Weighted ℓ^1 minimization techniques for compressed sensing and their applications

by

Yi Sui

M.Sc., The University of British Columbia, 2015
B.A. with Honors, University of California, Berkeley, 2012

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Spring 2020

Approval

Name:	**Yi Sui**
Degree:	**Doctor of Philosophy (Mathematics)**
Title:	**Weighted ℓ^1 minimization techniques for compressed sensing and their applications**
Examining Committee:	**Chair:** Ralf Wittenberg Associate Professor
	Ben Adcock Senior Supervisor Associate Professor
	Paul Tupper Committee Professor
	Manfred Trummer Internal Examiner Professor
	Mark Iwen External Examiner Associate Professor Department of Mathematics and Department of CMSE Michigan State University
Date Defended:	**April 17, 2020**

Abstract

Compressed sensing (CS) provides effective techniques for the recovery of a sparse vector from a small number of measurements by finding a solution to an underdetermined linear system. In recent years, CS has attracted substantial attention in applied mathematics, computer science and electrical engineering, and it has the potential to improve many applications, such as medical imaging and function approximation. One standard technique for solving the CS problem is ℓ^1 minimization; however, the performance of ℓ^1 minimization might be limited for many practical applications. Hence, in the past few years, there are many investigations into how to modify the ℓ^1 minimization approach so that better performance can be achieved. One such approach is weighted ℓ^1 minimization. In this thesis, we extend the weighted ℓ^1 minimization technique, traditionally used to solve the standard CS problem, to other applications. First, we develop a variance-based joint sparse (VBJS) algorithm based on weighted ℓ^1 minimization to solve the multiple measurement vector (MMV) problem. Unlike the standard $\ell^{2,1}$ minimization method for this problem, the VBJS method is easily parallelizable. Moreover, we observe through various numerical experiments that the VBJS method often uses fewer measurements to reach the same accuracy as the $\ell^{2,1}$ minimization method. Second, we apply weighted ℓ^1 minimization to the high-dimensional function approximation problem, focusing on the case of gradient-augmented measurements. The high-dimensional function approximation problem has many applications, including uncertainty quantification (UQ), where it arises in the task of approximating a quantity of interest (QoI) of a parametric differential equation (DE). For a fixed amount of computational cost, we see in various examples that, with additional gradient information, better approximation results are often achieved compared to non-gradient augmented sampling. Theoretically, we prove that, with the same sample complexity as the case of function samples only, the gradient-augmented problem gives a better error bound in a stronger Sobolev norm as opposed to an L^2 norm. Finally, we use the adjoint sensitivity analysis method to compute the gradient information. As we show, this method computes the gradient samples of the QoI of a parametric DE with around the same computational cost as computing the samples of the QoI itself. We apply this approach to several parametric DE problems, and numerically demonstrate the benefits of incorporating gradient information into the approximation procedure.

Keywords: Compressed sensing; Weighted ℓ^1 minimization; High-dimensional function approximation; Parametric DE; Multiple measurement vector problem

Dedication

To my mother and father

Acknowledgements

First of all, I would like to thank my supervisor Ben Adcock. I thank him for giving me the opportunity to work on my PhD with him. I thank him for introducing me to the interesting field of compressed sensing. His guidance, generous support and caring helped me go through various difficult times during my study. I enjoyed every conversation and discussion I had with him over the past few years. I also very much appreciate the "open door" policy he has. Ben is a good teacher and a good mentor. This thesis would not be possible without his help. I am very honored to be his first PhD student at SFU. I also would like to thank my committee member Paul Tupper, who always asks me lots of good questions and gives me useful feedback during the committee meeting. I appreciate that.

I also owe a thank to everyone in the Adcock's group. They are all good people and I have learned many things from each of them. I especially thank the former postdoc in the group, Simone Brugiapaglia, who is now an assistant professor at Concordia University. Simone and I had many discussions about mathematics and life when he was at SFU. I enjoyed all our discussions and learned many things from those discussions. Next, I would like to thank all fellow graduate students I meet at SFU. I also would like to thank all the staffs at the mathematics department of SFU. Finally, I would like to thank my examining committee members, Ralf Wittenberg, Ben Adcock, Paul Tupper, Manfred Trummer, Mark Iwen and the former chair Razvan Fetecau, for willing to be my examining committee.

Last but not least, I would like to thank my family. I thank my parents for giving me the opportunity to study abroad. I thank my parents for their endless love and continued support. I owe them a lot. I thank Hans Oeri for going through most parts of this long PhD journey with me.

This work is supported by an entrance scholarship from SFU and an NSERC PGSD scholarship.

Table of Contents

List of Tables

List of Figures

List of Notation

Chapter 2: The multiple measurement vector problem

- $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C \in \mathbb{C}^N$ – the set of C signals to recover
- $\boldsymbol{X} \in \mathbb{C}^{N \times C}$ – the matrix contains the signals to recover
- $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_C \in \mathbb{C}^m$ – C measurements vectors
- $\boldsymbol{Y} \in \mathbb{C}^{m \times C}$ – the matrix of measurements vector
- $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ – sampling matrix
- $\boldsymbol{n}_1, \ldots, \boldsymbol{n}_C \in \mathbb{C}^m$ – noise vectors
- $\boldsymbol{N} \in \mathbb{C}^{m \times C}$ – the matrix of noise vectors
- $\boldsymbol{w} = (w_i)_{i=1}^N$ – vector of positive weights
- ϵ – reciprocal weights parameter
- m – number of samples
- s – sparsity
- S – the support set
- $\boldsymbol{v} = (\check{v}_i)_{i=1}^N$ – variance vector
- $\boldsymbol{F} \in \mathbb{C}^{N \times N}$ – discrete Fourier transform (DFT) matrix
- $\boldsymbol{P}_\Omega \in \mathbb{C}^{m \times N}$ – the matrix selects rows of F with indices in Ω
- $\boldsymbol{g}_c \in \mathbb{C}^N$ – sensitivity profile
- $\boldsymbol{G}_c \in \mathbb{C}^{N \times N}$ – sensitivity profile matrix

Chapter 3: The high-dimensional function approximation problem

- y – one-dimensional variable
- $\boldsymbol{y} = (y_1, \ldots, y_d)$ – d-dimensional variable
- $D = (-1, 1)^d$ – domain
- $f(\boldsymbol{y}) : D \to \mathbb{C}$ – function to approximate
- $\boldsymbol{n} = (n_1, \ldots, n_d) \in \mathbb{N}_0^d$ – multi-index
- $\nu(y)$ – probability density function on $(-1, 1)$
- $\nu(\boldsymbol{y}) = \prod_{k=1}^d \nu(y_k)$ – tensor product probability density function on $D = (-1, 1)^d$
- $L^2_\nu(-1, 1)$, $L^2_\nu(D)$ – spaces of square-integrable functions with respect to ν
- $\{\phi_n\}_{n=0}^\infty$ – one-dimensional orthonormal basis of $L^2_\nu(-1, 1)$
- $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n} \in \mathbb{N}_0^d}$ – tensor-product orthonormal basis of $L^2_\nu(D)$, given by $\phi_{\boldsymbol{n}}(\boldsymbol{y}) = \prod_{k=1}^d \phi_{n_k}(y_k)$
- Λ – finite multi-index set

- N – cardinality of Λ
- Δ – finite multi-index set of coefficients with best or quasi-best s-term approximation
- m – number of samples
- $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ – sample points in D
- ∂_k – partial derivative with respect to y_k
- $\mu(y)$ - sampling measure on $(-1, 1)$
- $\mu(\boldsymbol{y}) = \prod_{i=1}^{d} \mu(y_i)$ - tensor product sampling measure on $D = (-1, 1)^d$

Chapter 4: Parametric differential equations

- y – one-dimensional parameter
- $\boldsymbol{y} = (y_1, \ldots, y_d)$ – d-dimensional parameter
- $\bar{y}$ – sample point of the one-dimensional parameter
- $\bar{\boldsymbol{y}}$ – sample point of the d-dimensional parameter
- x – one-dimensional physical variable for the parametric DEs
- $\boldsymbol{x} = (x_1, \ldots, x_n)$ – n-dimensional physical variable for the parametric DEs
- D – the d-dimensional parameter space
- Ω – the n-dimensional physical domain for the parametric DEs
- $\partial\Omega$ – the boundary for the physical domain Ω
- Γ_D – the boundary satisfies non-homogenous Dirichlet condition
- Γ_N – the boundary satisfies homogenous Newmann condition
- u – solution for the parametric DEs
- u_h – finite-dimensional approximated solution of u
- $\bar{u}$ – solution of u at the sample point $\bar{\boldsymbol{y}}$
- v – the test function
- $\mathcal{K}$ – field, either $\mathbb{R}$ or $\mathbb{C}$
- $\mathcal{H}$ – Hilbert space
- $\mathcal{X}$, $\mathcal{V}$, $\mathcal{Z}$ – Banach spaces
- $\mathcal{Y}$ – open subset of $\mathcal{X}$, space of the parameter
- $\mathcal{U}$ – open subset of $\mathcal{V}$, space of the solution u
- $\mathcal{V}'$ – the dual space of $\mathcal{V}$
- $q(\boldsymbol{y}) = Q(\boldsymbol{y}, u(\boldsymbol{y}))$ – the quantity of interest to recover
- $B[\cdot, \cdot]$ – the bilinear mapping
- $\langle \cdot, \cdot \rangle$ – $L^2(\Omega)$ inner product
- $(\cdot, \cdot)$ – the dual paring
- $\mathcal{U}_h$ – finite-dimensional subspace of $\mathcal{U}$
- $\{\varphi_j\}_{j=1}^{N}$ – a basis of $\mathcal{U}_h$
- $\boldsymbol{U}$ – the stiffness matrix
- F, S, T, G – operators
- F^*, S^*, T^*, G^* – the adjoint operators

Chapter 1

Introduction

The task of reconstructing a signal from a given number of measurements appears in signal processing problems. Mathematically, such a reconstruction problem can be understood as recovering a vector $\boldsymbol{x} \in \mathbb{C}^N$ from *measurements* $\boldsymbol{y} \in \mathbb{C}^m$ through a linear system

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ is the *sampling matrix*. Often, we allow the measurements to contain noise. Then, the linear system becomes

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e},$$

where $\boldsymbol{e} \in \mathbb{C}^m$ is the *noise vector*. Basic linear algebra suggests that, in order to recover $\boldsymbol{x}$, at least N measurements are needed. In fact, if $m < N$, the linear system to solve is *underdetermined* and has infinitely many solutions. Thus, we will not be able to recover $\boldsymbol{x}$ for the case $m < N$ without any additional information. About fifteen years ago, Candès, Romberg and Tao [32] and Donoho [53] published two separate works showing that, given that the vector $\boldsymbol{x}$ is sparse, it is possible to recover the vector with far fewer samples than this linear algebra intuition suggests. These two works introduced a new area of research called *compressed sensing* (CS), which provides efficient techniques for recovering sparse signals from a small number of measurements. CS techniques have many applications in various fields, such as medical image processing, sparse approximation, error correction, etc. For more information, see [59, §1.2].

1.1 The standard compressed sensing problem

Before introducing the standard compressed sensing (CS) problem, some notation and definitions are required. We use $[N]$ to denote the set $\{1, 2, \ldots, N\}$. We write $\bar{\Delta}$ for the complement $[N]\backslash\Delta$ of a set $\Delta \subseteq [N]$. Let $\boldsymbol{x} = (x_n)_{n=1}^N \in \mathbb{C}^N$. For $\Delta \subseteq [N]$, we define $\boldsymbol{x}_\Delta \in \mathbb{C}^N$

by

$$(\boldsymbol{x}_\Delta)_n = \begin{cases} x_n, & n \in \Delta \\ 0, & \text{otherwise} \end{cases}.$$

For a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ and a subset $\Delta \subseteq [N]$, we use $\boldsymbol{A}_\Delta$ to denote an $m \times N$ matrix having nth column equal to the nth column of matrix $\boldsymbol{A}$ whenever $n \in \Delta$ and zero otherwise. $|\Delta|$ denotes the cardinality of a set Δ. The *support* of $\boldsymbol{x}$ is defined as

$$\text{supp}(\boldsymbol{x}) = \{n : x_n \neq 0\} \subseteq \{1, \ldots, N\}.$$

Definition 1.1.1. *A vector $\boldsymbol{x} \in \mathbb{C}^N$ is s-sparse for some $1 \leq s \leq N$ if*

$$||\boldsymbol{x}||_0 := |\text{supp}(\boldsymbol{x})| \leq s.$$

In practice, it is rare that the vector to recover is exactly sparse. This motivates the following definition.

Definition 1.1.2. *Let $\boldsymbol{x} \in \mathbb{C}^N$ and $0 < p \leq \infty$. The ℓ^p norm of the best s-term approximation error is*

$$\sigma_s(\boldsymbol{x})_p := \inf\{\|\boldsymbol{x} - \boldsymbol{z}\|_p,\ \boldsymbol{z} \in \mathbb{C}^N \text{ is } s\text{-sparse}\}.$$

Informally, we say the vector $\boldsymbol{x} \in \mathbb{C}^N$ is *compressible* if $\sigma_s(\boldsymbol{x})_p$ decreases quickly in s.

For simplicity, now we consider the case of noiseless measurements and set up the standard CS problem. A standard CS problem consists in reconstructing an s-sparse vector $\boldsymbol{x} \in \mathbb{C}^N$ with a small number of measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{C}^m$ by solving a linear equation

$$\boldsymbol{A}\boldsymbol{z} = \boldsymbol{y}. \tag{1.1.1}$$

Some common choices of the sampling matrix $\boldsymbol{A}$ used in CS are a Bernoulli or Gaussian random matrix, subsampled discrete Fourier Transform (DFT) matrix, etc. Since $\boldsymbol{x}$ is sparse, it is natural for us to seek a solution of (1.1.1) by solving an ℓ^0 minimization problem

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} ||\boldsymbol{z}||_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{y}, \tag{1.1.2}$$

where $\|\boldsymbol{z}\|_0 = |\{n : z_n \neq 0\}|$. In fact, as shown in [59, §2.2], for a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ and an s-sparse vector $\boldsymbol{x} \in \mathbb{C}^N$, the following two properties are equivalent:

(i) The vector $\boldsymbol{x}$ is the unique s-sparse solution of (1.1.1), that is, $\{\boldsymbol{z} \in \mathbb{C}^N : \boldsymbol{A}\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x},\ \|\boldsymbol{z}\|_0 \leq s\} = \{\boldsymbol{x}\}$.

(ii) The vector $\boldsymbol{x}$ can be reconstructed as the unique solution of (1.1.2).

This ensures that the vector $\boldsymbol{x}$ can be recovered by solving the ℓ^0 minimization problem (1.1.2). However, the optimization problem (1.1.2) is non-convex and NP-hard. For more information on NP-hardness of ℓ^0 minimization, see [59, §2.3]. So, in practice, instead of (1.1.2), one often solves its convex relaxation: an ℓ^1 minimization problem. The ℓ^1 minimization problem is defined by

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} ||\boldsymbol{z}||_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{y}, \tag{1.1.3}$$

where $||\boldsymbol{z}||_1 = \sum_{n=1}^{N} |z_n|$. The problem (1.1.3) is also called *basis pursuit* (BP), which is first introduced by Chen, Donoho, and Saunders in [38]. A more general version of (1.1.3), which allows noisy measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, is defined by

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} ||\boldsymbol{z}||_1 \quad \text{subject to} \quad ||\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}||_2 \leq \eta, \tag{1.1.4}$$

for some $\eta \geq 0$. Note that we assume that the noise bound $\|\boldsymbol{e}\|_2 \leq \eta$ holds. This ℓ^1 minimization problem (1.1.4) is called *quadratically constrained basis pursuit* (QCBP).

It is expected that, by seeking a solution of the BP problem, we can recover the sparse vector $\boldsymbol{x}$. Figure 1.1 gives a simple illustration of the fact that the ℓ^1 norm promotes sparsity. However, it is not the case for the solution of the ℓ^2 minimization problem

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} ||\boldsymbol{z}||_2 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{y},$$

as shown on the right of Figure 1.1.

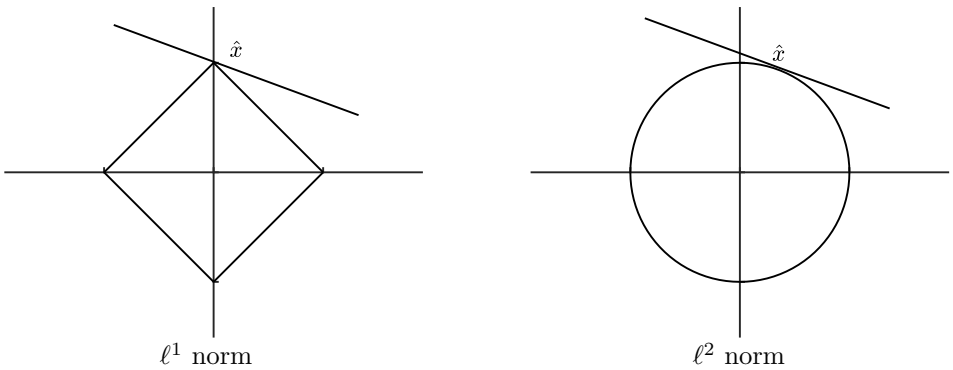


Figure 1.1: Illustration of the sparsity promoting of ℓ^1 norm. The line is the feasible set $\{\boldsymbol{z} : \boldsymbol{A}\boldsymbol{z} = \boldsymbol{y}\}$. The diamond (left) is the ℓ^1 ball and the circle (right) is the ℓ^2 ball. The intersection point $\hat{\boldsymbol{x}}$ is the minimal ℓ^1 norm solution (left) or ℓ^2 norm solution (right). The ℓ^1 norm solution is 1-sparse and the ℓ^2 norm solution is not sparse.

Note that the BP problem is closely related to the least absolute shrinkage and selection operator (LASSO) problem [113], which is often seen in the statistics literature [59, §3.1]. Beyond ℓ^1 minimization, various greedy algorithms can also be used to solve a standard

CS problem. Among them, two of the most common algorithms are: orthogonal matching pursuit (OMP) [116] and compressive sampling matching pursuit (CoSaMP) [100]. For references on other algorithms, see [59, §3].

1.2 The compressed sensing theory

After introducing the standard compressed sensing (CS) problem, now we switch our attention to CS theory, which provides recovery guarantees to ensure a stable and robust recovery of vector $\boldsymbol{x}$. In CS, *stable* recovery means that we can recover a vector $\boldsymbol{x}$ with an error controlled by the distance from $\boldsymbol{x}$ to the set of s-sparse vectors and *robust* recovery means that the distance from the recovered $\hat{\boldsymbol{x}}$ to the original $\boldsymbol{x}$ is controlled by the measurement error η [59, §4.2 & §4.3]. There are two types of recovery guarantees considered in CS, known as *uniform* and *nonuniform* recovery guarantees. Uniform recovery guarantees provide a condition on matrix $\boldsymbol{A}$ which ensures recovery for all vectors $\boldsymbol{x}$. On the other hand, nonuniform recovery guarantees provide conditions on matrix $\boldsymbol{A}$ and vector $\boldsymbol{x}$ which ensure recovery for a fixed vector $\boldsymbol{x}$. Thus, we can see that uniform guarantees are stronger than nonuniform guarantees, and uniform recovery guarantees imply nonuniform recovery guarantees [59, §9.2].

