

HIERARCHICAL GRADIENT DOMAIN
VECTOR FIELD PROCESSING

by

Sing Chun Lee

A dissertation submitted to Johns Hopkins University in conformity with the
requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

July, 2023

© 2023 Sing Chun Lee

All rights reserved

Abstract

Vector fields are a fundamental mathematical construct for describing flow-field-related problems in science and engineering. To solve these types of problems effectively on a discrete surface, various vector field representations are proposed using finite dimensional bases, a discrete connection, and an operator approach. Furthermore, for computational efficiency, quadratic Dirichlet energy is preferred to measure the smoothness of the vector field in the gradient domain. However, while quadratic energy gives a simple linear system, it does not support real-time vector field processing on a high-resolution mesh without extensive GPU parallelization. To this end, this dissertation describes an efficient hierarchical solver for vector field processing. Our method extends the successful multigrid design for interactive signal processing on meshes using an induced vector field prolongation combining it with novel speedup techniques. We formulate a general way for extending scalar field prolongation to vector fields. Focusing on triangle meshes, our convergence study finds that a standard multigrid does not achieve fast convergence due to the poorly-conditioned

ABSTRACT

system matrix. We observe a similar performance in standard single-level iterative methods such as the Jacobi, Gauss-Seidel, and conjugate gradient methods. Therefore, we compare three speedup techniques – successive over-relaxation, smoothed prolongation, and Krylov subspace update, and incorporate them into our solver. Finally, we demonstrate our solver on useful applications such as logarithmic map computation and discuss the applications to other hierarchies such as texture grids, followed by the conclusion and future work.

Primary Reader: Misha Kazhdan

Secondary Reader: Nassir Navab

External Readers: Laurent Younes, Szymon Rusinkiewicz

Dissertation Committee

Advisor

Michael (Misha) Kazhdan, Ph.D.

Professor

Department of Computer Science

Johns Hopkins University

Members

Nassir Navab, Ph.D.

Chair of Computer Aided Medical Procedures / Research Professor

School of Computation, Information and Technology / Department of Computer Science

Technische Universität München / Johns Hopkins University

Laurent Younes, Ph.D.

Professor, Director of the Center for Imaging Science

Department of Applied Mathematics and Statistics

Johns Hopkins University

Szymon Rusinkiewicz, Ph.D.

David M. Siegel '83 Professor, Director of Program in Robotics and Intelligent Systems

Department of Computer Science

Princeton University

Acknowledgments

Mere words cannot express my gratitude to everyone who helped and supported me during my Ph.D. journey. I especially want to thank my advisor **Misha Kazhdan** for guiding me through this journey, training me to be a better researcher, presenter, writer, and person, and, more importantly, for pointing me to the right door where my calling lies.

I am also grateful to my secondary advisor **Nassir Navab**, who accepted me as his student at Johns Hopkins University before I dived into the geometry world, for every interesting augmented reality project we did together, playing video games with me during retreats, and, above it all, for encouraging me to pursue what I would enjoy the most during my Ph.D. study.

Thanks to **Laurent Younes** and **Szymon Rusinkiewicz** for serving on my thesis committee and reading the thesis. Lastly, I would be remiss in not mentioning my family – my **mum** and two younger **brothers**. Their belief in me has kept me moving forward and eventually to prevail.

Dedication

To my mum, Hoi Ying Yip,
for supporting every decision I have made,
even during family hardship.
You shield me; I prevail and succeed.
Koran dankon, panjo!

Contents

Abstract	ii
Acknowledgments	v
List of Tables	xii
List of Figures	xiv
List of Algorithms	xix
1 Introduction	1
1.1 Motivation	2
1.2 Objective	4
1.3 Overview	5
2 Background	7
2.1 Gradient Domain Processing	8
2.2 Hierarchical Solver	10

CONTENTS

2.3	Vector Field Representations	13
3	Gradient Domain Formulation	15
3.1	Preliminaries	16
3.2	Signal Processing	21
3.2.1	The Gradient Domain Framework	21
3.2.2	Examples	21
3.2.2.1	Signal Modification	22
3.2.2.2	Signal Interpolation	23
3.2.2.3	Signal Fusion	24
3.2.3	A Fourier Perspective	25
3.3	Vector Fields Processing	26
3.3.1	The Gradient Domain Formulation	27
3.3.2	Applications	28
3.3.2.1	Vector Field Modification	29
3.3.2.2	Vector Field Interpolation	30
3.3.2.3	Other Applications	31
3.3.3	A Spectral Perspective	32
3.4	Discretization	32
3.5	Summary	33
4	A Hierarchical Approach	34

CONTENTS

4.1	A Two-Level V-Cycle Multigrid	35
4.2	Key Components	37
4.2.1	Relaxation Scheme	38
4.2.2	The Prolongation Matrix	40
4.2.3	Coarse Level Correction	41
4.3	Multigrid	43
4.4	Summary	46
5	Prolongation	47
5.1	Preliminaries	48
5.2	The Finite 1-Form Basis \mathcal{W}	51
5.2.1	Harmonic-Free 1-Form Basis	54
5.2.2	Whitney 1-Form Basis	60
5.3	Generalization	66
5.3.1	Basic Set-up	66
5.3.2	General Vector Space Construction	68
5.3.3	Applications	70
5.3.3.1	Harmonic-Free 1-Form	70
5.3.3.2	Whitney k -Form	73
5.4	Summary	75
6	Convergence Study	76

CONTENTS

6.1	Setup	77
6.1.1	The Models	77
6.1.2	0-Form Hierarchical Structures	78
6.1.2.1	Mesh Simplification	78
6.1.2.2	Sub-division Meshes	79
6.1.2.3	Intrinsic Self-Parameterization	81
6.2	Evaluation	82
6.2.1	0-Form System Convergence	83
6.2.2	1-Form System Convergence	85
6.3	Multigrid Components	89
6.3.1	Relaxation Schemes	89
6.3.2	Prolongation	94
6.3.3	Solution Update	99
6.4	Performance	103
6.4.1	Multigrid Cycles	106
6.4.2	Direct Solver	107
6.5	Summary	109
7	Applications	111
7.1	Parallel Transport	111
7.2	Optical Flow	114
7.3	Other Multigrid Systems	114

CONTENTS

8 Conclusion and Future Work	119
8.1 Conclusion	119
8.2 Future Work	120
Bibliography	123
A Input Hierarichial Meshes	138
B 1-Form System Convergence (Standard V-Cycle)	143
C Parameter Study	145
D Vita	171

List of Tables

6.1	This table shows the relative residual errors of the scalar field diffusion ('hat' basis), vector field diffusion (harmonic-free and Whitney bases) for timestep $\alpha = 0.0001$ after 50 iterations of multigrid V-Cycle using three hierarchical structures (HS) and the number of iterations required to reduce the relative residual error below 10^{-8} .	87
6.2	Convergence improvement using SOR with the over-relaxation weight $\omega_{sor} = 1.7$. The third and fourth columns show the changes in the relative residual errors after fifty iterations before and after the speed up, and the last two columns show the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .	92
6.3	This table shows the changes in convergence using prolongation smoothing with the damping weight $\omega_{dl} = 0.3$. The third and fourth columns are the changes in the relative residual errors after fifty iterations, and the last two columns are the changes in the number of iterations required to obtain accuracy below 10^{-8} .	98
6.4	Convergence improvement using Krylov subspace updates. The third and fourth columns show the improvement in the relative residual errors after fifty iterations before and after the speed up, and the last two columns are the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .	100
6.5	Convergence improvement using the combination of speed up techniques. The third and fourth columns show the improvement in the relative residual errors after fifty iterations before and after the speed up, and the last two columns are the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .	104

LIST OF TABLES

6.6	The relative residual errors of the first six iterations when the delta vector field is at points A , B , and C in the above inset figure and solving the diffusion problem using our hierarchical solver with and without the improvements.	106
6.7	This table lists, from left to right, the models, the system dimensions when using the harmonic-free and Whitney basis, the hierarchies, and the pre-processing and solving time in seconds of using the direct CHOLMOD solver and our hierarchical solver to solve for the diffusion problem step.	108
C.1	The convergence improvement using SOR with best over-relaxation weights ω_{SOR} shown in the third and fourth columns for the harmonic-free (HF) and Whitney (WN) bases. The next two columns show the changes in the relative residual errors after fifty iterations before and after the speed up, and the last two columns show the improvement in the number of iterations required to obtain an accuracy below 10^{-8}	155
C.2	The convergence improvement using Krylov subspace updates with best subspace dimensions (Dim.) shown in the third and fourth columns for the harmonic-free (HF) and Whitney (WN) bases. The next two columns show the changes in the relative residual errors after fifty iterations before and after the speed up, and the last two columns show the improvement in the number of iterations required to obtain an accuracy below 10^{-8}	170

List of Figures

1.1	Example applications of solving the vector diffusion equation as described in the work of Sharp <i>et al.</i> [SSC19] - parallel transport (left), logarithmic map (middle), and Karcher mean (right).	2
1.2	McCann and Pollard [MP08] combine the gradient domain paradigm and a hierarchical solver to develop an interactive painting application that allows artists to paint with digital brushes. The above figures show snapshots during a digital painting process.	3
3.1	Relation between d , δ , and \star on a 2-manifold \mathcal{M}	19
3.2	Smoothing (left) with $\lambda = 0$ and sharpening (right) with $\lambda = 2$ using gradient domain signal processing, where the signal is the mesh coordinate function. Image courtesy of [CRK16].	22
3.3	The contrast of the input image (left) is enhanced using gradient domain signal processing with the target 1-form proposed in [FLW02]. Image courtesy of [FLW02].	23
3.4	Shen <i>et al.</i> [SJZW07] use the gradient domain framework to remove the jumper from the image and fill in the content automatically. A user removes the jumper manually from an image (left-most), resulting in an input to the algorithm (second left), an image with a missing region. Then the algorithm fills the missing region iteratively, as illustrated by the last two images. Image courtesy of [SJZW07].	24
3.5	The input image (left) is fused seamlessly together to produce an image panoramic (right) by Kazhdan <i>et al.</i> [KH08]. Image courtesy of [KH08].	25
3.6	A vector field denoising result using Eq. 3.10. Image courtesy of [SWJG20].	29
3.7	Interpolation results from [FSDH07] with source/sink constraints (left) and an additional curve constraint (right). Image courtesy of [FSDH07].	30

LIST OF FIGURES

4.1	A two-level V-Cycle multigrid method composed of the three key components: i) a relaxation scheme, ii) a restriction/prolongation, and iii) a coarse level correction solve.	36
4.2	A graphical comparison of the three multigrid cycles for $L = 4$. These plots show the computation movement across the hierarchy.	46
5.1	A visualization of the exact and co-exact 1-form bases as vector fields on a flat triangle mesh. Considering a 0-form basis function centered at the interior vertex, we obtain two vector fields – the gradient (exact 1-form) and rotated gradient (co-exact 1-form) of the basis function. The first and second columns show the vector fields obtained when the function is the first-order Lagrange interpolant. The third and fourth columns show the vector fields for a second-order Lagrange interpolant.	55
5.2	A visualization of the Whitney 1-form basis function associated to an edge as a vector field on a flat triangle mesh using a linear and quadratic Lagrange basis.	60
6.1	Wire-frame visualizations of the eight models used in this study and their vertex counts. From left to right, the first row is generated models – the plane, sphere, torus, and 4-torus. The second row is scanned models – the hand, bimba, rooster, and fertility.	77
6.2	The hierarchical meshes of the bimba ¹ model obtained by using quadric error metric simplification algorithm, visualized with wire-frame on top of the meshes.	78
6.3	Sub-division meshes of the rooster ² model, visualized with wire-frame on top of the meshes.	79
6.4	Self-parameterization meshes of the fertility ³ model, visualized with wireframe on top of the meshes.	81
6.5	The convergence of the standard multigrid method using the ‘hat’ 0-form basis when solving an implicit scalar diffusion step with timestep α varying from 0.0001 to 0.01 using the mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP) hierarchies.	84
6.6	The convergence rate of the standard multigrid method when solving an implicit vector diffusion step with varies timestep α from 0.0001 to 0.01 using the hierarchies mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP), as well as the harmonic-free and Whitney 1-form basis.	86

LIST OF FIGURES

6.7	The convergence of damped Jacobi (DJ), successive over-relaxation (SOR), and conjugate gradient (CG) methods when solving an implicit vector diffusion step ($\alpha = 0.0001$) on the fertility model using the harmonic-free (HF) basis and the Whitney (WN) basis, for various relaxation weights $\omega_{DJ}, \omega_{SOR}$	90
6.8	The convergence in low/high-frequency spectrum of solving an implicit vector diffusion step with various timestep α on the fertility model using the Whitney basis using Gauss-Seidel (solid lines) and SOR (dotted lines) as the relaxation scheme.	94
6.9	Correlation matrices (Eq. 4.3) coarsest-finest visualized as images of all combinations. Black body colormap is used to visualize 0 as black, and 1 as white.	95
6.10	The convergence in low/high-frequency spectrum of solving an implicit vector diffusion step with various timestep α on the fertility model using the Whitney basis with (dotted lines) and without (solid lines) integrating Kyrlov subspace update.	103
6.11	The convergence comparison plots among V-Cycle, F-Cycle, and W-Cycle in error-time plot. The y-axis (the relative residual error) is plotted in the log scale.	107
7.1	Parallel transport results (top and bottom views) of our solver at every five iterations on the fertility model using the SP hierarchy and the Whitney basis, and the relative residual errors of the vector and scalar diffusion solves.	113
7.2	An example of texture grid hierarchy. The bi-linear-like seamless 0-form basis is defined on the teure grid nodes where the texture grid cells overlapped with the texture map of them model.	114
7.3	Logarithmic map results (front and back views, and on the texture) of our solver at every twenty iterations on the hand model the Whitney basis, and the relative residual errors of the vector and scalar diffusion solves. The first row shows the direct solver's result.	116
7.4	Logarithmic map results (front and back views, and on the texture) of our solver at every twenty iterations on the hand model the Whitney basis, and the relative residual errors of the vector and scalar diffusion solves. The first row shows the direct solver's result. (cont.)	117
A.1	The three mesh hierarchies of the plane model.	138
A.2	The three mesh hierarchies of the sphere model.	139
A.3	The three mesh hierarchies of the torus model.	139
A.4	The three mesh hierarchies of the 4-torus model.	140

LIST OF FIGURES

A.5	The three mesh hierarchies of the hand model.	140
A.6	The three mesh hierarchies of the bimba model.	141
A.7	The three mesh hierarchies of the rooster model.	141
A.8	The three mesh hierarchies of the fertility model.	142
B.1	The convergence rate of the standard multigrid method when solving an implicit vector diffusion step with varies timestep α from 0.0001 to 0.01 using the hierarchies mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP), as well as the harmonic-free and Whitney 1-form basis in six hundred iterations.	144
C.1	SOR convergence of solving a vector diffusion problem using the harmonic-free basis with different timestep α	146
C.2	SOR convergence of solving a vector diffusion problem using the Whitney basis with different timestep α	147
C.3	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.0001$ with different SOR relaxation weights.	149
C.4	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.001$ with different SOR relaxation weights.	150
C.5	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.01$ with different SOR relaxation weights.	151
C.6	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.0001$ with different SOR relaxation weights.	152
C.7	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.001$ with different SOR relaxation weights.	153
C.8	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.01$ with different SOR relaxation weights.	154
C.9	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.0001$ with different prolongation smoothing weights.	157
C.10	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.001$ with different prolongation smoothing weights.	158

LIST OF FIGURES

C.11	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.01$ with different prolongation smoothing weights.	159
C.12	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.0001$ with different prolongation smoothing weights.	160
C.13	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.001$ with different prolongation smoothing weights.	161
C.14	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.01$ with different prolongation smoothing weights.	162
C.15	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.0001$ with various Krylov subspace dimensions.	164
C.16	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.001$ with various Krylov subspace dimensions.	165
C.17	Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.01$ with various Krylov subspace dimensions.	166
C.18	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.0001$ with various Krylov subspace dimensions.	167
C.19	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.001$ with various Krylov subspace dimensions.	168
C.20	Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.01$ with various Krylov subspace dimensions.	169

List of Algorithms

1	A Two-Level V-Cycle Multigrid Method	37
2	A V-Cycle Multigrid Method	43
3	A F-Cycle Multigrid Method	44
4	A W-Cycle Multigrid Method	45

Chapter 1

Introduction

Vector, in Latin, meaning *one that carries or transports* [Kev23], was first introduced to mathematics and physics by Oliver Heaviside and Josiah Willard Gibbs near the end of the 19th century when they developed Vector Calculus for vector fields. Due to its simpler and cleaner notation than the preceding Hamilton's Quaternions and Clifford Algebras, vector fields, today, have become the main mathematical object to describe flow-field-related problems such as electromagnetism, fluid simulation, and shape deformation. As a result, vector field processing has come to be an essential task in contemporary science and engineering applications.

1.1 Motivation

Discrete Vector Field Processing

Owing to the importance of vector field processing, extensive work has been done on vector field design in discrete geometry processing, especially for triangulated surfaces. From finite elements approaches, through discrete exterior calculus and finite element exterior calculus, to spectral methods and machine learning techniques, researchers have developed new representations of vector fields on surfaces for specific applications, better accuracy, and efficient computation (reviewed in Chapter 2). Regardless of the choice of representation used, to manipulate and compare vector fields, well-defined differential operators are required. Having these well-defined discrete differential operators enables numerous applications.



Figure 1.1: Example applications of solving the vector diffusion equation as described in the work of Sharp *et al.* [SSC19] - parallel transport (left), logarithmic map (middle), and Karcher mean (right).

CHAPTER 1. INTRODUCTION

For example, one can solve the vector diffusion equation described in the work of Sharp *et al.* [SSC19] to perform parallel transport of a vector and compute the logarithmic map and the Karcher mean as illustrated in Fig. 1.1. This dissertation focuses on vector field manipulation using first-order differential operators. Specifically, we use the exterior derivative and co-derivative to formulate the vector field processing problem in the gradient domain (details in Chapter 3). In order to handle a large class of applications (related to first-order differential operators), we adapt the gradient domain paradigm from scalar field processing to vector field processing on surfaces.

Interactive Gradient Domain Scalar Field Processing



Figure 1.2: McCann and Pollard [MP08] combine the gradient domain paradigm and a hierarchical solver to develop an interactive painting application that allows artists to paint with digital brushes. The above figures show snapshots during a digital painting process.

The gradient domain paradigm has unified scalar field processing on surfaces for various applications such as smoothing, sharpening, and stitching. Several

CHAPTER 1. INTRODUCTION

works have combined this approach with a hierarchical solver to provide interactive applications such as interactive painting (Fig. 1.2) developed by McCann and Pollard [MP08].

Inspired by the significant speedup in scalar field processing using a hierarchical approach, this dissertation demonstrates an efficient hierarchical solver for gradient domain vector field processing in an attempt to produce real-time vector field processing applications. Furthermore, we describe an approach for extending scalar field prolongation matrices to vector fields, alleviating the need for creating hierarchies for vector field processing from scratch and ensuring that the prolongation commutes with fundamental operators like the differential.

1.2 Objective

To the best of our knowledge, this dissertation is the first to look into solving vector field processing using a hierarchical approach and generalize the 1-form prolongation design by leveraging the existing efficient 0-form hierarchical system. The objective of the dissertation is to answer the following questions:

- How do we perform vector field processing in the gradient domain?
- How do we design a hierarchy for vector field processing?
- How do we make the hierarchical solver converge quickly?

1.3 Overview

Chapter 2 begins with a literature review of gradient domain processing, hierarchical solver, and discrete vector field design. Next, Chapter 3 presents the classical formulation of scalar field gradient domain processing on surfaces with Riemannian metrics using the language of exterior calculus. We then present its extension to vector field processing using the exterior derivative and co-derivative.

Chapter 4 describes our hierarchical approach using the multigrid method. We start by considering a two-level system, then describe the three key components as well as speedup techniques in each component – relaxation schemes, smoothed prolongations, and Krylov subspace update method. Then, in Chapter 5, we present our general approach to constructing hierarchical vector field bases given a scalar field basis with a prolongation matrix. The approach defines a vector field prolongation matrix for the multigrid method.

Focusing on a triangle mesh and using existing scalar field hierarchical structures, Chapter 6 extensively studies the convergence of our solver, the effects of different speedup techniques and multigrid cycle types, and compares it to the direct CHOLMOD solver. In Chapter 7, we demonstrate the effectiveness of our approach for problems of logarithmic map computation and its applications to texture grid hierarchy for vector field processing.

CHAPTER 1. INTRODUCTION

We conclude in Chapter 8, summarizing our findings and discussing future work from both the practical and theoretical perspectives. Practically, it will be interesting to leverage the power of GPUs to accelerate computation and implement higher-dimensional k -form processing. Theoretically, we are interested in optimizing the 0-form prolongation coefficients for the best 1-form prolongation.

Chapter 2

Background

Our approach for processing vector fields on surfaces has three essential components – the formulation, the solver, and the vector field representation. First, we formulate vector field processing as a gradient domain problem for handling a wide range of vector field processing applications. Then, we discretize the energy using Galerkin’s formulation, resulting in solving a system of linear equations, which we solve using a hierarchical solver. We need a finite basis for Galerkin’s approach. Discretizing requires choosing a basis; to that, we choose a 1-form basis that is constructed from a 0-form basis. We will describe each component in detail in Chapters 3 to 5. In this chapter, we review the related literature of each component of our approach – *i.e.*, the gradient domain processing, the hierarchical solver, and the discrete vector field representation on surfaces.

2.1 Gradient Domain Processing

Gradient Domain Image Processing

Gradient domain processing is a powerful image editing technique that directly manipulates image gradients rather than image intensities. It has many applications, including image denoising [FLW02], inpainting [SJZW07], and stitching [KH08]. The motivation behind this approach came from the study of rendering high-dynamic range images [DW00]. In 2002, Fattal *et al.* [FLW02] first introduced the concept of using the gradient field to manipulate images. They proposed a variational approach to compress high-dynamic range images by minimizing the difference between the gradient fields of the input and output images. This approach immediately got the attention of the image processing research community. Pérez *et al.* [PGB03] adapted the approach for image editing; Levin *et al.* [LZPW04] used patch-based optimization in the gradient domain for seamless image stitching; Du *et al.* [DQLB05] assumed texture beneath shadow shares similar patterns to non-shadowed one and proposed a shadow removal algorithm by matching gradients to complement the lost radiance in shadow region; Agrawal *et al.* [ARNL05] presented a similar gradient projection scheme that keeps gradients coherent to remove reflections and highlights from flash images. They then extended the approach for edge suppression [ARC06] and advocated it as a general technique for gradient do-

CHAPTER 2. BACKGROUND

main image processing [AR07]. Bhat *et al.* [BCCZ08] analyzed this approach using Fourier analysis and proposed it as an optimization framework for image and video filtering [BZCC10]. In 2016, Shibata *et al.* [STO16] extended the framework for image reconstruction with constraints. Gradient domain image processing is ongoing research. In recent years, it has been combined with deep learning techniques. For example, GFNet [LL22] aimed to extract gradient features to compensate for the missing high-frequency face features in synthesized high-resolution images.

Gradient Domain Signal Processing

This powerful framework has also been applied to applications beyond image processing. For instance, Huang *et al.* [HSL*06] developed a subspace technique to minimize gradient domain energy with non-linear constraints to deform meshes; Lehtinen *et al.* [LKL*13] introduced a novel Metropolis rendering algorithm by sampling in the gradient domain and reconstructing the rendering image using the gradient domain technique and Markus *et al.* [KMA*15] extended this Metropolis approach to Monte Carlo sampling; Chuang *et al.* [CRK16] used it for geometry editing by treating mesh coordinates as input signal; and Prada *et al.* [PKCH18] extended the mesh signal processing framework to texture-domain processing.

Gradient Domain Vector Field Processing

While there is no unified framework for vector field processing, many vector field processing approaches such as [FSDH07], [BSEH17], and [SWJG20] can be expressed as gradient domain problems. Formulating the problem of vector field processing as a gradient domain problem (Chapter 3) allows us to carry over techniques and intuition from signal processing to this more challenging context.

2.2 Hierarchical Solver

Our goal is to perform vector field processing on surfaces efficiently. In general, this reduces to the solution of a large, sparse linear system. To solve a large-scale systems efficiently, one of the most effective ways is to use a hierarchical approach. In practice, multigrid has been the method of choice.

The Multigrid Method

In his early work [Bra77], Brandt introduces the concept of the multigrid as a hierarchical approach for solving elliptic partial differential equations. Subsequent refinement proposed by researchers improved the multigrid method's speed and accuracy, and it has become a standard hierarchical numerical solver as described in several textbooks [McC87, BHM00, BL11].

CHAPTER 2. BACKGROUND

In the geometry processing community, the multigrid method is applied to solve large-scale systems and create interactive applications. For example, [KH08] used the multigrid method and leveraged memory coherency to speed up large image stitching and tone mapping; [MP08] applied the multigrid method to develop interactive digital painting software, and [PKCH18] utilized the multigrid method to enable real-time computation such as geodesic in heat on textured meshes. The community also investigated efficient surface hierarchy for scalar field processing [LZBCJ21]. This dissertation applies the standard multigrid method to vector field processing.

The Multigrid Prolongation

An essential component of the multigrid method is a prolongation, which defines how the coarse level's solution is expressed in the fine level. The design of a good prolongation is an active research topic. Since this dissertation works on vector field processing, our review focuses on multigrid prolongation for vector fields. To the best of our knowledge, there are two methods for designing vector field prolongation. One is the smoothed aggregation used in the algebraic multigrid with edge elements [RS02, NP19]. This method constructs an initial prolongation matrix based on interpolation weights, smooths the matrix to reduce errors, and corrects it to satisfy specific properties (*e.g.*, commutativity between the prolongation and differential operators). This algebraic multigrid

CHAPTER 2. BACKGROUND

technique is a general approach to constructing a prolongation matrix given a connectivity matrix, which can be used to construct vector field prolongation. The other is the Whitney 1-form prolongation, which is implicitly defined once a 0-form prolongation is given (*e.g.*, [BHK20, Sec. 4.6]). In this dissertation, we formulate a general way for extending scalar field prolongation to vector field prolongation.

Speed-up Techniques

To further speed up the multigrid method, researchers have looked into different techniques. The Krylov subspace method is commonly used to update the solution after each iteration of the V-Cycle [ELM96, OW00, WLA*14]. On the other hand, a variety of smoothing techniques have been proposed. These are compared in [ABHT03, SSB15]. Furthermore, Hemker [Hem90] provides a formal study of convergence using Fourier analysis and concludes that necessary condition for an efficient prolongation is that it does not introduce high-frequency components when transferring information from the coarse level to the fine level. As a result, several approaches have been proposed to smooth the prolongation [VMB94, Lot23]. We study and incorporate these techniques into our hierarchical solver design in Chapter 4.

2.3 Vector Field Representations

In order to manipulate vector fields over a discrete surface, we need discrete representation. In their course, de Goes *et al.* [dGDT16] reviewed and summarized vector field design methods on triangle meshes, categorizing them based on the vector fields encoded on the triangle mesh. Concretely, they describe face-based, edge-based, and vertex-based representations.

Face-based representation includes [PP00, TLHD03, War07, DVPSH14, CDS10, SWJG20], in which the discretization guarantees continuity within each discrete face while leaving the continuity at the edges and vertices undefined.

Edge-based representation is continuous within each face and along (but not across) edges. For example, the finite element exterior calculus [AFW06, WWT*06] and discrete exterior calculus [FSDH07, CdGDS13] approaches preserve the continuity of the tangential components of the vector fields along each edge. As pointed out in [dGDT16], an operator approach [GMP*10, ABCCO13] reformulating the Killing operator in terms of the Hodge-Laplacian and using the average Gaussian curvature per edge (Weitzenböck identity) can be used to define a basis that is continuous along edges.

Vertex-based representations aim for continuity at vertices. For instance, Zhang *et al.* [ZMT06] use the polar map at each vertex to interpolate vector fields locally around the vertex, while Knöppel *et al.* [KCPS13] and Liu

CHAPTER 2. BACKGROUND

et al. [LTGD16] define a discrete connection to continuously transport vectors along edges and vertices. The gains in continuity provided by these methods come at the cost of introducing non-linearity.

In this dissertation, we discretize vector fields using a finite 1-form basis that is constructed from a 0-form finite basis. One advantage of this basis is that the 1-form prolongation is induced from the 0-form prolongation by construction. We describe and formulate this approach in Chapter 5.

Chapter 3

Gradient Domain Formulation

In this dissertation, we formulate vector field processing on surfaces using the gradient domain framework. Today, gradient domain processing is a standard framework (as in [BZCC10,STO16,CRK16]) commonly used for many signal processing applications in computer vision and graphics such as smoothing and sharpening [XLG*12, CRK16], content fusion [PGB03, WD11], stitching [LZPW04, KH08], and those mentioned in the previous chapter.

We begin by reviewing the standard framework and its applications to image processing and, more generally, the processing of signals on surfaces. Then, we use the framework to formulate vector field processing in the language of exterior calculus. We show some applications and conclude the chapter by discussing how the gradient domain formulation is discretized using a finite dimensional basis.

3.1 Preliminaries

While gradient domain processing was initially proposed for image processing, it has been widely adapted to signal processing on surfaces (concretely, Riemannian manifolds). Throughout this dissertation, we assume that our processing domain is a Riemannian 2-manifold (\mathcal{M}, g) with a metric tensor g . Frequently used notation is described and defined below.

(Co-)Vector Space

We denote by $\mathbb{T}\mathcal{M}$ the tangent bundle of \mathcal{M} , and at a point $p \in \mathcal{M}$, we denote by $\mathbb{T}_p\mathcal{M}$ the tangent space of \mathcal{M} at p . Similarly, we denote by $\mathbb{T}^*\mathcal{M}$ the co-tangent bundle of \mathcal{M} , and by $\mathbb{T}_p^*\mathcal{M}$ the co-tangent space of \mathcal{M} at p . Furthermore, we denote by g_p the metric at the point p .

Inner Product and Norm of (Co-)Vectors

Recall that the metric g_p is a symmetric positive-definite bilinear map that defines an inner product on $\mathbb{T}_p\mathcal{M}$. It takes a pair of tangent vectors and returns a real value. As such, we can also view it as a map from $\mathbb{T}_p\mathcal{M}$ to $\mathbb{T}_p^*\mathcal{M}$, so that its inverse g_p^{-1} defines an inner product on $\mathbb{T}_p^*\mathcal{M}$.

We denote by $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ the inner product and norm. For $\vec{u}, \vec{v} \in \mathbb{T}_p\mathcal{M}$,

CHAPTER 3. GRADIENT DOMAIN FORMULATION

these are defined as:

$$\langle \vec{u}, \vec{v} \rangle = g_p(\vec{u}, \vec{v}), \quad \|\vec{u}\| = \sqrt{\langle \vec{u}, \vec{u} \rangle}.$$

For $\mu, \nu \in \mathbf{T}_p^* \mathcal{M}$, their inner product and norm are:

$$\langle \mu, \nu \rangle = g_p^{-1}(\mu, \nu), \quad \|\mu\| = \sqrt{\langle \mu, \mu \rangle}.$$

Vector Field and 1-Form Duality

For any given $\vec{u} \in \mathbf{T}_p \mathcal{M}$, there exists a unique dual co-vector $\mu \in \mathbf{T}_p^* \mathcal{M}$ (the converse is also true) such that, for any $\vec{v} \in \mathbf{T}_p \mathcal{M}$, we have:

$$\mu(\vec{v}) = \langle \vec{u}, \vec{v} \rangle.$$

A vector field $\vec{v}(p)$ on \mathcal{M} is a map that assigns every point $p \in \mathcal{M}$ a vector $\vec{v}(p) \in \mathbf{T}_p \mathcal{M}$. Since each vector has a unique dual co-vector, a vector field $\vec{v}(p)$ has a unique co-vector field (a.k.a. 1-form) $\nu(p)$ that assigns every point $p \in \mathcal{M}$ a co-vector $\nu(p) \in \mathbf{T}_p^* \mathcal{M}$ such that $\nu(p)(\cdot) = \langle \vec{v}(p), \cdot \rangle$.

By exploiting this duality, **we describe vector fields using 1-forms.**

CHAPTER 3. GRADIENT DOMAIN FORMULATION

k -Form Function Spaces, Inner Product and Norm of k -Forms

We denote by $\Omega_k(\mathcal{M})$ the space of k -forms, $k = 0, 1, 2$. Notice that a scalar function on \mathcal{M} is a 0-form on \mathcal{M} , a 1-form on \mathcal{M} is the dual of a vector field on \mathcal{M} , and a 2-form on \mathcal{M} can also be thought of as a scalar function on \mathcal{M} .

For any two $\phi, \psi \in \Omega_0(\mathcal{M})$, $\mu, \nu \in \Omega_1(\mathcal{M})$, $\varrho, \vartheta \in \Omega_2(\mathcal{M})$, we denote their inner product and squared norm as:

$$\begin{aligned} \langle \phi, \psi \rangle &= \int_{\mathcal{M}} \langle \phi(p), \psi(p) \rangle dp & \|\phi\|^2 &= \int_{\mathcal{M}} \langle \phi(p), \phi(p) \rangle dp, \\ \langle \mu, \nu \rangle &= \int_{\mathcal{M}} \langle \mu(p), \nu(p) \rangle dp & \|\mu\|^2 &= \int_{\mathcal{M}} \langle \mu(p), \mu(p) \rangle dp, \\ \langle \varrho, \vartheta \rangle &= \int_{\mathcal{M}} \langle \varrho(p), \vartheta(p) \rangle dp & \|\varrho\|^2 &= \int_{\mathcal{M}} \langle \varrho(p), \varrho(p) \rangle dp. \end{aligned}$$

At a point p , for $\phi(p), \psi(p)$, $\langle \cdot, \cdot \rangle$ is the usual scalar product. For $\mu(p), \nu(p)$, $\langle \cdot, \cdot \rangle$ is the co-vector inner product we defined above. For $\varrho(p), \vartheta(p)$ on a 2-manifold \mathcal{M} , we define $\langle \cdot, \cdot \rangle$ as follows. Let $\varrho(p) = \rho(p)dx_p \wedge dy_p$ and $\vartheta(p) = \theta(p)dx_p \wedge dy_p$ expressed in some local co-frame (dx_p, dy_p) . We can express the metric g_p as a matrix \mathbf{g}_p using the co-frame. We define:

$$\langle \varrho(p), \vartheta(p) \rangle = \frac{\rho(p)\theta(p)}{\sqrt{\det \mathbf{g}_p}}.$$

Notice that we refer to these inner products when we describe the signal and vector field processing on \mathcal{M} .

CHAPTER 3. GRADIENT DOMAIN FORMULATION

Derivatives of k -Forms, Hodge Star Operator

$$\begin{array}{ccccc}
 \Omega_0(\mathcal{M}) & \xrightarrow{d} & \Omega_1(\mathcal{M}) & \xrightarrow{d} & \Omega_2(\mathcal{M}) \\
 \uparrow \star & \circlearrowleft & \uparrow \star & \circlearrowleft & \uparrow \star \\
 \Omega_2(\mathcal{M}) & \xrightarrow{\delta} & \Omega_1(\mathcal{M}) & \xrightarrow{\delta} & \Omega_0(\mathcal{M})
 \end{array}$$

Figure 3.1: Relation between d , δ , and \star on a 2-manifold \mathcal{M} .

In order to formulate a notion of geometry processing in the gradient domain, we need derivatives. Given a k -form, one can use the Hodge star operator that takes a k -form and returns a $(2 - k)$ -form. One can differentiate a k -form using the exterior derivative $d : \Omega_k(\mathcal{M}) \rightarrow \Omega_{k+1}(\mathcal{M})$, which returns a $(k + 1)$ -form; and also using the exterior co-derivative $\delta : \Omega_k(\mathcal{M}) \rightarrow \Omega_{k-1}(\mathcal{M})$, which returns a $(k - 1)$ -form. δ is defined as $\delta = (-1)^k \star^{-1} d \star$. Fig. 3.1 shows the relation between d , δ , and \star how they operate on a k -form and return another differential form. The Hodge-Laplacian of a k -form is given by $\Delta = \delta d + d \delta$. Note that, on a 2-manifold, for 0-forms $\Delta = \delta d$, and for 2-forms, $\Delta = d \delta$.

Exterior Calculus Operator Properties

We recall the following properties.

$$d \circ d = 0, \tag{3.1}$$

$$\delta \circ \delta = 0, \tag{3.2}$$

$$\star \delta = (-1)^k d \star, \tag{3.3}$$

CHAPTER 3. GRADIENT DOMAIN FORMULATION

$$\star d = (-1)^{k+1} \delta \star, \quad (3.4)$$

$$\star^{-1} = (-1)^{k(2-k)} \star, \quad (3.5)$$

and for any 1-form $\mu, \nu \in \Omega_1(\mathcal{M})$, k -form $K \in \Omega_k(\mathcal{M})$, and $(k+1)$ -form $P \in \Omega_{k+1}(\mathcal{M})$, we have

$$\langle \mu, \nu \rangle = \langle \star \mu, \star \nu \rangle, \quad (3.6)$$

$$\langle dK, P \rangle = \langle K, \delta P \rangle \quad \text{if } \partial \mathcal{M} = 0. \quad (3.7)$$

Exact and Co-exact k -forms, Hodge Decomposition

When a k -form is the differential of a $(k-1)$ -form, the k -form is called *exact*. Similarly, if a k -form is a co-differential of a $(k+1)$ -form, then the k -form is called *co-exact*. We say a k -form is harmonic, when both its differential and co-differential are zero. Any k -form can be orthogonally decomposed into its exact, co-exact, and harmonic component, known as the Hodge decomposition [RdL16]:

$$\Omega_k(\mathcal{M}) \equiv d(\Omega_{k-1}(\mathcal{M})) \oplus \delta(\Omega_{k+1}(\mathcal{M})) \oplus \mathcal{H}^k(\mathcal{M}), \quad (3.8)$$

where $\mathcal{H}^k(\mathcal{M})$ is the space of harmonic k -forms.

Further details on Riemannian geometry and exterior calculus can be found in Petersen's textbook [Pet06] and Burton's lecture notes [Bur03].

3.2 Signal Processing

3.2.1 The Gradient Domain Framework

In the context of a Riemannian 2-manifold (\mathcal{M}, g) , gradient domain processing of signals (0-forms) can be formulated as follows [PKCH18, Eq. 1]. We are given a target signal $\psi : \mathcal{M} \rightarrow \mathbb{R}$ and a target 1-form $\nu : \mathcal{M} \rightarrow \mathbb{T}^* \mathcal{M}$. The goal is to reconstruct a new signal $\phi : \mathcal{M} \rightarrow \mathbb{R}$ such that, simultaneously, it is similar to ψ and its differential $d\phi$ is similar to ν . To solve for ϕ , one minimizes the energy:

$$E(\phi) = \alpha \|\phi - \psi\|^2 + \beta \|d\phi - \nu\|^2. \quad (3.9)$$

The first summand is referred to as the screening term, which encourages similarity between the output and a prescribed input signal, while the second is the differential term which measures the extent to which the differential of the output matches the prescribed input 1-form. The parameters α and β balance between the two.

3.2.2 Examples

While Eq. 3.9 is used for numerous applications, the key to applying this framework is describing the desired signal modification in the gradient domain by a target 1-form ν . In many applications, we either begin with a single

CHAPTER 3. GRADIENT DOMAIN FORMULATION

target signal ψ and construct a target 1-form ν from $d\psi$ for making the desired changes, or we start with multiple target signals ψ_i , and create a target 1-form ν for fusing the ψ_i .

3.2.2.1 Signal Modification

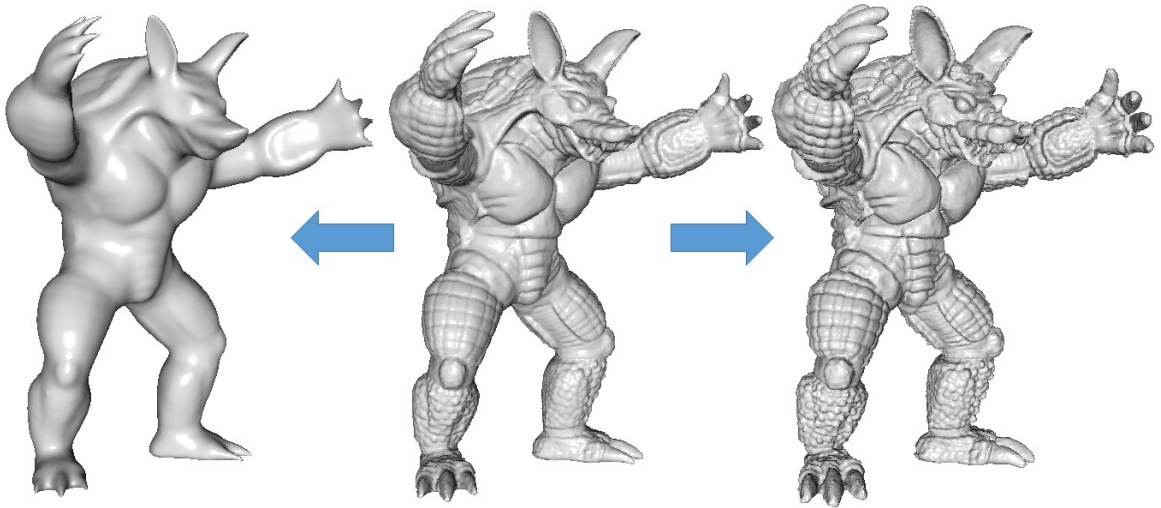


Figure 3.2: Smoothing (left) with $\lambda = 0$ and sharpening (right) with $\lambda = 2$ using gradient domain signal processing, where the signal is the mesh coordinate function. Image courtesy of [CRK16].

In applications such as smoothing and sharpening [CRK16]. One can generate a target 1-form ν by modulating (i.e. dampening or amplifying) the differential of the input:

$$\nu = \lambda d\psi.$$

Picking $0 \leq \lambda < 1$, Eq. 3.9 defines a smoothing energy. Similarly, for $\lambda > 1$, Eq. 3.9 defines sharpening energy. Fig. 3.2 shows the results from Chuang

CHAPTER 3. GRADIENT DOMAIN FORMULATION

et al. [CRK16] of smoothing and sharpening a mesh using $\lambda = 0$ and $\lambda = 2$. We recall that when $\lambda = 0$ and $\alpha = 1$, minimizing Eq. 3.9 is equivalent to diffusing the signal by performing an implicit step with step-size β .

Instead of a constant factor λ , one can create a target 1-form by multiplying the $d\psi$ with a spatially varying modulation function, as in high dynamic range compression [FLW02]. Fig. 3.3 shows results using this approach to improve the contrast of an image.



Figure 3.3: The contrast of the input image (left) is enhanced using gradient domain signal processing with the target 1-form proposed in [FLW02]. Image courtesy of [FLW02].

3.2.2.2 Signal Interpolation

The framework also allows interesting signal interpolation applications such as in-painting [SJZW07], shown in Fig. 3.4.

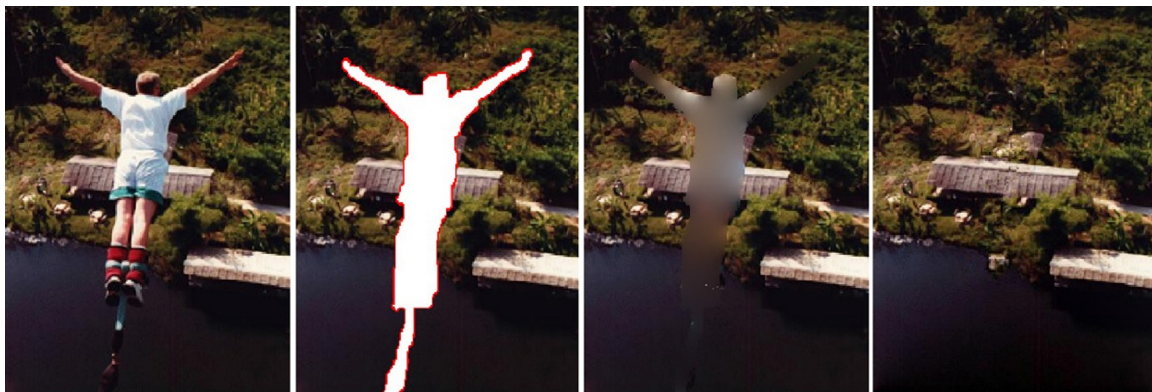


Figure 3.4: Shen *et al.* [SJZW07] use the gradient domain framework to remove the jumper from the image and fill in the content automatically. A user removes the jumper manually from an image (left-most), resulting in an input to the algorithm (second left), an image with a missing region. Then the algorithm fills the missing region iteratively, as illustrated by the last two images. Image courtesy of [SJZW07].

In in-painting, we are given an input image in which part of the content is missing (Fig. 3.4 second image). The goal is to fill in the missing part. Using the gradient domain framework, Shen *et al.* [SJZW07] create the target 1-form within the missing region by filling it iteratively from the boundary with the most similar image differential found in the input. Then, they solve for the 0-form (colors) that best fits the in-painted 1-form. The result is an image that seamlessly fills in the image using similar parts from the input (Fig. 3.4 last image).

3.2.2.3 Signal Fusion

Another common type of application is signal fusion, such as image stitching [LZPW04, KH08], seamless cloning [PGB03], and content fusion [WD11].

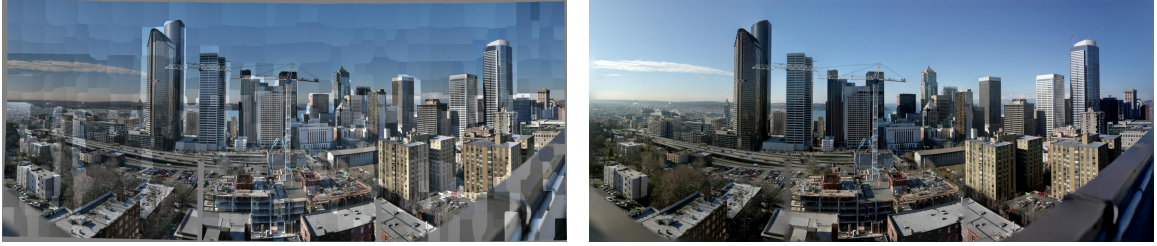


Figure 3.5: The input image (left) is fused seamlessly together to produce an image panoramic (right) by Kazhdan *et al.* [KH08]. Image courtesy of [KH08].

These applications begin with multiple target signals $\{\psi_i : \mathcal{M}_i \rightarrow \mathbb{R}\}$, where $\mathcal{M}_i \subset \mathcal{M}$ and $\mathcal{M}_i \cap \mathcal{M}_j = \emptyset$ for $i \neq j$. As in the previous examples, to use the gradient domain framework, one needs a target signal and a target 1-form. For image stitching, Levin *et al.* [LZPW04] combine $\{\psi_i\}$ to form a single target signal and construct a target 1-form from $\{d\psi_i\}$ by fusing the differentials and setting the differential across the boundaries of \mathcal{M}_i to 0, *i.e.*, encouraging the signal to be continuous across the seams. By solving for the best fit image, Kazhdan *et al.* [KH08] obtain a seamless image panoramic, as shown in Fig. 3.5.

3.2.3 A Fourier Perspective

How does gradient domain processing work, and why is it good for signal processing? Bhat *et al.* [BCCZ08] describe how gradient domain processing works using the frequency domain perspective. Concretely, when the target 1-form is a scalar multiple of the differential of another target signal (*i.e.*, $\nu = d\xi$,

CHAPTER 3. GRADIENT DOMAIN FORMULATION

$\xi \in \Omega_0(\mathcal{M})$, solving Eq. 3.9 is equivalent to blending the low-frequency components of ψ and the high-frequency components of ξ . That is, gradient domain processing creates a new signal by fusing the low/high-frequency components of two target signals.

This description continues to hold even when the gradient constraints are explicitly given by a 1-form μ , rather than implicitly as the differential of a 0-form. By applying the Hodge decomposition to a 1-form μ and denoting by $d\xi$ the exact component, minimizing Eq. 3.9 with $\nu = \mu$ is the same as minimizing it with $\nu = d\xi$ due to the orthogonality of the Hodge decomposition. As a result, one can interpret solving Eq. 3.9 as performing two steps. The first projects ν onto the exact 1-form space to obtain a solution whose differential is the exact part of ν , and the second performs the low/high-frequency component fusion described by Bhat *et al.* [BCCZ08].

