0 boundary conditions and thus also consider those boundary conditions. All three boundary conditions fix the transformation y to the boundary $\partial\Omega$ in a certain sense. Hence, a preregistration is useful for applications where large transformations are necessary to align the images. As boundary conditions further reduce the solution space, they may be interpreted as constraints.
Staggered grids are well known, see e.g. [112], for the discretization of divergence and curl. Hence, we use staggered grids for the discretization of the partial differential operator $\mathcal{B}$ from Tab. 2. For the discretization of the partial differential operator $\mathcal{B}$ with Dirichlet-0 and sliding boundary conditions we use an inner-(face-)staggered discretization. The reason for this is, that the inner-nodal positions naturally correspond to Dirichlet-0 boundary conditions. The centered positions are suitable for Dirichlet-0 and Neumann-0 boundary conditions, which makes an inner-staggered discretization appropriate for Dirichlet-0 and sliding boundary conditions. Moreover, for Neumann-0 boundary conditions we use a (face-)staggered discretization, as a nodal position naturally reflects the boundary conditions.
We discretize the differential operator $\mathcal{B}$, from Tab. 2 as follows: We define

$$
\begin{aligned}
B_h^{\text{div, } d=2} &:= \begin{pmatrix} I_2^c \otimes D_{11}^\eta & D_2^\eta \otimes D_1^c \\ D_2^c \otimes D_1^\eta & D_{22}^\eta \otimes I_1^c \end{pmatrix}, \\
B_h^{\text{curl, } d=2} &:= \begin{pmatrix} -D_2^c \otimes D_1^\eta & I_2^\eta \otimes D_{11}^c \\ -D_{22}^c \otimes I_1^\eta & D_2^\eta \otimes D_1^c \end{pmatrix}, \\
B_h^{\text{div, } d=3} &:= \begin{pmatrix} I_3^c \otimes I_2^c \otimes D_{11}^\eta & I_3^c \otimes D_2^\eta \otimes D_1^c & D_3^\eta \otimes I_2^c \otimes D_1^c \\ I_3^c \otimes D_2^c \otimes D_1^\eta & I_3^c \otimes D_{22}^\eta \otimes I_1^c & D_3^\eta \otimes I_2^c \otimes D_1^c \\ D_3^c \otimes I_2^c \otimes D_1^\eta & D_3^c \otimes D_2^\eta \otimes I_1^c & D_{33}^\eta \otimes I_2^c \otimes I_1^c \end{pmatrix}, \\
B_h^{\text{curl, } d=3} &:= \begin{pmatrix} 0 & -D_3^c \otimes I_2^\eta \otimes D_1^c & I_3^\eta \otimes D_2^c \otimes D_1^c \\ 0 & -D_3^c \otimes D_2^\eta \otimes I_1^c & I_3^\eta \otimes D_{22}^c \otimes I_1^c \\ 0 & -D_{33}^c \otimes I_2^\eta \otimes I_1^c & D_3^\eta \otimes D_2^c \otimes I_1^c \\ D_3^c \otimes I_2^c \otimes D_1^\eta & 0 & I_3^\eta \otimes I_2^c \otimes D_{11}^c \\ D_3^c \otimes D_2^c \otimes I_1^\eta & 0 & I_3^\eta \otimes D_2^c \otimes D_1^c \\ D_{33}^c \otimes I_2^c \otimes I_1^\eta & 0 & D_3^\eta \otimes I_2^c \otimes D_1^c \\ -I_3^c \otimes D_2^c \otimes D_1^\eta & I_3^c \otimes I_2^\eta \otimes D_{11}^c & 0 \\ -I_3^c \otimes D_{22}^c \otimes I_1^\eta & I_3^c \otimes D_2^\eta \otimes D_1^c & 0 \\ -D_3^c \otimes D_2^c \otimes I_1^\eta & D_3^c \otimes I_2^\eta \otimes D_1^c & 0 \end{pmatrix},
\end{aligned}
$$

where $I_j^c \in \mathbb{R}^{m_j, m_j}$ is the identity, $\otimes$ denotes the Kronecker-product, and the other concrete matrices are outlined in Tab. 10 for sliding boundary conditions.
For Dirichlet-0 boundary conditions, we extend the identity I_j^η and the second-order difference operator $D_{j,j}^\eta$ with zero rows, as we know that the grid points on the boundary

Table 10: *First- and second-order finite difference operators for cell-centered and inner nodal grids of size $m = m_1 \times \cdots \times m_d$ with grid width $h_j = (\omega_2^j - \omega_1^j)/m_j$ and sliding boundary conditions; $I_j^\eta \in \mathbb{R}^{m_j-1,m_j-1}$ is the usual identity; first-order: $D_j^c \in \mathbb{R}^{m_j-1,m_j}$ and $D_j^\eta \in \mathbb{R}^{m_j,m_j-1}$ and second-order: $D_{j,j}^c \in \mathbb{R}^{m_j,m_j}$ and $D_{j,j}^\eta \in \mathbb{R}^{m_j-1,m_j-1}$.*

$$D_j^c := \frac{1}{h_j}\begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}, \qquad D_j^\eta := \frac{1}{h_j}\begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & -1 \end{pmatrix},$$

$$D_{jj}^c := \frac{1}{h_j^2}\begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}, \quad D_{jj}^\eta := \frac{1}{h_j^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}.$$

are zero and the linear extension over the boundary leads to a vanishing second-order derivative on the boundary. The first-order difference operator D_j^η remains the same. Again due to the linear interpolation we add ± 2 in an additional row of the operator D_j^c for the boundary points and exchange the entry $(1,1)$ and (m_j, m_j) of the second-order difference operator $D_{j,j}^c$ with -3.
For Neumann-0 boundary conditions, we extend the identity I_j^η as we start on a nodal instead of an inner-nodal grid. Furthermore, we extend the first-order difference operator D_j^c with zero rows, as the first-order derivative vanishes. The second-order operator $D_{j,j}^c$ remains unchanged. As we start on a nodal grid the first-order finite difference operator D_j^η needs no predictions for the points closest to the boundary. For the second-order finite difference operator $D_{j,j}^\eta$ we use a linear interpolation for the boundary, i.e. the difference between the grid point on the boundary and its closest neighbor point.
With these, we define

$$B_h := \begin{pmatrix} \sqrt{\gamma} B_h^{\text{div}} \\ \sqrt{1-\gamma} B_h^{\text{curl}} \end{pmatrix}$$

and conclude

$$A_h = B_h^\top B_h = \gamma (B_h^{\text{div}})^\top B_h^{\text{div}} + (1-\gamma)(B_h^{\text{curl}})^\top B_h^{\text{curl}} \in \mathbb{R}^{N,N}.$$

This discretization of $\mathcal{A}$ based on the discretization of the differential operator $\mathcal{B}$ has the advantage, that A_h is ensured to be symmetric and positive semi-definite. These are properties that are favorable for the optimization, as they e.g. allow for conjugate gradient schemes [52, 53].

Finite Differences and Cancellation

Cancellation is a well-known phenomenon when using finite differences. Cancellation occurs when nearby quantities subject to rounding error are subtracted. The accurate digits cancel each other out, but digits contaminated by rounding error are left. For more details about cancellation we refer to [117].
Since finite differences relies on the difference of rounded function values, cancellation

might occur. It can be identified as the error is weighted by the factor $\frac{1}{h^p}$ where p is the order of the differential operator and $h \in \mathbb{R}$ the considered grid width.

We observe cancellation for $\gamma = 0.5$ for the proposed discretizations. We will exemplarily discuss the situation for the two-dimensional staggered grid discretization of the second-order divergence-curl (DC) regularizer with sliding boundary conditions. Without loss of generality we focus on the entry $\mathcal{A}_{12}$.
In the continuous setting we have

$$\begin{aligned}\mathcal{A}_{12} &= \gamma(\partial_{11}\partial_{12} + \partial_{12}\partial_{22}) - (1-\gamma)(\partial_{12}\partial_{11} + \partial_{22}\partial_{12}) \\ &= \gamma(\partial_{1112} + \partial_{1222}) - (1-\gamma)(\partial_{1112} + \partial_{1222})\end{aligned}$$

and hence $\mathcal{A}_{12} = 0$ for $\gamma = 0.5$. We will see that our discretization mimics the continuous operator, but has a cancellation error. With the notation

$$\begin{aligned} D_h^{11,1} &:= I_2^c \otimes D_{11}^\eta, & D_h^{12,1} &:= D_2^c \otimes D_1^\eta, & D_h^{22,1} &:= D_{22}^c \otimes I_1^\eta, \\ D_h^{11,2} &:= I_2^\eta \otimes D_{11}^c, & D_h^{12,2} &:= D_2^\eta \otimes D_1^c, & D_h^{22,2} &:= D_{22}^\eta \otimes I_1^c, \end{aligned} \tag{16}$$

we describe the block matrix B_h. The first two indices indicate the direction of the derivative and the third index indicates the component the finite difference matrix acts on. Then, we have for the discretization

$$(A_h)_{12} = \gamma((D_h^{11,1})^\top D_h^{12,2} + (D_h^{12,1})^\top D_h^{22,2}) - (1-\gamma)((D_h^{12,1})^\top D_h^{11,2} + (D_h^{22,1})^\top D_h^{12,2}). \tag{17}$$

In the special case $\gamma = 0.5$, we need

$$(D_h^{11,1})^\top D_h^{12,2} - (D_h^{12,1})^\top D_h^{11,2} \quad \text{as well as} \quad (D_h^{12,1})^\top D_h^{22,2} - (D_h^{22,1})^\top D_h^{12,2}$$

to vanish to mimic the behavior of the differential operator $\mathcal{A}$. We examine this in the following experiments. The experiments have surprising results, as the phenomenon of cancellation occurs only in a certain setup.

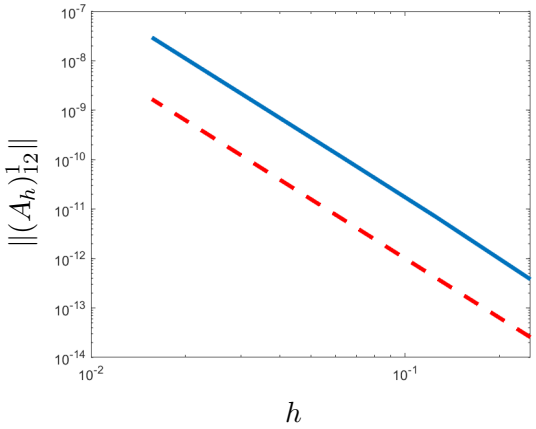

Figure 5.1.4: *Visualization of the observed cancellation in Exp. 5.1.4. Log-log plot of the grid width h against the error $\|(A_h)^1_{12}\|$ in blue together with a slope of -4 in red.*

Experiment 5.1.4 (Cancellation)**.** We determine the entry $(A_h)_{12}$ for $\gamma = 0.5$ in three variants:

- In the first variant we calculate the whole matrix A_h and pick out the entries, that correspond to the block $(A_h)_{12}$. For this variant we use the notation $(A_h)^1_{12}$.
- In the second variant we use the blocks from Eq. (16) of the block-matrix B_h. Here, we manually simulate the matrix-multiplication according to Eq. (17) and use the notation $(A_h)^2_{12}$.
- In the third variant we extract the first and second column of block-matrices from B_h and use $(A_h)^3_{12} = (B^1_h)^\top B^2_h$ where B^1_h is the first block column and B^2_h is the second block column.

We examine $(A_h)^k_{12}$ for $k = 1, 2, 3$. We observe that $(A_h)^1_{12} = (A_h)^3_{12} \neq 0$, but $(A_h)^2_{12} = 0$. Note, that we obtain a zero matrix in a continuous setting. Thus, both matrices $(A_h)^1_{12}, (A_h)^3_{12}$ are expected to vanish for a smaller grid width. We observe the opposite, due to cancellation. For $\Omega = [0, 1]^2$ with $m = 2^j(1, 1)$ with $j = 2, \dots, 8$, we plot the grid width h against the error $\|(A_h)^1_{12}\|$; see Fig. 5.1.4. The error is in $\mathcal{O}(\frac{1}{h^4})$ and thus fits the behavior of cancellation. We are surprised, that it only occurs for the variants, where we use matrix multiplication and not within the manual calculation of $(A_h)_{12}$.

In order to verify, that the error is actually cancellation, we carry out the following experiment with more precise calculations.

Experiment 5.1.5 (Precise Calculations)**.** We repeated Exp. 5.1.4 and used the Matlab [118] function 'vpa' in the first and in the third variant of the calculation of $(A_h)_{12}$. The function 'vpa' allows for more precise calculations. More specifically, we set the number of relevant digits to 100 and save the matrix B_h with 100 digits. In this setting all matrices, including $(A_h)^1_{12}$ and $(A_h)^3_{12}$, are zero-matrices. Thus, the occurring cancellation seems to be a problem of unprecise matrix multiplication within Matlab [118].

Numerical Consistency of the Proposed Discretizations

A consistent discretization ensures, that the application of the finite difference operator to a vector is related to the application of the differential operator to a mapping in the continuous setting. We have the following definition.

Definition 5.1.6 (Consistency of a Discretization, modified LeVeque [114, §2.8])**.** Let $x_h \in \mathbb{R}^N$ be a vector of grid points to a given grid and $A_h \in \mathbb{R}^{N,N}$ be a finite difference discretization of a differential operator $\mathcal{A} : H^k(\Omega, \mathbb{R}^d) \to H^0(\Omega, \mathbb{R}^d)$ for $k \in \mathbb{N}$. Furthermore, $u \in H^k(\Omega, \mathbb{R}^d)$ is a vector-field and $f = \mathcal{A}\,u$. Then the finite difference method is consistent with order $p \in \mathbb{Z}$, if

$$\|A_h u(x_h) - f(x_h)\| \in \mathcal{O}(h^p).$$

We illustrate numerical $\mathcal{O}(h^2)$ consistency for the proposed discretization with sliding boundary conditions. A rigorous proof, similar to [113, Ex. 2.3.1] is left to future work.

The following examples are chosen, such that they fulfill the boundary conditions, but are as simple as possible at the same time. In particular, we set the second component of the considered vector-field u to zero.

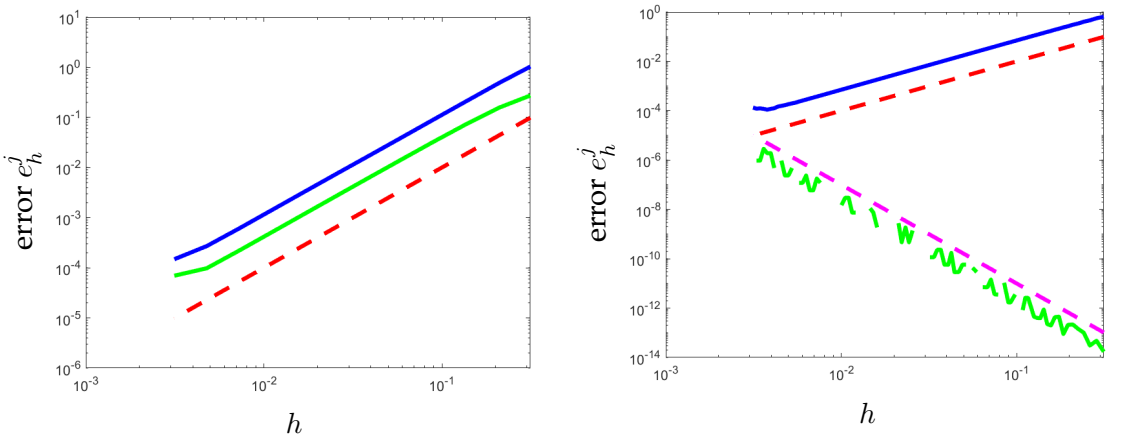

Figure 5.1.5: *Consistency check as described in Exp. 5.1.7 for the discretization with sliding boundary conditions with $\gamma = 0.1$ (left) and $\gamma = 0.5$ (right). Log-log plot of the grid width h against the first (blue) and second (green) component of the consistency error e_h^j. In red a slope of 2 is visualized and in magenta a slope of -4. Cancellation is visible for the second component for $\gamma = 0.5$, as no consistency error occurs. The discretization A_h of the fourth-order differential operator with sliding boundary conditions is thus consistent in $\mathcal{O}(h^2)$.*

Experiment 5.1.7 (Consistency for a Sliding Boundary Discretization)**.** We set $\Omega = [0,1]^2$ and

$$u(x) = \begin{pmatrix} \sin(\pi x_1)\cos(2\pi x_2) \\ 0 \end{pmatrix},$$

as u fulfills the sliding boundary conditions and $u \in C^\infty(\Omega, \mathbb{R}^2)$, we can apply the differential operator $\mathcal{A}$ and have

$$f = \begin{pmatrix} (5 + 16(1-\gamma))\pi^4 \sin(\pi x_1)\cos(2\pi x_2) \\ (2\gamma - 1)\pi^4 \cos(\pi x_1)\sin(2\pi x_2) \end{pmatrix}.$$

For a more detailed insight, we consider the components separately within the consistency check, i.e. we examine the error $e_h^j = \|(A_h u_h)^j - f_h^j\|_\infty$ for $j = 1, 2$, where $u_h = u(x_h)$ and $f_h = f(x_h)$. The grid width h is plotted against the consistency error e_h^j in Fig. 5.1.5. Here, we used the proposed discretization A_h with sliding boundary conditions and the parameter $\gamma = 0.1$ as well as $\gamma = 0.5$. For $\gamma = 0.1$ the error e_h^j is in $\mathcal{O}(h^2)$ for $j = 1, 2$. For $\gamma = 0.5$ the slope of the error is approximately -4 and not continuous for $j = 2$. Since the fourth-order differential operator approximates a constant (here zero) to all relevant digits, cancellation [117] occurs, that is weighted with the factor $\frac{1}{h^4}$. Again using the 'vpa' function of Matlab [118] leads to $e_h^2 = 0$. We expect the cancellation to also occurs for $\gamma \neq 0.5$, but not to be visible as the consistency error is larger. Hence, the discretization A_h is consistent in $\mathcal{O}(h^2)$ for sliding boundary conditions.

Remark 5.1.8 (3D Extension for the Consistency for a Sliding Boundary Discretization). Again, $\mathcal{O}(h^2)$ consistency results can be observed for the three-dimensional extension of

Ex. 5.1.7 with $\Omega = [0,1]^3$ and

$$u(x) = \begin{pmatrix} \sin(\pi x_1)\cos(2\pi x_2)\cos(\pi x_3) \\ 0 \\ 0 \end{pmatrix}, \quad \text{with}$$
$$f(x) = \begin{pmatrix} (6+24(1-\gamma))\pi^4 \sin(\pi x_1)\cos(2\pi x_2)\cos(\pi x_3) \\ 12(2\gamma-1)\pi^4 \cos(\pi x_1)\sin(2\pi x_2)\cos(\pi x_3) \\ 6(2\gamma-1)\pi^4 \cos(\pi x_1)\cos(2\pi x_2)\sin(\pi x_3) \end{pmatrix}.$$

Next, we support the findings for the spectral decomposition from Sec. 4.3 with a numerical example.

Experiment 5.1.9 (Eigenfunctions for $d = 2$)**.** We verify the proposed implementation and build on Ex. 4.3.3. For $m = 2^j(1,1)$ with $j = 1, 2, \ldots, 10$, i.e. $h = \frac{1}{2^j}(1,1)$, we plot the error

$$e_h := \|A_h v_h^k - \lambda^j v_h^k\|_\infty$$

against one component of h in a log-log-plot in Fig. 5.1.6. The error has slope 2, i.e. the error decreases in $\mathcal{O}(h^2)$. For very small h the error rises, which can be explained by cancellation [117]. These findings emphasize the consistency of the discretization with sliding boundary conditions and furthermore support the spectral decomposition from Sec. 4.3.

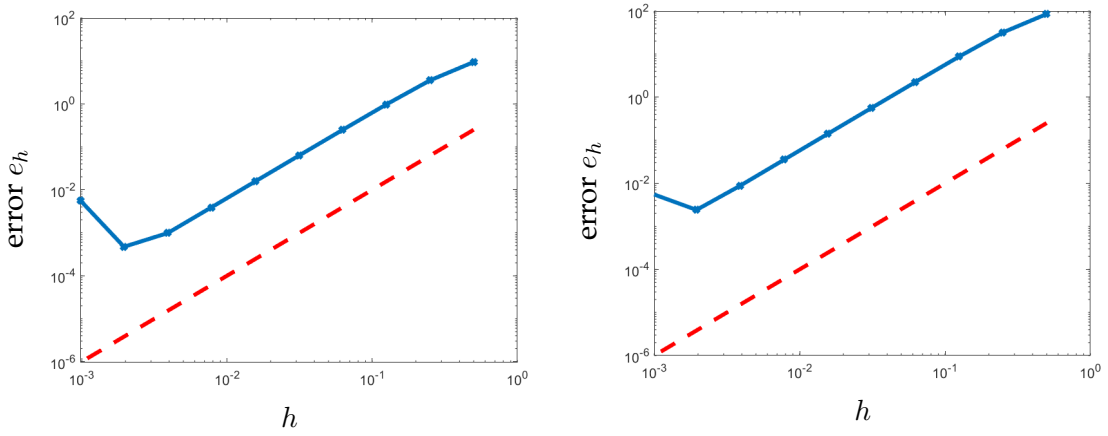

Figure 5.1.6: *Visualization of the consistency for the two eigenfunctions (left: $k = 1$ and right $k = 2$). For details see Exp. 5.1.9. The log-log plot of the error e_h against the grid width h with slope 2 suggest an h^2 approximation of the differential operator $\mathcal{A}$ from Tab. 3. This supports the consistency of the discretization and shows the spectral decomposition of the discretization with sliding boundary conditions.*

Remark 5.1.10. Ex. 5.1.9 was also extended to $d = 3$ and we obtained similar results.

Ex. 5.1.9 strongly suggest a similar relation between the continuous and the discrete spectral decomposition, as it has been proven for the elastic operator [107]. The utilized finite difference operator from Tab. 10 fulfill the relations

$$D_j^{\mathrm{c}} = -(D_j^{\mathrm{in}})^\top, \quad D_{jj}^{\mathrm{c}} = D_j^{\mathrm{in}} D_j^{\mathrm{c}} \quad \text{and} \quad D_{jj}^{\mathrm{in}} = D_j^{\mathrm{c}} D_j^{\mathrm{in}}$$

and hence fit into the discretization framework presented in [107]. A rigorous proof for the discrete spectral decomposition for the second-order DC regularizer with sliding boundary conditions is left to future work.

In contrast to the well-fitting results for sliding boundary conditions, we observe problems with the discretizations with Dirichlet-0 and Neumann-0 boundary conditions. In particular, we have no consistency in grid points close to the boundary.

Experiment 5.1.11 (Consistency for a Dirichlet-0 Discretization)**.** For a function which fulfills Dirichlet-0 boundary conditions we choose $u \in C^\infty(\Omega, \mathbb{R}^2)$ with

$$u(x) = \begin{pmatrix} \sin(\pi x_1)\sin(2\pi x_2) \\ 0 \end{pmatrix}.$$

Thus we obtain

$$f := \begin{pmatrix} (20 - 15\gamma)\pi^4 \sin(\pi x_1)\sin(2\pi x_2) \\ 10(1-2\gamma)\pi^4 \cos(\pi x_1)\cos(2\pi x_2) \end{pmatrix}.$$

In the consistency check, we observe a rising error for smaller grid width h; see Fig. 5.1.7. The difference images suggest, that the location of the problem is in the positions closest to the boundary, i.e. in the discretization of the boundary conditions. This is also confirmed by the log-log-plot in the top right, as the discretization restricted to the inner grid point positions has the expected $\mathcal{O}(h^2)$ consistency.

Remark 5.1.12. A similar behavior of the consistency error can be observed for the proposed discretization with Neumann-0 boundary conditions. Here, we used the example setting $u \in C^\infty(\Omega, \mathbb{R}^2)$ with

$$u(x) = \begin{pmatrix} \cos(\pi x_1)\cos(2\pi x_2) \\ 0 \end{pmatrix}, \quad \text{and } f = \begin{pmatrix} (5 + 16(1-\gamma))\pi^4 \sin(\pi x_1)\cos(2\pi x_2) \\ (2\gamma - 1)\pi^4 \cos(\pi x_1)\sin(2\pi x_2) \end{pmatrix}.$$

We observe that only the discretization with sliding boundary conditions is consistent. The detailed analysis of the discretization problems with Dirichlet-0 and Neumann-0 boundary conditions is the subject of future work.

Numerical Convergence of the Proposed Discretizations

Another criterion to measure the quality of a discretization is to measure, whether the solution of the finite difference scheme converges point-wise to the discretized continuous solution. This property is called convergence of a discretization and is defined as follows.