Before stating an example of a uniform recovery guarantee, we first need to define the *robust null space property*.

Definition 1.2.1. *The matrix $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ is said to satisfy the robust null space property (rNSP) of order s with constants $0 < \rho < 1$ and $\tau > 0$ if*

$$\|\boldsymbol{x}_\Delta\|_2 \leq \frac{\rho}{\sqrt{s}}\|\boldsymbol{x}_{\bar{\Delta}}\|_1 + \tau\|\boldsymbol{A}\boldsymbol{x}\|_2$$

for all $\boldsymbol{x} \in \mathbb{C}^N$ and $\Delta \subseteq [N]$ with $|\Delta| \leq s$.

The following theorem shows that the rNSP implies stable and robust recovery.

Theorem 1.2.2. *[7, Thm 5.14] Suppose the matrix $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ satisfies the rNSP of order s with constants $0 < \rho < 1$ and $\tau > 0$. For all $\boldsymbol{x} \in \mathbb{C}^N$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^m$, where $\|\boldsymbol{e}\|_2 \leq \eta$ for some $\eta \geq 0$, any minimizer $\hat{\boldsymbol{x}} \in \mathbb{C}^N$ of (1.1.4) satisfies*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq C_1\sigma_s(\boldsymbol{x})_1 + C_2\sqrt{s}\eta,$$
$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq C_3\frac{\sigma_s(\boldsymbol{x})_1}{\sqrt{s}} + C_4\eta,$$

where the constants C_1, C_2, C_3, C_4 are given by

$$C_1 = 2\left(\frac{1+\rho}{1-\rho}\right),\ C_2 = \frac{4\tau}{1-\rho},\ C_3 = \frac{(3\rho+1)(\rho+1)}{(1-\rho)},\ C_4 = \frac{(3\rho+5)\tau}{(1-\rho)}.$$

It is typically hard to obtain the rNSP of the matrix $\boldsymbol{A}$ directly. Instead, we show that the matrix $\boldsymbol{A}$ satisfies the property called *restricted isometry property* (RIP), which gives a sufficient condition for the rNSP to hold, and then leads to stable and robust recovery of $\boldsymbol{x}$. For details on how the RIP implies the rNSP, see [59, Thm. 6.13].

Definition 1.2.3. *Let $1 \leq s \leq N$. The sth restricted isometry constant (RIC) δ_s of a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ is the smallest $\delta \geq 0$ such that*

$$(1-\delta)\|\boldsymbol{x}\|_2^2 \leq ||\boldsymbol{A}\boldsymbol{x}||_2^2 \leq (1+\delta)\|\boldsymbol{x}\|_2^2,$$

for all s-sparse vectors $\boldsymbol{x} \in \mathbb{C}^N$. If $0 < \delta_s < 1$, then the matrix $\boldsymbol{A}$ is said to have the restricted isometry property (RIP) of order s.

Then, the following theorem gives a uniform guarantee for stable and robust recovery of $\boldsymbol{x}$.

Theorem 1.2.4. *[29, Thm 1.3] Suppose that $\boldsymbol{A} \in \mathbb{C}^{m\times N}$ satisfies the RIP of order 2s with constant*

$$\delta_{2s} < \sqrt{2} - 1.$$

Let $\boldsymbol{x} \in \mathbb{C}^N$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, where $\|\boldsymbol{e}\| \leq \eta$ for some $\eta \geq 0$. Then, a solution $\hat{\boldsymbol{x}} \in \mathbb{C}^N$ of (1.1.4) *satisfies*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq C_1\sigma_s(\boldsymbol{x})_1 + C_2\sqrt{s}\eta,$$
$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq C_3\frac{\sigma_s(\boldsymbol{x})_1}{\sqrt{s}} + C_4\eta,$$

where the constants C_1, C_2, C_3, C_4 depend on δ_{2s} only.

The concept of RIP is closely related to the uniform uncertainty principle (UUP), which is introduced by Candès and Tao in [34]. Later, they refined the UUP to be the RIP in [33], and proved that if the sampling matrix $\boldsymbol{A}$ satisfies $\delta_s + \delta_{2s} + \delta_{3s} \leq 1$, then by solving (1.1.3) we can recover any s-sparse signal $\boldsymbol{x}$ exactly. When there are noisy measurements, Candès, Romberg and Tao showed in [31] that, with $\delta_{3s} + 3\delta_{4s} < 2$, stable and robust recovery of any $\boldsymbol{x}$ can be achieved by solving (1.1.4). An improved condition of $\delta_{2s} < \sqrt{2} - 1$ (shown in the above theorem) was first obtained by Candès in [29].

As mentioned earlier, there are various ways to set the sampling matrix $\boldsymbol{A}$. We now present a uniform recovery guarantee for Gaussian or Bernoulli random matrix as an example.

Definition 1.2.5. *Let $\boldsymbol{A} \in \mathbb{R}^{m\times N}$ be a matrix.*

- *$\boldsymbol{A}$ is a Gaussian random matrix if its entries are independent standard Gaussian random variables.*

- $\boldsymbol{A}$ *is a Bernoulli random matrix if its entries are independent Rademacher random variables, i.e. taking values* ± 1 *with equal probability.*

Theorem 1.2.6. *[7, Thm 5.19] Let* $0 < \epsilon < 1$, $1 \leq s \leq N$ *and*

$$m \gtrsim s \cdot \log(eN/s) + \log(2\epsilon^{-1}). \tag{1.2.1}$$

Suppose that $\boldsymbol{A} = \frac{1}{\sqrt{m}}\tilde{\boldsymbol{A}}$, *where* $\tilde{\boldsymbol{A}} \in \mathbb{C}^{m\times N}$ *is a Gaussian or Bernoulli random matrix. Then the following holds with probability at least* $1-\epsilon$. *For any* $\boldsymbol{x} \in \mathbb{C}^N$ *and* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} \in \mathbb{C}^m$, *where* $\|\boldsymbol{e}\|_2 \leq \eta$ *for some* $\eta \geq 0$, *all minimizers* $\hat{\boldsymbol{x}} \in \mathbb{C}^N$ *of* (1.1.4) *satisfy*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \lesssim \sigma_s(\boldsymbol{x})_1 + \sqrt{s}\eta,$$
$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \lesssim \frac{\sigma_s(\boldsymbol{x})_1}{\sqrt{s}} + \eta,$$

where $\sigma_s(\boldsymbol{x})_1$ *is as in Definition 1.1.2.*

Note that, here and throughout this thesis, the notation $A \gtrsim B$ or $A \lesssim B$ is used to mean there exists a constant $c > 0$ independent of all parameters such that $A \geq cB$ or $A \leq cB$.

Theorem 1.2.6 should be understood as follows. With a high probability, stable and robust recovery of any $\boldsymbol{x} \in \mathbb{C}^N$ can be achieved via the QCBP problem, when the sample complexity (1.2.1) holds. Note that the sample complexity (1.2.1) depends linearly on s, up to log factors in N/s and the failure probability. Moreover, if $\boldsymbol{x}$ is exactly s-sparse, with noiseless measurements, exact recovery is obtained through BP problem.

Unlike uniform recovery guarantees, which are proved by showing that the matrix $\boldsymbol{A}$ satisfies the RIP, nonuniform recovery guarantees are proved by showing the existence of a *dual vector* (also called *dual certificate*). Note, for a complex number $z \in \mathbb{C}$, we define the sign of z by

$$\operatorname{sign}(z) = \frac{z}{|z|}, \quad z \in \mathbb{C}\backslash\{0\}, \quad \operatorname{sign}(0) = 0.$$

If $\boldsymbol{z} = (z_n)_{n=1}^N \in \mathbb{C}^N$ is a vector, then $\operatorname{sign}(\boldsymbol{z}) = (\operatorname{sign}(z_n))_{n=1}^N \in \mathbb{C}^N$ is the vector of component-wise signs.

The following theorem gives an example of a nonuniform recovery guarantee.

Theorem 1.2.7. *[59, Thm. 4.33] Let* $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N$ *be the columns of* $\boldsymbol{A} \in \mathbb{C}^{m\times N}$, *let* $\boldsymbol{x} \in \mathbb{C}^N$ *with* s *largest absolute entries supported on* Δ, *and let* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$ *with* $\|\boldsymbol{e}\|_2 \leq \eta$. *For* $\delta, \beta, \gamma, \theta, \tau \geq 0$ *with* $\delta < 1$, *assume that*

$$\|\boldsymbol{A}_\Delta^*\boldsymbol{A}_\Delta - I\|_2 \leq \delta, \quad \max_{l\in\bar{\Delta}} \|\boldsymbol{A}_\Delta^*\boldsymbol{a}_l\|_2 \leq \beta,$$

and there exists a vector $\boldsymbol{u} = \boldsymbol{A}^*\boldsymbol{h} \in \mathbb{C}^N$ *with* $\boldsymbol{h} \in \mathbb{C}^m$ *such that,*

$$\|\boldsymbol{u}_\Delta - \operatorname{sign}(\boldsymbol{x}_\Delta)\|_2 \leq \gamma, \quad \left\|\boldsymbol{u}_{\bar{\Delta}}\right\|_\infty \leq \theta, \quad \textit{and} \quad \|\boldsymbol{h}\|_2 \leq \tau\sqrt{s}.$$

If $\rho = \theta + \beta\gamma/(1-\delta) < 1$*, then any minimizer* $\hat{\boldsymbol{x}}$ *of* (1.1.4) *satisfies*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq C_1\sigma_s(\boldsymbol{x})_1 + (C_2 + C_3\sqrt{s})\eta,$$

for some constants $C_1, C_2, C_3 > 0$ *depending only on* $\delta, \beta, \gamma, \theta, \tau$*.*

Note that the vector $\boldsymbol{u}$ constructed here is the so-called *inexact dual vector*. We see that there is an extra factor of $\sqrt{s}$ in the error bound of Theorem 1.2.7 compared to the error bound from Theorem 1.2.4. This is a common difference between a nonuniform recovery guarantee and a uniform recovery guarantee, which happens because weaker conditions than the RIP are imposed on the matrix $\boldsymbol{A}$ [7, §5.4] [59, §4.4].

1.3 Weighted ℓ^1 minimization

As mentioned in §1.1, ℓ^1 minimization is the standard technique for solving the classical compressed sensing (CS) problem. However, it has been seen in many practical applications the performance of the ℓ^1 minimization technique could be poor. Thus, in recent years, there are many works on investigating how to modify the ℓ^1 minimization method so that a better performance could be achieved. One simple way to do this is to replace the ℓ^1 norm with the weighted ℓ^1 norm, where the weights are chosen to incorporate some prior knowledge of the problem being solved. This weighted ℓ^1 minimization technique has gained increasing attention in CS in the past few years. Recovering sparse signals via weighted ℓ^1 minimization with prior known support information has been widely studied. See [23, 35, 60, 78, 97, 121], for instance. Similar to the ℓ^1 minimization, the weighted ℓ^1 minimization problem has the form

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} ||\boldsymbol{z}||_{1,\boldsymbol{w}} \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{y},$$

where $||\boldsymbol{z}||_{1,\boldsymbol{w}} = \sum_{n=1}^N w_n|z_n|$ and $\boldsymbol{w}$ is a positive weight vector. With noisy measurements, the problem becomes

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} ||\boldsymbol{z}||_{1,\boldsymbol{w}} \quad \text{subject to} \quad ||\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}||_2 \leq \eta.$$

The idea of using weighted ℓ^1 minimization in CS was first introduced by Candès, Wakin and Boyd in [35]. In [35], the authors pointed out that, if the sparse structure of the signal is known, one should use small weights to encourage the nonzero entries and large weights to penalize the zero entries, so that a better recovery result can be obtained. Demonstrations

of the benefits of including weights were presented in [35] through a series of numerical experiments. A theoretical analysis of sparse recovery with weighted ℓ^1 minimization was shown in [60]. As proved in [60], with at least 50% of the accurate estimate of the partial support, a weaker RIP, compared to the one shown in [31], is sufficient for stable and robust recovery of $\boldsymbol{x}$ via weighted ℓ^1 minimization. A generalization of [60], which considers arbitrarily many distinct weights, was studied by Needell, Saab and Woolf in [99]. An example of random Gaussian recovery with weighted ℓ^1 minimization was studied in [90]. As shown in Theorem 5 of [90], compared to the standard ℓ^1 minimization, significantly fewer measurements than $m \gtrsim s\log(N/s)$ (as shown in Theorem 1.2.6) are sufficient for obtaining stable and robust recovery of $\boldsymbol{x}$ with Gaussian random matrix via weighted ℓ^1 minimization, especially when an accurate support estimate is known.

1.4 Contributions and outline

In this thesis, we extend the weighted ℓ^1 minimization technique, which is traditionally used to reconstruct a sparse vector in compressed sensing (CS), to other applications.

The main contributions of this thesis are:

- We develop a variance-based joint sparse (VBJS) method for solving the multiple measurements vector (MMV) problem, which is easily parallelizable and more effective compared to other standard methods.

- We apply weighted ℓ^1 minimization to the approximation of high-dimensional functions, with a focus on the case of gradient-augmented measurements. As we show numerically and theoretically, this leads to better approximations over the standard, non-gradient augmented case.

- We apply this approach to approximate quantities of interest (QoIs) of parametric differential equations (DEs). As we show, the gradient samples of these QoIs can be computed cheaply via the adjoint sensitivity analysis method. Numerically, this gradient-augmented approach gives better approximations than those obtained from sampling only the QoI itself.

The outline of this thesis is as follows:

- Chapter 2: In this chapter, we introduced a variance-based joint sparse algorithm for the multiple measurement vector problem, based on weighted ℓ^1 minimization. We illustrate the effectiveness of this new algorithm with a set of synthetic experiments and with applications to one-dimensional signal recovery and parallel Magnetic Resonance Imaging. The work presented in Chapter 2 is based on [6], co-authored with Ben Adcock, Ann Gelb and Guohui Song. Rodrigo Platte helped us run the numerical

experiments shown as Figure 2.3, Table 2.1 and Table 2.2 on a cluster at Arizona State University.

- Chapter 3: In Chapter 3, we apply the weighted ℓ^1 minimization technique to the high-dimensional function approximation problem. In particular, we work on the high-dimensional function approximation problem with gradient-augmented sampling. A set of nonuniform recovery guarantees for this gradient-augmented weighted ℓ^1 minimization problem, along with numerical demonstrations on the benefits of additional gradient sampling, are also presented in Chapter 3. The work presented in Chapter 3 is based on [8], co-authored with Ben Adcock.

- Chapter 4: In Chapter 4, we introduce the adjoint sensitivity analysis method for generating gradient samples. Various examples for approximating quantities of interest of parametric differential equations with gradient-augmented samples via weighted ℓ^1 minimization are also presented in Chapter 4.

- Chapter 5: Conclusions for this thesis and some topics for future work are presented in Chapter 5.

Chapter 2

The multiple measurement vector problem

The multiple measurement vector (MMV) problem arises in many applications such as Magnetoencephalography (MEG) [50], magnetic resonance imaging (MRI) [84, 87], video based face recognition [86], and synthetic aperture radar (SAR) imaging [22,39]. Unlike the standard compressed sensing (CS) problem, which aims to recover one sparse signal, the MMV problem in CS considers the recovery of a set of signals $\boldsymbol{x_1}, \ldots, \boldsymbol{x_C} \in \mathbb{C}^N$ with *joint sparsity*, which is a term introduced by Baron et al. in [15].

Definition 2.0.1. *A collection of vectors $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_C}\}$ is s-joint sparse if each vector is s-sparse and the support sets of those vectors overlap, i.e.*

$$|\mathrm{supp}(\boldsymbol{x}_1) \cup \ldots \cup \mathrm{supp}(\boldsymbol{x}_C)| \leq s,$$

where, for a vector $\boldsymbol{x} = (x_i)_{i=1}^N$, $\mathrm{supp}(\boldsymbol{x}) = \{i : x_i \neq 0\}$.

From Definition 2.0.1, we can see that signals with joint sparsity have similar supports.

Although these signals can be recovered separately with classical CS procedure, e.g. ℓ^1 minimization, we seek to develop an algorithm which can achieve the same level of accuracy as individual recovery with fewer measurements by exploiting the joint sparsity information. Within the past one and a half decades, there have been many works on developing algorithms for the MMV problem with joint sparsity exploitation. Typically, these algorithms are modifications of the algorithms for the standard CS problem. For instance, MMV basic matching pursuit (M-BMP) [50, §IV.A], MMV orthogonal matching pursuit (M-OMP) [50, §IV.B], MMV order recursive matching pursuit (M-ORMP) [50, §IV.C], simultaneous orthogonal matching pursuit (S-OMP) [116] and block MMV (BMMV) [61, §3] are extended versions of the matching pursuit (MP) algorithms for the standard compressed sensing problem. Simultaneous hard thresholding pursuit (SHTP) [58] and simultaneous iterative hard thresholding (S-IHT) [88] for the MMV problem are modified versions of the thresholding-based algorithms. For other algorithms, see [21, 131] and references therein.

Among all these algorithms, the most common approach is to solve an $\ell^{2,1}$ minimization problem:

$$\min_{\boldsymbol{Z}\in\mathbb{C}^{N\times C}} ||\boldsymbol{Z}||_{2,1} \text{ subject to } ||\boldsymbol{AZ}-\boldsymbol{Y}||_F \leq \eta. \tag{2.0.1}$$

Here, if $\boldsymbol{X} = (x_{ic})_{i,c=1}^{N,C} \in \mathbb{C}^{N\times C}$ is a matrix, we define the $\ell^{2,1}$ norm by

$$||\boldsymbol{X}||_{2,1} = \sum_{i=1}^{N}\left(\sum_{c=1}^{C}|x_{ic}|^2\right)^{1/2}.$$

The Frobenious nom of $\boldsymbol{X}$ is defined by

$$\|\boldsymbol{X}\|_F = \|\boldsymbol{X}\|_{2,2} = \left(\sum_{i=1}^{N}\sum_{c=1}^{C}|x_{ic}|^2\right)^{1/2}.$$

The measurement matrix $\boldsymbol{Y}$ is defined the same way as shown in §2.1.1.

This $\ell^{2,1}$ minimization approach for the MMV problem can be seen as an analogue for the popular ℓ^1 minimization procedure for single sparse vector recovery. For references on $\ell^{2,1}$ minimization and its generalization for the MMV problem, see [50, 54, 55, 115, 120, 123, 130]. Particularly, in [54], the authors provided a sufficient condition on robust and efficient recovery for a set of signals with block-sparse structure, which can be considered as a generalization of the MMV problem.

Although the $\ell^{2,1}$ minimization approach provides an effective way to solve the MMV problem, we should not ignore a drawback for this method is that it is difficult to parallelize, since the recovery of those signals is inherently coupled. For instance, one $\ell^{2,1}$ minimization problem requires solving an optimization problem of size $N \times C$, where C is the number of signals and N is the signal length. When the number of signals is large, the time used to solve the $\ell^{2,1}$ minimization problem becomes increasingly long. With this in mind, we aim to develop a new recovery algorithm which is easily parallelizable and can reach the same accuracy as $\ell^{2,1}$ minimization with fewer measurements. In this chapter, we will introduce a variance-based joint sparse algorithm, which is based on the weighted ℓ^1 minimization method for the standard CS problem, to solve the MMV problem. We will demonstrate through various numerical experiments that this variance-based joint sparse algorithm is favourable in terms of both computational time and accuracy compared to the $\ell^{2,1}$ minimization approach.