3.3 Vector Fields Processing

Gradient domain processing is a powerful framework that can be applied to many applications. In this dissertation, we use it to formulate vector field processing. Recall that we describe vector fields using 1-forms. Next, we will use d and δ to formulate gradient domain vector field processing.

3.3.1 The Gradient Domain Formulation

We are given a target 0-form φ , a target 1-form μ , and a target 2-form ϱ . As in signal processing (Eq. 3.9), the goal of gradient domain vector field processing is to construct a new 1-form ω such that its differential (*i.e.*, curl) $d\omega$ matches ϱ , its co-differential (*i.e.*, divergence) $\delta\omega$ matches φ , and is itself similar to μ . We can solve for ω by minimizing the energy:

$$E(\omega) = \alpha \|\omega - \mu\|^2 + \beta (\|d\omega - \varrho\|^2 + \|\delta\omega - \varphi\|^2). \quad (3.10)$$

The screening term (first term) promotes similarity between the result and the target 1-form. The two differential terms (second two terms) measure the extent to which the curl and divergence of the solution fit the prescribed curl and divergence. The parameters α, β balance between the two. Using this formulation, we can modify/interpolate/stitch vector fields by prescribing their curl and divergence.

Remarks

Another approach for formulating vector field processing in the gradient domain would be to use the covariant derivative and a target tensor field. The difference between tensor calculus and exterior calculus approaches are discussed by de Goes *et al.* [dGDT16] in their SIGGRAPH course. As described

CHAPTER 3. GRADIENT DOMAIN FORMULATION

by Knöppel *et al.* [KCPS13], the tensor approach includes both an *antiholomorphic energy* and a *holomorphic energy*, while the exterior approach only has the antiholomorphic energy. The antiholomorphic energy is an energy that measures the curl and divergence of a vector field, which we use in our energy for obtaining smooth vector fields, while the holomorphic energy measures other vector field quantities that we do not consider. The holomorphic energy is used as a conformal energy for surface parameterization [DMA02,LPRM02] and also for solving killing vector fields (which are both holomorphic and divergence-free) as in [BCBSG10, Eq. 8].

3.3.2 Applications

Like signal processing, Eq. 3.10 has many applications for vector field processing. Here, the main challenge is how to describe the desired vector field by a 2-form ϱ and a 0-form φ . Below we briefly describe a few applications of gradient domain vector field processing. In Chapter 7, we demonstrate additional applications in more detail.

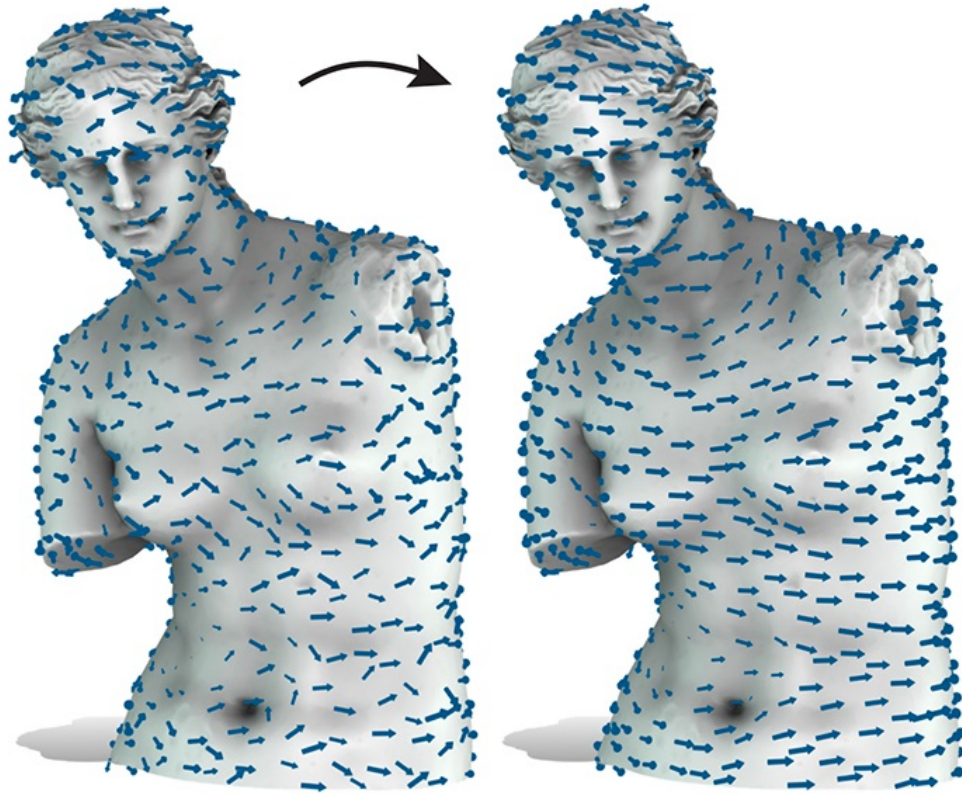


Figure 3.6: A vector field denoising result using Eq. 3.10. Image courtesy of [SWJG20].

3.3.2.1 Vector Field Modification

As in signal processing, if we generate the target 2-form and 0-form by modulating the differential and co-differential of the input 1-form:

$$\varrho = \lambda d\mu, \quad \varphi = \lambda \delta\mu.$$

Eq. 3.10 describes a smoothing/sharpening energy. When $\lambda = 0$ and $\alpha = 1$, minimizing Eq. 3.10 is the same as diffusing the vector field with an implicit

CHAPTER 3. GRADIENT DOMAIN FORMULATION

step of size β , a technique often used for denoising. Notice that since our energy only involves the antiholomorphic term, we perform a diffusion step with the Hodge-Laplacian, while using the energy with the holomorphic term would amount to performing a diffusion step with the Bochner-Laplacian.

Fig. 3.6 shows a denoising result from [SWJG20] using Eq. 3.10 as a diffusion energy. Note how the flow of the output vector field (right) matches the general flow of the input while removing the finer-grained perturbations.

3.3.2.2 Vector Field Interpolation

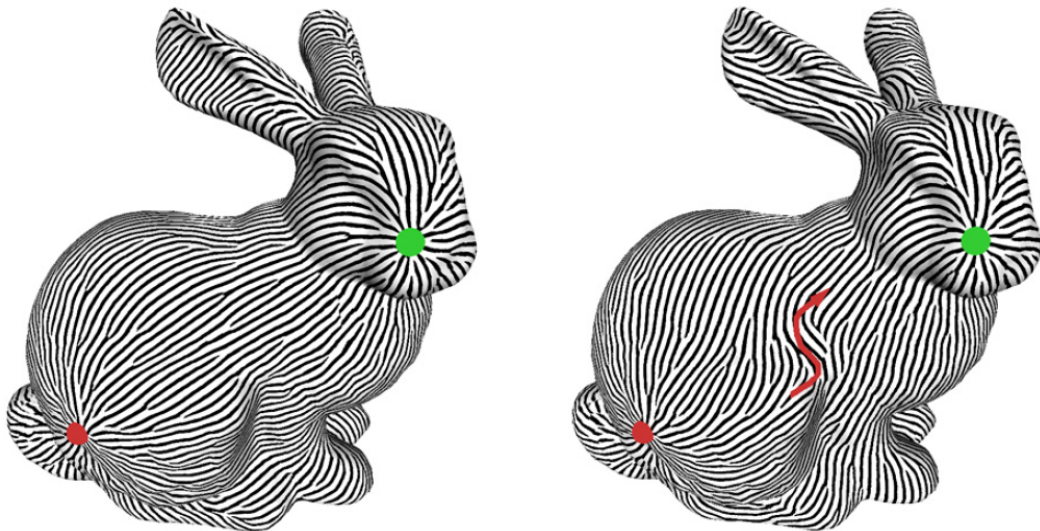


Figure 3.7: Interpolation results from [FSDH07] with source/sink constraints (left) and an additional curve constraint (right). Image courtesy of [FSDH07].

Eq. 3.10 (with a slight modification) also allows us to interpolate vector fields given a set of constraints. In vector field design such as [FSDH07], users can specify point constraints (specifying vortices and/or sources and sinks) or

CHAPTER 3. GRADIENT DOMAIN FORMULATION

curve constraints (prescribing directions) to create a smooth vector field on a surface as shown in Fig. 3.7. Notice that vortices can be prescribed by having the target 2-form φ (the curl) to be zero everywhere except at the vortices, while the sources and sinks can be prescribed by having the target 0-form ϱ (the divergence) to be zero everywhere except at the sources and sinks. However, supporting curve constraints requires modifying our formulation as the screening term constrains the vector field everywhere, not just at the curve positions. To support this, the gradient formulation needs to be extended to allow a spatially varying screening weight. (Concretely, α should be large along the curve and zero away from it.) Such a spatially adaptive screening weight was already considered for gradient domain signal processing on surfaces. (For example, Chuang *et al.* [CRK16] show how such an approach can be used to achieve feature aware smoothing.)

3.3.2.3 Other Applications

Since the computation of parallel transport, the logarithmic map, and the Karcher mean can be formulated in terms of vector field diffusion [SSC19], these, too, can be thought of as applications of gradient domain processing. However, the Bochner-Laplacian tends to be preferred for these applications. As described in [SSC19], an approximated curvature matrix can be used to correct the result (as the Bochner-Laplacian and Hodge-Laplacian differ by a

curvature term.)

3.3.3 A Spectral Perspective

The analysis in [BCCZ08] for Eq. 3.9 carries over to Eq. 3.10. Suppose the target 2-form ϱ is exact, i.e., we can write $\varrho = d\rho, \rho \in \Omega_1(\mathcal{M})$, and the target 0-form φ is co-exact, i.e., we can write $\varphi = \delta\nu, \nu \in \Omega_1(\mathcal{M})$. Then, solving Eq. 3.10 is equivalent to blending the high-frequency components of ρ and φ with the low frequency components of ν .

3.4 Discretization

Using the Euler-Lagrange formulation, the minimizer of Eq. 3.10 satisfies the following equation:

$$(1 + \alpha(\delta d + d\delta))\omega = \mu + \alpha(\delta\varrho + d\varphi). \quad (3.11)$$

To solve Eq. 3.11 in practice, we discretize using a finite basis $\mathcal{W} = \{\omega_i\}$ that spans a subspace of 1-forms on \mathcal{M} . Representing ω and μ in this subspace by $\omega = \sum_i w_i \omega_i, \mu = \sum_i u_i \omega_i$, where $\mathbf{w} = [w_i], \mathbf{u} = [u_i]$ are the coefficients of the 1-forms in the basis \mathcal{W} . Given the basis, we define the mass matrix \mathbf{M} , stiffness

CHAPTER 3. GRADIENT DOMAIN FORMULATION

matrix \mathbf{S} , and the load vector \mathbf{l} , whose entries are:

$$\mathbf{M}_{ij} = \int_{\mathcal{M}} \langle \omega_i, \omega_j \rangle dp, \quad \mathbf{S}_{ij} = \int_{\mathcal{M}} \langle d\omega_i, d\omega_j \rangle + \langle \delta\omega_i, \delta\omega_j \rangle dp,$$

$$\mathbf{l}_i = \int_{\mathcal{M}} \langle \delta\rho, \omega_i \rangle + \langle d\varphi, \omega_i \rangle dp,$$

and, using the weak formulation, the discretization of Eq. 3.11 in basis \mathcal{W} becomes:

$$(\mathbf{M} + \alpha\mathbf{S}) \mathbf{w} = \mathbf{M}\mathbf{u} + \alpha\mathbf{l}. \quad (3.12)$$

3.5 Summary

In this chapter, we described how to extend the gradient domain scalar field processing to vector field processing using the language of exterior calculus. We noted that this formulation only involves the antiholomorphic energy (*i.e.*, curl and divergence), and when solving for diffusion equation, this formulation is equivalent to using the Hodge-Laplacian. In contrast, the tensor calculus approach, which involves the holomorphic energy, describes diffusion using the Bochner-Laplacian. We showed some examples of applications of this formulation and noted that the spectral perspective interpretation of 0-form (scalar field) processing carries over to the 1-form (vector field) processing formulation. We concluded by discretizing the energy using a finite element basis.

Chapter 4

A Hierarchical Approach

How to efficiently solve a system of linear equations like Eq. 3.12 has been actively researched in numerical computation. In this context, the multigrid method is a successful and popular algorithm that speeds up the processing time to achieve real-time interaction. For example, both artistic painting [MP08] and texture domain processing [PKCH18] utilize the multigrid method to develop interactive signal processing applications. In this chapter, we first review a two-level V-Cycle multigrid method. Then, we discuss the key components of the multigrid method and techniques for accelerating each component. Finally, we present the multi-level multigrid method.

4.1 A Two-Level V-Cycle Multigrid

Recall that we discretize Eq. 3.10 using a finite basis \mathcal{W} , where we now denote by n the number of basis functions in \mathcal{W} . We rewrite Eq. 3.12 and denote it as the fine level linear system $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{A} = \mathbf{M} + \alpha \mathbf{S}, \quad \mathbf{b} = \mathbf{M}\mathbf{u} + \alpha \mathbf{l}, \quad \mathbf{x} = \mathbf{w}.$$

The multigrid method solves this fine level linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ using a coarsening hierarchy of discretizations (in this section, we look at a two-level hierarchy). The coarse level linear system $\hat{\mathbf{A}} \in \mathbb{R}^{\hat{n} \times \hat{n}}$, $\hat{\mathbf{b}} \in \mathbb{R}^{\hat{n}}$ is defined by a full rank prolongation matrix $\mathbf{P} \in \mathbb{R}^{n \times \hat{n}}$, where $\hat{n} < n$:

$$\hat{\mathbf{A}} = \mathbf{P}^\top \mathbf{A} \mathbf{P}, \quad \hat{\mathbf{b}} = \mathbf{P}^\top (\mathbf{b} - \mathbf{A}\mathbf{x}).$$

We will describe how to define \mathbf{P} for solving Eq. 3.12 later in Chapter 5. The central idea of the multigrid method can be thought of as efficiently solving the low-frequency (respectively high-frequency) components of the fine system at the coarse (respectively fine) level – solving the high-frequency part at the fine level using a fast relaxation scheme and solving the low-frequency part at the coarse level using a direct solver. Fig. 4.1 illustrates a two-level V-Cycle multigrid method. A V-Cycle consists of a restriction phase (R-i and R-ii), a coarse

CHAPTER 4. HIERARCHICAL APPROACH

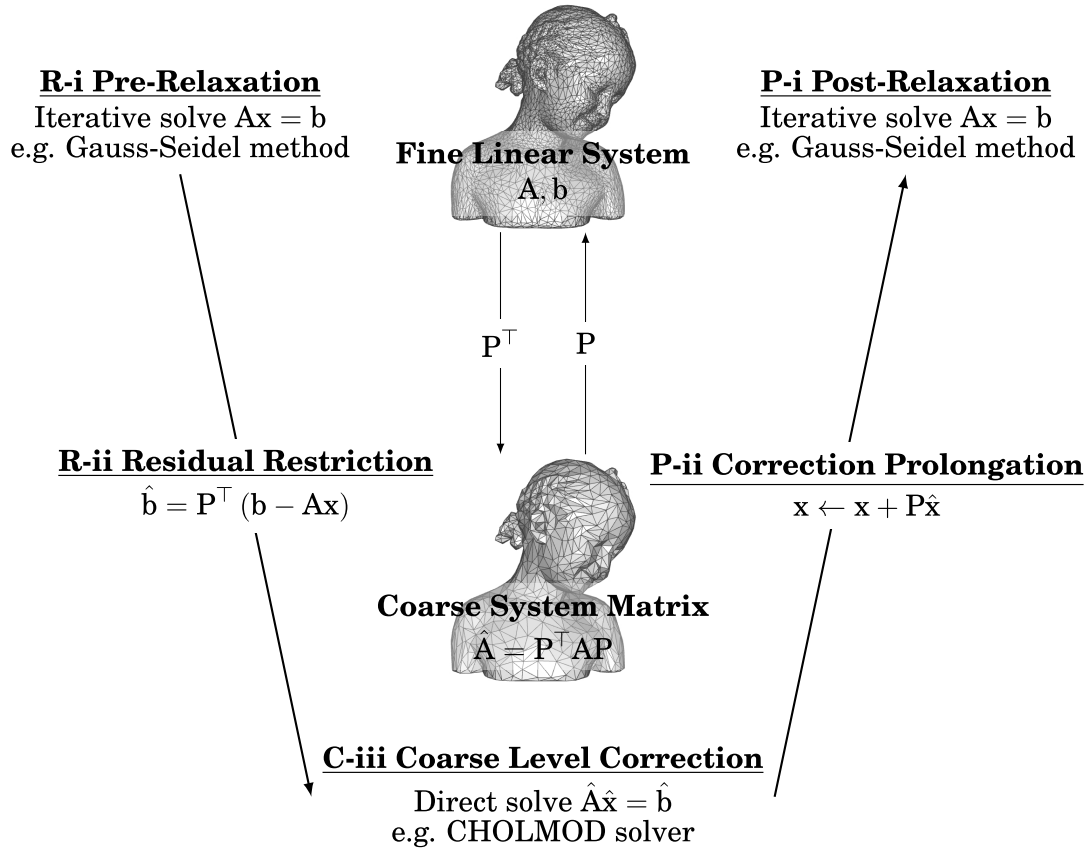


Figure 4.1: A two-level V-Cycle multigrid method composed of the three key components: i) a relaxation scheme, ii) a restriction/prolongation, and iii) a coarse level correction solve.

level correction solve (C-iii), and a prolongation phase (P-i and P-ii). Given a current estimate x , during the restriction phase, R-i refines x using a fast relaxation scheme, then R-ii restricts the residual $(b - Ax)$ to the coarse system using the restriction matrix P^T . At the coarse level, C-iii solves for the correction \hat{x} using the coarse system matrix \hat{A} and the restricted residual \hat{b} . During the prolongation phase, P-ii interpolates the coarse correction \hat{x} to the fine level and uses it to correct the current estimate x , then P-i further refines x using

CHAPTER 4. HIERARCHICAL APPROACH

the fast relaxation scheme. The two-level V-Cycle multigrid is summarized in Algorithm 1. In practice, one can recursively use another V-Cycle for C-iii until the system is small enough for a quick direct solve, resulting in efficient multi-level V-Cycle multigrid methods.

Algorithm 1 A Two-Level V-Cycle Multigrid Method

1: **Given**
2: A The fine level system matrix
3: b The fine level right hand side
4: P The prolongation matrix
5: **Compute**
6: $\hat{A} = P^T A P$

7: **Input/Output**
8: x The current estimate

9: **Restriction**
10: R-i $x \leftarrow \text{PreRelaxation}(A, x, b)$
11: R-ii $\hat{b} = P^T (b - Ax)$
12: **Correction**
13: C-iii $\hat{x} = \text{CorrectionSolve}(\hat{A}, \hat{b})$
14: **Prolongation**
15: P-ii $x \leftarrow x + P\hat{x}$
16: P-i $x \leftarrow \text{PostRelaxation}(A, x, b)$

4.2 Key Components

The multigrid method has three components:

1. A **relaxation scheme** that (iteratively) refines the current estimate.
2. A **prolongation matrix** that describes the coarse basis vectors as linear combination of the finer ones. It has three functions:

CHAPTER 4. HIERARCHICAL APPROACH

- It restricts the fine level residual to the coarse level,
- It defines the coarse system matrix, and
- It prolongs the coarse level correction to the fine level.

3. A **coarse level correction** that (directly) solves the restricted residual.

4.2.1 Relaxation Scheme

An efficient relaxation scheme should be fast. Following the spectral analysis approach in [BL11, Eq. 3.1], we analyze the scheme by considering its convergence in the low/high-frequency spectrum. Given a system matrix $\mathbf{A} = \mathbf{M} + \alpha\mathbf{S}$, a spectral decomposition $\mathbf{e}_{i=1}^n$. We express the ground-truth \mathbf{x}^* and the current estimate \mathbf{x} in the spectral basis:

$$\mathbf{x}^* = \sum_i x_i^* \mathbf{e}_i, \quad \mathbf{x} = \sum_i x_i \mathbf{e}_i.$$

We define the low/high-frequency relative residual errors as:

$$E_l(\mathbf{x}) = \sum_{i=1}^k (x_i^* - x_i)^2 \frac{\mathbf{e}_i^\top \mathbf{A} \mathbf{e}_i}{(\mathbf{x})^\top \mathbf{A} \mathbf{x}}, \quad E_h(\mathbf{x}) = \sum_{i=k+1}^n (x_i^* - x_i)^2 \frac{\mathbf{e}_i^\top \mathbf{A} \mathbf{e}_i}{(\mathbf{x})^\top \mathbf{A} \mathbf{x}}. \quad (4.1)$$

In this dissertation, we study the common relaxation schemes using the split methods: Jacobi, damped Jacobi, Gauss-Seidel, and successive over-relaxation (SOR). Given a symmetric square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with entries A_{ij} , a right

CHAPTER 4. HIERARCHICAL APPROACH

hand side $\mathbf{b} \in \mathbb{R}^n$ and a current estimate $\mathbf{x} \in \mathbb{R}^n$. We split the system matrix \mathbf{A} into its diagonal components \mathbf{D} and lower triangular components \mathbf{L} :

$$\mathbf{D} = \begin{bmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{nn} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ A_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & 0 \end{bmatrix}.$$

(Note that since \mathbf{A} is symmetric, $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{L}^\top$.) These common relaxation schemes have the same iterative update step:

$$\mathbf{x} \leftarrow \mathbf{N}^{-1}(\mathbf{b} - (\mathbf{A} - \mathbf{N})\mathbf{x}), \quad (4.2)$$

where $\mathbf{N} \in \mathbb{R}^{n \times n}$ is:

Damped Jacobi : $\mathbf{N} = \frac{1}{\omega_{DJ}}\mathbf{D}$, where $\omega_{DJ} \neq 0$.

[when $\omega_{DJ} = 1$, it is the standard **Jacobi**]

SOR : $\mathbf{N} = \mathbf{L} + \frac{1}{\omega_{SOR}}\mathbf{D}$, where $\omega_{SOR} \neq 0$.

[when $\omega_{SOR} = 1$, **SOR** is known as **Gauss-Seidel**]

In addition, we also study an exact method: the conjugate gradient method.

4.2.2 The Prolongation Matrix

The prolongation matrix is crucial to the multigrid method as it defines the coarse resolution systems and defines how constraints and solutions are shared between levels of the hierarchy. A good prolongation matrix should define a coarse basis that is as low-frequency as possible. Using spectral analysis, we analyze how well this prolonged solution approximates the fine system solution in its low frequency. We compute both the first k smallest eigenvectors \mathbf{v}_i and \mathbf{u}_j of the generalized eigenvalue problem $\mathbf{S}\mathbf{v}_i = \lambda_i\mathbf{M}\mathbf{v}_i$ for the fine system and $\mathbf{P}^\top\mathbf{S}\mathbf{u}_j = \gamma_j\mathbf{P}^\top\mathbf{M}\mathbf{u}_j$ for the prolonged coarse system¹. Then, we can compute the eigenvector-correlation matrix \mathbf{C} , whose entries are:

$$C_{ij} = \mathbf{v}_i\mathbf{M}\mathbf{u}_j. \quad (4.3)$$

The closer \mathbf{C} is to a block identity matrix is, the lower-frequency the prolonged coarse basis is. This, in turn, implies a more efficient multigrid.

In this dissertation, we study the prolongation smoothing borrowed from algebraic multigrid for speed-up. Given a prolongation matrix \mathbf{P}_{l+1}^l that prolongs from level $l+1$ to level l , where level $l+1$ is coarser than level l , we smooth the

¹Note that: we solve for $\mathbf{P}^\top\mathbf{S}\tilde{\mathbf{P}}\tilde{\mathbf{u}}_j = \gamma_j\mathbf{P}^\top\mathbf{M}\tilde{\mathbf{P}}\tilde{\mathbf{u}}_j$ at the coarser level, then we prolong it back $\mathbf{u}_j = \tilde{\mathbf{P}}\tilde{\mathbf{u}}_j$, giving us $\mathbf{P}^\top\mathbf{S}\mathbf{u}_j = \gamma_j\mathbf{P}^\top\mathbf{M}\mathbf{u}_j$.

CHAPTER 4. HIERARCHICAL APPROACH

prolongation matrix by setting

$$\mathbf{P}_{l+1}^l \leftarrow \mathbf{S}_l \mathbf{P}_{l+1}^l, \quad (4.4)$$

where the \mathbf{S}_l is the damped Jacobi smoother defined in [VMB94, Eq. 4.6]. In addition, to maintain the sparsity, instead of using the strongly-coupled neighborhood in [VMB94, Eq. 4.2], we discard the entries of the smoothed prolongations where their original value is zero before smoothing.

4.2.3 Coarse Level Correction

Complementing the relaxation scheme, the coarser level correction solves the residual problem. The coarse level correction should be computed quickly. When the dimension of the coarse system is small enough, we can use a direct solver such as CHOLMOD solver. Otherwise, we can recursively apply the algorithm, resulting in a multi-level multigrid method as described later in Section 4.3. Similar to the relaxation scheme, we can analyze the effectiveness of the coarse level correction using spectral analysis (Eq. 4.1).

To further accelerate the convergence, we integrate a low-cost solution update method inspired by the Krylov subspace method.

CHAPTER 4. HIERARCHICAL APPROACH

Krylov Subspace Updates

Given a system matrix A and a right hand side b , the *Krylov subspace* is defined as $\{b, Ab, A^2b, \dots\}$. Oosterlee *et al.* [OW00] showed that the estimate after each V-Cycle of multigrid forms a Krylov subspace. We define the Krylov subspace as $\mathcal{K} = \text{span}(\{\mathbf{x}^{(i)}\})$, where $\mathbf{x}^{(i)}$ is the estimated solution after the i -th V-Cycle. Given an exact solution \mathbf{x}^* such that $A\mathbf{x}^* = b$, and \mathcal{K} at the end of each V-Cycle, we solve for the best solution $\mathbf{x} \in \mathcal{K}$ minimizing:

$$E(\mathbf{x}) = (\mathbf{x} - \mathbf{x}^*)^\top A (\mathbf{x} - \mathbf{x}^*).$$

Let β be the coefficients of \mathbf{x} in \mathcal{K} , i.e., $\mathbf{x} = \sum_i \beta_i \mathbf{x}^{(i)}$. Then, minimizing the above energy is equivalent to solve:

$$\tilde{A}\beta = \tilde{b}, \tag{4.5}$$

where the entries of \tilde{A} and of \tilde{b} , denoted by \tilde{A}_{ij} and \tilde{b}_i are:

$$\tilde{A}_{ij} = (\mathbf{x}^{(i)})^\top A \mathbf{x}^{(j)}, \quad \tilde{b}_i = b^\top \mathbf{x}^{(i)}.$$

Using the solution of Eq. 4.5, we update the solution at the end of the i -th V/F/W-Cycle: $\mathbf{x}^{(i)} = \mathbf{x}$. We denote this step by KS.

4.3 Multigrid

In this section, we present general multigrid algorithms using V-Cycle (Algorithm 2), F-Cycle (Algorithm 3), and W-Cycle (Algorithm 4) solvers with integrated speedup techniques. We assume a coarsening hierarchy from level 1 (the finest) to L (the coarsest) and denote by \mathbf{P}_{l+1}^l the prolongation matrix from $l + 1$ level to l level. The system matrix at level l is defined by $\mathbf{A}_l = (\mathbf{P}_l^1)^\top \mathbf{A}_1 \mathbf{P}_l^1$ where $\mathbf{P}_l^1 = \mathbf{P}_2^1 \cdots \mathbf{P}_l^{l-1}$, $\mathbf{P}_1^1 = \mathbf{I}$, $1 \leq l \leq L$.

V-Cycle

Algorithm 2 A V-Cycle Multigrid Method

- 1: **Given**
- 2: $\mathbf{A}_1, \mathbf{b}_1$ The finest level system matrix and right hand side
- 3: \mathbf{P}_{l+1}^l The prolongation matrices, $l = 1, 2, \dots, L - 1$
- 4: **Compute**
- 5: $\mathbf{A}_l = (\mathbf{P}_l^1)^\top \mathbf{A}_1 \mathbf{P}_l^1, l = 2, \dots, L$

- 6: **Input/Ouput**
- 7: \mathbf{x}_1 The current/updated estimate

- 8: **VCycle**($l = 1$)
- 9: **Restriction**
- 10: R-i $\mathbf{x}_l \leftarrow \text{PreRelaxation}(\mathbf{A}_l, \mathbf{x}_l, \mathbf{b}_l)$ (Eq. 4.2)
- 11: R-ii $\mathbf{b}_{l+1} = (\mathbf{P}_{l+1}^l)^\top (\mathbf{b}_l - \mathbf{A}_l \mathbf{x}_l)$
- 12: **Correction**
- 13: **if** ($l + 1 == L$) **then**
- 14: C-iii $\mathbf{x}_{l+1} = \text{CorrectionSolve}(\mathbf{A}_{l+1}, \mathbf{b}_{l+1})$
- 15: **else**
- 16: C-iii **VCycle**($l + 1$)
- 17: **end if**
- 18: **Prolongation**
- 19: P-ii $\mathbf{x}_l \leftarrow \mathbf{x}_l + \mathbf{P}_{l+1}^l \mathbf{x}_{l+1}$
- 20: P-i $\mathbf{x}_l \leftarrow \text{PostRelaxation}(\mathbf{A}_l, \mathbf{x}_l, \mathbf{b}_l)$ (Eq. 4.2)
- 21: KS $\mathbf{x}_1 \leftarrow \text{KrylovSubspaceUpdate}$ (Eq. 4.5)

CHAPTER 4. HIERARCHICAL APPROACH

F-Cycle

Algorithm 3 A F-Cycle Multigrid Method

```

1: Given
2:    $A_1, b_1$    The finest level system matrix and right hand side
3:    $P_{l+1}^l$    The prolongation matrices,  $l = 1, 2, \dots, L - 1$ 
4: Compute
5:    $A_l = (P_l^1)^\top A_1 P_l^1, l = 2, \dots, L$ 

```

```

6: Input/Output
7:    $x_1$        The current/updated estimate

```

```

8: FCycle( $l = 1$ )
9:   Restriction
10:    R-i    $x_l \leftarrow \text{PreRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
11:    R-ii   $b_{l+1} = (P_{l+1}^l)^\top (b_l - A_l x_l)$ 
12:  Correction
13:    if ( $l + 1 == L$ ) then
14:      C-iii  $x_{l+1} = \text{CorrectionSolve}(A_{l+1}, b_{l+1})$ 
15:    else
16:      C-iii FCycle( $l + 1$ )
17:    end if
18:  Prolongation
19:    P-ii    $x_l \leftarrow x_l + P_{l+1}^l x_{l+1}$ 
20:    P-i     $x_l \leftarrow \text{PostRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
21:  Restriction
22:    R-i    $x_l \leftarrow \text{PreRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
23:    R-ii   $b_{l+1} = (P_{l+1}^l)^\top (b_l - A_l x_l)$ 
24:  Correction
25:    if ( $l + 1 == L$ ) then
26:      C-iii  $x_{l+1} = \text{CorrectionSolve}(A_{l+1}, b_{l+1})$ 
27:    else
28:      C-iii VCycle( $l + 1$ ) (Algorithm 2)
29:    end if
30:  Prolongation
31:    P-ii    $x_l \leftarrow x_l + P_{l+1}^l x_{l+1}$ 
32:    P-i     $x_l \leftarrow \text{PostRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
33: KS     $x_1 \leftarrow \text{KrylovSubspaceUpdate}$  (Eq. 4.5)

```

CHAPTER 4. HIERARCHICAL APPROACH

W-Cycle

Algorithm 4 A W-Cycle Multigrid Method

```

1: Given
2:    $A_1, b_1$    The finest level system matrix and right hand side
3:    $P_{l+1}^l$    The prolongation matrices,  $l = 1, 2, \dots, L - 1$ 
4: Compute
5:    $A_l = (P_l^1)^\top A_1 P_l^1, l = 2, \dots, L$ 

```

```

6: Input/Output
7:    $x_1$        The current/updated estimate

```

```

8: WCycle( $l = 1$ )
9:   Restriction
10:    R-i     $x_l \leftarrow \text{PreRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
11:    R-ii    $b_{l+1} = (P_{l+1}^l)^\top (b_l - A_l x_l)$ 
12:  Correction
13:    if ( $l + 1 == L$ ) then
14:      C-iii  $x_{l+1} = \text{CorrectionSolve}(A_{l+1}, b_{l+1})$ 
15:    else
16:      C-iii WCycle( $l + 1$ )
17:    end if
18:  Prolongation
19:    P-ii    $x_l \leftarrow x_l + P_{l+1}^l x_{l+1}$ 
20:    P-i     $x_l \leftarrow \text{PostRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
21:  Restriction
22:    R-i     $x_l \leftarrow \text{PreRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
23:    R-ii    $b_{l+1} = (P_{l+1}^l)^\top (b_l - A_l x_l)$ 
24:  Correction
25:    if ( $l + 1 == L$ ) then
26:      C-iii  $x_{l+1} = \text{CorrectionSolve}(A_{l+1}, b_{l+1})$ 
27:    else
28:      C-iii WCycle( $l + 1$ )
29:    end if
30:  Prolongation
31:    P-ii    $x_l \leftarrow x_l + P_{l+1}^l x_{l+1}$ 
32:    P-i     $x_l \leftarrow \text{PostRelaxation}(A_l, x_l, b_l)$  (Eq. 4.2)
33: KS      $x_1 \leftarrow \text{KrylovSubspaceUpdate}$  (Eq. 4.5)

```

Chapter 5

Prolongation

The challenge in implementing a multigrid solver is designing a prolongation matrix between the fine and coarse spaces. For signal processing with a 0-form basis, several 0-form prolongation matrices have been designed for real-time applications such as [PKCH18, LZBCJ21]. To solve Eq. 3.12 using the multigrid method (Algorithms 2 to 4), we show how existing 0-form hierarchies can be leveraged to design associated 1-form hierarchies. Concretely, we consider the case of a 1-form basis that can be constructed using a 0-form basis, such as the harmonic-free 1-form basis and the Whitney 1-form basis. We show that the 1-form basis construction induces a 1-form prolongation matrix derived from the 0-form prolongation matrix, which we can then use to define a hierarchy on the 1-forms. We conclude by showing that both the 1-form hierarchies using the harmonic-free 1-form basis and the Whitney 1-form basis

CHAPTER 5. PROLONGATION

are realizations of a more general formulation that can be expressed in terms of functors on the categories of vector spaces.

5.1 Preliminaries

We begin by reviewing some mathematical definitions required for our discussion. In what follows, we assume all vector spaces are finite-dimensional.

Dual, Direct Sum, Tensor Product, and Wedge Product

Given a vector space V , we denote by V^* the dual space of linear functionals on V . In addition, given vector spaces V_0 and V_1 , we denote by $V_0 \oplus V_1$ the direct sum, by $V_0 \otimes V_1$ the tensor product, and by $V_0 \wedge V_1$ the wedge product, of V_0 and V_1 .

Induced Maps

Given vector spaces U and V and a linear map $\mathcal{L} : U \rightarrow V$, we recall that there is a canonical dual map \mathcal{L}^* between the dual spaces:

$$\mathcal{L}^* : V^* \rightarrow U^*.$$

Similarly, given vector spaces $U_i, V_i, i = 0, 1$, and linear maps $\mathcal{L}_i : U_i \rightarrow V_i$, there are canonical maps on the direct sum, the tensor product, and the wedge

CHAPTER 5. PROLONGATION

product:

$$\mathcal{L}_0 \circledast \mathcal{L}_1 : U_0 \circledast U_1 \rightarrow V_0 \circledast V_1,$$

where $\circledast = \oplus, \otimes, \wedge$.

A Functor and Its Induced Maps

We refer to a map taking a vector space to the composition of direct sums and tensor products of the vector space as a *functor on the category of vector spaces* or simply *functor* for short:

$$\mathcal{F}(V) = \oplus_i \otimes^{l_i} V, \tag{5.1}$$

where $i, l_i \in \mathbb{N}$.

If \mathcal{F} is a functor on the category of vector spaces, then \mathcal{F} is also a functor on the category of linear maps between vector spaces. Concretely, given vector spaces U and V and a linear map $\mathcal{L} : U \rightarrow V$, the functor \mathcal{F} defines a linear map:

$$\mathcal{F}(\mathcal{L}) : \mathcal{F}(U) \rightarrow \mathcal{F}(V).$$

Additionally, given $\hat{V} \subset V$ and denoting by $\mathcal{L} : \hat{V} \hookrightarrow V$ the (trivial) injection of

CHAPTER 5. PROLONGATION

the subspace \hat{V} into the space V , then \mathcal{F} induces an injective map:

$$\mathcal{F}(\mathcal{L}) : \mathcal{F}(\hat{V}) \hookrightarrow \mathcal{F}(V). \quad (5.2)$$

Bilinear Forms and Linear Maps

Given a bilinear form $\mathcal{B} : V \times V \rightarrow \mathbb{R}$, we can also think of it as a linear map $\mathcal{B} : V \rightarrow V^*$. For example, the metric g_p on \mathcal{M} :

$$g_p : \mathbf{T}_p\mathcal{M} \times \mathbf{T}_p\mathcal{M} \rightarrow \mathbb{R}, \quad (\vec{u}, \vec{v}) \mapsto \langle \vec{u}, \vec{v} \rangle,$$

can be thought of as a map from the tangent space $\mathbf{T}_p\mathcal{M}$ into the co-tangent space $\mathbf{T}_p^*\mathcal{M}$, defined by:

$$g_p : \mathbf{T}_p\mathcal{M} \rightarrow \mathbf{T}_p^*\mathcal{M}, \quad \vec{u} \mapsto \langle \vec{u}, \cdot \rangle.$$

Bases, Vector Spaces and Matrices as Linear Maps

Given a basis $\{v_i\}_{i=1}^n$, that spans the vector space V , there exists a canonical dual basis $\{v_i^*\}$ for the dual vector space V^* , defined by:

$$v_i^*(v_j) = \delta_{ij},$$

CHAPTER 5. PROLONGATION

where δ_{ij} is the Kronecker delta. Then, expressing $v \in V$ with respect to the basis $\{v_i\}$ by the coefficient vector $\mathbf{v} \in \mathbb{R}^n$ and $v^* \in V^*$ with respect to the basis $\{v_i^*\}$ by $\mathbf{v}^* \in \mathbb{R}^n$, we can use a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ to express the linear map $\mathcal{B} : V \rightarrow V^*$ with respect to the two bases.

5.2 The Finite 1-Form Basis \mathcal{W}

The System Matrix

As described in Section 3.4, we discretize Eq. 3.11:

$$(1 + \alpha(\delta d + d\delta))\omega = \mu + \alpha(\delta \varrho + d\varphi),$$

using a finite basis $\mathcal{W} = \{\omega_i\}_{i=1}^n$ of 1-forms on the surface \mathcal{M} . We denote by V the vector space spanned by \mathcal{W} . This discretization gives the linear system Eq. 3.12, where the mass and stiffness matrices are defined as:

$$\mathbf{M}_{ij} = \int_{\mathcal{M}} \langle \omega_i, \omega_j \rangle dp, \quad \mathbf{S}_{ij} = \int_{\mathcal{M}} \langle d\omega_i, d\omega_j \rangle + \langle \delta\omega_i, \delta\omega_j \rangle dp. \quad (5.3)$$

More generally, given a bilinear form $\mathcal{A} : V \times V \rightarrow \mathbb{R}$, the associated matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is given by:

$$\mathbf{A}_{ij} = \mathcal{A}(\omega_i, \omega_j).$$

CHAPTER 5. PROLONGATION

For Eq. 3.11, the bilinear form $\mathcal{A} : V \times V \rightarrow \mathbb{R}$ is:

$$\mathcal{A}(\mu, \nu) = \langle (1 + \alpha(\delta d + d\delta))\mu, \nu \rangle.$$

The Prolongation Matrix

We consider the case when we are given a coarse basis $\hat{\mathcal{W}} = \{\hat{\omega}_j\}_{j=1}^{\hat{n}}$ with $\hat{\omega}_j \in V$, and we denote by \hat{V} the subspace spanned by $\hat{\mathcal{W}}$. Since $\hat{\omega}_j \in V$, every $\hat{\omega}_j$ can be written as a linear combination of ω_i , i.e., $\hat{\omega}_j = \sum_{i=1}^n \mathbf{P}_{ij}\omega_i$. The matrix $\mathbf{P} \in \mathbb{R}^{n \times \hat{n}}$ of these coefficients, \mathbf{P}_{ij} , is the *prolongation matrix*. Its associated prolongation operator $\mathcal{P} : \hat{V} \hookrightarrow V$ is an injection of \hat{V} into V .

Likewise, we can express the bilinear form \mathcal{A} with respect to $\hat{\mathcal{W}}$ and its dual using a matrix $\hat{\mathbf{A}} \in \mathbb{R}^{\hat{n} \times \hat{n}}$ (we refer to it as the coarse system matrix). By plugging the expansion of the $\{\hat{\omega}_j\}$ in terms of the $\{\omega_i\}$, the coefficients of $\hat{\mathbf{A}}$ are:

$$\begin{aligned} \hat{\mathbf{A}}_{ij} &= \mathcal{A}(\hat{\omega}_i, \hat{\omega}_j) \\ &= \sum_k \sum_l \mathbf{P}_{ki} \mathbf{P}_{lj} \mathcal{A}(\omega_k, \omega_l) \\ &= \sum_k \sum_l \mathbf{P}_{ki} \mathbf{P}_{lj} \mathbf{A}_{kl}. \end{aligned}$$

Therefore, we can simply express the coarse system matrix $\hat{\mathbf{A}}$ using the fine

CHAPTER 5. PROLONGATION

system matrix A and the prolongation matrix P :

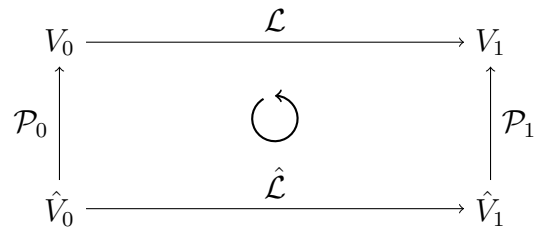
$$\hat{A} = P^T A P. \quad (5.4)$$

The Requirement on Prolongation Matrices

In the multigrid prolongation design, it often requires that the prolongation matrices commute with the discrete differential operator, such as in [RS02, Eq. 15]. The reason is that given two vector spaces $V_i, i = 0, 1$ (e.g., the spaces of 0-forms and 1-forms), a linear map (e.g., a differential operator) between them $\mathcal{L} : V_0 \rightarrow V_1$, and two prolongations $\mathcal{P}_i : \hat{V}_i \hookrightarrow V_i$ defining the coarse subspaces $\hat{V}_i \subset V_i$, if $\hat{\mathcal{L}} : \hat{V}_0 \rightarrow \hat{V}_1$ (defined implicitly by restriction, i.e., $\hat{\mathcal{L}} := \mathcal{L}|_{\hat{V}_0}$) is a well-defined linear map on \hat{V}_i , then $\hat{\mathcal{L}}$ must satisfy:

$$\mathcal{L} \circ \mathcal{P}_0 = \mathcal{P}_1 \circ \hat{\mathcal{L}},$$

as depicted in the below composition diagram:



When expressing using matrices with respect to the fine and coarse bases, denoting by L and \hat{L} the discretized linear maps, and by P_0 and P_1 the prolon-

CHAPTER 5. PROLONGATION

gation matrices, the above equation implies the commutativity requirement stated in [RS02, Eq. 15], *i.e.*,

$$\mathbf{L} \cdot \mathbf{P}_0 = \mathbf{P}_1 \cdot \hat{\mathbf{L}}. \quad (5.5)$$

We will discuss this requirement further when we generalize the basis construction. Now, we review two 1-form bases that are constructed from a 0-form basis, and see how the construction induces a 1-form prolongation matrix \mathcal{P}_1 from the 0-form prolongation matrix \mathcal{P}_0 and how the construction satisfies the commutativity requirement (Eq. 5.5).

5.2.1 Harmonic-Free 1-Form Basis

The Basis Construction

The first 1-form basis we consider is the harmonic-free basis (a.k.a. the gradient and the rotated-gradient basis [PP00]). Recall that the 1-form space, $\Omega_1(\mathcal{M})$, can be decomposed into exact, $d(\Omega_0(\mathcal{M}))$, co-exact, $\delta(\Omega_2(\mathcal{M}))$, and harmonic, $\mathcal{H}^1(\mathcal{M})$, 1-form subspaces by the Hodge decomposition (Eq. 3.8).

Given a 0-form basis $\{\phi_i\}$ spanning a finite-dimensional subspace of $\Omega_0(\mathcal{M})$, using the exterior derivative d , we obtain a basis that spans a subspace of exact 1-forms $d(\Omega_0(\mathcal{M}))$:

$$\mathcal{W}_{\mathcal{E}} := \{d\phi_i\},$$

CHAPTER 5. PROLONGATION

and using the Hodge star operator \star and exterior co-derivative δ , we obtain a basis that spans a subspace of co-exact 1-forms $\delta \star (\Omega_0(\mathcal{M}))$:

$$\mathcal{W}_{\bar{\mathcal{E}}} := \{\delta \star \phi_i\}.$$

Together these form a harmonic-free 1-form basis:

$$\mathcal{W}_{\mathcal{HF}} := \mathcal{W}_{\mathcal{E}} \cup \mathcal{W}_{\bar{\mathcal{E}}}. \quad (5.6)$$

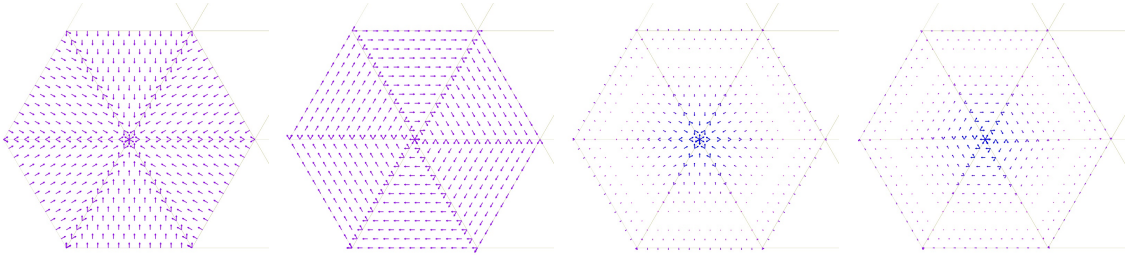


Figure 5.1: A visualization of the exact and co-exact 1-form bases as vector fields on a flat triangle mesh. Considering a 0-form basis function centered at the interior vertex, we obtain two vector fields – the gradient (exact 1-form) and rotated gradient (co-exact 1-form) of the basis function. The first and second columns show the vector fields obtained when the function is the first-order Lagrange interpolant. The third and fourth columns show the vector fields for a second-order Lagrange interpolant.

Fig. 5.1 illustrates the corresponding vector functions of $\mathcal{W}_{\mathcal{HF}}$ using linear and quadratic Lagrange elements on a flat triangle mesh. In this representation, we have a smooth vector field per face, but these are discontinuous at the boundaries. A limitation of this basis is that it cannot represent harmonic

CHAPTER 5. PROLONGATION

vector fields.

The Induced 1-Form Prolongation Matrix

Suppose we are given a fine 0-form basis $\{\phi_i\}_{i=1}^m$ spanning a subspace of 0-forms V and a 0-form prolongation matrix $\mathbf{P} = [\mathbf{P}_{ij}] \in \mathbb{R}^{m \times \hat{m}}$, which we use to define the coarse 0-form basis $\{\hat{\phi}_i = \sum_j \mathbf{P}_{ji} \phi_j\}_{i=1}^{\hat{m}}$ spanning a coarse subspace \hat{V} .