Definition 5.1.13 (Convergence of a Discretization, [114, §2.9])**.** Let $x_h \in \mathbb{R}^N$ be a vector of grid points to a given grid and $A_h \in \mathbb{R}^{N,N}$ be a finite difference discretization of a differential operator $\mathcal{A} : H^k(\Omega, \mathbb{R}^d) \to H^0(\Omega, \mathbb{R}^d)$ for $k \in \mathbb{N}$. Furthermore, $u \in H^k(\Omega, \mathbb{R}^d)$ is a vector-field and $f = \mathcal{A}\,u$. A finite difference method is convergent with order p, if

$$\|u(x_h) - (A_h)^{-1} f(x_h)\| \in \mathcal{O}(h^p).$$

Again, we concentrate on numerical convergence in this section and a rigorous proof is left to future work.
For the proposed discretization with sliding boundary conditions we observe an $\mathcal{O}(h^2)$ convergent discretization.

Experiment 5.1.14 (Convergence of a Discretization with Sliding Boundary Conditions)**.** We continue Ex. 5.1.7 and determine the convergence error $\epsilon_h^j := \|u_h^j - ((A_h)^{-1} f_h)^j\|_\infty, j = 1, 2$ for the sliding boundary discretization. To determine $(A_h)^{-1} f_h$ we used the build-in Matlab [118] solver 'backslash'. For $\gamma = 0.1$ both components of the error have an $\mathcal{O}(h^2)$ convergence and for $\gamma = 0.5$ cancellation occurs again for ϵ_h^2; see Fig. 5.1.8 (top row). Using the 'vpa' function again leads to the expected $\epsilon_h^2 = 0$.

As the proposed Neumann-0 and Dirichlet-0 discretizations are not consistent, we cannot expect them to be convergent. Surprisingly the Neumann-0 discretization seems to have a perturbed $\mathcal{O}(h^4)$ convergence, whereas the discretization with Dirichlet-0 boundary conditions is not convergent at all.

Experiment 5.1.15 (Convergence of a Discretization with Neumann-0 Boundary Conditions)**.** We continue the example, that was described in Rem. 5.1.12 and determine the convergence error for Neumann-0 boundary conditions. We observe, that the convergence error is not stable but is approximately in $\mathcal{O}(h^4)$; see Fig. 5.1.8 (bottom, left).

Experiment 5.1.16 (Convergence of a Discretization with Dirichlet-0 Boundary Conditions)**.** We continue Ex. 5.1.11 and determine the convergence error for a discretization with Dirichlet-0 boundary conditions. Clearly, the discretization is not convergent; see Fig. 5.1.8 (bottom, right).

Conclusion

We define the discrete analogue of the second-order DC regularizer from Def. 2.2.1 applied to the displacement $u \in C^2(\Omega, \mathbb{R}^d)$

$$S(u_h; \gamma) := \frac{1}{2} \overline{h} \|B_h u_h\|_2^2, \quad \gamma \in [0, 1] \tag{18}$$

with $\overline{h} := \prod_{k=1}^d h_k$. We propose finite difference operators B_h that rely on a staggered grid discretization with three different boundary conditions. We use short finite differences with Dirichlet-0, Neumann-0 and sliding boundary conditions.
The numerical examples support the suitability of the discretization with sliding boundary conditions, as the discretization is consistent and convergent. For the proposed discretizations for Dirichlet-0 and Neumann-0 boundary conditions we can neither ensure consistency nor convergence. The numerical examples suggest, that the boundary conditions are not implemented correctly. Using linear interpolation to model the value of grid points outside the domain might be the problem. These examples thus strongly suggest the use of the discretization with sliding boundary conditions.

5.1.3 The Discrete Image Registration Problem

Next, we outline the discretization of the data-fitting term SSD $\mathcal{D}^{\text{SSD}}$ from Def. 2.1.3 and follow the ideas from [5, §7.1]. Based on a centered grid and the midpoint quadrature

rule (15), we obtain the following definition of the discretized SSD. Let $x_h \in \mathbb{R}^N$ be a given vector of grid points. Let $y \in C^2(\Omega, \mathbb{R})$, and $\mathcal{R}, \mathcal{T} \in I(d) \cap C^2(\Omega, \mathbb{R})$ then for $y_h = y(x_h)$, $T_y = \mathcal{T}(y(x_h))$ and $R = \mathcal{R}(x_h)$, the discretized SSD is

$$D^{\text{SSD}} : \mathbb{R}^N \to \mathbb{R}, \quad D^{\text{SSD}}(y_h) = \frac{1}{2}\overline{h}\|T_y - R\|_2^2 \tag{19}$$

with $\overline{h} := \prod_{k=1}^{d} h_k$.

We obtain the following discrete image registration problem, which is a discrete analogue of Eq. (10). For a given grid Ω_h, $x_h \in \mathbb{R}^N$ is a vector of grid points, D^{SSD} is the discretized SSD (19) and S is the discretized second-order DC regularizer (18). For given $\alpha \in \mathbb{R}_{\geq 0}$, $\gamma \in [0, 1]$ we minimize

$$J : \mathbb{R}^N \to \mathbb{R}, \quad J(u_h) = D^{\text{SSD}}(x_h + u_h) + \alpha S(u_h; \gamma). \tag{20}$$

5.2 Considered Methods of Numerical Optimization

The discretized minimization problem (20) is finite dimensional and can be solved with standard algorithms from numerical optimization. We refer to [53, 119] for an introduction to optimization methods.

Standard Gauss-Newton optimization schemes [53, §10.3] from the family of least-square Newton approaches are used to compute a minimizer of the discretized image registration problem (20). As the problem formulation is based on a sum of squares, Gauss-Newton algorithms are suitable optimizers, as e.g. demonstrated in [46, 120, 121]. Gauss-Newton schemes are based on first-order derivatives and use an approximation of the Hessian [53, p.254]. This approximation is practical for image registration problems as the Hessian of the data-fitting term is data-dependent and it is generally not guaranteed to be a symmetric and positive definite matrix. The approximation ensures these properties. Thus, a search direction for a minimizer is obtained.
A disadvantage of Gauss-Newton optimization is that convergence [53, p.255-257] can be slow and is not even guaranteed. This occurs if an initial guess is far from the minimum or the approximation of the Hessian is ill-conditioned. A preregistration potentially leads to an initial guess close to the minimum. On the other hand Gauss-Newton optimization can approach quadratic convergence, or at least rapid local convergence, if the first order term dominate the second-order term; see [53, p.257] for details.

To further simplify the algorithms, inexact Gauss-Newton schemes [53, p.257] are commonly used for variational image registration; see e.g. [46]. Inexact has the meaning that the potentially large linear system

$$H(y_h)v_h = -\nabla J(y_h), \tag{21}$$

is not solved exactly for the desired descent direction v_h within the Gauss-Newton iteration, but with an iterative procedure. Here, H is the approximation of the Hessian and ∇J the gradient of the discretized objective function. As the linear system (21) needs to be solved in each iteration step, efficient linear solvers are needed to obtain a reasonable performance of the Gauss-Newton algorithm. A direct solution with e.g. Gaussian elimination or LU factorization [122], is not feasible for most image registration problems within standard software as in Matlab [118]. In this thesis, we consider conjugate gradient and multigrid solver, as they are the standard solver in FAIR [5].

Conjugate gradient solvers [52, 53], are suitable for the linear system (21), as the approximation of the Hessian H is symmetric and positive semi-definite by design. The method converges to the solution of the linear system (21) in at most N steps, where N is the number of unknowns [53, Thm. 5.1]. In fact, if the approximation of the Hessian has multiple eigenvalues, much less than N iterations are needed; see [53, p.112] for details.

As the construction of a multigrid solver is one of the overarching goals of this thesis, we refer to Ch. 6 for an introduction to multigrid.

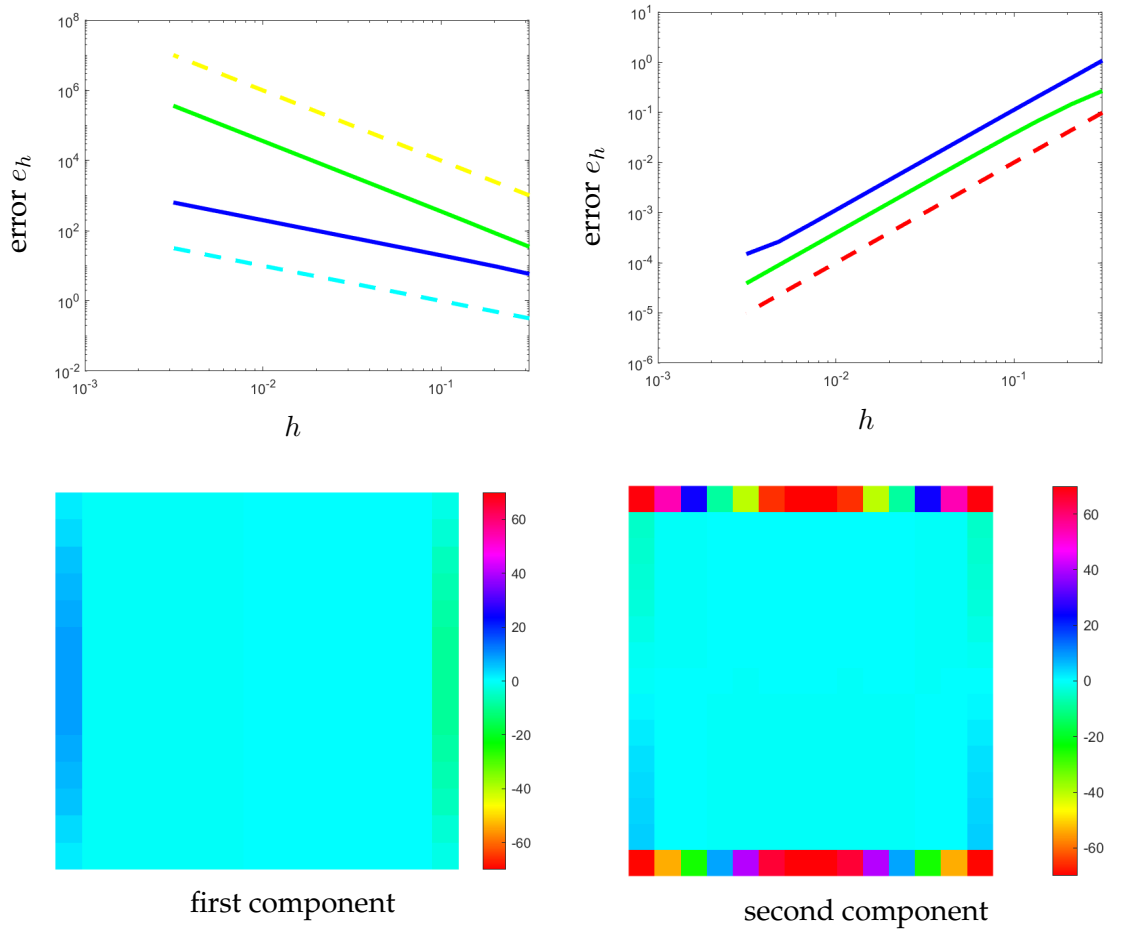

Figure 5.1.7: *Visualization of the consistency check for a discretization with Dirichlet-0 boundary conditions. For details see Exp. 5.1.11. Top row: Log-log plot of the grid width h against the first (blue) and second (green) component of the consistency error e_h. Left the components of the error and right the components of the boundary-omitted components of the error. A slope of 2 is visualized in red, a slope of -1 in yellow and a slope of -2 in cyan. Bottom row: The first (left) and second (right) components of the difference $A_h u_h - f_h$. The visualizations suggest a problem in the discretization of the boundary.*

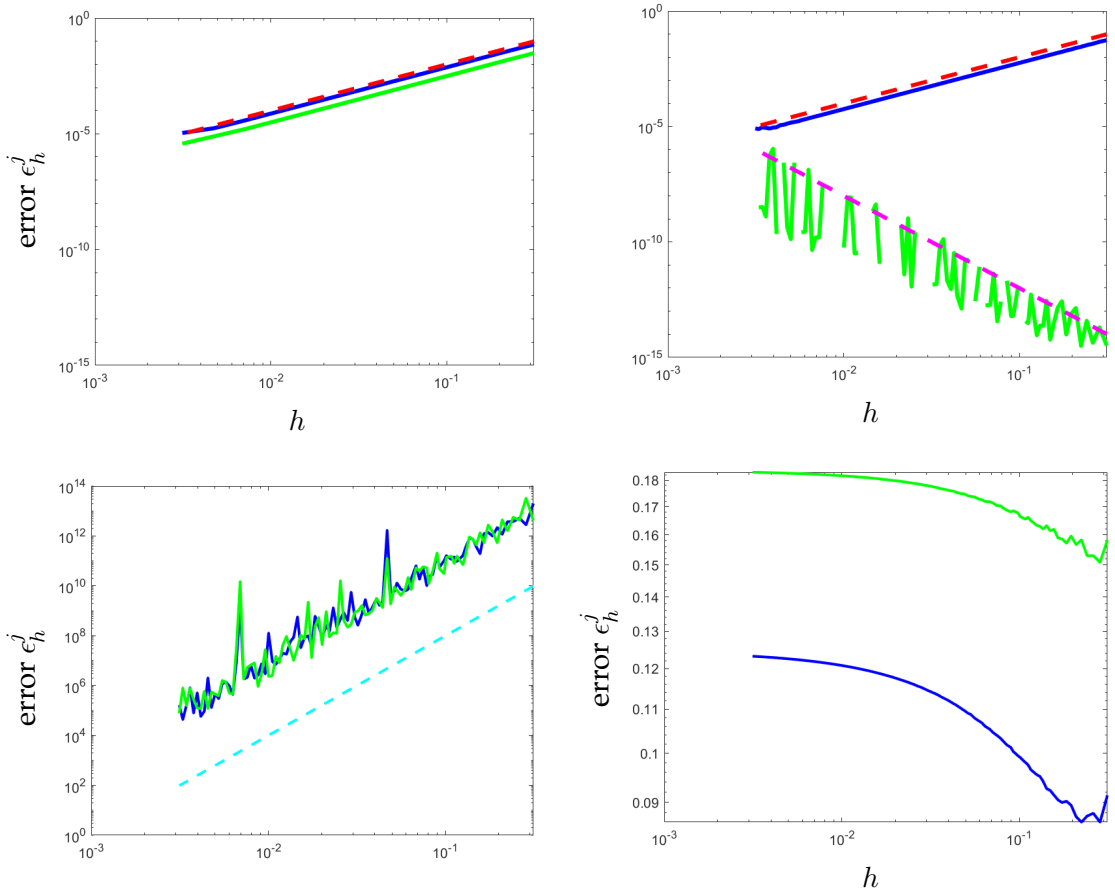

Figure 5.1.8: *Visualization of the convergence check for the discretization with sliding boundary conditions (top), Neumann-0 (bottom, left) and Dirichlet-0 (bottom, right) boundary conditions; see Exp. 5.1.14, Exp. 5.1.15 and Exp. 5.1.16 for details. Log-log plot of the grid width h against the error ϵ_h^j for the first (blue, $j = 1$) and second (green, $j = 2$) component. For sliding boundary conditions we use $\gamma = 0.1$ (top, left) and $\gamma = 0.5$ (top, right), for the other discretization we have $\gamma = 0.1$. The slopes of 2 in red, of -4 in magenta and of 4 in cyan help to emphasize, that the discretization with sliding boundary conditions is $\mathcal{O}(h^2)$ convergent, whereas the Neumann-0 discretization can be approximated by an h^4 term. The discretization with Dirichlet-o boundary conditions is not convergent at all.*

6 Fundamental Concepts of Multigrid

In this chapter, the basic concepts of multigrid methods are introduced to pave the way for an introduction of a multigrid solver for an image registration problem with second order divergence-curl (DC) regularization in the following Ch. 7. Multigrid solvers are of particular interest, as they solve linear systems of equations, such as in Eq. (21), with complexity $\mathcal{O}(N)$, where N is the number of unknowns.

Multigrid methods are efficient iterative solution methods for a large class of problems, that can be traced back to the solution of a linear system of equations. The underlying idea is to combine smoothing effects of classical iterative solvers, such as Jacobi-type iteration, with a representation of the problem on several discretizations. We study geometric multigrid solvers in this thesis. Furthermore, we illustrate the idea of multigrid methods using the Poisson equation as an example. The content of this chapter is mainly based on [31, §§1-2, §8] if not stated otherwise. For a more detailed introduction to multigrid methods, we refer to [31, 32, 33, 123].

The structure of this chapter is as follows: In Sec. 6.1, the fundamental concepts of multigrid methods are outlined on a two-grid cycle, the simplest multigrid solver. Moreover, the main components of a multigrid framework and a generalization to the two most common multigrid architectures, a V- and a W-cycle, are presented in more detail in Sec. 6.2. Additionally, local Fourier analysis (LFA), a tool to examine convergence properties of a multigrid solver in a simplified way, is presented in Sec. 6.3.

6.1 A Two-Grid Cycle

In this section, we provide an overview about multigrid and its components. For the ease of presentation, we consider the simplest multigrid solver, a two-grid cycle, to explain the basic idea of multigrid. This section is based on [31, §2.2].

Multigrid solvers were originally designed for partial differential equations (PDEs), but the concepts were extended to a broad area of other problem-types. Here, we concentrate on a discretized version of a differential equations. The purpose of a multigrid solver is i.e. to find an approximation of the solution of a linear system of equations of the form

$$A_h u_h = f_h, \tag{22}$$

where $u_h, f_h \in \mathbb{R}^N$ are discretizations of functions $u, f : \Omega \to \mathbb{R}^d$, i.e. $u_h := u(x_h) \in \mathbb{R}^N$ for a vector of grid points x_h. The matrix $A_h \in \mathbb{R}^{N,N}$ is a discretization of a differential operator on the grid Ω_h. In this thesis, we assume A_h to be a finite difference operator. Moreover, in the context of multigrid methods, we consider the representation of the finite difference matrix $A_h \in \mathbb{R}^{N,N}$ with stencils. Stencils are defined as follows.

Definition 6.1.1 (Difference stencils, modified Trottenberg [31, p. 297])**.** Each entry $A_h^{k,l}$ of a finite difference operator can be represented by difference stencils, i.e. $A_h^{k,l} \hat{=} [s_\kappa^{k,l}]_h,\ \kappa \in \mathbb{Z}^d$ where

$$A_h^{k,l} y_h^l(x) = \sum_{\kappa \in V} s_\kappa^{k,l} y_h^l(x + \kappa h), \quad k, l = 1, \dots, d.$$

Here, the vector y_h is a discretization of a function, $x \in X$ is a grid point and $\kappa h = (\kappa_1 h_1, \ldots, \kappa_d h_d)$, where h is the grid width. Moreover, $V \subset \mathbb{Q}^d$ is a finite index set.

Note, that the coefficients potentially differ for grid points close to the boundary and hence one single stencil per entry is usually not enough to represent a difference operator.
The following example gives an intuition to stencil notation.

Example 6.1.2 (Discretization of the Poisson Equation)**.** The continuous Poisson equation has the form

$$\Delta u(x) = f(x) \quad \text{for } x \in \Omega = (0,1)^2$$

for $u \in C^2(\Omega, \mathbb{R}^2)$ with Dirichlet-0 boundary conditions, i.e. $u|_{\partial\Omega} = 0$. The components u_1, u_2 are uncoupled and hence can be solved independently of each other. For the discretization we consider an inner-nodal grid with grid size $m = (8,8)$, i.e. grid width $h = \frac{1}{8}(1,1)$. For the discretization of the differential operator

$$\mathcal{A} = \Delta = \begin{pmatrix} \partial_{11} + \partial_{22} & 0 \\ 0 & \partial_{11} + \partial_{22} \end{pmatrix}$$

we consider the standard central finite differences and hence the operator

$$D_{jj} = \frac{1}{(\frac{1}{8})^2} \begin{pmatrix} 2 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & -1 & 2 & -1 & & & & \\ & & -1 & 2 & -1 & & & \\ & & & -1 & 2 & -1 & & \\ & & & & -1 & 2 & -1 & \\ & & & & & -1 & 2 & \end{pmatrix} \in \mathbb{R}^{7,7}.$$

From this, we deduce the corresponding difference operator

$$A_h = \Delta_h = \begin{pmatrix} I \otimes D_{11} + D_{22} \otimes I & 0 \\ 0 & I \otimes D_{11} + D_{22} \otimes I \end{pmatrix} \in \mathbb{R}^{98,98},$$

where $I \in \mathbb{R}^{7,7}$ is the identity matrix. As $h_1 = h_2$, the entries $(A_h)_{1,1} = (A_h)_{2,2}$ can also be displayed by the five-point stencils

$$\frac{1}{(\frac{1}{8})^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}$$

or its analogues for the boundary, as e.g.

$$\frac{1}{(\frac{1}{8})^2} \begin{bmatrix} & -1 & \\ 0 & 4 & -1 \\ & -1 & \end{bmatrix} \quad \text{or} \quad \frac{1}{(\frac{1}{8})^2} \begin{bmatrix} & 0 & \\ 0 & 4 & -1 \\ & -1 & \end{bmatrix}.$$

In a next step, we reformulate the linear problem from Eq. (22). As the solution is obtained iteratively, for each approximation u_h obtained from an iterative solver, the error and the residual are defined as follows.

Definition 6.1.3 (Error and Residual, Trottenberg [31, Eq. (2.2.2) and (2.2.3)])**.** For an

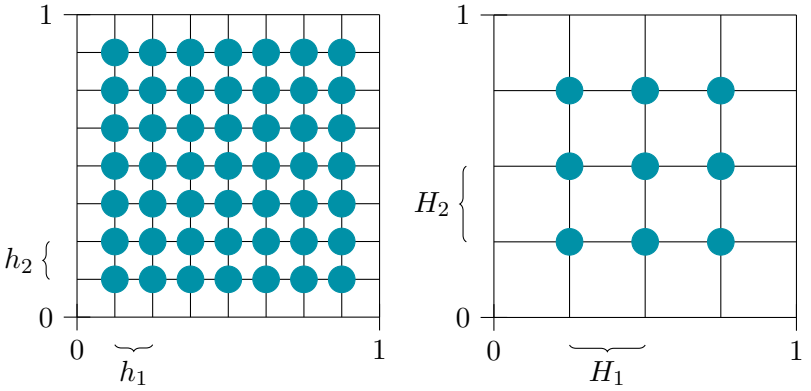

Figure 6.1.1: *Inner-nodal grid Ω_h (left) for $\Omega = (0,1)^2$ with grid size $m = (8,8)$, i.e. grid width $h = (\frac{1}{8}, \frac{1}{8})$ and its coarse grid Ω_H (right) with grid size $M = (4,4)$, i.e. grid width $H = (\frac{1}{4}, \frac{1}{4})$.*

approximation u_h of the exact solution u_h^* of the linear system Eq. (22) the error is

$$e_h := u_h^* - u_h$$

and the residual (sometimes also called defect) is

$$r_h := f_h - A_h u_h.$$

Starting with the reformulation, the original equation (22) can be expressed as the so called defect equation

$$A_h e_h = r_h, \tag{23}$$

where e_h is the error and r_h the residual of an approximation u_h according to Def. 6.1.3.

The linear system (22) is discretized within a multigrid framework. More precisely, a hierarchical structure of the grids is formulated. For the ease of presentation, we assume to have a grid Ω_h with a grid size $m \in \mathbb{N}^d$ that is a multiple of two. For a given grid Ω_h, its corresponding coarse grid Ω_H is defined as follows.

Definition 6.1.4 (Coarse Grids, modified Trottenberg [31, p. 17])**.** Given a grid Ω_h, see Sec. 5.1, then its coarse grid Ω_H has the same domain Ω, a grid size M where $2M = m$ and the same grid type.

Note, that the set of grid points X from the coarse grid Ω_H can be deduced from the domain, the grid size and the grid type. Furthermore, for the grid width H of the coarse grid it holds $H = 2h$.

Example 6.1.5 (Coarse Grid)**.** We illustrate the basic idea of a coarse grid, for $\Omega = (0,1)^2, m = (8,8)$ for the inner-nodal grid type. A visualization is provided in Fig. 6.1.1. The grid points of the grid Ω_h are in the set

$$X_h = \{x \in (0,1)^2 | x = \frac{1}{8}(j,k), j,k = 1,2,\ldots,7\}$$

with $N = |X_h| = 49$. The corresponding coarse grid Ω_H with $m = (2, 2)$ has the set of grid points

$$X_H = \{x \in (0,1)^2 | x = \frac{1}{4}(j,k), j,k = 1,2,3\}$$

with $|X_H| = 9$. Furthermore, $X_H \subset X_h$ holds as we have an inner-nodal discretization.

As multigrid is based on the smoothing of the high frequency components, we want to give an intuition of the idea of high and low frequencies in the following example. A proper definition follows later on.