2.1 The variance-based joint sparse (VBJS) recovery

In this section, we will set up the multiple measurement vector (MMV) problem and introduce the variance-based joint sparse (VBJS) recovery algorithm.

2.1.1 The set-up of the MMV problem

In this chapter, we consider the recovery of a set of $C \geq 1$ signals with joint sparsity, which are denoted as $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C \in \mathbb{C}^N$. Note that, in practice, these signals often are not sparse in nature themselves, but under some orthogonal sparsifying transform, e.g. discrete cosine transform (DCT) or wavelet transform [83]. Whenever necessary, we write

$$\boldsymbol{X} = [\boldsymbol{x}_1 | \cdots | \boldsymbol{x}_C] \in \mathbb{C}^{N \times C},$$

for the corresponding matrix to be recovered. Moreover, we consider measurements of the form

$$\boldsymbol{y}_c = \boldsymbol{A}_c \boldsymbol{x}_c + \boldsymbol{n}_c, \qquad c = 1, \ldots, C, \tag{2.1.1}$$

where $\boldsymbol{n}_1, \ldots, \boldsymbol{n}_C \in \mathbb{C}^m$ are *noise* vectors and $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_C \in \mathbb{C}^{m \times N}$ are taken as the standard *sampling matrices* in compressed sensing (CS), e.g. subsampled discrete Fourier transform (DFT) matrix. In this chapter, we shall assume that there is a known priori noise bound for each signal, i.e.

$$\|\boldsymbol{n}_c\|_2 \leq \eta_c, \quad c = 1, \ldots, C.$$

Moreover, often it will be the case that

$$\boldsymbol{A} = \boldsymbol{A}_1 = \cdots = \boldsymbol{A}_C,$$

although this condition is not necessary for the developments that follow. In this chapter, we only consider this latter case. Thus, now we may rewrite (2.1.1) as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{N},$$

where

$$\boldsymbol{Y} = [\boldsymbol{y}_1 | \cdots | \boldsymbol{y}_C] \in \mathbb{C}^{m \times C}, \qquad \boldsymbol{N} = [\boldsymbol{n}_1 | \cdots | \boldsymbol{n}_C] \in \mathbb{C}^{m \times C}.$$

With this in hand, the objective for us is to recover $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C$ (or equivalently $\boldsymbol{X}$) from the measurements $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_C$ (or $\boldsymbol{Y}$). Here, we consider the undetermined setting where m (the number of measurements) is much smaller than N (the signal dimension).

2.1.2 The variance-based joint sparse (VBJS) recovery algorithm

Before introducing the variance-based joint sparse (VBJS) recovery algorithm for solving (2.1.1), we will make two more assumptions:

1. The supports of the vectors are similar, i.e. $\text{supp}(\boldsymbol{x}_1) \approx \text{supp}(\boldsymbol{x}_2) \approx \cdots \approx \text{supp}(\boldsymbol{x}_C)$. Equivalently, the joint sparsity of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C$ does not greatly exceed the sparsity of each of the individual vectors.

2. The coefficients of the vectors are reasonably distinct. Specifically, the vector $\boldsymbol{v} = (v_i)_{i=1}^N$ of element-wise variances

$$v_i = \frac{1}{C}\sum_{c=1}^{C}(x_{ic})^2 - \left(\frac{1}{C}\sum_{c=1}^{C}x_{ic}\right)^2, \qquad i = 1, \ldots, N,$$

is nonzero, and we have $\operatorname{supp}(\boldsymbol{v}) \approx \bigcup_{c=1}^{C} \operatorname{supp}(\boldsymbol{x}_c)$.

Both assumptions are reasonable in practice. The first assumption is taken from the definition for joint sparsity. As mentioned earlier, in this chapter, we work on recovering a set of signals with joint sparsity. In other words, this first assumption is the starting assumption for the problems to be considered in this chapter. The second assumption gives a necessary requirement to see the benefits of joint sparsity. If all vectors $\boldsymbol{x}_c$, $c = 1, \ldots, C$, are identical, with the same sampling matrix $\boldsymbol{A}$, we expect to get exactly the same measurements vectors $\boldsymbol{y}_c$. There is no additional joint sparsity information conveyed to the MMV problem. We should emphasize that, when either of these assumptions does not hold, the VBJS recovery algorithm will still succeed. However, none of the joint sparsity recovery algorithms expect to achieve the same level of accuracy with fewer measurements than the individual recovery of the signals $\boldsymbol{x}_c$.

The VBJS recovery algorithm is described as follows:

1. Recover the vectors $\boldsymbol{x}_c$, $c = 1, \ldots, C$ separately using the standard ℓ^1 minimization:

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} \|\boldsymbol{z}\|_1 \text{ subject to } \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}_c\|_2 \le \eta_c, \qquad c = 1, \ldots, C,$$

where $\eta_1, \cdots, \eta_c$ are the noise bounds for each signal. The recovered results are denoted by $\check{\boldsymbol{x}}_1, \ldots, \check{\boldsymbol{x}}_C$.

2. Compute the element-wise variance of the vectors $\check{\boldsymbol{x}}_c = (\check{x}_{ic})_{i=1}^N$, $c = 1, \ldots, C$. That is, compute $\boldsymbol{v} = (\check{v}_i)_{i=1}^N$, where

$$\check{v}_i = \frac{1}{C}\sum_{c=1}^{C}(\check{x}_{ic})^2 - \left(\frac{1}{C}\sum_{c=1}^{C}\check{x}_{ic}\right)^2, \qquad i = 1, \ldots, N.$$

3. The two assumptions made above suggest that $\boldsymbol{v}$ should carry information about the shared support of the $\boldsymbol{x}_c$. Specifically, $\check{v}_i$ should be large when the index i belongs to this support, and $\check{v}_i \approx 0$ otherwise. Hence we compute a vector of nonnegative weights $\boldsymbol{w} = (w_i)_{i=1}^N$ based on this information, where $w_i \ge 0$. In particular, we choose small w_i when $\check{v}_i$ is large and large w_i when $\check{v}_i \approx 0$.

4. Solve C weighted ℓ^1 minimization problems to get the final reconstruction of each vector $\boldsymbol{x}_c$:

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} \|\boldsymbol{z}\|_{1,\boldsymbol{w}} \text{ subject to } \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}_c\|_2 \le \eta_c, \qquad c=1,\dots,C,$$

where the weighted $\ell_{1,\boldsymbol{w}}$ norm is defined by

$$\|\boldsymbol{z}\|_{1,\boldsymbol{w}} = \sum_{i=1}^{N} w_i|z_i|.$$

We denote the final reconstructions as $\hat{\boldsymbol{x}}_1,\dots,\hat{\boldsymbol{x}}_c$.

We observe that, comparing with the $\ell^{2,1}$ minimization approach, the VBJS algorithm is easily parallelizable, since the computationally intensive steps (steps 1 and 4) each require to solve C separate (weighted) ℓ^1 minimization problems. Steps 2 and 3 require communications between cores, but are extremely cheap in comparison. Thus, for the problem with a large number of C, we expect to see a significant reduction in computational time when using the VBJS method, compared to using $\ell^{2,1}$ minimization.

In step 2, we compute the element-wise variance vector $\boldsymbol{v} = (\check{v}_i)_{i=1}^N$ so that we can get an estimate of the joint support information of the $\boldsymbol{x}_c$ based on $\boldsymbol{v}$. To be more specific, if $\check{v}_i$ is large, then the index i belongs to the shared support. If $\check{v}_i$ is small, then the index i is not in the support. Then, in step 4, we put large weights at the indices which are not in the support to enforce small entries. Conversely, we put small weights at the indices which are in the support to encourage nonzero entries. The idea of using weighted ℓ^1 minimization with weights chosen based on shared support information in step 4 is inspired by [60].

2.1.3 Different weighting strategies

The key in step 3 of the VBJS algorithm is to compute a vector of weights $\boldsymbol{w}\in\mathbb{R}^N$, which can capture the shared support information of the vectors $\boldsymbol{x}_c$ reasonably well, so that a better recovery of $\boldsymbol{x}_c$ in step 4 than in step 1 can be achieved. Now, in this subsection, we shall describe two different strategies for picking $\boldsymbol{w}$, which will be examined numerically in §2.2.1.

- *Cutoff weights*:

  We get an estimation of the shared support set Γ by setting $i\in\Gamma$ when

  $$\check{v}_i \ge \gamma, \quad \text{for a fixed parameter } \gamma>0. \tag{2.1.2}$$

  Then, we define the weights vector $\boldsymbol{w}$ by

  $$w_i = \begin{cases} 1, & i\notin\Gamma \\ \sigma, & i\in\Gamma \end{cases},$$

  where σ is a fixed small positive number.

Note that since the parameters γ, σ are fixed here, the cutoff weights may not be the best idea when there are various scales in the variance $\boldsymbol{v}$. Thus, we should seek for a different strategy, which defines weights w_i directly based on the variance vector $\boldsymbol{v}$.

- *Reciprocal weights*:

  We fix a tolerance $\epsilon > 0$, and set the weights

  $$w_i = \frac{1}{\check{v}_i + \epsilon}. \tag{2.1.3}$$

  By setting weights this way, we see that if $\check{v}_i$ is small, which suggests that the index i is not in the support set, then w_i is large. Conversely, if $\check{v}_i$ is large, then w_i is small. This reciprocal weighting strategy is inspired by [35].

2.2 Numerical results

We now present several synthetic experiments to illustrate the effectiveness of the VBJS algorithm. For comparative purposes, we also consider (i) separate recovery of the C signals via ℓ^1 minimization (equivalent to step 1 of the VBJS algorithm); (ii) recovery of individual signals via two-step reweighted ℓ^1 minimization (details in §2.2.2); and (iii) joint sparse recovery via $\ell^{2,1}$ minimization (as in (2.0.1)). Our comparison is accomplished by analyzing the performance of each method on randomly generated sets of sparse vectors of a given size N. Specifically, for each fixed m (number of measurements) and s (sparsity), we proceed as follows:

1. Fix a number of trials T. For each trial $t = 1, \ldots, T$:

   (i) Generate a support set $S \subseteq \{1, \ldots, N\}$ uniformly at random with size $|S| = s$.

   (ii) Define vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C$ such that $\text{supp}(\boldsymbol{x}_1) = \ldots = \text{supp}(\boldsymbol{x}_C) = S$. The nonzero entries x_{ic}, $c = 1, \ldots, C$, $i \in S$, are drawn independently from the standard normal distribution.

   (iii) Generate a sampling matrix $\boldsymbol{A}$ and compute measurements $\boldsymbol{y}_c = \boldsymbol{A}\boldsymbol{x}_c$, $c = 1, \ldots, C$.

   (iv) Compute the reconstructions $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_C$ using the desired method (ℓ^1 minimization, two-step reweighted ℓ^1 minimization, VBJS or $\ell^{2,1}$ minimization).

   (v) Compute the normalized error $E_t = \sqrt{\sum_{c=1}^{C} \|\boldsymbol{x}_c - \hat{\boldsymbol{x}}_c\|_2^2 / \sum_{c=1}^{C} \|\boldsymbol{x}_c\|_2^2}$ for each method.

   Finally, average the recovery error E_t over the trials, $E = \frac{1}{T}(E_1 + \ldots + E_T)$.

2. Repeat step 1 for different values of s and m as required.

Note that it is possible to compute other quantities of interest, such as the computational time, in a similar manner. We may also compute the empirical success probability p, defined as the fraction of trials which successfully recover the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C$ within a given tolerance, i.e. $E_t < \text{tol}$ for some fixed tolerance tol.

There are various options for generating the sampling matrix in step (iii). Since it frequently arises in applications, we choose $\boldsymbol{A}$ to be a subsampled DFT matrix. That is, we construct a set $\Omega \subseteq \{1, \ldots, N\}$ of size m uniformly at random and let

$$\boldsymbol{A} = \frac{1}{\sqrt{m}} \boldsymbol{P}_\Omega \boldsymbol{F},$$

where $\boldsymbol{F} \in \mathbb{C}^{N \times N}$ is the DFT matrix and $\boldsymbol{P}_\Omega \in \mathbb{C}^{m \times N}$ is the matrix that selects rows of $\boldsymbol{F}$ corresponding to the indices in Ω. The factor $\frac{1}{\sqrt{m}}$ is a normalization constant, and ensures that $\mathbb{E}(\boldsymbol{A}^* \boldsymbol{A}) = I$. The above procedure also requires a number of parameters. Throughout, we shall choose them as $N = 256$, $T = 20$ and $\text{tol} = 10^{-3}$, which is consistent with similar experiments performed in, for example, [98].

We also require a numerical solver for all four of the optimization problems considered: ℓ^1 minimization, two-step reweighted ℓ^1 minimization, weighted ℓ^1 minimization and $\ell^{2,1}$ minimization. Unless otherwise specified, we use the SPGL1 package [118, 119] with its default parameter values, except for the maximum number of iterations which is set to 10,000. Since the data in this experiment is noiseless, we solve equality-constrained minimization problems (i.e. $\eta = 0$ or $\eta_c = 0$ respectively). All numerical experiments are performed on a MacBook Pro with 2 cores, a 2.9GHz Intel Core i7 processor, and 8BG DDR3 RAM, unless specified.

2.2.1 Comparison with different weighting strategies

Figure 2.1 plots the recovery error and success probability versus the number of measurements m for a fixed sparsity s using the VBJS algorithm with different values of cutoff parameter γ (defined as (2.1.2)). For simplicity, we set the corresponding weights $\boldsymbol{w}$ with $\sigma = \gamma$. In all the plots, the usual transition behavior in success probability is observed as the number of measurements increases. However, the recovery result is sensitive to the value of γ. For all experiments, we see improvements of the phase transitions when γ decreases from 1 to 10^{-1}. But, the phase transitions get worse as the value of γ decreases beyond 10^{-1}.

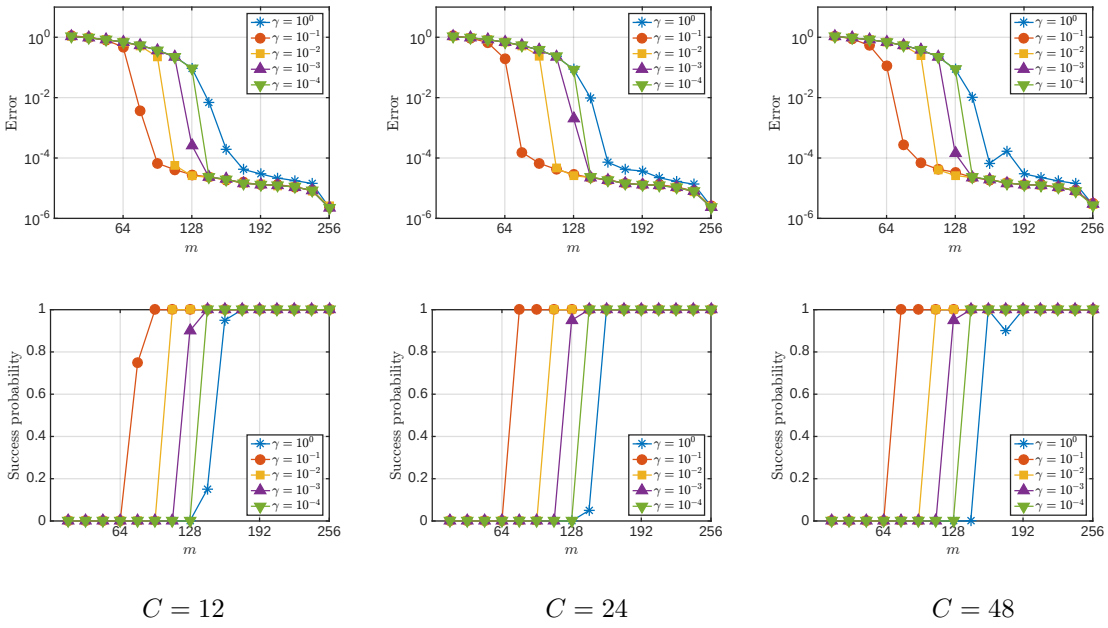


Figure 2.1: Recovery error (top row) and success probability (bottom row) against m for VBJS on randomly-generated sparse vectors with sparsity $s = 64$. The cutoff weights were used with various values of γ.

Figure 2.2 plots the recovery error and success probability versus m for a fixed sparsity s using VBJS with various different values of the weighting parameter ϵ in (2.1.3). As seen from the results with cutoff weights, the usual phase transition behavior is observed as the number of measurements passes through a certain threshold. As expected, larger values of ϵ yield worse phase transitions, since the prior information obtained from the variance vector is less heavily exploited. However, we observe that decreasing ϵ beyond 10^{-2} does not improve the recovery results. This suggests that, in practice, the reciprocal weights is more reliable than the cutoff weights. With this in mind, for remaining experiments in this section, we will use the reciprocal weights with $\epsilon = 10^{-2}$.

2.2.2 Comparison with other methods and solvers

In this subsection, we compare the VBJS method with three other methods:

1. separate recovery of the individual signals via ℓ^1 minimization (i.e. step 1 of the VBJS algorithm),

2. two-step reweighted ℓ^1 minimization of the individual signals (see below),

3. joint sparse recovery via $\ell^{2,1}$ minimization (as in (2.0.1)).

Note that method 2 is similar to the VBJS method, except that the weights do not use any joint information. Instead, following the reweighted ℓ^1 minimization procedure [35], for

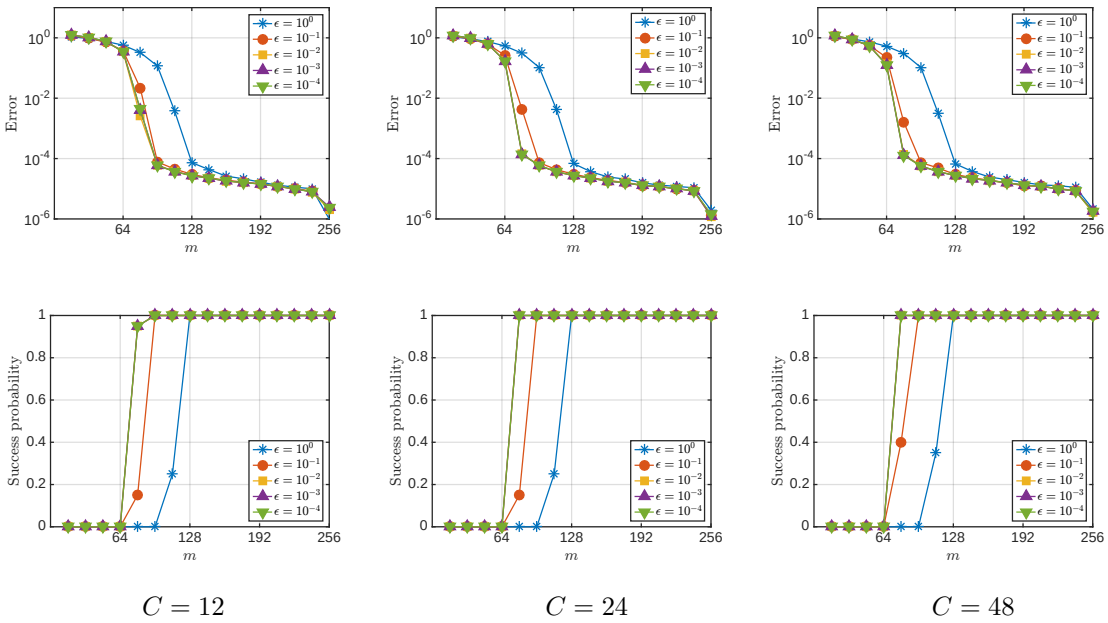


Figure 2.2: Recovery error (top row) and success probability (bottom row) against m for VBJS on randomly-generated sparse vectors with sparsity $s = 64$. The reciprocal weights were used with various values of ϵ.

each signal, the weights $\boldsymbol{w} = \boldsymbol{w}_c = (w_{ic})_{i=1}^{N}$ are chosen according to

$$w_{ic} = \frac{1}{|\check{x}_{ic}| + \epsilon}.$$

For accurate approximations of $|\check{x}_{ic}|$, reweighted ℓ^1 minimization achieves better performance over ℓ^1 minimization for recovery of a single vector. Note that we limit ourselves to two steps for consistency with the VBJS method.