Using the coarse 0-form basis $\{\hat{\phi}_i\}_{i=1}^{\hat{m}}$, we can construct the coarse exact and co-exact 1-form bases:

$$\begin{aligned}\hat{\mathcal{W}}_{\mathcal{E}} &= \{d\hat{\phi}_j\} = \left\{ \sum_i \mathbf{P}_{ij} d\phi_i \right\}, \\ \hat{\mathcal{W}}_{\bar{\mathcal{E}}} &= \{\delta \star \hat{\phi}_j\} = \left\{ \sum_i \mathbf{P}_{ij} \delta \star \phi_i \right\}, \\ \hat{\mathcal{W}}_{\mathcal{HF}} &= \hat{\mathcal{W}}_{\mathcal{E}} \cup \hat{\mathcal{W}}_{\bar{\mathcal{E}}}.\end{aligned}$$

We denote by $V_{\mathcal{HF}}$ the 1-form vector space spanned by $\mathcal{W}_{\mathcal{HF}}$ and by $\hat{V}_{\mathcal{HF}}$ the one spanned by $\hat{\mathcal{W}}_{\mathcal{HF}}$. This construction induces a 1-form prolongation

$$\mathbf{P}_{\mathcal{HF}} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \in \mathbb{R}^{2m \times 2\hat{m}}.$$

CHAPTER 5. PROLONGATION

Fine System Matrices

Using the exterior calculus operator properties (Eqns. 3.1 to 3.6), we obtain:

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_0 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_0 \end{bmatrix},$$

where the coefficients of \mathbf{S}_0 and \mathbf{B}_0 are

$$[\mathbf{S}_0]_{ij} = \int_{\mathcal{M}} \langle d\phi_i, d\phi_j \rangle dp, \quad [\mathbf{B}_0]_{ij} = \int_{\mathcal{M}} \langle \delta d\phi_i, \delta d\phi_j \rangle dp.$$

Note that, in this case, the diagonal blocks of the 1-form mass matrix \mathbf{M} are the 0-form stiffness matrix \mathbf{S}_0 . Computing the coefficients of 1-form stiffness matrix \mathbf{S} is challenging as it requires taking the second derivative of the ϕ_i . We use the following to approximate \mathbf{S} .

Notice that $\delta d : \Omega_0(\mathcal{M}) \rightarrow \Omega_0(\mathcal{M})$. For any $f, g \in \Omega_0(\mathcal{M})$, let $\tilde{f} = \delta df$, $\tilde{g} = \delta dg$.

We express them with respect to the basis $\{\phi_i\}$ by projecting them on V . Let \mathbf{f} , \mathbf{g} , $\tilde{\mathbf{f}}$, $\tilde{\mathbf{g}}$ be the coefficient vectors of the projections with respect to $\{\phi_i\}$. Then, we have:

$$\tilde{\mathbf{f}} = \mathbf{M}_0^{-1} \cdot \mathbf{S}_0 \mathbf{f}, \quad \tilde{\mathbf{g}} = \mathbf{M}_0^{-1} \cdot \mathbf{S}_0 \mathbf{g}.$$

Therefore,

$$\langle \delta df, \delta dg \rangle \approx \tilde{\mathbf{f}}^\top \mathbf{M}_0 \tilde{\mathbf{g}} = \mathbf{f}^\top \mathbf{S}_0 \cdot \mathbf{M}_0^{-1} \cdot \mathbf{S}_0 \mathbf{g}.$$

CHAPTER 5. PROLONGATION

This gives us an expression for computing \mathbf{S} :

$$\mathbf{S} \approx \mathbf{S}_0 \cdot \mathbf{M}_0^{-1} \cdot \mathbf{S}_0.$$

For simplicity, we lump \mathbf{M}_0 to be a diagonal matrix so that it is trivial to invert.

Coarse System Matrices

By Eq. 5.4, we can construct the coarse system matrices using the induced 1-form prolongation matrix $\mathbf{P}_{\mathcal{H}\mathcal{F}}$:

$$\begin{aligned} \hat{\mathbf{M}} = \mathbf{P}_{\mathcal{H}\mathcal{F}}^\top \cdot \mathbf{M} \cdot \mathbf{P}_{\mathcal{H}\mathcal{F}} &= \begin{bmatrix} \mathbf{P}^\top \cdot \mathbf{S}_0 \cdot \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^\top \cdot \mathbf{S}_0 \cdot \mathbf{P} \end{bmatrix}, \\ \hat{\mathbf{S}} = \mathbf{P}_{\mathcal{H}\mathcal{F}}^\top \cdot \mathbf{S} \cdot \mathbf{P}_{\mathcal{H}\mathcal{F}} &= \begin{bmatrix} \mathbf{P}^\top \cdot \mathbf{B}_0 \cdot \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^\top \cdot \mathbf{B}_0 \cdot \mathbf{P} \end{bmatrix}. \end{aligned}$$

The Commutativity Requirement

For this basis, we have two linear maps to consider – the 0-form exterior derivative operator $d : \Omega_0(\mathcal{M}) \rightarrow \Omega_1(\mathcal{M})$ and the 1-form Hodge star operator $\star : \Omega_1(\mathcal{M}) \rightarrow \Omega_1(\mathcal{M})$. We can express them with respect to the fine and coarse bases (and their dual), giving us matrices $\mathbf{D} \in \mathbb{R}^{2m \times m}$, $\hat{\mathbf{D}} \in \mathbb{R}^{2\hat{m} \times \hat{m}}$, $\mathbf{H} \in \mathbb{R}^{2m \times 2m}$,

CHAPTER 5. PROLONGATION

$\hat{\mathbf{H}} \in \mathbb{R}^{2\hat{m} \times 2\hat{m}}$:

$$\mathbf{D} = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix}, \quad \hat{\mathbf{D}} = \begin{bmatrix} \mathbf{I}_{\hat{m}} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \\ -\mathbf{I}_m & \mathbf{0} \end{bmatrix}, \quad \hat{\mathbf{H}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{\hat{m}} \\ -\mathbf{I}_{\hat{m}} & \mathbf{0} \end{bmatrix},$$

where \mathbf{I}_k is a $k \times k$ identity matrix. Our goal is to show that the commutativity requirement, Eq. 5.5, is satisfied for these two linear maps. *i.e.*,

$$\mathbf{D} \cdot \mathbf{P} = \mathbf{P}_{\mathcal{HF}} \cdot \hat{\mathbf{D}} \quad \text{and} \quad \mathbf{H} \cdot \mathbf{P}_{\mathcal{HF}} = \mathbf{P}_{\mathcal{HF}} \cdot \hat{\mathbf{H}}.$$

These are obvious:

$$\mathbf{D} \cdot \mathbf{P} = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{P} = \begin{bmatrix} \mathbf{P} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{\hat{m}} \\ \mathbf{0} \end{bmatrix} = \mathbf{P}_{\mathcal{HF}} \cdot \hat{\mathbf{D}},$$

and

$$\mathbf{H} \cdot \mathbf{P}_{\mathcal{HF}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \\ -\mathbf{I}_m & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{P} \\ -\mathbf{P} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} & \mathbf{I}_{\hat{m}} \\ -\mathbf{I}_{\hat{m}} & \mathbf{0} \end{bmatrix} = \mathbf{P}_{\mathcal{HF}} \cdot \hat{\mathbf{H}}.$$

5.2.2 Whitney 1-Form Basis

The Basis Construction

Another common choice is the Whitney 1-form basis [Whi05], which is also defined in terms of a 0-form basis. Given a 0-form basis $\{\phi_i\}$,

$$\mathcal{W}_{\mathcal{WN}} := \{\phi_{ij} = \phi_i \cdot d\phi_j - \phi_j \cdot d\phi_i \mid i < j \text{ and } \text{supp}(\phi_i) \cap \text{supp}(\phi_j) \neq \emptyset\}. \quad (5.7)$$

Fig. 5.2 illustrates the corresponding vector fields derived from linear and

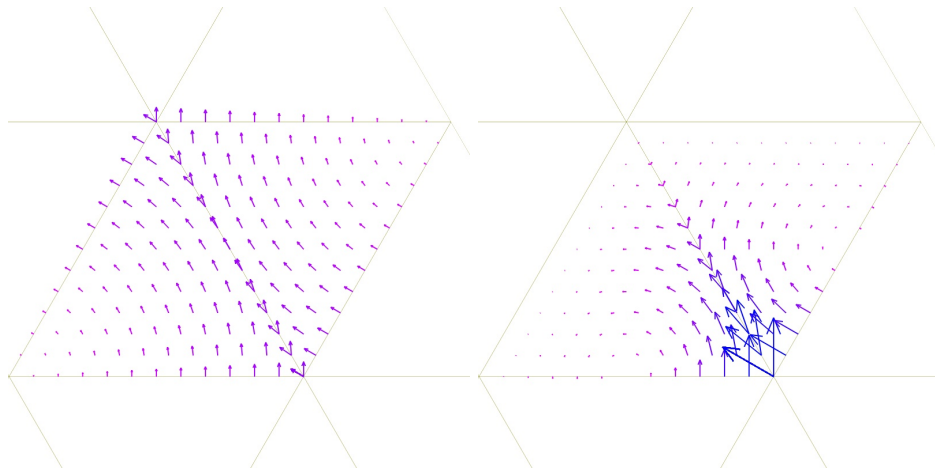


Figure 5.2: A visualization of the Whitney 1-form basis function associated to an edge as a vector field on a flat triangle mesh using a linear and quadratic Lagrange basis.

quadratic Lagrange Whitney basis function associated to an edge of a flat triangle mesh. Though the vector fields are still not continuous across the edge, their tangential components are.

CHAPTER 5. PROLONGATION

The Induced 1-Form Prolongation Matrix

Similar to the harmonic-free 1-form basis, given a fine 0-form basis $\{\phi_i\}_{i=1}^m$ that spans a fine subspace of 0-forms V and a 0-form prolongation matrix $\mathbf{P} = [\mathbf{P}_{ij}] \in \mathbb{R}^{m \times \hat{m}}$, which defines a coarse 0-form basis $\{\hat{\phi}_i = \sum_j \mathbf{P}_{ji} \phi_j\}_{i=1}^{\hat{m}}$ that spans a coarse subspace of 0-forms \hat{V} , one can construct the coarse Whitney 1-form basis by setting:

$$\hat{\mathcal{W}}_{\mathcal{WN}} = \{\hat{\phi}_{mn} = \hat{\phi}_m \cdot d\hat{\phi}_n - \hat{\phi}_n \cdot d\hat{\phi}_m \mid m < n \text{ and } \text{supp}(\hat{\phi}_m) \cap \text{supp}(\hat{\phi}_n) \neq \emptyset\}.$$

Likewise, we denote by $V_{\mathcal{WN}}$ the fine 1-form subspace spanned by $\mathcal{W}_{\mathcal{WN}}$ and by $\hat{V}_{\mathcal{WN}}$ the coarse 1-form subspace by $\hat{\mathcal{W}}_{\mathcal{WN}}$. To simplify the notation, we denote by $\mathbf{i} = (i_1, i_2)$ a multi-index, and write $\phi_{\mathbf{i}}$ instead of $\phi_{i_1 i_2}$. By direct expansion, we can express the coarse 1-form in terms of the fine one:

$$\begin{aligned} \hat{\phi}_{\mathbf{i}} &= \hat{\phi}_{i_1} \cdot d\hat{\phi}_{i_2} - \hat{\phi}_{i_2} \cdot d\hat{\phi}_{i_1} \\ &= \sum_{j_1} \sum_{j_2} \mathbf{P}_{j_1 i_1} \mathbf{P}_{j_2 i_2} \phi_{j_1} \cdot d\phi_{j_2} - \sum_{j_1} \sum_{j_2} \mathbf{P}_{j_2 i_2} \mathbf{P}_{j_1 i_1} \phi_{j_2} \cdot d\phi_{j_1} \\ &= \sum_{j_1} \sum_{j_2} \mathbf{P}_{j_1 i_1} \mathbf{P}_{j_2 i_2} \phi_{\mathbf{j}} \\ &= \sum_{\mathbf{j} | j_1 < j_2} \mathbf{P}_{j_1 i_1} \mathbf{P}_{j_2 i_2} \phi_{\mathbf{j}} + \sum_{\mathbf{j} | j_1 > j_2} \mathbf{P}_{j_1 i_1} \mathbf{P}_{j_2 i_2} \phi_{\mathbf{j}} \quad [\phi_{\mathbf{j}} = 0 \text{ if } j_1 = j_2] \\ &= \sum_{\mathbf{j} | j_1 < j_2} \mathbf{P}_{j_1 i_1} \mathbf{P}_{j_2 i_2} \phi_{\mathbf{j}} - \sum_{\mathbf{j} | j_2 > j_1} \mathbf{P}_{j_2 i_1} \mathbf{P}_{j_1 i_2} \phi_{\mathbf{j}} \quad [\phi_{j_1 j_2} = -\phi_{j_2 j_1}] \\ &= \sum_{\mathbf{j} | j_1 < j_2} (\mathbf{P}_{j_1 i_1} \mathbf{P}_{j_2 i_2} - \mathbf{P}_{j_2 i_1} \mathbf{P}_{j_1 i_2}) \phi_{\mathbf{j}}. \end{aligned}$$

CHAPTER 5. PROLONGATION

As a result, this construction induces a 1-form prolongation $\mathbf{P}_{\mathcal{W}\mathcal{N}}$, whose coefficients are given by:

$$\mathbf{P}_{\mathbf{ij}} = \mathbf{P}_{i_1j_1}\mathbf{P}_{i_2j_2} - \mathbf{P}_{i_2j_1}\mathbf{P}_{i_1j_2}. \quad (5.8)$$

Fine System Matrices

In this case, we have:

$$[\mathbf{M}]_{\mathbf{ij}} = \int_{\mathcal{M}} \langle \phi_{\mathbf{i}}, \phi_{\mathbf{j}} \rangle dp \quad [\mathbf{S}]_{\mathbf{ij}} = \underbrace{\int_{\mathcal{M}} \langle d\phi_{\mathbf{i}}, d\phi_{\mathbf{j}} \rangle dp}_{[\mathbf{S}_{div}]_{\mathbf{ij}}} + \underbrace{\int_{\mathcal{M}} \langle \delta\phi_{\mathbf{i}}, \delta\phi_{\mathbf{j}} \rangle dp}_{[\mathbf{S}_{curl}]_{\mathbf{ij}}}.$$

By definition, $\delta = -\star^{-1}d\star$, so we can compute \mathbf{S}_{curl} by setting:

$$[\mathbf{S}_{curl}]_{\mathbf{ij}} = \int_{\mathcal{M}} \langle \star^{-1}d\star\phi_{\mathbf{i}}, \star^{-1}d\star\phi_{\mathbf{j}} \rangle dp.$$

If the 0-form basis $\{\phi_{\mathbf{i}}\}$ forms a partition of unity (*i.e.*, $\sum_{\mathbf{i}} \phi_{\mathbf{i}} = 1$), then, for a 0-form f expressed with respect to the basis $\{\phi_{\mathbf{i}}\}$ by the coefficient vector $\mathbf{f} = [f_{\mathbf{i}}]$, we have

$$\begin{aligned} df &= \sum_{i_2} f_{i_2} \cdot d\phi_{i_2} \\ &= \sum_{i_2} f_{i_2} \left(\sum_{i_1} \phi_{i_1} \right) \cdot d\phi_{i_2} - \sum_{i_2} f_{i_2} \phi_{i_2} \cdot d \left(\sum_{i_1} \phi_{i_1} \right) \\ &= \sum_{i_2} \sum_{i_1} f_{i_2} \phi_{i_1} \\ &= \sum_{\mathbf{i} | i_1 < i_2} f_{i_2} \phi_{\mathbf{i}} + \sum_{\mathbf{i} | i_1 > i_2} f_{i_2} \phi_{\mathbf{i}} \end{aligned}$$

CHAPTER 5. PROLONGATION

$$\begin{aligned}
 &= \sum_{\mathbf{i}|i_1 < i_2} f_{i_2} \phi_{\mathbf{i}} - \sum_{\mathbf{i}|i_2 > i_1} f_{i_1} \phi_{\mathbf{i}} \\
 &= \sum_{\mathbf{i}|i_1 < i_2} (f_{i_2} - f_{i_1}) \phi_{\mathbf{i}}.
 \end{aligned}$$

Thus, we can use a matrix \mathbf{D} to express the 0-form exterior derivative d with respect to the bases $\{\phi_i\}$ and $\{\phi_{\mathbf{i}}\}$, with entries:

$$\mathbf{D}_{ij} = \begin{cases} -1 & \text{if } i_1 = j, \\ 1 & \text{if } i_2 = j, \\ 0 & \text{otherwise.} \end{cases} \quad (5.9)$$

For the Whitney basis, the Hodge star operators can be discretized by the mass matrices [TKB99]. By denoting the 0-form mass matrix \mathbf{M}_0 , we can use the below expression for computing \mathbf{S}_{curl} :

$$\mathbf{S}_{curl} = \mathbf{M} \cdot \mathbf{D} \cdot (\mathbf{M}_0)^{-1} \cdot \mathbf{D}^\top \cdot \mathbf{M}.$$

Again, we lump \mathbf{M}_0 to be a diagonal matrix so that it is trivial to invert.

CHAPTER 5. PROLONGATION

Coarse System Matrices

Using the induced 1-form prolongation $P_{\mathcal{WN}}$ defined in Eq. 5.8, Eq. 5.4 gives the coarse 1-form mass and stiffness matrices:

$$\hat{\mathbf{M}} = \mathbf{P}_{\mathcal{WN}}^\top \cdot \mathbf{M} \cdot \mathbf{P}_{\mathcal{WN}}, \quad \hat{\mathbf{S}} = \mathbf{P}_{\mathcal{WN}}^\top \cdot \mathbf{S} \cdot \mathbf{P}_{\mathcal{WN}}.$$

The Commutativity Requirement

Now, we will show the commutativity requirement (Eq. 5.5) is also satisfied for the Whitney 1-form basis. In this case, we consider the linear map $d : \Omega_0(\mathcal{M}) \rightarrow \Omega_1(\mathcal{M})$.

We assume that the 0-form basis $\{\phi_i\}$ forms a partition of unity (*i.e.*, $\sum_i \phi_i = 1$) and the 0-form prolongation matrix preserves the partition of unity. *i.e.*, $\sum_i P_{ji} = 1$. So, we have:

$$\sum_i \hat{\phi}_i = \sum_i \sum_j P_{ji} \phi_j = \sum_j \left(\sum_i P_{ji} \right) \phi_j = 1.$$

As discussed before, if the basis forms a partition of unity, we can express the 0-form exterior derivative operator d with respect to the fine and coarse bases using matrices \mathbf{D} and $\hat{\mathbf{D}}$ as defined in Eq. 5.9. Our goal is then to show that the prolongation matrices \mathbf{P} and $\mathbf{P}_{\mathcal{WN}}$ commute with the discrete derivative

CHAPTER 5. PROLONGATION

matrices D and \hat{D} . *i.e.*,

$$D \cdot P = P_{\mathcal{WN}} \cdot \hat{D}.$$

We can show it by direct expansion.

$$\begin{aligned}
 [D \cdot P]_{ij} &= \sum_k D_{ik} P_{kj} \\
 &= P_{i2j} - P_{i1j} && \text{[Def. of } D_{ik}] \\
 &= \left(\sum_{k_1} P_{i_1 k_1} \right) P_{i_2 j} - \left(\sum_{k_2} P_{i_2 k_2} \right) P_{i_1 j} && \text{[} \sum_j P_{ij} = 1] \\
 &= \sum_{k_1} P_{i_1 k_1} P_{i_2 j} - \sum_{k_2} P_{i_2 k_2} P_{i_1 j} \\
 &= \sum_{k_1} \sum_{k_2} P_{i_1 k_1} P_{i_2 k_2} \hat{D}_{k_j} && \text{[Def. of } \hat{D}_{k_j}] \\
 &= \sum_{\mathbf{k}|k_1 < k_2} P_{i_1 k_1} P_{i_2 k_2} \hat{D}_{\mathbf{k}j} + \sum_{\mathbf{k}|k_1 > k_2} P_{i_2 k_2} P_{i_1 k_1} \hat{D}_{\mathbf{k}j} && \text{[} \hat{D}_{\mathbf{k}j} = 0 \text{ if } k_1 = k_2] \\
 &= \sum_{\mathbf{k}|k_1 < k_2} P_{i_1 k_1} P_{i_2 k_2} \hat{D}_{\mathbf{k}j} - \sum_{\mathbf{k}|k_2 < k_1} P_{i_2 k_2} P_{i_1 k_1} \hat{D}_{(k_2, k_1)j} && \text{[} \hat{D}_{(k_1, k_2)j} = -\hat{D}_{(k_2, k_1)j}] \\
 &= \sum_{\mathbf{k}|k_1 < k_2} P_{i_1 k_1} P_{i_2 k_2} \hat{D}_{\mathbf{k}j} - \sum_{\mathbf{k}|k_1 < k_2} P_{i_2 k_1} P_{i_1 k_2} \hat{D}_{\mathbf{k}j} \\
 &= \sum_{\mathbf{k}|k_1 < k_2} (P_{i_1 k_1} P_{i_2 k_2} - P_{i_2 k_1} P_{i_1 k_2}) \hat{D}_{\mathbf{k}j} \\
 &= \sum_{\mathbf{k}|k_1 < k_2} P_{i\mathbf{k}} \hat{D}_{\mathbf{k}j} \\
 &= [P_{\mathcal{WN}} \cdot \hat{D}]_{ij}.
 \end{aligned}$$

5.3 Generalization

There are certain similarities in constructing the harmonic-free and Whitney 1-form bases, as both induce a prolongation matrix that can be used to define the coarse vector space and system matrices, and the prolongation matrices commute with related linear maps. We generalize this construction using the language of category theory similar to [LS20], which provides a category theoretical interpretation of the Galerkin finite element method.

5.3.1 Basic Set-up

Vector Space Constructions by Functors

Looking back, to generalize the constructions we have seen, we can try to use functors on the category of vector spaces. Given a vector space V , such as a subspace of 0-forms spanned by a basis $\{\phi_i\}$, and a prolongation $\mathcal{P} : \hat{V} \hookrightarrow V$ defining the coarse subspace \hat{V} , we “construct” two vector spaces $V_i = \mathcal{F}_i(V)$ using two functors (Eq. 5.1) $\mathcal{F}_i, i = 0, 1$. In what we discussed above, \mathcal{F}_0 is typically the identity (so that V_0 is the space of 0-forms) and V_1 is a space of 1-form. However, we will look at the more general case as it will allow us to generalize it to k -forms.

CHAPTER 5. PROLONGATION

Sufficiency Conditions for Linear Map Restrictibility

As described by Eq. 5.2, the functors \mathcal{F}_i also define canonical prolongations $\mathcal{P}_i = \mathcal{F}_i(\mathcal{P}) : \mathcal{F}_i(\hat{V}) \hookrightarrow \mathcal{F}_i(V)$. In this construction, the two prolongations are defined independently of each other. However, in many applications, the spaces V_0 and V_1 are related by a linear map, $\mathcal{L} : V_0 \rightarrow V_1$ (e.g. the exterior derivative from k -forms to $(k + 1)$ -forms or the Hodge star mapping k -forms to $(n - k)$ -forms). In this context, we would like to be able to define a linear operator on the coarser spaces, $\hat{\mathcal{L}} : \hat{V}_0 \rightarrow \hat{V}_1$.

For $\hat{\mathcal{L}}$ to be well-defined, its image on \hat{V}_0 must be in \hat{V}_1 . Therefore, the necessary and sufficient condition for $\hat{\mathcal{L}}$ to be well-defined is

$$\mathcal{L}(\hat{V}_0) \subset \hat{V}_1. \tag{5.10}$$

In this case, we can simply let $\hat{\mathcal{L}}$ be the restriction $\hat{\mathcal{L}} = \mathcal{L}|_{\hat{V}_0}$.

The Commutativity Requirement

Assuming $\hat{\mathcal{L}}$ is well defined, the prolongations automatically satisfy the property: $\mathcal{L} \circ \mathcal{P}_0 = \mathcal{P}_1 \circ \hat{\mathcal{L}}$, and thus satisfy the commutativity requirement (in terms of their matrix representation.)

In other words, one would need to define the functors carefully such that Eq. 5.10 is satisfied.

5.3.2 General Vector Space Construction

In our context, the set-up is more complicated as $\mathcal{F}_i(V)$ (a composition of direct sum and tensor product of V) is, in general, a “bigger” space than the spaces of k -forms V_i . Instead of $V_i = \mathcal{F}_i(V)$ as described above, we have V_i “inside” $\mathcal{F}_i(V)$.

To address this, we reproduce the discussion above when the linear map $\mathcal{L} : V_0 \rightarrow V_1$ can be pulled back to a linear map $\bar{\mathcal{L}} : \mathcal{F}_0(V) \rightarrow \mathcal{F}_1(V)$ and describe the necessary and sufficient condition for the coarse linear operator $\hat{\mathcal{L}} : \hat{V}_0 \rightarrow \hat{V}_1$ to be well-defined. We conclude by showing that for both the harmonic-free and the Whitney basis, the required conditions are met, and the associated linear operator can be defined between the coarse subspaces.

Vector Space Constructions by Functors

In addition to the functors \mathcal{F}_i , we define the vector spaces $V_i = M_i \circ \mathcal{F}_i(V)$, where $M_i : \mathcal{F}_i(V) \rightarrow V_i$ are maps that pull a linear map $\mathcal{L} : V_0 \rightarrow V_1$ back to a linear map $\bar{\mathcal{L}} : \mathcal{F}_0(V) \rightarrow \mathcal{F}_1(V)$. Along the same lines, we define the subspaces $\hat{V}_i = M_i \circ \mathcal{F}_i(\hat{V})$ and the functors \mathcal{F}_i induce canonical prolongations $\mathcal{F}_i(\mathcal{P}) : \hat{V}_i \hookrightarrow V_i$.

CHAPTER 5. PROLONGATION

Sufficiency Conditions for Linear Map Restrictibility

Given a linear map $\mathcal{L} : V_0 \rightarrow V_1$, we define a coarse linear map by restriction $\hat{\mathcal{L}} = \mathcal{L}|_{\hat{V}_0}$. $\hat{\mathcal{L}}$ is well-defined only if $\mathcal{L}(\hat{V}_0) \subset \hat{V}_1$.

Claim 1. Sufficiency conditions for $\mathcal{L}(\hat{V}_0) \subset \hat{V}_1$ are:

1. There exists a map $\bar{\mathcal{L}} : \mathcal{F}_0(V) \rightarrow \mathcal{F}_1(V)$ such that $\mathcal{L} \circ M_0 = M_1 \circ \bar{\mathcal{L}}$:

$$\begin{array}{ccc}
 \mathcal{F}_0(V) & \xrightarrow{\quad \bar{\mathcal{L}} \quad} & \mathcal{F}_1(V) \\
 M_0 \downarrow & \circlearrowleft & \downarrow M_1 \\
 V_0 & \xrightarrow{\quad \mathcal{L} \quad} & V_1
 \end{array}$$

2. The restriction of $\bar{\mathcal{L}}$ to $\mathcal{F}_0(\hat{V})$ lies in $\mathcal{F}_1(\hat{V})$:

$$\bar{\mathcal{L}}(\mathcal{F}_0(\hat{V})) \subset \mathcal{F}_1(\hat{V}).$$

Proof. Given $\hat{v}_0 \in \hat{V}_0 = M_0 \circ \mathcal{F}_0(\hat{V})$, there exists $\omega \in \mathcal{F}_0(\hat{V})$ such that $M_0(\omega) = \hat{v}_0$. Since (by condition 2) we have $\bar{\mathcal{L}}(\omega) \in \bar{\mathcal{L}}(\mathcal{F}_0(\hat{V})) \subset \mathcal{F}_1(\hat{V})$, we can leverage the commutativity (condition 1) to get:

$$\mathcal{L}(\hat{v}_0) = \mathcal{L} \circ M_0(\omega) = M_1 \circ \bar{\mathcal{L}}(\omega) \in M_1 \circ \bar{\mathcal{L}}(\mathcal{F}_0(\hat{V})) \subset M_1 \circ \mathcal{F}_1(\hat{V}) = \hat{V}_1,$$

as desired. □

The Commutativity Requirement

Likewise, the condition above ensures $\hat{\mathcal{L}}$ is well defined, and the prolongations automatically satisfy the property: $\mathcal{L} \circ \mathcal{P}_0 = \mathcal{P}_1 \circ \hat{\mathcal{L}}$, and thus satisfy the commutativity requirement (in terms of their matrix representation.)

In other words, using this general approach, one would need to define the functors and the pull-back maps carefully such that the conditions in *Claim 1* are satisfied.

5.3.3 Applications

Assume that we discretize the 0-form space $\Omega_0(\mathcal{M})$ using a subspace V of scalar functions (0-forms) on a triangle mesh (e.g., the piecewise linear “hat” basis $\{\phi_i\}$). We show that both the harmonic-free 1-form and the Whitney 1-form are realizations of the above general construction.

5.3.3.1 Harmonic-Free 1-Form

Basis Construction

We construct the 0-form subspace V_1 and 1-form subspace V_2 by setting

$$\mathcal{F}_0(V) = V, \quad M_0(\phi) = \phi \quad \text{and} \quad \mathcal{F}_1(V) = V \oplus V, \quad M_1(\phi, \psi) = d\phi + \delta \star \psi.$$

CHAPTER 5. PROLONGATION

These give

$$V_0 = M_0 \circ \mathcal{F}_0(V) = V \quad \text{and} \quad V_1 = M_1 \circ \mathcal{F}_1(V) = d(V) \oplus \delta \star (V),$$

where V_1 is the definition of the harmonic-free 1-form described in Section 5.2.1.

Sufficiency Conditions

As in Section 5.2.1, we consider two linear maps – the 0-form exterior derivative operator $d : \Omega_0(\mathcal{M}) \rightarrow \Omega_1(\mathcal{M})$ and the 1-form Hodge star operator $\star : \Omega_1(\mathcal{M}) \rightarrow \Omega_1(\mathcal{M})$.

0-form Exterior Derivative: $\mathcal{L} = d : V_0 \rightarrow V_1$

We show that the above construction satisfies both sufficiency conditions.

In this case, we can take:

$$\bar{\mathcal{L}}(\phi) = (\phi, 0).$$

Condition 1: $\mathcal{L} \circ M_0 = M_1 \circ \bar{\mathcal{L}}$

The L.H.S. and R.H.S. are the same after direct expansion.

$$\mathcal{L} \circ M_0(\phi) = \mathcal{L}(\phi) = d\phi \quad \text{and} \quad M_1 \circ \bar{\mathcal{L}}(\phi) = M_1(\phi, 0) = d\phi.$$

CHAPTER 5. PROLONGATION

Condition 2: $\bar{\mathcal{L}}(\mathcal{F}_0(\hat{V})) \subset \mathcal{F}_1(\hat{V})$

For any $\hat{v} \in \hat{V}$, we have $\bar{\mathcal{L}}(\mathcal{F}_0(\hat{v})) = \bar{\mathcal{L}}(\hat{v}) = (\hat{v}, 0) \in \hat{V} \oplus \hat{V} = \mathcal{F}_1(\hat{V})$.

1-Form Hodge Star: $\mathcal{L} = \star : V_1 \rightarrow V_1$

In this case, we set

$$\bar{\mathcal{L}}(\phi, \psi) = (-\psi, \phi).$$

Note that, in this case, \mathcal{L} is mapping from V_1 to V_1 . We show the commutativity conditions are met.

Condition 1: $\mathcal{L} \circ M_1 = M_1 \circ \bar{\mathcal{L}}$

Using exterior calculus operator properties Eqns. 3.3, 3.4, and 3.5, we have the L.H.S.:

$$\mathcal{L} \circ M_1(\phi, \psi) = \mathcal{L}(d\phi + \delta \star \psi) = \star d\phi + \star \delta \star \psi = \delta \star \phi - d\psi,$$

and the R.H.S.:

$$M_1 \circ \bar{\mathcal{L}}(\phi, \psi) = M_1(-\psi, \phi) = -d\psi + \delta \star \phi.$$

CHAPTER 5. PROLONGATION

Condition 2: $\bar{\mathcal{L}}(\mathcal{F}_1(\hat{V})) \subset \mathcal{F}_1(\hat{V})$

For any $\hat{v} \in \hat{V}$, we have $\bar{\mathcal{L}}(\mathcal{F}_1(\hat{v})) = \bar{\mathcal{L}}(\hat{v}, \hat{v}) = (-\hat{v}, \hat{v}) \in \hat{V} \oplus \hat{V} = \mathcal{F}_1(\hat{V})$.

5.3.3.2 Whitney k -Form

Basis Construction

For the Whitney basis, our generalization applies to the general k -forms. We can obtain the k -form subspace V_k from the 0-form subspace V by setting

$$\mathcal{F}_k(V) = \bigotimes_{i=0}^k V, \quad M_k(\phi_0, \dots, \phi_k) = \sum_{j=0}^k (-1)^j \phi_j \cdot d\phi_0 \wedge \dots \wedge d\phi_{j-1} \wedge d\phi_{j+1} \wedge \dots \wedge d\phi_k.$$

These give

$$V_k = M_k \circ \mathcal{F}_k(V),$$

which is the space of Whitney k -forms.

Sufficiency Conditions

In this case, as in Section 5.2.2, we consider the k -form exterior derivative operator $d : \Omega_k(\mathcal{M}) \rightarrow \Omega_{k+1}(\mathcal{M})$.

CHAPTER 5. PROLONGATION

***k*-form Exterior Derivative:** $\mathcal{L} = d : V_k \rightarrow V_{k+1}$

Assuming the constant function is in V_k and \hat{V}_k , to show the construction satisfies the conditions, we set

$$\bar{\mathcal{L}}(\phi_0, \dots, \phi_k) = (k, \phi_0, \dots, \phi_k).$$

Condition 1: $\mathcal{L} \circ M_k = M_{k+1} \circ \bar{\mathcal{L}}$

Through direct expansion, we have

$$\begin{aligned} \mathcal{L} \circ M_k(\phi_0, \dots, \phi_k) &= \mathcal{L}\left(\sum_{j=0}^k (-1)^j \phi_j \cdot d\phi_0 \wedge \dots \wedge d\phi_{j-1} \wedge d\phi_{j+1} \dots \wedge d\phi_k\right) \\ &= \sum_{j=0}^k (-1)^j \cdot d\phi_j \wedge d\phi_0 \wedge \dots \wedge d\phi_{j-1} \wedge d\phi_{j+1} \dots \wedge d\phi_k \\ &= k \cdot d\phi_0 \wedge \dots \wedge d\phi_k, \end{aligned}$$

and

$$\begin{aligned} M_{k+1} \circ \bar{\mathcal{L}}(\phi_0, \dots, \phi_k) &= M_{k+1}(k, \phi_0, \dots, \phi_k) \\ &= k \cdot d\phi_0 \wedge \dots \wedge d\phi_k + \sum_{j=1}^{k+1} (-1)^j \phi_j \cdot 0 \wedge d\phi_0 \wedge \dots \\ &= k \cdot d\phi_0 \wedge \dots \wedge d\phi_k. \end{aligned}$$

Condition 2: $\bar{\mathcal{L}}\left(\mathcal{F}_k(\hat{V})\right) \subset \mathcal{F}_{k+1}(\hat{V})$

$$\bar{\mathcal{L}}\left(\mathcal{F}_k(\hat{V})\right) = \bar{\mathcal{L}}\left(\bigotimes_{i=0}^k \hat{V}\right) = \{k\} \otimes \bigotimes_{i=0}^k \hat{V} \subset \bigotimes_{i=0}^{k+1} \hat{V} = \mathcal{F}_{k+1}(\hat{V}).$$

5.4 Summary

We reviewed two ways to construct a 1-form basis from a 0-form basis. In both cases, given a 0-form prolongation matrix, the construction induces a 1-form prolongation matrix, which we will use for implementing a multigrid solver. We generalized the construction by formulating it in the language of category theory and described the sufficiency conditions for linear operators defined over the fine space to be restrictable, via the induced prolongation, to the coarse space. The sufficiency conditions guarantee the commutativity between the induced prolongations and linear operators, which is often a desired requirement for the multigrid prolongation design.

Chapter 6

Convergence Study

In this chapter, we study the convergence of the multigrid method described in Chapter 4, solving the gradient domain problem formulated in Chapter 3 with the the induced 1-form prolongation matrix presented in Chapter 5. Our hierarchical solver is built on top of a 0-form system. To evaluate the convergence, we use three different existing hierarchical 0-form systems and two 1-form bases. We compare different multigrid settings as described in Chapter 4 and choose the best setting for the problem and compare its performance with the other iterative and direct solvers.

6.1 Setup

6.1.1 The Models

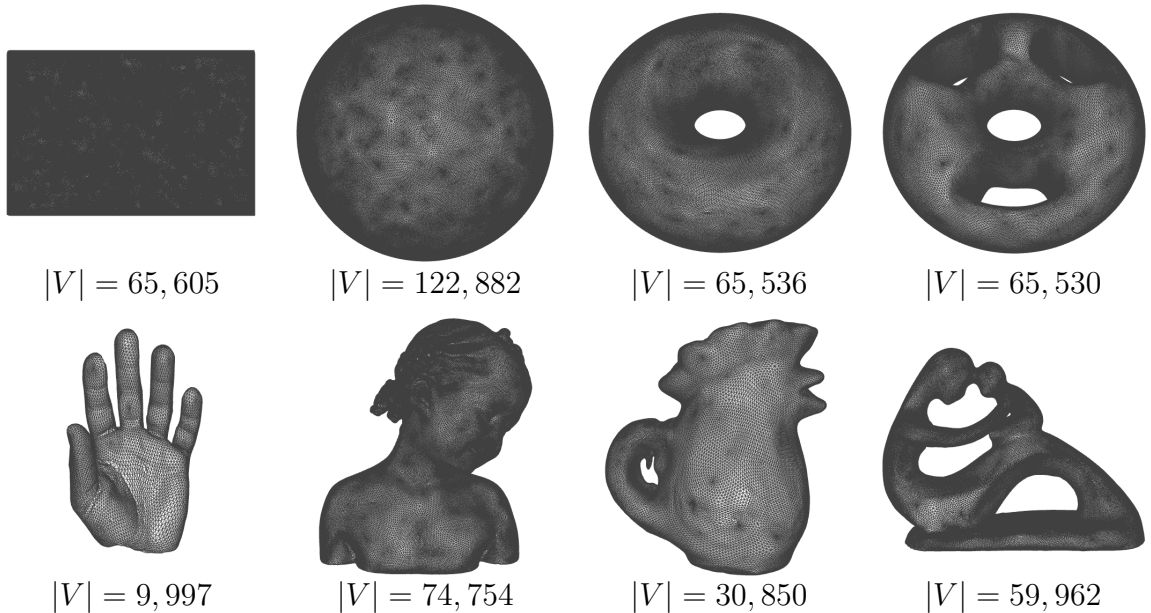


Figure 6.1: Wire-frame visualizations of the eight models used in this study and their vertex counts. From left to right, the first row is generated models – the plane, sphere, torus, and 4-torus. The second row is scanned models – the hand, bimba, rooster, and fertility.

We use eight models for the study, including four generated simple shapes – the plane, the sphere, the torus, and the 4-torus – and four scanned meshes – the hand, the bimba, the rooster, and the fertility. These are the topology of genus-0 with boundary, genus-0 without boundary, genus-1, and genus-4, as shown in Fig. 6.1 with their number of vertices.

6.1.2 0-Form Hierarchical Structures

To implement our 1-form hierarchical solver, we need an initial 0-form hierarchy. Specifically, we need a 0-form basis and a 0-form prolongation matrix for the multigrid method. Then, we can construct the 1-form bases and prolongation matrices described in Sections 5.2.1 and 5.2.2. In this study, we consider three 0-form hierarchies described in the literature, and we use the ‘hat’ basis functions to define the space of 0-forms at the finest resolution. Below, we describe how we obtain these hierarchies and the 0-form prolongation matrices.

6.1.2.1 Mesh Simplification

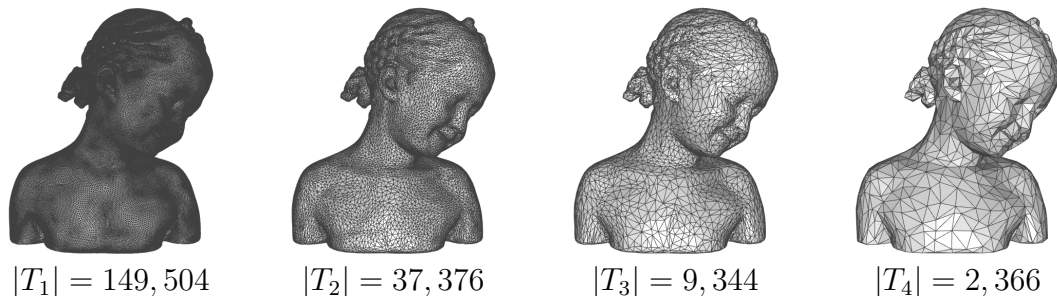


Figure 6.2: The hierarchical meshes of the bimba¹model obtained by using quadric error metric simplification algorithm, visualized with wireframe on top of the meshes.

The first hierarchical structure is the simplest one. We obtain a mesh hierarchy by mesh simplification using Quadric Error Metric (QEM) [GH97]. Given an input triangle mesh M with $|T|$ faces, we simplify the mesh L times using the

¹We selectively show the hierarchy on one model. A complete visual comparison is provided in Appendix A.

CHAPTER 6. CONVERGENCE STUDY

QEM simplification algorithm, as implemented in MeshLab [CCC*08], with the number of triangles reduced by a factor of four at each simplification, i.e., $|T_{l+1}| = |T_l|/4$. Fig. 6.2 shows the mesh hierarchy obtained by applying the method to the bimba model.

To define the 0-form prolongation, we proceed as follows. For each vertex at the finer mesh, we find the closest point on the coarse mesh. For each coarse ‘hat’ basis function, we evaluate it at the closest point as the prolongation matrix coefficient. In practice, since the ‘hat’ basis function only has a local support, the resulting prolongation matrix is sparse.

6.1.2.2 Sub-division Meshes

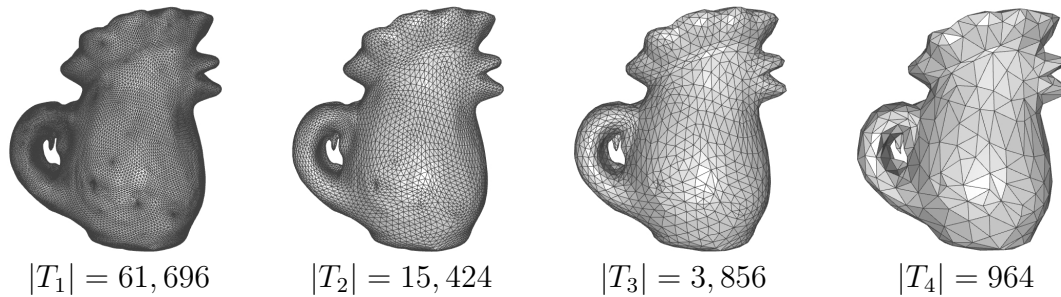


Figure 6.3: Sub-division meshes of the rooster²model, visualized with wire-frame on top of the meshes.

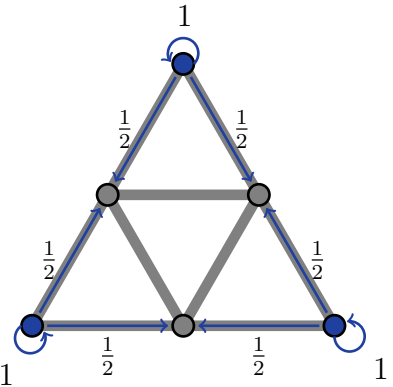
The second hierarchical structure uses both mesh simplification and mesh sub-division [Loo87]. After applying the mesh simplification steps described above,

²We selectively show the hierarchy on one model. A complete visual comparison is provided in Appendix A.

CHAPTER 6. CONVERGENCE STUDY

given a simplified mesh M_L with $|T|/4^L$ triangles, we apply L iterations of subdivision to M_L to get back a mesh with the original number of triangles but having a subdivision structure. After each subdivision iteration, we project the new vertices onto the original mesh M . Note that the projection step may introduce self-intersecting triangles, especially around the mesh boundaries. We manually inspect the mesh and re-adjust vertex positions to ensure the mesh has no self-intersections. This generates a sequence of meshes M_1, M_2, \dots, M_L with M_1 the finest mesh and M_L the coarsest. Fig. 6.3 show an example of sub-division mesh of the rooster model.

In the sub-division scheme, each triangle is locally split into four using the edge mid points. This gives a simple 0-form prolongation stencil, as shown in the inset. For each coarse vertex (blue) on a coarse triangle, after the sub-division, it has



three fine supporting vertices on the fine triangle – itself and the two newly added mid-edge vertices (gray). The prolongation weights for the coarse vertex are 1 for itself (blue to blue), and $\frac{1}{2}$ for the others (blue to gray). The prolongation matrix is found by assembling these local stencil weights. Note that, by construction, the row sum of the prolongation matrix equals one.

CHAPTER 6. CONVERGENCE STUDY

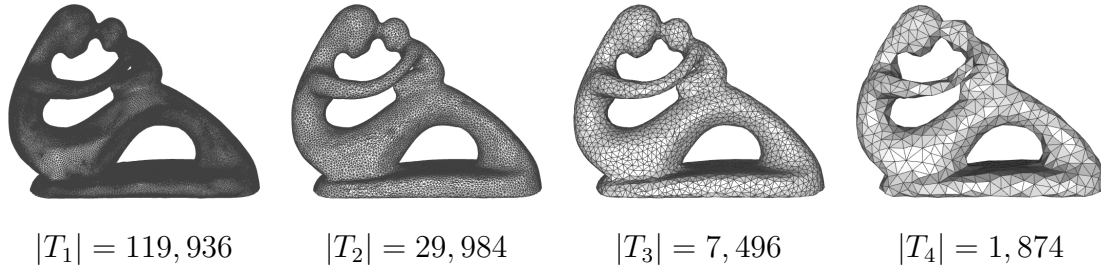


Figure 6.4: Self-parameterization meshes of the fertility³model, visualized with wireframe on top of the meshes.

6.1.2.3 Intrinsic Self-Parameterization

The third hierarchical structure is the intrinsic self-parameterization of Liu *et al.* [LZBCJ21], which has been shown to outperform other hierarchical structures for signal processing with the multigrid method. During mesh simplification, this method optimizes the map between the original and simplified meshes. With this optimized map, one can express the fine vertices in the coarse mesh using the barycentric coordinates. A 0-form prolongation is then obtained by evaluating coarse ‘hat’ basis functions at the mapped fine vertices. Fig. 6.4 shows the hierarchy created using the intrinsic self-parameterization on the fertility model. We use the implementation of Liu *et al.* [LZBCJ21] to generate this hierarchy and the associated 0-form prolongation matrices.

²We selectively show the hierarchy on one model. A complete visual comparison is provided in Appendix A.

Remarks

We pre-process the input models by first generating the sub-division hierarchy, then we create the mesh simplification and self-intrinsic parameterization hierarchies using the finest mesh of the sub-division hierarchy. By doing so, we make sure all hierarchies share the same initial discretization domain. *i.e.*, all hierarchies solve the same linear system defined on the same input mesh.

6.2 Evaluation

Using the ‘hat’ 0-form basis, hierarchies, and prolongation matrices described above, we construct the 1-form bases (harmonic-free and Whitney) and 1-form prolongation matrices. Then using these 1-form bases, we compute the 1-form mass and stiffness matrices. For evaluation, we generate the hierarchies at a depth of four (*i.e.*, $L = 4$), and we solve an implicit step of vector diffusion. We randomly select a point on the mesh and set a delta vector field (with an arbitrarily chosen tangent direction) as the input signal to diffuse. We use a direct solver to compute the ground-truth \mathbf{x}^* :

$$\mathbf{A} = \mathbf{M} + \alpha\mathbf{S}, \quad \mathbf{b} = \mathbf{M}\mathbf{x}^*,$$

CHAPTER 6. CONVERGENCE STUDY

where $\mathbf{A}, \mathbf{M}, \mathbf{S} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, are as described in Eq. 3.12, and α is the time-step of the diffusion.

To measure convergence, given an estimate \mathbf{x} , we compute the relative residual error:

$$E(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{x}^*)^\top \mathbf{A}(\mathbf{x} - \mathbf{x}^*)}{(\mathbf{x}^*)^\top \mathbf{A} \mathbf{x}^*}.$$

We set the initial guess $\mathbf{x} = 0$ so that $E(\mathbf{x})$ is always one initially and should decrease to zero during the iterative process. Note that, expressing the estimate \mathbf{x} using the spectral basis, $E(\mathbf{x})$ can be split into the low/high frequency residual errors defined in Eq. 4.1. We use three random seeds and take the average residual measurements as our results.