Example 6.1.6 (High and Low Frequencies)**.** We consider the Poisson equation and follow [31, p.16-17]. It is well-known that the discrete operator $A_h = \Delta_h$ has the discrete eigenfunctions

$$\phi_h^k(x) = \phi_h^{k_1,k_2}(x) = I_2 \otimes \sin(k_1 \pi x_1) \sin(k_2 \pi x_2),$$

for $k_i = 1, 2, \dots m_i - 1$. On the coarse grid, i.e. $x \in X_H$ we have the so called aliasing of the frequencies $\phi_h^k(x)$. In particular,

$$\phi_h^k(x) = -\phi_h^{m_1-k_1,k_2}(x) = -\phi_h^{k_1,m_2-k_2}(x) = \phi_h^{m_1-k_1,m_2-k_2}(x) \quad \text{for } x \in X_H,$$

i.e. these four eigenfunctions cannot be distinguished on the coarse grid Ω_H. We call the eigenfunctions with $k_i < \frac{m_i}{2}$ for $i = 1, 2$ low frequency components and the remaining ones high frequency components. The high frequency components are not visible on the coarse grid.

Multigrid solvers consist of an efficient interplay between the two concepts of error smoothing and coarse grid correction. Error smoothing describes the process of an iterative solver being applied to a linear system leading to a decrease of the high frequency components of the error. The coarse grid correction reformulates the linear system with respect to the error and solves the reformulated linear system on a coarser discretization. The advantage of this procedure is, that the solution of a linear system with a coarser discretization is less complex, compared to the original discretization.

A two-grid cycle is the smallest version of a multigrid solver. We comment on the components of a multigrid cycle, but more details are provided in Sec. 6.2. A two-grid cycle consists of the following steps:

1. **Pre-smoothing:** For a suitable smoothing procedure, the high frequency components of the error e_h, see Def. 6.1.3, of the current approximation u_h decrease. The smoothing is repeated for $n_1 \in \mathbb{N}$ iterations.
2. **Restriction:** The linear system is restricted to a discretization on the coarse grid. This is the so called coarse grid principle, i.e. the solution of a linear system, discretized on a coarse grid Ω_H, is less expensive than the solution on the original discretization.
3. **Coarse grid solution:** In particular, the linear system that is solved on the coarse grid discretization is the defect equation $A_H e_H = r_H$ and the solution e_H is called the coarse grid solution.
4. **Prolongation:** The coarse grid solution e_H of the defect equation is transferred back to a discretization on the original grid Ω_h with the so called prolongation operator P_H^h.

5. **Correction:** The approximated error e_h is added to the smoothed approximation u_h. This update is also called correction step.
6. **Post-smoothing:** As the approximated error e_h might contain some high frequency components, due to the grid transfers, another smoothing step is applied to the corrected approximation u_h for $n_2 \in \mathbb{N}$ iterations.

6.2 Multigrid Components and Generalization

In this section, we present the components: smoothing, prolongation and restriction as well as coarse grid operators for a multigrid algorithm. For each of the components common choices are provided as examples. Moreover, we present the two most common architectures of a general multigrid solver.

6.2.1 Smoothing

Smoothing is one of the two main strategies of a multigrid solver. A smoother is an iterative solver for the linear system in Eq. (22), that leads to a reduction of the error, see Def. 6.1.3, in the high frequency components. Whether an iterative solver is a smoother, is problem-specific.

We assume that an iterative procedure can be formulated in the form

$$A_h^+ u_h^{k+1} + A_h^- u_h^k = f_h, \quad \text{with} \quad A_h^+ + A_h^- = A_h, \tag{24}$$

where $A_h \in \mathbb{R}^{N,N}$ is the discretized differential operator from Eq. (22), u_h^{k+1} is the updated approximation and u_h^k is the approximation before the iteration. Since we want to solve for the updated approximation u_h^{k+1}, we assume that A_h^+ is invertible. The representation of A_h via $A_h^+, A_h^- \in \mathbb{R}^{N,N}$ is called splitting. From this splitting, the so called smoothing operator is defined as follows.

Definition 6.2.1 (Smoothing Operator, Trottenberg [31, p. 298])**.** Given a splitting of the operator A_h as in Eq. (24) for an iterative solver with $(A^+h)^{-1}$ existing, then the smoothing operator S_h is defined as

$$S_h := -(A_h^+)^{-1} A_h^-.$$

Note, that the name smoothing operator already suggests that the considered iterative procedure is a smoother for a certain problem. This is not the case for a general iterative solver, as a smoother is problem specific. In particular, the smoothing operator $S_h \in \mathbb{R}^{N,N}$ is used to determine if the underlying iterative solver is in fact a smoother. In Sec. 6.3, we will present a local Fourier analysis of the smoothing operator S_h, as a tool to determine, if a certain iterative procedure is a smoother for a specific problem.

The ω-damped Jacobi type iteration is presented in the following and its smoothing operator is explicitly determined. For more details and especially more advanced solvers the reader is referred to [31, 123, 124]. Jacobi-type solvers are based on the concept of splitting the matrix $A_h \in \mathbb{R}^{N,N}$ from Eq. (22) up into

$$A_h = D_h - L_h - U_h,$$

where D_h is the diagonal of A_h, and $-L_h$ and $-U_h$ are the strictly lower and upper triangular parts of A_h. This representation is used to define the iterative update. Assuming that D_h^{-1} exists, Eq. (22) is rewritten as

$$u_h = D_h^{-1}(L_h + U_h)u_h + D_h^{-1} f_h.$$

For a given start vector u_h^0 the deduced k-th ω-damped Jacobi-type iteration has the form

$$u_h^{k+1} = [(1-\omega)I_h + \omega D_h^{-1}(L_h + U_h)]u_h^k + \omega D_h^{-1} f_h \tag{25}$$

with a weighting (damping) factor $\omega \in (0, 1]$. The case $\omega = 1$ is called the classical Jacobi-type iteration.
To determine the smoothing operator from Def. 6.2.1, we find that the splitting matrices from Eq. (24) have the form

$$A_h^+ = \frac{1}{\omega} D_h, \quad A_h^- = -(L_h + U_h + \frac{1-\omega}{\omega} D_h)$$

for ω-damped Jacobi-type iterations. For the ω-damped Jacobi-type relaxation, the smoothing operator $S_h(\omega)$ from Def. 6.2.1 depends on the relaxation parameter ω. It therefore depends on the choice of the relaxation parameter ω, whether the ω-damped Jacobi-type iteration is a smoother for a certain problem.

The following example illustrates that the ω-damped Jacobi-type iteration is a smoothing procedure for the Poisson-equation, for $\omega \in (0, 1]$.

Example 6.2.2 (Smoothing Operator)**.** We outline the ω-damped Jacobi-type iteration for the Poisson equation; see [31, Ex. 4.3.3]. We have $D = 4I$ and hence

$$A_h^+ = \frac{4}{h^2\omega} I \quad \text{and} \quad A_h^- = A_h - \frac{4}{h^2\omega} I.$$

For the ease of presentation we focus on the inner grid points and obtain the stencil representation

$$(A_h^+)_{jj} = \frac{1}{h^2} \begin{bmatrix} 0 & \frac{0}{\omega} & 0 \\ \frac{4}{\omega} & & \\ 0 & & \end{bmatrix} \quad \text{and} \quad (A_h^-)_{jj} = \frac{1}{h^2} \begin{bmatrix} & \frac{-1}{\omega} & \\ -1 & \frac{4(\omega-1)}{\omega} & -1 \\ & \frac{-1}{\omega} & \end{bmatrix} \quad \text{for } j = 1, 2$$

and $(A_h^{\pm})_{ij} = 0$ for $i \neq j$. Thus, the smoothing operator has the stencil entries

$$(S_h)_{jj} = \frac{\omega}{4} \begin{bmatrix} & \frac{1}{\omega} & \\ 1 & \frac{4(1-\omega)}{\omega} & 1 \\ & \frac{1}{\omega} & \end{bmatrix} \quad \text{for } j = 1, 2$$

and $(S_h)_{ij} = 0$ for $i \neq j$.
Note, that the eigenfunction ϕ_h^k of the smoothing operator S_h are the same as those of the of the Laplace operator. The corresponding eigenvalues are

$$\lambda_k(S_h) = 1 - \frac{\omega}{2}(2 - \cos(k_1\pi h_1) - \cos(k_2\pi h_2)), \quad k_i = 1, \ldots, m_i - 1;$$

see [31, p.30]. The eigenvectors of the Laplace operator A_h correspond to the eigenfunctions of the continuous Poisson problem. As we can expand fairly arbitrary functions using this set of eigenfunctions, it is also possible to expand arbitrary vectors in terms

of a set of eigenvectors [123, p.19]. Let e_h^0 be the initial guess in the ω-damped Jacobi-type iteration. The vector can be represented by the eigenvectors of the finite difference operator A_h, i.e.

$$e_h^0(x) = \sum_k \alpha_k \phi_h^k(x)$$

for coefficients α_k with $k = (k_1, k_2)^\top$ and $k_1 = 1, \ldots, m_1 - 1, k_2 = 1, \ldots, m_2 - 1$. We use the relation $e_h^m = (S_h)^m e_h^0$, see [123, p.19], and obtain

$$e_h^m(x) = \sum_{k,l} \alpha_k (S_h)^m \phi_h^k(x) = \sum_k \alpha_{k,l} (\lambda_k(S_h))^m \phi_h^k(x),$$

where $(\lambda_{k,l}(S_h)) = 1 - \frac{\omega}{2}(2 - \cos(k_1 \pi h) - \cos(k_2 \pi h))$ are the eigenvalues of the smoothing operator. As the objective of the smoother is a reduction of the high frequency components in the error,

$$\max_k |(\lambda_k(S_h))| < 1$$

needs to be ensured for $\frac{m_i}{2} \leq k_i < m_i - 1$. In fact, $|(\lambda_k(S_h))| < 1$ holds for $\omega \in (0, 1]$. Hence, the ω-damped Jacobi iteration with $\omega \in (0, 1]$ is a smoother for the Poisson equation.

This example demonstrates, that the known discrete eigenfunctions of the finite difference operator often are transferable to the smoothing operator. This enables the analysis of the smoothing properties of the smoothing operator via its eigenvalues.

6.2.2 Restriction and Prolongation

The restriction of the residual and the prolongation of the approximated error are crucial components of the coarse grid principle. The task of the restriction operator is to map the discretization of a function on a grid Ω_h to a discretization on the coarse grid Ω_H. The prolongation operator has the opposite task: it maps a discretization of a function on a coarse grid Ω_H to a discretization on the grid Ω_h. These operators for restriction and prolongation are called transfer operators and described in more detail in this section.

Definition 6.2.3 (Transfer Operators, modified Trottenberg [31, p. 42])**.** For a discretization f_h of a function on the grid Ω_h a restriction operator $R_h^H \in \mathbb{R}^{N,n}$ is an operator that maps f_h to a discretization on the coarse grid Ω_H. The prolongation operator $P_H^h \in \mathbb{R}^{n,N}$ is an operator that maps a discretization f_H of a function on the coarse grid Ω_h to a discretization f_h on the grid Ω_h.

Moreover, we present some of the most common choices for transfer operators based on [31, § 2.3.2, §2.8.4]. Particularly, one-dimensional transfer operators are presented, as we use Kronecker products to obtain transfer operators for the two- and three-dimensional setting.

Restriction Operators

In the following, we present specific choices for restriction operators R_h^H for a one-dimensional problem. For the ease of presentation we assume the underlying grids to be large enough and the grid size m to be a multiple of 2 such that the presented transfer

operators are defined. Based on the grid type of the grid used for the discretization of the functions, different restriction options are distinguished.

A nodal discretization leads to straightforward transfer operators, as the grid points of the coarse grid are a subset of the grid points of the original grid. The injection operator, see visualization in Fig. 6.2.1 top left, uses the subset relation. A major drawback of injection restriction, is the loss of the information of the high frequency components. A more sophisticated restriction operator is the full-weighting, see Fig. 6.2.1 (top, right), that incorporates the information of the neighboring grid points.
The following example demonstrates the effect of losing the high frequency information.

Example 6.2.4 (Restriction Operators on a Nodal Grid)**.** Let $\Omega = [0,1]$, $m = 4$, i.e. $M = 2$. We have given the function $f(x) = \sin(2\pi x)$. Thus, the nodal grid points $x_h \in \mathbb{R}^5$ with the discretized function $f_h \in \mathbb{R}^5$ are

$$x_h = \left(0, \tfrac{1}{4}, \tfrac{1}{2}, \tfrac{3}{4}, 1\right)^\top \in \mathbb{R}^5, \quad \text{with} \quad f_h = \left(0, 1, 0, -1, 0\right)^\top \in \mathbb{R}^5.$$

Hence, with the injection operator

$$(R^{\mathrm{n}})_h^H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3,5},$$

we obtain the discretized coarse grid function

$$f_H = \left(0, 0, 0\right)^\top \in \mathbb{R}^3$$

corresponding to the grid points $x_H = (0, \frac{1}{2}, 1)^\top$. The information about the oscillations of the high frequency sines are completely lost. In this example the problem of losing the high frequency components also occurs for the full-weighting operator

$$(R^{\mathrm{n}})_h^H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3,5}$$

due to the symmetry of the sine.

For a centered discretization the simplest strategy is averaging; see Fig. 6.2.1 (middle, left). A strategy to incorporate more neighboring information is derived from the corresponding linear interpolation, hence we call it linear weighting. Note that due to the consideration of several neighboring points, we have to make assumptions about the boundary behavior of the discretized function that is restricted. Here, we consider Neumann-0 and Dirichlet-0 boundary conditions. A visualization is given in Fig. 6.2.1.

Example 6.2.5 (Restriction Operators on a Centered Grid)**.** Let $\Omega = [0,1]$. To incorporate four neighboring points, we set $m = 8$ and $f(x) = \sin(4\pi x)$. With a cell centered grid we have the centered grid points $x_h \in \mathbb{R}^8$ with the discretized function $f_h \in \mathbb{R}^8$ with

$$\begin{aligned} x_h &= \left(\tfrac{1}{16}, \tfrac{3}{16}, \tfrac{5}{16}, \tfrac{7}{16}, \tfrac{9}{16}, \tfrac{11}{16}, \tfrac{13}{16}, \tfrac{15}{16}\right)^\top \\ f_h &= \left(\tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}\right)^\top. \end{aligned}$$

The application of the averaging operator

$$(R^{\mathrm{c}})_h^H = \frac{1}{2}\begin{pmatrix} 1 & 1 & & & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & & 1 & 1 \end{pmatrix} \in \mathbb{R}^{4,8}$$

leads to

$$f_H = \left(\tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}\right)^\top.$$

Assuming Neumann-0 boundary conditions for f, we use the linear weighting restriction operator of the form

$$(R^{\mathrm{c,N}})_h^H = \frac{1}{8}\begin{pmatrix} 5 & 4 & 1 & & & & & \\ & & 1 & 3 & 3 & 1 & & \\ & & & & 1 & 3 & 3 & 1 \\ & & & & & & 1 & 4 & 5 \end{pmatrix} \in \mathbb{R}^{4,8}.$$

We end up with

$$f_H = \left(\tfrac{1}{\sqrt{2}}, -\tfrac{1}{2\sqrt{2}}, \tfrac{1}{2\sqrt{2}}, -\tfrac{1}{\sqrt{2}}\right)^\top.$$

Using Dirichlet-0 boundary conditions with the linear weighting restriction operator

$$(R^{\mathrm{c,D}})_h^H = \frac{1}{8}\begin{pmatrix} 2 & 3 & 1 & & & & & \\ & & 1 & 3 & 3 & 1 & & \\ & & & & 1 & 3 & 3 & 1 \\ & & & & & & 1 & 3 & 2 \end{pmatrix} \in \mathbb{R}^{4,8},$$

we have

$$f_H = \left(\tfrac{1}{2\sqrt{2}}, -\tfrac{1}{2\sqrt{2}}, \tfrac{1}{2\sqrt{2}}, -\tfrac{1}{2\sqrt{2}}\right)^\top.$$

The incorporation of more neighboring grid points clearly leads to a more smoothed coarse grid discretization. Moreover, different assumptions about the boundary change the obtained coarse grid discretization.

Prolongation Operators

The presented prolongation operators are based on linear interpolation, i.e. a linear polynomial can be interpolated exactly. A visualization for linear prolongation on a nodal grid and on a centered grid is given in Fig. 6.2.2. Here again for a centered discretization we distinguish between Neumann-0 and Dirichlet-0 boundary conditions.
The following examples illustrate the prolongation.

Example 6.2.6 (Prolongation Operator on a Nodal Grid)**.** We continue Ex. 6.2.4. Starting with the coarse grid discretization $f_H = 0 \in \mathbb{R}^3$ the application of the linear prolongation operator

$$(P^n)_H^h = \frac{1}{2}\begin{pmatrix} 2 & & \\ 1 & 1 & \\ & 2 & \\ & 1 & 1 \\ & & 2 \end{pmatrix} \in \mathbb{R}^{5,3}$$

leads to $f_h = 0 \in \mathbb{R}^5$. This shows, that the lost information cannot be recovered.

Example 6.2.7 (Prolongation Operators on a Centered Grid)**.** We continue Ex. 6.2.5. Given the coarse grid discretization

$$f_H = \left(\tfrac{1}{\sqrt{2}}, -\tfrac{1}{2\sqrt{2}}, \tfrac{1}{2\sqrt{2}}, -\tfrac{1}{\sqrt{2}}\right)^\top,$$

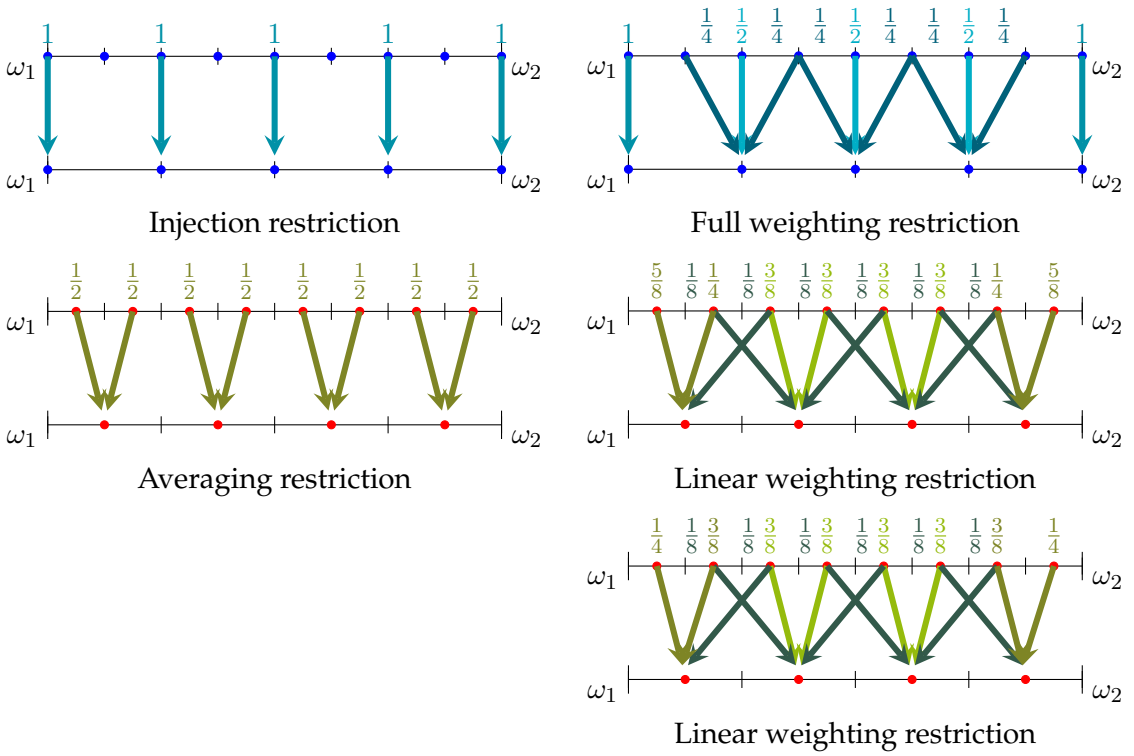

Figure 6.2.1: *Visualization of the restriction operator through their weights acting on the discretization on the original grid Ω_h leading to a discretization on the coarse grid Ω_H. Restriction operators acting on a nodal (top) or centered (middle, bottom) discretization. For a nodal discretization: Injection restriction (left) and full weighting (right). For a centered discretization: Averaging restriction (left) and linear weighting (right) with Neumann-0 (middle) or Dirichlet-0 (bottom) boundary conditions.*

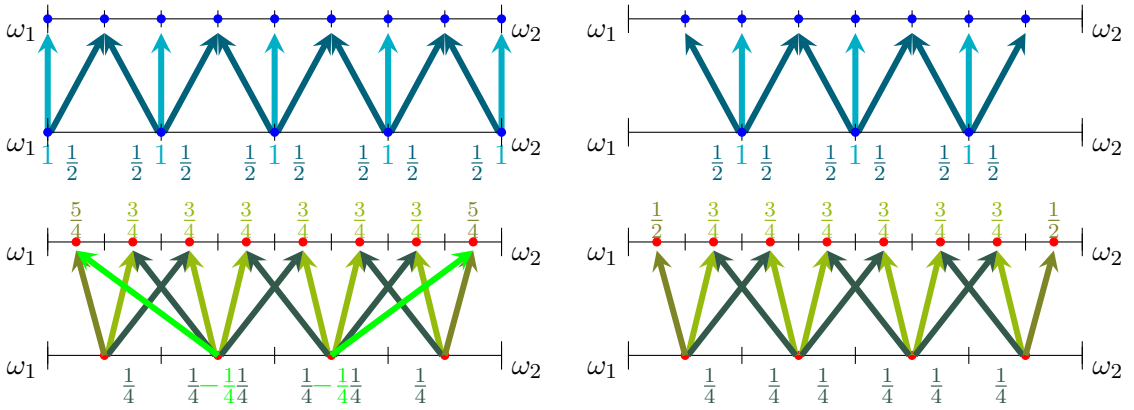

Figure 6.2.2: *Visualization of the linear prolongation operator through their weights acting on the discretization on the coarse grid Ω_H leading to a discretization on the original grid Ω_h. Prolongation operators acting on a discretization of a function with nodal (top left) inner-nodal (top right) or centered (bottom) grid type. For the centered discretization we distinguish between a Neumann-0 (left) or a Dirichlet-0 boundary (right).*

where the function f has Neumann-0 boundary conditions, we apply the linear prolongation operator with

$$(P^{c,N})_H^h = \frac{1}{4}\begin{pmatrix} 5 & -1 & & & & & & \\ 3 & 1 & & & & & & \\ & 1 & 3 & & & & & \\ & 3 & 1 & & & & & \\ & & & 1 & 3 & & & \\ & & & 3 & 1 & & & \\ & & & & & 1 & 3 & \\ & & & & & -1 & 5 \end{pmatrix} \in \mathbb{R}^{8,4}$$

and thus obtain

$$f_h = \frac{1}{8\sqrt{2}}\left(11, 5, 2, -2, 2, -2, -5, -11\right)^\top.$$

Starting with the coarse grid discretization

$$f_H = \left(\tfrac{1}{2\sqrt{2}}, -\tfrac{1}{2\sqrt{2}}, \tfrac{1}{2\sqrt{2}}, -\tfrac{1}{2\sqrt{2}}\right)^\top$$

and applying the linear prolongation operator with Dirichlet-0 boundary conditions

$$(P^{c,D})_H^h = \frac{1}{4}\begin{pmatrix} 2 & & & & & & & \\ 3 & 1 & & & & & & \\ & 1 & 3 & & & & & \\ & 3 & 1 & & & & & \\ & & & 1 & 3 & & & \\ & & & 3 & 1 & & & \\ & & & & & 1 & 3 & \\ & & & & & & 2 \end{pmatrix} \in \mathbb{R}^{8,4}$$

we obtain

$$f_h = \frac{1}{8\sqrt{2}}\left(2, 2, -2, 2, -2, 2, -2, -2\right)^\top.$$

Note, that the transposed prolongation operator is a restriction operator and vice versa. Furthermore, if it holds

$$P_H^h = 2^d (R_h^H)^\top,$$

for a prolongation P_H^h and a restriction R_h^H operator, we call them corresponding transfer operators. The rows of the transfer operators each sum up to 1. Since 2^d is the factor that describes the change in the number of cells between the coarse grid Ω_H and the original grid Ω_h, we need this factor in the equation. The linear prolongation operator corresponds to the full-weighting restriction operator on a nodal and to the linear weighting restriction operator on a centered grid.

6.2.3 Coarse Grid Operators

Another crucial step within the coarse grid correction, is to define a discretization of the differential operator $\mathcal{A}$ on the coarse grid Ω_H. The so called coarse grid operator A_H is used to solve the coarse grid solution. As the coarse grid problem, should represent the fine grid problem on a different discretization, the choice of the coarse grid operator is also a crucial component of multigrid.

A natural and very simple choice for the coarse grid operator A_H is to compute the discretization of the differential operator $\mathcal{A}$ on the coarse grid Ω_H. The main advantage of the direct computation is the simple implementation.

Another option is to use transfer operators P_H^h, R_h^H, see Def. 6.2.3, and to compute the so called Galerkin operator

$$A_H = R_h^H A_h P_H^h$$

instead. Note, that it is important to choose corresponding transfer operators for the Galerkin operator, as outlined in [123, pp. 84-91]. The advantage of the Galerkin operator is, that favorable properties of A_h such as symmetry and positive definiteness are transmitted automatically [31, p. 274]. The disadvantage is that the structure of the matrix is blended more and hence the potentially sparse structure is lost. This makes the Galerkin approach more time and memory consuming [31, p.278]. For more details about the properties of a Galerkin coarse grid operator, we refer to e.g. [31, §7.7].