Figure 2.3 reports the recovery error, success probability and average computational time versus m for a fixed sparsity s for each of the four methods. Unsurprisingly, ℓ^1 minimization has the lowest computational time, since it requires only C ℓ^1 minimization solves of size N which are done in parallel, followed by the VBJS ($2C$ solves of size N done in parallel) and then $\ell^{2,1}$ minimization (one solve of size NC). As expected, the two-step reweighted (rw) ℓ^1 minimization uses similar computational time as the VBJS since both methods involve $2C$ solves of size N done in parallel. The VBJS method also achieves the best performance. For instance, with $C = 24$ signals, successful recovery requires only around 80 measurements per signal, in comparison to around 128 for $\ell^{2,1}$ minimization. In other words, the $\ell^{2,1}$ minimization requires over 50% additional measurements to recover the same signals accurately.

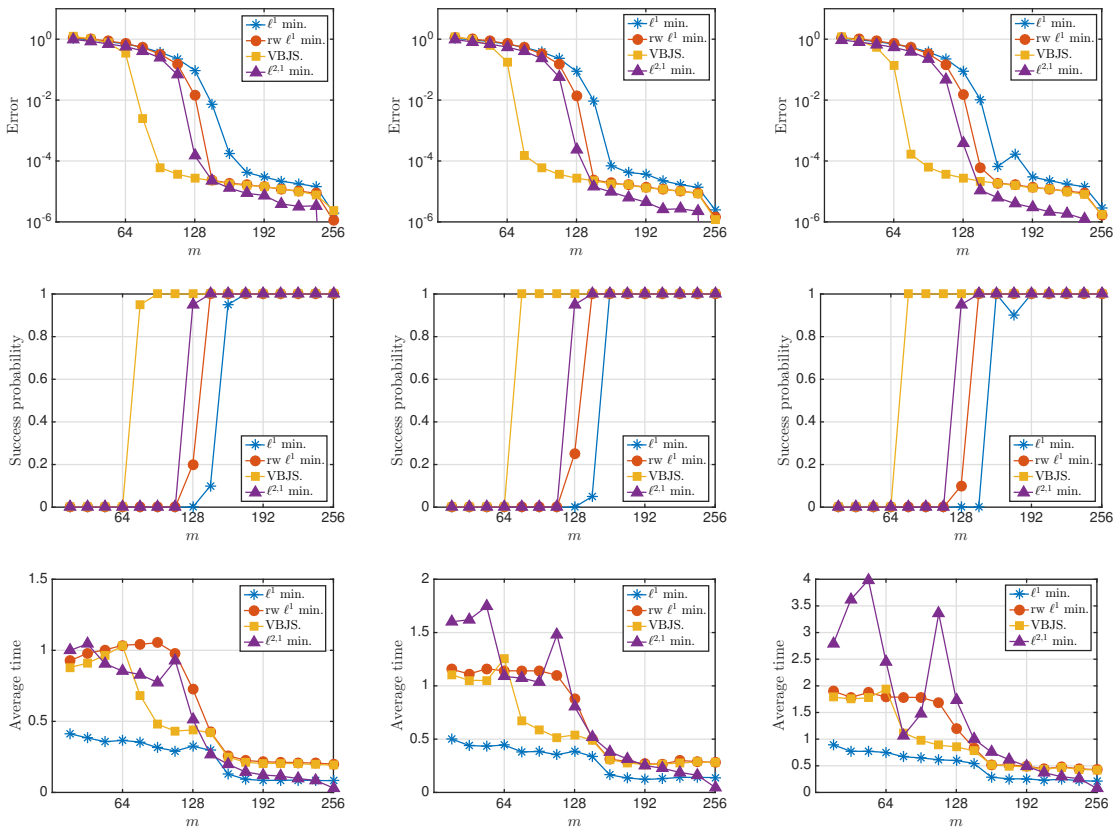


Figure 2.3: Comparison of ℓ^1 minimization, two-step reweighted (rw) ℓ^1 minimization, VBJS and $\ell^{2,1}$ minimization for sparsity $s = 64$. The rows show the error (top), success probability (middle) and average time (bottom) versus m for each method. For this and the results shown in Tables 2.1 and 2.2 computations were performed on a cluster with 48 physical cores (96 logical cores), Intel Xeon E5-4657L v2 processors, 2.90GHz, and 512GB of RAM memory.

The full phase transition plots, which show the success probability v.s. s/N and m/N, and the phase transition curves, which are the curves showing the phase transition from successful recovery to unsuccessful recovery, for each method are shown in Figures 2.4 and 2.5, which give a further illustration of the benefit of the VBJS method. It is worth pointing out that, as shown in both figures, the VBJS method exhibits a phase transition curve which is close to the optimal $m = s$ line. For more information on phase transition, see [52].

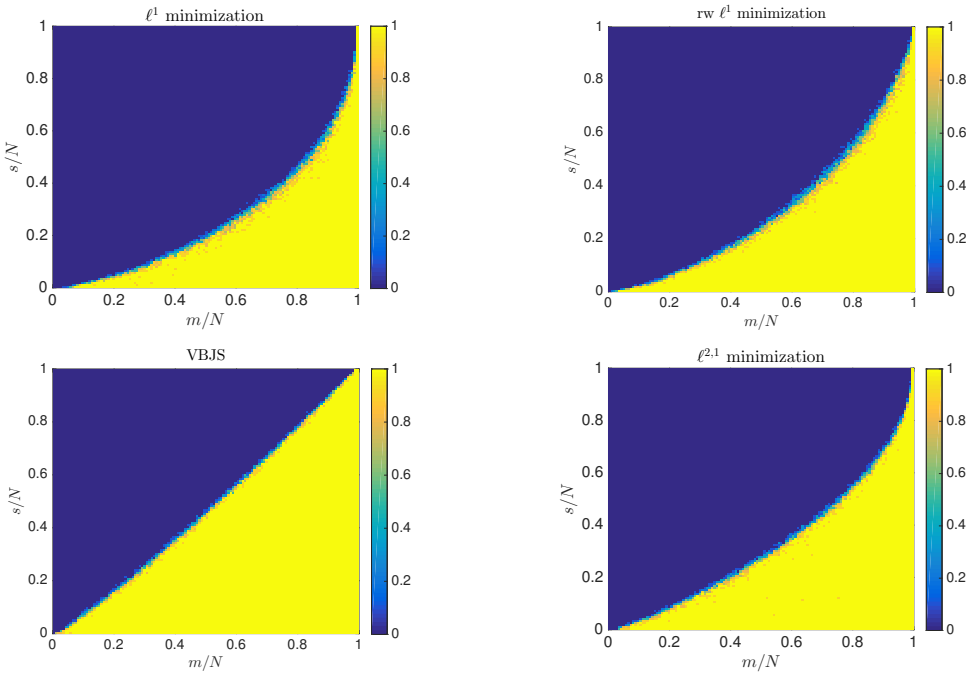
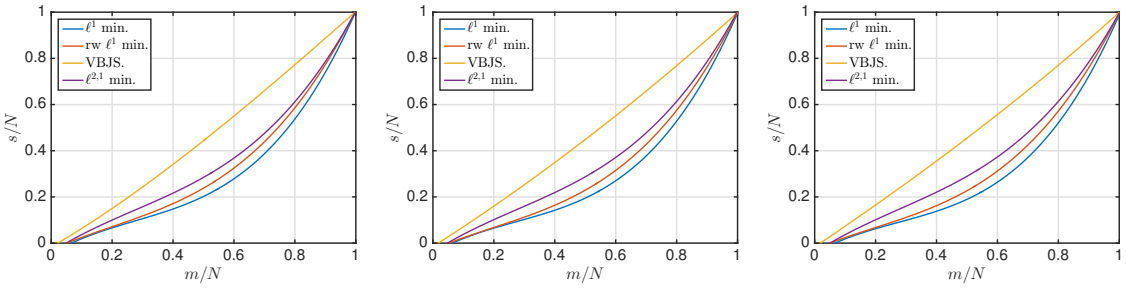


Figure 2.4: Phase transition diagrams for ℓ^1 minimization, rw ℓ^1 minimization, VBJS and $\ell^{2,1}$ minimization for $C = 12$ signals using $T = 10$ trials. The diagrams show the success probability for values $1 \leq s \leq N$ and $1 \leq m \leq N$.



$C = 12$ $C = 24$ $C = 48$

Figure 2.5: Phase transition curves for ℓ^1 minimization, rw ℓ^1 minimization, VBJS and $\ell^{2,1}$ minimization using $T = 10$ trials. The curves show the phase transition from successful recovery (below the line) to unsuccessful recovery (above the line). The criterion for successful recovery used was an empirical success probability $p > 0.75$.

Up to this point, we have used the SPGL1 package to solve all the optimization problems. For completeness, in Figure 2.6, we repeat the experiments of Figure 2.3 using the YALL1 [129] and CVX [65] packages. Similar results are produced by these packages, with VBJS giving a consistently better phase transition than ℓ^1 minimization, rw ℓ^1 minimization and $\ell^{2,1}$ minimization in all cases.

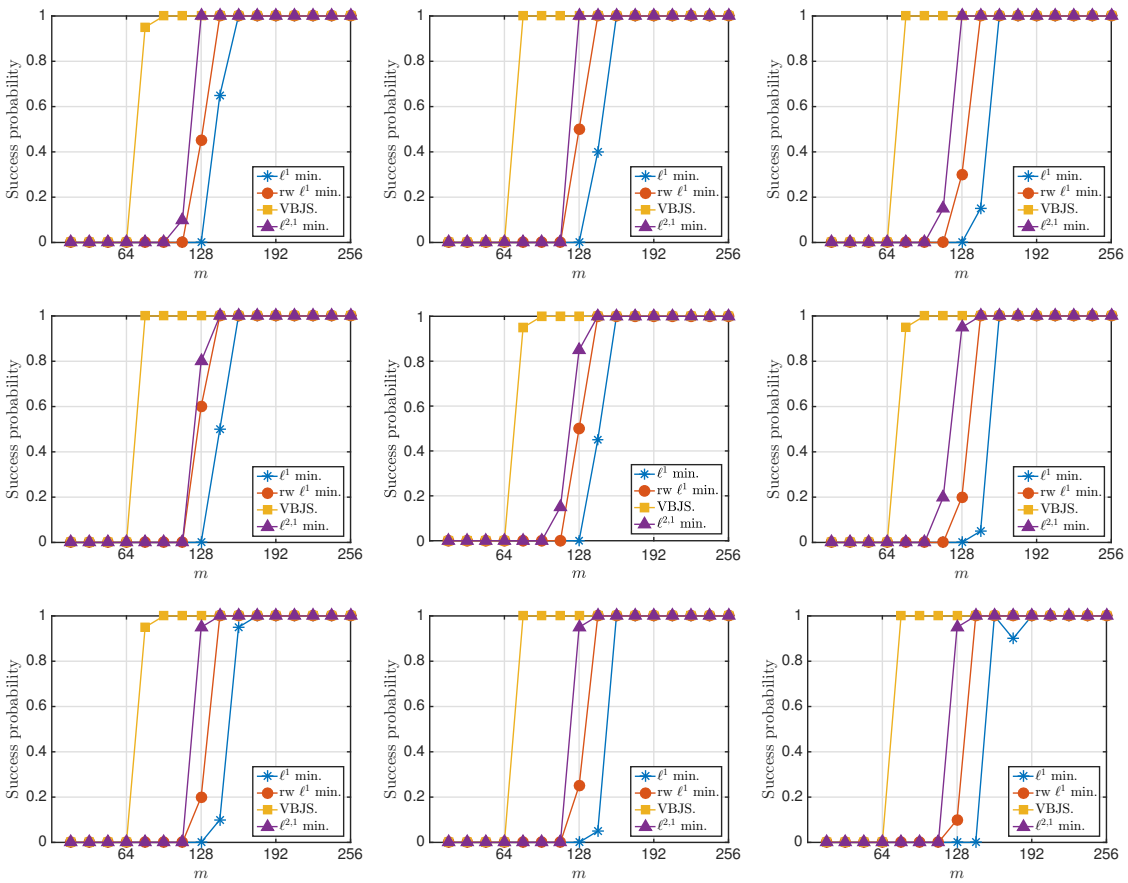


Figure 2.6: Comparison of ℓ^1 minimization, rw ℓ^1 minimization, VBJS and $\ell^{2,1}$ minimization for sparsity $s = 64$ using CVX (top row), YALL1 (middle row) and SPGL1 (bottom row). The plots show the success probability versus m for each method and package.

2.2.3 Signals with partially overlapping supports

Up to now, in all numerical experiments, the signals $\boldsymbol{x}_c$ are assumed to be s-sparse and have a common support S with $|S| = s$. In practice, it may be more realistic to assume that only a fraction of the support is shared. We next present several experiments of this scenario.

To model this set-up, we introduce a parameter $0 < \tau \leq 1$ corresponding to the fraction of shared supports. We then replace steps (i) and (ii) used in the previous experiments with the following:

(i′) Generate a support set $S \subseteq \{1, \ldots, N\}$ uniformly at random with size $|S| = \lfloor \tau s \rfloor$. Generate supports sets $S_1, \ldots, S_C \subseteq \{1, \ldots, N\} \backslash S$ uniformly and independently at random with size $|S_c| = s - \lfloor \tau s \rfloor$ for $c = 1, \ldots, C$.

(ii′) Define vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C$ such that $\mathrm{supp}(\boldsymbol{x}_c) = S \cup S_c$, $c = 1, \ldots, C$. The nonzero entries x_{ic}, $c = 1, \ldots, C$, $i \in S$, are drawn from the standard normal distribution.

In other words, each signal has roughly τs elements in the common support S, and $(1-\tau)s$ elements in a unique support S_c. Note that $\tau = 1$ corresponds to the set-up for the previous experiments.

The top row of Figure 2.7 show the success probability for the VBJS method with various different values of τ using weights as in (2.1.3). As is evident, the performance of the VBJS method quickly declines as τ decreases. However, this is due to the choice of weights, which, while well suited to the $\tau = 1$ case, produce incorrectly scaled weights to effectively handle the partially shared support case. Fortunately, a different weighting strategy can overcome this issue (shown as the bottom row of Figure 2.7). The new weighting strategy, called *energy weights*, consists of five steps shown as follows:

1. Choose $\delta \in (0,1)$ and $\sigma \in (0,1)$.
2. Sort the variance vector $\boldsymbol{v}$ into decreasing order to form a new vector $\boldsymbol{v}^*$. Let I be the index set of sorted indices, i.e. $\boldsymbol{v}_i^* = \boldsymbol{v}_{I(i)}$.
3. Determine the smallest $K \leq N$ such that
$$\sum_{k=1}^{K} \boldsymbol{v}_k^* \geq (1-\delta) \sum_{n=1}^{N} v_n.$$
4. Define the set $T := \{I(1), \cdots, I(K)\}$.
5. Choose weights $w_i = \sigma$ if $i \in T$ and $w_i = 1$ if $i \notin T$.

Unlike the previous weights, the energy weights strategy constructs a candidate support set T that capture all but δ of the variance information, and then uses a binary weighting strategy based on whether an index i is inside or outside T. Note that a similar approach has been used in [60]. This new weighting strategy requires choosing two parameters δ and σ. However, as shown in Figure 2.7, the performance is fairly insensitive to the choice of the parameter σ. Meanwhile, in Figure 2.8, a similar behavior is seen with respect to the other parameter δ. Henceforth, we use the values $\delta = 0.05$ and $\sigma = 10^{-2}$. Finally, we note that it is still possible for T to be non-unique, for example if $\boldsymbol{v}_i = \boldsymbol{v}_j$ for some $i, j \in [1, \ldots, N]$. This is generally unlikely to happen, and most often should only affect the case where there is very small overlap of support (i.e., τ is small).

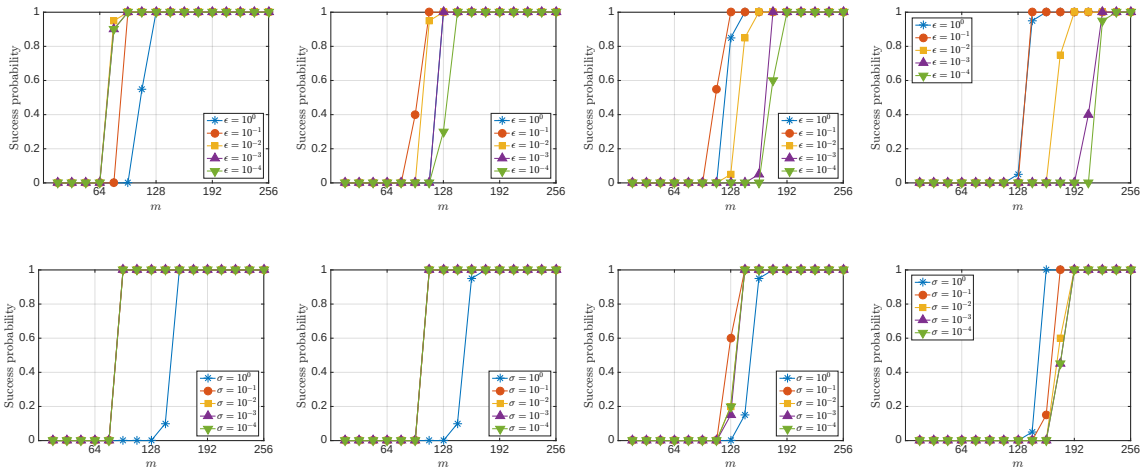


Figure 2.7: Comparison of the success probability for two weighting strategies with various τ. Results for weights as in (2.1.3) are shown in the top, and the energy weights with $\delta = 0.05$ are shown in the bottom.