6.2.1 0-Form System Convergence

For context, we begin by considering scalar field diffusion using the 0-form system, where convergence is known to be good. We use the standard V-Cycle multigrid method with (four iterations of) Gauss-Seidel as the relaxation scheme and no additional speed-ups. We solve the diffusion problem using the three hierarchies – mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP). Fig. 6.5 shows the relative residual errors when solving with different timesteps α . The plots are drawn on linear-log (x-y) axes so that the exponential decay expected of the multigrid method is

CHAPTER 6. CONVERGENCE STUDY

Scalar Field Diffusion Convergence

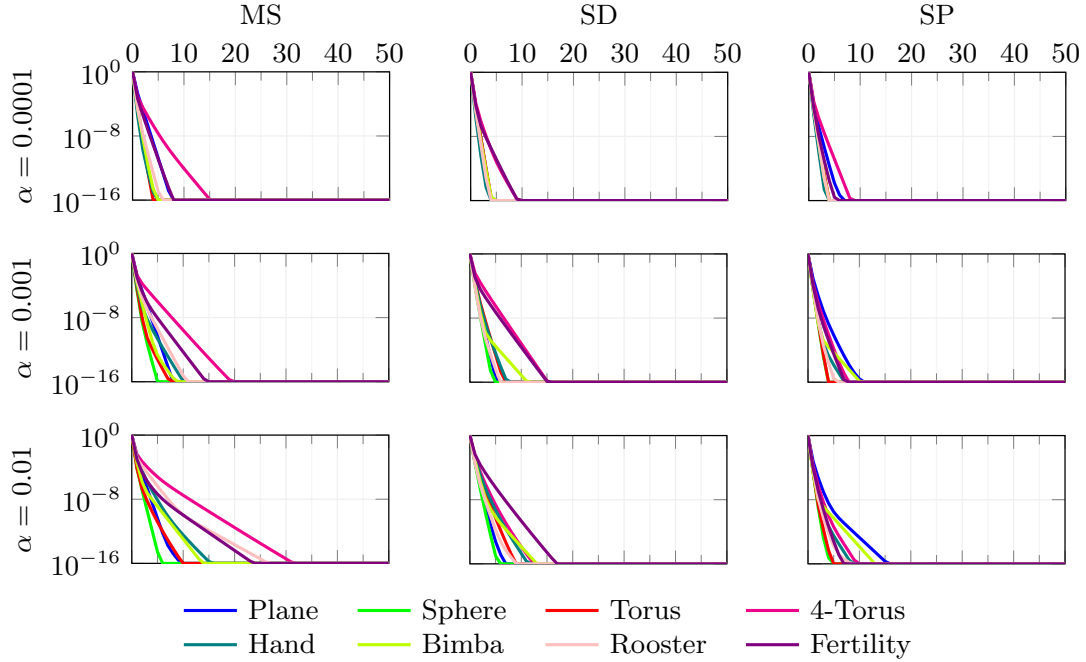


Figure 6.5: The convergence of the standard multigrid method using the ‘hat’ 0-form basis when solving an implicit scalar diffusion step with timestep α varying from 0.0001 to 0.01 using the mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP) hierarchies.

manifest as a curve with a constant slope (up to machine precision – we cap the minimum at 10^{-16} in the plots).

Although performance deteriorates as α increases, the 0-form system consistently achieves errors below 10^{-8} within five iterations using SP and eight iterations using SD. For MS, only the 4-torus model fails to achieve errors below 10^{-8} within eight iterations. We note that MS does not perform as well as the others. Nevertheless, our 0-form solver implementation matches the performance described in literature such as [PKCH18].

6.2.2 1-Form System Convergence

Next, we consider a similar set-up for vector fields. Using the same setting, *i.e.*, a V-Cycle with Gauss-Seidel relaxation and no additional speed-ups, to solve the diffusion problem using the 1-form system with both the harmonic-free and Whitney 1-form basis. We observe a significant deterioration in performance. As shown in Fig. 6.6, this gets worse as α is increased. The negative effect on convergence varies with the topology of the input meshes, the choice of 1-form basis, and the hierarchical structures. In general, the Whitney basis performs better than the harmonic-free basis. However, when using the MS and SP hierarchies, as α is increased, the performance deterioration of the Whitney basis is worse than the harmonic-free basis. Focusing on the Whitney basis, when α is small (bottom first row), some combinations of models and hierarchies (*e.g.*, hand + SD) still achieve errors below 10^{-8} within seven iterations, and all do so within fifty iterations. However, when α is big (bottom last row), no model does⁴.

The convergence for the Whitney basis is consistent across different hierarchies. However, this is not the case when using the harmonic-free basis. For the harmonic-free basis, the SP hierarchy outperforms the others, particularly as α is increased. Aside from the torus, the solver does not achieve 10^{-8} errors within ten iterations.

⁴In Appendix B, we show the convergence plots in six hundred iterations and discuss its convergence in the long run.

CHAPTER 6. CONVERGENCE STUDY

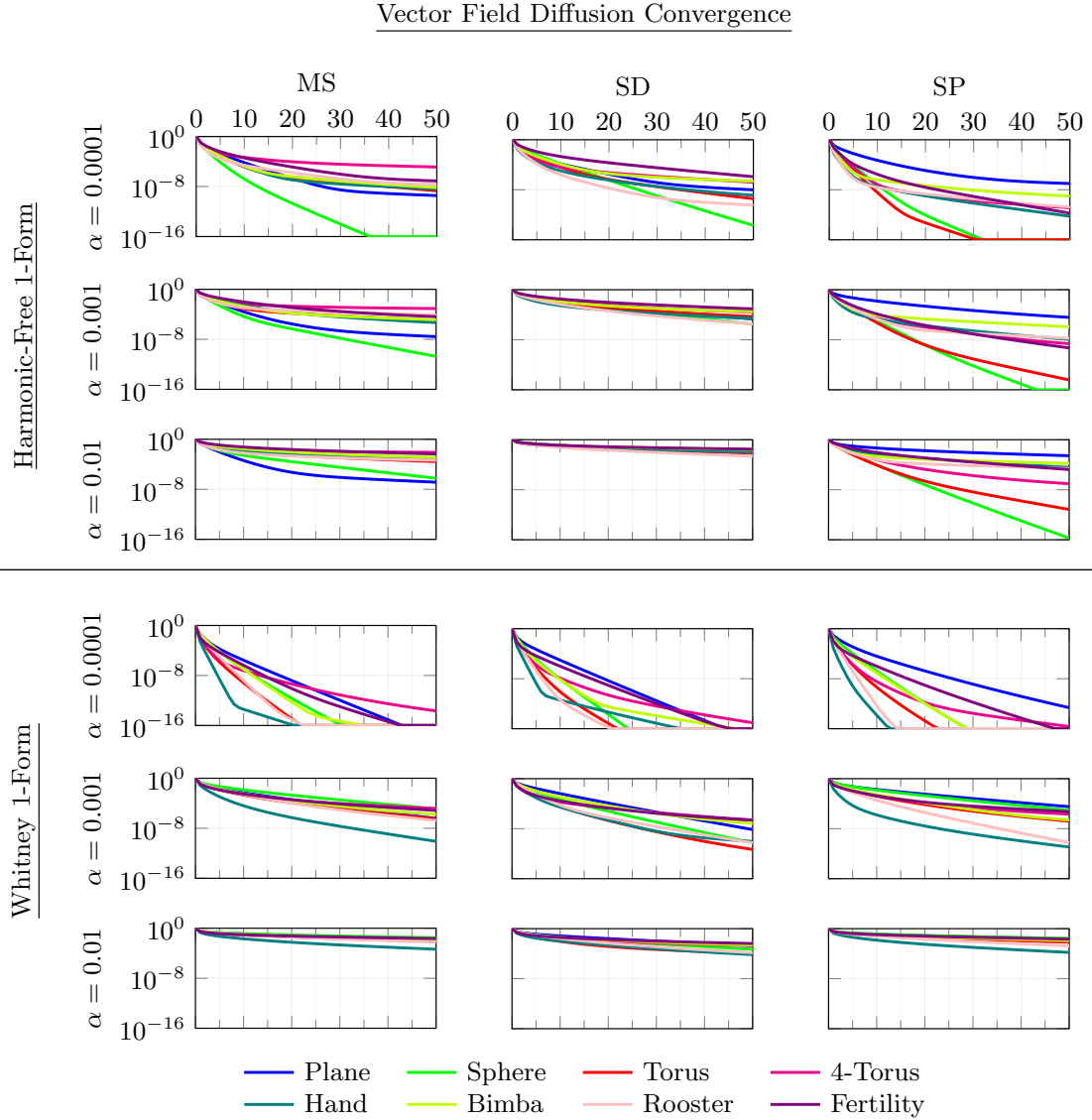


Figure 6.6: The convergence rate of the standard multigrid method when solving an implicit vector diffusion step with varies timestep α from 0.0001 to 0.01 using the hierarchies mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP), as well as the harmonic-free and Whitney 1-form basis.

Notwithstanding the deterioration in convergence, the empirical results show that using the self-intrinsic map does not always improve convergence over other hierarchies for vector field processing, which is consistent with the

CHAPTER 6. CONVERGENCE STUDY

discussion of Liu *et al.* [LZBCJ21, Sec. 8].

	HS	$E(\mathbf{x})$ after 50 itrns.			# of itrns $E(\mathbf{x}) < 10^{-8}$		
		'hat'	harmonic-free	Whitney	'hat'	harmonic-free	Whitney
Plane	MS	$1.4 \cdot 10^{-31}$	$3.6 \cdot 10^{-10}$	$6.1 \cdot 10^{-19}$	4	27	18
	SD	$2.2 \cdot 10^{-31}$	$9.7 \cdot 10^{-9}$	$2.9 \cdot 10^{-18}$	2	50	20
	SP	$2.6 \cdot 10^{-31}$	$1.0 \cdot 10^{-7}$	$2.3 \cdot 10^{-13}$	3	> 50	26
Sphere	MS	$2.6 \cdot 10^{-31}$	$4.5 \cdot 10^{-21}$	$1.8 \cdot 10^{-24}$	2	15	13
	SD	$2.2 \cdot 10^{-31}$	$2.0 \cdot 10^{-14}$	$1.4 \cdot 10^{-30}$	2	27	10
	SP	$1.7 \cdot 10^{-31}$	$1.6 \cdot 10^{-22}$	$9.2 \cdot 10^{-26}$	2	12	13
Torus	MS	$1.6 \cdot 10^{-31}$	$2.1 \cdot 10^{-9}$	$2.7 \cdot 10^{-27}$	2	40	9
	SD	$1.3 \cdot 10^{-31}$	$3.8 \cdot 10^{-10}$	$3.1 \cdot 10^{-26}$	2	35	8
	SP	$1.2 \cdot 10^{-31}$	$1.3 \cdot 10^{-21}$	$8.2 \cdot 10^{-27}$	2	10	8
4-Torus	MS	$2.4 \cdot 10^{-31}$	$1.3 \cdot 10^{-5}$	$2.1 \cdot 10^{-14}$	6	> 50	14
	SD	$2.1 \cdot 10^{-31}$	$1.6 \cdot 10^{-7}$	$9.0 \cdot 10^{-16}$	4	> 50	11
	SP	$2.5 \cdot 10^{-31}$	$1.3 \cdot 10^{-11}$	$2.2 \cdot 10^{-16}$	4	14	9
Hand	MS	$2.7 \cdot 10^{-32}$	$3.9 \cdot 10^{-9}$	$6.5 \cdot 10^{-22}$	2	40	5
	SD	$3.3 \cdot 10^{-32}$	$1.3 \cdot 10^{-9}$	$3.7 \cdot 10^{-19}$	2	37	5
	SP	$2.0 \cdot 10^{-32}$	$5.9 \cdot 10^{-13}$	$1.5 \cdot 10^{-26}$	2	14	5
Bimba	MS	$1.2 \cdot 10^{-31}$	$7.1 \cdot 10^{-9}$	$9.2 \cdot 10^{-19}$	2	47	12
	SD	$8.3 \cdot 10^{-32}$	$2.4 \cdot 10^{-7}$	$4.1 \cdot 10^{-17}$	2	> 50	10
	SP	$7.7 \cdot 10^{-32}$	$9.8 \cdot 10^{-10}$	$3.0 \cdot 10^{-22}$	2	30	12
Rooster	MS	$5.2 \cdot 10^{-32}$	$2.1 \cdot 10^{-8}$	$9.0 \cdot 10^{-25}$	3	> 50	9
	SD	$6.8 \cdot 10^{-32}$	$3.5 \cdot 10^{-11}$	$1.2 \cdot 10^{-21}$	2	22	7
	SP	$8.9 \cdot 10^{-32}$	$1.7 \cdot 10^{-11}$	$3.1 \cdot 10^{-30}$	2	13	7
Fertility	MS	$1.1 \cdot 10^{-31}$	$7.9 \cdot 10^{-8}$	$3.3 \cdot 10^{-18}$	4	> 50	15
	SD	$1.8 \cdot 10^{-31}$	$1.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-17}$	3	> 50	17
	SP	$1.3 \cdot 10^{-31}$	$1.8 \cdot 10^{-12}$	$2.4 \cdot 10^{-17}$	3	23	17

Table 6.1: This table shows the relative residual errors of the scalar field diffusion ('hat' basis), vector field diffusion (harmonic-free and Whitney bases) for timestep $\alpha = 0.0001$ after 50 iterations of multigrid V-Cycle using three hierarchical structures (HS) and the number of iterations required to reduce the relative residual error below 10^{-8} .

CHAPTER 6. CONVERGENCE STUDY

Table 6.1 summarizes and compares the convergence between the 0-form and 1-form systems for $\alpha = 0.0001$. We compare the convergence in two ways. First, we evaluate the relative residual error after a fixed number of iterations. We use fifty iterations and list the results in the column titled ‘ $E(\mathbf{x})$ after 50 itrs.’. Second, we check the number of iterations required to achieve a target relative residual error. In this case, we choose 10^{-8} (the accuracy level that we consider the solver converges) and lists the results in the column titled ‘# of itrs $E(\mathbf{x}) < 10^{-8}$ ’.

The 0-form system takes at most six iterations to converge across all combinations, but the performance of the 1-form systems deteriorates significantly – using the Whitney basis requires as many as twenty-six iterations, and using harmonic-free can sometimes fail to converge. Similarly, after fifty iterations, the relative residual errors of the 1-form systems are much higher than those of the 0-form.

Our goal is to speed up the convergence of the 1-form systems to obtain performance comparable to that of the 0-form system, *i.e.*, to converge (at the accuracy of 10^{-8}) within **six** iterations.

6.3 Multigrid Components

To design an efficient hierarchical solver for vector field processing (as fast as the scalar field processing), we need to address the problem of poor convergence. We consider the different multigrid components and explore the different options and speed-up techniques described in Chapter 4. We analyze each component and choose the best multigrid setting for faster convergence. In our coming discussion, we use the fertility model as an example because it performs poorly for both bases and has a more interesting topology than the worst model (plane). We also focus the discussion on $\alpha = 0.0001$. Appendix C provides details for all models and different timesteps.

6.3.1 Relaxation Schemes

As described in Section 4.2.1, we compare the damped Jacobi (DJ), successive over-relaxation (SOR), and conjugate gradient (CG) methods as the relaxation scheme. We analyze the performance of these schemes in two ways. First, we evaluate the extent to which these schemes speed up the solver. Second, we follow the best practice suggestions in [BL11] and use spectral decomposition to study the convergence (Eq. 4.1), *i.e.*, how effectively individual schemes reduce the low/high-frequency components of the residual.

Relaxation Convergence Comparison

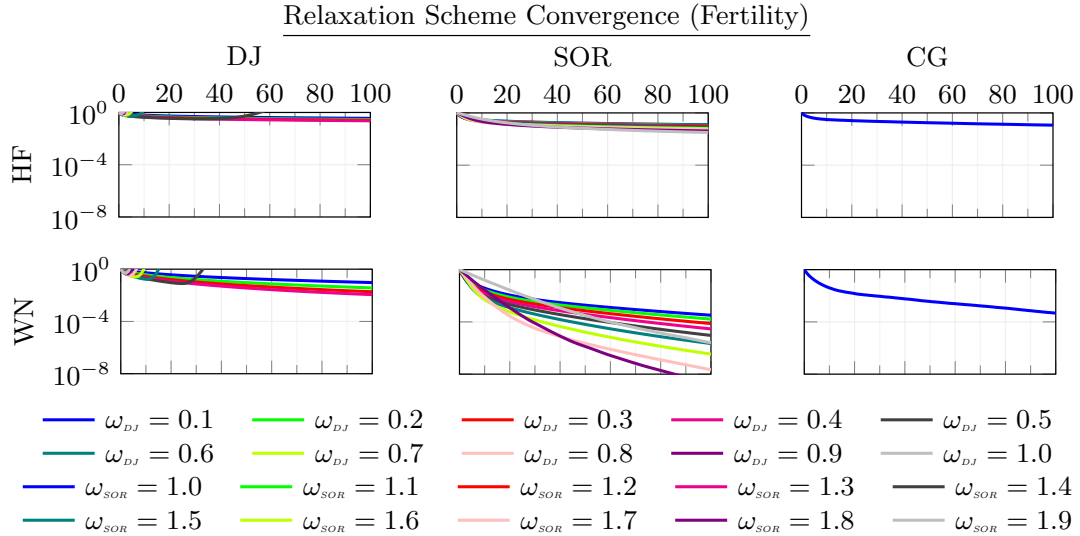


Figure 6.7: The convergence of damped Jacobi (DJ), successive over-relaxation (SOR), and conjugate gradient (CG) methods when solving an implicit vector diffusion step ($\alpha = 0.0001$) on the fertility model using the harmonic-free (HF) basis and the Whitney (WN) basis, for various relaxation weights $\omega_{DJ}, \omega_{SOR}$.

To evaluate the convergence of different relaxation schemes, we solve the same diffusion problem described above. However, we solve it directly on the input models instead of using the hierarchical solver. We find that SOR converges faster than DJ and CG.

Fig. 6.7 shows the convergence of DJ with $\omega_{DJ} = 0.1, 0.2 \dots, 1$, SOR with $\omega_{SOR} = 1, 1.1, \dots, 1.9$, and CG methods for solving the diffusion problem on the fertility model. We find that, for the harmonic-free basis, the convergence of the relaxation schemes appears to be independent of the relaxation weights; while, for the Whitney basis, convergence performance varies with the weights.

The DJ methods converge best with a small ω_{DJ} (less than 0.5) and SOR con-

CHAPTER 6. CONVERGENCE STUDY

verges more quickly than the other relaxation schemes for all over-relaxation weights ω_{SOR} . The results indicate that replacing the Gauss-Seidel relaxation scheme of the multigrid method with SOR could speed up the convergence.

Convergence Improvement using SOR

Fixing SOR as the relaxation scheme, we return to the question of the solver's performance. We find that, depending on the input models and the choice of the hierarchy and basis, the best relaxation weight ranges from 1.4 to 1.9. Table 6.2 shows the improvement of using SOR with $\omega_{SOR} = 1.7$ (convergence plots of different ω_{SOR} are provided in Appendix C.) The table consists of six columns. The first two columns describe the models and the hierarchies (HS). The third and fourth columns show the changes in the relative residual errors after fifty iterations. In contrast, the last two columns show the improvement in the number of iterations required for convergence (errors below 10^{-8}). Both are presented in the format of *before* \Rightarrow *after*.

When comparing the relative residual errors after fifty iterations, on average, using SOR has over 10^6 and 10^3 orders of magnitude improvement for the Whitney and harmonic-free basis. We notice that, for the rooster model with the SP hierarchy, the residual error goes up after using SOR. However, it is still significantly small.

Overall, using SOR as the relaxation scheme significantly improves conver-

CHAPTER 6. CONVERGENCE STUDY

	HS	$E(\mathbf{x})$ after 50 itrs.		# of itrs $E(\mathbf{x}) < 10^{-8}$	
		harmonic-free	Whitney	harmonic-free	Whitney
Plane	MS	$3.6 \cdot 10^{-10} \Rightarrow 1.5 \cdot 10^{-16}$	$6.1 \cdot 10^{-19} \Rightarrow 9.5 \cdot 10^{-31}$	27 \Rightarrow 7	18 \Rightarrow 5
	SD	$9.7 \cdot 10^{-09} \Rightarrow 3.9 \cdot 10^{-09}$	$2.9 \cdot 10^{-18} \Rightarrow 1.5 \cdot 10^{-27}$	50 \Rightarrow 43	20 \Rightarrow 5
	SP	$1.0 \cdot 10^{-07} \Rightarrow 3.7 \cdot 10^{-10}$	$2.3 \cdot 10^{-13} \Rightarrow 9.6 \cdot 10^{-29}$	> 50 \Rightarrow 29	26 \Rightarrow 5
Sphere	MS	$4.5 \cdot 10^{-21} \Rightarrow 2.4 \cdot 10^{-24}$	$1.8 \cdot 10^{-24} \Rightarrow 1.3 \cdot 10^{-30}$	13 \Rightarrow 6	13 \Rightarrow 7
	SD	$2.0 \cdot 10^{-14} \Rightarrow 6.5 \cdot 10^{-24}$	$1.4 \cdot 10^{-30} \Rightarrow 1.4 \cdot 10^{-30}$	27 \Rightarrow 14	10 \Rightarrow 5
	SP	$1.6 \cdot 10^{-22} \Rightarrow 6.9 \cdot 10^{-24}$	$9.2 \cdot 10^{-26} \Rightarrow 1.1 \cdot 10^{-30}$	12 \Rightarrow 6	13 \Rightarrow 7
Torus	MS	$2.1 \cdot 10^{-09} \Rightarrow 1.6 \cdot 10^{-12}$	$2.7 \cdot 10^{-27} \Rightarrow 8.9 \cdot 10^{-31}$	40 \Rightarrow 29	9 \Rightarrow 6
	SD	$3.8 \cdot 10^{-10} \Rightarrow 1.1 \cdot 10^{-13}$	$3.1 \cdot 10^{-26} \Rightarrow 1.4 \cdot 10^{-30}$	35 \Rightarrow 22	8 \Rightarrow 5
	SP	$1.3 \cdot 10^{-21} \Rightarrow 1.1 \cdot 10^{-24}$	$8.2 \cdot 10^{-27} \Rightarrow 1.3 \cdot 10^{-30}$	10 \Rightarrow 6	8 \Rightarrow 6
4-Torus	MS	$1.3 \cdot 10^{-05} \Rightarrow 7.6 \cdot 10^{-06}$	$2.1 \cdot 10^{-14} \Rightarrow 1.0 \cdot 10^{-18}$	> 50 \Rightarrow > 50	14 \Rightarrow 14
	SD	$1.6 \cdot 10^{-07} \Rightarrow 8.9 \cdot 10^{-08}$	$9.0 \cdot 10^{-16} \Rightarrow 2.7 \cdot 10^{-21}$	> 50 \Rightarrow > 50	11 \Rightarrow 12
	SP	$1.3 \cdot 10^{-11} \Rightarrow 1.3 \cdot 10^{-12}$	$2.2 \cdot 10^{-16} \Rightarrow 3.6 \cdot 10^{-24}$	14 \Rightarrow 22	9 \Rightarrow 9
Hand	MS	$3.9 \cdot 10^{-09} \Rightarrow 2.1 \cdot 10^{-12}$	$6.5 \cdot 10^{-22} \Rightarrow 6.3 \cdot 10^{-19}$	40 \Rightarrow 16	5 \Rightarrow 4
	SD	$1.3 \cdot 10^{-09} \Rightarrow 6.0 \cdot 10^{-12}$	$3.7 \cdot 10^{-19} \Rightarrow 4.8 \cdot 10^{-24}$	37 \Rightarrow 21	5 \Rightarrow 4
	SP	$5.9 \cdot 10^{-13} \Rightarrow 8.5 \cdot 10^{-15}$	$1.5 \cdot 10^{-26} \Rightarrow 1.9 \cdot 10^{-24}$	14 \Rightarrow 8	5 \Rightarrow 4
Bimba	MS	$7.1 \cdot 10^{-09} \Rightarrow 2.6 \cdot 10^{-09}$	$9.2 \cdot 10^{-19} \Rightarrow 3.3 \cdot 10^{-24}$	47 \Rightarrow 34	12 \Rightarrow 6
	SD	$2.4 \cdot 10^{-07} \Rightarrow 5.1 \cdot 10^{-08}$	$4.1 \cdot 10^{-17} \Rightarrow 2.3 \cdot 10^{-23}$	> 50 \Rightarrow > 50	10 \Rightarrow 5
	SP	$9.8 \cdot 10^{-10} \Rightarrow 9.1 \cdot 10^{-10}$	$3.0 \cdot 10^{-22} \Rightarrow 1.7 \cdot 10^{-28}$	30 \Rightarrow 24	12 \Rightarrow 5
Rooster	MS	$2.1 \cdot 10^{-08} \Rightarrow 3.3 \cdot 10^{-08}$	$9.0 \cdot 10^{-25} \Rightarrow 3.9 \cdot 10^{-27}$	> 50 \Rightarrow > 50	9 \Rightarrow 5
	SD	$3.5 \cdot 10^{-11} \Rightarrow 2.1 \cdot 10^{-10}$	$1.2 \cdot 10^{-21} \Rightarrow 6.4 \cdot 10^{-25}$	22 \Rightarrow 14	7 \Rightarrow 4
	SP	$1.7 \cdot 10^{-11} \Rightarrow 5.5 \cdot 10^{-11}$	$3.1 \cdot 10^{-30} \Rightarrow 7.8 \cdot 10^{-23}$	13 \Rightarrow 12	7 \Rightarrow 4
Fertility	MS	$7.9 \cdot 10^{-08} \Rightarrow 1.2 \cdot 10^{-09}$	$3.3 \cdot 10^{-18} \Rightarrow 3.0 \cdot 10^{-29}$	> 50 \Rightarrow 30	15 \Rightarrow 8
	SD	$1.2 \cdot 10^{-06} \Rightarrow 1.1 \cdot 10^{-09}$	$3.2 \cdot 10^{-17} \Rightarrow 1.6 \cdot 10^{-27}$	> 50 \Rightarrow 41	17 \Rightarrow 8
	SP	$1.8 \cdot 10^{-12} \Rightarrow 6.0 \cdot 10^{-22}$	$2.4 \cdot 10^{-17} \Rightarrow 6.8 \cdot 10^{-30}$	23 \Rightarrow 14	17 \Rightarrow 9

Table 6.2: Convergence improvement using SOR with the over-relaxation weight $\omega_{SOR} = 1.7$. The third and fourth columns show the changes in the relative residual errors after fifty iterations before and after the speed up, and the last two columns show the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .

CHAPTER 6. CONVERGENCE STUDY

gence for all three hierarchies. For the Whitney basis, five of the eight models achieve the desired convergence (errors below 10^{-8} within six iterations), while the sphere requires an additional iteration and the fertility needs two to three more iterations.

We note that although the different relaxation schemes did not improve convergence when used in a single-level hierarchy, they do appear to improve convergence when used in a multi-level hierarchy. Details can be found in Appendix C (Fig. C.3 and Table C.1.) A few of the combinations (*e.g.*, torus + SP) even achieve the desired convergence; however, some combinations (such as bimba + SD) still fail to converge within fifty iterations.

Spectral Analysis

We follow the recommendation of Brandt and Lvine [BL11] and use spectral analysis to separately study the low- and high-frequency residual errors $E_l(\mathbf{x})$ and $E_h(\mathbf{x})$ (Eq. 4.1) for the first fifty iterations, comparing using Gauss-Seidel ($\omega_{SOR} = 1$) and SOR ($\omega_{SOR} = 1.7$) as the relaxation scheme in the hierarchical solver. As before, we use the Whitney 1-form basis and solve the diffusion problem on the fertility model.

The solid and dotted lines in Fig. 6.8 are $E_l(\mathbf{x})$ (blue) and $E_h(\mathbf{x})$ (green) of Gauss-Seidel and SOR, respectively. We observe that the convergence of $E_l(\mathbf{x})$ and $E_h(\mathbf{x})$ are similar in both cases (the gaps between $E_l(\mathbf{x})$ and $E_h(\mathbf{x})$ are the

CHAPTER 6. CONVERGENCE STUDY

same before and after the speedup), across different hierarchies and timesteps α , meaning that using SOR as the relaxation scheme robustly speeds up the solver. The speed up is independent of the choice of the hierarchies and the solution's frequency distribution in the spectral basis.

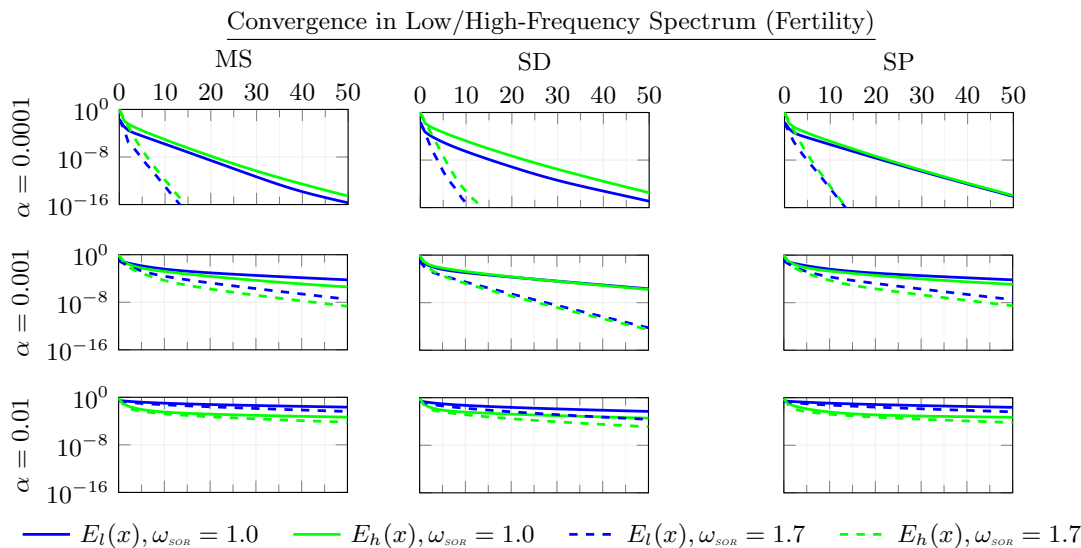


Figure 6.8: The convergence in low/high-frequency spectrum of solving an implicit vector diffusion step with various timestep α on the fertility model using the Whitney basis using Gauss-Seidel (solid lines) and SOR (dotted lines) as the relaxation scheme.

6.3.2 Prolongation

Next, we analyze the prolongation matrix using Eq. 4.3 and plot the low-frequency correlation matrix using the eigenfunctions associated with the first 200 smallest eigenvalues. We observe that these matrices seem to be correlated to the solver convergence.

CHAPTER 6. CONVERGENCE STUDY

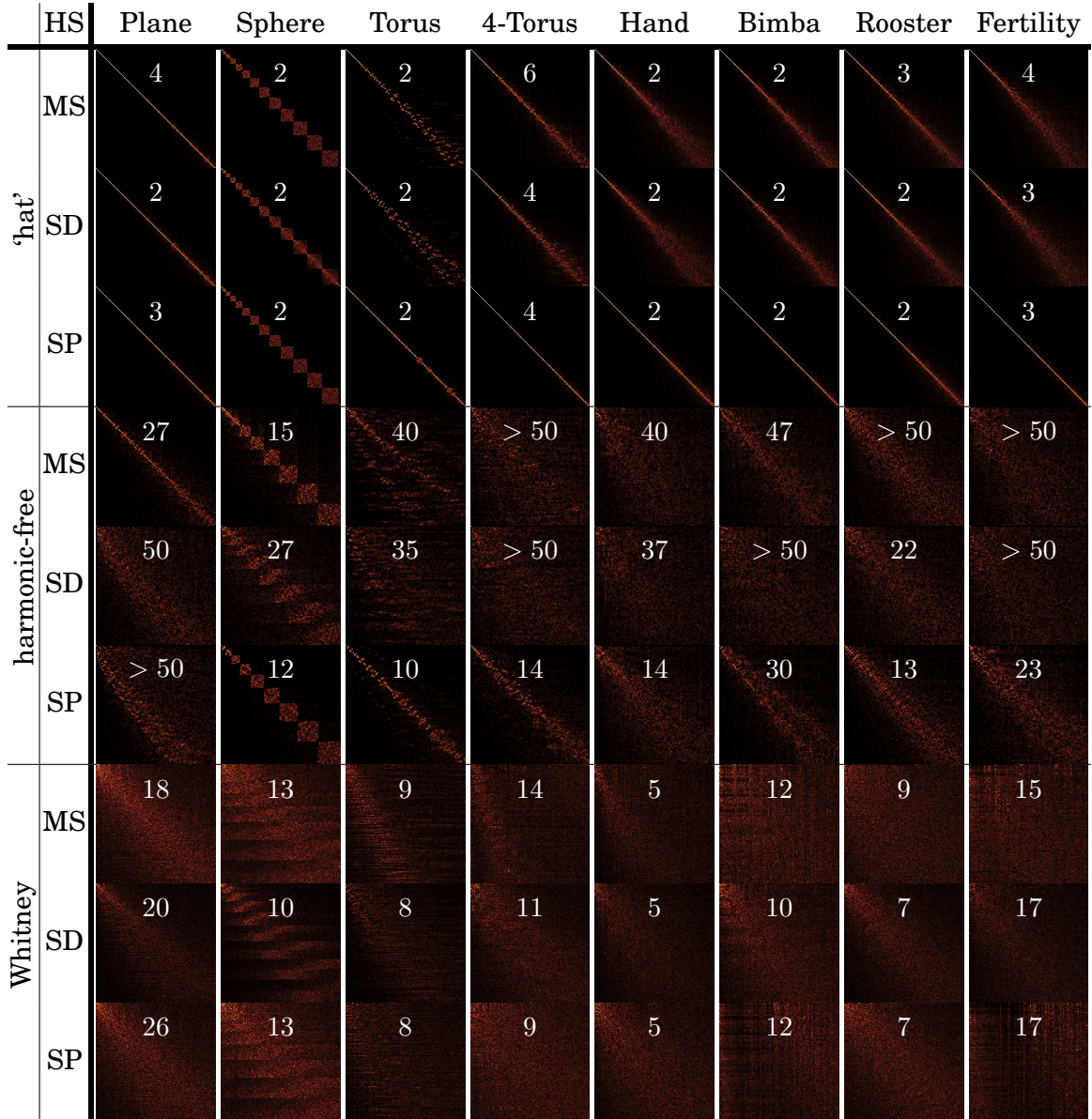


Figure 6.9: Correlation matrices (Eq. 4.3) coarsest-finest visualized as images of all combinations. Black body colormap is used to visualize 0 as black, and 1 as white.

Fig. 6.9 shows these correlation matrices (normalized by the maximum matrix entry), as images (using black body colormap to visualize 0 as black, and 1 as white), of the coarsest level ($l = 4$) to the finest level ($l = 1$) of all com-

CHAPTER 6. CONVERGENCE STUDY

binations. The number of iterations to reduce the residual error below 10^{-8} is displayed on top of the images. Recall that Eq. 4.3 are also the coefficients of the coarsest level eigenfunctions expressed with respect to the finest level eigenfunctions. Therefore, by examining these low-frequency correction matrices, we obtain an understanding to what extent the low-frequency eigenfunctions at the coarsest level are composed of the low-frequency eigenfunctions at the finest level.

For the 0-form system (the ‘hat’ basis), the low-frequency correlation matrices are almost diagonal, meaning that the low-frequency coarse eigenfunctions are mainly composed of the low-frequency fine eigenfunctions. In this case, the coarse system primarily solves the low-frequency of the fine system. Notice that the superior performance of SP is reflected by these correlation matrices – of the three hierarchies, the matrix generated by SP is the most diagonal one.

As expected, given the less efficient convergence, most of these 1-form correlation matrices (the harmonic-free and Whitney basis) are not close to diagonal. Yet, when they are close, such as the combinations plane + MS and torus + SP using the harmonic-free basis, their convergence also tends to be better (*i.e.*, plane + SD/SP and torus + MS/SD.) This seems to be true also for the Whitney basis, although the low-frequency correlation matrices are more evenly spread. For example, sphere + SD converges faster than sphere + MS/SP. For the Whitney basis, the shape of the “diagonalness” is similar among MS/SD/SP, which

CHAPTER 6. CONVERGENCE STUDY

is reflected in their similar convergence.

We note that while the low-frequency correlation matrices could appear to be predictive of the relative performance for different variants of a single basis, they are less predictive when comparing the behavior of different bases. For example, using the Whitney basis gives faster convergence, but their low-frequency correlation matrices appear less diagonal than those of the harmonic-free basis. Our analysis focuses on the eigenfunctions associated with the first 200 smallest eigenvalues. It may be interesting to analyze the correlation matrix using the full spectral basis.

Changes in Convergence using Prolongation Smoothing

As described in Section 4.2.2, there is a technique from the algebraic multigrid method to smooth the prolongation matrix using damped Jacobi. In our exploration of this technique, we did not find it universally improved either the “diagonalness” of the correlation matrices or the convergence of the solver. Specifically, for the harmonic-free basis, it destroys the diagonal structure and does not improve the convergence. However, it does help in the case of the Whitney basis for some models. Table 6.3 presents the changes in convergence of using prolongation smoothing with the damping weight $\omega_{d,j} = 0.3$. Similar to Table 6.2, the table shows the changes in the relative residual errors after fifty iterations and the number of iterations required for convergence (*i.e.*, errors

CHAPTER 6. CONVERGENCE STUDY

	HS	$E(\mathbf{x})$ after 50 itrs.		# of itrs $E(\mathbf{x}) < 10^{-8}$	
		harmonic-free	Whitney	harmonic-free	Whitney
Plane	MS	$3.6 \cdot 10^{-10} \Rightarrow 1.8 \cdot 10^{-03}$	$6.1 \cdot 10^{-19} \Rightarrow 1.3 \cdot 10^{-26}$	27 \Rightarrow > 50	18 \Rightarrow 6
	SD	$9.7 \cdot 10^{-09} \Rightarrow 4.0 \cdot 10^{-03}$	$2.9 \cdot 10^{-18} \Rightarrow 8.7 \cdot 10^{-18}$	50 \Rightarrow > 50	20 \Rightarrow 7
	SP	$1.0 \cdot 10^{-07} \Rightarrow 1.1 \cdot 10^{-03}$	$2.3 \cdot 10^{-13} \Rightarrow 2.6 \cdot 10^{-22}$	> 50 \Rightarrow > 50	26 \Rightarrow 6
Sphere	MS	$4.5 \cdot 10^{-21} \Rightarrow 1.9 \cdot 10^{-03}$	$1.8 \cdot 10^{-24} \Rightarrow 9.2 \cdot 10^{-31}$	13 \Rightarrow > 50	13 \Rightarrow 9
	SD	$2.0 \cdot 10^{-14} \Rightarrow 8.2 \cdot 10^{-03}$	$1.4 \cdot 10^{-30} \Rightarrow 1.7 \cdot 10^{-30}$	27 \Rightarrow > 50	10 \Rightarrow 10
	SP	$1.6 \cdot 10^{-22} \Rightarrow 1.7 \cdot 10^{-03}$	$9.2 \cdot 10^{-26} \Rightarrow 6.8 \cdot 10^{-31}$	12 \Rightarrow > 50	13 \Rightarrow 9
Torus	MS	$2.1 \cdot 10^{-09} \Rightarrow 1.6 \cdot 10^{-03}$	$2.7 \cdot 10^{-27} \Rightarrow 1.2 \cdot 10^{-21}$	40 \Rightarrow > 50	9 \Rightarrow 15
	SD	$3.8 \cdot 10^{-10} \Rightarrow 1.1 \cdot 10^{-02}$	$3.1 \cdot 10^{-26} \Rightarrow 3.3 \cdot 10^{-20}$	35 \Rightarrow > 50	8 \Rightarrow 16
	SP	$1.3 \cdot 10^{-21} \Rightarrow 9.2 \cdot 10^{-04}$	$8.2 \cdot 10^{-27} \Rightarrow 9.6 \cdot 10^{-31}$	10 \Rightarrow > 50	8 \Rightarrow 9
4-Torus	MS	$1.3 \cdot 10^{-05} \Rightarrow 3.3 \cdot 10^{-03}$	$2.1 \cdot 10^{-14} \Rightarrow 2.2 \cdot 10^{-10}$	> 50 \Rightarrow > 50	14 \Rightarrow 34
	SD	$1.6 \cdot 10^{-07} \Rightarrow 1.3 \cdot 10^{-02}$	$9.0 \cdot 10^{-16} \Rightarrow 1.4 \cdot 10^{-09}$	> 50 \Rightarrow > 50	11 \Rightarrow 41
	SP	$1.3 \cdot 10^{-11} \Rightarrow 8.0 \cdot 10^{-04}$	$2.2 \cdot 10^{-16} \Rightarrow 3.7 \cdot 10^{-14}$	14 \Rightarrow > 50	9 \Rightarrow 19
Hand	MS	$3.9 \cdot 10^{-09} \Rightarrow 5.5 \cdot 10^{-05}$	$6.5 \cdot 10^{-22} \Rightarrow 3.3 \cdot 10^{-24}$	40 \Rightarrow > 50	5 \Rightarrow 7
	SD	$1.3 \cdot 10^{-09} \Rightarrow 7.3 \cdot 10^{-04}$	$3.7 \cdot 10^{-19} \Rightarrow 6.0 \cdot 10^{-19}$	37 \Rightarrow > 50	5 \Rightarrow 6
	SP	$5.9 \cdot 10^{-13} \Rightarrow 2.0 \cdot 10^{-05}$	$1.5 \cdot 10^{-26} \Rightarrow 1.7 \cdot 10^{-28}$	14 \Rightarrow > 50	5 \Rightarrow 4
Bimba	MS	$7.1 \cdot 10^{-09} \Rightarrow 2.1 \cdot 10^{-03}$	$9.2 \cdot 10^{-19} \Rightarrow 7.7 \cdot 10^{-21}$	47 \Rightarrow > 50	12 \Rightarrow 10
	SD	$2.4 \cdot 10^{-07} \Rightarrow 5.2 \cdot 10^{-03}$	$4.1 \cdot 10^{-17} \Rightarrow 1.2 \cdot 10^{-16}$	> 50 \Rightarrow > 50	10 \Rightarrow 11
	SP	$9.8 \cdot 10^{-10} \Rightarrow 1.4 \cdot 10^{-03}$	$3.0 \cdot 10^{-22} \Rightarrow 6.4 \cdot 10^{-24}$	30 \Rightarrow > 50	12 \Rightarrow 9
Rooster	MS	$2.1 \cdot 10^{-08} \Rightarrow 3.2 \cdot 10^{-04}$	$9.0 \cdot 10^{-25} \Rightarrow 3.7 \cdot 10^{-27}$	> 50 \Rightarrow > 50	9 \Rightarrow 7
	SD	$3.5 \cdot 10^{-11} \Rightarrow 2.3 \cdot 10^{-03}$	$1.2 \cdot 10^{-21} \Rightarrow 5.5 \cdot 10^{-24}$	22 \Rightarrow > 50	7 \Rightarrow 7
	SP	$1.7 \cdot 10^{-11} \Rightarrow 6.7 \cdot 10^{-05}$	$3.1 \cdot 10^{-30} \Rightarrow 2.9 \cdot 10^{-31}$	13 \Rightarrow > 50	7 \Rightarrow 5
Fertility	MS	$7.9 \cdot 10^{-08} \Rightarrow 1.5 \cdot 10^{-03}$	$3.3 \cdot 10^{-18} \Rightarrow 7.9 \cdot 10^{-18}$	> 50 \Rightarrow > 50	15 \Rightarrow 17
	SD	$1.2 \cdot 10^{-06} \Rightarrow 7.4 \cdot 10^{-03}$	$3.2 \cdot 10^{-17} \Rightarrow 1.7 \cdot 10^{-14}$	> 50 \Rightarrow > 50	17 \Rightarrow 23
	SP	$1.8 \cdot 10^{-12} \Rightarrow 4.9 \cdot 10^{-04}$	$2.4 \cdot 10^{-17} \Rightarrow 2.5 \cdot 10^{-22}$	23 \Rightarrow > 50	17 \Rightarrow 14

Table 6.3: This table shows the changes in convergence using prolongation smoothing with the damping weight $\omega_{Dj} = 0.3$. The third and fourth columns are the changes in the relative residual errors after fifty iterations, and the last two columns are the changes in the number of iterations required to obtain accuracy below 10^{-8} .

CHAPTER 6. CONVERGENCE STUDY

below 10^{-8}) of all variants of models, hierarchies, and bases.

As prolongation smoothing destroys the diagonal structure of the correlation matrices for the harmonic-free basis, we observe much slower convergence for this basis. No combination converges within fifty iterations, and the relative residual errors are much higher than without using the smoothing. Yet, for the Whitney basis, prolongation smoothing improves the convergence of the plane model significantly by a factor of three. However, it does not significantly improve solver performance for other models (*e.g.*, rooster), and it slows down the performance for some models (*e.g.*, torus). Appendix C provides plots of convergences using prolongation smoothing with different damping weights $\omega_{D,J}$ and timesteps α .

6.3.3 Solution Update

In addition to classical techniques from the literature, we propose applying Krylov subspace updates to the multigrid method as described in Section 4.2.3. This is a low-cost update method as we keep the subspace dimension small compared to the dimension of the system. We evaluate the extent to which this improves performance, varying the subspace dimension from 1 to 15.