Example 6.2.8 (Coarse Grid Operators)**.** In this example, we demonstrate two different coarse grid operators for the Poisson equation, given a coarse grid Ω_H on the domain $\Omega = (0,1)^2$ with grid size $M = (4,4)$ and an inner-nodal discretization. Direct calculation of the finite difference operator $A_H = I_2 \otimes (I_2 \otimes D_{11} + D_{22} \otimes I_1)$ leads to

$$I_2 \otimes D_{11} + D_{22} \otimes I_1 = \frac{1}{2^4} \begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix}.$$

In contrast the Galerkin based approach leads to a matrix $A_H = I_2 \otimes B$ with

$$B := \frac{1}{2^{10}} \begin{pmatrix} 12 & -2 & 0 & -2 & -1 & 0 & 0 & 0 & 0 \\ -2 & 12 & -2 & -1 & -2 & -1 & 0 & 0 & 0 \\ 0 & -2 & 12 & 0 & -1 & -2 & 0 & 0 & 0 \\ -2 & -1 & 0 & 12 & -2 & 0 & -2 & -1 & 0 \\ -1 & -2 & -1 & -2 & 12 & -2 & -1 & -2 & -1 \\ 0 & -1 & -2 & 0 & -2 & 12 & 0 & -1 & -2 \\ 0 & 0 & 0 & -2 & -1 & 0 & 12 & -2 & 0 \\ 0 & 0 & 0 & -1 & -2 & -1 & -2 & 12 & -2 \\ 0 & 0 & 0 & 0 & -1 & -2 & 0 & -2 & 12 \end{pmatrix}.$$

Here, we used a full-weighting restriction and a bilinear prolongation operator for the calculation of the Galerkin operator. The Galerkin approach leads to more averaging, compared to direct calculation. This has the advantage, that the coarse grid solution is smoother.

6.2.4 General Multigrid Architecture

The two-grid cycle is the simplest multigrid architecture. A general multigrid cycle can be deduced from its recursive calling. In this section, more details are provided about the two most common multigrid architectures: the V- and W-cycle. This section is based on the ideas of [31, §2.4].

Within a two-grid cycle the linear system

$$A_H e_H^k = r_H^k$$

needs to be solved for a given coarse grid Ω_H. A recursive call of another two-grid cycle is repeated until the coarsest grid is reached. The coarsest grid depends on factors such as the grid type of the discretization. A scalar equation or a very small linear system is obtained on the coarsest grid discretization. The cycle depth l coincides with the number of recursive calls. A recursive call of the two-grid cycle, to solve the linear system, leads to the the classical multigrid V-cycle. Another common option is a so called W-cycle, where the coarse grid correction is repeated twice. In Fig. 6.2.3 both architectures are visualized. Hence, the cycle index c, indicating the number of coarse grid corrections, is one for a V-cycle and two for a W-cycle.

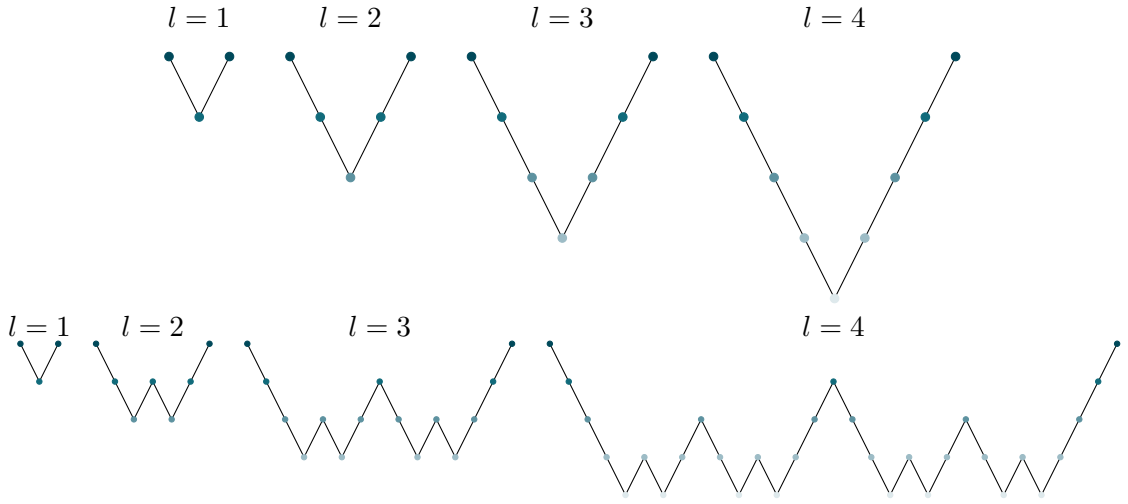

Figure 6.2.3: *Top: V-cycles with cycle depth $l = 1, 2, 3, 4$. Copied from [31, Fig. 2.8]. Bottom: W-cycles with cycle depth $l = 1, 2, 3, 4$. Copied from [31, Fig. 2.8].*

6.3 Local Fourier Analysis

The objective of local Fourier analysis (LFA), also called local mode analysis, is to determine the convergence behavior of a geometric multigrid algorithm in a simplified way. It is not completely rigorous, unless the Fourier modes are eigenvectors of the smoothing operator, see e.g. [125], which is not generally the case. Nevertheless, LFA is useful for the high frequency components of the error, as they are expected to resemble the eigenvectors of the smoothing operator [123, p.49]. Another strategy is to define another Fourier basis; see e.g. [44]. In this thesis, we consider the simplest standard setting and assume, that the Fourier modes resemble the eigenvectors of the smoothing operator.
An introduction to LFA methods in multigrid can be found in [57]. Note, that LFA is an own area of research in multigrid methods; see e.g. [125, 44]. We follow the ideas and definitions of LFA described in [31, §4.7, §8.3, §2.9]. After an introduction to the fundamental concepts, we analyze the smoother within the multigrid algorithm and assume an ideal coarse grid correction. This is also called Fourier one-grid analysis as only the operators of the original grid are considered. In particular, the h-ellipticity measure and the LFA version of the smoothing factor are presented.

6.3.1 Fundamental Concepts of Local Fourier Analysis

The term 'local' enters into play, as it is assumed, that smoothing is a local process, as each unknown is updated using information from nearby neighbors [123, p.49]. Hence, boundary phenomena are neglected and a so called infinite grid is considered, i.e. only inner grid points exist. This simplifies the analysis. Thereupon, the objective of LFA is to determine the quantitative convergence behavior and efficiency, that an appropriate multigrid algorithm can attain, if a proper boundary treatment is included.

For a given grid Ω_h, see Sec. 5.1, we define its infinite grid G_h, by extending its set of grid points X with grid points outside the domain Ω. Furthermore, we notice that an inner-nodal grid type cannot be distinguished from a nodal grid type, considering an infinite grid. The same holds for the inner-staggered grid type for $d = 2$ and inner-face- or inner-edge-staggered grid for $d = 3$. Moreover, the concept of coarse grids, see

Def. 6.1.4, can also be applied to infinite grids. An infinite coarse grid G_H is the infinite grid of the coarse grid Ω_H corresponding to the grid Ω_h. For the ease of presentation $h = h_1 = h_2 (= h_3)$ is assumed in LFA.

In LFA we consider the so called Fourier modes.

Definition 6.3.1 (Fourier Modes, modified Trottenberg [31, p. 297])**.** For an infinite grid G_h, a Fourier mode is the mapping

$$\phi_h : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d, \quad \phi_h(\theta, x) := 1^\top \cdot \mathrm{e}^{\frac{\mathrm{i}\theta^\top x}{h}},$$

where $1 = (1, \dots, 1) \in \mathbb{R}^d$, θ is a frequency in $\mathbb{R}^d$ and $x \in \mathbb{R}^d$ is a spatial component.

Due to the 2π-periodicity of the Fourier mode in terms of the frequency θ, we can restrict $\theta \in [-\pi, \pi)^d$. For a given infinite grid G_h, we consider the spatial component to be a grid point of the corresponding infinite coarse grid, i.e. $x \in G_H$. Then only the frequency components $\phi_h(\theta, x)$ with $\theta \in [\frac{-\pi}{2}, \frac{\pi}{2})^d$ are distinguishable, as $H = 2h$. Hence, we distinguish between the high and low frequency components of the Fourier mode $\phi_h(\theta, \cdot)$ on the grid G_h with respect to the coarse grid G_H.

Definition 6.3.2 (High and Low Frequency Components of a Fourier Mode, Trottenberg [31, Def. 4.2.1])**.** Considering a Fourier mode; see Def. 6.3.1. For a frequency $\theta \in T^{\text{low}} := \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^d$, the Fourier mode evaluated in $\phi_h(\theta, \cdot)$ is called a low frequency component of the Fourier mode ϕ_h. Otherwise, i.e. $\theta \in T^{\text{high}} := [-\pi, \pi)^d \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^d$, is a high frequency component of the Fourier mode ϕ_h.

The frequencies $\theta \in T^{\text{low}}$ are called low frequencies and the frequencies $\theta \in T^{\text{high}}$ are called high frequencies. A visualization of the low and high frequency domains for $d = 2, 3$ is given in Fig. 6.3.1.

Remark 6.3.3. We will use the substitution $\nu = 1 - \cos(\theta)$, hence it is useful to keep in mind, that $\nu \in [0, 2]^d$ for the frequency $\theta \in [-\pi, \pi]^d$. Thus, we adapt the definition and have: $\phi_h(\nu, \cdot)$ is a low frequency component of the Fourier mode ϕ_h, iff $\nu \in V^{\text{low}} := [0, 1)^d$ and a high frequency component of the Fourier mode ϕ_h, iff $\nu \in V^{\text{high}} := [0, 2]^d \setminus [0, 1)^d$.

Theorem 6.3.4 (Symbols/Formal Eigenvalues, Trottenberg [31, p. 297])**.** Let A_h from Eq. (22) be a finite difference operator, which can be described by a difference stencil; see Def. 6.1.1. Furthermore, X is a set of grid points for a given infinite grid G_h. A Fourier mode $\phi_h(\theta, x)$, see Def 6.3.1, with $\theta \in [-\pi, \pi)^d$ and $x \in X$ is a formal eigenfunction of the discrete operator A_h as

$$A_h \phi_h(\theta, x) = \underbrace{\begin{pmatrix} \tilde{A}^{1,1}(\theta) & \dots & \tilde{A}^{1,d}(\theta) \\ \vdots & & \vdots \\ \tilde{A}^{d,1}(\theta) & \dots & \tilde{A}^{d,d}(\theta) \end{pmatrix}}_{=:\tilde{A}(\theta) \in \mathbb{C}^{d,d}} \phi_h(\theta, x), \quad x \in X$$

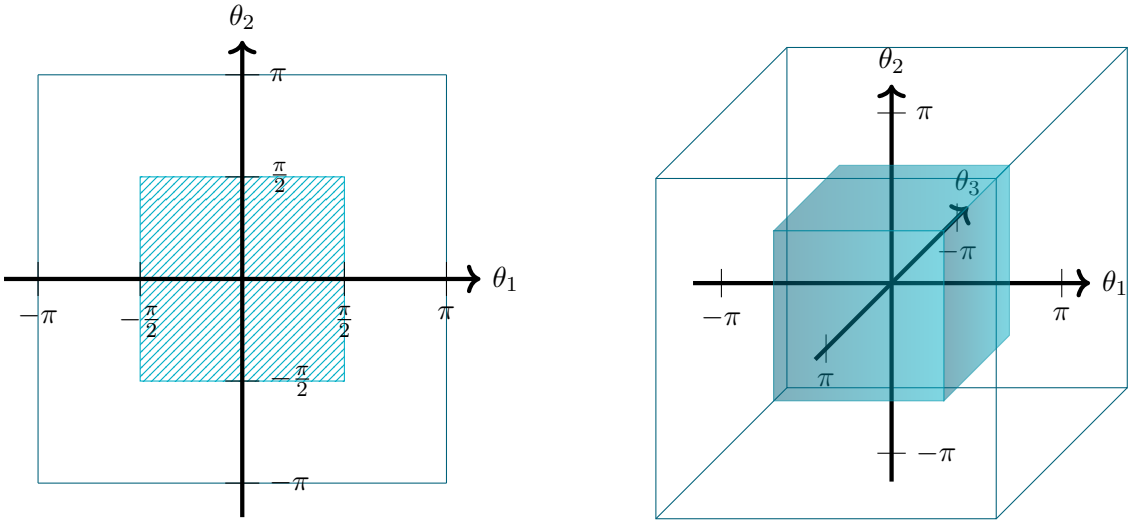

Figure 6.3.1: *Visualization of the low frequency domain* $T^{low} := \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^d$ *(blue) and the high frequency domain* $T^{high} := [-\pi, \pi)^d \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^d$ *(white) for* $d = 2$ *(left) and* $d = 3$ *(right).*

holds for $\tilde{A} : [-\pi, \pi]^d \to \mathbb{C}^{d,d}$ with the entries

$$\tilde{A}^{k,l}(\theta) = \sum_{\kappa} s_{\kappa}^{k,l} \mathrm{e}^{\mathrm{i}\theta^\top \kappa}.$$

We call the matrix $\tilde{A}(\theta)$ the formal eigenvalue or the symbol of the discretized differential operator A_h.

The main advantage of representing the finite difference operator by its symbol is the reduction in size. Instead of analyzing a matrix of the size $N \times N$, where N is the number of grid points and thus potentially large, we can analyze a $d \times d$ matrix for a given frequency θ.

Example 6.3.5 (Symbol)**.** We determine the symbol $\tilde{A}$ corresponding to the Laplace operator $A_h = -\Delta_h$; see [31, Ex. 4.2.1]. We use the five-point stencil for the diagonal entries and have

$$\begin{aligned}
\tilde{A}^{jj} &= \frac{1}{h^2}(4 - (\mathrm{e}^{\mathrm{i}\theta_1} + \mathrm{e}^{-\mathrm{i}\theta_1} + \mathrm{e}^{\mathrm{i}\theta_2} + \mathrm{e}^{-\mathrm{i}\theta_2})) \\
&= \frac{2}{h^2}(2 - (\cos(\theta_1) + \cos(\theta_2))) \\
&= \frac{2(\nu_1 + \nu_2)}{h^2}
\end{aligned}$$

with the substitution $\nu = 1 - \cos(\theta)$. Hence, the symbol is

$$\tilde{A}(\nu) = \frac{2}{h^2} \begin{pmatrix} (\nu_1 + \nu_2) & 0 \\ 0 & (\nu_1 + \nu_2) \end{pmatrix}.$$

6.3.2 h-Ellipticity Measure

The h-ellipticity of a discretized differential operator A_h is a necessary and sufficient condition for the existence of a point-wise smoothing procedure for a multigrid solver; see [31, Thm. 4.7.1]. If the h-ellipticity measure of a discretized differential operator A_h is bounded away from zero, there are no high frequency components in the kernel of the operator A_h, that cannot be reduced on a coarser grid. This implies, that there exists a smoothing procedure for a multigrid solver and furthermore, that the discretization is appropriate for a multigrid solver. The concept of h-ellipticity will be presented in more detail in the following, based on the concepts presented in [31, §4.7,§8.3].

Definition 6.3.6 (h-Ellipticity, Trottenberg [31, p. 300])**.** The h-ellipticity measure $E_h \in [0,1]$ of the discretized differential operator A_h from Eq. (22) is defined as

$$E_h(A_h) := \frac{\min\{|\det \tilde{A}(\theta)| : \theta \in T^{\text{high}}\}}{\max\{|\det \tilde{A}(\theta)| : -\pi < \theta_j \leq \pi\}},$$

where $\tilde{A}(\theta)$ represents the symbol of A_h, see Def. 6.3.4, and $\theta \in T^{\text{high}}$ are the high frequencies; see Def. 6.3.2.

An operator A_h is called h-elliptic or has a good measure of h-ellipticity, if there is a constant such that

$$E_h(A_h) \geq \text{const.} > 0.$$

The denominator is a scaling factor to ensure $E_h(A_h) \in [0,1]$.
The lack of h-ellipticity can be due to the fact that the discretized differential operator A_h is not elliptic itself or due to an instability of the discretization of A_h, i.e. due to e.g. large central differences; see [31, Ex. 4.7.2]. Hence, a smoothing and thus an efficient multigrid solver is not ensured.

Remark 6.3.7 (Trottenberg [31, Rem. 4.7.3]). If the h-ellipticity measure $E_h(A) > 0$, but is very small, it is not possible to construct an efficient pointwise smoother. In that case, small highly oscillating variations in the residuum f_h, i.e. the right hand-side of Eq. (22), may result in a large oscillations of the discrete solution u_h.

Example 6.3.8 (h-Ellipticity)**.** We determine the h-ellipticity measure for the Laplace operator $A_h = \Delta_h$; see [31, modified Ex. 4.7.1]. The determinant of the symbol reads as $\det \tilde{A}(\nu) = \frac{4}{h^4}(\nu_1 + \nu_2)^2$. Thus, the minimization of $\det \tilde{A}(\nu)$ with respect to $\nu \in [0,2]^2 \backslash [0,1)^2$ leads to the minimum $\det \tilde{A}((1,0)^\top) = \frac{4}{h^4}$. The maximization of $\det \tilde{A}(\nu)$ with respect to $\nu \in [0,2]^2$ leads to the maximum $\det(\tilde{A})((2,2)^\top) = \frac{64}{h^4}$. Hence, the h-ellipticity measure is

$$E_h(\Delta_h) = \frac{\frac{4}{h^4}}{\frac{64}{h^4}} = \frac{1}{16}.$$

6.3.3 Local Smoothing Factor

The local smoothing factor is the simplest quantitative predictor of the asymptotic convergence speed per multigrid cycle obtainable [33]. The convergence of a multigrid algorithm builds on the convergence of the corresponding smoothing procedure. Hence, the local smoothing factor is used to validate the performance of a multigrid solver.

Using the splitting from Eq. (24) the formal eigenvalues, see Def. 6.3.4, to the stencil notations of the operators $A_h^-, A_h^+ \in \mathbb{R}^{N,N}$ are used. Applying the operators to the formal eigenfunctions $\phi_h(\theta, x)$ it holds

$$A_h^- \phi_h(\theta, x) = \tilde{A}^- \phi_h(\theta, x),$$
$$A_h^+ \phi_h(\theta, x) = \tilde{A}^+ \phi_h(\theta, x),$$

where $\tilde{A}^-, \tilde{A}^+ \in \mathbb{C}^{d,d}$ are the symbols of the operators A_h^-, A_h^+, respectively.

Definition 6.3.9 (Local Smoothing Factor, Trottenberg [31, p. 298])**.** For the splitting from Eq. (24) with $\left(\tilde{A}^+(\theta)\right)^{-1}$ existing and eigenfunctions $\phi(\theta, \cdot)$ of the smoothing operator S_h, see Def. 6.2.1, we define the local smoothing factor

$$\mu_{\text{loc}} = \sup\{|\lambda(\theta)| : \det(\lambda(\theta)\tilde{A}^+(\theta) + \tilde{A}^-(\theta)) = 0, \theta \in T^{\text{high}}\}. \tag{26}$$

The local smoothing factor μ_{loc} depends on the parameter ω for the ω-damped Jacobi-type iteration. As $\mu_{\text{loc}}(S_h) < 1$ must be fulfilled for a converging smoothing procedure with smoothing operator S_h, a concrete choice for the parameter ω in the ω-damped Jacobi-type iteration can be deduced.

Example 6.3.10 (Local Smoothing Factor)**.** We determine the local smoothing factor for the Laplace operator $A_h = \Delta_h$ in a ω-damped Jacobi-type iteration; see [31, modified Ex. 4.3.5]. The symbols of the splitting matrices A_h^+ and A_h^- are

$$\tilde{A}^+ = \frac{4}{\omega h^2} I \quad \text{and} \quad \tilde{A}^-(\nu) = \frac{2}{h^2}(\nu_1 + \nu_2 - \frac{2}{\omega})I.$$

Hence, from

$$\lambda(\nu)\tilde{A}^+ + \tilde{A}^- = \frac{2}{h^2}(\frac{2}{\omega}(\lambda(\nu) - 1) + \nu_1 + \nu_2)I$$

we have to solve $\det(\lambda(\nu)\tilde{A}^+ + \tilde{A}^-) = 0$ and hence have

$$(\frac{2}{\omega}(\lambda(\nu) - 1) + \nu_1 + \nu_2)^2 = 0$$

with the root $\lambda^*(\nu) = 1 - \frac{(\nu_1+\nu_2)\omega}{2}$. Thus, the smoothing factor is

$$\mu_{\text{loc}} = \max\{|1 - \frac{(\nu_1 + \nu_2)\omega}{2}| : \nu \in [0, 2)^2 \backslash [0, 1)^2\} = \max\{|1 - \frac{\omega}{2}|, |1 - 2\omega|\}.$$

The same results can also be obtained in the rigorous smoothing analysis. This illustrates the reliability of LFA.

Furthermore, the local smoothing factor can be used as an approximation to the convergence speed. More precisely,

$$(\mu_{\text{loc}})^n \tag{27}$$

is assumed to be an approximation, where $n = n_1 + n_2$ is the number of all pre- and post-smoothing steps per multigrid cycle; see [33, p. 235].

7 A Multigrid Method for Image Registration with Second-Order Divergence-Curl Regularization

In this chapter, we present a novel multigrid method for the discrete image registration problem (20) with second-order divergence-curl (DC) regularization. Within the presented Gauss-Newton optimization, we solve the linear system (21) with a multigrid solver. To the best of our knowledge a multigrid solver for this specific regularization has not been introduced so far. We provide a local Fourier analysis (LFA), that provides a strong theoretical fundament for the proposed multigrid solver. Moreover, we present a comprehensive validation with numerical experiments of the proposed multigrid solver and compare the proposed multigrid solver to a state of the art multigrid solver.

Within the analysis, we focus on the regularizer of the image registration problem (1) alone, and ignore the influence of the data-fitting term. The focus on the regularizer is a common strategy for the analysis of multigrid solvers; see e.g. [46, 47, 48].
The considered linear system (21) can be interpreted as a discretized fourth-order partial differential equation (PDE), due to the second-order DC regularizer. Most existing fourth-order multigrid schemes, such as [76, 77, 78], rely on the splitting of the operator, as these lower-order schemes are more efficient. Here, we propose a multigrid solver for the fourth-order system. The aim is a proof of concept, i.e. to show the suitability of the coupled fourth-order operator for multigrid.

LFA ensures the suitability of the proposed discretization of the regularizer and the convergence of the smoother from a theoretical point of view. In particular, we determine the h-ellipticity measure, see Thm. 7.2.2 / Thm. 7.2.3, as well as the local smoothing factor, see Thm. 7.2.4 / Thm. 7.2.7 in a closed form for $d = 2$ and $d = 3$.
Usually, LFA is only provided for $d = 2$, see e.g. [55, 46], or in case of [47] only for $d = 3$. Our three-dimensional LFA results are not a straight-forward extension of the two-dimensional case, which emphasizes the importance of LFA for both cases, to get more insight into the underlying problem. In contrast to other contributions in this field, see e.g. [47, 77, 78, 55], we determine the local smoothing factor both explicitly and numerically.

A comprehensive validation of the proposed multigrid solver is a proof of concept. We provide numerous experiments for the two-dimensional setting to evaluate the convergence, the complexity as well as the efficiency of the proposed multigrid solver for different multigrid architectures.

In addition, a comparison to a state of the art multigrid solver gives an intuition for the findings of the proposed multigrid solver. In particular, we discuss the LFA results and compare them to the LFA of the coupled second-order elastic operator. The elastic regularizer [46] shares the following properties with the proposed second-order DC regularizer: It is physically inspired and thus enables a coupling of the components. Furthermore, the proposed discretization by [46] is based on a staggered-grid discretization. The main difference of the elastic and the second-order DC regularizer is their order. Whereas the elastic regularizer is of first order and thus the linear system can be interpreted as a second-order PDE within the multigrid framework, the second-order DC regularizer corresponds to a fourth-order PDE.
The previous numerical experiments are repeated with the elastic regularizer. Surprisingly, we observe that the proposed multigrid solver is almost as efficient as the elastic multigrid solver in the experiments.

This chapter is organized as follows: In Sec. 7.1, the general multigrid framework and in particular the multigrid components and the cycle architecture for the fourth-order linear system obtained from image registration with second-order DC regularization are presented. Moreover, LFA gives a theoretical foundation for the suitability of the proposed smoother in Sec. 7.2. Numerous experiments validate the proposed multigrid solver in Sec. 7.3. A comparison to the well known elastic multigrid solver is presented in Sec. 7.4.