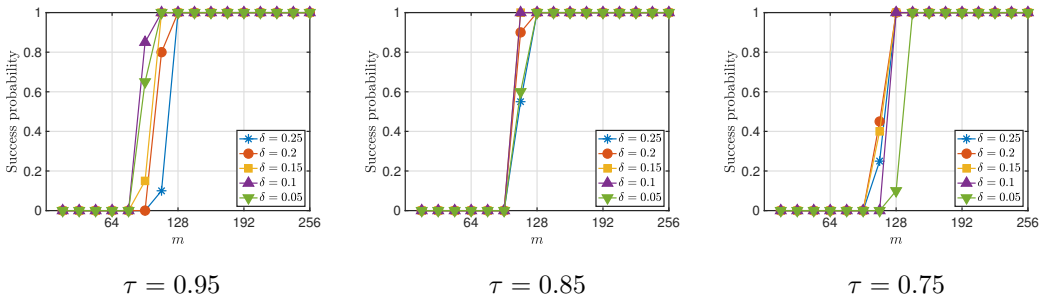
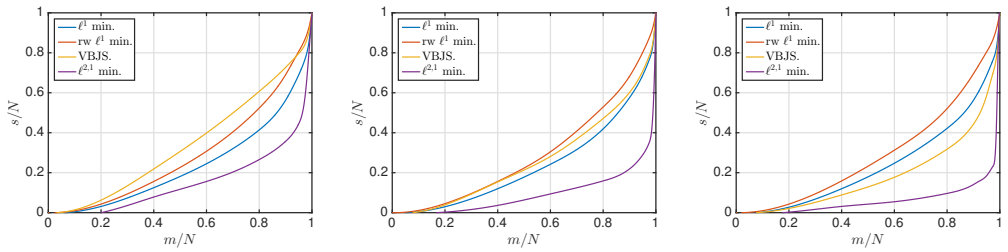


Figure 2.8: Comparison of the success probability with varying δ and $\sigma = 10^{-2}$ for various τ.

Figure 2.9 compares this weighting strategy with the other three methods (defined in §2.2.2) for various different τ. For larger τ, we see the VBJS method offers the best performance. On the other hand, as τ decreases its performance in comparison to ℓ^1 minimization and the two-step reweighted ℓ^1 minimization declines. This is to be expected: as the fraction of shared support decreases, there is less benefit to promoting joint sparsity structure. Interestingly, $\ell^{2,1}$ minimization offers very poor performance, even when τ is close to one.



$\tau = 0.9$ $\tau = 0.75$ $\tau = 0.5$

Figure 2.9: Phase transition curves for ℓ^1 minimization, rw ℓ^1 minimization, VBJS and $\ell^{2,1}$ minimization with $\delta = 0.05$ and $\sigma = 10^{-2}$ for various τ. The criterion for successful recovery is defined the same ways as in Figure 2.5.

2.3 Application to one-dimensional signal recovery

In this section, we consider using the VBJS method to recover a sequence of one-dimensional signals, and we are interested in comparing the recovery results of the VBJS method to $\ell^{2,1}$ minimization. The sequence of one-dimensional signals to recover is defined by

$$f(i,c) = \begin{cases} 2\cos\left(\frac{ic}{CN}\right), & 0 \le i < \frac{N}{\sqrt{8}}, \\ \sin\left(\frac{ic}{CN}\right), & \frac{N}{\sqrt{8}} \le i < \frac{N}{\sqrt{2}}, \\ -\frac{c}{C}, & \frac{N}{\sqrt{2}} \le i \le N. \end{cases}$$

for $i = 1, \ldots, N$ and $c = 1, \ldots, C$. Thus, the (i,c)th position of the signal matrix $\boldsymbol{X}$ is defined by $x_{ic} = f(i,c)$.

Recall, signals are often not sparse in nature, but under some orthogonal sparsifying transform [83]. For this experiment, we sparsify this set of signals with the Haar wavelet transform [89, §7.2.2] by using the Rice wavelet toolbox [14]. This toolbox is also used for those experiments shown in §2.4 and §2.5. Now the ℓ^1 minimization problems to solve in step 1 of the VBJS method become

$$\min_{\boldsymbol{z} \in \mathbb{C}^N} ||\boldsymbol{\Phi z}||_1 \quad \text{subject to} \quad \boldsymbol{A z} = \boldsymbol{y}_c, \qquad c = 1, \ldots, C,$$

where $\boldsymbol{\Phi} \in \mathbb{C}^{N \times N}$ is the Haar wavelet transform basis. The same modifications apply to the weighted ℓ^1 minimization problems to solve in step 4 and the $\ell^{2,1}$ minimization problems. In step 4 of the VBJS method, we use reciprocal weights with $\epsilon = 10^{-2}$. We construct a subsampled DFT matrix $\boldsymbol{A} \in \mathbb{C}^{m \times N}$ by keeping the frequencies

$$\Omega = \Omega_1 \cup \Omega_2,$$

where $\Omega_1 = \{-m/4, \ldots, m/4-1\}$ contains all the lowest $m/2$ frequencies of the DFT matrix and $\Omega_2 \subseteq \{-N/2, \ldots, N/2-1\} \backslash \Omega_1$ contains $m/2$ frequencies chosen uniformly at random.

Figure 2.10 shows the error versus m and the signal-to-error ratio (SER) versus m for $C = 20$ and $N = 256$, when either the VBJS method or $\ell^{2,1}$ minimization is applied. Here, the SER is computed as

$$\text{SER} = -20 \log_{10} \left(\left\| \boldsymbol{X} - \hat{\boldsymbol{X}} \right\|_F / \|\boldsymbol{X}\|_F \right).$$

The experiment is repeated for 10 times and the recovery error is computed the same way as in §2.2. We see that, with small number of measurements, the VBJS method has a clear improvement compared to the standard $\ell^{2,1}$ minimization. The improvement is maximized when $m = 32$. However, as the number of measurements increases, the improvement lessens. For $m \geq 48$, both methods have about the same performance.

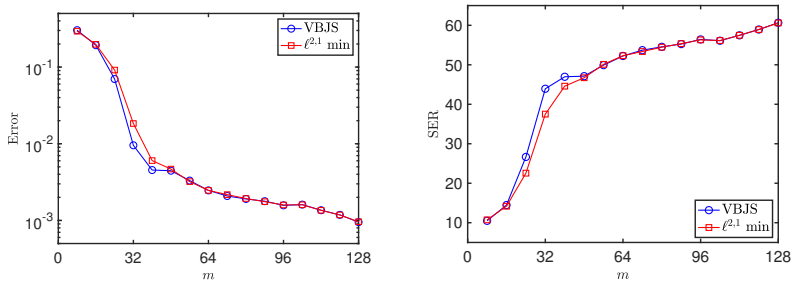


Figure 2.10: Recovery error (left) and SER $= -20 \log_{10} \left(\left\| \boldsymbol{X} - \hat{\boldsymbol{X}} \right\|_F / \|\boldsymbol{X}\|_F \right)$ in dB (right), where $\hat{\boldsymbol{X}}$ is the recovered signal matrix, with VBJS and $\ell^{2,1}$ minimization under Haar wavelet transform.

Snapshots of recovered signals when either the VBJS method or $\ell^{2,1}$ minimization is applied with $m = 32$ are shown in Figure 2.11. It can be clearly seen from those plots that the recovered results with the VBJS method are more accurate than $\ell^{2,1}$ minimization. This result matches with what we have found in Figure 2.10.

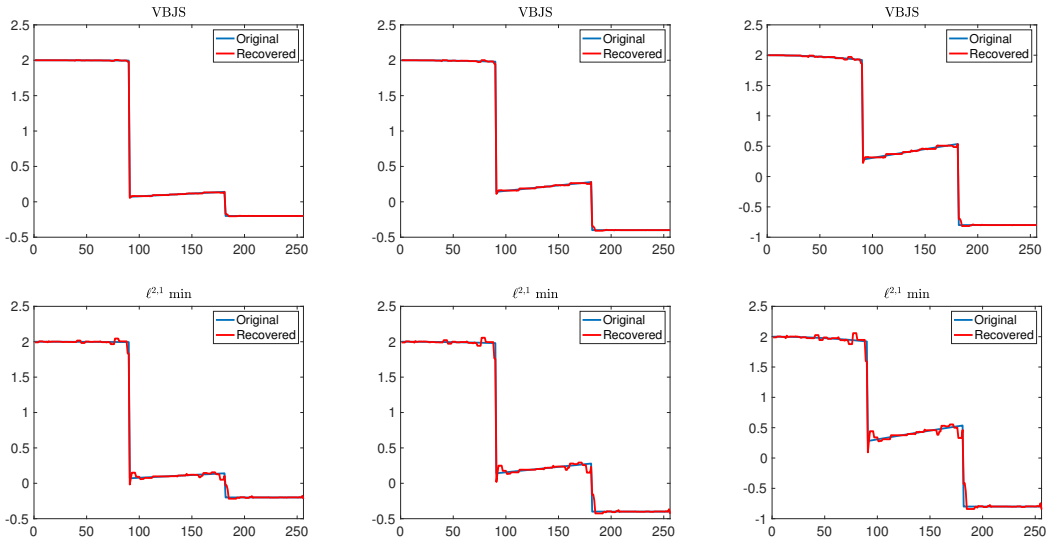


Figure 2.11: Comparison of the recovered results with VBJS and $\ell^{2,1}$ minimization for $c = 4, 8, 16$.

2.4 Application to parallel Magnetic Resonance Imaging

In this section, we consider applying the VBJS method for parallel Magnetic Resonance Imaging (MRI) recovery. Unlike the standard MRI machine, a parallel MRI machine has multiple sensors to acquire data simultaneously which can provide more overall measurements for image recovery. Parallel MRI can be considered as an application of the parallel acquisition model [44]. Note that the gradient-augmented high-dimensional function approximation problem, which will be studied in Chapter 3, can be reformulated as a parallel acquisition model. For details, see §3.8.1. Here, we apply the standard discrete parallel MRI model, which has also been considered in [45,68]. Assume $\boldsymbol{x} \in \mathbb{C}^N$ is the vectorized image to recover. If the parallel MRI machine has in total C coils, then the measurements acquired in cth coil are given by

$$\boldsymbol{y}_c = \boldsymbol{A}\boldsymbol{G}_c\boldsymbol{x} + \boldsymbol{n}_c \in \mathbb{C}^m,$$

where matrix $\boldsymbol{A} = \frac{1}{\sqrt{m}}\boldsymbol{P}_\Omega\boldsymbol{F}$ is a normalized subsampled DFT with size $m \times N$ and $\boldsymbol{n}_c$ is the noise vector. The matrix $\boldsymbol{G}_c = \text{diag}(\boldsymbol{g}_c) \in \mathbb{C}^{N\times N}$ is a diagonal matrix intrinsic to the particular coil, where the vector $\boldsymbol{g}_c \in \mathbb{C}^N$ is known as the *sensitivity profile*. The sensitivity profile is a complex function that usually attenuates the image away from the physical location of the coil. We define the coil images as

$$\boldsymbol{x}_c = \boldsymbol{G}_c\boldsymbol{x},$$

i.e. the overall image $\boldsymbol{x}$ multiplied by the sensitivity profile matrix $\boldsymbol{G}_c$. Then, the measurements acquired in cth coil can also be written as $\boldsymbol{y}_c = \boldsymbol{A}\boldsymbol{x}_c + \boldsymbol{n}_c$. Figure 2.12 gives an example of sensitivity profiles and coil images for $C = 4$. The complex sensitivity profiles are computed using the Biot–Savart law, as described in [68]. Here, we use the MRI Phantom package [62, 68] to generate coil images of the GLPU phantom and the complex sensitivity profiles. Those images are color coded using the domain coloring method for visualizing complex numbers [125, §2.5].

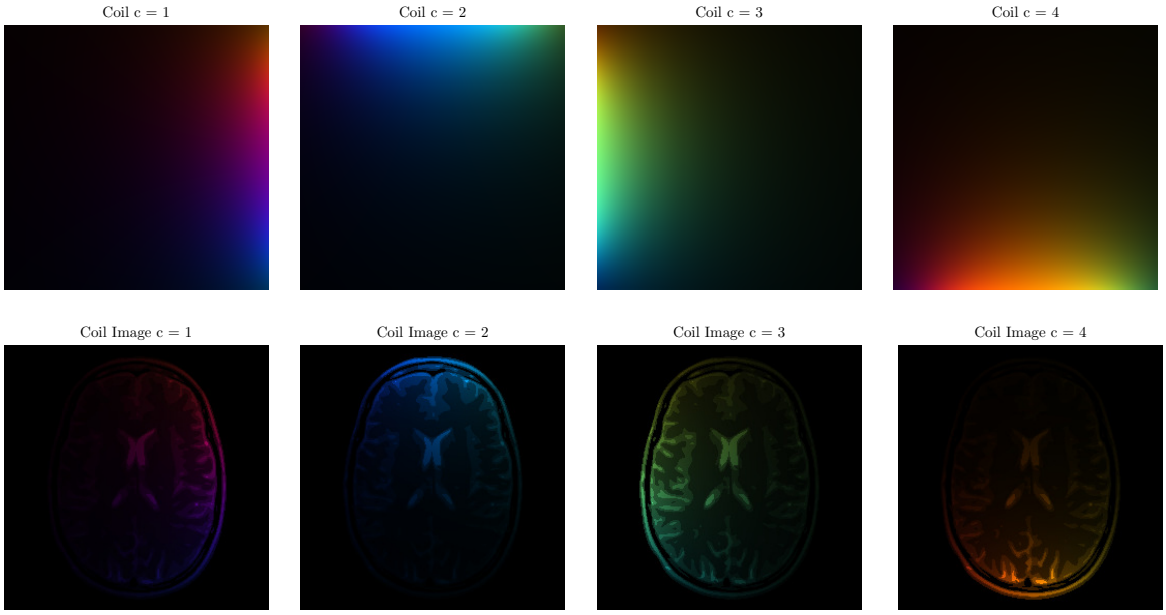


Figure 2.12: Complex sensitivity profiles (top) and coil images (bottom).

There are many methods available for reconstructing parallel MRI images, which can be simply divided into two categories [45]:

(i) *Coil-by-coil* image reconstruction methods, such as GeneRalized Auto-calibrating Partially Parallel Acquisitions (GRAPPA) [67] and iTerative Self-consistent Parallel Imaging Reconstruction (SPIRiT) [82].

(ii) Single image reconstruction methods, such as SENSitivity Encoding SENSE method [103].

Here, we focus primarily on coil-by-coil methods. There are two stages involved in those methods: One first computes approximate coil images $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_C$, and then combines them to get an approximation $\hat{\boldsymbol{x}}$ to the overall image $\boldsymbol{x}$. The first stage can be performed using $\ell^{2,1}$ minimization. The $\ell^{2,1}$ minimization technique is first introduced to the parallel MRI context in [85] and known as the calibration-less multi-coil (CaLM) MR reconstruction method. In this numerical experiment, we compare $\ell^{2,1}$ minimization with the VBJS method.

An advantage of coil-by-coil recovery methods is that they can avoid calibration of the sensitivity profiles $\boldsymbol{G}_c$. Typically, this calibration step requires an additional pre-scan, which can be time-consuming [45]. Moreover, if the sensitivity profiles are not well estimated during the pre-scan, then a poor recovery result of the overall image can be expected [7]. To avoid this, the second stage (recovery of $\boldsymbol{x}$ from the recovered coil images) is performed using a sum-of-squares procedure:

$$\hat{x}_i = \sqrt{\sum_{c=1}^{C} |\hat{x}_{ic}|^2}, \qquad i = 1, \ldots, N. \tag{2.4.1}$$

Unfortunately, this procedure can introduce additional inhomogeneity artifacts to the reconstruction caused by the geometry of the coil. For more details on this, see [45, §VII.B]. Since our main objective in this section is to compare methods for recovering the coil images, we shall avoid the sum-of-squares procedure, and instead use the least-squares fit

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{z} \in \mathbb{C}^N}{\operatorname{argmin}} \sum_{c=1}^{C} \|\boldsymbol{G}_c \boldsymbol{z} - \hat{\boldsymbol{x}}_c\|_2^2. \tag{2.4.2}$$

Note that since matrices $\boldsymbol{G}_c$ are diagonal, the solution of (2.4.2) can be conveniently expressed as

$$\hat{x}_i = \frac{\sum_{c=1}^{C} \hat{x}_{ic} \overline{g_{ic}}}{\sum_{c=1}^{C} |g_{ic}|^2}, \qquad i = 1, \ldots, N,$$

where $\boldsymbol{g}_c = (g_{ic})_{i=1}^N$. This procedure avoids the inhomogeneity artifacts introduced into the sum-of-squares, at the expense of having to know (or pre-compute) the sensitivity profiles.

In the following experiment, we compare the recovery of the analytical phantom image shown in Figure 2.13 (from [68]) using $\ell^{2,1}$ minimization and the VBJS method, followed by the least-squares fit (2.4.2) in both cases. We use the same measurements for each method, taken as radial line sampling in Fourier space (see Figure 2.13). This is a typical sampling procedure for parallel MRI reconstruction. We assume there is Gaussian random noise with variance 10^{-3} added into these measurements. We vary the number of radial lines to taken and the number of coils to see how they change the recovery results. To be more specific, in this experiment, the number of lines is varied from 37 (corresponding to 29.7% sampling of whole Fourier space) to 93 (corresponding to 64.6% sampling), and the number of coils are $C = 8, 16, 32$. As is standard in sparse MRI reconstruction, in both cases, we use the db4 wavelets [89, §7.2.3] as the sparsifying transform. For the VBJS method, we use the reciprocal weights shown as (2.1.3), which we have found to give slightly better reconstruction error than the energy weights introduced in §2.2.3.

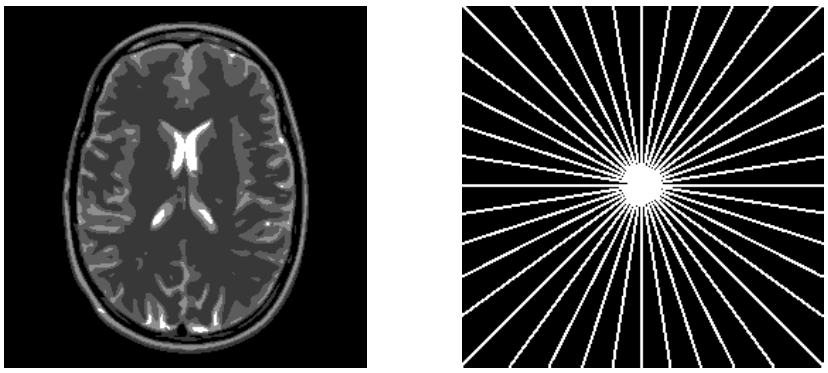


Figure 2.13: 256×256 phantom image (left) and radial sampling map (right).

Tables 2.1 and 2.2 show the signal-to-error ratio (SER) and computational time for both methods. Here, the SER is defines as

$$SER = -20 \log_{10} \left(\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 / \|\boldsymbol{x}\|_2 \right).$$

In the same manner as the experiments shown in §2.2, the VBJS method both requires less time to compute the reconstruction and achieves a consistently higher SER. In particular, for large numbers of coils and radial lines, the time saving is by a factor of between 2 and 4.

$C = 8$

no. of lines	37	45	53	61	69	77	85	93
VBJS SER	21.32	23.54	25.95	28.48	30.93	33.70	36.58	39.32
$\ell^{2,1}$ min SER	20.77	22.93	25.08	27.43	29.66	32.32	35.08	37.73

$C = 16$

no. of lines	37	45	53	61	69	77	85	93
VBJS SER	21.20	23.41	25.73	28.19	30.59	33.31	36.14	38.81
$\ell^{2,1}$ min SER	20.76	22.94	25.08	27.40	29.69	32.38	35.09	37.55

$C = 32$

no. of lines	37	45	53	61	69	77	85	93
VBJS SER	21.20	23.41	25.73	28.19	30.58	33.29	36.16	38.81
$\ell^{2,1}$ min SER	20.75	22.87	25.06	27.45	29.67	32.32	35.09	37.60

Table 2.1: Signal-to-error ratio (SER) $= -20 \log_{10} \left(\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 / \|\boldsymbol{x}\|_2 \right)$ in dB for each method, where $\hat{\boldsymbol{x}}$ is the recovered image and C is the number of coils.