CHAPTER 6. CONVERGENCE STUDY

	HS	$E(\mathbf{x})$ after 50 itr.		# of itr $E(\mathbf{x}) < 10^{-8}$	
		harmonic-free	Whitney	harmonic-free	Whitney
Plane	MS	$3.6 \cdot 10^{-10} \Rightarrow 1.5 \cdot 10^{-15}$	$6.1 \cdot 10^{-19} \Rightarrow 3.9 \cdot 10^{-24}$	27 \Rightarrow 7	18 \Rightarrow 5
	SD	$9.7 \cdot 10^{-09} \Rightarrow 1.0 \cdot 10^{-14}$	$2.9 \cdot 10^{-18} \Rightarrow 1.0 \cdot 10^{-20}$	50 \Rightarrow 15	20 \Rightarrow 4
	SP	$1.0 \cdot 10^{-07} \Rightarrow 2.0 \cdot 10^{-14}$	$2.3 \cdot 10^{-13} \Rightarrow 1.6 \cdot 10^{-22}$	> 50 \Rightarrow 13	26 \Rightarrow 5
Sphere	MS	$4.5 \cdot 10^{-21} \Rightarrow 2.2 \cdot 10^{-24}$	$1.8 \cdot 10^{-24} \Rightarrow 1.2 \cdot 10^{-30}$	13 \Rightarrow 6	13 \Rightarrow 7
	SD	$2.0 \cdot 10^{-14} \Rightarrow 4.6 \cdot 10^{-22}$	$1.4 \cdot 10^{-30} \Rightarrow 5.6 \cdot 10^{-31}$	27 \Rightarrow 9	10 \Rightarrow 5
	SP	$1.6 \cdot 10^{-22} \Rightarrow 2.9 \cdot 10^{-24}$	$9.2 \cdot 10^{-26} \Rightarrow 7.1 \cdot 10^{-31}$	12 \Rightarrow 6	13 \Rightarrow 6
Torus	MS	$2.1 \cdot 10^{-09} \Rightarrow 6.7 \cdot 10^{-16}$	$2.7 \cdot 10^{-27} \Rightarrow 1.7 \cdot 10^{-24}$	40 \Rightarrow 16	9 \Rightarrow 8
	SD	$3.8 \cdot 10^{-10} \Rightarrow 8.9 \cdot 10^{-17}$	$3.1 \cdot 10^{-26} \Rightarrow 1.2 \cdot 10^{-25}$	35 \Rightarrow 13	8 \Rightarrow 6
	SP	$1.3 \cdot 10^{-21} \Rightarrow 2.6 \cdot 10^{-24}$	$8.2 \cdot 10^{-27} \Rightarrow 1.5 \cdot 10^{-29}$	10 \Rightarrow 6	8 \Rightarrow 7
4-Torus	MS	$1.3 \cdot 10^{-05} \Rightarrow 1.5 \cdot 10^{-11}$	$2.1 \cdot 10^{-14} \Rightarrow 1.7 \cdot 10^{-18}$	> 50 \Rightarrow 30	14 \Rightarrow 11
	SD	$1.6 \cdot 10^{-07} \Rightarrow 1.2 \cdot 10^{-13}$	$9.0 \cdot 10^{-16} \Rightarrow 2.7 \cdot 10^{-17}$	> 50 \Rightarrow 25	11 \Rightarrow 10
	SP	$1.3 \cdot 10^{-11} \Rightarrow 1.1 \cdot 10^{-15}$	$2.2 \cdot 10^{-16} \Rightarrow 1.3 \cdot 10^{-21}$	14 \Rightarrow 13	9 \Rightarrow 9
Hand	MS	$3.9 \cdot 10^{-09} \Rightarrow 9.8 \cdot 10^{-16}$	$6.5 \cdot 10^{-22} \Rightarrow 1.1 \cdot 10^{-21}$	40 \Rightarrow 11	5 \Rightarrow 4
	SD	$1.3 \cdot 10^{-09} \Rightarrow 9.2 \cdot 10^{-17}$	$3.7 \cdot 10^{-19} \Rightarrow 5.1 \cdot 10^{-21}$	37 \Rightarrow 11	5 \Rightarrow 4
	SP	$5.9 \cdot 10^{-13} \Rightarrow 3.1 \cdot 10^{-19}$	$1.5 \cdot 10^{-26} \Rightarrow 7.4 \cdot 10^{-27}$	14 \Rightarrow 7	5 \Rightarrow 3
Bimba	MS	$7.1 \cdot 10^{-09} \Rightarrow 8.5 \cdot 10^{-15}$	$9.2 \cdot 10^{-19} \Rightarrow 1.2 \cdot 10^{-20}$	47 \Rightarrow 15	12 \Rightarrow 6
	SD	$2.4 \cdot 10^{-07} \Rightarrow 3.5 \cdot 10^{-13}$	$4.1 \cdot 10^{-17} \Rightarrow 3.2 \cdot 10^{-19}$	> 50 \Rightarrow 20	10 \Rightarrow 6
	SP	$9.8 \cdot 10^{-10} \Rightarrow 1.6 \cdot 10^{-14}$	$3.0 \cdot 10^{-22} \Rightarrow 1.9 \cdot 10^{-22}$	30 \Rightarrow 12	12 \Rightarrow 5
Rooster	MS	$2.1 \cdot 10^{-08} \Rightarrow 2.5 \cdot 10^{-12}$	$9.0 \cdot 10^{-25} \Rightarrow 9.2 \cdot 10^{-28}$	> 50 \Rightarrow 16	9 \Rightarrow 5
	SD	$3.5 \cdot 10^{-11} \Rightarrow 2.4 \cdot 10^{-12}$	$1.2 \cdot 10^{-21} \Rightarrow 3.8 \cdot 10^{-23}$	22 \Rightarrow 13	7 \Rightarrow 4
	SP	$1.7 \cdot 10^{-11} \Rightarrow 1.7 \cdot 10^{-12}$	$3.1 \cdot 10^{-30} \Rightarrow 2.0 \cdot 10^{-30}$	13 \Rightarrow 9	7 \Rightarrow 4
Fertility	MS	$7.9 \cdot 10^{-08} \Rightarrow 2.0 \cdot 10^{-14}$	$3.3 \cdot 10^{-18} \Rightarrow 5.3 \cdot 10^{-21}$	> 50 \Rightarrow 17	15 \Rightarrow 8
	SD	$1.2 \cdot 10^{-06} \Rightarrow 1.2 \cdot 10^{-13}$	$3.2 \cdot 10^{-17} \Rightarrow 2.2 \cdot 10^{-20}$	> 50 \Rightarrow 17	17 \Rightarrow 9
	SP	$1.8 \cdot 10^{-12} \Rightarrow 8.1 \cdot 10^{-18}$	$2.4 \cdot 10^{-17} \Rightarrow 2.6 \cdot 10^{-22}$	23 \Rightarrow 9	17 \Rightarrow 9

Table 6.4: Convergence improvement using Krylov subspace updates. The third and fourth columns show the improvement in the relative residual errors after fifty iterations before and after the speed up, and the last two columns are the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .

CHAPTER 6. CONVERGENCE STUDY

Convergence Improvement

In general, keeping the subspace dimension at ten works best for most combinations. We note that when the subspace becomes close to linearly dependent, our implementation does not obtain a numerically stable update. To address this, we skip the update when the numerical solution does not give a smaller residual. In Appendix C, we see from the plots that it happens after around ten to twenty iterations. After inspection, we find that it happens when the determinant of the system matrix of Krylov subspace updates is less than 10^{-16} .

Table 6.4 shows the convergence improvement using Krylov subspace updates with the subspace dimension equal to ten (see Appendix C for convergence plots of different subspace dimensions and timesteps α .) Similar to previous result tables, we show the improvement in the relative residual errors after fifty iterations and in the number of iterations for convergence (*i.e.*, to achieve errors below 10^{-8}).

For the Whitney basis, the improvement of integrating the Krylov subspace update is similar to that of using SOR as the relaxation scheme. The number of iterations to bring the residual errors below 10^{-18} is, on average, reduced by a factor of two to three. Besides the higher-genus models (4-tours and fertility), torus + MS/SP and sphere + MS, for all models the solver converges within six iterations.

CHAPTER 6. CONVERGENCE STUDY

In contrast, for the harmonic-free basis, although most combinations do not meet the six iterations goal, all combinations now converge within fifty iterations. In this case, the improvement of integrating the Krylov subspace update is more significant than using SOR as the relaxation scheme.

In the long run (after fifty iterations), all combinations obtain smaller relative residual errors for the harmonic-free basis. However, for the Whitney basis, we note that the hand + MS and torus + MS/SD encounter a slight slowdown. We note that the increases in errors of these cases are $6.5 \cdot 10^{-22} \Rightarrow 1.1 \cdot 10^{-21}$, $2.7 \cdot 10^{-27} \Rightarrow 1.7 \cdot 10^{-24}$, and $3.1 \cdot 10^{-26} \Rightarrow 1.2 \cdot 10^{-25}$, which are negligibly small.

Spectral Analysis

As above, we also analyze convergence after integrating Krylov subspace updates using spectral analysis. We would like to see if Krylov subspace updates selectively improve either the low- or the high-frequencies. Likewise, we plot the low/high-frequency residual errors ($E_l(\mathbf{x})$ in blue and $E_h(\mathbf{x})$ in green) as shown in Fig. 6.10, where the dotted lines represent results after the integration of Krylov subspace update while the solid lines represent that of the standard multigrid method. We observe a similar behavior when using SOR as the relaxation scheme. The changes in slopes of $E_l(\mathbf{x})$ before and after the speedup are similar to those of $E_h(\mathbf{x})$ across different hierarchies and timesteps.

CHAPTER 6. CONVERGENCE STUDY

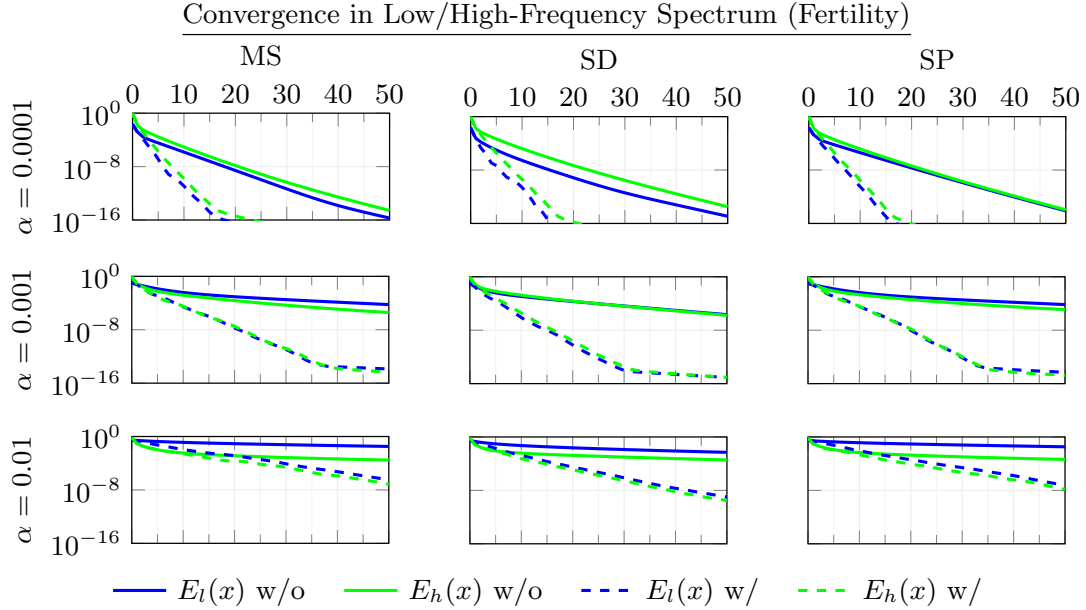


Figure 6.10: The convergence in low/high-frequency spectrum of solving an implicit vector diffusion step with various timestep α on the fertility model using the Whitney basis with (dotted lines) and without (solid lines) integrating Krylov subspace update.

It means that integrating the Krylov subspace update into the solver affects the low- and high-frequencies equilaterally.

6.4 Performance

By combining the above improvements (*i.e.*, SOR, prolongation smoothing, and Krylov subspace updates), we obtain the overall convergence improvement for each combination, as shown in Table 6.5. When using the Whitney basis, we obtain the desired convergence for all models with the exception of the 4-torus (which still needs seven iterations to converge using the SP hierarchy.)

CHAPTER 6. CONVERGENCE STUDY

	HS	$E(\mathbf{x})$ after 50 itr.		# of itr $E(\mathbf{x}) < 10^{-8}$	
		harmonic-free	Whitney	harmonic-free	Whitney
Plane	MS	$3.6 \cdot 10^{-10} \Rightarrow 3.9 \cdot 10^{-21}$	$6.1 \cdot 10^{-19} \Rightarrow 8.0 \cdot 10^{-27}$	27 \Rightarrow 6	18 \Rightarrow 4
	SD	$9.7 \cdot 10^{-09} \Rightarrow 6.1 \cdot 10^{-16}$	$2.9 \cdot 10^{-18} \Rightarrow 6.6 \cdot 10^{-24}$	50 \Rightarrow 12	20 \Rightarrow 4
	SP	$1.0 \cdot 10^{-07} \Rightarrow 6.5 \cdot 10^{-16}$	$2.3 \cdot 10^{-13} \Rightarrow 7.4 \cdot 10^{-30}$	> 50 \Rightarrow 13	26 \Rightarrow 4
Sphere	MS	$4.5 \cdot 10^{-21} \Rightarrow 5.8 \cdot 10^{-24}$	$1.8 \cdot 10^{-24} \Rightarrow 1.3 \cdot 10^{-30}$	13 \Rightarrow 6	13 \Rightarrow 5
	SD	$2.0 \cdot 10^{-14} \Rightarrow 1.8 \cdot 10^{-23}$	$1.4 \cdot 10^{-30} \Rightarrow 8.5 \cdot 10^{-31}$	27 \Rightarrow 11	10 \Rightarrow 4
	SP	$1.6 \cdot 10^{-22} \Rightarrow 6.7 \cdot 10^{-25}$	$9.2 \cdot 10^{-26} \Rightarrow 1.1 \cdot 10^{-30}$	12 \Rightarrow 5	13 \Rightarrow 5
Torus	MS	$2.1 \cdot 10^{-09} \Rightarrow 3.5 \cdot 10^{-18}$	$2.7 \cdot 10^{-27} \Rightarrow 1.2 \cdot 10^{-30}$	40 \Rightarrow 16	9 \Rightarrow 6
	SD	$3.8 \cdot 10^{-10} \Rightarrow 2.7 \cdot 10^{-19}$	$3.1 \cdot 10^{-26} \Rightarrow 8.7 \cdot 10^{-31}$	35 \Rightarrow 15	8 \Rightarrow 5
	SP	$1.3 \cdot 10^{-21} \Rightarrow 4.7 \cdot 10^{-24}$	$8.2 \cdot 10^{-27} \Rightarrow 1.2 \cdot 10^{-30}$	10 \Rightarrow 5	8 \Rightarrow 5
4-Torus	MS	$1.3 \cdot 10^{-05} \Rightarrow 7.0 \cdot 10^{-11}$	$2.1 \cdot 10^{-14} \Rightarrow 1.3 \cdot 10^{-20}$	> 50 \Rightarrow 33	14 \Rightarrow 9
	SD	$1.6 \cdot 10^{-07} \Rightarrow 1.9 \cdot 10^{-14}$	$9.0 \cdot 10^{-16} \Rightarrow 4.4 \cdot 10^{-28}$	> 50 \Rightarrow 25	11 \Rightarrow 8
	SP	$1.3 \cdot 10^{-11} \Rightarrow 1.9 \cdot 10^{-15}$	$2.2 \cdot 10^{-16} \Rightarrow 8.5 \cdot 10^{-25}$	14 \Rightarrow 8	9 \Rightarrow 7
Hand	MS	$3.9 \cdot 10^{-09} \Rightarrow 3.6 \cdot 10^{-20}$	$6.5 \cdot 10^{-22} \Rightarrow 5.4 \cdot 10^{-23}$	40 \Rightarrow 10	5 \Rightarrow 2
	SD	$1.3 \cdot 10^{-09} \Rightarrow 9.2 \cdot 10^{-18}$	$3.7 \cdot 10^{-19} \Rightarrow 6.7 \cdot 10^{-27}$	37 \Rightarrow 11	5 \Rightarrow 2
	SP	$5.9 \cdot 10^{-13} \Rightarrow 5.7 \cdot 10^{-23}$	$1.5 \cdot 10^{-26} \Rightarrow 1.3 \cdot 10^{-31}$	14 \Rightarrow 7	5 \Rightarrow 3
Bimba	MS	$7.1 \cdot 10^{-09} \Rightarrow 5.9 \cdot 10^{-15}$	$9.2 \cdot 10^{-19} \Rightarrow 6.8 \cdot 10^{-26}$	47 \Rightarrow 12	12 \Rightarrow 5
	SD	$2.4 \cdot 10^{-07} \Rightarrow 1.1 \cdot 10^{-12}$	$4.1 \cdot 10^{-17} \Rightarrow 3.5 \cdot 10^{-23}$	> 50 \Rightarrow 24	10 \Rightarrow 5
	SP	$9.8 \cdot 10^{-10} \Rightarrow 1.7 \cdot 10^{-15}$	$3.0 \cdot 10^{-22} \Rightarrow 1.2 \cdot 10^{-27}$	30 \Rightarrow 12	12 \Rightarrow 5
Rooster	MS	$2.1 \cdot 10^{-08} \Rightarrow 8.7 \cdot 10^{-12}$	$9.0 \cdot 10^{-25} \Rightarrow 5.8 \cdot 10^{-31}$	> 50 \Rightarrow 15	9 \Rightarrow 4
	SD	$3.5 \cdot 10^{-11} \Rightarrow 8.8 \cdot 10^{-13}$	$1.2 \cdot 10^{-21} \Rightarrow 1.3 \cdot 10^{-25}$	22 \Rightarrow 17	7 \Rightarrow 4
	SP	$1.7 \cdot 10^{-11} \Rightarrow 5.6 \cdot 10^{-13}$	$3.1 \cdot 10^{-30} \Rightarrow 2.1 \cdot 10^{-29}$	13 \Rightarrow 7	7 \Rightarrow 3
Fertility	MS	$7.9 \cdot 10^{-08} \Rightarrow 6.7 \cdot 10^{-16}$	$3.3 \cdot 10^{-18} \Rightarrow 4.2 \cdot 10^{-30}$	> 50 \Rightarrow 17	15 \Rightarrow 6
	SD	$1.2 \cdot 10^{-06} \Rightarrow 1.5 \cdot 10^{-15}$	$3.2 \cdot 10^{-17} \Rightarrow 6.0 \cdot 10^{-29}$	> 50 \Rightarrow 20	17 \Rightarrow 6
	SP	$1.8 \cdot 10^{-12} \Rightarrow 1.4 \cdot 10^{-22}$	$2.4 \cdot 10^{-17} \Rightarrow 1.0 \cdot 10^{-30}$	23 \Rightarrow 9	17 \Rightarrow 6

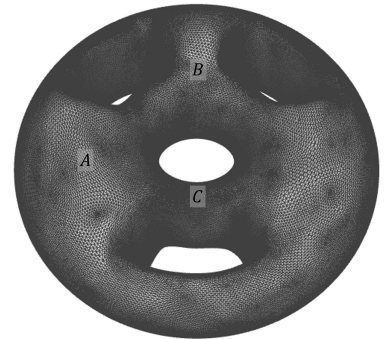
Table 6.5: Convergence improvement using the combination of speed up techniques. The third and fourth columns show the improvement in the relative residual errors after fifty iterations before and after the speed up, and the last two columns are the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .

CHAPTER 6. CONVERGENCE STUDY

Unfortunately, for the harmonic-free basis, even choosing the best hierarchy, five of the eight models fail to converge within six iterations, with the worst model, fertility, requiring as many as thirteen iterations. We note that combining the two techniques may not always give better results. For example, rooster + Whitney basis + SD needs three iterations to converge when only using SOR as the relaxation scheme for speedup, but it takes four iterations when using the combination. Nevertheless, overall, combining the speedup techniques improves convergence for most models.

Remarks

In practice, we find that the solver convergence may be affected by the surface curvatures and the quality of the triangulation. Table 6.6 shows the relative residual errors of the first six iterations when solving the diffusion problem (using the combination 4-torus + Whitney + SP) with the delta signal at points A , B , and C as shown in the inset. We observe that when the surface is umbilic (A and B), the solver converges quickly. However, when not umbilic (C), the solver converges more slowly, even when incorporating the improvements described above. Consequently, we suspect that the solver convergence might be related to the change in the curvatures. In the same way, we observe a similar slowdown when the delta vector



CHAPTER 6. CONVERGENCE STUDY

	<i>A</i>		<i>B</i>		<i>C</i>	
	w/o speed up	w/ speed up	w/o speed up	w/ speed up	w/o speed up	w/ speed up
0	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$	$1.00 \cdot 10^0$
1	$4.74 \cdot 10^{-3}$	$6.82 \cdot 10^{-3}$	$1.24 \cdot 10^{-2}$	$7.69 \cdot 10^{-3}$	$6.58 \cdot 10^{-2}$	$4.38 \cdot 10^{-2}$
2	$3.20 \cdot 10^{-4}$	$5.39 \cdot 10^{-5}$	$1.10 \cdot 10^{-3}$	$1.14 \cdot 10^{-4}$	$1.81 \cdot 10^{-2}$	$5.16 \cdot 10^{-3}$
3	$4.99 \cdot 10^{-5}$	$9.91 \cdot 10^{-7}$	$1.58 \cdot 10^{-4}$	$3.77 \cdot 10^{-6}$	$7.37 \cdot 10^{-3}$	$4.47 \cdot 10^{-4}$
4	$1.18 \cdot 10^{-5}$	$2.60 \cdot 10^{-8}$	$2.89 \cdot 10^{-5}$	$1.69 \cdot 10^{-7}$	$3.50 \cdot 10^{-3}$	$2.53 \cdot 10^{-5}$
5	$3.58 \cdot 10^{-6}$	$3.01 \cdot 10^{-10}$	$6.07 \cdot 10^{-6}$	$1.33 \cdot 10^{-8}$	$1.79 \cdot 10^{-3}$	$2.17 \cdot 10^{-6}$
6	$1.26 \cdot 10^{-6}$	$5.31 \cdot 10^{-12}$	$1.43 \cdot 10^{-6}$	$1.63 \cdot 10^{-9}$	$9.60 \cdot 10^{-4}$	$1.96 \cdot 10^{-7}$

Table 6.6: The relative residual errors of the first six iterations when the delta vector field is at points *A*, *B*, and *C* in the above inset figure and solving the diffusion problem using our hierarchical solver with and without the improvements.

field resides near the boundary of the mesh.

6.4.1 Multigrid Cycles

We compare the performance of three standard multigrid cycles – the V-Cycle, F-Cycle, and W-Cycle – of our accelerated multigrid solver using the combination 4-torus + Whitney on all three hierarchies. Although F-Cycle and W-Cycle take fewer iterations than V-Cycle to obtain the same level of accuracy, when comparing the performance in time, V-Cycle outperforms the others. Fig. 6.11 shows the convergence in time (instead of the number of iterations.) The time is measured on a PC with an Intel(R) Core(TM) i7-7700HQ 2.80 GHz CPU. It shows that V-Cycle converges faster than the others, particularly when α is small.

CHAPTER 6. CONVERGENCE STUDY

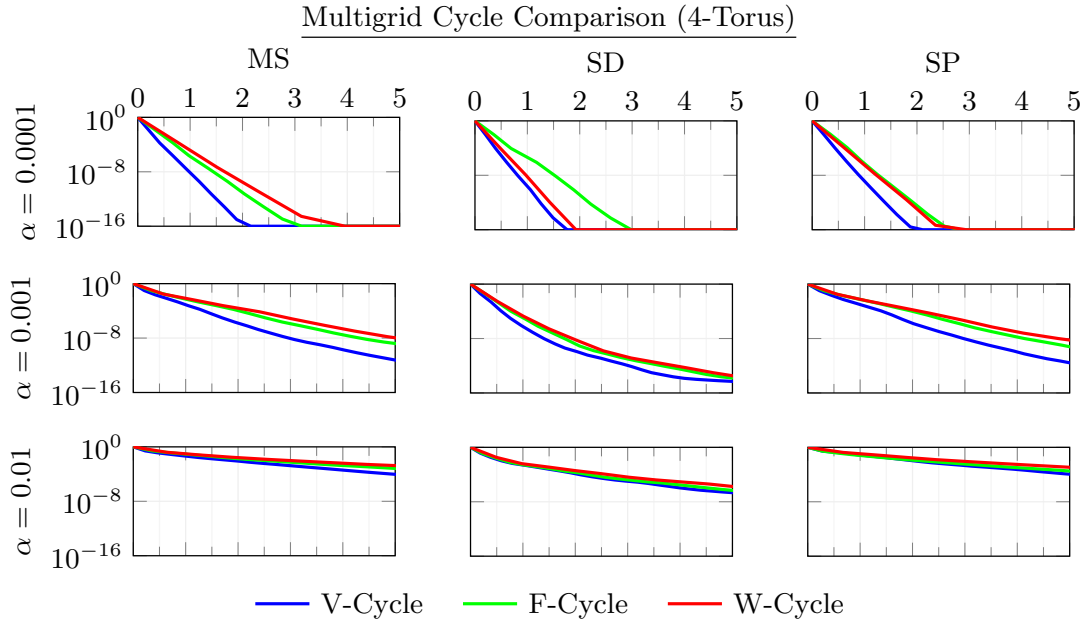


Figure 6.11: The convergence comparison plots among V-Cycle, F-Cycle, and W-Cycle in error-time plot. The y-axis (the relative residual error) is plotted in the log scale.

6.4.2 Direct Solver

The last part of the convergence study compares our solver's performance to a direct CHOLMOD solver [CDHR08]. We solve a diffusion problem with a small timestep $\alpha = 0.0001$ and measure the time the direct solver needs and the time until our solve obtains a solution whose relative residual error is less than 10^{-8} . Table 6.7 gives the details.

In this measurement, we exclude the mass and stiffness matrix computation times because both methods require the same matrices. We see that CHOLMOD consistently obtains a solution within 0.1 seconds, while its pre-processing time (matrix factorization) varies with the size of the systems. It

CHAPTER 6. CONVERGENCE STUDY

	System Dimensions		HS	CHOLMOD		Our Solver	
	harmonic-free	Whitney		harmonic-free	Whitney	harmonic-free	Whitney
Plane	131,210	195,588	MS	1.6 : 0.1	6.1 : 0.1	1.5 : 0.7	4.1 : 0.9
			SD	1.7 : 0.1	6.5 : 0.1	1.4 : 1.3	2.8 : 0.8
			SP	1.6 : 0.1	6.0 : 0.1	1.6 : 1.1	4.0 : 0.9
Sphere	245,764	368,640	MS	5.0 : 0.7	13.8 : 0.2	2.8 : 3.9	6.1 : 2.5
			SD	5.0 : 0.7	14.4 : 0.2	2.0 : 2.2	5.1 : 1.5
			SP	5.2 : 0.7	14.2 : 0.2	3.0 : 3.9	6.0 : 2.4
Torus	131,072	196,608	MS	2.6 : 0.1	7.0 : 0.1	1.8 : 1.4	3.4 : 1.3
			SD	2.6 : 0.1	7.2 : 0.1	1.3 : 1.7	2.6 : 1.1
			SP	2.6 : 0.1	7.0 : 0.1	1.8 : 0.8	3.5 : 1.4
4-Torus	131,060	196,608	MS	2.6 : 0.1	6.7 : 0.1	1.9 : 2.0	3.4 : 1.3
			SD	2.8 : 0.1	6.9 : 0.1	1.5 : 2.2	2.6 : 1.3
			SP	2.6 : 0.1	6.7 : 0.1	1.8 : 2.7	4.1 : 0.8
Hand	19,994	29,900	MS	0.6 : 0.0	0.6 : 0.0	0.3 : 0.2	0.7 : 0.1
			SD	0.3 : 0.0	0.7 : 0.0	0.3 : 0.2	0.3 : 0.1
			SP	0.6 : 0.0	0.7 : 0.0	0.3 : 0.2	1.0 : 0.1
Bimba	149,508	224,256	MS	2.0 : 0.1	7.8 : 0.1	1.9 : 1.4	5.1 : 1.3
			SD	2.1 : 0.1	7.4 : 0.1	1.4 : 1.9	3.6 : 1.0
			SP	2.0 : 0.1	7.4 : 0.1	2.0 : 0.9	4.5 : 1.3
Rooster	61,700	92,544	MS	1.0 : 0.0	3.1 : 0.0	1.1 : 0.5	1.9 : 0.3
			SD	1.2 : 0.0	3.1 : 0.0	0.7 : 0.6	1.5 : 0.3
			SP	1.1 : 0.0	3.3 : 0.0	1.3 : 0.4	1.8 : 0.3
Fertility	119,924	179,904	MS	1.7 : 0.1	5.9 : 0.1	2.0 : 1.3	3.3 : 1.3
			SD	1.8 : 0.1	6.4 : 0.1	1.3 : 1.6	3.4 : 1.2
			SP	1.7 : 0.1	6.9 : 0.1	1.8 : 0.8	3.4 : 1.3

Table 6.7: This table lists, from left to right, the models, the system dimensions when using the harmonic-free and Whitney basis, the hierarchies, and the pre-processing and solving time in seconds of using the direct CHOLMOD solver and our hierarchical solver to solve for the diffusion problem step.

takes substantially longer when the system dimension is higher (e.g., sphere + Whitney).

CHAPTER 6. CONVERGENCE STUDY

For our hierarchical solver, our pre-processing step computes the prolongation matrices, the coarse level mass and stiffness matrices, and the index coloring for SOR parallelization. This step also initializes the coarsest level direct solver. In general, the SD hierarchy has a faster pre-processing step because its prolongation matrices are sparser. Similarly, comparing the two bases with roughly the same system size (sphere + harmonic-free vs bimba + Whitney), we see that the harmonic-free system also has a faster pre-processing because of the sparsity of the prolongation matrices. Therefore, even though the harmonic-free basis requires more iterations to converge, it takes roughly the same time to obtain the solution at the same level of accuracy.

We notice that our implementation of SOR relaxation is not optimal, which takes roughly the same amount of time as the direct solver needs to solve for the solution. However, our solver requires considerably less pre-processing time, particularly when the system dimension is large. We notice that the current implementation is far from interactive. Future work is needed to optimize the performance.

6.5 Summary

We studied the convergence of our proposed solver and discovered that the 1-form system is more difficult to solve efficiently using a hierarchical ap-

CHAPTER 6. CONVERGENCE STUDY

proach. Via spectral analysis, we noticed that the 1-form prolongation matrix behaves differently than the 0-form prolongation matrix. The low-frequency of the 0-form coarse system is consistent with the low-frequency of the fine system, whereas that of the 1-form coarse system appears to be less so.

To mitigate the poor convergence, we investigated three speed up techniques and found that using SOR as the relaxation scheme and integrating a Krylov subspace update method noticeably improves the convergence, while using the smoothed prolongation also helps for some input models. After the speedup, when solving for a small timestep implicit diffusion step ($\alpha = 0.0001$), except for the 4-torus model, which requires seven iterations to converge, all models meet the desired convergence – obtaining an accuracy of 10^{-8} within six iterations.

In terms of performance, we found that V-Cycle is better than other multi-grid cycles. Compared to a direct CHOLMOD solver, however, our solver requires more time to obtain a solution, though it takes less pre-processing time, particularly when the system dimension is large. This could make it beneficial for applications requiring dynamic changes of the system matrix, such as smoothing/sharpening, where users may want to interactively change the system matrix to explore different smoothing/sharpening effects.

Chapter 7

Applications

In this chapter, using our hierarchical solver, we demonstrate an application – parallel transport (via the Heat Method [SSC19]), and discuss other applications related to our previous work [LK19], and to other multigrid systems [PKCH18, KLAKE23].

7.1 Parallel Transport

In the work of Sharp *et al.* [SSC19, Alg. 1], the authors show that by solving diffusion equations, one can parallel transport a vector on a 2-manifold. We apply our solver to this application using the combination fertility + Whitney + SP. The implementation requires diffusing one vector field and two scalar fields (one corresponding to the magnitude of the vector and one a delta function.) We

CHAPTER 7. APPLICATIONS

denote by $E_{vec}(\mathbf{x})$, $E_{mag}(\mathbf{x})$ and $E_{delta}(\mathbf{x})$ their relative residual errors. Note that, we do not perform the curvature correction described in [SSC19, Sec. 6.1.1].

For the timestep α , the suggestion in [SSC19, Sec. 7.3] is to use the square of the average edge length, which is $\alpha = 2.2 \cdot 10^{-5}$. However, in practice, we find this value is too small, resulting in numerical instability. Instead, we use $2.2 \cdot 10^{-4}$.

The first column of Fig. 7.1 shows the number of iterations. The second and third columns depict the results of parallel transporting a vector on the fertility model using our solver and the difference from the direct solver result, from the top view; while the next two columns show the same from the bottom view. Note that since our initial guess is zero, at iteration 0, the differences (in the third column and fifth column) show the direct solver result. The last three columns are the relative residual errors $E_{vec}(\mathbf{x})$, $E_{mag}(\mathbf{x})$, and $E_{delta}(\mathbf{x})$.

Our solver quickly obtains an accuracy of 10^{-8} within six iterations, and already close to 10^{-16} after about fifteen iterations. After the ten iterations, the region near the source delta vector field (visualized as a green arrow) is the same as the direct solver's (*i.e.*, no red arrows in the third column), but the region far way from the source is still converging (such as the bottom shown in the fifth column), even though the residual error is very small (sixth column). We obtain a visually indistinguishable solution (apart from the cut locus) using thirty iterations. For the cut locus to be the same as the direct solver's result,

CHAPTER 7. APPLICATIONS

our solver takes fifty iterations.

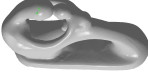
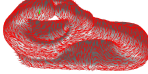

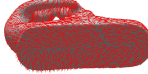
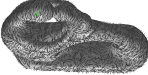
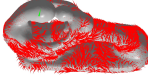


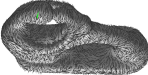
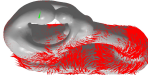
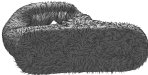
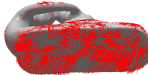
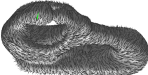
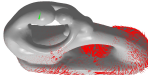

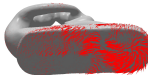
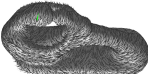
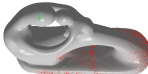
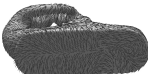

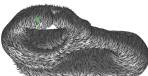
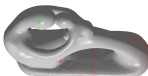
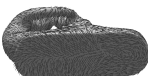
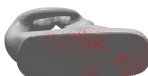
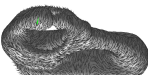
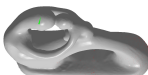
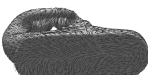
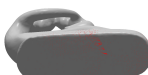
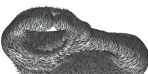
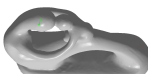
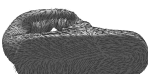

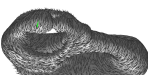
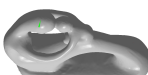
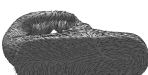

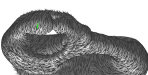
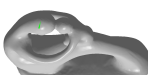
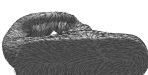

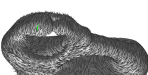
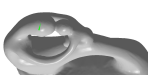
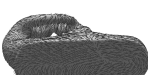

Itrs.	Top View		Bottom View		$E_{vec}(\mathbf{x})$	$E_{mag}(\mathbf{x})$	$E_{delta}(\mathbf{x})$
0					1	1	1
5					$4.6 \cdot 10^{-7}$	$5.4 \cdot 10^{-21}$	$6.1 \cdot 10^{-21}$
10					$1.0 \cdot 10^{-12}$	$2.0 \cdot 10^{-31}$	$2.7 \cdot 10^{-31}$
15					$2.5 \cdot 10^{-16}$	$1.8 \cdot 10^{-31}$	$1.7 \cdot 10^{-31}$
20					$1.3 \cdot 10^{-18}$	$2.3 \cdot 10^{-31}$	$2.1 \cdot 10^{-31}$
25					$7.3 \cdot 10^{-21}$	$2.4 \cdot 10^{-31}$	$2.7 \cdot 10^{-31}$
30					$4.6 \cdot 10^{-23}$	$1.9 \cdot 10^{-31}$	$2.8 \cdot 10^{-31}$
35					$3.9 \cdot 10^{-25}$	$2.2 \cdot 10^{-31}$	$2.8 \cdot 10^{-31}$
40					$1.0 \cdot 10^{-26}$	$2.1 \cdot 10^{-31}$	$2.6 \cdot 10^{-31}$
45					$7.3 \cdot 10^{-28}$	$1.8 \cdot 10^{-31}$	$2.0 \cdot 10^{-31}$
50					$6.7 \cdot 10^{-29}$	$1.7 \cdot 10^{-31}$	$2.3 \cdot 10^{-31}$

Figure 7.1: Parallel transport results (top and bottom views) of our solver at every five iterations on the fertility model using the SP hierarchy and the Whitney basis, and the relative residual errors of the vector and scalar diffusion solves.

7.2 Optical Flow

In our earlier work [LK19] on establishing correspondences between genus-zero shapes, one key component was performing optical flow over the sphere. We use optical flow to establish the correspondences between two spherical images. In that work, the optical flow field is represented using the harmonic-free basis and is computed hierarchically. At each level, a direct solver is used to solve the linearized optical flow equation iteratively. At each iteration, the system matrix needs to be updated. At finer levels, the system updates and direct solves become expensive. We believe that these expensive system updates and direct solves can be replaced by our hierarchical solvers.

7.3 Other Multigrid Systems

Texture Grid Basis

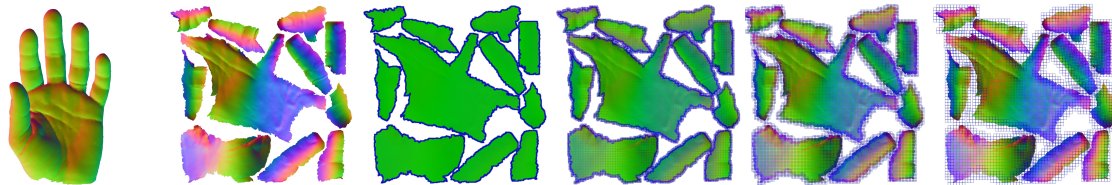


Figure 7.2: An example of texture grid hierarchy. The bi-linear-like seamless 0-form basis is defined on the texture grid nodes where the texture grid cells overlapped with the texture map of them model.

Prada *et al.* [PKCH18] developed a novel seamless 0-form basis over a tex-

CHAPTER 7. APPLICATIONS

ture domain and leveraged the hierarchical structure of the texture grid to support interactive signal processing applications. Fig. 7.2 shows an example of the texture grid hierarchy of the hand model. Instead of using the hat basis, we have begun exploring using this bi-linear-like seamless 0-form basis (which also forms a partition of unity) and use the texel 0-form prolongation matrix defined in [PKCH18] to construct a texture grid 1-form basis and prolongation matrix (that satisfies the sufficiency conditions *Claim 1*). By doing so, we hope to leverage the regular structure of texture grids to further speed up the multigrid performance.

We attempted to implement this approach. Figs. 7.3 and 7.4 demonstrate a preliminary example of computing the logarithmic map (by the vector heat method) using the Whitney 1-form basis and prolongation described above, without any multigrid speedup. In these figures, we show the results of the direct solver and of our solver (at every twenty iterations). We observe a similar slow convergence to solving the vector transport problem on a triangle mesh. Without any speedup, after twenty iterations, the region near the source is the same as the result of the direct solver. However, regions near the cut locus take longer to converge. Our solver takes about a hundred iterations to obtain a result that is close to the direct solver's.

CHAPTER 7. APPLICATIONS



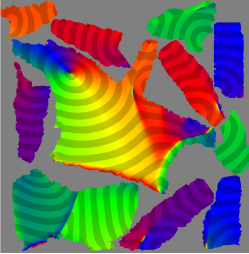





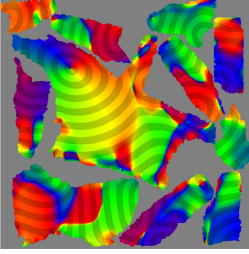


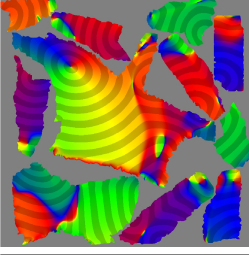


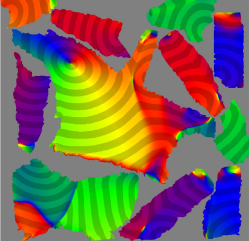
	Front View	Back View	Texture	$E_{vec}(\mathbf{x})$	$E_{mag}(\mathbf{x})$	$E_{delta}(\mathbf{x})$
CHOLMOD				0	0	0
Ours (Itr. 0)				1	1	1
Ours (Itr. 20)				$2.4 \cdot 10^{-11}$	$2.7 \cdot 10^{-23}$	$1.3 \cdot 10^{-23}$
Ours (Itr. 40)				$7.1 \cdot 10^{-15}$	$3.7 \cdot 10^{-30}$	$9.4 \cdot 10^{-30}$
Ours (Itr. 60)				$1.3 \cdot 10^{-17}$	$3.6 \cdot 10^{-30}$	$8.5 \cdot 10^{-30}$

Figure 7.3: Logarithmic map results (front and back views, and on the texture) of our solver at every twenty iterations on the hand model the Whitney basis, and the relative residual errors of the vector and scalar diffusion solves. The first row shows the direct solver’s result.

CHAPTER 7. APPLICATIONS



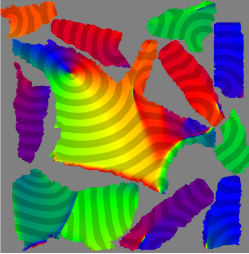


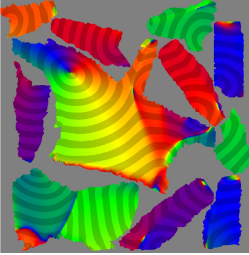


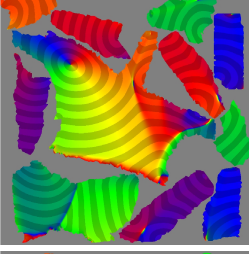


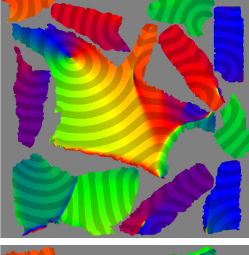


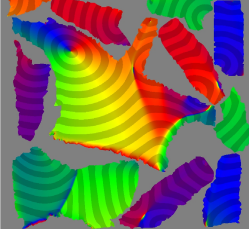
	Front View	Back View	Texture	$E_{vec}(\mathbf{x})$	$E_{mag}(\mathbf{x})$	$E_{delta}(\mathbf{x})$
CHOLMOD				0	0	0
Ours (Itr. 80)				$1.6 \cdot 10^{-19}$	$3.6 \cdot 10^{-30}$	$8.8 \cdot 10^{-30}$
Ours (Itr. 100)				$7.1 \cdot 10^{-20}$	$3.5 \cdot 10^{-30}$	$8.3 \cdot 10^{-30}$
Ours (Itr. 120)				$7.0 \cdot 10^{-20}$	$3.5 \cdot 10^{-30}$	$9.0 \cdot 10^{-30}$
Ours (Itr. 140)				$6.9 \cdot 10^{-20}$	$3.5 \cdot 10^{-30}$	$8.6 \cdot 10^{-30}$

Figure 7.4: Logarithmic map results (front and back views, and on the texture) of our solver at every twenty iterations on the hand model the Whitney basis, and the relative residual errors of the vector and scalar diffusion solves. The first row shows the direct solver’s result. (cont.)

CHAPTER 7. APPLICATIONS

k -Form Basis

Our approach is not limited to a 1-form basis. Recently, our work on Poisson manifold reconstruction [KLAK23] generalizes Poisson surface reconstruction (PSR) by fitting the normal of a co-dimension k manifold to the wedge product of the gradients of k scalar functions (recall that in PSR, we look for a scalar function whose gradient fits well the normal of a surface.) In this work, to reconstruct the manifold of co-dimension two (*i.e.*, its normal can be described by 2-forms), we construct a 2-form basis over Euclidean space by first defining a 0-form basis and then taking the wedge product of the gradients of pairs of basis functions. Leveraging bilinearity, we extend the prolongation matrix defined for the 0-form basis to a prolongation matrix defined on the 2-form basis, giving us a 2-form hierarchy.

Similarly, we can construct a k -form basis by taking the wedge product of the gradients of k -pairs of the 0-form basis, and the multi-linearity allows us to define the k -form prolongation matrix from the 0-form prolongation matrix. This, in turn, allows us to design an efficient multigrid solver that can be used to reconstruct manifolds of arbitrary co-dimension.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

To the best of our knowledge, this dissertation is the first to look into the *hierarchical* approach for vector field processing using the finite element exterior calculus gradient domain formulation. We first went over related literature, then we reviewed the gradient domain formulation for vector field processing, which could be generalized to k -form processing. Next, we described the general multigrid method for solving a linear system with speed up techniques, and we studied two existing 1-form bases – harmonic-free and Whitney, which could be constructed from a 0-form basis. Given a 0-form prolongation matrix, we show that these constructions implicitly induce 1-form prolongation matrices which use to design a multigrid solver. We also generalized this con-

CHAPTER 8. CONCLUSION AND FUTURE WORK

struction using category theory and described the sufficiency conditions for the commutativity between the induced prolongations and linear operators defined on the related vector spaces. Lastly, we extensively studied the convergence of our solver and demonstrated our approach for a few applications.

8.2 Future Work

Our approach extends existing efficient 0-form basis and hierarchical solvers to 1-form (also to k -form) hierarchical solvers, which opens the door to leveraging existing 0-form processing techniques for higher form processing. However, empirically we have found that the resulting solvers are not as efficient as we would like. Even though we mitigate the problem by integrating speedup techniques and achieve comparable convergence for small diffusion timestep, the following challenges remain:

Additional Speedups

To further address the convergence problems, one practical and straightforward direction to speed up the solver by leveraging the GPU computation power. On average, the current implementation takes about 0.2 seconds for each V-Cycle iteration. The bottleneck is the relaxation and coarse-fine matrix-vector computation. These two steps could be made faster using the GPU.

CHAPTER 8. CONCLUSION AND FUTURE WORK

Alternatively, another standard method to cope with a slow iterative method is preconditioning. However, a good preconditioner is difficult to find. In order to preserve the sparsity (hence the efficiency of solving sparse system), one promising preconditioner to look into is the sparse approximate inverse preconditioner ([ST23, Chap. 11]). This preconditioner solves for an approximate inverse that has the same sparsity pattern as the system matrix. In the applied numerical mathematics community, Benzi and Tûma review the method and discuss the implementation details in [BT99]. In the future, we could borrow this technique and see if it helps with convergence.

Optimal Prolongation Matrix

Another way to address the convergence issue is to deal with the prolongation matrix. We have observed the difference between the efficient 0-form prolongation matrix and the less effective 1-form prolongation matrix using the correlation matrix. An interesting research question to study is: “How do we optimize a 0-form prolongation matrix such that the induced 1-form prolongation has a more diagonal-like coarse-fine correlation matrix?” We believe such a 1-form prolongation matrix would give better convergence.

CHAPTER 8. CONCLUSION AND FUTURE WORK

k-Form Processing

Since our formulation is general, the natural direction to continue is to extend the implementation to higher dimensional k -form processing. For example, we can extend it to four dimensional (space + time) and use the system to solve Maxwell's equations by modeling the electromagnetic tensor as a 2-form $F = E + B$, where

$$E = E_x dt \wedge dx + E_y dt \wedge dy + E_z dt \wedge dz, \quad B = B_x dt \wedge dx + B_y dt \wedge dy + B_z dt \wedge dz.$$

Then, Maxwell's equations can be expressed by using the exterior derivative and Hodge star [Car03, Sec. 1.8]:

$$dF = 0, \quad d \star F = 0.$$

Extending our approach to k -form processing may enable many applications to solving physical equations.

Bibliography

- [ABCCO13] AZENCOT O., BEN-CHEN M., CHAZAL F., OVSJANIKOV M.: An operator approach to tangent vector field processing. *Computer Graphics Forum* 32, 5 (2013), 73–82. doi:10.1111/cgf.12174.
- [ABHT03] ADAMS M., BREZINA M., HU J., TUMINARO R.: Parallel multi-grid smoothing: Polynomial versus gauss–seidel. *J. Comput. Phys.* 188, 2 (jul 2003), 593–610. doi:10.1016/S0021-9991(03)00194-3.
- [AFW06] ARNOLD D. N., FALK R. S., WINTHER R.: Finite element exterior calculus, homological techniques, and applications. *Acta Numerica* 15 (2006), 1–155. doi:10.1017/S0962492906210018.
- [AR07] AGRAWAL A., RASKAR R.: Gradient domain manipulation techniques in vision and graphics. ICCV 2007 Course, Last visited 2023-02-10, 2007. URL: <http://www.amitkagrawal.com/ICCV2007Course/>.