7.1 The Proposed Multigrid Solver

In this chapter, we propose a multigrid solver for the linear system (21) of the Gauss-Newton iteration. We focus on the second-order divergence-curl (DC) regularizer and omit the data-fitting term and thus obtain a linear system of the form (22), where A_h is a discretization of the fourth-order differential operator from Tab. 3. For the ease of presentation, we focus on the inner-staggered grid discretization with Dirichlet-0 boundary conditions as proposed in Sec. 5.1.2. Staggered grid discretizations are known to avoid possible so called checkerboard instabilities in multigrid approaches [31, p. 314].

In the following we present concrete choices for the components of the proposed multigrid solver. Smoothing, restriction and prolongation as well as the coarse grid operator characterize a multigrid solver; see Sec. 6.2. Moreover, the cycle architecture of the multigrid framework is specified.

Smoothing

For the smoothing, an ω-damped Jacobi-type iteration, see Sec. 6.2.1, is used.
Cor. 7.2.5/ Cor. 7.2.8 give a criterion for a proper choice of the relaxation parameter ω in dependence of the parameter γ. We recall, that the parameter γ is weighting the divergence- and curl-part of the second-order DC regularization.
Usually for the number of smoothing steps $n_1 + n_2 \leq 3$ is a good compromise between convergence speed and complexity; see [31, §3.3.1] for a theoretical verification. Thus, we also ensure $n_1 + n_2 \leq 3$ for our experiments in Sec. 7.3. In particular, we test the impact of a different number of smoothing steps to the average defect reduction. The average defect reduction factor is a measure for the convergence of the multigrid solver. As the order of magnitude of the average defect reduction factor remains the same with a different number of smoothing steps, a small number of smoothing steps, e.g. $n_1 = n_2 = 1$, is preferred.

Restriction and Prolongation Operator

Due to the staggered grids, different directions of a component have different grid positions and the transfer operators are obtained through Kronecker products; see Tab. 11. The prolongation operators for both grid position, the inner-nodal and centered direction, are bi-/ trilinear interpolation; see Fig. 6.2.2.
For the restriction, the full-weighting restriction operator is used for the inner-nodal direction and the linear-weighting restriction operator is used for the centered direction; see Fig. 6.2.1. As known from [33, Eq. (4.3)], the transfer operator should be chosen in such a way, that the high frequencies are not magnified. For the discretized fourth-order finite difference operator A_h, the usage of the full /linear- weighting restriction and the bi/tri-linear prolongation are a borderline case, but are appropriate. Furthermore, at

Table 11: *Transfer operators for $d = 2, 3$. The transfer operators are restriction operators $T_i^c = (R^c)_h^H \in \mathbb{R}^{\frac{m_i}{2}, m_i}$ and $T_i^{in} = (R^{in})_h^H \in \mathbb{R}^{\frac{m_i}{2}-1, m_i-1}$ or prolongation operators $T_i^c = (P^c)_H^h \in \mathbb{R}^{m_i, \frac{m_i}{2}}$ and $T_i^{in} = (P^{in})_H^h \in \mathbb{R}^{m_i-1, \frac{m_i}{2}-1}$.*

$$
\begin{aligned}
d = 2: \quad T &:= \begin{pmatrix} T_2^c \otimes T_1^{in} & \\ & T_2^{in} \otimes T_1^c \end{pmatrix}, \\
d = 3: \quad T &:= \begin{pmatrix} T_3^c \otimes T_2^c \otimes T_1^{in} & & \\ & T_3^c \otimes T_2^{in} \otimes T_1^c & \\ & & T_3^{in} \otimes T_2^c \otimes T_1^c \end{pmatrix}
\end{aligned}
$$

least two cycles of the multigrid solver are performed to ensure [33, Eq. (4.4)]. Otherwise large errors might occur through the contribution of the high frequencies to the low frequencies. We refer to [33, §4.3] for a detailed presentation of the conditions for transfer operators.
Here, we assumed, that the discretization u_h of the desired displacement has Dirichlet-0 boundary conditions. Hence, our transfer operators are implemented with Dirichlet-0 boundary conditions. An extension to Neumann-0 or sliding boundary conditions is straightforward by using the corresponding transfer operators.

Coarse grid operator

The coarse grid operator, i.e. the discretized differential operator A_H, is computed directly for the second-order DC regularizer.
For the application of the multigrid solver to the whole image registration a strategy for the data-fitting term is needed. As the data-fitting term depends on the images, a direct computation would lead to a new interpolation of the images on the coarse grid. This makes a direct computation of the coarse grid operator computational expensive. Hence, for the data fitting term we follow the approach proposed in [46], which relies on a variation of the Galerkin based approach; see Sec. 6.2.3. More precisely, an approximation of the Hessian of the data-fitting term, in particular the diagonal, is determined for the discretization on the original grid. As the diagonal can be handled the same way as a vector, the restriction and prolongation operator can be applied directly to the diagonal. Hence, the coarse grid operator for the image registration problem has the form

$$A_H = R_h^H \text{diag}\left((\nabla^2 D)_h\right) P_H^h + \alpha (\nabla^2 S)_H,$$

where $(\nabla^2 D)_h$ is the Hessian of the data-fitting term for a discretization on the grid Ω_h. Further, $(\nabla^2 S)_H$ is the Hessian of the second-order DC regularizer discretized on the coarse grid Ω_H and R_h^H, P_H^h are the transfer operators, described in the last paragraph about the transfer operators.

Cycle Architecture

We consider both standard architectures, the V- as well as the W-cycle, in our experiments. We determine the average error reduction for both cycle architectures in Sec. 7.3. As the W-cycle does not lead to a better average defect reduction factor, the use of a V-cycle is recommended due to the lower complexity; see [31, Eq. (2.4.14)].

7.2 Local Fourier Analysis of the Proposed Multigrid Solver

A local Fourier analysis (LFA), see Sec. 6.3, of the second-order DC regularizer and in particular finite difference operator A_h from Sec. 5.1.2 is presented. We use LFA to ensure the suitability of the proposed inner-staggered grid discretization for a multigrid solver and to demonstrate that an ω-damped Jacobi-type iteration as a possible smoother. The LFA of second-order DC regularizer is a main result of this thesis and was first presented in our contribution [81] for a two-dimensional setting.

Our LFA of the second-order DC regularizer, for two- as well as the three-dimensional setting, includes:

1. the stencil notation of the corresponding discretized fourth-order differential operator from Sec. 5.1.2,
2. the h-ellipticity of the fourth-order operator and hence its suitability for multigrid methods,
3. the explicit local smoothing factor for an ω-damped Jacobi-type iteration and
4. an upper bound for the relaxation parameter ω of the ω-damped Jacobi-type iteration.

7.2.1 Stencil Notation

First, the stencil notation, see Def. 6.1.1 of the discretized fourth-order operator A_h, introduced in Sec. 5.1, is determined. Here, only the stencils corresponding to the inner grid points are relevant, as LFA assumes infinite grids.

Theorem 7.2.1. For $u \in C^6(\Omega, \mathbb{R}^d)$, with u bounded, we define an $\mathcal{O}(h^2)$ approximation $A_h \in \mathbb{R}^{d,d}$ of the differential operator $\mathcal{A}$ from Tab. 3 with entries $A_h^{j,k}, j, k = 1, \dots, d$, that are defined by the stencils listed in Tab. 12 for $d = 2$ and in Tab. 13 for $d = 3$.

Table 12: *Stencil notation of the entries $A_h^{k,l}, k, l = 1, 2$ of the fourth-order finite-difference operator A_h.*

$$
\begin{aligned}
A_h^{1,1} &:= \frac{1}{h^4} \begin{bmatrix} & \gamma & \\ 1 & -4\gamma-2 & 1 \\ (1-\gamma) \; -4(1-\gamma)-2 & 10 & -4(1-\gamma)-2 \; (1-\gamma) \\ 1 & -4\gamma-2 & 1 \\ & \gamma & \end{bmatrix}, \\
A_h^{2,2} &:= \left(A_h^{1,1}\right)^\top, \\
A_h^{1,2} &:= \frac{(2\gamma-1)}{h^4} \begin{bmatrix} & 1 & -1 & \\ 1 & -6 & 6 & -1 \\ -1 & 6 & -6 & 1 \\ & -1 & 1 & \end{bmatrix} = A_h^{2,1}.
\end{aligned}
$$

Proof. The proof follows a well known scheme, see e.g. [114, p. 5], and is exemplarily outlined for the stencil $A_h^{1,1}$ for $d = 2$. The remaining entries are along the same lines and the extension to the three-dimensional setting is straightforward.

Table 13: *Stencil notation of the entries $A_h^{k,l}, k,l = 1,2,3$ of the fourth-order discretized differential operator A_h, obtained from second-order DC regularization. Note, that the stencil representation changes in size for the different entries, in particular $A_h^{1,1}, A_h^{2,2}, A_h^{3,3} \in \mathbb{R}^{5,5,5}$ whereas $A_h^{1,2} = A_h^{2,1} \in \mathbb{R}^{4,4,3}$, $A_h^{1,3} = A_h^{3,1} \in \mathbb{R}^{4,3,4}$ and $A_h^{2,3} = A_h^{3,2} \in \mathbb{R}^{3,4,4}$.*

$$
\begin{aligned}
(A^{1,1})_{\kappa_1,\kappa_2,1} &:= \tfrac{1}{h^4}\begin{bmatrix} & & 0 & & \\ & & 0 & & \\ 0 & 0 & (1-\gamma) & 0 & 0 \\ & & 0 & & \\ & & 0 & & \end{bmatrix} = (A^{1,1})_{\kappa_1,\kappa_2,5} = (A^{2,2})_{\kappa_1,\kappa_2,1} = (A^{2,2})_{\kappa_1,\kappa_2,5},\\
(A^{1,1})_{\kappa_1,\kappa_2,2} &:= \tfrac{1}{h^4}\begin{bmatrix} & & 0 & & \\ 0 & 2(1-\gamma) & -8(1-\gamma)-2 & 2(1-\gamma) & 0 \\ & & 1 & & \\ & & 0 & & \end{bmatrix}\\
&= (A^{1,1})_{\kappa_1,\kappa_2,4} = (A^{2,2})_{\kappa_1,\kappa_2,2}^\top = (A^{2,2})_{\kappa_1,\kappa_2,4}^\top\\
(A^{1,1})_{\kappa_1,\kappa_2,3} &:= \tfrac{1}{h^4}\begin{bmatrix} & & \gamma & & \\ & 1 & -4\gamma-4 & 1 & \\ (1-\gamma) & -8(1-\gamma)-2 & 20(1-\gamma)+6\gamma+8 & -8(1-\gamma)-2 & (1-\gamma) \\ & 1 & -4\gamma-4 & 1 & \\ & & \gamma & & \end{bmatrix} = (A^{2,2})_{\kappa_1,\kappa_2,3}^\top,\\
(A^{3,3})_{\kappa_1,\kappa_2,1} &:= \tfrac{1}{h^4}\begin{bmatrix} & & 0 & & \\ & & 0 & & \\ 0 & 0 & \gamma & 0 & 0 \\ & & 0 & & \\ & & 0 & & \end{bmatrix} = (A^{3,3})_{\kappa_1,\kappa_2,5},\\
(A^{3,3})_{\kappa_1,\kappa_2,2} &:= \tfrac{1}{h^4}\begin{bmatrix} & & 0 & & \\ & & 1 & & \\ 0 & 1 & 4\gamma-4 & 1 & 0 \\ & & 1 & & \\ & & 0 & & \end{bmatrix} = (A^{3,3})_{\kappa_1,\kappa_2,4}\\
(A^{3,3})_{\kappa_1,\kappa_2,3} &:= \tfrac{1}{h^4}\begin{bmatrix} & & (1-\gamma) & & \\ & 2(1-\gamma) & -8(1-\gamma)-2 & 2(1-\gamma) & \\ (1-\gamma) & -8(1-\gamma)-2 & 20(1-\gamma)+6\gamma+8 & -8(1-\gamma) & (1-\gamma) \\ & 2(1-\gamma) & -8(1-\gamma)-2 & 2(1-\gamma) & \\ & & (1-\gamma) & & \end{bmatrix}\\
(A^{1,2})_{\kappa_1,\kappa_2,1} &:= \tfrac{(2\gamma-1)}{h^4}\begin{bmatrix} 0 & 0 & & & \\ 0 & 1 & -1 & 0 & \\ 0 & -1 & 1 & 0 & \\ & 0 & 0 & & \end{bmatrix} = (A^{1,2})_{\kappa_1,\kappa_2,3}\\
(A^{1,2})_{\kappa_1,\kappa_2,2} &:= \tfrac{(2\gamma-1)}{h^4}\begin{bmatrix} & 1 & -1 & \\ 1 & -8 & 8 & -1 \\ -1 & 8 & -8 & 1 \\ & -1 & 1 & \end{bmatrix},\\
(A^{1,3})_{\kappa_1,\kappa_2,1} &:= \tfrac{(2\gamma-1)}{h^4}\begin{bmatrix} & 0 & & \\ 0 & 1 & 0 & \\ 0 & -1 & 0 & \\ & 0 & & \end{bmatrix} = (A^{1,3})_{\kappa_1,\kappa_2,4} = (A^{2,3})_{\kappa_1,\kappa_2,1}^\top = (A^{2,3})_{\kappa_1,\kappa_2,4}^\top\\
(A^{1,3})_{\kappa_1,\kappa_2,2} &:= \tfrac{(2\gamma-1)}{h^4}\begin{bmatrix} & 1 & & \\ 1 & -8 & 1 & \\ -1 & 8 & -1 & \\ & -1 & & \end{bmatrix} = (A^{1,3})_{\kappa_1,\kappa_2,3} = (A^{2,3})_{\kappa_1,\kappa_2,2}^\top = (A^{2,3})_{\kappa_1,\kappa_2,3}^\top.
\end{aligned}
$$

Let $u = (u_1, u_2)^\top \in C^6(\Omega, \mathbb{R}^2)$ be a displacement, which is bounded in the neighborhood of $x = (x_1, x_2)^\top \in \Omega$. Taylor approximation [126, Eq. (5.81)] of a component of the deformation $u_j, j = 1,2$ leads to

$$
\begin{aligned}
u_j(x_1 + ah, x_2 + bh) = u_j &+ h(a\partial_1 + b\partial_2)u_j + \frac{h^2}{2}(a\partial_1 + b\partial_2)^2 u_j\\
&+ \frac{h^3}{3!}(a\partial_1 + b\partial_2)^3 u_j + \frac{h^4}{4!}(a\partial_1 + b\partial_2)^4 u_j\\
&+ \frac{h^5}{5!}(a\partial_1 + b\partial_2)^5 u_j + \mathcal{O}(h^6)
\end{aligned}
$$

for $a, b \in \{\frac{-3}{2}, -1, 0, \frac{-1}{2}, \frac{1}{2}, 1, \frac{3}{2}\}$.

Hence, for the entry $A_h^{1,1}$ acting on the first component of the deformation u_1 for the blue

Table 14: *Stencil $A_h^{1,1}$ for the two-dimensional setting, with color coded entries for the proof of Thm. 7.2.1.*

$$\frac{1}{h^4}\begin{bmatrix} (1-\gamma) & 1 & \gamma \\ & -4(1-\gamma)-2 & -4\gamma-2 \\ & 1 & 10 \\ & & -4\gamma-2 \\ & & \gamma \end{bmatrix} \begin{bmatrix} & 1 & \\ (1-\gamma) & -4(1-\gamma)-2 & 1 \\ & 1 & \end{bmatrix}$$

entries according to Tab. 14, we obtain

$$\begin{aligned}\ell^1 :&= (-4\gamma-2)(u_1(x_1+h,x_2)+u_1(x_1-h,x_2))+\gamma(u_1(x_1+2h,x_2)+u_1(x_1-2h,x_2))\\ &= (-6\gamma-4)u_1 - 2h^2\partial_{11}u_1 + (\gamma-\frac{1}{6})h^4\partial_{1111}u_1 + \mathcal{O}(h^6).\end{aligned}$$

Similar, for the orange entries of the stencil $A_h^{1,1}$, we have

$$\begin{aligned}\ell^2 :&= (-4(1-\gamma)-2)(u_1(x_1,x_2+h)+u_1(x_1,x_2-h))\\ &\quad + (1-\gamma)(u_1(x_1,x_2+2h)+u_1(x_1,x_2-2h))\\ &= (-6(1-\gamma)-4)u_1 - 2h^2\partial_{22}u_1 + ((1-\gamma)-\frac{1}{6})h^4\partial_{2222}u_1 + \mathcal{O}(h^6).\end{aligned}$$

For the green entries of the stencil, we have

$$\begin{aligned}\ell^3 :&= u_1(x_1+h,x_2+h)+u_1(x_1-h,x_2-h)+u_1(x_1+h,x_2-h)+u_1(x_1-h,x_2+h)\\ &= 4u_1 + 2h^2(\partial_{11}+\partial_{22})u_1 + \frac{h^4}{6}(\partial_{1111}+6\partial_{1122}+\partial_{2222})u_1 + \mathcal{O}(h^6).\end{aligned}$$

Putting it all together, with the central entry we get

$$\begin{aligned}A_h^{1,1}u_1 &= \frac{1}{h^4}(10u_1+\ell^1+\ell^2+\ell^3)\\ &= \frac{1}{h^4}(\gamma h^4\partial_{1111}+(1-\gamma)h^4\partial_{2222}+h^4\partial_{1122}+\mathcal{O}(h^6))u_1\\ &= (\mathcal{A})^{1,1}u_1+\mathcal{O}(h^2).\end{aligned}$$

The discretization of the other entries of $\mathcal{A}$ works in the same way. □

The non-diagonal entries $A_h^{k,l}$ for $k \neq l$ have no central entry and thus lead to a change of the grid of the discretization, which is due to the staggered grid discretization.

As shown above, the discretization has a second-order consistency, if restricted to the inner grid points. We already observed the $\mathcal{O}(h^2)$ consistency in the experiments in Sec. 5.1.2.

7.2.2 h-Ellipticity

A stable discretization of the differential operator $\mathcal{A}$ is needed to obtain an efficient multigrid algorithm [33]. As discussed in Sec. 6.3 a measure for the stability of the fourth-order difference operator A_h is the h-ellipticity measure E_h; see Def. 6.3.6. Hence, the h-ellipticity of the stencil notation from Thm. 7.2.1 is explicitly determined, to verify the discretization. In particular, we show, that the h-ellipticity measure is independent of the constant γ and bounded away from zero. Thus, the discretized differential operator A_h is h-elliptic for all choices of the parameter γ. In essence, the proposed staggered grid discretization for the second-order DC regularizer from Sec. 5.1.2 is appropriate for a multigrid framework and the existence of a pointwise smoother is ensured. This is one of the main results of this thesis.

h-Ellipticity in the Two-Dimensional Setting

We start with the two-dimensional setting and examine the h-ellipticity of the finite difference operator A_h, from Tab. 12.

Theorem 7.2.2. The h-ellipticity measure, see Def. 6.3.6, of the discretized operator A_h with stencil notation as in Thm. 7.2.1 is

$$E_h(A_h) = \frac{1}{256}.$$

Proof. First, we determine the symbol of the discretized fourth-order differential operator A_h according to Def. 6.3.4 for the stencil notation from Tab. 12. Starting with the entry $A_h^{1,1}$, the symbol is

$$\begin{aligned}
\tilde{A}^{1,1}(\theta) = \frac{1}{h^4}\Big(& 10 - (4\gamma+2)\left(\mathrm{e}^{-\mathrm{i}\theta_1} + \mathrm{e}^{\mathrm{i}\theta_1}\right) - (4(1-\gamma)+2)\left(\mathrm{e}^{-\mathrm{i}\theta_2} + \mathrm{e}^{\mathrm{i}\theta_2}\right) \\
& + \gamma\left(\mathrm{e}^{-\mathrm{i}2\theta_1} + \mathrm{e}^{\mathrm{i}2\theta_1}\right) + (1-\gamma)\left(\mathrm{e}^{-\mathrm{i}2\theta_2} + \mathrm{e}^{\mathrm{i}2\theta_2}\right) \\
& + \left(\mathrm{e}^{-\mathrm{i}(\theta_1+\theta_2)} + \mathrm{e}^{\mathrm{i}(\theta_1+\theta_2)}\right) + \left(\mathrm{e}^{-\mathrm{i}(\theta_1-\theta_2)} + \mathrm{e}^{\mathrm{i}(\theta_1-\theta_2)}\right)\Big) \\
= \frac{4}{h^4}\Big(& 2 + \gamma\cos(\theta_1)\big(\cos(\theta_1) - 2\big) - \cos(\theta_1) + \cos(\theta_1)\cos(\theta_2) \\
& + (1-\gamma)\cos(\theta_2)\big(\cos(\theta_2) - 2\big) - \cos(\theta_2)\Big).
\end{aligned}$$

With the substitution $\nu_j = 1 - \cos\theta_j = 2\sin^2\frac{\theta_j}{2}$ from Rem. 6.3.3, we obtain

$$\tilde{A}^{1,1}(\nu) = \frac{4}{h^4}(\gamma\nu_1^2 + \nu_1\nu_2 + (1-\gamma)\nu_2^2).$$

Similar, the symbol

$$\tilde{A}^{2,2}(\nu) = \frac{4}{h^4}((1-\gamma)\nu_1^2 + \nu_1\nu_2 + \gamma\nu_2^2)$$

of the entry $A_h^{2,2}$ is calculated. For the entry $A_h^{1,2}$ the symbol is

$$\begin{aligned}\tilde{A}^{1,2}(\theta) = \frac{(2\gamma-1)}{h^4}\Bigg(&6\left(\mathrm{e}^{\mathrm{i}\frac{1}{2}(\theta_1-\theta_2)}\mathrm{e}^{-\mathrm{i}\frac{1}{2}(\theta_1-\theta_2)}\right) - 6\left(\mathrm{e}^{\mathrm{i}\frac{1}{2}(\theta_1+\theta_2)} + \mathrm{e}^{-\mathrm{i}\frac{1}{2}(\theta_1+\theta_2)}\right)\\ &+ \left(\mathrm{e}^{\mathrm{i}\frac{1}{2}(3\theta_1+\theta_2)} + \mathrm{e}^{-\mathrm{i}\frac{1}{2}(3\theta_1+\theta_2)}\right) - \left(\mathrm{e}^{\mathrm{i}\frac{1}{2}(3\theta_1-\theta_2)} + \mathrm{e}^{-\mathrm{i}\frac{1}{2}(3\theta_1-\theta_2)}\right)\\ &+ \left(\mathrm{e}^{\mathrm{i}\frac{1}{2}(\theta_1+3\theta_2)} + \mathrm{e}^{-\mathrm{i}\frac{1}{2}(\theta_1+3\theta_2)}\right) - \left(\mathrm{e}^{\mathrm{i}\frac{1}{2}(\theta_1-3\theta_2)} + \mathrm{e}^{-\mathrm{i}\frac{1}{2}(\theta_1-3\theta_2)}\right)\Bigg)\\ = \frac{8(2\gamma-1)}{h^4}&\cdot \sin(\frac{\theta_1}{2})\sin(\frac{\theta_2}{2})\left(2\sin^2(\frac{\theta_1}{2}) + 2\sin^2(\frac{\theta_2}{2})\right).\end{aligned}$$

Again, the substitution $\nu_j = 1-\cos\theta_j = 2\sin^2\frac{\theta_j}{2}$ leads to

$$\tilde{A}^{1,2}(\nu) = \frac{4(2\gamma-1)}{h^4}\sqrt{\nu_1\nu_2}\,(\nu_1+\nu_2)\,.$$

Therefore, we end up with the symbol

$$\tilde{A}(\nu) = \begin{pmatrix}\tilde{A}^{1,1}(\nu) & \tilde{A}^{1,2}(\nu)\\ \tilde{A}^{2,1}(\nu) & \tilde{A}^{2,2}(\nu)\end{pmatrix} = \frac{4}{h^4}\begin{pmatrix}\gamma\nu_1^2+\nu_1\nu_2+(1-\gamma)\nu_2^2 & (2\gamma-1)\sqrt{\nu_1\nu_2}(\nu_1+\nu_2)\\ (2\gamma-1)\sqrt{\nu_1\nu_2}(\nu_1+\nu_2) & (1-\gamma)\nu_1^2+\nu_1\nu_2+\gamma\nu_2^2\end{pmatrix}.$$

For the determinant, we obtain

$$\det(\tilde{A})(\nu) = \frac{16}{h^8}\gamma(1-\gamma)(\nu_1+\nu_2)^4. \tag{28}$$

In order to determine the h-ellipticity measure according to Def. 6.3.6, the determinant $\det(\tilde{A})(\nu)$ from Eq. (28) is minimized with respect to $\nu \in [0,2]^2\backslash[0,1)^2$. Thus, the minimum is e.g. attained in $\nu = (1,0)^\top$ with

$$\det(\tilde{A})(1,0) = \frac{16}{h^8}\gamma(1-\gamma).$$

Next, we determine the maximum of the determinant with respect to $\nu \in [0,2]^2$. The maximal value

$$\frac{16}{h^8}\cdot 256\gamma(1-\gamma)$$

is attained in $\nu = (2,2)^\top$. Therefore, the h-ellipticity measure is

$$E_h(A_h) = \frac{\min\{|\det(\tilde{A})(\nu)| : \nu\in[0,2]^2\backslash[0,1)^2\}}{\max\{|\det(\tilde{A})(\nu)| : \nu\in[0,2]\}} = \frac{1}{256}.$$

□

As the h-ellipticity measure is bounded away from zero, the proposed staggered grid discretization from Tab. 12 is suitable for the construction of a multigrid solver and the existence of a pointwise smoother is ensured.

h-Ellipticity in the Three-Dimensional Setting

We extend the examination of the h-ellipticity of the finite difference operator A_h to the three-dimensional setting.