$C = 8$

no. of lines	37	45	53	61	69	77	85	93
VBJS time	113.10	55.96	48.20	42.44	39.87	28.95	32.92	25.64
$\ell^{2,1}$ min time	70.81	83.03	71.38	81.20	34.91	49.47	42.25	39.17

$C = 16$

no. of lines	37	45	53	61	69	77	85	93
VBJS time	56.59	40.79	34.98	35.19	30.53	27.57	27.86	21.38
$\ell^{2,1}$ min time	67.87	142.70	81.64	68.70	82.22	78.63	51.58	60.94

$C = 32$

no. of lines	37	45	53	61	69	77	85	93
VBJS time	67.35	65.32	49.67	46.80	36.21	27.58	29.95	25.92
$\ell^{2,1}$ min time	101.89	103.95	103.99	154.28	83.43	100.08	110.93	77.32

Table 2.2: Computational time (in seconds) for each method, where C is the number of coils.

2.5 Application to color image

Finally, in this section, we apply the VBJS method to color image recovery. Color images have three channels - Red (R), Green (G) and Blue (B), which are highly correlated. It means that, at position i, if there is a large coefficient in the R channel, then it is likely to have a large coefficient at the same location in G and B channels [83]. Thus, we can reformulate a color image recovery problem as a multiple measurement vector (MMV) problem.

In this experiment, we compare the recovery of the colored Shepp-Logan phantom image shown in Figure 2.14 by using the VBJS method, $\ell^{2,1}$ minimization and ℓ^1 minimization (step 1 of the VBJS method). We use the same noiseless measurements for each method with the sampling matrix taken to be a subsampled DFT matrix $\boldsymbol{A} \in \mathbb{C}^{m \times N}$. To form this matrix $\boldsymbol{A}$, we keep the frequencies $\Omega = \Omega_1 \cup \Omega_2$, where Ω_1 contains $m/2$ samples as a circle centered at the zero frequency of the DFT matrix and Ω_2 contains $m/2$ samples uniformly random sampled outside of the circle. An example of the sampling pattern shown in Figure 2.14. Here, we apply the same sparsifying transform and weighting strategy as in §2.3.

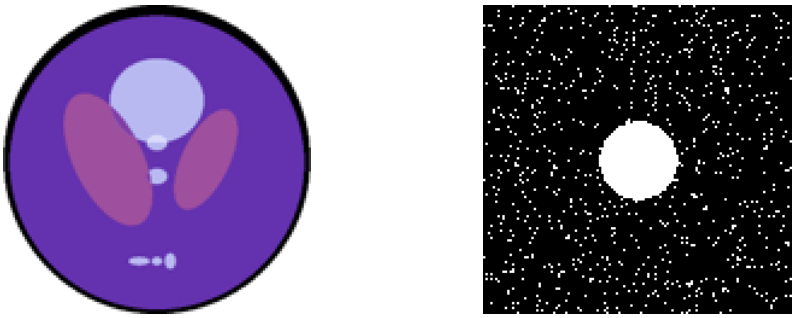


Figure 2.14: 128×128 Shepp-Logan phantom image (left) and sampling pattern (right).

Table 2.3 shows the signal-to-error ratio (SER), defined the same way as in §2.3, for all methods with varying percentages of sampling. As we expected, both the VBJS method and $\ell^{2,1}$ minimization give better recovery results than ℓ^1 minimization. However, surprisingly, the VBJS method always gives less accurate recovery results compared to $\ell^{2,1}$ minimization.

sampling percentage	15%	20%	25%	30%	35%
ℓ_1 min SER	19.71	23.78	27.07	34.42	39.84
VBJS method SER	19.88	24.25	27.78	35.62	41.42
$\ell^{2,1}$ min SER	20.39	24.77	28.52	37.34	43.06

Table 2.3: Signal-to-error ratio (SER) in dB for all methods with various percentages of sampling.

Chapter 3

The high-dimensional function approximation problem

In many science and engineering problems, approximating an analytic and high-dimensional function f from a limited number of measurements is often required. Recall that a function f is analytic if it can be locally expressed as a convergent Taylor series [114, Chpt. 8]. Analytic functions are infinitely differentiable; therefore, extremely smooth. This high-dimensional function approximation problem has gained a lot attention in recent years, driven by the fact that it has various applications including uncertainty quantification (UQ), risk assessment, optimization and control. In UQ, a high-dimensional function approximation problem often arises when computing a quantity of interest (QoI) for a parametric partial differential equation (PDE). In this case, we have a high-dimensional parametric PDE:

$$\mathcal{L}u(\boldsymbol{y};\boldsymbol{x}) = 0,$$

where $\mathcal{L}$ is a differential operator, $\boldsymbol{x}$ is a set of physical variables, and $\boldsymbol{y} \in \mathbb{R}^d$ for $d \gg 1$, is a high-dimensional parameter vector defined on the domain D. Here, the task is to approximate the QoI, $q(\boldsymbol{y}) = Q(u(\boldsymbol{y}, \cdot))$, from a small number of discrete samples. Note that we often have a limited number of samples available due to the fact that each sampling requires a solve of the parametric PDE, which can be very expensive computationally. The operator Q is defined as a linear functional acting on the solution u of the parametric PDE. For example, a typical QoI to compute is the solution u of the parametric PDE at a given point in the physical domain. It can often be shown that the QoI is an analytic and high-dimensional function in the parameter domain [16, 43]. Thus, such a QoI approximation problem can be considered as an analytic and high-dimensional function approximation problem.

As a typical set-up, we assume this analytic, high-dimensional function f to approximate can be expressed as an expansion in an orthogonal multivariate polynomial basis, that is

$$f = \sum_{\boldsymbol{n} \in \mathbb{N}_0^d} c_{\boldsymbol{n}} \phi_{\boldsymbol{n}}.$$

This set-up is often referred to as *polynomial chaos expansion*, which is first introduced by Wiener in [126] and has been used in many subsequent works. See [1, 2, 41], for instance. In order to approximate f, we first draw m samples $f(\boldsymbol{y}_1), \ldots, f(\boldsymbol{y}_m)$ randomly and independently with respect to some probability measure. The goal is to obtain an approximation of f by recovering its s-term expansion coefficients in a finite index set accurately from those samples of f. This approximation problem is often solved by applying the least-square method [40, 47, 48, 71, 94–96, 109]. However, in recent years, there is an increasing focus on applying compressed sensing (CS) techniques, e.g. the ℓ^1 minimization or weighted ℓ^1 minimization technique, to solve this problem [1, 2, 70, 101, 106–108, 112, 128], given that an analytic, high-dimensional function often has a nearly sparse representation in certain orthogonal polynomial basis, e.g. tensor Legendre or Chebyshev polynomials [2, 43]. For more details and discussions related to the convergence rate on function approximation with orthogonal polynomials, see [5] [110, Chpt. 8] [114, Chpt. 8].

In this chapter, we will apply the weighted ℓ^1 minimization technique in compressed sensing to the high-dimensional function approximation problem. In particular, we will work on the gradient-augmented high-dimensional function approximation problem, where both function value f and its gradient value ∇f are sampled. Note that this gradient-augmented high-dimensional approximation problem can be viewed as a multivariate extension of the Hermite interpolation problem from numerical analysis. This gradient-augmented problem comes across in many UQ applications (See [102], for example, and references therein), since it is often the case in practice that gradient samples can be computed with about the same amount of cost as computing the function samples using the adjoint sensitivity analysis method [110, §10.2] [28, §IV]. Thus, for high-dimensional problems, with a fixed amount of computational cost, we expect to obtain substantially more information by considering a gradient-augmented problem compared to the unaugmented problem. The objective for this chapter is to demonstrate that, with additional gradient information, a better recovery result can be obtained. Moreover, we will provide a theoretical justification on how extra gradient information improves the recovery. As a further note, in Chapter 4, we will show in detail how to compute the gradient samples with the adjoint sensitivity analysis method and several examples on recovering quantities of interest of parametric differential equations will also be presented.

3.1 Previous work

Recovery of sparse polynomial chaos expansions with compressed sensing (CS) has been widely studied in recent years. Theoretical guarantees for sparse univariate Legendre polynomial expansions recovery via unweighted ℓ^1 minimization were first presented in [107]. The first work concerned the problem of recovering sparse multivariate trigonometric polynomial expansions through unweighted ℓ^1 minimization was presented in [106]. As an extension of [106] and [107], theoretical results for approximating sparse multivariate trigonometric and Legendre polynomials expansions via weighted ℓ^1 minimization were presented in [108]. For uniform recovery guarantees for high-dimensional function approximation with lower sets assumption via weighted ℓ^1 minimization, see [43]. A related work on nonuniform recovery guarantee for high-dimensional function approximation via weighted ℓ^1 minimization was established in [2].

In recent years the gradient-augmented high-dimensional function approximation problem has gained increasing attention. Tang made a first investigation on using gradient-augmented sampling for sparse Legendre approximation via ℓ^1 regularization in [112]. The first theoretical result for recovery of Hermite polynomial expansions with gradient-augmented unweighted ℓ^1 minimization was provided in [102]. The authors have shown in [102] that, with gradient-augmented sampling, a better null space property and a smaller coherence are obtained, which provide sufficient conditions for a successful recovery. For details on how the null space property implies a stable and robust recovery, we refer to Chapter 1. Related work on gradient-augmented unweighted ℓ^1 recovery of sparse Jacobi polynomial chaos expansions can be found in [69]. For the work on gradient-augmented unweighted ℓ^1 recovery of sparse trigonometric polynomial expansions, see [127].

Finally, for high-dimensional function approximation problem with applications in UQ, see [3, 12, 16, 40, 92, 95, 104]. For gradient-augmented problem with applications in UQ, see [9, 80, 81, 102].

3.2 Notation

In this section, we will define some notation. Throughout this chapter, $\boldsymbol{y} = (y_1, \ldots, y_d) \in D$ denotes the d-dimensional variable, where $D = (-1, 1)^d$ is the d-dimensional domain. For the one-dimensional case, we have $y \in (-1, 1)$. We write $\nu(y)$ for a probability density function on $(-1, 1)$ and $\nu(\boldsymbol{y}) = \prod_{i=1}^{d} \nu(y_i)$ for the corresponding tensor-product probability density function on D. The square-integrable polynomials spaces with respect to ν are denoted by $L^2_\nu(D)$ and $L^2_\nu(-1, 1)$ respectively. We write $\|\cdot\|_{L^2(D)}$ and $\|\cdot\|_{L^2(-1,1)}$ for the corresponding norms, and the inner products are written as $\langle \cdot, \cdot \rangle_{L^2(D)}$ and $\langle \cdot, \cdot \rangle_{L^2(-1,1)}$.

Let $\{\phi_n(y)\}_{n=0}^{\infty}$ be a one-dimensional orthonormal basis of $L^2_\nu(-1,1)$. The corresponding tensor-product orthonormal basis $\{\phi_{\boldsymbol{n}}(\boldsymbol{y})\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ of $L^2_\nu(D)$ is then written as

$$\phi_{\boldsymbol{n}}(\boldsymbol{y}) = \prod_{k=1}^{d} \phi_{n_k}(y_k), \qquad \boldsymbol{n} = (n_1,\dots,n_d) \in \mathbb{N}_0^d,$$

where $\boldsymbol{n} = (n_1,\dots,n_d)$ is a multi-index in $\mathbb{N}_0^d$. Here, the orthonormal bases we are particularly interested in are Legendre and Chebyshev polynomial bases, which are special cases of the ultraspherical and Jacobi families of polynomials.

Jacobi polynomials. For parameters $\alpha,\beta > -1$ and $n \in \mathbb{N}_0$, let $P_n^{(\alpha,\beta)}$ be the Jacobi polynomial of degree n. Jacobi polynomials are orthogonal on $(-1,1)$ with respect to the weight function $\omega^{(\alpha,\beta)}(y) = (1-y)^\alpha(1+y)^\beta$, and satisfy

$$\langle P_n^{(\alpha,\beta)}, P_m^{(\alpha,\beta)}\rangle_{L^2_{\omega^{(\alpha,\beta)}}} = \kappa_n^{(\alpha,\beta)}\delta_{n,m},$$

where

$$\kappa_n^{(\alpha,\beta)} = \frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1}\frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(n+1)\Gamma(n+\alpha+\beta+1)}.$$

These polynomials satisfy a three-term recurrence relation:

$$\begin{aligned}
&P_0^{(\alpha,\beta)}(y) = 1, \qquad P_1^{(\alpha,\beta)}(y) = \frac{1}{2}(\alpha+\beta+2)y + \frac{1}{2}(\alpha-\beta),\\
&2n(n+\alpha+\beta)(2n+\alpha+\beta-2)P_n^{(\alpha,\beta)}(y)\\
&\qquad = (2n+\alpha+\beta-1)\left((2n+\alpha+\beta)(2n+\alpha+\beta-2)y + \alpha^2-\beta^2\right)P_{n-1}^{(\alpha,\beta)}(y)\\
&\qquad - 2(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)P_{n-2}^{(\alpha,\beta)}(y), \quad n = 2,3,4,\dots
\end{aligned}$$

Let $c^{(\alpha,\beta)} = \int_{-1}^{1}\omega^{(\alpha,\beta)}(y)\,\mathrm{d}y$. We define a probability density function $\nu(y) = \frac{\omega^{(\alpha,\beta)}(y)}{c^{(\alpha,\beta)}}$. Then the corresponding orthonormal Jacobi polynomials with respect to $\nu(y)$ are given by

$$\phi_n(y) = \frac{P_n^{(\alpha,\beta)}(y)}{\sqrt{\kappa_n^{(\alpha,\beta)}c^{(\alpha,\beta)}}}, \qquad n \in \mathbb{N}_0.$$

That is,

$$\langle \phi_n, \phi_m\rangle_{L^2_\nu} = \delta_{n,m}, \quad \text{for } n,m \in \mathbb{N}_0.$$

Ultraspherical polynomials. Ultraspherical polynomials belong to the family of Jacobi polynomials, and they are Jacobi polynomials with $\alpha = \beta$. Ultraspherical polynomials are orthogonal on $(-1,1)$ with respect to the weight function $\omega^{(\alpha,\beta)}(y) = (1-y^2)^\alpha$.

Legendre polynomials. Legendre polynomials belong to the family of Jacobi polynomials. They are Jacobi polynomials with $\alpha = \beta = 0$ and are orthogonal on $(-1, 1)$ with respect to the weight function $\omega^{(\alpha,\beta)}(y) = 1$.

Chebyshev polynomials. Chebyshev polynomials belong to the family of Jacobi polynomials. They are Jacobi polynomials with $\alpha = \beta = -\frac{1}{2}$ and are orthogonal on $(-1, 1)$ with respect to the weight function $\omega^{(\alpha,\beta)}(y) = (1 - y^2)^{-1/2}$. Chebyshev polynomials satisfy the general formula

$$P_n^{(-1/2,-1/2)}(y) = \cos(n\Theta), \quad \Theta = \arccos(y), \quad \text{for } n \geq 0.$$

For more information on the family of Jacobi polynomials, see [111, Chpt. IV].

We write the function to approximate as $f : D \to \mathbb{C}$. We approximate f with an expansion in the basis $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$, that is

$$f = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}.$$

Thus, in order to approximate f, we need to reconstruct the vector of coefficients $\boldsymbol{c} = (c_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathbb{N}_0^d} \in \ell^2(\mathbb{N}_0^d)$. However, $\boldsymbol{c}$ is an infinite vector. In order to recover it, we first need to truncate the expansion of f using a finite multi-index set $\Lambda \subset \mathbb{N}_0^d$ with cardinality $N = |\Lambda|$ and consider recovering the coefficient vector in this finite set Λ. We also use Δ to denote a finite multi-index set, typically of size $|\Delta| = s$, which corresponds to the coefficients of f that give the best or quasi-best s-term approximation. Or, more frequently in this chapter, the best or quasi-best s-term approximation in lower sets, which will be defined later.

We consider approximating f with m samples taken at points denoted by $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$. We choose those sampling points randomly and independently with respect to some probability measure. We let $\mu(y)$ denote the sampling probability measure on $(-1, 1)$. The corresponding tensor-product sampling probability measure is then written as $\mu(\boldsymbol{y}) = \prod_{i=1}^d \mu(y_i)$. Note that, typically, but not always, we have $\mu = \nu$.

The ℓ^2 norm and inner product on either $\mathbb{C}^N$ or $\ell^2(\mathbb{N}_0^d)$ are denoted by $\|\cdot\|_2$ and $\langle\cdot,\cdot\rangle$ respectively. We write $\|\cdot\|_{1,\boldsymbol{w}}$ for the norm on the weighted space $\ell^1_{\boldsymbol{w}}(\mathbb{N}_0^d)$ as

$$\|\boldsymbol{c}\|_{1,\boldsymbol{w}} = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} w_{\boldsymbol{n}}|c_{\boldsymbol{n}}|,$$

where $\boldsymbol{w} = (\boldsymbol{w}_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathbb{N}_0^d}$ is an infinite vector of positive weights. For finite vectors of positive weights in $\mathbb{R}^N$, the $\|\cdot\|_{1,\boldsymbol{w}}$ norm is defined likewise.

Finally, we let ∂_k be the partial derivative operator with respect to y_k, i.e. $\partial/\partial y_k$, for $k = 1, \ldots, d$. For $k = 0$, ∂_0 simply denotes the identity operator, i.e. $\partial_0 f = f$.

3.3 Formulation as a weighted ℓ^1 minimization problem

In this section, we will show how to formulate the high-dimensional function approximation problem as a weighted ℓ^1 minimization problem. As defined in §3.2, $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ denotes a tensor-product orthonormal basis of $L^2_\nu(D)$, where ν is a tensor-product probability density function. Then we can write any $f \in L^2_\nu(D)$ as

$$f = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}, \qquad \boldsymbol{c}_n = \langle f, \phi_{\boldsymbol{n}}\rangle_{L^2_\nu(D)}. \tag{3.3.1}$$

The finite vector of coefficients is denoted by $\boldsymbol{c} = (c_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathbb{N}_0^d} \in \ell^2(\mathbb{N}_0^d)$. It has been indicated in §3.2, in order to approximate f, we first need to truncate the expansion (3.3.1) using a finite multi-index set Λ. The truncated expansion is written as

$$f = f_\Lambda + e_\Lambda = \sum_{\boldsymbol{n}\in\Lambda} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}} + \sum_{\boldsymbol{n}\notin\Lambda} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}. \tag{3.3.2}$$

For reasons discussed in §3.4, given $s \geq 1$, we choose Λ as the hyperbolic cross index set of degree s:

$$\Lambda = \Lambda_s^{\mathrm{HC}} = \left\{\boldsymbol{n} \in \mathbb{N}_0^d : \prod_{k=1}^d (n_k + 1) \leq s + 1\right\}. \tag{3.3.3}$$

Let

$$\boldsymbol{n}_1, \ldots, \boldsymbol{n}_N, \tag{3.3.4}$$

be an ordering of the multi-indices in Λ. Then we write $\boldsymbol{c}_\Lambda = (c_{\boldsymbol{n}})_{\boldsymbol{n}\in\Lambda} = (c_{\boldsymbol{n}_j})_{j=1}^N \in \mathbb{C}^N$ for the corresponding finite vector of coefficients. Here and throughout this chapter we shall index over the multi-index set Λ or the index set $\{1, \ldots, N\}$ (using (3.3.4)) interchangeably. The meaning will be clear from the context.