BIBLIOGRAPHY

- [ARC06] AGRAWAL A., RASKAR R., CHELLAPPA R.: Edge suppression by gradient field transformation using cross-projection tensors. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2* (USA, 2006), CVPR '06, IEEE Computer Society, p. 2301–2308. doi: 10.1109/CVPR.2006.106.
- [ARNL05] AGRAWAL A., RASKAR R., NAYAR S. K., LI Y.: Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph.* 24, 3 (jul 2005), 828–835. doi: 10.1145/1073204.1073269.
- [BCBSG10] BEN-CHEN M., BUTSCHER A., SOLOMON J., GUIBAS L.: On discrete killing vector fields and patterns on surfaces. *Computer Graphics Forum* 29, 5 (2010), 1701–1711. doi:10.1111/j.1467-8659.2010.01779.x.
- [BCCZ08] BHAT P., CURLESS B., COHEN M., ZITNICK C. L.: Fourier analysis of the 2d screened poisson equation for gradient domain problems. In *Computer Vision – ECCV 2008* (Berlin, Heidelberg, 2008), Forsyth D., Torr P., Zisserman A., (Eds.), Springer Berlin Heidelberg, pp. 114–128. doi:10.1007/978-3-540-88688-4_9.
- [BHKB20] BUNGE A., HERHOLZ P., KAZHDAN M., BOTSCH M.: Polygon

BIBLIOGRAPHY

- laplacian made simple. *Computer Graphics Forum* 39, 2 (2020), 303–313. doi:10.1111/cgf.13931.
- [BHM00] BRIGGS W. L., HENSON V. E., MCCORMICK S. F.: *A Multigrid Tutorial, Second Edition*, second ed. Society for Industrial and Applied Mathematics, 2000. doi:10.1137/1.9780898719505.
- [BL11] BRANDT A., LIVNE O. E.: *Multigrid Techniques*. Society for Industrial and Applied Mathematics, 2011. doi:10.1137/1.9781611970753.
- [Bra77] BRANDT A.: Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 31, 138 (1977), 333–390. doi:10.2307/2006422.
- [BSEH17] BRANDT C., SCANDOLO L., EISEMANN E., HILDEBRANDT K.: Spectral processing of tangential vector fields. *Comput. Graph. Forum* 36, 6 (sep 2017), 338–353. doi:10.1111/cgf.12942.
- [BT99] BENZI M., TÛMA M.: A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics* 30, 2 (1999), 305–340. doi:10.1016/S0168-9274(98)00118-4.
- [Bur03] BURTON D.: A primer on exterior differential calculus. *Theoret-*

BIBLIOGRAPHY

- ical and Applied Mechanics* 30 (2003), 85–162. doi:10.2298/TAM0302085B.
- [BZCC10] BHAT P., ZITNICK C. L., COHEN M., CURLESS B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph.* 29, 2 (apr 2010). doi:10.1145/1731047.1731048.
- [Car03] CARROLL S.: *Spacetime and Geometry: An Introduction to General Relativity*. Benjamin Cummings, 2003. URL: <http://www.amazon.com/Spacetime-Geometry-Introduction-General-Relativity/dp/0805387323>.
- [CCC*08] CIGNONI P., CALLIERI M., CORSINI M., DELLEPIANE M., GANOVELLI F., RANZUGLIA G.: MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference* (2008), Scarano V., Chiara R. D., Erra U., (Eds.), The Eurographics Association. doi:10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.
- [CdGDS13] CRANE K., DE GOES F., DESBRUN M., SCHRÖDER P.: Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses* (New York, NY, USA, 2013), SIG-

BIBLIOGRAPHY

- GRAPH '13, Association for Computing Machinery. doi:10.1145/2504435.2504442.
- [CDHR08] CHEN Y., DAVIS T. A., HAGER W. W., RAJAMANICKAM S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.* 35, 3 (oct 2008). doi:10.1145/1391989.1391995.
- [CDS10] CRANE K., DESBRUN M., SCHRÖDER P.: Trivial connections on discrete surfaces. *Computer Graphics Forum* 29, 5 (2010), 1525–1533. doi:10.1111/j.1467-8659.2010.01761.x.
- [CRK16] CHUANG M., RUSINKIEWICZ S., KAZHDAN M.: Gradient-domain processing of meshes. *Journal of Computer Graphics Techniques (JCGT)* 5, 4 (December 2016), 44–55. doi:10.5281/zenodo.7953865.
- [dGDT16] DE GOES F., DESBRUN M., TONG Y.: Vector field processing on triangle meshes. In *ACM SIGGRAPH 2016 Courses* (New York, NY, USA, 2016), SIGGRAPH '16, Association for Computing Machinery. doi:10.1145/2897826.2927303.
- [DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *Computer Graphics Forum* 21, 3 (2002), 209–218. doi:10.1111/1467-8659.00580.

BIBLIOGRAPHY

- [DQLB05] DU Z., QIN X., LIN H., BAO H.: Shadow removal in gradient domain. In *Image Analysis and Recognition* (Berlin, Heidelberg, 2005), Kamel M., Campilho A., (Eds.), Springer Berlin Heidelberg, pp. 107–115. doi:10.1007/11559573_14.
- [DVPSH14] DIAMANTI O., VAXMAN A., PANOZZO D., SORKINE-HORNUNG O.: Designing n-polyvector fields with complex polynomials. *Computer Graphics Forum* 33, 5 (2014), 1–11. doi:10.1111/cgf.12426.
- [DW00] DICARLO J. M., WANDELL B. A.: Rendering high dynamic range images. In *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications* (San Jose, CA, United States, 2000), Sampat N., Yeh T., Blouke M. M., Sampat N., Jr. G. M. W., Yeh T., (Eds.), vol. 3965, International Society for Optics and Photonics, SPIE, pp. 392 – 401. doi:10.1117/12.385456.
- [ELM96] ELMAN H. C.: Multigrid and krylov subspace methods for the discrete stokes equations. *International Journal for Numerical Methods in Fluids* 22, 8 (1996), 755–770. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0363%2819960430%2922%3A8%3C755%3A%3AAID-FLD377%3E3.0.CO%3B2-1>.

BIBLIOGRAPHY

- [FLW02] FATTAL R., LISCHINSKI D., WERMAN M.: Gradient domain high dynamic range compression. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2002), SIGGRAPH '02, Association for Computing Machinery, p. 249–256. doi:10.1145/566570.566573.
- [FSDH07] FISHER M., SCHRÖDER P., DESBRUN M., HOPPE H.: Design of tangent vector fields. *ACM Trans. Graph.* 26, 3 (jul 2007), 56–es. doi:10.1145/1276377.1276447.
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., p. 209–216. doi:10.1145/258734.258849.
- [GMP*10] GAWLIK E. S., MULLEN P., PAVLOV D., MARSDEN J. E., DESBRUN M.: Geometric, variational discretization of continuum theories. *Physica D: Nonlinear Phenomena* 240 (2010), 1724–1760. doi:10.1016/j.physd.2011.07.011.
- [Hem90] HEMKER P. W.: On the order of prolongations and restrictions in multigrid procedures. *J. Comput. Appl. Math.* 32, 3 (nov 1990), 423–429. doi:10.1016/0377-0427(90)90047-4.

BIBLIOGRAPHY

- [HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L.-Y., TENG S.-H., BAO H., GUO B., SHUM H.-Y.: Subspace gradient domain mesh deformation. *ACM Trans. Graph.* 25, 3 (jul 2006), 1126–1134. doi:10.1145/1141911.1142003.
- [KCPS13] KNÖPPEL F., CRANE K., PINKALL U., SCHRÖDER P.: Globally optimal direction fields. *ACM Trans. Graph.* 32, 4 (jul 2013). doi:10.1145/2461912.2462005.
- [Kev23] KEVIN D. MAHONEY AND THE LATDICT GROUP.: Lat-dict, 2023. Last visited 2023-02-02. URL: <https://www.latin-dictionary.net/search/latin/vector>.
- [KH08] KAZHDAN M., HOPPE H.: Streaming multigrid for gradient-domain operations on large images. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–10. doi:10.1145/1360612.1360620.
- [KLAK23] KOHLBRENNER M., LEE S. C., ALEXA M., KAZHDAN M.: Poisson Manifold Reconstruction - Beyond Co-dimension One. *Computer Graphics Forum* (2023). doi:10.1111/cgf.14907.
- [KMA*15] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Trans. Graph.* 34, 4 (jul 2015). doi:10.1145/2766997.

BIBLIOGRAPHY

- [LK19] LEE S. C., KAZHDAN M.: Dense point-to-point correspondences between genus-zero shapes. *Computer Graphics Forum* 38, 5 (2019), 27–37. doi:10.1111/cgf.13787.
- [LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain metropolis light transport. *ACM Trans. Graph.* 32, 4 (jul 2013). doi:10.1145/2461912.2461943.
- [LL22] LUO S., LU J.: Gfnet: A gradient information compensation-based face super-resolution network. *IEEE Access* 10 (2022), 8073–8080. doi:10.1109/ACCESS.2022.3143499.
- [Loo87] LOOP C.: *Smooth Subdivision Surfaces Based on Triangles*. PhD thesis, University of Utah, January 1987. URL: <https://www.microsoft.com/en-us/research/publication/smooth-subdivision-surfaces-based-on-triangles/>.
- [Lot23] LOTTES J.: Optimal polynomial smoothers for multigrid v-cycles, 2023. doi:10.48550/arXiv.2202.08830.
- [LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* 21, 3 (jul 2002), 362–371. doi:10.1145/566654.566590.

BIBLIOGRAPHY

- [LS20] LAHTINEN V., STENVALL A.: A category theoretical interpretation of discretization in galerkin finite element method. *Mathematische Zeitschrift* 296, 3 (Dec 2020), 1271–1285. doi:10.1007/s00209-020-02456-1.
- [LTGD16] LIU B., TONG Y., GOES F. D., DESBRUN M.: Discrete connection and covariant derivative for vector field analysis and design. *ACM Trans. Graph.* 35, 3 (mar 2016). doi:10.1145/2870629.
- [LZBCJ21] LIU H.-T. D., ZHANG J. E., BEN-CHEN M., JACOBSON A.: Surface multigrid via intrinsic prolongation. *ACM Trans. Graph.* 40, 4 (jul 2021). doi:10.1145/3450626.3459768.
- [LZPW04] LEVIN A., ZOMET A., PELEG S., WEISS Y.: Seamless image stitching in the gradient domain. In *Computer Vision - ECCV 2004* (Berlin, Heidelberg, 2004), Pajdla T., Matas J., (Eds.), Springer Berlin Heidelberg, pp. 377–389. doi:10.1007/978-3-540-24673-2_31.
- [McC87] MCCORMICK S. F.: *Multigrid Methods*. Society for Industrial and Applied Mathematics, 1987. doi:10.1137/1.9781611971057.
- [MP08] MCCANN J., POLLARD N. S.: Real-time gradient-domain painting. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–7. doi:10.1145/1360612.1360692.

BIBLIOGRAPHY

- [NP19] NAPOV A., PERRUSSEL R.: Revisiting aggregation-based multigrid for edge elements. *Electron. Trans. Numer. Anal.* 51 (2019), 118–134. doi:10.1553/etna_vol51s118.
- [OW00] OOSTERLEE C. W., WASHIO T.: Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows. *SIAM Journal on Scientific Computing* 21, 5 (2000), 1670–1690. doi:10.1137/S1064827598338093.
- [Pet06] PETERSEN P.: *Riemannian Geometry*. Springer, New York, NY, 2006. doi:10.1007/978-0-387-29403-2.
- [PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Trans. Graph.* 22, 3 (jul 2003), 313–318. doi:10.1145/882262.882269.
- [PKCH18] PRADA F., KAZHDAN M., CHUANG M., HOPPE H.: Gradient-domain processing within a texture atlas. *ACM Trans. Graph.* 37, 4 (jul 2018). doi:10.1145/3197517.3201317.
- [PP00] POLTHIER K., PREUSS E.: Variational approach to vector field decomposition. In *Data Visualization 2000* (Vienna, 2000), de Leeuw W. C., van Liere R., (Eds.), Springer Vienna, pp. 147–155. doi:10.1007/978-3-7091-6783-0_15.

BIBLIOGRAPHY

- [RdL16] RIBEIRO P. C., DE CAMPOS VELHO H. F., LOPES H.: Helmholtz–hodge decomposition and the analysis of 2d vector field ensembles. *Computers & Graphics* 55 (2016), 80–96. doi:10.1016/j.cag.2016.01.001.
- [RS02] REITZINGER S., SCHÖBERL J.: An algebraic multigrid method for finite element discretizations with edge elements. *Numerical Linear Algebra with Applications* 9, 3 (2002), 223–238. doi:10.1002/nla.271.
- [SJZW07] SHEN J., JIN X., ZHOU C., WANG C. C.: Gradient based image completion by solving the poisson equation. *Computers & Graphics* 31, 1 (2007), 119–126. doi:https://doi.org/10.1016/j.cag.2006.10.004.
- [SSB15] SUNDAR H., STADLER G., BIROS G.: Comparison of multigrid algorithms for high-order continuous finite element discretizations. *Numerical Linear Algebra with Applications* 22, 4 (2015), 664–680. doi:10.1002/nla.1979.
- [SSC19] SHARP N., SOLIMAN Y., CRANE K.: The vector heat method. *ACM Trans. Graph.* 38, 3 (jun 2019). doi:10.1145/3243651.
- [ST23] SCOTT J., TUMA M.: *Sparse Approximate Inverse Precondition-*

BIBLIOGRAPHY

- ers*. Springer International Publishing, Cham, 2023, pp. 205–221. doi:10.1007/978-3-031-25820-6_11.
- [STO16] SHIBATA T., TANAKA M., OKUTOMI M.: Gradient-domain image reconstruction framework with intensity-range and base-structure constraints. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, 2016), IEEE, pp. 2745–2753. doi:10.1109/CVPR.2016.300.
- [SWJG20] STEIN O., WARDETZKY M., JACOBSON A., GRINSPUN E.: A simple discretization of the vector dirichlet energy. *Computer Graphics Forum* 39, 5 (2020), 81–92. doi:10.1111/cgf.14070.
- [TKB99] TARHASAARI T., KETTUNEN L., BOSSAVIT A.: Some realizations of a discrete hodge operator: a reinterpretation of finite element techniques [for em field analysis]. *IEEE Transactions on Magnetism* 35, 3 (1999), 1494–1497. doi:10.1109/20.767250.
- [TLHD03] TONG Y., LOMBEYDA S., HIRANI A. N., DESBRUN M.: Discrete multiscale vector field decomposition. *ACM Trans. Graph.* 22, 3 (jul 2003), 445–452. doi:10.1145/882262.882290.
- [Van92] VANĚK P.: Acceleration of convergence of a two-level algorithm by smoothing transfer operators. *Applications of Mathematics* 37 (1992), 265–274. doi:10.21136/am.1992.104509.

BIBLIOGRAPHY

- [Van12] VANĚK P.: Smoothed prolongation multigrid with rapid coarsening and massive smoothing. *Applications of Mathematics* 57, 1 (Feb 2012), 1–10. doi:10.1007/s10492-012-0001-3.
- [VMB94] VANEK P., MANDEL J., BREZINA M.: Algebraic multigrid on unstructured meshes. *UCD/CCM Report 34* (1994), 123–146. URL: <http://www-math.ucdenver.edu/ccm/reports/rep034.ps.gz>.
- [War07] WARDETZKY M.: Discrete differential operators on polyhedral surfaces - convergence and approximation. In *Discrete Differential Operators on Polyhedral Surfaces - Convergence and Approximation* (Berlin, 2007), Dissertationen FU, pp. 1–105. doi:10.17169/refubium-7719.
- [WD11] WANG O., DAVIS J.: Gradient domain hdr compositing. In *ACM SIGGRAPH 2011 Posters* (New York, NY, USA, 2011), SIGGRAPH '11, Association for Computing Machinery. doi:10.1145/2037715.2037755.
- [Whi05] WHITNEY H.: *Geometric Integration Theory*. Dover Books on Mathematics. Dover Publications, 2005. URL: <https://books.google.com/books?id=UvxUAAAAYAAJ>.
- [WLA*14] WILLIAMS S., LIJEWSKI M., ALMGREN A., STRAALEN B. V.,

BIBLIOGRAPHY

- CARSON E., KNIGHT N., DEMMEL J.: s-step krylov subspace methods as bottom solvers for geometric multigrid. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium* (2014), pp. 1149–1158. doi:10.1109/IPDPS.2014.119.
- [WWT*06] WANG K., WEIWEI, TONG Y., DESBRUN M., SCHRÖDER P.: Edge subdivision schemes and the construction of smooth vector fields. *ACM Trans. Graph.* 25, 3 (jul 2006), 1041–1048. doi:10.1145/1141911.1141991.
- [XLG*12] XIE Z.-F., LAU R. W. H., GUI Y., CHEN M.-G., MA L.-Z.: A gradient-domain-based edge-preserving sharpen filter. *The Visual Computer* 28, 12 (Dec 2012), 1195–1207. doi:10.1007/s00371-011-0668-6.
- [ZDWT19] ZHAO R., DESBRUN M., WEI G.-W., TONG Y.: 3d hodge decompositions of edge- and face-based vector fields. *ACM Trans. Graph.* 38, 6 (nov 2019). doi:10.1145/3355089.3356546.
- [ZMT06] ZHANG E., MISCHAIKOW K., TURK G.: Vector field design on surfaces. *ACM Trans. Graph.* 25, 4 (oct 2006), 1294–1326. doi:10.1145/1183287.1183290.

Appendix A

Input Hierarchical Meshes

This appendix shows a side-by-side visual comparison of the three hierarchies mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP) on the eight input models.

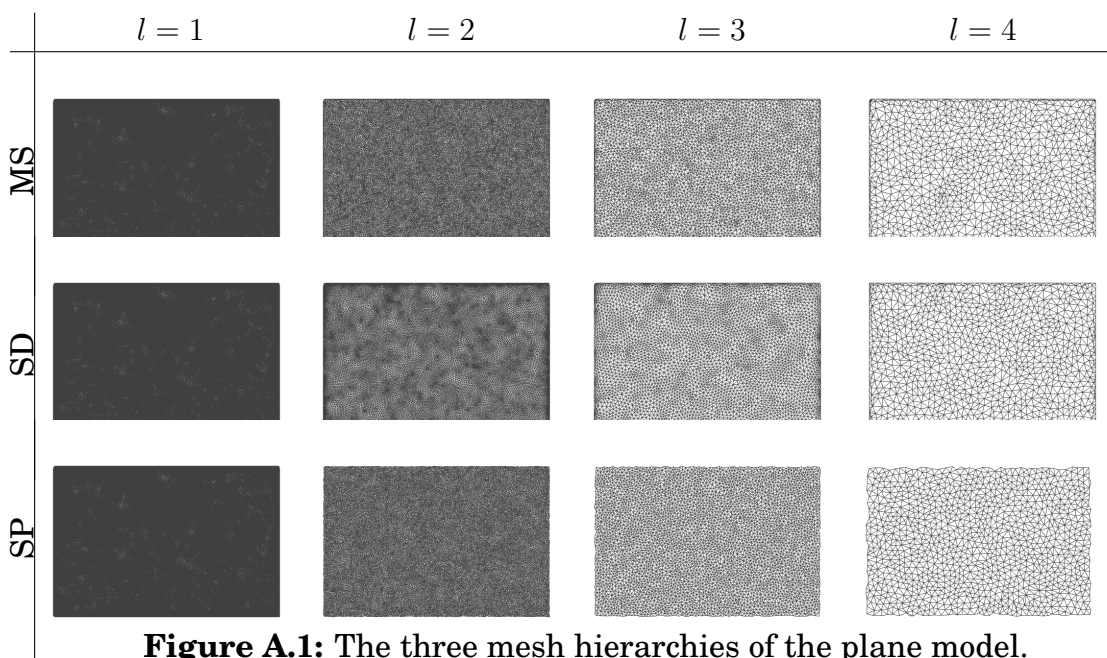


Figure A.1: The three mesh hierarchies of the plane model.

APPENDIX A. INPUT HIERARICHIAL MESHES

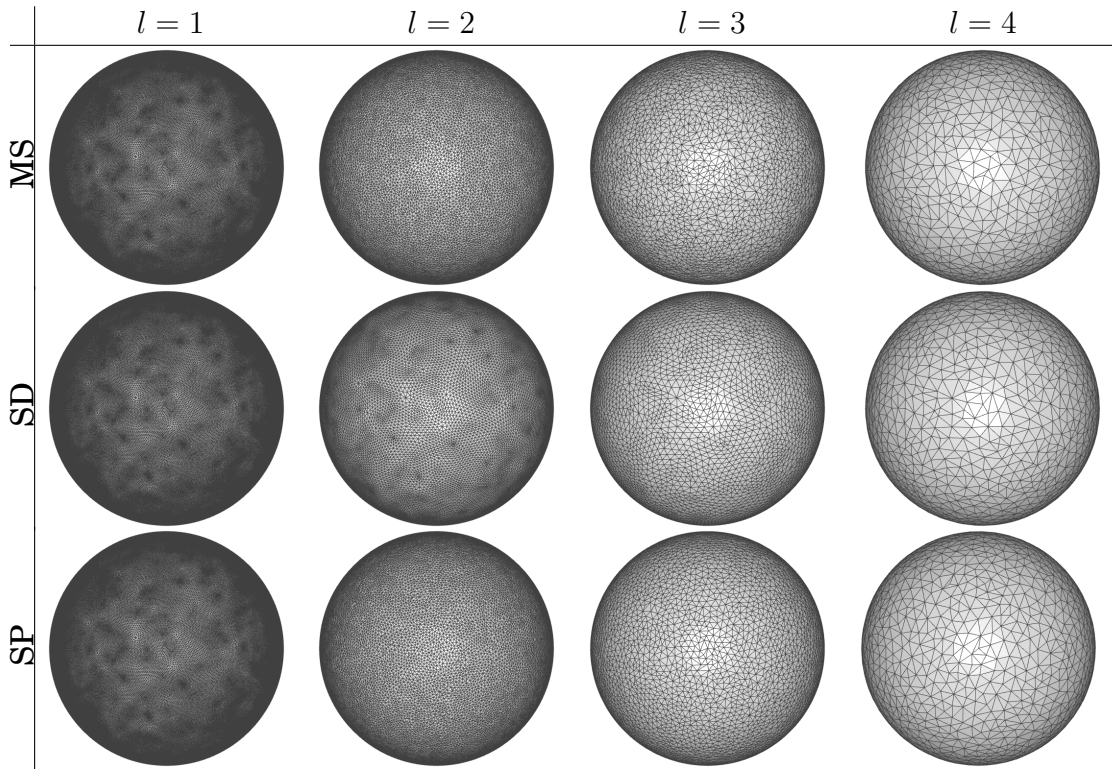


Figure A.2: The three mesh hierarchies of the sphere model.

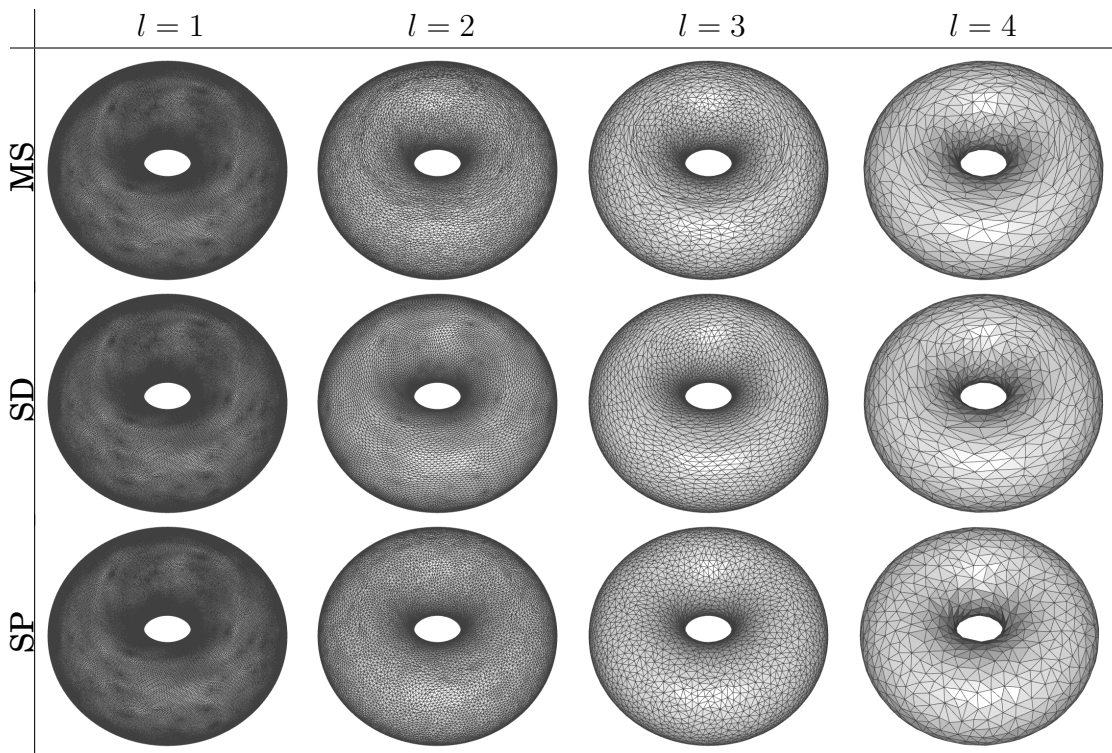


Figure A.3: The three mesh hierarchies of the torus model.

APPENDIX A. INPUT HIERARICHIAL MESHES

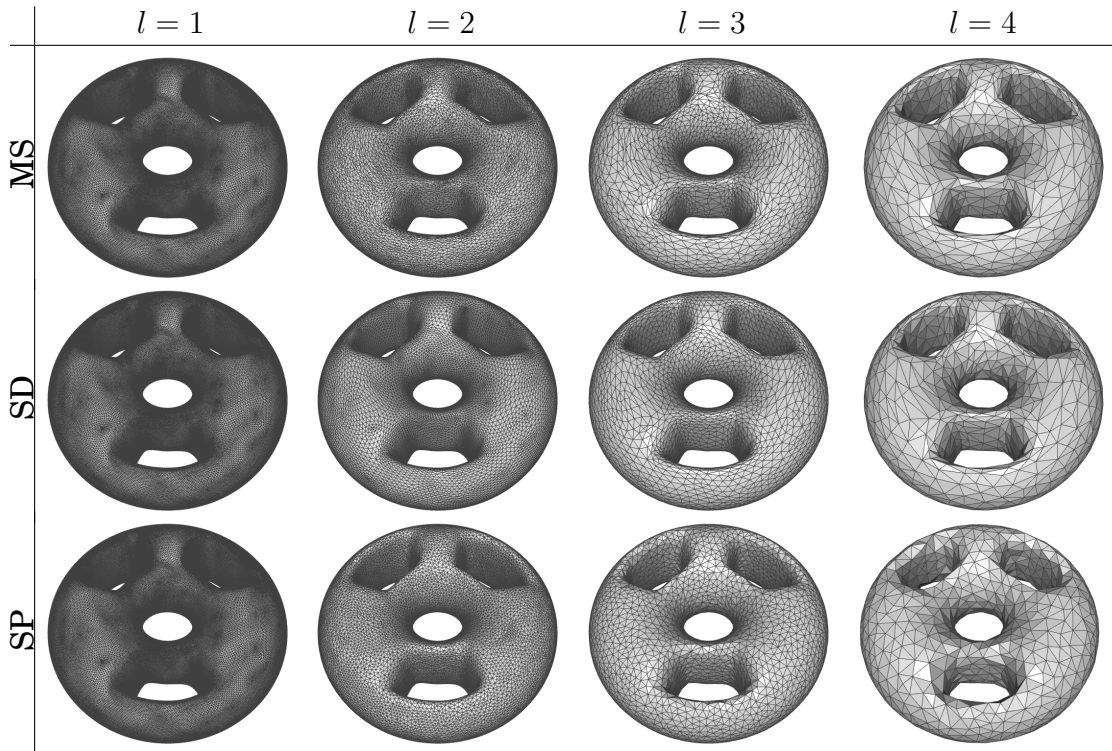


Figure A.4: The three mesh hierarchies of the 4-torus model.

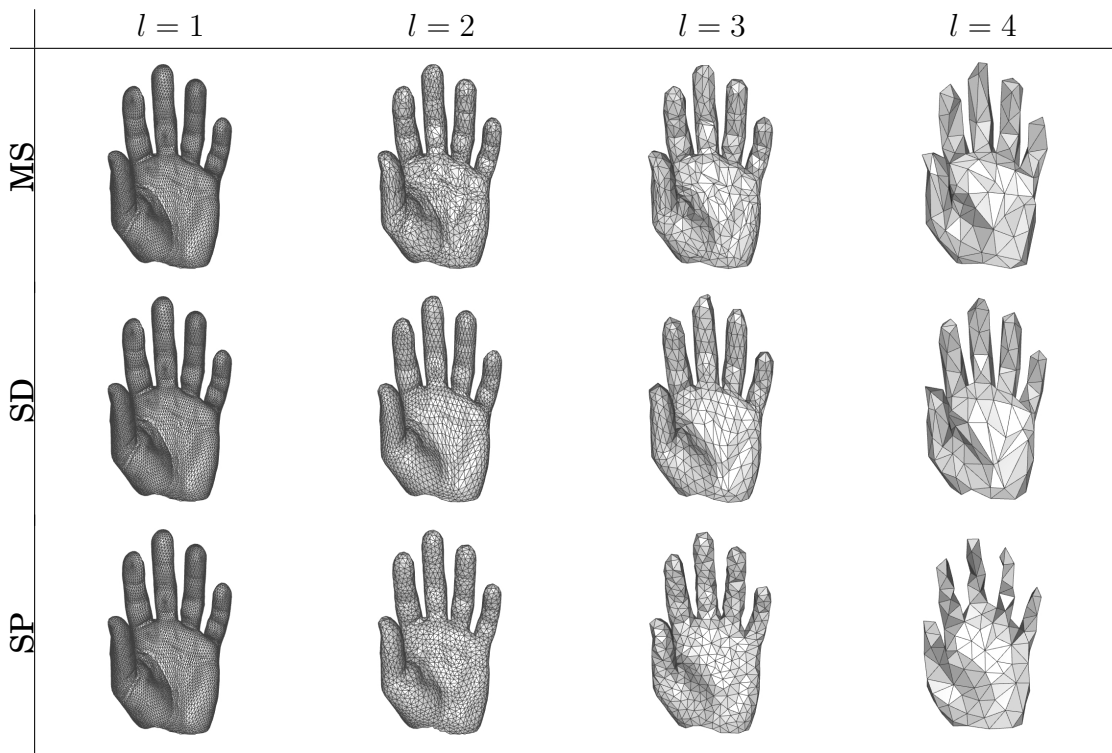


Figure A.5: The three mesh hierarchies of the hand model.

APPENDIX A. INPUT HIERARICHIAL MESHES

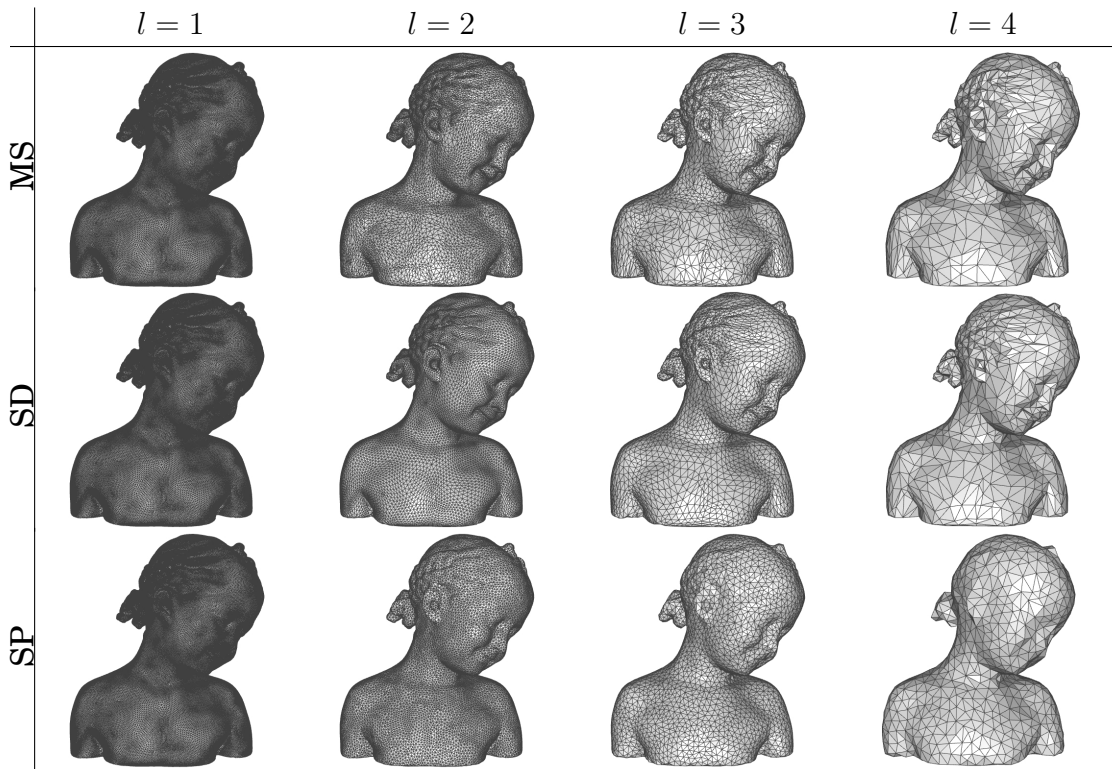


Figure A.6: The three mesh hierarchies of the bimba model.

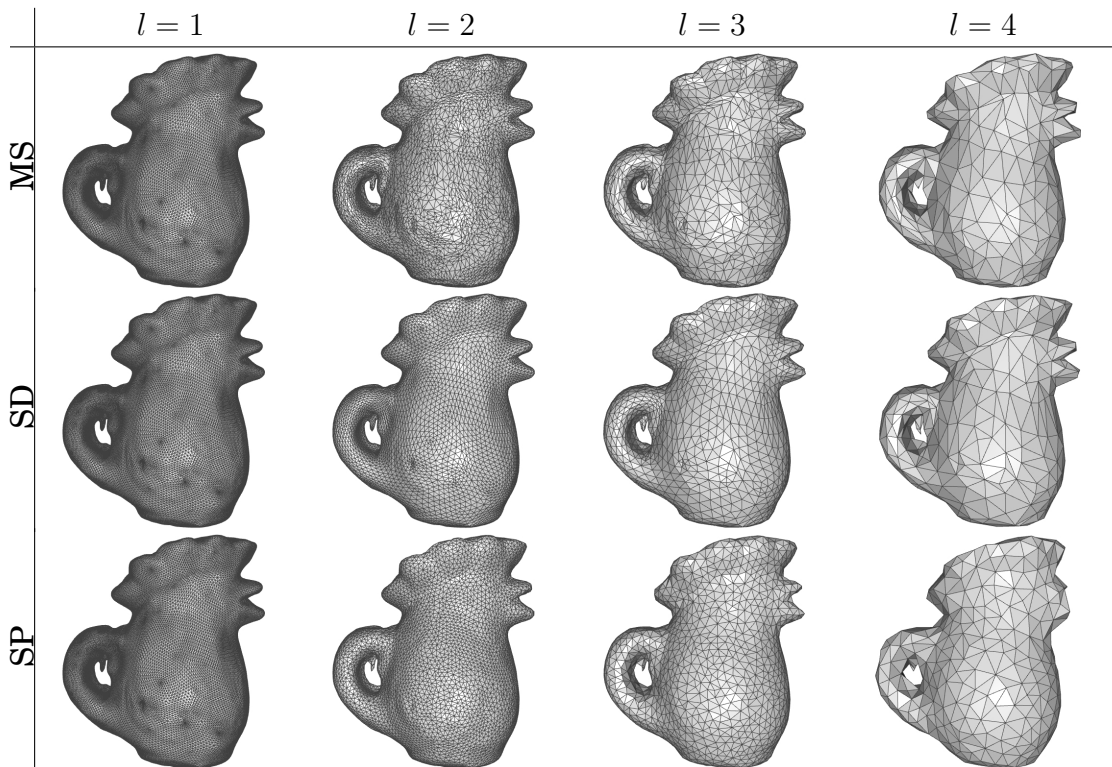


Figure A.7: The three mesh hierarchies of the rooster model.

APPENDIX A. INPUT HIERARICHIAL MESHES

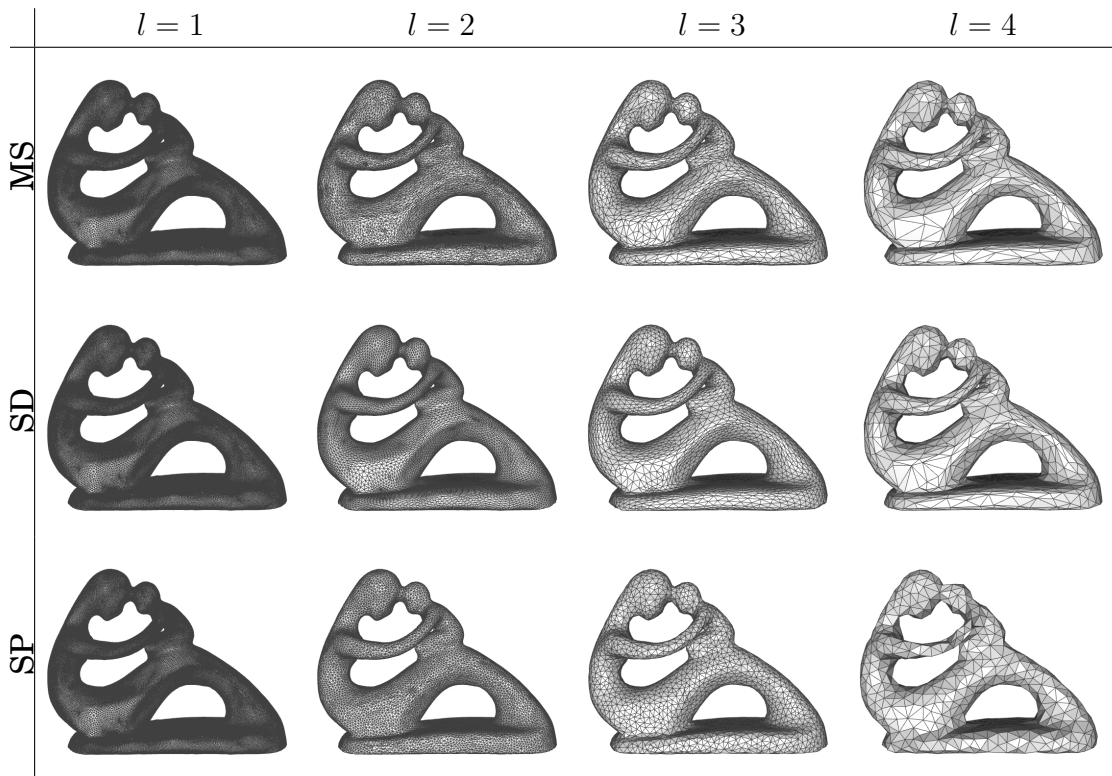


Figure A.8: The three mesh hierarchies of the fertility model.

Appendix B

1-Form System Convergence (Standard V-Cycle)

In this appendix, we provide the convergence plots in a long run (six hundred iterations) in Fig. B.1. We observe that the convergence highly depends on the choice of the hierarchy and the basis. The SP hierarchy gives faster and more consistent convergence across all models. In general, when the timestep α is small, using the Whitney basis converges faster than the harmonic-free basis, and the choice of hierarchy has less impact on the Whitney basis. However, when α is big, using harmonic-free basis + MS/SP converges faster than using the Whitney basis. The effect of the input models on the convergence also depends on the choice of the hierarchy and basis. There is no consistent observable pattern. Yet, we note that, for the Whitney basis, the higher-genus

APPENDIX B. 1-FORM SYSTEM CONVERGENCE (STANDARD V-CYCLE)

models (4-torus and fertility) tend to have slower convergence, while, for the harmonic basis, the sphere and torus tend to have faster convergence.

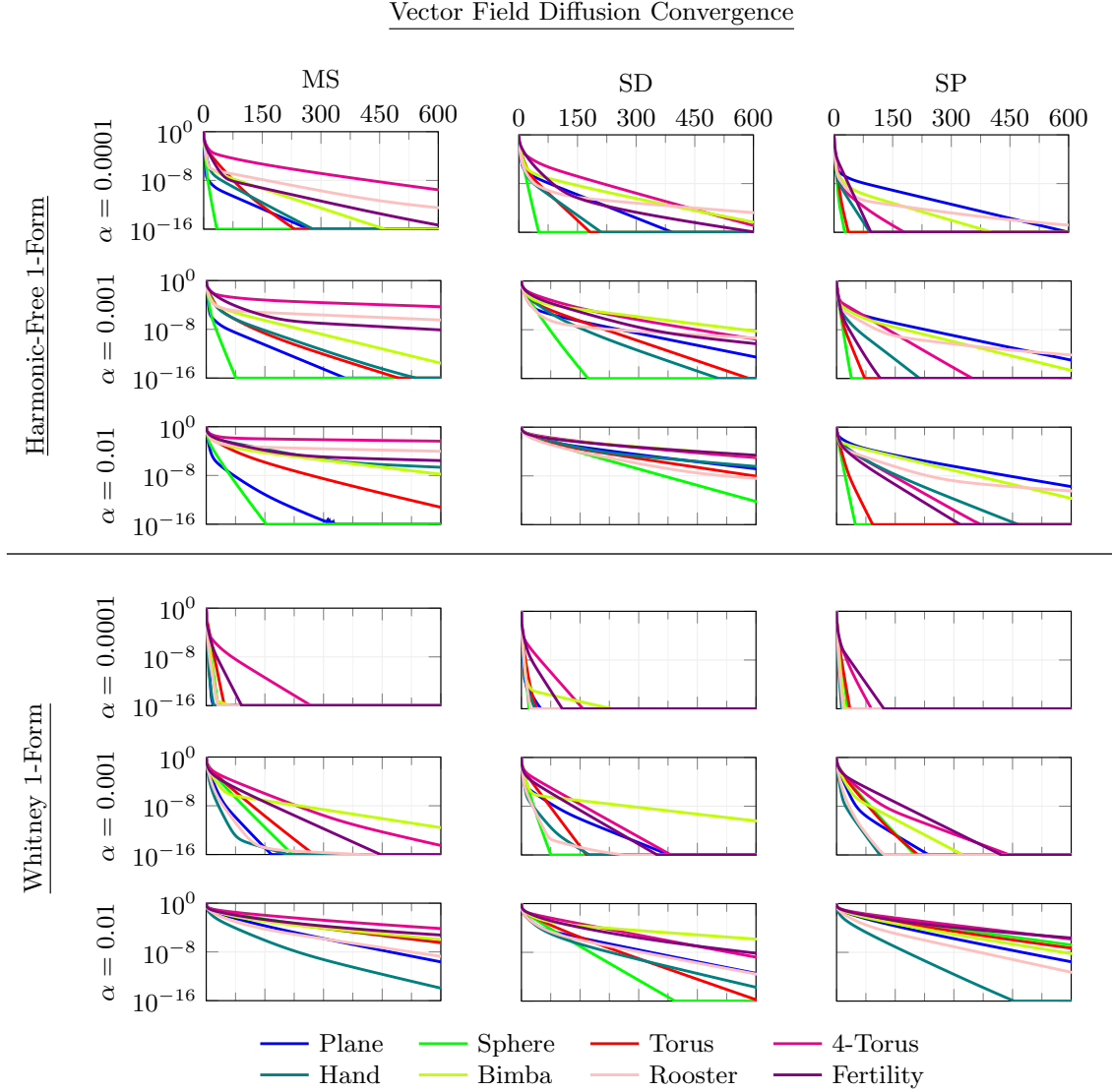


Figure B.1: The convergence rate of the standard multigrid method when solving an implicit vector diffusion step with varies timestep α from 0.0001 to 0.01 using the hierarchies mesh simplification (MS), sub-division (SD), and self-intrinsic parameterization (SP), as well as the harmonic-free and Whitney 1-form basis in six hundred iterations.

Appendix C

Parameter Study

In Chapter 6, we study and explore different multigrid components and conclude that successive over-relaxation (SOR) works best as the relaxation scheme, smoothing the prolongation matrix provides a small improvement in convergence for some models, and the integration of Krylov subspace solution update significantly improves the performance. This appendix supplements details of the study for different models and parameters.

SOR Relaxation Weight

The relaxation weight does not have a significant effect on the harmonic-free basis, as depicted in Fig. C.1. Nevertheless, for the Whitney basis, Fig. C.2 shows that using the relaxation weight between 1.6 and 1.8 consistently speeds up the convergence for all models. However, as α is increased, the improvement

APPENDIX C. PARAMETER STUDY

is negligible.

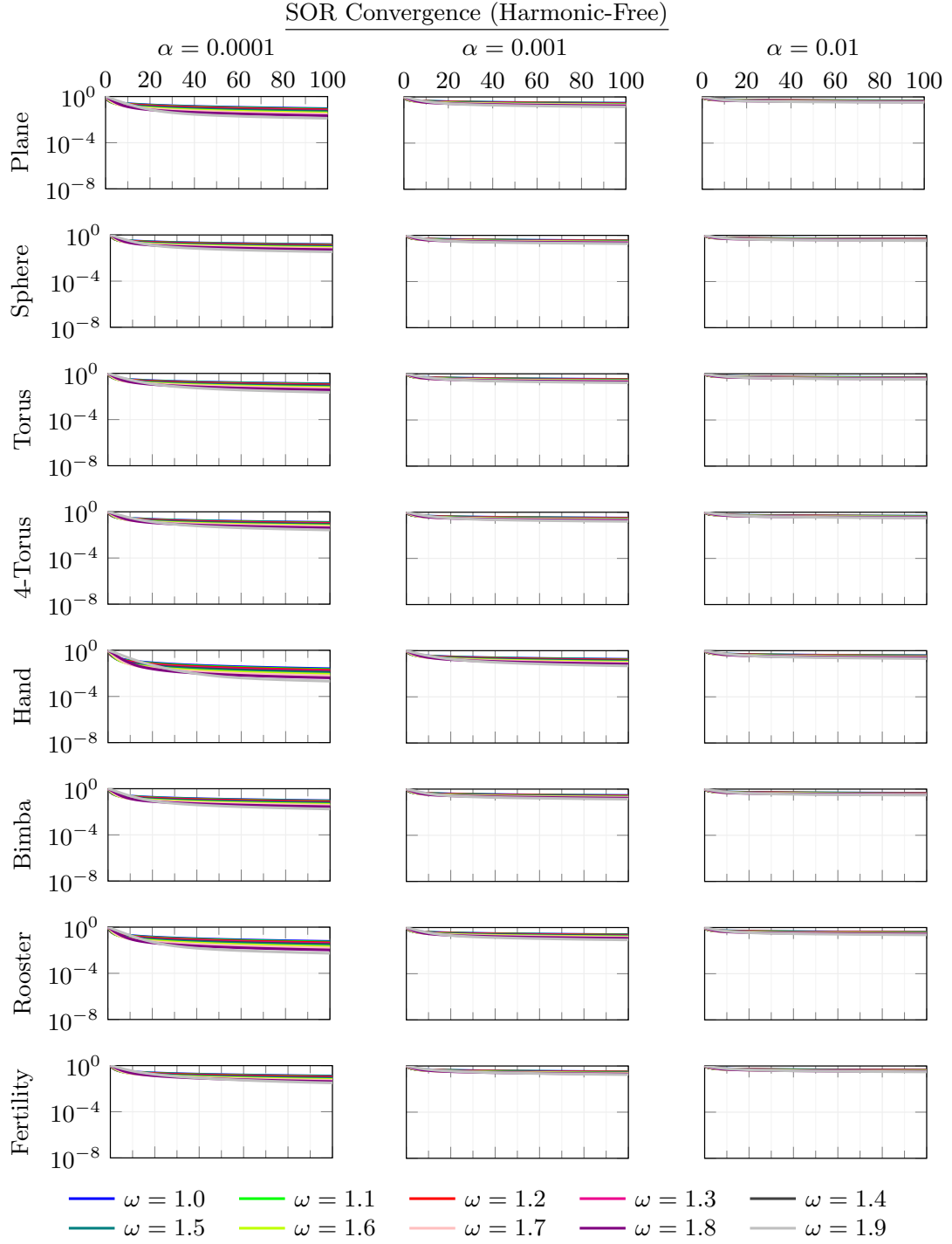


Figure C.1: SOR convergence of solving a vector diffusion problem using the harmonic-free basis with different timestep α .