Theorem 7.2.3. The h-ellipticity measure, see Def. 6.3.6, of the discretized operator A_h with stencil notation as in Thm. 7.2.1 is

$$E_h(A_h) = \frac{1}{46656}.$$

Proof. Since the proof for $d = 3$ is carried out in the same way as in the two-dimensional setting, only a few intermediate steps are presented. Using the stencils from Tab. 13 we obtain the symbols

$$\begin{aligned}
\tilde{A}^{1,1}(\nu) &= \frac{4}{h^4}\left((1-\gamma)(\nu_2+\nu_3)^2 + \nu_1(\gamma\nu_1+\nu_2+\nu_3)\right),\\
\tilde{A}^{2,2}(\nu) &= \frac{4}{h^4}\left((1-\gamma)(\nu_1+\nu_3)^2 + \nu_2(\nu_1+\gamma\nu_2+\nu_3)\right),\\
\tilde{A}^{3,3}(\nu) &= \frac{4}{h^4}\left((1-\gamma)(\nu_1+\nu_2)^2 + \nu_3(\nu_1+\nu_2+\gamma\nu_3)\right),\\
\tilde{A}^{1,2}(\nu) &= \frac{4(2\gamma-1)}{h^4}\sqrt{\nu_1\nu_2}(\nu_1+\nu_2+\nu_3) = \tilde{A}^{2,1}(\nu),\\
\tilde{A}^{1,3}(\nu) &= \frac{4(2\gamma-1)}{h^4}\sqrt{\nu_1\nu_3}(\nu_1+\nu_2+\nu_3) = \tilde{A}^{3,1}(\nu) \quad \text{and}\\
\tilde{A}^{2,3}(\nu) &= \frac{4(2\gamma-1)}{h^4}\sqrt{\nu_2\nu_3}(\nu_1+\nu_2+\nu_3) = \tilde{A}^{3,2}(\nu).
\end{aligned}$$

For the calculation of the determinant, we used Mathematica [127], and obtain

$$\det(\tilde{A})(\nu) = \frac{2^6}{h^{12}}\gamma(1-\gamma)^2(\nu_1+\nu_2+\nu_3)^6.$$

Thus, the h-ellipticity measure of the discretized differential operator A_h for $d = 3$ is

$$E_h(A_h) = \frac{\frac{2^6}{h^{12}}\gamma(1-\gamma)^2}{\frac{2^6}{h^{12}}\gamma(1-\gamma)^2 6^6} = \frac{1}{6^6} = \frac{1}{46656}.$$

□

Note, that the determinant $\det(\tilde{A})$ weights the preliminary factor of the divergence and the curl-part equally in the two-dimensional case. In contrast, the preliminary factor of the curl-part, $1-\gamma$, has a quadratic impact, whereas the preliminary factor of the divergence-part, γ, only has a linear impact in the three-dimensional setting. This again emphasizes, that the three-dimensional results are not a straightforward extension of the two-dimensional findings, as the curl is fundamentally different.

The h-ellipticity measure is rather close to zero in the three-dimensional setting. This is mainly caused by the fourth-order of the finite difference operator. Nevertheless, this indicates, that there exist eigenvalues close to zero for high frequencies. From the spectral decomposition of the continuous fourth-order differential operator from Thm. 4.1.9

together with the knowledge of the discrete spectral decomposition of the elastic operator [107], we suspect, that the norm of a sine is close to zero. A spectral decomposition of the discrete fourth-order operator A_h is an urgent task for further research.

7.2.3 Local Smoothing Factor

Next, we determine the local smoothing factor μ_{loc}, see Def. 6.3.9, of the ω-damped Jacobi-type iteration explicitly, to show, that this iteration is a proper smoother for a suitable choice of ω. This is another main result of this thesis.

The ω-relaxed Jacobi-type update, see Eq. (25), is given via

$$u_h^{k+1} = u_h^k + \omega(z_h^{k+1} - u_h^k), \quad \text{where} \quad z_h^{k+1} = \left(I - \frac{h^4}{a(\gamma)} A_h\right) u_h^k + \frac{h^4}{a(\gamma)} f_h \tag{29}$$

with

$$a(\gamma) = \begin{cases} 10 & \text{for } d = 2, \\ 20(1-\gamma) + 6\gamma + 8 & \text{for } d = 3, \end{cases}$$

where A_h is the fourth-order discretized differential operator with the stencil notation as in Thm. 7.2.1. Note, that this simple notation is only possible due to the splitting with the diagonal matrix with the simple form $D_h = \frac{a(\gamma)}{h^4} I$. Thus, the smoothing operator, see Def. 6.2.1, has the form

$$S_h = I - \frac{\omega h^4}{a(\gamma)} A_h. \tag{30}$$

For this smoothing operator, we determine the smoothing factor.

The Local Smoothing Factor in a Two-Dimensional Setting

We start the examination of the local smoothing factor in a two-dimensional setting.

Theorem 7.2.4. The local smoothing factor μ_{loc}, see Def. 6.3.9; for the smoothing operator S_h from Eq. (30) is

$$\mu_{\text{loc}}(S_h) = \max\left\{ \left| 1 - \frac{16\omega}{5}(1 + \rho_\gamma) \right|, \left| 1 - \frac{\omega}{5}(1 - \rho_\gamma) \right| \right\}$$

with $\rho_\gamma := \sqrt{1 - 4\gamma(1-\gamma)}$ and $\gamma \in (0, 1)$.

Proof. The representation of the local smoothing factor from Eq. (26) is used. The splitting matrices, see Eq. (29), are

$$A_h^+ = \frac{a(\gamma)}{\omega h^4} I \quad \text{and} \quad A_h^- = A_h - \frac{a(\gamma)}{\omega h^4} I$$

where A_h is the discretization with stencil notation according to Thm. 7.2.1. The symbols, see Def. 6.3.4,

$$\tilde{A}^+ = \frac{a(\gamma)}{\omega h^4} I \quad \text{and} \quad \tilde{A}^- = \tilde{A} - \frac{a(\gamma)}{\omega h^4} I,$$

are determined, where $\tilde{A}$ is the symbol of A_h, that we already determined in the proof of Thm. 7.2.2.
The determinant $\det(\lambda(\nu)) := \det(\lambda(\nu)\tilde{A}^+ + \tilde{A}^-)$ is

$$\begin{aligned}\det(\lambda(\nu)) &= \det \tilde{A}(\nu) + \frac{40(\lambda(\nu)-1)}{\omega h^8}(\tilde{A}^{1,1}(\nu) + \tilde{A}^{2,2}(\nu)) + \frac{100(\lambda(\nu)-1)^2}{\omega^2 h^8} \\ &= \frac{16\gamma(1-\gamma)(\nu_1+\nu_2)^4}{h^8} + \frac{40(\lambda(\nu)-1)(\nu_1+\nu_2)^2}{\omega h^8} + \frac{100(\lambda(\nu)-1)^2}{\omega^2 h^8}\end{aligned}$$

with the roots $\lambda^*(\nu) = 1 - \frac{\omega}{5}(\nu_1+\nu_2)^2\left(1 \pm \sqrt{1-4\gamma(1-\gamma)}\right)$.
Hence, with $\rho_\gamma := \sqrt{1-4\gamma(1-\gamma)}$ the smoothing factor is

$$\begin{aligned}\mu_{loc}(S_h) &= \sup\{|\lambda^*(\nu)| : \nu \in [0,2]^2 \backslash (0,1)^2\} \\ &= \max\left\{\left|1 - \frac{16\omega}{5}(1+\rho_\gamma)\right|, \left|1 - \frac{\omega}{5}(1-\rho_\gamma)\right|\right\}.\end{aligned}$$

□

Note, that in contrast to the h-ellipticity measure, the local smoothing factor μ_{loc} of the ω-damped Jacobi-type relaxation depends on the parameter $\gamma \in (0,1)$.
For a given parameter γ and a given relaxation parameter ω we can deduce, if the ω-damped Jacobi-type iteration is a smoother. We need $\mu_{\text{loc}} < 1$ for the convergence of the iterative procedure, i.e. for a smoother. In particular, we observe, that the relaxation parameter ω needs to be chosen small enough, to obtain $\mu_{\text{loc}} < 1$. For an easier understanding we deduce an upper bound for the relaxation parameter ω in dependence of the parameter γ.

Corollary 7.2.5. According to Thm. 7.2.4, the ω-damped Jacobi-type relaxation is a suitable smoothing procedure, iff

$$\omega < \omega^*(\gamma) = \frac{5}{8(1+\sqrt{1-4\gamma(1-\gamma)})}. \tag{31}$$

The upper bound $\omega^*(\gamma)$ is visualized in Fig. 7.2.1 on the left side for the two-dimensional setting. If the relaxation parameter ω is chosen according to Cor. 7.2.5, then the ω-damped Jacobi-type iteration is a smoother for a multigrid algorithm solving the linear system of the form of Eq. (22) with the fourth-order discretized differential operator A_h.

We further provide a sanity check to emphasize the importance of choosing an appropriate relaxation parameter ω.

Experiment 7.2.6 (Suitable Choice of the Relaxation Parameter ω)**.** For $m = [128, 128]$ and $\gamma = 0.1$ the right-hand side is set to $f_h = 0$ and A_h is the fourth-order finite difference operator obtained from second-order DC regularization with a discretization as described in Sec. 5.1.2. The upper bound from Cor. 7.2.5 is $\omega^*(0.1) = \frac{25}{72}$. For $\omega = 0.3 < \omega^*(0.1)$ and $\omega = 0.5 > \omega^*(0.1)$ the relative residuum $r = \frac{\|f_h - A_h u_h\|}{\|f_h\|}$ is plotted against the number of iterations in Fig. 7.2.2. For the suitable relaxation parameter $\omega = 0.3$ the residuum decreases. This enhances, that the ω-damped Jacobi-type iteration is an appropriate smoother. In contrast, the residuum increases for the non-suitable choice $\omega = 0.5$.

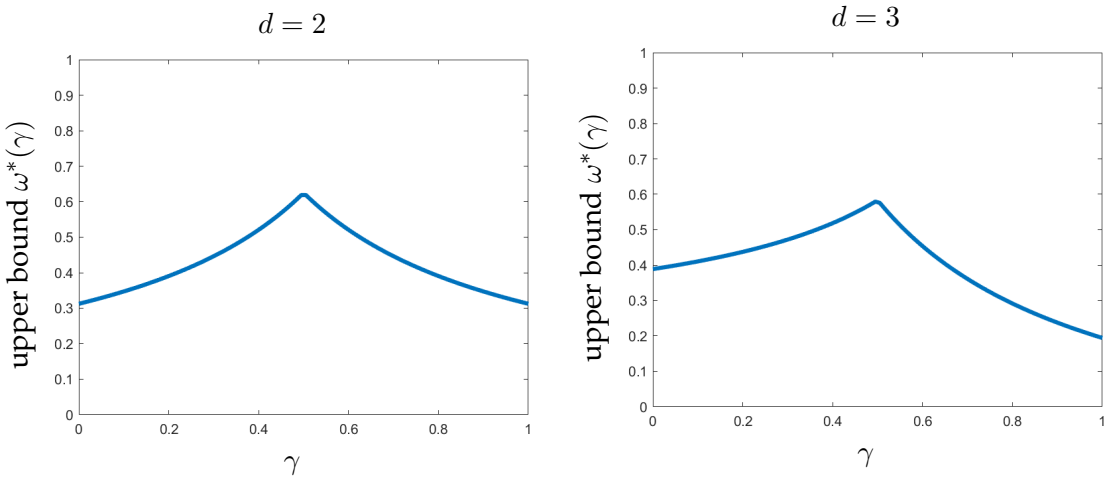

Figure 7.2.1: *Visualization of the upper bound $\omega^*(\gamma)$ according to Eq. (31)/ Eq. (32) for a proper choice of the parameter ω in dependence of the parameter $\gamma \in (0,1)$ for $d = 2$ (left) and $d = 3$ (right).*

The Local Smoothing Factor in a Three-Dimensional Setting

We extend the findings for the local smoothing factor to the three-dimensional setting.

Theorem 7.2.7. The local smoothing factor μ_{loc}, see Def. 6.3.9, for the smoothing operator S_h from Eq. (30) is

$$\mu_{\text{loc}}(S_h) = \max\left\{\left|1 - \frac{72\omega \max\{\gamma, 1-\gamma\}}{7(2-\gamma)}\right|, \left|1 - \frac{2\omega \min\{\gamma, 1-\gamma\}}{7(2-\gamma)}\right|\right\}$$

with $\gamma \in (0,1)$.

Proof. Analogously to the two-dimensional setting, we use the representation of the local smoothing factor from Eq. (26). Here, we compute the roots of the determinant $\det(\lambda(\nu))$ with Mathematica [127] and obtain

$$\begin{aligned}
\lambda_1^*(\nu) &= 1 - \frac{6\omega \max\{\gamma, 1-\gamma\}}{21(2-\gamma)}(\nu_1+\nu_2+\nu_3)^2,\\
\lambda_2^*(\nu) &= 1 - \frac{6\omega \min\{\gamma, 1-\gamma\}}{21(2-\gamma)}(\nu_1+\nu_2+\nu_3)^2,\\
\lambda_3^*(\nu) &= 1 - \frac{6\omega(1-\gamma)}{21(2-\gamma)}(\nu_1+\nu_2+\nu_3)^2.
\end{aligned}$$

Thus, the smoothing factor is

$$\begin{aligned}
\mu_{loc}(S_h) &= \sup\{|\lambda_j^*(\nu)| : \nu \in [0,2]^3 \backslash (0,1)^3, j = 1,2,3\}\\
&= \max\left\{\left|1 - \frac{72\omega \max\{\gamma, 1-\gamma\}}{7(2-\gamma)}\right|, \left|1 - \frac{2\omega \min\{\gamma, 1-\gamma\}}{7(2-\gamma)}\right|\right\}.
\end{aligned}$$

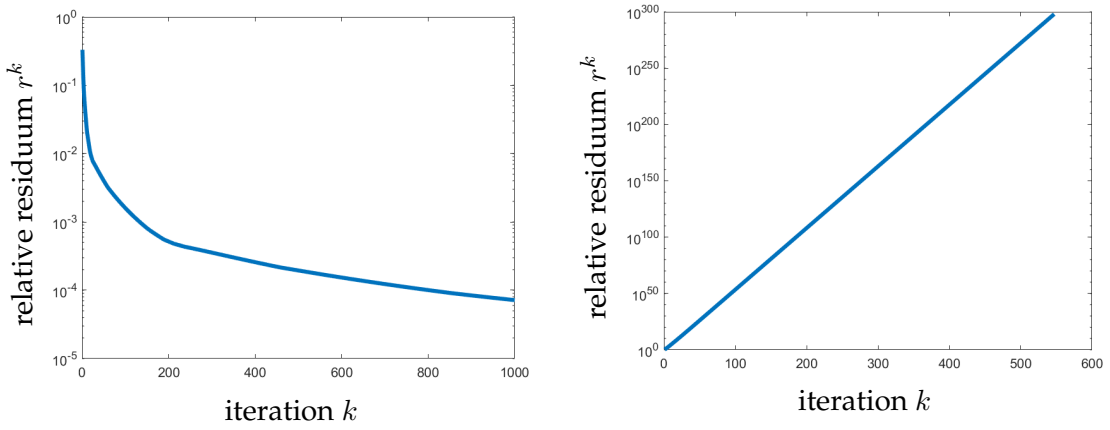

Figure 7.2.2: *Visualization of the relative residuum against the number of iterations for an ω-damped Jacobi-type iterations as described in Exp. 7.2.6. The relative residuum decreases for $\omega = 0.3 < \omega^*$ (left) and rises for $\omega = 0.5 > \omega^*$ (right). Hence, for $\omega = 0.3$ the ω-relaxed Jacobi-type iteration is a smoothing procedure, but not for $\omega = 0.5$.*

□

We determined the local smoothing factor explicitly for the three-dimensional setting. From this, we again deduce the upper bound for the relaxation parameter ω of the ω-damped Jacobi-type iteration.

Corollary 7.2.8. According to Thm. 7.2.4, the ω-damped Jacobi-type relaxation is a suitable smoothing procedure, iff

$$\omega < \omega^*(\gamma) = \frac{7(2-\gamma)}{36\max\{\gamma, 1-\gamma\}}. \tag{32}$$

The upper bound for the three-dimensional setting is visualized on the right side of Fig. 7.2.1.

Table 15: *Approximation of the asymptotic convergence speed with the local smoothing factor μ_{loc}^n and n overall smoothing steps with the damping parameter $\omega = 0.3$ and the DC regularizer parameter $\gamma = 0.1$ for $d = 2, 3$.*

$n =$	1	2	3
$d = 2$	0.9880	0.9604	0.9412
$d = 3$	0.9955	0.9909	0.9865

Furthermore, we know from (27), that the local smoothing factor μ_{loc} can be used to determine an approximation to the convergence speed of the multigrid algorithm.

Example 7.2.9 (Theoretical Approximation of the Convergence Speed)**.** Let A_h be the fourth-order finite difference operator from Sec. 5.1 with $\gamma = 0.1$. The local smoothing factor for an ω-damped Jacobi-type relaxation with relaxation parameter $\omega = 0.3$ is $\mu_{\text{loc}} = \frac{247}{250} = 0.988$ for $d = 2$; see Thm. 7.2.4. For the three-dimensional setting we obtain and

$\mu_{\text{loc}} = \frac{662}{655} = 0.9955$; see Thm. 7.2.7. Hence, for n_1 pre- and n_2 post smoothing steps, i.e. $n = n_1 + n_2$ overall smoothing steps with $n = 1, 2, 3$, we obtain an approximation to the convergence speed as listed in Tab. 15. As the smoothing factors are close to one, a fast converging, i.e. an efficient multigrid solver is not expected.

7.3 Validation of the Proposed Multigrid Solver

In this section, we validate the proposed multigrid solver for image registration with second-order divergence-curl (DC) regularization. Specifically, we present numerical experiments to validate the convergence, complexity and efficiency of the proposed multigrid solver.
In the first place, the following numerical experiments verify the suitability of the proposed multigrid solver for the fourth-order PDE obtained from image registration with the proposed second-order DC regularizer. Furthermore, a different number of smoothing steps and different architectures are compared against each other. For the ease of presentation, we limit the experiments to $d = 2$.

Implementation Details

Our implementation is in Matlab [118]. It builds on the FAIR package [5], that provides a multigrid solver implementation for image registration with elastic and curvature regularization. The FAIR multigrid framework is based on matrix-free concepts; see [5, 120]. Our implementation of the multigrid solver for the fourth-order PDE obtained from second-order DC regularization fits into the FAIR framework and thus is also based on matrix-free implementations.

Validation of the Convergence

Next to the smoothing factor, the average defect reduction factor is an approximation of the convergence speed. It is defined as follows.

Definition 7.3.1 (Average Defect Reduction Factors, Trottenberg [31, Eq. (2.5.1)])**.** For a multigrid algorithm, the average defect reduction factor over $k \in \mathbb{N}$ iterates is determined by

$$q^k := \sqrt[k]{\frac{\|r_h^k\|}{\|r_h^0\|}},$$

where $r_h^0 \neq 0$ is the initial residual and r_h^k the residual at the k-th iteration.

For k large enough, the average defect reduction factor q^k is an approximation of the convergence speed; see [31, p. 54]. Note, that the first few iterations might not reflect the asymptotic convergence behavior, hence there exist other definitions of the average defect reduction factor, as e.g. the stabilized average defect reduction factor

$$\hat{q}^{(k_0,k)} := \sqrt[k-k_0]{\frac{\|r_h^k\|}{\|r_h^{k_0}\|}} \tag{33}$$

for a small number $k_0 < k$.

Furthermore, it is well known that the convergence of a multigrid solver does not depend on the grid size [31, p. 55]. This property is called h-independent convergence. Thus, for different grid sizes, the average defect reduction factors should be in a similar range.

Table 16: *Average defect reduction factor $q^{(20)}$ and stabilized average defect reduction factor $\hat{q}^{(5,20)}$ for grid sizes $m^j = 2^j(4,4), j = 0, \cdots 7$; see Exp. 7.3.2.*

$j =$	0	1	2	3	4	5	6	7
$q^{(20)}$	0.8404	0.8378	0.8376	0.8372	0.8371	0.8367	0.8370	0.8365
$\hat{q}^{(5,20)}$	0.8844	0.8756	0.8751	0.8740	0.8743	0.8731	0.8733	0.8737

In a first step, h-independence of the proposed multigrid algorithm is demonstrated. In particular, we ensure, that the average defect reduction factors from Def. 7.3.1 and Eq. (33) are in the same order of magnitude for different grid sizes.

Experiment 7.3.2 (h-Independence)**.** As suggested in [31, p. 54], we set the initial guess u_h^0 to a random vector and the right-hand side from Eq. (22) to $f_h = 0$. We apply an V(1,1) cycle and determine the average defect reduction factor q and the stabilized average defect reduction factor $\hat{q}$ for different grid sizes $m^j = 2^j(4,4), j = 0, 1, \cdots 7$ on a domain $\Omega = [0,1]^2$. Due to the inner-staggered discretization, the number of unknowns is $2^{2j+4} - 2^{j+2}$. The grid width is $h^j = \frac{1}{2^{j+2}}$. The quantities are determined after $k = 20$ iterations and for the stabilized factor with $k_0 = 5$. The experiments were repeated 100 times. The averaged values are summarized in Tab. 16. The magnitude of both average defect reduction factors stay the same for all grid sizes m^j. This illustrates the h-independence of the proposed multigrid algorithm.

Table 17: *Average defect reduction $q^{(20)}$ and stabilized average defect reduction $\hat{q}^{(20)}$ of a $V(n_1, n_2)$- or $W(n_1, n_2)$-cycle with n_1 pre- and n_2 post-smoothing steps. All quantities are measured 100 times and averaged; see Exp. 7.3.3. The smallest, i.e. best approximated convergence rates are highlighted in blue.*

	$(n_1, n_2) =$	(1,0)	(2,0)	(1,1)	(2,1)	(1,2)	(3,0)
	$(\mu_{\text{loc}})^{n_1+n_2}$	0.9800	0.9604	0.9604	0.9412	0.9412	0.9412
V-cycle	$q^{(20)}$	0.8602	0.8378	0.8375	0.8279	0.8277	0.8280
	$\hat{q}^{(5,20)}$	0.8829	0.8748	0.8746	0.8791	0.8789	0.8781
W-cycle	$q^{(20)}$	0.8561	0.8388	0.8398	0.8310	0.8322	0.8303
	$\hat{q}^{(5,20)}$	0.8990	0.8908	0.8920	0.8821	0.8832	0.8810

Next, the average defect reduction factors are compared for different numbers of pre- and postsmoothing steps. Both quantities are a practical approximate of the convergence speed, whereas the approximation with the local smoothing factor from Ex. 7.2.9 is a theoretical approximation. We compare the theoretical to the practical approximation.
Furthermore, a V-cycle architecture is compared against a W-cycle architecture. Note, that the average defect reduction factors are used as a criterion for convergence speed here. Neither a concrete number of pre- and postsmoothing steps, nor the W-cycle leads to a significant better convergence speed. Hence, in terms of complexity, a V-cycle with a small number of smoothing steps is recommended.

Experiment 7.3.3 (Convergence for V- and W-Cycles)**.** We extend Exp. 7.3.2 and consider a different number of pre- and post-smoothing steps for a V- and a W-cycle. The grid size is $m = (128, 128)$ and the average defect reduction $\hat{q}^{(20)}$ from Def. 7.3.1 and the stabilized average defect reduction $\hat{q}^{(5,20)}$ from Eq. (33) with $k_0 = 5, k = 20$ are determined. The quantities are measured for 100 times, then averaged and are summarized in Tab. 17. In general, we observe, that the practical approximations of the convergence speed are better, than the theoretical approximations through the smoothing factor. The results for the V- and W-cycle have the same order of magnitude. This suggests the application of a V-cycle, as it is less computational expensive, but leads to the same convergence speed.

Note, that the received convergence rates are a proof of concept of the proposed multigrid solver. Nevertheless, they are not desirable for real-time applications, as a smaller convergence factor is needed for an efficient multigrid framework. Still, the average defect reduction factors are bounded away from one and hence suggest a converging multigrid algorithm.

Validation of the Complexity

We demonstrate that the complexity of the proposed multigrid solver is linear in the number of unknowns. The linear complexity is expected for V- as well as W-cycles; see [31, §2.4.3] for details. This complexity is the reason why multigrid methods are asymptotically optimal methods.
We determine the number of floating point operations (flops) to verify the complexity of the proposed multigrid solver with different architectures. Note, that the number of flops does not depend on the discretized differential operator A_h.