Let μ be another tensor-product probability density function on D. For technical reasons, we assume that

$$\sup_{\boldsymbol{y}\in D} \sqrt{\nu(\boldsymbol{y})/\mu(\boldsymbol{y})}|\phi_{\boldsymbol{n}}(\boldsymbol{y})| < \infty, \qquad \forall \boldsymbol{n} \in \mathbb{N}_0^d. \tag{3.3.5}$$

Note that this condition holds in particular when $\mu = \nu$ and the $\phi_{\boldsymbol{n}}$ are polynomials. Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \in D$ be sample points, drawn independently and randomly according to μ. If

$$\boldsymbol{A} = \frac{1}{\sqrt{m}} \left(\phi_{\boldsymbol{n}_j}(\boldsymbol{y}_i)\right)_{i,j=1}^{m,N} \in \mathbb{C}^{m\times N}, \tag{3.3.6}$$

is the sampling matrix, then we have a system of linear equations

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{c}_\Lambda + \boldsymbol{e}, \tag{3.3.7}$$

where $\boldsymbol{f} \in \mathbb{C}^m$ and $\boldsymbol{e} \in \mathbb{C}^m$ are given by

$$\boldsymbol{f} = \frac{1}{\sqrt{m}} \left(f(\boldsymbol{y}_i)\right)_{i=1}^m, \qquad \boldsymbol{e} = \frac{1}{\sqrt{m}} \left(e_\Lambda(\boldsymbol{y}_i)\right)_{i=1}^m. \tag{3.3.8}$$

Suppose the tail error $\boldsymbol{e}$ satisfies

$$\|\boldsymbol{e}\|_2 \leq \eta, \tag{3.3.9}$$

for some known $\eta \geq 0$. Given a vector of weights $\boldsymbol{w} = (w_{\boldsymbol{n}})_{\boldsymbol{n}\in\Lambda}$ with $w_{\boldsymbol{n}} \geq 1$, $\forall \boldsymbol{n}$, we consider solving the weighted ℓ^1 minimization problem

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} \|\boldsymbol{z}\|_{1,\boldsymbol{w}} \text{ subject to } \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{f}\|_2 \leq \eta. \tag{3.3.10}$$

If $\hat{\boldsymbol{c}} \in \mathbb{C}^N$ is a minimizer of this problem, then the resulting approximation to f is given by

$$\hat{f} = \sum_{\boldsymbol{n}\in\Lambda} \hat{c}_{\boldsymbol{n}} \phi_{\boldsymbol{n}}. \tag{3.3.11}$$

Note that, in practice, a tail error bound such as (3.3.9) may not be available, since the error $\boldsymbol{e}$ depends on the unknown function f. For theoretical results on sparse recovery under unknown errors, see [3, 4, 27].

Before stating a recovery guarantee for the weighted ℓ^1 minimization problem (3.3.10), we need to introduce several additional definitions. First, for a vector of weights $\boldsymbol{w} = (w_{\boldsymbol{n}})$ with $w_{\boldsymbol{n}} > 0$ and a set $\Delta \subset \mathbb{N}_0^d$, the *weighted cardinality* of Δ is defined by

$$|\Delta|_{\boldsymbol{w}} = \sum_{\boldsymbol{n}\in\Delta} w_{\boldsymbol{n}}^2. \tag{3.3.12}$$

Given ν, μ and $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ as in §3.2, we define the *intrinsic weights* $\boldsymbol{u} = (u_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathbb{N}_0^d}$ as

$$u_{\boldsymbol{n}} = \sup_{\boldsymbol{y}\in D} \sqrt{\nu(\boldsymbol{y})/\mu(\boldsymbol{y})} |\phi_{\boldsymbol{n}}(\boldsymbol{y})|. \tag{3.3.13}$$

As mentioned in [2, 4], for tensor Chebyshev polynomials with samples drawn from the Chebyshev measure, we have

$$u_{\boldsymbol{n}} = \sup_{\boldsymbol{y}\in D} |\phi_{\boldsymbol{n}}(\boldsymbol{y})| = 2^{|\boldsymbol{n}|_0/2},$$

where $|\boldsymbol{n}|_0 = |\{k : n_k \neq 0\}|$ for $\boldsymbol{n} = (n_1, \ldots, n_d) \in \mathbb{N}_0^d$. For tensor Legendre polynomials with samples drawn from the uniform measure, we have

$$u_{\boldsymbol{n}} = \sup_{\boldsymbol{y}\in D} |\phi_{\boldsymbol{n}}(\boldsymbol{y})| = \prod_{k=1}^d \sqrt{2n_k + 1}.$$

With this in hand, we have the following recovery guarantee for the non-gradient augmented weighted ℓ^1 problem (3.3.10):

Theorem 3.3.1. *[4, Thm. 2] Let $\Lambda \subset \mathbb{N}_0^d$ with $|\Lambda| = N$, $0 \leq \epsilon \leq e^{-1}$, $\eta \geq 0$, $\boldsymbol{w} = (w_{\boldsymbol{n}})_{\boldsymbol{n}\in\Lambda}$ be a vector of weights with $w_{\boldsymbol{n}} \geq 1$, $\forall \boldsymbol{n}$, $\boldsymbol{c} \in \ell^2(\mathbb{N}_0^d)$ and $\Delta \subset \Lambda$, $\Delta \neq \emptyset$ be any fixed set. Draw $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ independently according to the measure ν. Let $\boldsymbol{A}$, $\boldsymbol{f}$ and $\boldsymbol{e}$ be as in* (3.3.6) *and* (3.3.8) *respectively and suppose that η satisfies* (3.3.9)*. Then, with probability at least $1-\epsilon$, any minimizer $\hat{\boldsymbol{c}}$ of* (3.3.10) *satisfies*

$$\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2 \lesssim \lambda\sqrt{|\Delta|_{\boldsymbol{w}}}\left(\eta + \|\boldsymbol{c} - \boldsymbol{c}_\Lambda\|_{1,\boldsymbol{u}}\right) + \|\boldsymbol{c} - \boldsymbol{c}_\Delta\|_{1,\boldsymbol{w}},$$

provided

$$m \gtrsim \left(|\Delta|_{\boldsymbol{u}} + \max_{\boldsymbol{n}\in\Lambda\setminus\Delta}\left\{\frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2}\right\}|\Delta|_{\boldsymbol{w}}\right) L, \tag{3.3.14}$$

where $\lambda = 1 + \frac{\sqrt{\log(\epsilon^{-1})}}{\log\left(2n\sqrt{|\Delta|_{\boldsymbol{w}}}\right)}$ and $L = \log(\epsilon^{-1})\log\left(2n\sqrt{|\Delta|_{\boldsymbol{w}}}\right)$.

Theorem 3.3.1 provides a nonuniform recovery guarantee for the unaugmented weighted ℓ^1 minimization problem (3.3.10). Suppose the coefficient vector $\boldsymbol{c}$ is exactly sparse. With $\Delta = \operatorname{supp}(\boldsymbol{c})$ and $\eta = 0$, the exact recovery of $\boldsymbol{c}$ is obtained when condition (3.3.14) holds. Moreover, in order to minimize the right-hand side of (3.3.14), we should pick $\boldsymbol{w} = \boldsymbol{u}$, so that the second term inside of the parentheses is equal to the first term inside of the parentheses. In other words, the intrinsic weights are the optimal weights for the problem. The same conclusion could be reached from the main result for the gradient-augmented weighted ℓ^1 minimization problem, stated as Theorem 3.7.1.

3.4 Lower sets

In [59, Chpt. 12], the authors presented theoretical results for recovering an approximately sparse vector $\boldsymbol{x} \in \mathbb{C}^N$ using ℓ^1 minimization when the sampling matrix $\boldsymbol{A}$ is assumed to be associated to a bounded orthonormal system. Note that the ℓ^1 minimization problem solved there can be seen as the weighted ℓ^1 minimization problem (3.3.10) with $w_n = 1$. Those results shown in [59, Chpt. 12] imply that, for sparse recovery of an orthonormal polynomial expansion, a sample complexity

$$m \gtrsim K^2 s \times \text{ log factors} \tag{3.4.1}$$

is sufficient to attain the best s-term approximation with ℓ^1 minimization. Here, $K = \sup_{\boldsymbol{n}\in\Lambda}\|\phi_{\boldsymbol{n}}\|_{L^\infty(D)}$ denotes the bound for the underlying orthonormal basis [2, 4, 43]. As shown in [4,41,43,49,128], when the best s-term approximation of a d-dimensional multivariate polynomial expansion is considered, this sample complexity can become exponentially-

large in d. For example, when using Chebyshev polynomials with samples drawn from the Chebyshev measure, this sample complexity becomes $m \gtrsim 2^d \times \log$ factors. Similarly, for Legendre polynomials with sampling with respect to the Chebyshev measure, the best known sample complexity is $m \gtrsim 3^d \times \log$ factors [2, 43, 107]. These results indicate that high-dimensional function approximation with ℓ^1 minimization may suffer from the curse of dimensionality in the sample complexity [4, 43]. The terminology, *curse of dimensionality*, is introduced by Bellman in [19], which describes the exponential blow-up behavior of the complexity with increasing dimension of the problem [4, 43].

However, recent work [2, 4, 43] has shown that such sample complexity is not sharp. We are able to recover polynomial coefficients with a much lower sample complexity by exploiting the additional sparse structure that polynomial coefficients of analytic and high-dimensional functions possess. Specifically, the *lower set* structure:

Definition 3.4.1. *A set $\Delta \subseteq \mathbb{N}_0^d$ is lower if whenever $\boldsymbol{n} = (n_1, \ldots, n_d) \in \Delta$ and $\boldsymbol{n}' = (n_1', \ldots, n_d') \in \mathbb{N}_0^d$ satisfies $n_k' \leq n_k$, $k = 1, \ldots, d$, then $\boldsymbol{n}' \in \Delta$.*

Lower sets (also known as monotone or downward closed sets) have been studied extensively in the context of multivariate polynomial approximation [2,4,40,41,49]. In particular, it has been shown in [42] that, for recovering solutions of a broad class of parametric PDEs, there exist sequences of lower sets of cardinality s which have the same approximation error decay rate as those of the best s-term approximation.

Note that the union of all lower sets of size at most s is precisely the *hyperbolic cross* index set of size s

$$\bigcup \{\Delta : |\Delta| \leq s,\ \Delta \text{ lower}\} = \Lambda_s^{\mathrm{HC}}, \tag{3.4.2}$$

where

$$\Lambda_s^{\mathrm{HC}} = \left\{ \boldsymbol{n} \in \mathbb{N}_0^d : \prod_{k=1}^{d} (n_k + 1) \leq s + 1 \right\}. \tag{3.4.3}$$

Thus, unless specified, we take the hyperbolic cross index set of size s as the finite multi-indices set from where the approximation of f is sought, i.e. $\Lambda = \Lambda_s^{\mathrm{HC}}$. Then, we can estimate the approximation error in terms of the $\ell^1_{\boldsymbol{w}}$-norm error of the best lower s-term approximation of $\boldsymbol{c}$:

$$\sigma_{s,L}(\boldsymbol{c})_{1,\boldsymbol{w}} = \inf \left\{ \|\boldsymbol{c} - \boldsymbol{z}\|_{1,\boldsymbol{w}} :\ \boldsymbol{z} \in \ell^1_{\boldsymbol{w}}(\mathbb{N}_0^d),\ |\mathrm{supp}(\boldsymbol{z})| \leq s,\ \mathrm{supp}(\boldsymbol{z}) \text{ lower} \right\}. \tag{3.4.4}$$

Here, $\mathrm{supp}(\boldsymbol{z}) = \{\boldsymbol{n} : z_{\boldsymbol{n}} \neq 0\}$ is the set of indices where $z_{\boldsymbol{n}}$ is nonzero. As mentioned above, for functions arising as solutions of parametric PDEs, $\sigma_{s,L}(\boldsymbol{c})_{1,\boldsymbol{w}}$ is a reasonable surrogate for the true best s-term approximation

$$\sigma_s(\boldsymbol{c})_{1,\boldsymbol{w}} = \inf \left\{ \|\boldsymbol{c} - \boldsymbol{z}\|_{1,\boldsymbol{w}} :\ \boldsymbol{z} \in \ell^1_{\boldsymbol{w}}(\mathbb{N}_0^d),\ |\mathrm{supp}(\boldsymbol{z})| \leq s \right\}.$$

A series of recent works [2, 4, 43] have studied the problem of high-dimensional function approximation in lower sets via weighted ℓ^1 minimization problem. The following theorem gives the result for approximating with tensor-product Chebyshev or Legendre polynomial expansions:

Theorem 3.4.2. *[4, Thm. 3] Let $\Lambda = \Lambda_s^{HC}$ defined as (3.4.3) with $|\Lambda| = N$, $0 < \epsilon < e^{-1}$, $\eta \geq 0$, $\boldsymbol{c} \in \ell^2(\mathbb{N}_0^d)$ and $\Delta \subseteq \Lambda$ be a lower set and $|\Delta| \leq s$ with $s \geq 2$. Draw $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ independently according to the measure ν. Let $\boldsymbol{A}$, $\boldsymbol{f}$ and $\boldsymbol{e}$ be as in (3.3.6) and (3.3.8) respectively and suppose that the tail error satisfies $\|\boldsymbol{e}\|_2 \leq \eta$. Set weights $\boldsymbol{w} = \boldsymbol{u} = (u_{\boldsymbol{n}})_{\boldsymbol{n}\in\Lambda}$ with $u_{\boldsymbol{n}} = \|\phi_{\boldsymbol{n}}\|_{L^\infty}$, and let $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ be the tensor Legendre or Chebyshev basis. Then, any minimizer $\hat{\boldsymbol{c}}$ of (3.3.10) satisfies*

$$\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2 \lesssim \lambda s^{\gamma/2} \left(\eta + \|\boldsymbol{c} - \boldsymbol{c}_\Lambda\|_{1,\boldsymbol{u}}\right) + \sigma_{s,L}(\boldsymbol{c})_{1,\boldsymbol{u}},$$

where $\sigma_{s,L}(\boldsymbol{c})_{1,\boldsymbol{u}}$ is defined as (3.4.4), with probability at least $1 - \epsilon$, provided

$$m \gtrsim s^\gamma \log(\epsilon^{-1}) \min\{d + \log(s), \log(2d)\log(s)\}, \tag{3.4.5}$$

where $\lambda = 1 + \frac{\sqrt{\log(\epsilon^{-1})}}{\log(s)}$ and $\gamma = \log(3)/\log(2)$ or $\gamma = 2$ in the Chebyshev or Legendre case respectively.

Theorem 3.4.2 has shown that quasi-best s-term approximation in lower sets can be obtained by solving a weighted ℓ^1 minimization problem with a suitable choice of weights. Note that the error bound obtained here is a mixed (ℓ^2, ℓ^1) error bound, which is standard for nonuniform recovery guarantees in compressed sensing. This result gives a quasi-best s-term approximation, in the sense that the error is bounded by the best lower s-term approximation up to a constant factor. It is also worth pointing out that, if the vector $\boldsymbol{c}$ is exactly s-sparse and $\eta = 0$, then an exact recovery of $\boldsymbol{c}$ is obtained. Moreover, it can be clearly seen that the sample complexity, shown as (3.4.5), is significantly smaller than (3.4.1), since (3.4.5) is at most logarithmic in the dimension d and polynomial in s.

As a final remark, Theorem 3.4.2 gives a nonuniform recovery guarantee. Uniform recovery guarantees for this problem have been shown in [4, 43]. Compared to the nonuniform recovery guarantee, the uniform recovery guarantee gives a better error bound by a factor of $1/(s^{\gamma/2})$ when a larger log factor in sample complexity is applied. It is a common difference between nonuniform recovery guarantees and uniform recovery guarantees in compressed sensing.

For more references on high-dimensional function approximation in low sets with applications in parametric PDEs, see [40–42, 49].

3.5 The set-up for the gradient-augmented problem

In the previous sections, we have derived the non-gradient augmented weighted ℓ^1 minimization problem and introduced the concept of lower sets. In this section, we will set up the gradient-augmented weighted ℓ^1 minimization problem, which will be solved in order to approximate the high-dimensional function. Here, we aim to use those additional gradient samples to improve the accuracy of the approximation.

3.5.1 Sturm–Liouville eigenfunctions

The main tool that will be used to define the gradient-augment problem is Sturm–Liouville (SL) theory. Thus, we will review SL theory first.

Recall that a SL problem is an eigenvalue problem of the form

$$-(\chi u')' + \zeta u = \lambda \nu u, \tag{3.5.1}$$

where coefficients χ, ζ, and ν are three real-valued functions in $(-1,1)$. The coefficient χ is continuously differentiable, strictly positive in $(-1,1)$ and continuous in $[-1,1]$, ζ is continuous, nonnegative and bounded in $[-1,1]$ and ν is continuous, nonnegative and integrable over $(-1,1)$. A SL problem is singular when χ vanishes at the boundary, i.e. $\chi(\pm 1) = 0$. Such a problem has a countable set of nonnegative real eigenvalues $0 \leq \lambda_0 < \lambda_1 < \dots$ and the corresponding eigenfunctions $\{\phi_n\}_{n\in\mathbb{N}_0}$ forms an orthogonal basis of $L^2_\nu(-1,1)$ [36, §2.2].

Of relevance to this chapter, the classical orthogonal polynomials are all singular Sturm–Liouville eigenfunctions:

Legendre polynomials. These are Sturm–Liouville eigenfunctions corresponding to

$$\chi(y) = \frac{1}{2}(1-y^2), \qquad \nu(y) = \frac{1}{2}, \qquad \zeta(y) = 0.$$

The corresponding eigenvalues are $\lambda_n = n(n+1)$. Note that it is customary to write $\chi(y) = 1-y^2$ and $\nu(y) = 1$ here. We have normalized by $1/2$ so that ν is a probability density function.

Chebyshev polynomials. These are Sturm–Liouville eigenfunctions corresponding to

$$\chi(y) = \frac{\sqrt{1-y^2}}{\pi}, \qquad \nu(y) = \frac{1}{\pi\sqrt{1-y^2}}, \qquad \zeta(y) = 0.$$

The corresponding eigenvalues are $\lambda_n = n^2$.

Jacobi polynomials. These are Sturm–Liouville eigenfunctions corresponding to

$$\chi(y) = \frac{1}{c^{(\alpha,\beta)}}(1-y)^{\alpha+1}(1+y)^{\beta+1}, \qquad \nu(y) = \frac{(1-y)^{\alpha}(1+y)^{\beta}}{c^{(\alpha,\beta)}}, \qquad \zeta(y) = 0. \tag{3.5.2}$$

where $\alpha, \beta > -1$ and $c^{(\alpha,\beta)} = \int_{-1}^{1}(1-y)^{\alpha}(1+y)^{\beta}\,\mathrm{d}y$. The corresponding eigenvalues are

$$\lambda_n^{(\alpha,\beta)} = n(n+\alpha+\beta+1). \tag{3.5.3}$$

Note that Jacobi polynomials include both Legendre and Chebyshev polynomials as the special cases $\alpha = \beta = 0$ and $\alpha = \beta = -1/2$ respectively.