APPENDIX C. PARAMETER STUDY

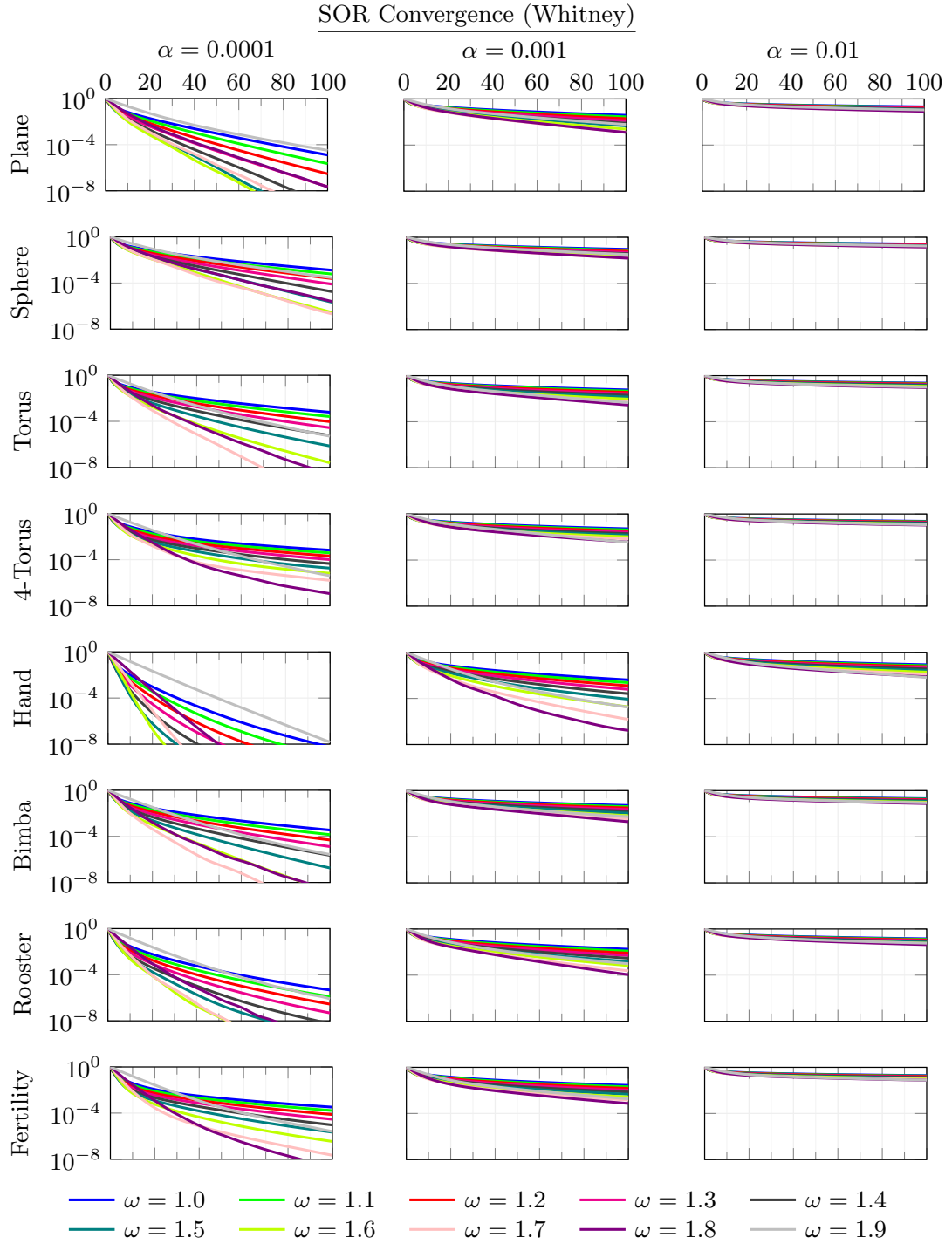


Figure C.2: SOR convergence of solving a vector diffusion problem using the Whitney basis with different timestep α .

APPENDIX C. PARAMETER STUDY

SOR Relaxation Improvement

Figs. C.3 to C.5 and Figs. C.6 to C.8 demonstrate the convergence of our solver using different SOR relaxation weights for the harmonic-free and Whitney bases. While $\omega_{SOR} = 1.7$ works for most models, we also explore the parameter and find the optimal ω_{SOR} for each combination of models, hierarchies, bases, and timesteps α . In practice, we find a bigger ω_{SOR} tends to work better as α_{SOR} is increased; and smaller ω_{SOR} is better for meshes with boundaries (such as plane and hand).

For the harmonic-free basis, as discussed, although SOR does not have much difference when used in a single-level solver, it does improve the multi-level solver convergence. For a few combinations (such as plane + MS and sphere + SP), it also helps slightly, even for a larger timestep ($\alpha = 0.1$). However, comparatively, the improvement is not as significant as that of the Whitney basis.

For the Whitney basis, we observe substantial improvement in using SOR as the relaxation scheme for $\alpha = 0.0001$. The improvement is considerably consistent across different hierarchies and timesteps. It is still reasonably helpful for $\alpha = 0.001$. However, when $\alpha \geq 0.01$, the improvement is inconsiderable.

Table C.1 shows the convergence improvement using the optimal relaxation weight for each combination for $\alpha = 0.0001$.

APPENDIX C. PARAMETER STUDY

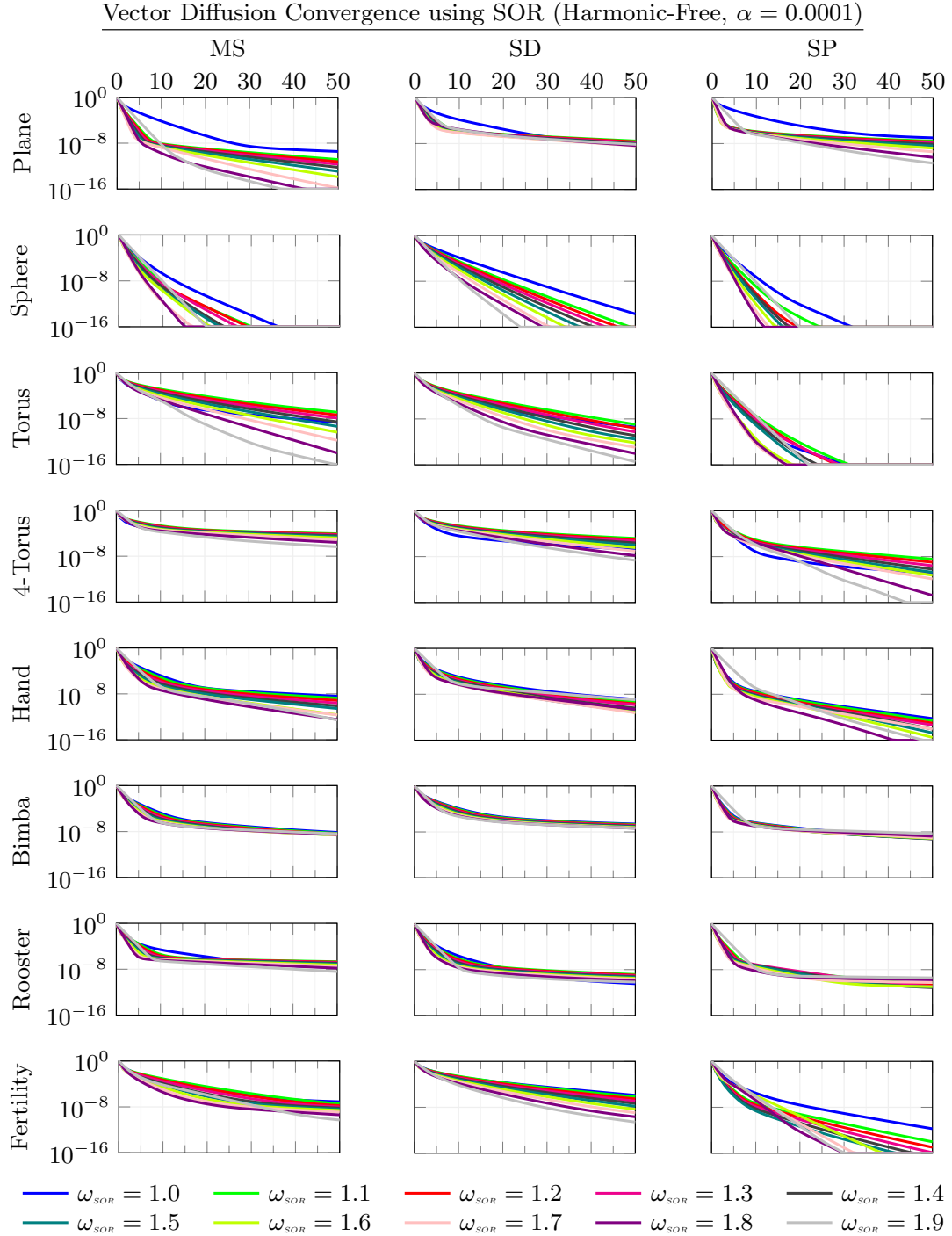


Figure C.3: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.0001$ with different SOR relaxation weights.

APPENDIX C. PARAMETER STUDY

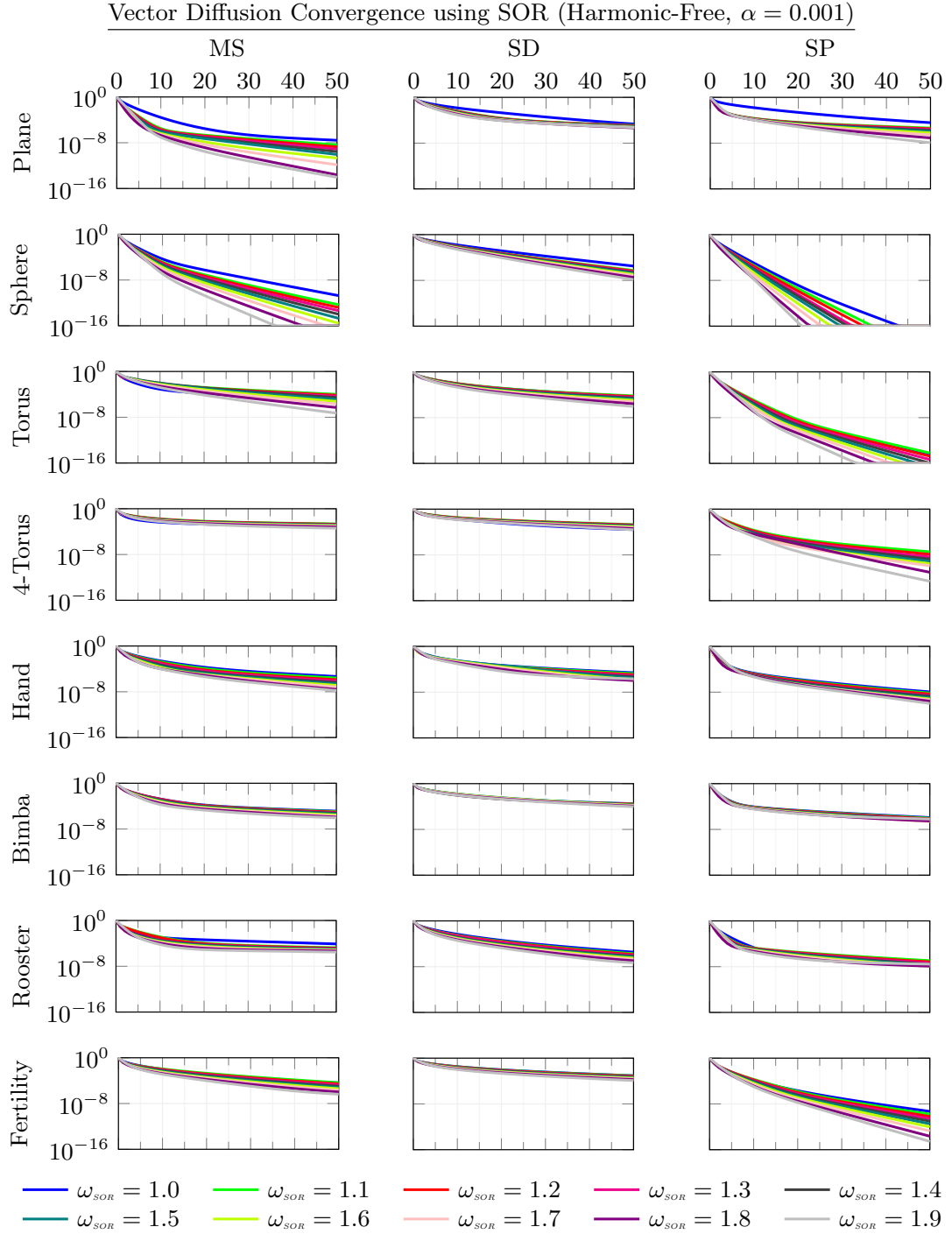


Figure C.4: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.001$ with different SOR relaxation weights.

APPENDIX C. PARAMETER STUDY

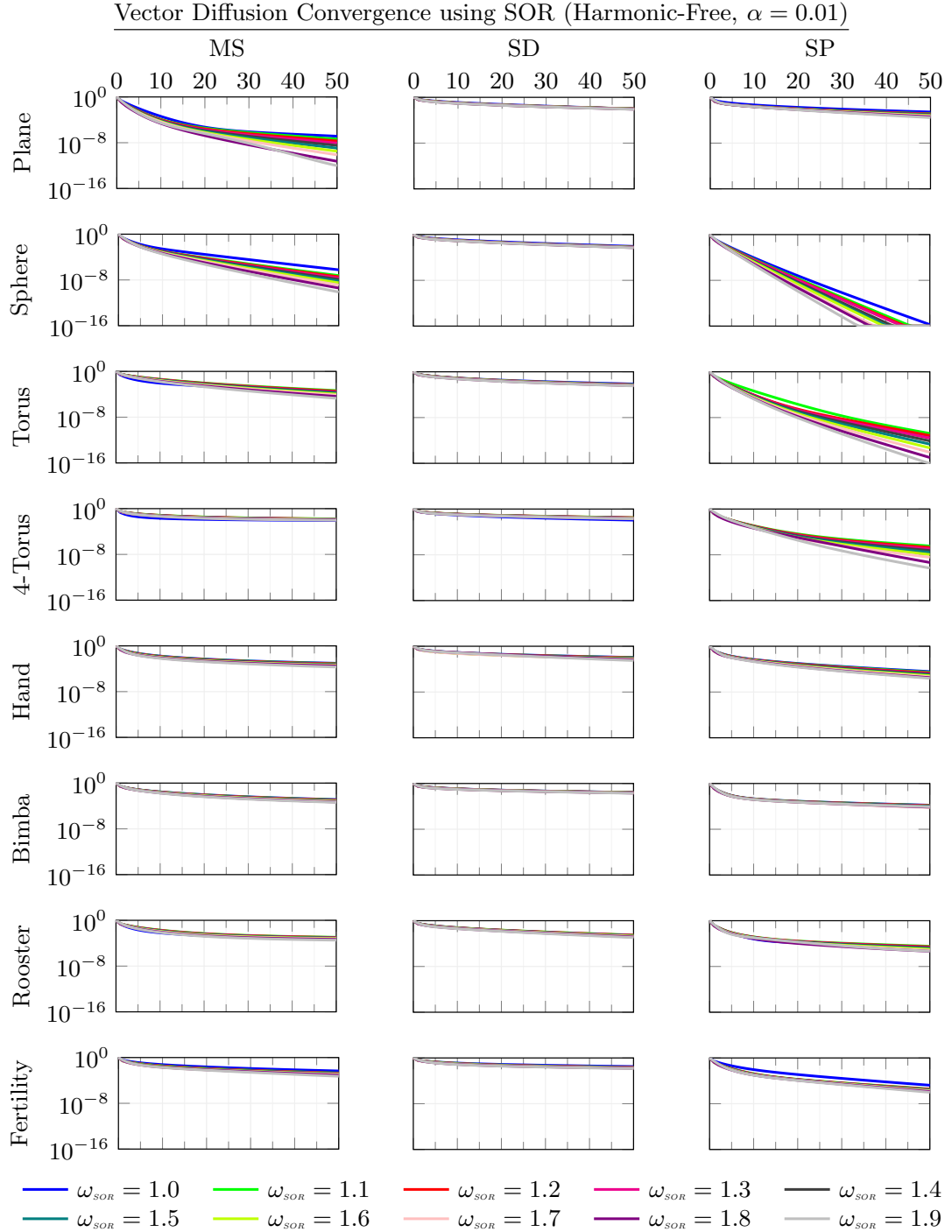


Figure C.5: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.01$ with different SOR relaxation weights.

APPENDIX C. PARAMETER STUDY

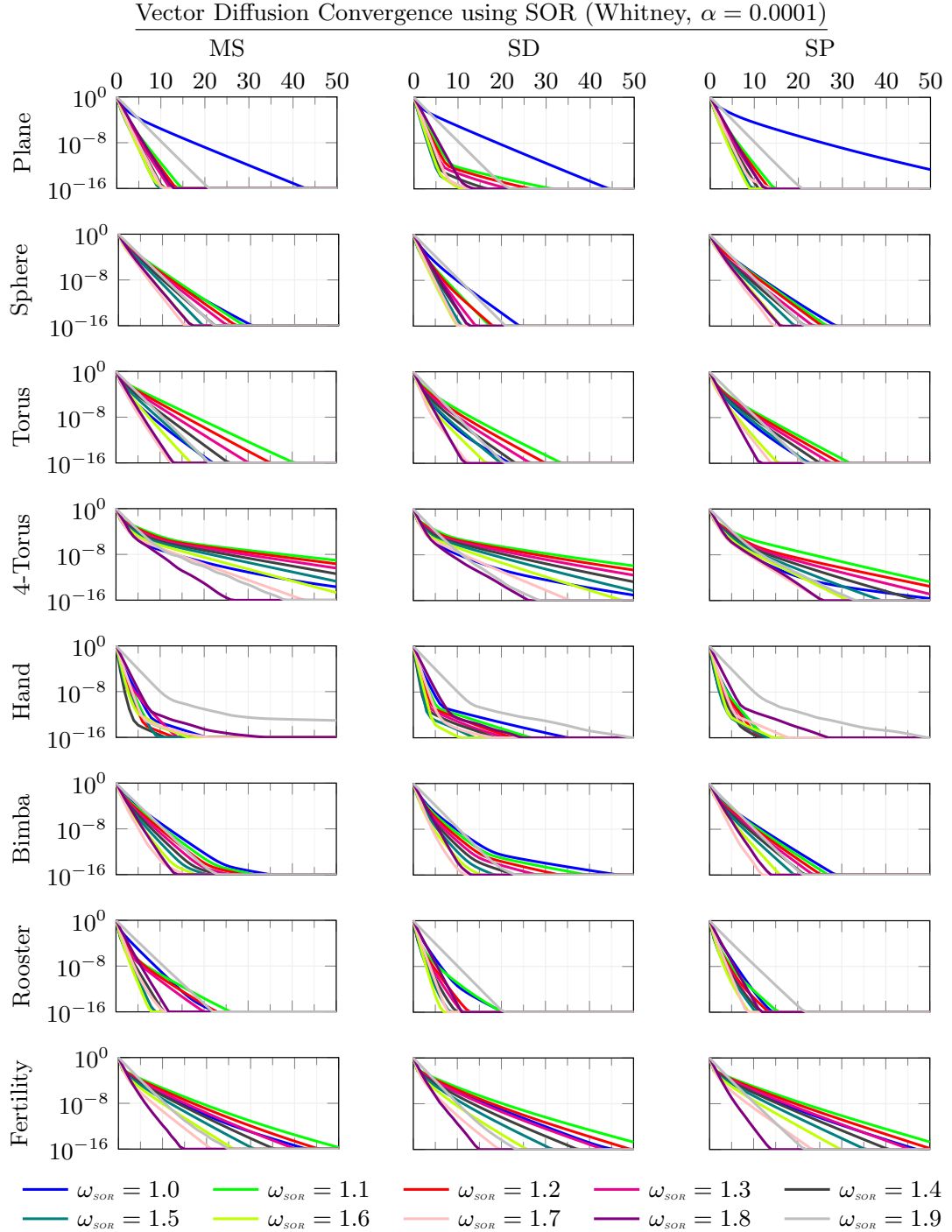


Figure C.6: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.0001$ with different SOR relaxation weights.

APPENDIX C. PARAMETER STUDY

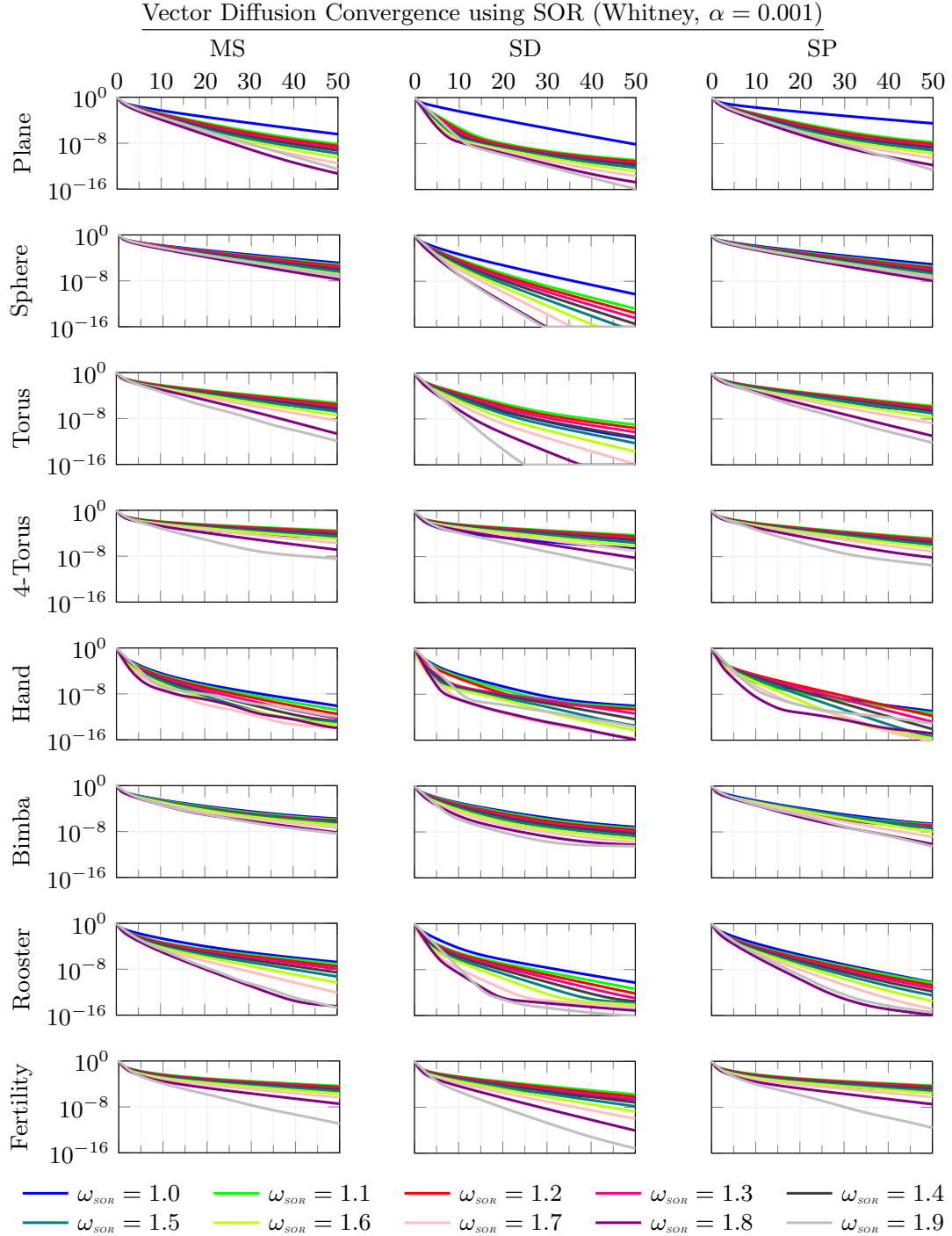


Figure C.7: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.001$ with different SOR relaxation weights.

APPENDIX C. PARAMETER STUDY

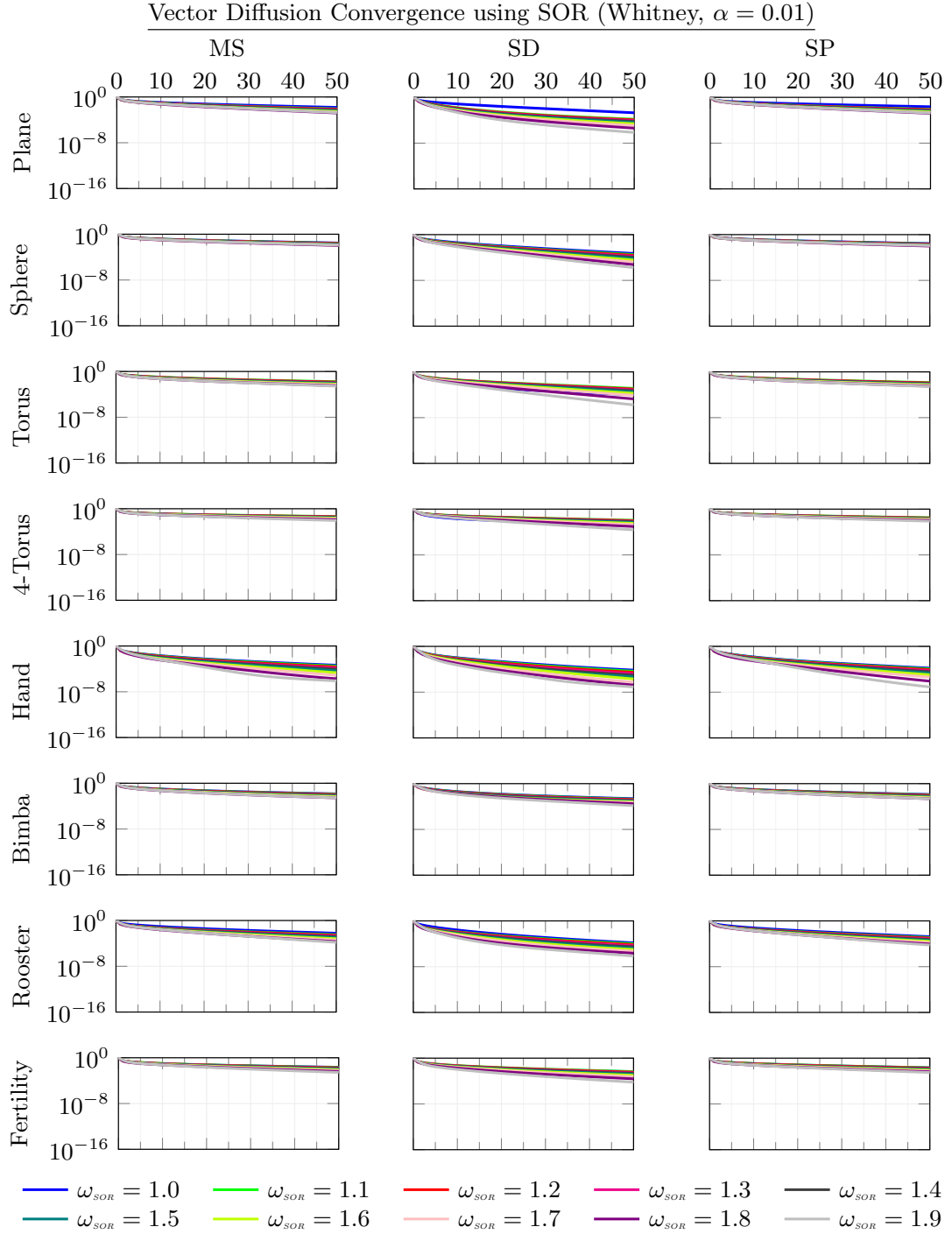


Figure C.8: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.01$ with different SOR relaxation weights.

APPENDIX C. PARAMETER STUDY

	HS	ω_{SOR}		$E(\mathbf{x})$ after 50 itr. s.		# of itr. s $E(\mathbf{x}) < 10^{-8}$					
		HF	WN	HF		WN		HF		WN	
Plane	MS	1.8	1.4	$3.6 \cdot 10^{-10} \Rightarrow 5.1 \cdot 10^{-18}$	$6.1 \cdot 10^{-19} \Rightarrow 1.2 \cdot 10^{-28}$	27	\Rightarrow	6	18	\Rightarrow	5
	SD	1.7	1.4	$9.7 \cdot 10^{-09} \Rightarrow 3.9 \cdot 10^{-09}$	$2.9 \cdot 10^{-18} \Rightarrow 1.5 \cdot 10^{-23}$	50	\Rightarrow	43	20	\Rightarrow	4
	SP	1.9	1.6	$1.0 \cdot 10^{-07} \Rightarrow 3.8 \cdot 10^{-12}$	$2.3 \cdot 10^{-13} \Rightarrow 6.2 \cdot 10^{-29}$	> 50	\Rightarrow	21	26	\Rightarrow	4
Sphere	MS	1.7	1.7	$4.5 \cdot 10^{-21} \Rightarrow 2.4 \cdot 10^{-24}$	$1.8 \cdot 10^{-24} \Rightarrow 1.3 \cdot 10^{-30}$	13	\Rightarrow	6	13	\Rightarrow	7
	SD	1.9	1.5	$2.0 \cdot 10^{-14} \Rightarrow 2.6 \cdot 10^{-24}$	$1.4 \cdot 10^{-30} \Rightarrow 8.4 \cdot 10^{-31}$	27	\Rightarrow	11	10	\Rightarrow	5
	SP	1.7	1.7	$1.6 \cdot 10^{-22} \Rightarrow 6.9 \cdot 10^{-24}$	$9.2 \cdot 10^{-26} \Rightarrow 1.1 \cdot 10^{-30}$	12	\Rightarrow	6	13	\Rightarrow	7
Torus	MS	1.9	1.7	$2.1 \cdot 10^{-09} \Rightarrow 8.7 \cdot 10^{-17}$	$2.7 \cdot 10^{-27} \Rightarrow 8.9 \cdot 10^{-31}$	40	\Rightarrow	18	9	\Rightarrow	6
	SD	1.9	1.7	$3.8 \cdot 10^{-10} \Rightarrow 3.5 \cdot 10^{-16}$	$3.1 \cdot 10^{-26} \Rightarrow 1.4 \cdot 10^{-30}$	35	\Rightarrow	16	8	\Rightarrow	5
	SP	1.7	1.7	$1.3 \cdot 10^{-21} \Rightarrow 1.1 \cdot 10^{-24}$	$8.2 \cdot 10^{-27} \Rightarrow 1.3 \cdot 10^{-30}$	10	\Rightarrow	6	8	\Rightarrow	6
4-Torus	MS	1.9	1.8	$1.3 \cdot 10^{-05} \Rightarrow 4.5 \cdot 10^{-07}$	$2.1 \cdot 10^{-14} \Rightarrow 2.3 \cdot 10^{-20}$	> 50	\Rightarrow	> 50	14	\Rightarrow	10
	SD	1.9	1.8	$1.6 \cdot 10^{-07} \Rightarrow 2.2 \cdot 10^{-09}$	$9.0 \cdot 10^{-16} \Rightarrow 4.4 \cdot 10^{-27}$	> 50	\Rightarrow	44	11	\Rightarrow	9
	SP	1.0	1.7	$1.3 \cdot 10^{-11} \Rightarrow 1.3 \cdot 10^{-11}$	$2.2 \cdot 10^{-16} \Rightarrow 3.6 \cdot 10^{-24}$	14	\Rightarrow	14	9	\Rightarrow	9
Hand	MS	1.8	1.4	$3.9 \cdot 10^{-09} \Rightarrow 3.4 \cdot 10^{-13}$	$6.5 \cdot 10^{-22} \Rightarrow 3.2 \cdot 10^{-22}$	40	\Rightarrow	14	5	\Rightarrow	2
	SD	1.7	1.4	$1.3 \cdot 10^{-09} \Rightarrow 6.0 \cdot 10^{-12}$	$3.7 \cdot 10^{-19} \Rightarrow 3.7 \cdot 10^{-26}$	37	\Rightarrow	21	5	\Rightarrow	2
	SP	1.7	1.3	$5.9 \cdot 10^{-13} \Rightarrow 8.5 \cdot 10^{-15}$	$1.5 \cdot 10^{-26} \Rightarrow 1.8 \cdot 10^{-30}$	14	\Rightarrow	8	5	\Rightarrow	3
Bimba	MS	1.5	1.7	$7.1 \cdot 10^{-09} \Rightarrow 2.5 \cdot 10^{-09}$	$9.2 \cdot 10^{-19} \Rightarrow 3.3 \cdot 10^{-24}$	47	\Rightarrow	34	12	\Rightarrow	6
	SD	1.8	1.6	$2.4 \cdot 10^{-07} \Rightarrow 4.3 \cdot 10^{-08}$	$4.1 \cdot 10^{-17} \Rightarrow 1.4 \cdot 10^{-23}$	> 50	\Rightarrow	> 50	10	\Rightarrow	5
	SP	1.4	1.7	$9.8 \cdot 10^{-10} \Rightarrow 6.0 \cdot 10^{-10}$	$3.0 \cdot 10^{-22} \Rightarrow 1.7 \cdot 10^{-28}$	30	\Rightarrow	24	12	\Rightarrow	5
Rooster	MS	1.9	1.5	$2.1 \cdot 10^{-08} \Rightarrow 3.8 \cdot 10^{-09}$	$9.0 \cdot 10^{-25} \Rightarrow 3.5 \cdot 10^{-30}$	> 50	\Rightarrow	41	9	\Rightarrow	4
	SD	1.9	1.6	$3.5 \cdot 10^{-11} \Rightarrow 6.6 \cdot 10^{-11}$	$1.2 \cdot 10^{-21} \Rightarrow 7.4 \cdot 10^{-25}$	22	\Rightarrow	11	7	\Rightarrow	3
	SP	1.8	1.4	$1.7 \cdot 10^{-11} \Rightarrow 1.7 \cdot 10^{-10}$	$3.1 \cdot 10^{-30} \Rightarrow 2.2 \cdot 10^{-29}$	13	\Rightarrow	9	7	\Rightarrow	4
Fertility	MS	1.8	1.8	$7.9 \cdot 10^{-08} \Rightarrow 4.1 \cdot 10^{-10}$	$3.3 \cdot 10^{-18} \Rightarrow 3.8 \cdot 10^{-30}$	> 50	\Rightarrow	25	15	\Rightarrow	7
	SD	1.9	1.8	$1.2 \cdot 10^{-06} \Rightarrow 2.7 \cdot 10^{-11}$	$3.2 \cdot 10^{-17} \Rightarrow 7.2 \cdot 10^{-30}$	> 50	\Rightarrow	30	17	\Rightarrow	7
	SP	1.5	1.8	$1.8 \cdot 10^{-12} \Rightarrow 2.2 \cdot 10^{-18}$	$2.4 \cdot 10^{-17} \Rightarrow 1.1 \cdot 10^{-30}$	23	\Rightarrow	9	17	\Rightarrow	6

Table C.1: The convergence improvement using SOR with best over-relaxation weights ω_{SOR} shown in the third and fourth columns for the harmonic-free (HF) and Whitney (WN) bases. The next two columns show the changes in the relative residual errors after fifty iterations before and after the speed up, and the last two columns show the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .

APPENDIX C. PARAMETER STUDY

Prolongation Smoothing Improvement

Figs. C.9 to C.11 and Figs. C.12 to C.14 illustrate the effect of prolongation smoothing on the convergence of our solver for the harmonic-free and Whitney bases when solving the diffusion problem. Although prolongation smoothing is a standard technique in algebraic multigrid to construct smooth prolongation, we find that this technique does not help with the convergence for the harmonic-free basis. As described in Chapter 6, prolongation smoothing destroys the diagonal-like structure of the coarse-fine eigenfunction correlation matrix, resulting in poor solver performance.

Nevertheless, in the case of the Whitney basis, it helps for certain models (plane, torus, hand, and bimba), in particular, the plane. However, the smoothing weight varies with the input models and the choice of the hierarchy.

Similar to using SOR as the relaxation scheme, the improvement narrows down as α is increased. However, unlike SOR, for obtaining our desired convergence (accuracy of 10^{-8} within six iterations), prolongation smoothing yields typically zero to two iterations improvement (except the plane).

APPENDIX C. PARAMETER STUDY

Vector Diffusion Convergence using Prolongation Smoothing (Harmonic-Free, $\alpha = 0.0001$)

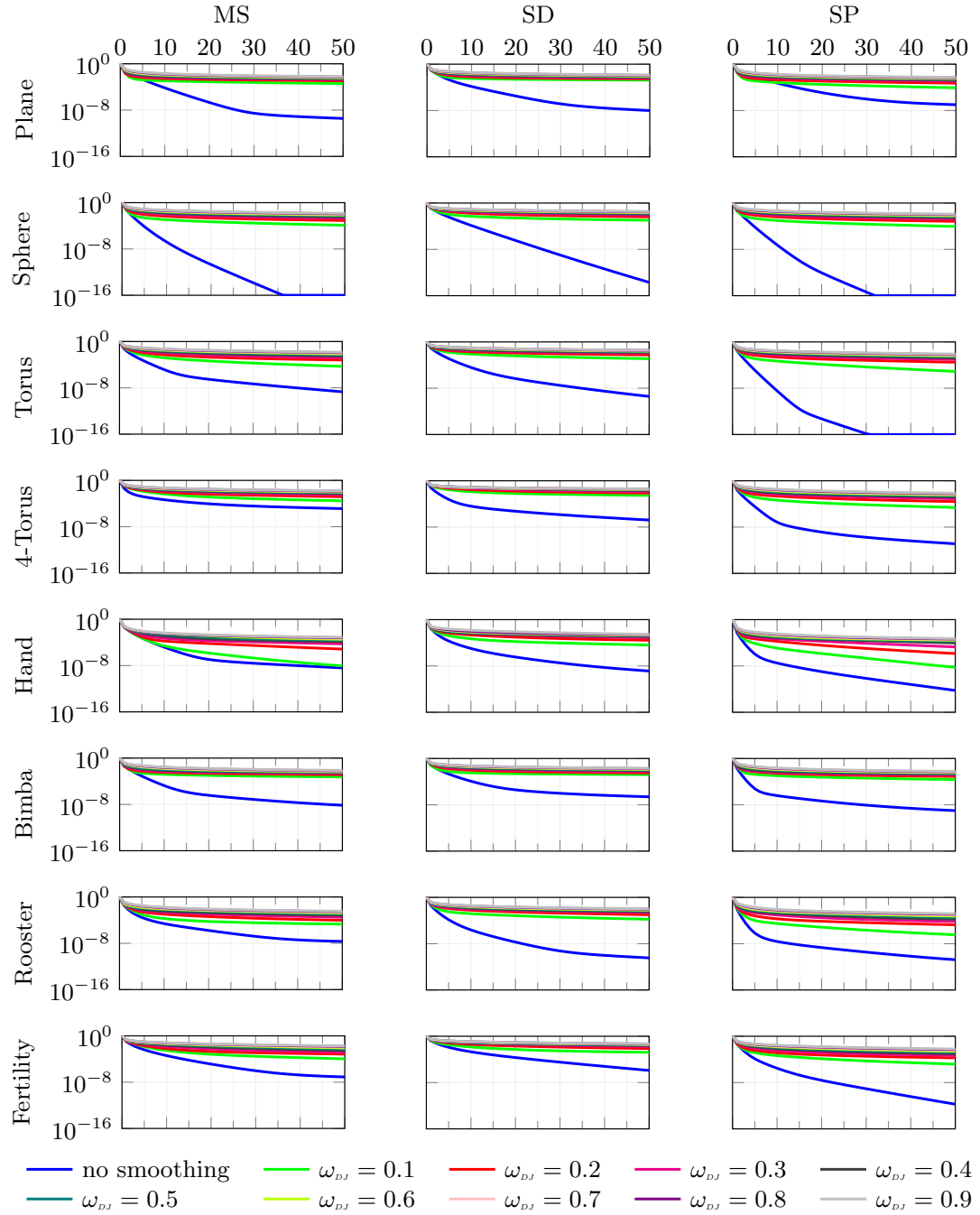


Figure C.9: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.0001$ with different prolongation smoothing weights.

APPENDIX C. PARAMETER STUDY

Vector Diffusion Convergence using Prolongation Smoothing (Harmonic-Free, $\alpha = 0.001$)

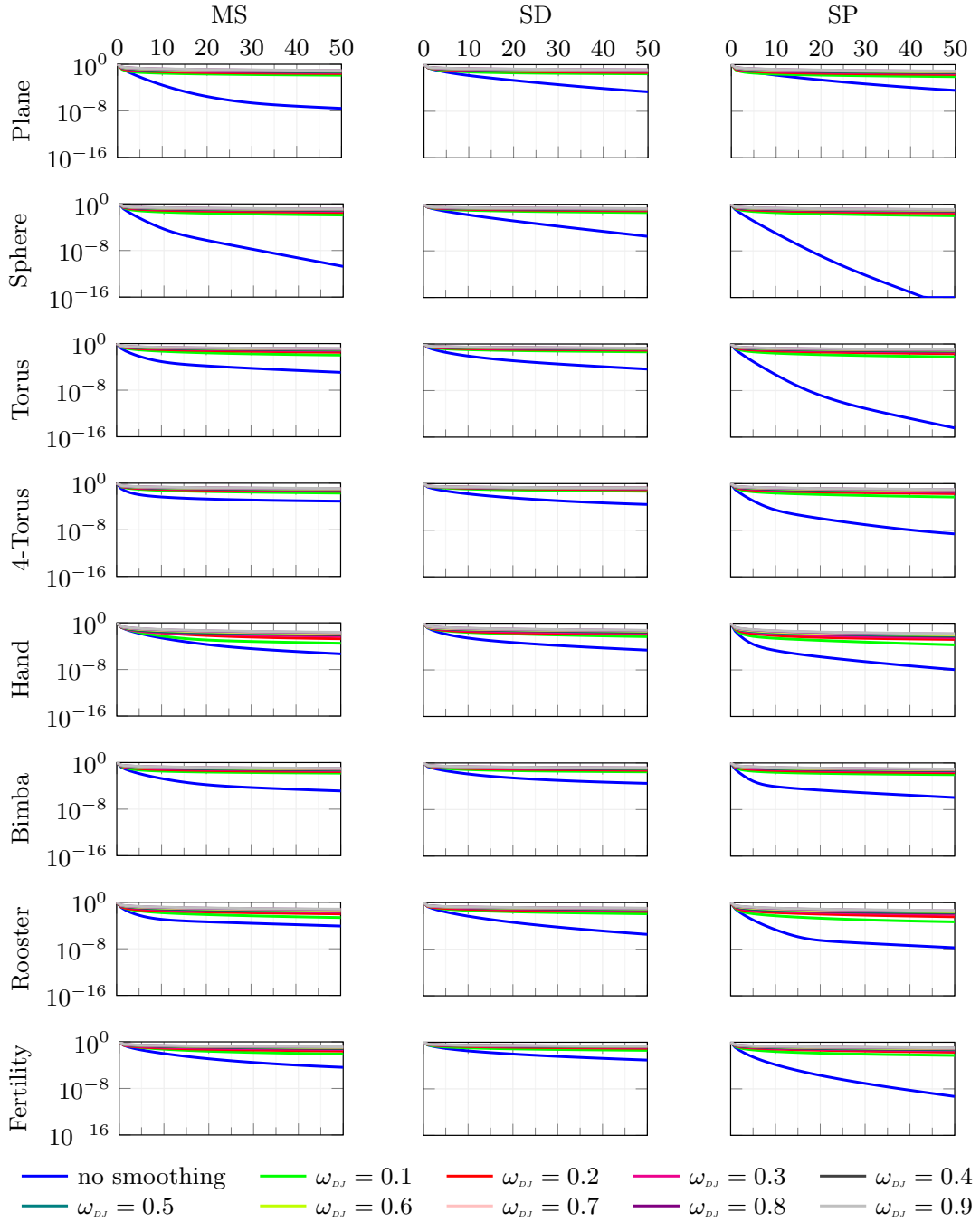


Figure C.10: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.001$ with different prolongation smoothing weights.

APPENDIX C. PARAMETER STUDY

Vector Diffusion Convergence using Prolongation Smoothing (Harmonic-Free, $\alpha = 0.01$)

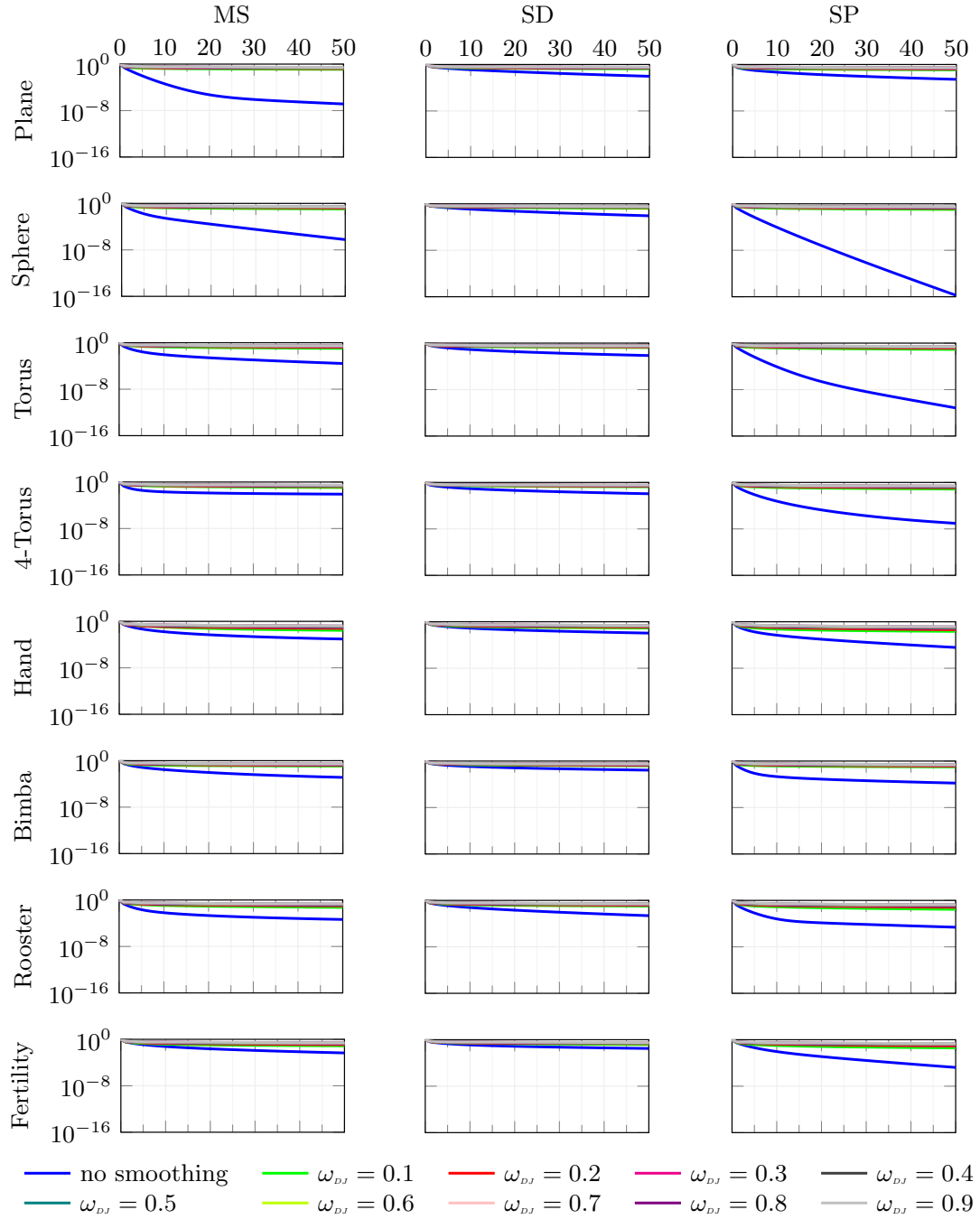


Figure C.11: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.01$ with different prolongation smoothing weights.