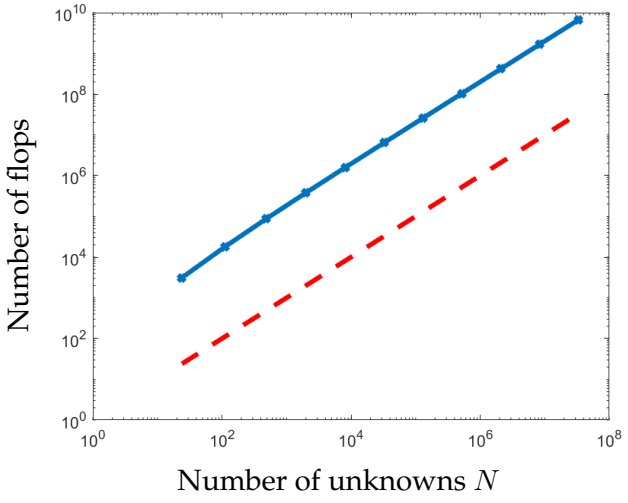

Figure 7.3.1: *Visualization of linear complexity; see Exp. 7.3.4 for details. Log-log plot of the number of flops against the number of unknowns. The number of flops of a single V-cycle (in blue) rises with a slope of one (in red, dotted).*

Experiment 7.3.4 (Complexity)**.** To determine the number of flops, we use the lightspeed package [128] and our matrix-based implementation of our proposed multigrid solver. For different grid sizes $m^j = 2^j(1, 1),\ j = 2, 3 \ldots, 12$, i.e. $N^j = 2^{2j+1} - 2^{j+1}$ unknowns,

we determine the number of flops for a single V-cycle. The number of flops rises linear with the number of unknowns; see Fig. 7.3.1. This emphasizes the complexity of $\mathcal{O}(N)$ of the multigrid algorithm.

Validation of the Efficiency

The efficiency describes the trade-off between complexity and convergence speed, as a solution should be obtained with as little computation as possible but at the same time as close to the exact solution as possible.
Therefore, we measure the runtime and compare the proposed multigrid solver against a conjugate gradient solver. Conjugate gradient solvers are one of the most commonly used solution strategies in image registration and are further known to be competitive to multigrid solver; see [32, p. 105].

In order to determine the error as a criterion, we change the experimental setup, as a ground truth solution is needed. We reuse the experimental setup from the consistency checks in Sec. 5.1.2 as they provide a ground truth solution.

Table 18: *Comparison of a multigrid (MG) and a conjugate gradient (CG) solver for the coupled fourth-order problem from Exp. 7.3.5. For a given tolerance, i.e. a given upper bound for the reduction of the relative residuum, the runtime (t in seconds), number of iterations (k), the relative residuum after k iterations and the norm of the error are determined. The quantities are measured 80 times and averaged.*

	tol $= 10^{-1}$				tol $= 10^{-2}$			
	t [s]	k	$\frac{\|r^k\|_2}{\|r^0\|_2}$	$\|e^k\|_2$	t [s]	k	$\frac{\|r^k\|_2}{\|r^0\|_2}$	$\|e^k\|_2$
MG	0.07	1	$1.43 \cdot 10^{-2}$	63.28	596.58	11138	$1 \cdot 10^{-2}$	$7.02 \cdot 10^{3}$
CG	18.59	5139	$9.96 \cdot 10^{-2}$	$2.90 \cdot 10^{4}$	44.69	12716	$9.8 \cdot 10^{-3}$	$3.00 \cdot 10^{4}$

Experiment 7.3.5 (Efficiency of Multigrid and Conjugate Gradient Solver)**.** The proposed multigrid solver is compared to the build-in Matlab [118] conjugate gradient solver. We set the experimental setup to $\Omega = [0, 2\pi]^2$ and

$$u_h^* = \begin{pmatrix} \sin(x_1)\sin(2x_2) \\ 0 \end{pmatrix}, \quad f_h = \begin{pmatrix} (20 - 15\gamma)\sin(x_1)\sin(2x_2) \\ 10(1 - 2\gamma)\cos(x_1)\cos(2x_2) \end{pmatrix}.$$

From the ground truth solution u_h^* we determine the error $e^k = u^* - u^k$ for the k-th iteration; see Def. 6.1.3. The grid size is set to $m = (128, 128)$ and a $V(1,1)$-cycle is used for the multigrid solver. The tolerance for the relative residual reduction of the solvers is set to $\text{tol}_1 = 10^{-1}$ in a first experiment and to $\text{tol}_2 = 10^{-2}$ in a second experiment. The number of iterations as well as the runtime is measured for both solvers. Additionally, we determine the actual relative residuum $\frac{\|r^k\|_2}{\|r^0\|_2}$. The runtime, as well as the number of iterations, the actual obtained relative residuum and the norm of the error are averaged over 80 measurements. Tab. 18 lists the averaged results. The increasing norm of the error for the smaller tolerance might be surprising, at first sight. But as the algorithms reduces the relative residuum and do not know the actual error, the norm of the error is not guaranteed to be monotonically decreasing for a converging algorithm. We observe, that the norm of the error is smaller for the multigrid solver in this experiment. Furthermore, we observe, that due to the poor convergence rate a large number of iterations is

needed for the multigrid solver, which also leads to a rather high runtime. Even though the number of iterations is smaller for the multigrid solver. In terms of the runtime the use of the conjugate gradient is clearly preferable for a large number of iterations. Note, that the build-in conjugate gradient solver of Matlab is optimized and hence highly efficient.

This experiment again stresses that the proposed multigrid solver is a proof of concept, but not very efficient. Due to the poor convergence speed the multigrid solver is not recommended for a high number of iterations as the expected runtime is very large.

7.4 Comparison of the Proposed Multigrid Solver with the Elastic Multigrid Solver

We compare the proposed multigrid solver to the state of the art elastic multigrid solver as proposed in [46], to get an intuition about the obtained results. Furthermore, we build on the implementation of the multigrid solver for the elastic regularizer in FAIR [5], that is based on a staggered grid discretization. We compare both the theoretical results of the local Fourier analysis (LFA) from Sec. 7.2 to a LFA of the elastic operator [46] and further the results of the numerical experiments to those of the elastic operator.

7.4.1 Comparison of the Results of Local Fourier Analysis

In the following we take a closer look at different aspects of LFA of the elastic operator. For the stencil notation of the staggered-grid discretization of the elastic operator A_h^{elas} we refer to [46]. We present the h-ellipticity and state, that the ω-damped Jacobi-type iteration is a smoother for a suitable choice of the Lamé constants λ, μ. For the ease of presentation and as the proofs follow the same scheme as in Sec. 7.2, we omit them in this section.

Comparison of the h-Ellipticity

First, we compare the h-ellipticity measure from Thm. 7.2.2 / Thm. 7.2.3. The h-ellipticity measure was already determined for the two-dimensional elastic finite difference operator A_h^{elas} in [46]. We formulate it with a straight-forward extension for the three-dimensional setting.

Theorem 7.4.1. The h-ellipticity measure, see Def. 6.3.6, of the discretized operator A_h^{elas} is

$$E_h(A_h^{\text{elas}}) = \begin{cases} \frac{1}{16}, & \text{for } d = 2, \\ \frac{1}{216}, & \text{for } d = 3. \end{cases}$$

Note, that the result of the h-ellipticity measure $E_h(A_h^{\text{elas}})$ is not surprising for the two-dimensional case, as it goes in line with the result of Ex. 6.3.8. This is expected as the elastic operator A_h^{elas} simplifies to the Laplace operator for the unnatural choice $\lambda = -\mu$ of the Lamé constants.

We compare the h-ellipticity measure $E_h(A_h^{\text{elas}})$ from Thm. 7.4.1 with the h-ellipticity measure $E_h(A_h)$ as stated in Thm. 7.2.2/ Thm. 7.2.3.

Clearly, the proposed finite difference operator A_h, as well as the elastic finite difference operator A_h^{elas} are h-elliptic and thus suitable discretizations for a multigrid framework. Moreover, the h-ellipticity measure E_h is independent of the constant γ in case of the operator A_h, but also independent of the Lamé constants in case of the elastic operator A_h^{elas}.
However, for the three-dimensional setting we observe, that the h-ellipticity measure E_h is much smaller for the operator A_h, than for the elastic operator A_h^{elas}. This is mainly because A_h is a discretization of a fourth-order differential operator, whereas A_h^{elas} is a discretized second-order differential operator. An h-ellipticity measure close to zero, might indicate that the proposed discretization is not suitable for smoothing, i.e. for a reliable multigrid solver. This suggests, that the elastic multigrid solver is more efficient than the proposed multigrid solver. A splitting of the fourth-order operator A_h could improve the efficiency of a corresponding multigrid scheme.

Comparison of the Local Smoothing Factor

We further compare the local smoothing factor μ_{loc} for an ω-damped Jacobi-type iteration. Note, that the local smoothing factor is not stated in [46] for the elastic operator A_h^{elas} and hence we present new result. As the proof follows the same concept as in Thm. 7.2.4 / Thm. 7.2.7, we omit it here.

The smoothing operator, see Def. 6.2.1, for an ω-damped Jacobi-type iteration applied to a linear system with the elastic operator A_h^{elas} has the form

$$S_h^{\text{elas}} = I - \frac{\omega h^2}{b(\mu,\lambda)} A_h^{\text{elas}}, \tag{34}$$

where

$$b(\mu,\lambda) = \begin{cases} 2(3\mu+\lambda) & \text{for } d=2, \\ 2(4\mu+\lambda) & \text{for } d=3. \end{cases}$$

Theorem 7.4.2. The local smoothing factor, see Def. 6.3.9, for the smoothing operator from Eq. (34) is

$$\mu_{\text{loc}}(S_h^{\text{elas}}) = \begin{cases} \max\left\{|1-4\omega\frac{2\mu+\lambda}{(3\mu+\lambda)}|, |1-\omega\frac{\mu}{(3\mu+\lambda)}|\right\} & \text{for } d=2, \\ \max\left\{|1-6\omega\frac{2\mu+\lambda}{(4\mu+\lambda)}|, |1-\omega\frac{\mu}{(4\mu+\lambda)}|\right\} & \text{for } d=3 \end{cases}$$

for positive Lamé-constants μ, λ.

We see, that the choice of the relaxation parameter ω depends on the ratio of the Lamé constants. This means, that a direct comparison to the smoothing factors in Thm. 7.2.4/ Thm. 7.2.7 is difficult. Hence, we consider a concrete example.

Example 7.4.3 (Local Smoothing Factor of the Elastic Multigrid Solver)**.** For the standard setting of the Lamé constants, i.e. $\lambda = 0, \mu = 1$, and a relaxation parameter $\omega = 0.5$, we end up with

$$\mu_{\text{loc}}(S_h^{\text{elas}}) = \begin{cases} \frac{5}{6} & \text{for } d=2, \\ \frac{7}{8} & \text{for } d=3. \end{cases}$$

Table 19: *Approximation* $(\mu_{\text{loc}})^n$ *of the asymptotic convergence speed for a multigrid algorithm for the elastic operator with the local smoothing factor* μ_{loc} *and* n *overall smoothing steps with the relaxation parameter* $\omega = 0.5$ *and the Lamé-constants* $\lambda = 0, \mu = 1$ *for* $d = 2, 3$.

$n =$	1	2	3
$d = 2$	0.8333	0.6944	0.5786
$d = 3$	0.8750	0.7656	0.6699

Clearly, the ω-damped Jacobi-type iteration is a smoother for these parameters. As known the approximation (27), the local smoothing factor μ_{loc} is an approximation to the convergence speed. Therefore, we determine the approximations for different numbers of n_1 pre- and n_2 post-smoothing steps for a certain setting of the Lamé-constants. For $n = n_1 + n_2$ overall smoothing steps with $n = 1, 2, 3$, we obtain the approximations as listed in Tab. 19.

By comparing the results from Ex. 7.4.3 to those of Ex. 7.2.9 we observe, that the local smoothing factor is smaller for the elastic multigrid framework. Hence, it can be expected, that the elastic multigrid scheme is more efficient than the proposed multigrid scheme. The reason is again the fourth-order of the operator A_h, compared to the second-order of the elastic operator A_h^{elas}. But also the elastic multigrid framework has a smoothing factor rather close to one. Thus, we expect to need a few iterations to get close to the solution. This verifies that ω-damped Jacobi-type iterations, lead to poor smoothing factors for strongly coupled components [47, p.6]. For more efficient smoothers, more advanced iterative schemes are needed. Nevertheless, the findings with the simple ω-damped Jacobi-type smoother proof the general suitability for multigrid.

Moreover, we determine an upper bound for the choice of the relaxation parameter ω of the elastic smoothing operator (34) and compare the upper bound for a concrete choice of the Lamé constants.

Corollary 7.4.4. According to Thm. 7.4.2, the ω-damped Jacobi-type iteration is a suitable smoothing procedure, iff

$$\omega < \omega^*(\mu, \lambda),$$

where

$$\omega^*(\mu, \lambda) = \begin{cases} \frac{(3\mu+\lambda)}{2(2\mu+\lambda)} & \text{for } d = 2, \\ \frac{(4\mu+\lambda)}{3(2\mu+\lambda)} & \text{for } d = 3. \end{cases}$$

The standard setting $\lambda = 0, \mu = 1$ needs a relaxation parameter $\omega < \frac{3}{4}$ for $d = 2$ and $\omega < \frac{2}{3}$ for $d = 3$ for the ω-damped Jacobi-type iteration to be a smoother. This is fulfilled for the standard setting of $\omega = 0.5$. As demonstrated in Exp. 7.2.6 the choice $\omega = 0.5$ is not suitable for the proposed multigrid framework for the second-order DC regularizer. The relaxation parameter needs to be smaller for almost all $\gamma \in [0, 1]$. The relaxation parameter can be interpreted as a weighting factor of the new update or as a step size. If the relaxation parameter is small, we stay near the old iterate. Thus, a too small relaxation parameter is expected to slow the efficiency of the multigrid algorithm down.

7.4.2 Comparison of the Results of Numerical Experiments

We repeat some of the numerical experiments from Sec. 7.3 for the elastic multigrid solver to interpret the experimental results of the proposed multigrid solver.

Table 20: *Average defect reduction factor $\hat{q}^{(20)}$ and the stabilized average defect reduction factor $\hat{q}^{(5,20)}$ for different grid sizes $m = 2^j(16,16),\ j = 0,1,\dots,5$ for the multigrid solver for the elastic regularizer; see Exp. 7.4.5.*

$j =$	0	1	2	3	4	5
$q^{(20)}$	0.8158	0.8263	0.8240	0.8239	0.8268	0.8288
$\hat{q}^{(5,20)}$	0.8614	0.8727	0.8704	0.8723	0.8723	0.8735

First, we repeat Exp. 7.3.2, i.e. the h-independence experiment, for the elastic multigrid solver.

Experiment 7.4.5 (h-Independence of the Elastic Multigrid Solver)**.** We set the initial guess u_h^0 to a random vector and consider the homogeneous linear system, i.e. $f_h = 0$. In Tab. 20 the averaged measurements for the average defect reduction factor $\hat{q}^{(20)}$ and the stabilized average defect reduction factor $\hat{q}^{(5,20)}$ after $k = 20$ iterations with $k_0 = 5$ for a $V(1,0)$-cycle for different grid sizes $m^j = 2^j(16,16),\ j = 0,1,\dots,5$ are given. Due to the staggered grid discretization, we have $2^{2j+9} + 2^{j+5}$ unknowns. Again, the results are obtained as 100 averaged measurements.

The elastic multigrid solver has an average defect reduction factor and a stabilized average defect reduction factor in the same order of magnitude for all grid sizes m. This resembles the proposed multigrid solver. Whereas the average defect reduction factor is slightly smaller for the elastic multigrid solver than for the proposed multigrid solver, the stabilized average defect reduction factor is very similar for both multigrid solvers. This suggests a similar convergence of the proposed and of the elastic solver, which is not expected from LFA; see Sec. 7.4.1.

Table 21: *Average defect reduction $q^{(20)}$ and stabilized average defect reduction $\hat{q}^{(20)}$of a $V(n_1,n_2)$- or $W(n_1,n_2)$-cycle for a multigrid solver for the elastic operator with n_1 pre- and n_2 post-smoothing steps. Both quantities are measured 100 times and averaged; see Exp. 7.4.6. The smallest, i.e. best approximated convergence rates are highlighted in blue.*

	$(n_1,n_2) =$	(1,0)	(2,0)	(1,1)	(2,1)	(1,2)	(3,0)
	$(\mu_{\text{loc}})^{n_1+n_2}$	0.8333	0.6944	0.6944	0.5786	0.5786	0.5786
V-cycle	$q^{(20)}$	0.8200	0.8018	0.8017	0.7949	0.7950	0.7950
	$\hat{q}^{(5,20)}$	0.8640	0.8627	0.8625	0.8630	0.8631	0.8632
W-cycle	$q^{(20)}$	0.8142	0.7942	0.7973	0.7892	0.7915	0.7872
	$\hat{q}^{(5,20)}$	0.8568	0.8538	0.8575	0.8565	0.8590	0.8541

Second, we further compare the solvers and repeat the convergence experiment, i.e. Exp. 7.3.3, for the elastic multigrid solver.

Experiment 7.4.6 (Convergence of the Elastic Multigrid Solver)**.** We build on the setting from Exp. 7.4.5 and consider different numbers of n_1 pre- and n_2 post-smoothing steps

for a fixed grid size $m = (128, 128)$ and compare a V- against a W-cycle architecture. As expected, the stabilized average defect reduction $\hat{q}^{(5,20)}$ leads to larger values, than the average defect reduction $\hat{q}^{(20)}$; see Tab. 21. Furthermore, we compare these numerical approximations of the convergence speed to the theoretical approximation (27). We observe, that the order of magnitude only coincides for $n = n_1 + n_2 = 1$, as the theoretical reduction with more smoothing steps can numerically not be reached. Again the V- and W-cycle achieve the same order of magnitude.

A comparison between the results of Exp. 7.3.3 and Exp. 7.4.6 again suggests, that the stabilized average defect reduction factor $\hat{q}^{(5,20)}$ is almost the same for both solvers. The average defect reduction $\hat{q}^{(20)}$ is slightly smaller, for the elastic solver, which suggests that the solver is most efficient within the first few iterates. The expected convergence rate from LFA cannot be reached by the elastic solver, whereas the proposed multigrid solver outperforms the expected convergence rates from LFA.

7.5 Conclusion

In this chapter we provided a multigrid solver for image registration with second-order DC regularization and analyzed its suitability with LFA, a comprehensive validation and further compared these results to a state of the art multigrid solver.
In particular, we showed through LFA, that the discretized differential operator A_h is h-elliptic and that the ω-damped Jacobi-type iteration is a suitable smoother for an appropriate choice of the relaxation parameter. We explicitly provided an upper bound for the relaxation parameter in dependence of the parameter γ. Note, that it is known, that the simple point-wise Jacobi-type iterations lead to a poor smoothing factor for coupled components; see [47]. Nevertheless, the findings are a proof of concept, that the staggered discretization of the second-order DC regularizer is in general suitable for multigrid solvers.
Furthermore, we compared the theoretical LFA results and the experimental results of the proposed multigrid solver to those, of the well-known first-order elastic regularizer from [46]. For the second-order DC regularizer a coupled fourth-order PDE is obtained, whereas the elastic regularization leads to a coupled second-order PDE. The elastic regularizer was chosen, as its discretization is also staggered and coupled in the components. The comparison of the LFA results predicts a better convergence speed for the elastic multigrid solver, than for the proposed multigrid solver, due to the different orders. Nevertheless, we observe similar convergence rates in our numerical experiments. Another more advanced smoother for coupled approaches is expected to lead to a more efficient solver.
A comparison with a conjugate gradient solver further demonstrates the poor efficiency, due to a long runtime.
Moreover, the counting of the flops shows, that the solver has an $\mathcal{O}(N)$ complexity, where N is the number of unknowns. In conclusion, the proposed second-order DC multigrid solver, though not optimal, turned out to be functioning.

8 Discussion and Future Research

In this thesis we studied second-order divergence-curl (DC) regularization for variational image registration. This chapter highlights the knowledge, that we gained and serves to discuss the proposed work. Furthermore, we provide ideas for further research, that is related to this thesis.

8.1 Discussion

The choice of a suitable regularizer is a demanding task in the field of variational image registration. Regularization ensures the existence of a solution, but also models the space of solutions. Typical regularizers are based on derivatives to ensure a certain smoothness of the transformation. In this thesis, we studied a higher-order regularizer, which comes along with the advantage of a corresponding interpolation energy, that can e.g. be used in combination with a landmark-based pre-registration in the context of variational image registration. Moreover, the second-order DC regularizer is inspired by the physics of fluids and thus enables an interpretation and furthermore a coupling in the components. Both properties, the second-order as well as the coupling in the components pose challenges, when it comes to theoretical analysis and numerical treatment.

The contributions of this thesis are twofold:

First, we studied the second-order DC regularizer in a continuous setting.
We determined the natural boundary conditions of the continuous second-order DC regularizer. As these are coupled in the components and rather complex to implement, we suggest a second-order DC regularization with sliding boundary conditions.
We presented the spectral decomposition of the second-order DC regularizer with sliding boundary conditions. The knowledge of the eigenfunctions enables an understanding of the frequency space. This is in particular helpful, as multigrid solvers, as also proposed for the second-order DC regularizer, rely on smoothing of the high frequencies. The assumption of sliding boundary conditions simplifies the discretization, but changes the regularization. The restriction of a fixed boundary is not suitable for all image registration applications.

Second, we presented a framework for the numerical solution of the considered image registration problem with second-order DC regularization.
We presented discretizations of second-order DC regularizers based on finite differences and staggered grids. The discretization for sliding boundary conditions is appropriate, as the discretized version of the regularizer is shown to be consistent and convergent. The close relation between the continuous regularizer and its discrete analogue was further emphasized, as numerical experiments demonstrated the transferability of the spectral decomposition to the discrete setting.
We proposed a multigrid solver for the fourth-order equation, that is e.g. obtained within Gauss-Newton optimization schemes of the variational problem. A local Fourier analysis gives a solid theoretical foundation for the proposed multigrid framework. We ensured the h-ellipticity and thus the suitability of the proposed discretization for a multigrid scheme. Moreover, we determined for a simple ω-Jacobi-type iteration the smoothing factor and thus demonstrated its smoothing capability. We successfully implemented a multigrid solver for variational image registration with second-order DC regularization, that fits into the FAIR [5] framework. We assume that more advanced smoothers

could further improve the convergence speed and thus the efficiency. It is common for other fourth-order equations, to split the equation into a system of second-order equations to speed up the multigrid performance; see e.g. [55]. Thus, it cannot be expected, that the proposed multigrid solver for a fourth-order equation is highly efficient. Nevertheless, the findings of this thesis emphasize the suitability of the second-order DC regularizer for multigrid schemes and motivate for further research.

8.2 Future Research

The presented study of second-order DC regularization in variational image registration has much potential for future development. Remaining problems, that have not yet been solved are:

- Based on Ch. 3 : A general analysis of the kernel of the second-order DC regularizer. We assume, that certain harmonic functions, as for the curvature regularizer [89], are in the kernel of the second-order DC regularizer. With the knowledge of the concrete kernel a suitable restriction of the space of displacements as in [89] can be formulated. Our restriction to the space with sliding boundary conditions in Ch. 4 is a first approach.
- Based on Ch. 4:
  - A spectral decomposition of the second-order DC regularizer with periodic boundary conditions, in additions to the sliding boundary conditions, as demonstrated for the elastic operator in [1, § 9.2.3]. This enables the usage of fast Fourier transforms to solve the inverse image registration problem in complexity $\mathcal{O}(N \log N)$, where N is the number of image voxels; see [1, §9.5].
  - A rigorous proof for the relation of the continuous and discrete spectral decomposition of the second-order DC regularizer with sliding BC. Exp. 4.3.3 suggests a direct connection analogous to the elastic regularizer [107]. The knowledge would enable an interpretation of the discrete fourth-order operator, that can e.g. be used for multigrid.
- Based on Ch. 5:
  - A discretization of the second-order DC regularizer with the natural boundary conditions from Thm. 3.1.2/Thm. 3.1.3 e.g. with the ghost-cell method [103, 129]. This would enable a comparison between the solutions of the problem with natural to the problem with sliding boundary conditions.
  - A discretization with finite elements, instead of finite differences, as proposed for the hyperelastic regularizer [47]. This discretization could probably improve the h-ellipticity measure of the second-order DC regularizer in Ch. 7. Note, that standard discretizations with conformal finite elements are not expected to be suitable for fourth-order differential equations, but non-conformal methods are known to ensure that C^1 elements are obtained; see [130] for details.
- Based on Ch. 7:
  - An adaptation of the multigrid framework from Ch. 7 to the discretization with sliding boundary conditions. In particular, the restriction and prolongation operators need to be adapted. As the proposed discretization with sliding

boundary conditions is the only consistent and convergent discretization, it is only meaningful to use it. Furthermore, it would enable an interpretation of the frequencies from Thm. 4.3.1 as outlined above.