APPENDIX C. PARAMETER STUDY

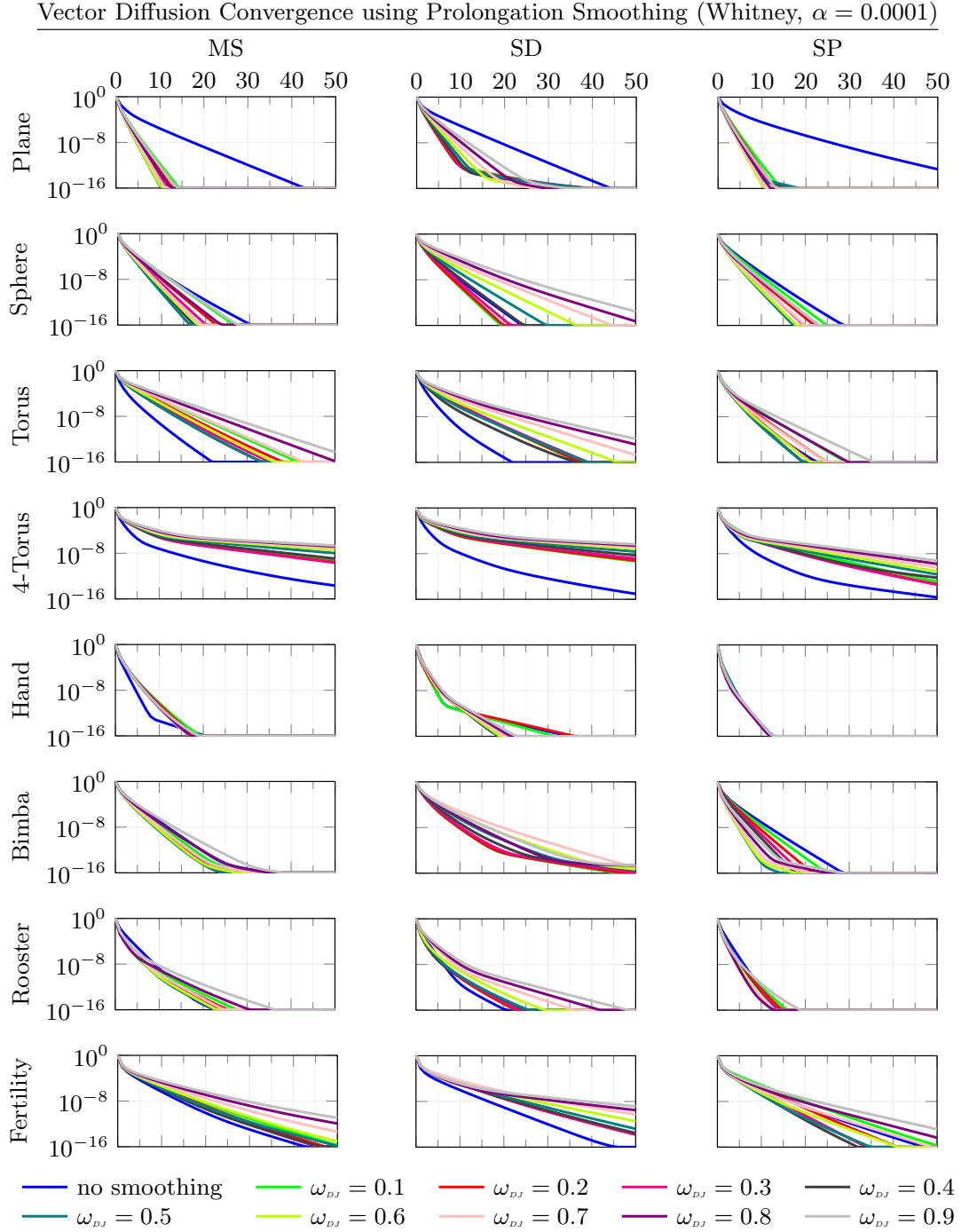


Figure C.12: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.0001$ with different prolongation smoothing weights.

APPENDIX C. PARAMETER STUDY

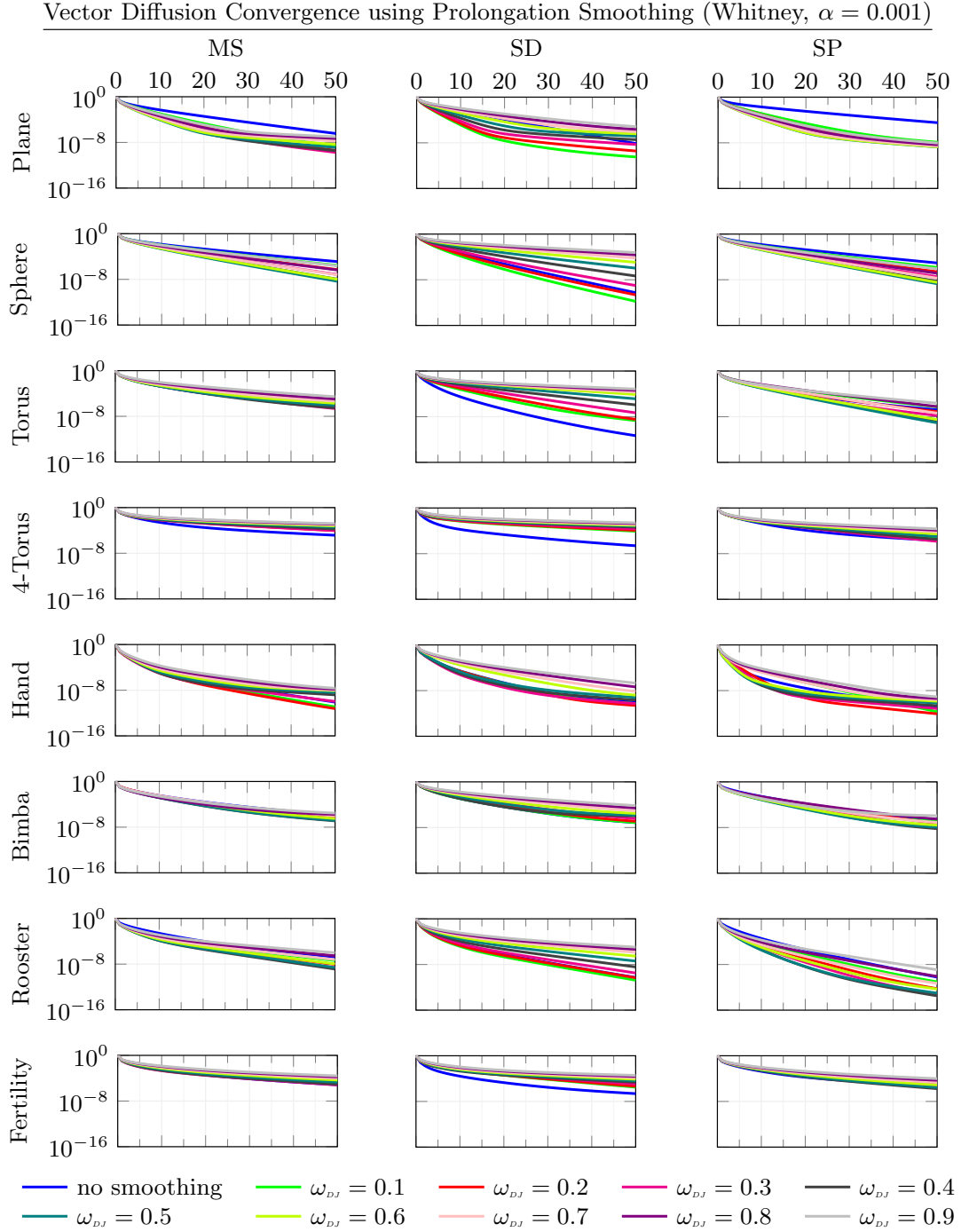


Figure C.13: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.001$ with different prolongation smoothing weights.

APPENDIX C. PARAMETER STUDY

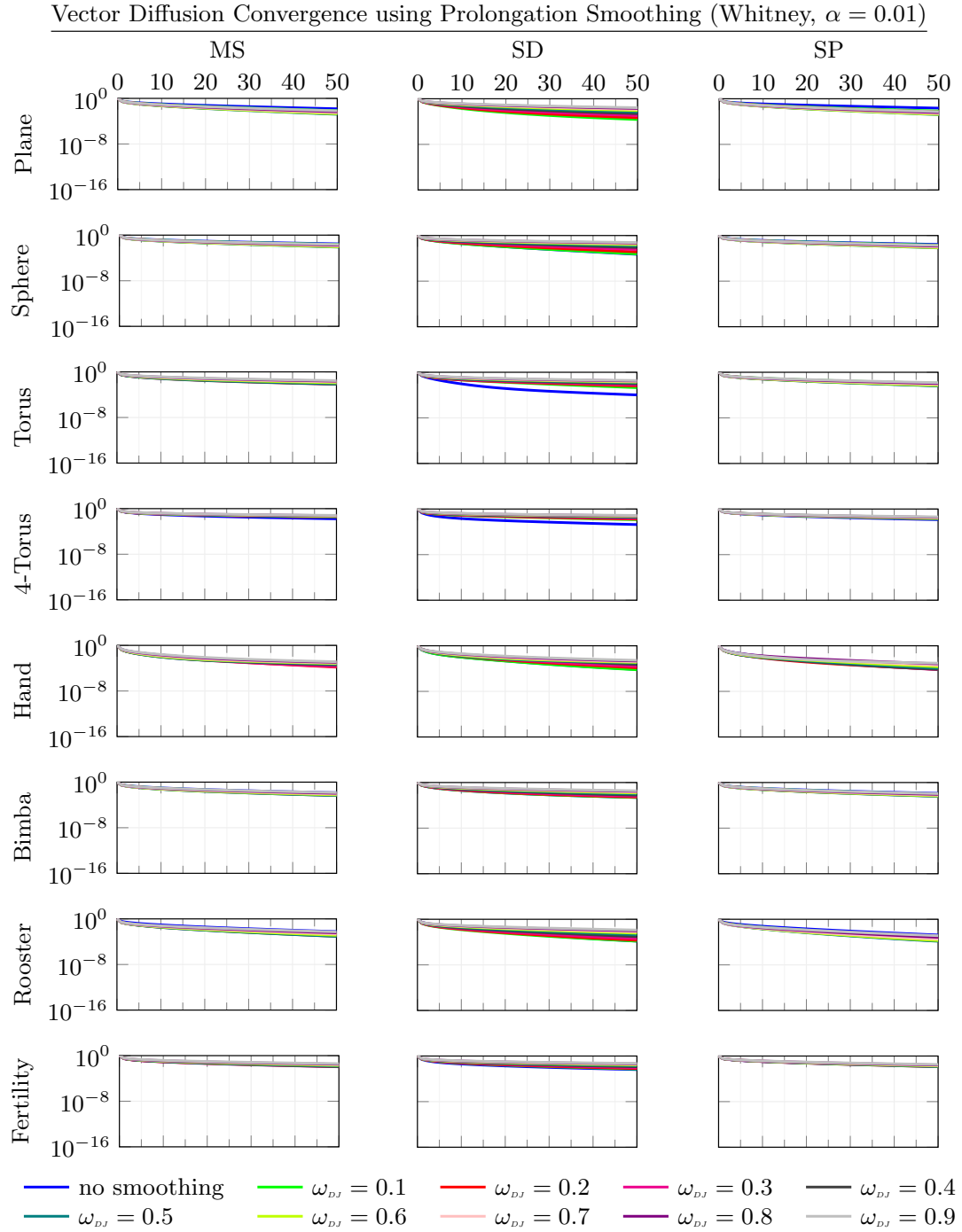


Figure C.14: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.01$ with different prolongation smoothing weights.

APPENDIX C. PARAMETER STUDY

Krylov Subspace Updates Improvement

Figs. C.15 to C.17 and Figs. C.18 to C.20 are plots of convergence of our solver after integrating the Krylov subspace updates, demonstrating the effect of various subspace dimensions. This technique substantially improves the convergence for both harmonic-free and Whitney basis consistently across different hierarchies, and models. Furthermore, it also improves the convergence even for a larger α .

One limitation of this technique is that the subspace quickly becomes close to linearly dependent when the successive solutions are relatively close to each other. In this case, our implementation obtains numerically unstable results. In the figures, there is an obvious change of slope of the curves throughout the iterations. That is when our solver struggles to obtain a better solution using Krylov subspace updates.

For most combinations and timesteps, we observe no significant difference when the subspace dimension is bigger than three. Our empirical results show that keeping the subspace dimension at ten works well for most combinations. However, for individual combinations such as hand + Whitney + MS and 4-torus + harmonic-free + SP, using another subspace dimension might gain additional one to two iterations improvement.

Table C.2 shows the convergence improvement using the optimal subspace dimension for each combination for $\alpha = 0.0001$.

APPENDIX C. PARAMETER STUDY

Vector Diffusion Convergence using Krylov Subspace Update (Harmonic-Free, $\alpha = 0.0001$)

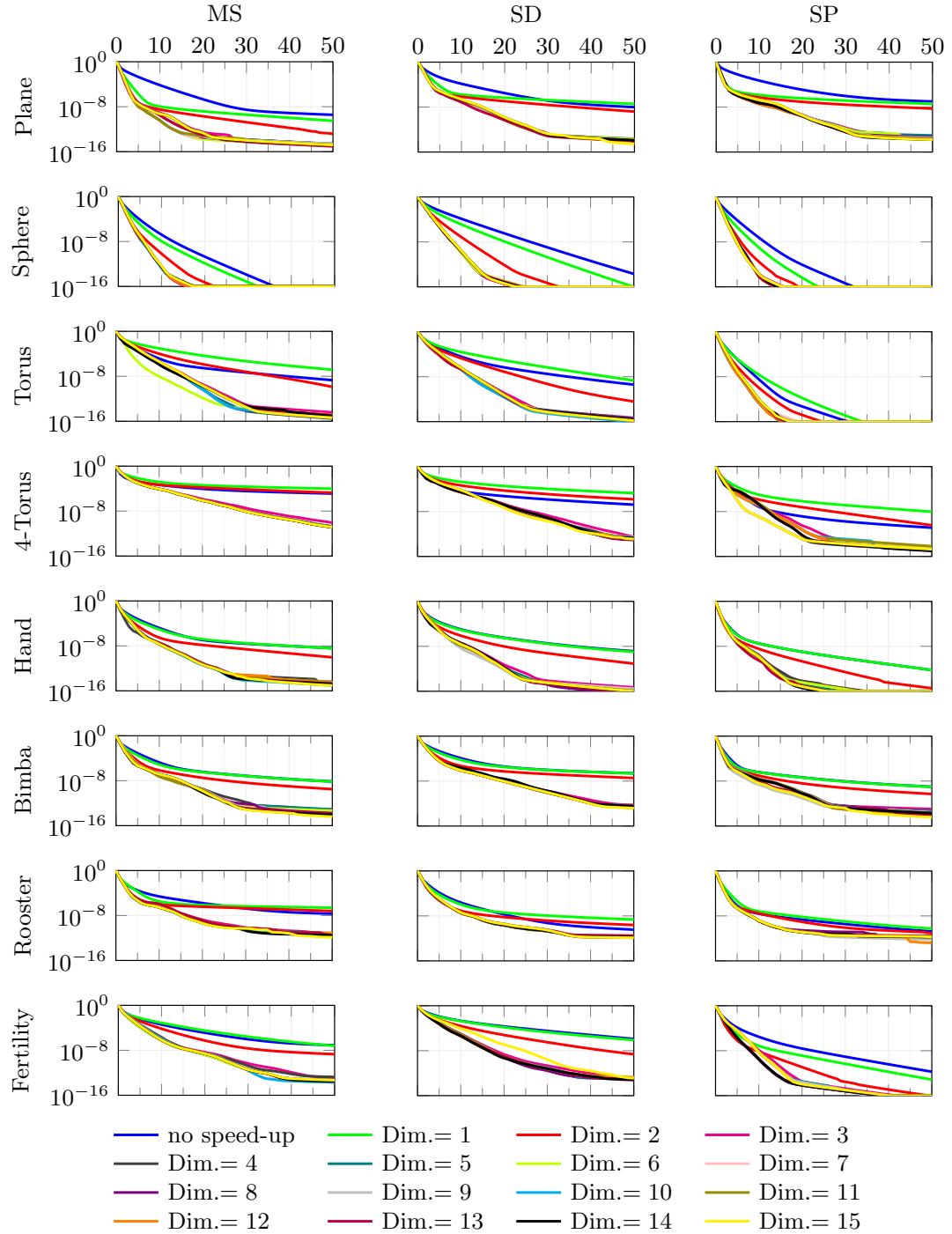


Figure C.15: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.0001$ with various Krylov subspace dimensions.

APPENDIX C. PARAMETER STUDY

Vector Diffusion Convergence using Krylov Subspace Update (Harmonic-Free, $\alpha = 0.001$)

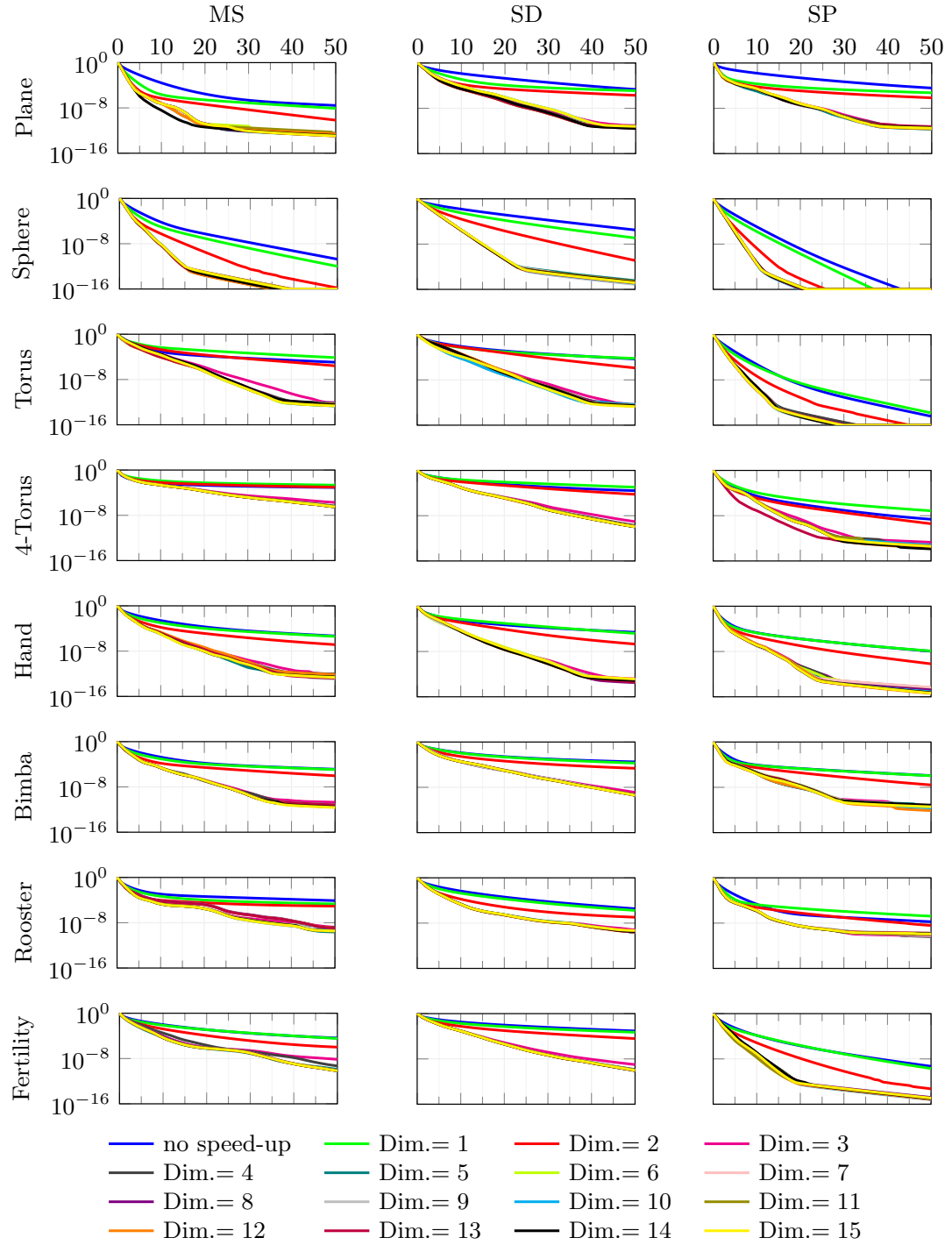


Figure C.16: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.001$ with various Krylov subspace dimensions.

APPENDIX C. PARAMETER STUDY

Vector Diffusion Convergence using Krylov Subspace Update (Harmonic-Free, $\alpha = 0.01$)

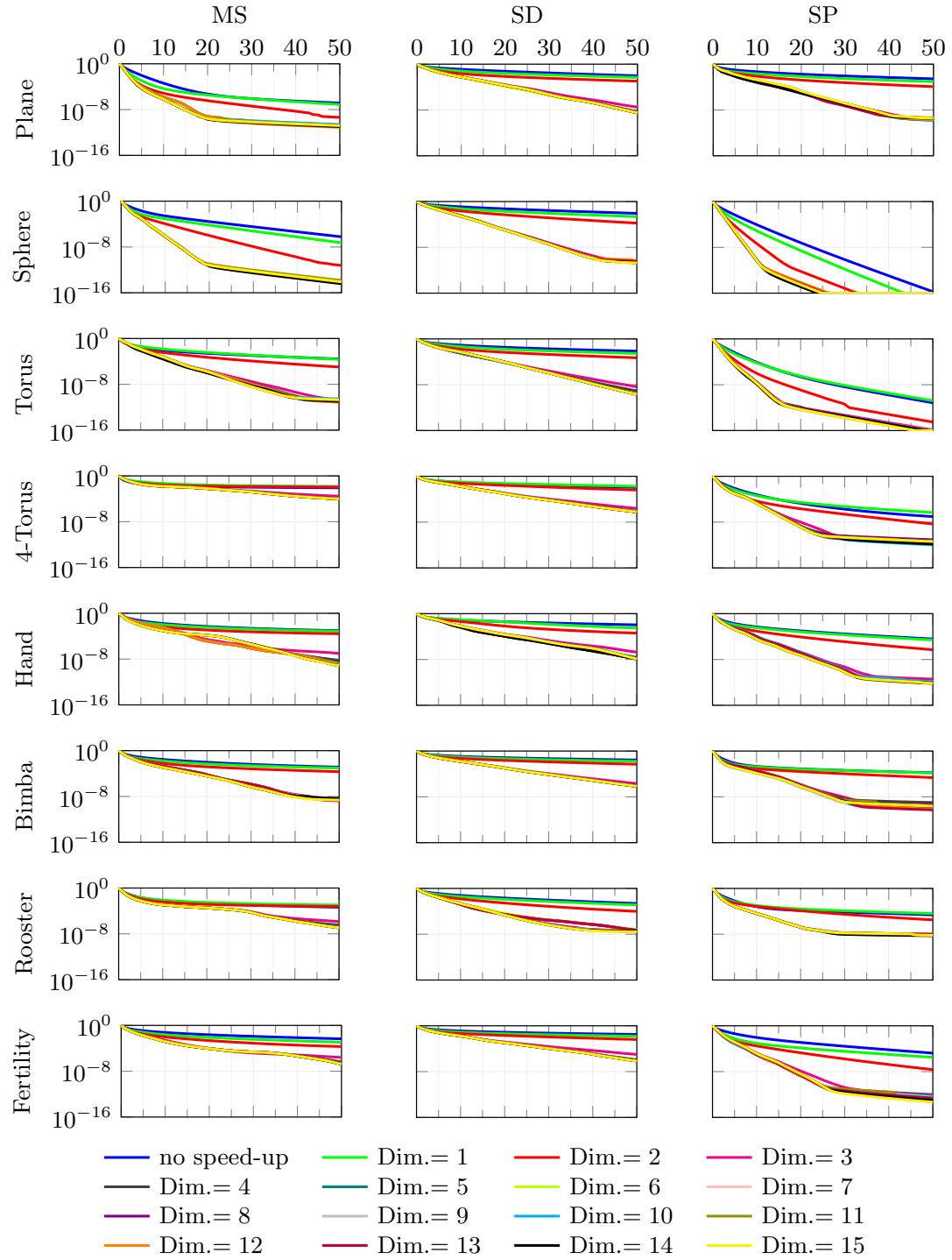


Figure C.17: Convergence of solving a vector diffusion problem using the harmonic-free basis and timestep $\alpha = 0.01$ with various Krylov subspace dimensions.

APPENDIX C. PARAMETER STUDY

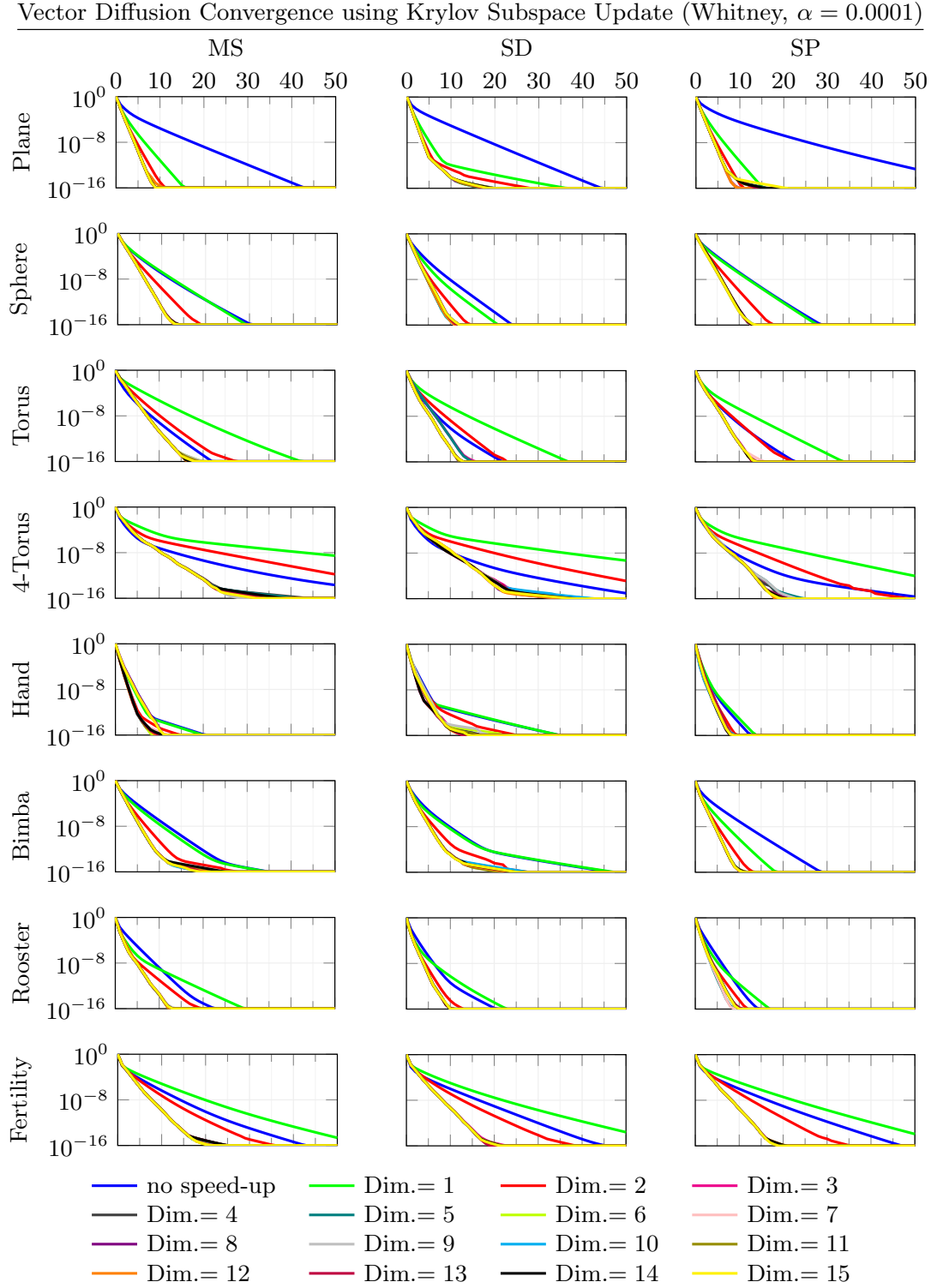


Figure C.18: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.0001$ with various Krylov subspace dimensions.

APPENDIX C. PARAMETER STUDY

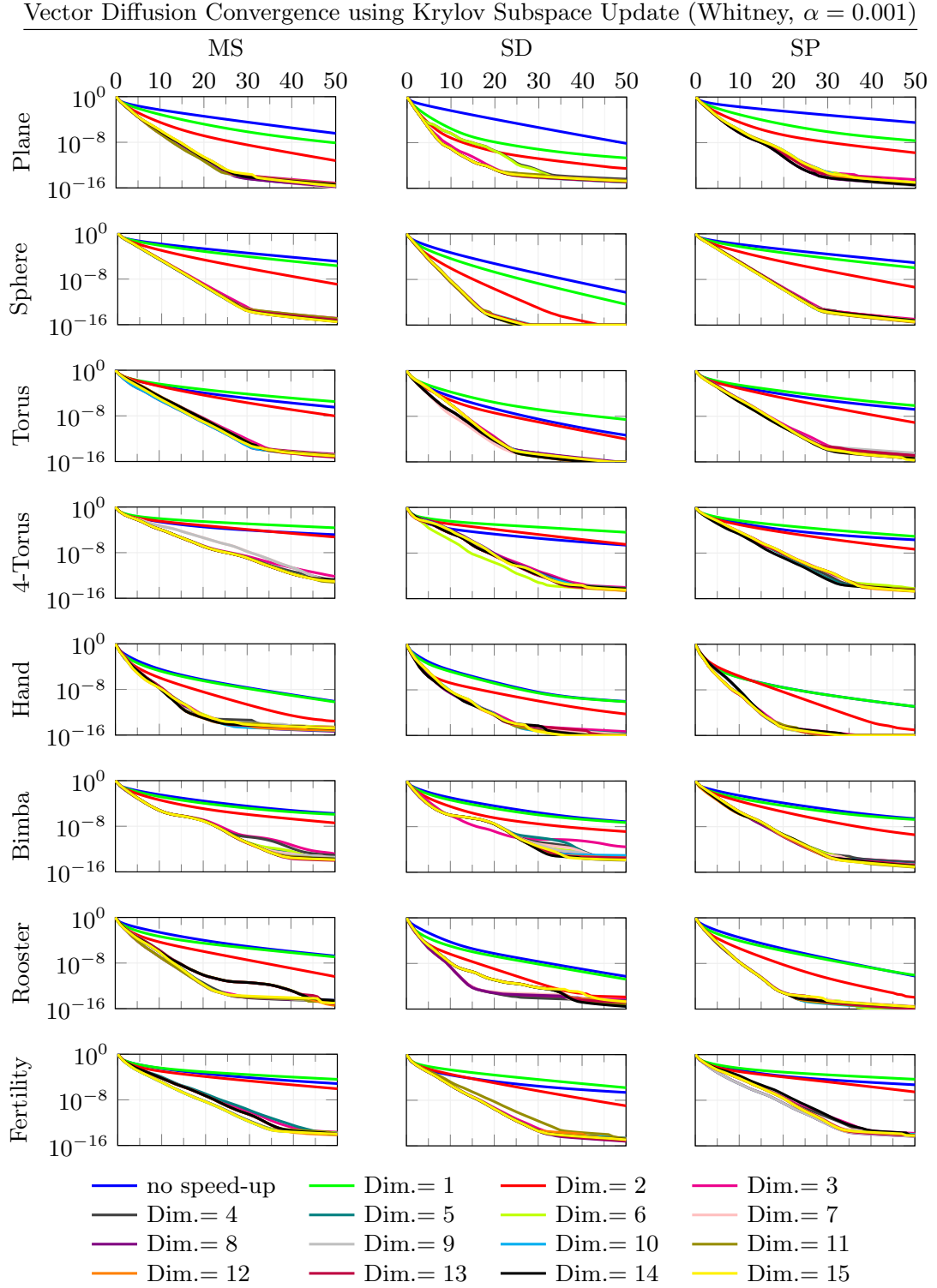


Figure C.19: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.001$ with various Krylov subspace dimensions.

APPENDIX C. PARAMETER STUDY

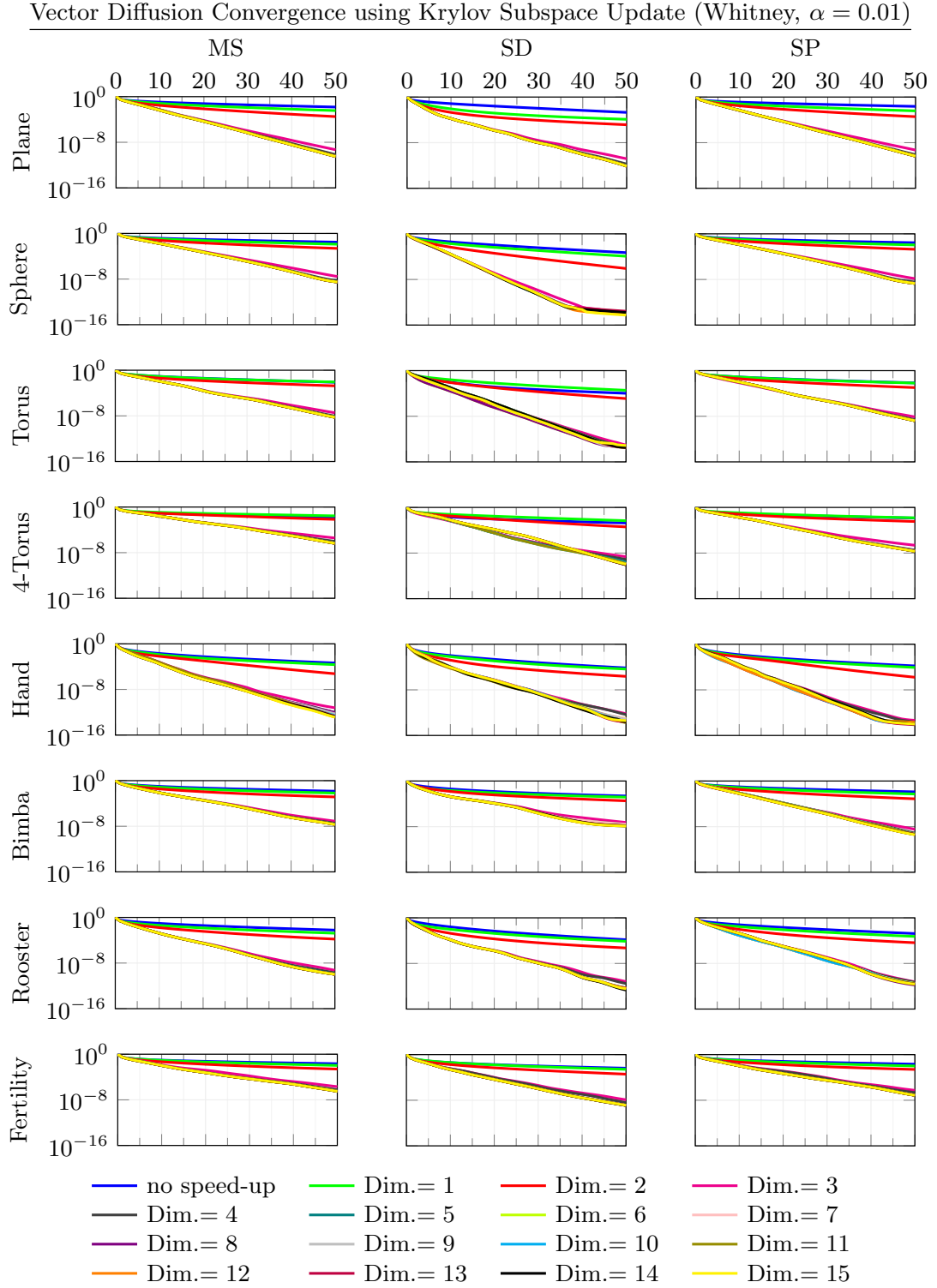


Figure C.20: Convergence of solving a vector diffusion problem using the Whitney basis and timestep $\alpha = 0.01$ with various Krylov subspace dimensions.

APPENDIX C. PARAMETER STUDY

	HS	Dim.		$E(\mathbf{x})$ after 50 itrs.		# of itrs $E(\mathbf{x}) < 10^{-8}$	
		HF	WN	HF	WN	HF	WN
Plane	MS	7	10	$3.6 \cdot 10^{-10} \Rightarrow 2.3 \cdot 10^{-15}$	$6.1 \cdot 10^{-19} \Rightarrow 3.9 \cdot 10^{-24}$	27	\Rightarrow 5 18 \Rightarrow 5
	SD	13	10	$9.7 \cdot 10^{-09} \Rightarrow 7.0 \cdot 10^{-15}$	$2.9 \cdot 10^{-18} \Rightarrow 1.0 \cdot 10^{-20}$	50	\Rightarrow 13 20 \Rightarrow 4
	SP	10	10	$1.0 \cdot 10^{-07} \Rightarrow 2.0 \cdot 10^{-14}$	$2.3 \cdot 10^{-13} \Rightarrow 1.6 \cdot 10^{-22}$	> 50	\Rightarrow 13 26 \Rightarrow 5
Sphere	MS	10	10	$4.5 \cdot 10^{-21} \Rightarrow 2.2 \cdot 10^{-24}$	$1.8 \cdot 10^{-24} \Rightarrow 1.2 \cdot 10^{-30}$	13	\Rightarrow 6 13 \Rightarrow 7
	SD	6	10	$2.0 \cdot 10^{-14} \Rightarrow 5.0 \cdot 10^{-22}$	$1.4 \cdot 10^{-30} \Rightarrow 5.6 \cdot 10^{-31}$	27	\Rightarrow 8 10 \Rightarrow 5
	SP	11	10	$1.6 \cdot 10^{-22} \Rightarrow 5.6 \cdot 10^{-24}$	$9.2 \cdot 10^{-26} \Rightarrow 7.1 \cdot 10^{-31}$	12	\Rightarrow 5 13 \Rightarrow 6
Torus	MS	6	10	$2.1 \cdot 10^{-09} \Rightarrow 1.5 \cdot 10^{-15}$	$2.7 \cdot 10^{-27} \Rightarrow 1.7 \cdot 10^{-24}$	40	\Rightarrow 10 9 \Rightarrow 8
	SD	10	10	$3.8 \cdot 10^{-10} \Rightarrow 8.9 \cdot 10^{-17}$	$3.1 \cdot 10^{-26} \Rightarrow 1.2 \cdot 10^{-25}$	35	\Rightarrow 13 8 \Rightarrow 6
	SP	10	10	$1.3 \cdot 10^{-21} \Rightarrow 2.6 \cdot 10^{-24}$	$8.2 \cdot 10^{-27} \Rightarrow 1.5 \cdot 10^{-29}$	10	\Rightarrow 6 8 \Rightarrow 7
4-Torus	MS	10	10	$1.3 \cdot 10^{-05} \Rightarrow 1.5 \cdot 10^{-11}$	$2.1 \cdot 10^{-14} \Rightarrow 1.7 \cdot 10^{-18}$	> 50	\Rightarrow 30 14 \Rightarrow 11
	SD	15	10	$1.6 \cdot 10^{-07} \Rightarrow 1.3 \cdot 10^{-13}$	$9.0 \cdot 10^{-16} \Rightarrow 2.7 \cdot 10^{-17}$	> 50	\Rightarrow 22 11 \Rightarrow 10
	SP	15	5	$1.3 \cdot 10^{-11} \Rightarrow 1.9 \cdot 10^{-15}$	$2.2 \cdot 10^{-16} \Rightarrow 2.7 \cdot 10^{-20}$	14	\Rightarrow 8 9 \Rightarrow 8
Hand	MS	10	12	$3.9 \cdot 10^{-09} \Rightarrow 9.8 \cdot 10^{-16}$	$6.5 \cdot 10^{-22} \Rightarrow 2.4 \cdot 10^{-21}$	40	\Rightarrow 11 5 \Rightarrow 3
	SD	9	12	$1.3 \cdot 10^{-09} \Rightarrow 2.0 \cdot 10^{-16}$	$3.7 \cdot 10^{-19} \Rightarrow 2.5 \cdot 10^{-21}$	37	\Rightarrow 10 5 \Rightarrow 3
	SP	13	10	$5.9 \cdot 10^{-13} \Rightarrow 7.1 \cdot 10^{-20}$	$1.5 \cdot 10^{-26} \Rightarrow 7.4 \cdot 10^{-27}$	14	\Rightarrow 6 5 \Rightarrow 3
Bimba	MS	11	10	$7.1 \cdot 10^{-09} \Rightarrow 9.1 \cdot 10^{-15}$	$9.2 \cdot 10^{-19} \Rightarrow 1.2 \cdot 10^{-20}$	47	\Rightarrow 13 12 \Rightarrow 6
	SD	10	10	$2.4 \cdot 10^{-07} \Rightarrow 3.5 \cdot 10^{-13}$	$4.1 \cdot 10^{-17} \Rightarrow 3.2 \cdot 10^{-19}$	> 50	\Rightarrow 20 10 \Rightarrow 6
	SP	15	10	$9.8 \cdot 10^{-10} \Rightarrow 3.7 \cdot 10^{-15}$	$3.0 \cdot 10^{-22} \Rightarrow 1.9 \cdot 10^{-22}$	30	\Rightarrow 9 12 \Rightarrow 5
Rooster	MS	14	10	$2.1 \cdot 10^{-08} \Rightarrow 3.1 \cdot 10^{-12}$	$9.0 \cdot 10^{-25} \Rightarrow 9.2 \cdot 10^{-28}$	> 50	\Rightarrow 15 9 \Rightarrow 5
	SD	10	10	$3.5 \cdot 10^{-11} \Rightarrow 2.4 \cdot 10^{-12}$	$1.2 \cdot 10^{-21} \Rightarrow 3.8 \cdot 10^{-23}$	22	\Rightarrow 13 7 \Rightarrow 4
	SP	14	10	$1.7 \cdot 10^{-11} \Rightarrow 1.7 \cdot 10^{-12}$	$3.1 \cdot 10^{-30} \Rightarrow 2.0 \cdot 10^{-30}$	13	\Rightarrow 8 7 \Rightarrow 4
Fertility	MS	10	10	$7.9 \cdot 10^{-08} \Rightarrow 2.0 \cdot 10^{-14}$	$3.3 \cdot 10^{-18} \Rightarrow 5.3 \cdot 10^{-21}$	> 50	\Rightarrow 17 15 \Rightarrow 8
	SD	10	10	$1.2 \cdot 10^{-06} \Rightarrow 1.2 \cdot 10^{-13}$	$3.2 \cdot 10^{-17} \Rightarrow 2.2 \cdot 10^{-20}$	> 50	\Rightarrow 17 17 \Rightarrow 9
	SP	8	10	$1.8 \cdot 10^{-12} \Rightarrow 8.2 \cdot 10^{-18}$	$2.4 \cdot 10^{-17} \Rightarrow 2.6 \cdot 10^{-22}$	23	\Rightarrow 8 17 \Rightarrow 9

Table C.2: The convergence improvement using Krylov subspace updates with best subspace dimensions (Dim.) shown in the third and fourth columns for the harmonic-free (HF) and Whitney (WN) bases. The next two columns show the changes in the relative residual errors after fifty iterations before and after the speed up, and the last two columns show the improvement in the number of iterations required to obtain an accuracy below 10^{-8} .

Sing Chun LEE



ABOUT ME

My passion led me to study Mathematics, Biomedical Computing, and Computer Science on three continents. I grew up in Asia, received my master in Europe, and currently finishing my Ph.D. in the USA. Each culture I lived in shaped who I am today. My research interest includes image processing, augmented reality, and geometry processing. I devoted myself largely to teaching during my Ph.D., and the University recognized the effort, earning me Professor Joel Dean Excellence in Teaching Award. I will continue my passion as an Assistant Professor at Bucknell University.

EDUCATION

JOHNS HOPKINS UNIVERSITY	BALTIMORE, MD, USA
<i>Dissertation: Hierarchical Gradient-Domain Vector Field Processing</i>	
Ph.D. Computer Science	2016–2023
M.Sc. Computer Science (GPA: 4.00)	2016–2017
TECHNISCHE UNIVERSITÄT MÜNCHEN	MUNICH, BAVARIA, GERMANY
<i>Thesis: Integration of RGBD Camera and Mobile C-arms – Calibration, Accuracy, and Applications</i>	
M.Sc. Biomedical Computing (GPA: 3.67)	2013–2016
THE CHINESE UNIVERSITY OF HONG KONG	HONG KONG
<i>Final-Year-Project: Image Upsampling via Tight Frames</i>	
B.Eng. Information Engineering (GPA: 3.65)	2006–2010
B.Sc. Mathematics (GPA: 3.51)	2006–2009

TEACHING EXPERIENCE

JOHNS HOPKINS UNIVERSITY	BALTIMORE, MD, USA
<ul style="list-style-type: none">Intermediate Programming (Fall 2020, Spring 2020, Spring 2021, Summer 2021)	Co-instructor
<ul style="list-style-type: none">HEART: Polygon Mesh Processing (Fall 2019, Fall 2021)	Instructor
<ul style="list-style-type: none">HEART: Geometric Computing and Its Applications (Fall 2020)	Instructor
<ul style="list-style-type: none">Application of Augmented Reality (Spring 2023)	Teaching assistant
<ul style="list-style-type: none">Augmented Reality (Spring 2017, 2018, 2019, 2022)	Teaching assistant
<ul style="list-style-type: none">Computer Graphics (Fall 2018, 2022)	Teaching assistant
<ul style="list-style-type: none">Computer Vision (Fall 2021)	Teaching assistant
TECHNISCHE UNIVERSITÄT MÜNCHEN	MUNICH, BAVARIA, GERMANY
<ul style="list-style-type: none">Patterns in Software Engineering (Winter 2013)	Course ssistant

WORK EXPERIENCE

BALGRIST UNIVERSITY HOSPITAL	ZURICH, SWITZERLAND
Visiting Ph.D. Student	2019/06–2019/08
INTUITIVE SURGICAL INC.	SUNNYVALE, CA, USA
Computer Graphics and Visualization Software Engineer Intern	2017/05–2018/01
JOHNS HOPKINS UNIVERSITY	BALTIMORE, MD, USA
Research Intern	2015/04–2015/12

MEDABILITY GMBH/NARVIS LAB Software Engineer	MUNICH, BAVARIA, GERMANY 2014/08–2015/03
HELMHOLTZ ZENTRUM MÜNCHEN Hilfswissenschaftler	MUNICH, BAVARIA, GERMANY 2014/08–2015/03
AZEUS SYSTEM LIMITED System Analyst	HONG KONG 2011/02–2013/07
ASM PACIFIC TECHNOLOGY LIMITED Computer Vision Engineer	HONG KONG 2010/05–2011/01

AWARDS

Professor Joel Dean Excellence in Teaching Award (2022)
 MICCAI workshop AECAI Best Paper Award (2017)
 NASA Space Apps Challenge/Hackathon Global Norminee (2016)

VOLUNTEERS

ExCamp (2021,2022) TReND(2022) Thread (2019,2020) SABES (2016,2017)

SELECTED PUBLICATIONS

- [FLW*19] FINK L., LEE S. C., WU J. Y., LIU X., SONG T., VELIKOVA Y., STAMMINGER M., NAVAB N., UNBERATH M.: Lumipath – towards real-time physically-based rendering on embedded devices. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019* (Cham, 2019), Shen D., Liu T., Peters T. M., Staib L. H., Essert C., Zhou S., Yap P.-T., Khan A., (Eds.), Springer International Publishing, pp. 673–681. doi:10.1007/978-3-030-32254-0_75.
- [KLAK23] KOHLBRENNER M., LEE S. C., ALEXA M., KAZHDAN M.: Poisson Manifold Reconstruction - Beyond Co-dimension One. *Computer Graphics Forum* (2023). doi:10.1111/cgf.14907.
- [LFF*16] LEE S. C., FUERST B., FOTOUHI J., FISCHER M., OSGOOD G., NAVAB N.: Calibration of rgb-d camera and cone-beam ct for 3d intra-operative mixed reality visualization. *International Journal of Computer Assisted Radiology and Surgery* 11, 6 (Jun 2016), 967–975. doi:10.1007/s11548-016-1396-1.
- [LFT*17] LEE S. C., FUERST B., TATENO K., JOHNSON A., FOTOUHI J., OSGOOD G., TOMBARI F., NAVAB N.: Multi-modal imaging, model-based tracking, and mixed reality visualisation for orthopaedic surgery. *Healthcare Technology Letters* 4, 5 (2017), 168–173. doi:10.1049/htl.2017.0066.
- [LK19] LEE S. C., KAZHDAN M.: Dense point-to-point correspondences between genus-zero shapes. *Computer Graphics Forum* 38, 5 (2019), 27–37. doi:10.1111/cgf.13787.
- [LSF*20] LEE S. C., SEIBOLD M., FÜRNSTAHL P., FARSHAD M., NAVAB N.: Pivot calibration concept for sensor attached mobile c-arms. In *Medical Imaging 2020: Image-Guided Procedures, Robotic Interventions, and Modeling* (2020), Fei B., Linte C. A., (Eds.), vol. 11315, International Society for Optics and Photonics, SPIE, p. 1131503. doi:10.1117/12.2547581.
- [WLZ*17] WANG Z., LEE S. C., ZHONG F., NAVARRO-ALARCON D., LIU Y.-H., DEGUET A., KAZANZIDES P., TAYLOR R. H.: Image-based trajectory tracking control of 4-dof laparoscopic instruments using a rotation distinguishing marker. *IEEE Robotics and Automation Letters* 2, 3 (2017), 1586–1592. doi:10.1109/LRA.2017.2676350.

Last updated: July 31, 2023