  - An extension of the multigrid framework, to more advanced smoothing strategies, such as Gauss-Seidel iterations, as well as an examination of higher-order prolongation/interpolation operators; see [31]. This is a promising next step to improve the convergence speed of the multigrid framework.
  - A splitting of the fourth-order equation into a second-order one based on e.g. the Helmholtz decomposition used in [29]. We expect the splitting to speed-up the performance of a corresponding multigrid framework, as it is observable for the biharmonic equation; see [31, §8.4].

- In general: A comprehensive evaluation of the second-order DC regularizer in comparison to other regularization strategies in the context of image registration. In [28] and [29] some image registration applications were demonstrated for the second-order DC regularizer. Nevertheless a comprehensive comparison to other regularizers would enable an understanding for the applications where the regularizer is superior to other regularization strategies.

The contribution of this thesis is the study of the second-order DC regularization in image registration. We examined the continuous regularizer and determined its natural boundary conditions, gave a first intuition for its kernel and presented a spectral decomposition for certain boundary conditions. Furthermore, we proposed a discretize-then-optimize approach for the solution of the considered image registration problem. This includes a discretization of the regularizer and the presentation of a novel multigrid method. Overall, the findings of this thesis motivate for further studies of the second-order DC regularizer.

Bibliography

[1] Jan Modersitzki. *Numerical methods for image registration.* Numer. Math. Sci. Comput. Oxford: Oxford University Press, 2004.

[2] Joseph V Hajnal and Derek LG Hill. *Medical image registration*. Boca Raton, FL: CRC Press, 2001.

[3] Nikos Paragios, James Duncan, and Nicholas Ayache, editors. *Handbook of biomedical imaging. Methodologies and clinical research*. New York, NY: Springer, 2015.

[4] Jacqueline Le Moigne, Nathan S. Netanyahu, and Roger D. Eastman, editors. *Image registration for remote sensing*. Cambridge: Cambridge University Press, 2011.

[5] Jan Modersitzki. *FAIR. Flexible algorithms for image registration.*, volume 6 of *Fundam. Algorithms*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2009.

[6] Heinz Handels. *Medizinische Bildverarbeitung*. Leipzig: Teubner, 2000.

[7] Ke Chen, Lok Ming Lui, and Jan Modersitzki. Image and surface registration. In *Processing, analyzing and learning of images, shapes, and forms. Part 2*, pages 579–611. Amsterdam: Elsevier/North Holland, 2019.

[8] JB Antoine Maintz and Max A Viergever. A survey of medical image registration. *Med. Image Anal.*, 2(1):1–36, 1998.

[9] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image Vis. Comput.*, 21(11):977–1000, 2003.

[10] Max A Viergever, JB Antoine Maintz, Stefan Klein, Keelin Murphy, Marius Staring, and Josien PW Pluim. A survey of medical image registration–under review. *Med. Image Anal.*, 33:140–144, 2016.

[11] Bernd Fischer and Jan Modersitzki. Ill-posed medicine – an introduction to image registration. *Inverse Probl.*, 24(3):16, 2008. Id/No 034008.

[12] Francisco PM Oliveira and Joao Manuel RS Tavares. Medical image registration: a review. *Comput. Methods Biomech. Biomed. Engin.*, 17(2):73–93, 2014.

[13] Xingyu Jiang, Jiayi Ma, Guobao Xiao, Zhenfeng Shao, and Xiaojie Guo. A review of multimodal image matching: Methods and applications. *Inf. Fusion*, 73:22–71, 2021.

[14] Anna Reithmeir, Veronika Spieker, Vasiliki Sideri-Lampretsa, Daniel Rueckert, Julia A Schnabel, and Veronika A Zimmer. From model based to learned regularization in medical image registration: A comprehensive review. *arXiv preprint arXiv:2412.15740*, 2024.

[15] Luminita A Vese and Carole Le Guyader. *Variational methods in image processing*. Chapman Hall/CRC Math. Comput. Imaging Sci. Ser. Boca Raton, FL: CRC Press, 2016.

[16] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, and Frank Lenzen. *Variational methods in imaging*, volume 167 of *Appl. Math. Sci.* New York, NY: Springer, 2009.

[17] Guoli Song, Jianda Han, Yiwen Zhao, Zheng Wang, and Huibin Du. A review on medical image registration as an optimization problem. *Curr. Med. Imaging Rev.*, 13(3):274–283, 2017.

[18] Grant Haskins, Uwe Kruger, and Pingkun Yan. Deep learning in medical image registration: a survey. *Mach. Vis. Appl.*, 31(1):8, 2020.

[19] Yabo Fu, Yang Lei, Tonghe Wang, Walter J Curran, Tian Liu, and Xiaofeng Yang. Deep learning in medical image registration: a review. *Phys. Med. Biol.*, 65(20):20TR01, 2020.

[20] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Optical flow modeling and computation: A survey. *Comput. Vis. Image Underst.*, 134:1–21, 2015.

[21] Leon Bungert and Martin Burger. Solution paths of variational regularization methods for inverse problems. *Inverse Probl.*, 35(10):36, 2019. Id/No 105012.

[22] Lars Ruthotto. *Hyperelastic image registration*. PhD thesis, Westfälische Wilhelms-Universität Münster, 2012.

[23] Luca Amodei and Mohammed-Najib Benbourhim. A vector spline approximation. *J. Approx. Theory*, 67(1):51–79, 1991.

[24] Luca Amodei and Mohammed-Najib Benbourhim. A vector spline approximation with application to meteorology. In *Curves and surfaces, Pap. Int. Conf., Chamonix-Mont-Blanc/Fr. 1990*, pages 5–10. Elsevier, 1991.

[25] Fabrice Dodu and Christophe Rabut. Vectorial interpolation using radial-basis-like functions. *Comput. Math. Appl.*, 43(3-5):393–411, 2002.

[26] Mohammed-Najib Benbourhim and Abderrahman Bouhamidi. Approximation of vectors fields by thin plate splines with tension. *J. Approx. Theory*, 136(2):198–229, 2005.

[27] David Suter. Motion estimation and vector splines. In *Proceedings of IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 939–942. IEEE, 1994.

[28] Carlos Oscar Sánchez Sorzano, Philippe Thévenaz, and Michael Unser. Elastic registration of biological images using vector-spline regularization. *IEEE Trans. Biomed. Eng.*, 52(4):652–663, 2005.

[29] Paris Tzitzimpasis, Mario Ries, Bas W Raaymakers, and Cornel Zachiu. Generalized div-curl based regularization for physically constrained deformable image registration. *Sci. Rep.*, 14(1):15002, 2024.

[30] Bryan E Schmidt and Jeffrey A Sutton. A physical interpretation of regularization for optical flow methods in fluids. *Exp. Fluids*, 62(2):34, 2021.

[31] Ulrich Trottenberg, Cornelius W Oosterlee, and Anton Schuller. *Multigrid*. Orlando, FL: Academic Press, 2001.

[32] Pieter Wesseling. *An introduction to multigrid methods*. Chichester: John Wiley & | Sons Ltd., 1992.

[33] Achi Brandt. Guide to multigrid development. In *Multigrid methods, Proc. Conf., Köln-Porz 1981, Lect. Notes Math. 960*, pages 220–312. Springer, 1982.

[34] Radii Petrovich Fedorenko. The speed of convergence of one iterative process. *U.S.S.R. Comput. Math. Math. Phys.*, 4(3):227–235, 1966.

[35] Nikolai Sergeevich Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *U.S.S.R. Comput. Math. Math. Phys.*, 6(5):101–135, 1966.

[36] Achi Brandt. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In *Proc. third internat. Conf. numer. Methods Fluid Mech., Univ. Paris 1972, 1, Lecture Notes Phys. 18*, pages 82–89. Springer, 1973.

[37] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, 31:333–390, 1977.

[38] Achi Brandt. Multilevel adaptive computations in fluid dynamics. *AIAA J.*, 18:1165–1172, 1980.

[39] Wolfgang Hackbusch. On the multi-grid method applied to difference equations. *Computing*, 20:291–306, 1978.

[40] Wolfgang Hackbusch. On the convergence of multi-grid iterations. *Beitr. Numer. Math.*, 9:213–239, 1981.

[41] Irad Yavneh. Why multigrid methods are so efficient. *Comput. Sci. Eng.*, 8(6):12–22, 2006.

[42] Dewen Yushu and Karel Matouš. The image-based multiscale multigrid solver, preconditioner, and reduced order model. *J. Comput. Phys.*, 406:23, 2020. Id/No 109165.

[43] Ana Budiša, Xiaozhe Hu, Miroslav Kuchta, Kent-Andre Mardal, and Ludmil Zikatanov. Algebraic multigrid methods for metric-perturbed coupled problems. *SIAM J. Sci. Comput.*, 46(3):a1461–a1486, 2024.

[44] Álvaro Pé de la Riva, Carmen Rodrigo, Francisco J Gaspar, James H Adler, Xiaozhe Hu, and Ludmil Zikatanov. A local Fourier analysis for additive Schwarz smoothers. *Comput. Math. Appl.*, 158:13–20, 2024.

[45] Pierre Hellier, Christian Barillot, Etienne Mémin, and Patrick Pérez. Medical image registration with robust multigrid techniques. In *Proc. Med. Image Comput. Comput. Assist. Interv.*, pages 680–687. Springer, 1999.

[46] Eldad Haber and Jan Modersitzki. A multilevel method for image registration. *SIAM J. Sci. Comput.*, 27(5):1594–1607, 2006.

[47] Lars Ruthotto, Chen Greif, and Jan Modersitzki. A stabilized multigrid solver for hyperelastic image registration. *Numer. Linear Algebra Appl.*, 24(5):e2095, 14, 2017.

[48] Noppadol Chumchob and Ke Chen. An augmented lagrangian method for solving total variation (TV)-based image registration model. *J. Algorithm. Comput. Technol.*, 14:1–16, 2020.

[49] Max Blendowski, Lasse Hansen, and Mattias P Heinrich. Weakly-supervised learning of multi-modal features for regularised iterative descent in 3D image registration. *Med. Image Anal.*, 67:101822, 2021.

[50] Paul Striewski and Benedikt Wirth. Elastic 3D-2D image registration. *J. Math. Imaging Vis.*, 64(5):443–462, 2022.

[51] Johannes Lotz. *Combined Local and Global Image Registration and its Application to Large-Scale Images in Digital Pathology*. PhD thesis, Institute of Mathematics and Image Computing, University of Lübeck, 2020.

[52] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49:409–436, 1952.

[53] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. New York, NY: Springer, 2nd ed. edition, 2006.

[54] Huan Han, Zhengping Wang, and Yimin Zhang. Multiscale approach for three-dimensional conformal image registration. *SIAM J. Imaging Sci.*, 15(3):1431–1468, 2022.

[55] Harald Köstler, Kai Ruhnau, and Roman Wienands. Multigrid solution of the optical flow system using a combined diffusion-and curvature-based regularizer. *Numer. Linear Algebra Appl.*, 15(2-3):201–218, 2008.

[56] Adrian Celaya, Beatrice Riviere, and David Fuentes. FMG-net and W-net: Multigrid inspired deep learning architectures for medical imaging segmentation. *arXiv preprint arXiv:2304.02725*, 2023.

[57] Roman Wienands and Wolfgang Joppich. *Practical Fourier analysis for multigrid methods*, volume 4 of *Numer. Insights*. Boca Raton, FL: Chapmaqn & Hall/CRC, 2005.

[58] Sandeep N Gupta and Jerry L Prince. On div-curl regularization for motion estimation in 3-D volumetric imaging. In *Proc. Int. Conf. Image Proc.*, volume 1, pages 929–932. IEEE, 1996.

[59] Timo Kohlberger, Christoph Schnörr, and Etienne Mémin. Variational dense motion estimation using a div-curl high-order regularization. In *Proc. ISSPA*, volume 1, pages 641–644. IEEE, 2003.

[60] Thomas Corpetti, Étienne Mémin, and Patrick Pérez. Dense estimation of fluid flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):365–380, 2002.

[61] Jing Yuan, Christoph Schnörr, and Etienne Mémin. Discrete orthogonal decomposition and variational fluid flow estimation. *J. Math. Imaging Vis.*, 28(1):67–80, 2007.

[62] Fang Chen and David Suter. Image coordinate transformation based on div-curl vector splines. In *Proc. Int. Conf. Pattern Recognit.*, volume 1, pages 518–520. IEEE, 1998.

[63] Ignacio Arganda-Carreras, Carlos OS Sorzano, Roberto Marabini, José María Carazo, Carlos Ortiz-de Solorzano, and Jan 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.

[64] Ching-Wei Wang, Eric Budiman Gosno, and Yen-Sheng Li. Fully automatic and robust 3D registration of serial-section microscopic images. *Sci. Rep.*, 5(1):1–14, 2015.

[65] Mohammed Oulhadj, Jamal Riffi, Khodriss Chaimae, Adnane Mohamed Mahraz, Bennis Ahmed, Ali Yahyaouy, Chraibi Fouad, Abdellaoui Meriem, Benatiya Andaloussi Idriss, and Hamid Tairi. Diabetic retinopathy prediction based on deep learning and deformable registration. *Multim. Tools Appl.*, 81(20):28709–28727, 2022.

[66] Osman Akdag, Stefano Mandija, Pim TS Borman, Paris Tzitzimpasis, Astrid LHMW van Lier, Rick Keesman, Bas W Raaymakers, and Martin F Fast. Evaluation of the impact of cardiac implantable electronic devices on cine MRI for real-time adaptive cardiac radioablation on a 1.5 T MR-linac. *Med. Phys.*, 52(1):99–112, 2025.

[67] Stefan Henn, Thorsten Schormann, Knut Engler, Karl Zilles, and Kristian Witsch. Elastische Anpassung in der digitalen Bildverarbeitung auf mehreren Auflösungsstufen mit Hilfe von Mehrgitterverfahren. In *Mustererkennung 1997, Informatik aktuell*, pages 392–399. Springer, Berlin, Heidelberg, 1997.

[68] Stefan Henn and Kristian Witsch. A multigrid approach for minimizing a nonlinear functional for digital image matching. *Computing*, 64(4):339–348, 2000.

[69] Stefan Henn and Kristian Witsch. Iterative multigrid regularization techniques for image matching. *SIAM J. Sci. Comput.*, 23(4):1077–1093, 2001.

[70] Eldad Haber, Raya Horesh, and Jan Modersitzki. Numerical optimization for constrained image registration. *Numer. Linear Algebra Appl.*, 17(2-3):343–359, 2010.

[71] Darko Zikic, Wolfgang Wein, Ali Khamene, Dirk-André Clevert, and Nassir Navab. Fast deformable registration of 3D-ultrasound data using a variational approach. In *Proc. Med. Image. Comput. Comput. Assist. Interv.*, pages 915–923. Springer, 2006.

[72] Claudia Frohn-Schauf, Stefan Henn, and Kristian Witsch. Multigrid based total variation image registration. *Comp. Vis. Sci.*, 11(2):101–113, 2008.

[73] Huan Han and Andong Wang. A fast multi grid algorithm for 2D diffeomorphic image registration model. *J. Comput. Appl. Math.*, 394:19, 2021. Id/No 113576.

[74] Huan Han, Zhengping Wang, and Yimin Zhang. Multiscale approach for two-dimensional diffeomorphic image registration. *Multiscale Model. Simul.*, 19(4):1538–1572, 2021.

[75] Zijuan Ding, Huan Han, and Huinan Wang. Two-dimensional diffeomorphic model for multi-modality image registration. *Comput. Appl. Math.*, 42(1):34, 2023. Id/No 17.

[76] Stefan Henn. A multigrid method for a fourth-order diffusion equation with application to image processing. *SIAM J. Sci. Comput.*, 27(3):831–849, 2005.

[77] Noppadol Chumchob, Ke Chen, and Carlos Brito-Loeza. A fourth-order variational image registration model and its fast multigrid algorithm. *Multiscale Model. Simul.*, 9(1):89–128, 2011.

[78] Noppadol Chumchob and Ke Chen. A robust multigrid approach for variational image registration models. *J. Comput. Appl. Math.*, 236(5):653–674, 2011.

[79] Patrick Héas and Etienne Mémin. Three-dimensional motion estimation of atmospheric layers from image sequences. *IEEE Trans. Geosci. Remote Sens.*, 46(8):2385–2396, 2008.

[80] Saskia Neuber, Pia F Schulz, Sven Kuckertz, and Jan Modersitzki. Segmentation-inspired image registration. In *BVM Workshop*, pages 205–210. Springer, 2024.

[81] Saskia Neuber and Jan Modersitzki. A multigrid approach for fourth-order equations in image registration. In *Proc. SSVM*, pages 391–403. Springer, 2025.

[82] Martin Burger, Jan Modersitzki, and Lars Ruthotto. A hyperelastic regularization energy for image registration. *SIAM J. Sci. Comput.*, 35(1):b132–b148, 2013.

[83] J Michael Fitzpatrick, Derek LG Hill, and Calvin R Maurer. Image registration. In *Handbook of medical imaging*, volume 2, pages 447–513. Washington, USA: SPIE, 2000.

[84] A Ardeshir Goshtasby. *Image registration. Principles, tools and methods.* Adv. Comput. Vis. Pattern Recognit. London: Springer, 2012.

[85] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, pages 49–52, 1902.

[86] Bernd Fischer and Jan Modersitzki. Curvature based image registration. *J. Math. Imaging Vis.*, 18(1):81–85, 2003.

[87] Berthold K P Horn and Brian G Schunck. Determining optical flow. *Artif. Intell.*, 17(1-3):185–203, 1981.

[88] Chaim Broit. *Optimal registration of deformed images*. PhD thesis, University of Pennsylvania, 1981.

[89] Stefan Henn and Kristian Witsch. A variational image registration approach based on curvature scale space. In *Proc. Scale space and PDE methods in computer vision*, pages 143–154. Berlin: Springer, 2005.

[90] Noppadol Chumchob. Vectorial total variation-based regularization for variational image registration. *IEEE Trans. Image Process.*, 22(11):4551–4559, 2013.

[91] Stefan Heldmann and Nils Papenberg. A variational approach for volume-to-slice registration. In *Proc. SSVM*, pages 624–635. Springer, 2009.

[92] Pia F Schulz, Florian Mannel, and Jan Modersitzki. Appropriate order of regularization in 3D/2D image registration. In *Proc. SSVM*, pages 377–390. Springer, 2025.

[93] Jean Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constr. Theory Funct. several Variables, Proc. Conf. Oberwolfach 1976, Lect. Notes Math. 571*, pages 85–100, 1977.

[94] Bernd Fischer and Jan Modersitzki. In *Proc. SPIE. Medical Imaging 2003: Image Processing*, volume 5032, pages 1037–1048. SPIE, 2003.

[95] Wolfgang Hackbusch. *Elliptic differential equations. Theory and numerical treatment*, volume 18 of *Springer Ser. Comput. Math.* Berlin: Springer, 2nd edition edition, 2017.

[96] Thomas Pock, Martin Urschler, Christopher Zach, Reinhard Beichel, and Horst Bischof. A duality based algorithm for TV-L^1-optical-flow image registration. In *Proc. Med. Image Comput. Comput. Assist. Interv.*, pages 511–518. Springer, 2007.

[97] Susan Jane Colley and Santiago Cañez. *Vector calculus*, volume 2. Prentice Hall Upper Saddle River, NJ, 1998.

[98] Lizhong Peng and Lei Yang. The curl in seven dimensional space and its applications. *Approximation Theory Appl.*, 15(3):66–80, 1999.

[99] Fred L Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 11(6):567–585, 1989.

[100] Markus Ekvall, Ludvig Bergenstråhle, Alma Andersson, Paulo Czarnewski, Johannes Olegård, Lukas Käll, and Joakim Lundeberg. Spatial landmark detection and tissue registration with deep learning. *Nat. Methods*, 21(4):673–679, 2024.

[101] Michael Grinfeld, editor. *Mathematical tools for physicists. Encyclopedia of applied physics*. Weinheim: Wiley-VCH, 2nd updated and extended ed. edition, 2015.

[102] Konrad Königsberger. *Analysis 2*. Springer-Lehrb. Berlin: Springer, 5., korrigierte Aufl. edition, 2004.

[103] Ronald P. Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152(2):457–492, 1999.

[104] Alexander Schmidt-Richberg, René Werner, Heinz Handels, and Jan Ehrhardt. Estimation of slipping organ motion by registration with direction-dependent regularization. *Med. Image Anal.*, 16(1):150–159, 2012.

[105] Ole Gildemeister, Johannes Bostelmann, Pia F Schulz, Andra Oltmann, Phillip Rostalski, Jan Modersitzki, and Jan Lellmann. Time-continuous sliding motion image registration using stationary velocity fields for respiratory motion interpolation. In *Proc. SSVM*, pages 404–416. Springer, 2025.

[106] David M. Day and Louis A. Romero. An analytically solvable eigenvalue problem for the linear elasticity equations. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA, 2004.

[107] Jan Modersitzki. The spectral decomposition of the continuous and discrete linear elasticity operators with sliding boundary conditions. *SIAM J. Matrix Anal. Appl.*, 45(1):134–147, 2024.

[108] E. Ahusborde. Legendre spectral methods for the -grad (div) operator with free boundary conditions. *Numer. Algorithms*, 52(2):151–171, 2009.

[109] Lawrence C. Evans. *Partial differential equations*, volume 19 of *Grad. Stud. Math.* Providence, RI: American Mathematical Society (AMS), 2nd ed. edition, 2010.

[110] Michael Hinze and Arnd Rösch. Discretization of optimal control problems. In *Constrained optimization and optimal control for partial differential equations*, pages 391–430. Basel: Birkhäuser, 2012.

[111] John T Betts and Stephen L Campbell. Discretize then optimize. In *Proc. SIAM. Mathematics for industry: challenges and frontiers*, pages 140–157. Philadelphia, PA: SIAM, 2005.

[112] Eldad Haber and Jan Modersitzki. Numerical methods for volume preserving image registration. *Inverse Probl.*, 20(5):1621–1638, 2004.

[113] James W Thomas. *Numerical partial differential equations: Finite difference methods*, volume 22 of *Texts Appl. Math.* New York, NY: Springer-Verlag, 1995.

[114] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations. Steady-state and time-dependent problems*. Philadelphia, PA: SIAM, 2007.

[115] Hans Petter Langtangen and Svein Linge. *Finite difference computing with PDEs. A modern software approach*, volume 16 of *Texts Comput. Sci. Eng.* Cham: Springer Open, 2017.

[116] Philip J. Davis and Philip Rabinowitz. *Methods of numerical integration*. Mineola, NY: Dover Publications, corrected reprint of the 1984 2nd ed. edition, 2007.

[117] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Comput. Surv.*, 23(1):5–48, 1991.

[118] The MathWorks Inc. Matlab version: 9.13.0 (r2022b), 2022.

[119] John E. jun. Dennis and Robert B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.*, volume 16 of *Class. Appl. Math.* Philadelphia, PA: SIAM, repr. edition, 1996.

[120] Lars König, Jan Rühaak, Alexander Derksen, and Jan Lellmann. A matrix-free approach to parallel and memory-efficient deformable image registration. *SIAM J. Sci. Comput.*, 40(3):b858–b888, 2018.

[121] Andreas Mang and Lars Ruthotto. A Lagrangian Gauss-Newton-Krylov solver for mass- and intensity-preserving diffeomorphic image registration. *SIAM J. Sci. Comput.*, 39(5):b860–b885, 2017.

[122] Josef Stoer and Roland Z Bulirsch. *Introduction to numerical analysis. Transl. from the German by R. Bartels, W. Gautschi, and C. Witzgall.*, volume 12. New York, NY: Springer, 3rd ed. edition, 2002.

[123] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A multigrid tutorial.* Philadelphia, PA: SIAM, 2nd ed. edition, 2000.

[124] Wolfgang Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Stuttgart: Teubner, 2., überarb. u. erw. edition, 1993.

[125] Carmen Rodrigo, Francisco J. Gaspar, and Ludmil T. Zikatanov. On the validity of the local Fourier analysis. *J. Comput. Math.*, 37(3):340–348, 2019.

[126] Vladimir A. Zorich. *Mathematical analysis I. Translated from the 4th and 6th Russian editions by Roger Cooke and Octavio T. Paniagua*. Universitext. Berlin: Springer, 2nd edition edition, 2015.

[127] Wolfram Research, Inc. Mathematica, Version 14.2, 2024.

[128] Tom Minka. Lightspeed. https://github.com/tminka/lightspeed/, 2016.

[129] Armando Coco and Giovanni Russo. Finite-difference ghost-point multigrid methods on Cartesian grids for elliptic problems in arbitrary domains. *J. Comput. Phys.*, 241:464–501, 2013.

[130] Dietrich Braess. *Finite elements. Theory, fast solvers, and applications in solid mechanics. Transl. from the German by Larry L. Schumaker.* Cambridge: Cambridge University Press, 2nd ed. edition, 2001.