For the remainder of this chapter, we assume that the orthonormal basis $\{\phi_n\}_{n=0}^{\infty}$ introduced in §3.2 arises as the eigenfunctions of a singular Sturm–Liouville problem (3.5.1). For convenience, we also assume that

$$\zeta(y) = 0. \tag{3.5.4}$$

This is not strictly necessary for what follows. However, it holds for all polynomials we will consider in this chapter; specifically, the orthogonal polynomials discussed above.

3.5.2 Sobolev orthogonality

The main advantage for setting up gradient-augmented problem with Sturm–Liouville (SL) theory is that the derivatives of Sturm–Liouville eigenfunctions are also orthogonal in a particular weighted L^2 space. Details on this will be shown in this subsection. Note that this weighted L^2 space does not usually coincide with the original weighted space $L^2_\nu(-1,1)$. However, two exceptions are the Fourier basis and Hermite polynomial basis, which have been studied in [127] and [102] respectively. The change of weight that occurs in the general case requires some additional scaling when deriving the gradient-augmented problem. See §3.5.3 for more details.

Multiplying equation (3.5.1) by $\overline{\phi_m}$, integrating by parts and using the fact that $\chi(\pm 1) = 0$ since the problem is assumed to be singular, we get

$$\begin{aligned}\int_{-1}^{1} -(\chi(y)\phi_n'(y))'\overline{\phi_m(y)}\,\mathrm{d}y &= -\chi(y)\phi_n'(y)\overline{\phi_m(y)}\Big|_{-1}^{1} + \int_{-1}^{1}\chi(y)\phi_n'(y)\overline{\phi_m'(y)}\,\mathrm{d}y \\ &= \lambda_n \int_{-1}^{1}\nu(y)\phi_n(y)\overline{\phi_m(y)}\,\mathrm{d}y,\end{aligned}$$

Hence, we can see that the derivatives ϕ_n' are orthogonal in $L^2_\chi(-1,1)$, since

$$\int_{-1}^{1}\chi(y)\phi_n'(y)\overline{\phi_m'(y)}\,\mathrm{d}y = \lambda_n\delta_{n,m}, \qquad n,m = 0,1,\ldots. \tag{3.5.5}$$

With this in hand, we define a weighted Sobolev space

$$\tilde{H}^1(-1,1) = \left\{ f \in L^2_\nu(-1,1) : f' \in L^2_\chi(-1,1) \right\},$$

with norm and inner product

$$\|f\|^2_{\tilde{H}^1(-1,1)} = \|f\|^2_{L^2_\nu(-1,1)} + \|f'\|^2_{L^2_\chi(-1,1)}, \qquad \langle f, g\rangle_{\tilde{H}^1(-1,1)} = \langle f, g\rangle_{L^2_\nu(-1,1)} + \langle f', g'\rangle_{L^2_\chi(-1,1)}.$$

It follows from (3.5.5) that the polynomials

$$\psi_n(y) = \frac{1}{\sqrt{1+\lambda_n}}\phi_n(y), \qquad n = 0, 1, 2 \ldots,$$

are an orthonormal system in $\tilde{H}^1(-1,1)$. Moreover, they form an orthonormal basis of $\tilde{H}^1(-1,1)$.

For $d \geq 2$ dimensions, we define the weighted Sobolev space as

$$\tilde{H}^1(D) = \left\{ f \in L^2_\nu(D) : \partial_k f \in L^2_{\nu_k}(D),\ k = 0, \ldots, d \right\}, \tag{3.5.6}$$

where $\nu_k(\boldsymbol{y})$ is the weight function given by

$$\nu_0(\boldsymbol{y}) = \nu(\boldsymbol{y}) = \prod_{j=1}^{d} \nu(y_j), \qquad \nu_k(\boldsymbol{y}) = \chi(y_k) \prod_{\substack{j=1 \\ j\neq k}}^{d} \nu(y_j), \quad k = 1, \ldots, d.$$

The associated norm and inner product are

$$\|f\|^2_{\tilde{H}^1(D)} = \sum_{k=0}^{d} \|\partial_k f\|^2_{L^2_{\nu_k}(D)}, \qquad \langle f, g\rangle_{\tilde{H}^1(D)} = \sum_{k=0}^{d} \langle \partial_k f, \partial_k g\rangle_{L^2_{\nu_k}(D)},$$

respectively. Furthermore, the functions

$$\psi_{\boldsymbol{n}}(\boldsymbol{y}) = \frac{1}{\sqrt{1+\lambda_{\boldsymbol{n}}}}\phi_{\boldsymbol{n}}(\boldsymbol{y}), \quad \boldsymbol{n} \in \mathbb{N}_0^d,$$

where

$$\lambda_{\boldsymbol{n}} = \sum_{k=1}^{d} \lambda_{n_k}, \tag{3.5.7}$$

form an orthonormal basis of $\tilde{H}^1(D)$.

Since it will be useful later, we now make one further observation. Let $g \in \tilde{H}^1(D)$. By assumption, we have $g \in L^2_\nu(D)$ and we may write

$$g = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}, \qquad c_{\boldsymbol{n}} = \langle g, \phi_{\boldsymbol{n}}\rangle_{L^2_\nu(D)},$$

so that

$$\|g\|_{L^2_\nu(D)}^2 = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} |c_{\boldsymbol{n}}|^2.$$

Moreover, due to the orthogonality relations, the coefficients of g with respect to the basis $\psi_{\boldsymbol{n}}$ are

$$\langle g, \psi_{\boldsymbol{n}}\rangle_{\tilde{H}^1(D)} = \sqrt{1+\lambda_{\boldsymbol{n}}}c_{\boldsymbol{n}}.$$

In particular,

$$\|g\|_{\tilde{H}^1(D)}^2 = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} (1+\lambda_{\boldsymbol{n}})|c_{\boldsymbol{n}}|^2.$$

3.5.3 The gradient-augmented weighted ℓ^1 minimization problem

With the weighted Sobolev space introduced, we are now ready to formulate the gradient-augmented weighted ℓ^1 minimization problem.

Following the notation defined in §3.2, we define the sampling matrices by

$$\boldsymbol{A}_k = \frac{1}{\sqrt{m}}\left(\frac{\partial \phi_{\boldsymbol{n}_j}(\boldsymbol{y}_i)}{\partial y_k}\right)_{i=1,j=1}^{m,N} \in \mathbb{C}^{m\times N}, \qquad k = 0,\dots,d,$$

where $k=0$ denotes the case that no partial derivative is taken, i.e.

$$\boldsymbol{A}_0 = \frac{1}{\sqrt{m}}\left(\phi_{\boldsymbol{n}_j}(\boldsymbol{y}_i)\right)_{i=1,j=1}^{m,N} \in \mathbb{C}^{m\times N}.$$

Then, we have a system of linear equations for the gradient-augmented problem

$$\frac{1}{\sqrt{m}}\left(\partial_k f(\boldsymbol{y}_i)\right)_{i=1}^m = \boldsymbol{A}_k \boldsymbol{c}_\Lambda + \frac{1}{\sqrt{m}}\left(\partial_k e_\Lambda(\boldsymbol{y}_i)\right)_{i=1}^m, \tag{3.5.8}$$

where $\boldsymbol{c}_\Lambda$ denotes the vector of coefficients of f corresponding to the finite multi-indices set Λ and the truncation error e_Λ is defined as in (3.3.2). For reasons that will become clear in a moment, we let

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{T}_0\boldsymbol{A}_0 \\ \boldsymbol{T}_1\boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{T}_d\boldsymbol{A}_d \end{bmatrix} \in \mathbb{C}^{(d+1)m\times N},$$

where $\boldsymbol{T}_k = \operatorname{diag}\left(\left(\sqrt{\tau_k(\boldsymbol{y}_i)}\right)_{i=1}^m\right) \in \mathbb{C}^{m\times m}$ are diagonal scaling matrices, and the τ_k are defined by

$$\tau_0(\boldsymbol{y}) = \frac{\prod_{j=1}^d \nu(y_j)}{\prod_{j=1}^d \mu(y_j)} = \frac{\nu_0(\boldsymbol{y})}{\mu(\boldsymbol{y})}, \qquad \tau_k(\boldsymbol{y}) = \frac{\chi(y_k)\prod_{j=1,j\neq k}^d \nu(y_j)}{\prod_{j=1}^d \mu(y_j)} = \frac{\nu_k(\boldsymbol{y})}{\mu(\boldsymbol{y})}, \quad k=1,\dots,d.$$

As we will show in (3.5.12), the diagonal scaling matrices $\boldsymbol{T}_k$ are used to ensure that $\bar{\boldsymbol{A}}^*\bar{\boldsymbol{A}}$ is diagonal in expectation. With this in hand, we have a system of linear equations for the gradient-augmented problem as

$$\boldsymbol{f} = \bar{\boldsymbol{A}}\boldsymbol{c}_\Lambda + \boldsymbol{e}, \tag{3.5.9}$$

where

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_0 \\ \vdots \\ \boldsymbol{f}_d \end{bmatrix}, \qquad \boldsymbol{f}_k = \frac{1}{\sqrt{m}} \left(\sqrt{\tau_k(\boldsymbol{y}_i)} \partial_k f(\boldsymbol{y}_i) \right)_{i=1}^m, \tag{3.5.10}$$

and

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_0 \\ \vdots \\ \boldsymbol{e}_d \end{bmatrix}, \qquad \boldsymbol{e}_k = \frac{1}{\sqrt{m}} \left(\sqrt{\tau_k(\boldsymbol{y}_i)} \partial_k e_\Lambda(\boldsymbol{y}_i) \right)_{i=1}^m.$$

As for the unaugmented problem, we shall assume that the tail error satisfies

$$\|\boldsymbol{e}\|_2 \leq \eta, \tag{3.5.11}$$

for some known $\eta \geq 0$, which is implied by the condition

$$\sup_{y \in D} \sum_{k=0}^{d} \tau_k(y) |\partial_k e_\Lambda(y)|^2 \leq \eta^2.$$

Recall that the sample point $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ are independently and identically distributed according to the probability density μ. Due to the diagonal scaling matrices $\boldsymbol{T}_k$ and the Sobolev orthogonality of the basis functions, we have

$$\begin{aligned} \mathbb{E}\left(\bar{\boldsymbol{A}}^*\bar{\boldsymbol{A}}\right)_{\boldsymbol{n},\boldsymbol{n}'} &= \sum_{k=0}^{d} \int_D \partial_k \phi_{\boldsymbol{n}}(\boldsymbol{y}) \overline{\partial_k \phi_{\boldsymbol{n}'}(\boldsymbol{y})} \tau_k(\boldsymbol{y}) \mu(\boldsymbol{y}) \, dy \\ &= \sum_{k=0}^{d} \int_D \partial_k \phi_{\boldsymbol{n}}(\boldsymbol{y}) \overline{\partial_k \phi_{\boldsymbol{n}'}(\boldsymbol{y})} \nu_k(\boldsymbol{y}) \, dy = (1 + \lambda_{\boldsymbol{n}}) \, \delta_{\boldsymbol{n},\boldsymbol{n}'}. \end{aligned} \tag{3.5.12}$$

Then, we introduce a diagonal scaling matrix $\boldsymbol{Q} = \operatorname{diag}\left(\sqrt{1 + \lambda_{\boldsymbol{n}}}\right)_{\boldsymbol{n} \in \Lambda}$, so that the scaled matrix

$$\boldsymbol{A} = \bar{\boldsymbol{A}}\boldsymbol{Q}^{-1}, \tag{3.5.13}$$

satisfies $\mathbb{E}(\boldsymbol{A}^*\boldsymbol{A}) = \boldsymbol{I}$. This condition allows us to prove that any vector $\boldsymbol{c}_\Lambda$ could be reconstructed by solving the optimization problem (3.5.14) with enough measurements [30].

With this new sampling matrix $\boldsymbol{A}$ in hand, we can now formulate the gradient-augmented weighted ℓ^1 minimization problem as

$$\min_{\boldsymbol{z}\in\mathbb{C}^N} \|\boldsymbol{z}\|_{1,\boldsymbol{w}} \text{ subject to } \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{f}\|_2 \leq \eta, \tag{3.5.14}$$

If the minimization problem (3.5.14) has a solution $\hat{\boldsymbol{z}}$, then we have $\hat{\boldsymbol{c}} = \boldsymbol{Q}^{-1}\hat{\boldsymbol{z}}$ as an approximation of the true coefficients $\boldsymbol{c}_\Lambda$. Then the corresponding approximation of the original function f is given as

$$\hat{f} = \sum_{\boldsymbol{n}\in\Lambda} \hat{c}_{\boldsymbol{n}}\phi_{\boldsymbol{n}}.$$

Finally, we have the following observation. If f_Λ is defined as in (3.3.2), then, due to the Sobolev orthogonality, we have that

$$\left\|f_\Lambda - \hat{f}\right\|_{\tilde{H}^1(D)} = \|\boldsymbol{Q}(\boldsymbol{c}_\Lambda - \hat{\boldsymbol{c}})\|_2 = \|\boldsymbol{z}_\Lambda - \hat{\boldsymbol{z}}\|_2, \tag{3.5.15}$$

where $\boldsymbol{z}_\Lambda = \boldsymbol{Q}\boldsymbol{c}_\Lambda$ are the coefficients vector of f with respect to the Sobolev orthogonal basis $\{\psi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$. Note that, since the analysis of the gradient-augmented problem (3.5.14) will provide an error bound for $\|\boldsymbol{z}_\Lambda - \hat{\boldsymbol{z}}\|_2$, we can use this bound to get an approximation error bound in the Sobolev norm $\tilde{H}^1(D)$ by considering (3.5.15).

3.6 Numerical results

As mentioned at the beginning, the objective for this chapter is to use additional gradient information to enhance the accuracy of the approximation to the high-dimensional function f. In this section, we wish to show numerically that an accuracy improvement is achieved with gradient-augmented sampling, when tensor-product Legendre and Chebyshev polynomials are used as the approximating polynomials. For all experiments shown in this section, we have assumed that the computational cost of computing the gradient is the same as the cost of computing function values. It is a reasonable assumption in certain uncertainty quantification (UQ) applications. For instance, when considering approximating a quantity of interest (QoI) of a parametric differential equation, it is known that the gradient samples of the QoI can be computed using about the same amount of computational cost as computing the QoI samples via the adjoint sensitivity analysis method [102]. For details on how to compute the gradient samples of the QoI with adjoint sensitivity analysis method and on the computational cost of computing the gradient samples of the QoI, see §4.2 and §4.7. Thus, in this section, we model the total cost of computing the gradient-augmented measurements as

$$\tilde{m} = m_o + m_g, \tag{3.6.1}$$

where m_o is the number of function samples and m_g is the number of the gradient samples. Unless otherwise specified, the gradient samples are measured at the same points as the function samples. This is a reasonable assumption since, as will be shown in §4.3, in order to compute the gradient samples with the adjoint sensitivity analysis method, we first need to compute the function samples. Hence, it will be more expensive to measure the gradient samples at different points to the function samples. Note that, for the unaugmented problem, the total computational cost is just $\tilde{m} = m_o$.

Throughout this section, we solve the weighted ℓ^1 minimization problem using the SPGL1 package [118, 119] with a maximum number of 10,000 iterations and $\eta = 10^{-12}$. We choose the truncated index set Λ as the hyperbolic cross index set of degree s. Different values of s are picked in different experiments so that $N = |\Lambda| \approx 2500$. For Figures 3.1–3.8, the $\tilde{H}^1$ norm approximation error is computed on a fixed grid of $4|\Lambda|$ points drawn according to the uniform density for Legendre polynomials and the Chebyshev density for Chebyshev polynomials. The error is averaged over 10 trials. All numerical experiments are performed on Matlab R2019a [91].

3.6.1 Approximation error in the $\tilde{H}^1$ norm

In this subsection, we will show how the $\tilde{H}^1$ norm approximation error is calculated numerically. Suppose we have an error grid containing K points, $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K \in D$. These points are independently and identically distributed according to the probability measure μ. Let

$$\bar{\mathbf{e}} = \begin{bmatrix} \boldsymbol{T}_0 \mathbf{e}_0 \\ \boldsymbol{T}_1 \mathbf{e}_1 \\ \vdots \\ \boldsymbol{T}_d \mathbf{e}_d \end{bmatrix} \in \mathbb{C}^{(K(d+1))\times 1},$$

where diagonal scaling matrices $\boldsymbol{T}_k = \operatorname{diag}\left(\left(\sqrt{\tau_k(\boldsymbol{y}_i)}\right)_{i=1}^K\right) \in \mathbb{C}^{K\times K}$ with τ_k defined as in §3.5.3 and

$$\mathbf{e}_k = \frac{1}{\sqrt{K}}\left(\partial_k f(\boldsymbol{y}_i) - \partial_k \hat{f}(\boldsymbol{y}_i)\right)_{i=1}^K, \quad k = 0, \ldots, d.$$

Here, the approximations of $(\partial_k f(\boldsymbol{y_i}))_{i=1}^K$ are defined by

$$\left(\partial_k \hat{f}(\boldsymbol{y_i})\right)_{i=1}^K = \boldsymbol{G}_k \hat{\boldsymbol{c}},$$

where the matrices $\boldsymbol{G}_k = \left(\frac{\partial \phi_{\boldsymbol{n}_j}(\boldsymbol{y}_i)}{\partial y_k}\right)_{i=1,j=1}^{K,N} \in \mathbb{C}^{K\times N}$ and $\hat{\boldsymbol{c}} \in \mathbb{C}^N$ denotes the approximation to the true coefficients vector $\boldsymbol{c}_\Lambda$. Then, we have

$$\begin{aligned}\mathbb{E}\left(\bar{\mathbf{e}}^*\bar{\mathbf{e}}\right) &= \sum_{k=0}^{d}\sum_{i=1}^{K}\mathbb{E}\left(\overline{(\mathbf{e}_k(\boldsymbol{y}_i))}\mathbf{e}_k(\boldsymbol{y}_i)\tau_k(\boldsymbol{y}_i)\right) = \sum_{k=0}^{d} K\int_D \overline{(\mathbf{e}_k(\boldsymbol{y}))}\mathbf{e}_k(\boldsymbol{y})\tau_k(\boldsymbol{y})\mu(\boldsymbol{y})\ d\boldsymbol{y}\\ &= \sum_{k=0}^{d}\int_D \left|\partial_k f(\boldsymbol{y}) - \partial_k \hat{f}(\boldsymbol{y})\right|^2 \nu_k(\boldsymbol{y})\ d\boldsymbol{y} = \sum_{k=0}^{d}\left\|\partial_k f - \partial_k \hat{f}\right\|^2_{L^2_{\nu_k}(D)} = \left\|f-\hat{f}\right\|^2_{\tilde{H}^1(D)}.\end{aligned}$$

When the number of samples K is large, with the law of large numbers, we have

$$\begin{aligned}\left\|f-\hat{f}\right\|^2_{\tilde{H}^1(D)} &\approx \sum_{k=0}^{d}\|T_k\mathbf{e}_k\|^2\\ \Rightarrow \left\|f-\hat{f}\right\|_{\tilde{H}^1(D)} &\approx \left(\sum_{k=0}^{d}\|T_k\mathbf{e}_k\|^2\right)^{1/2}.\end{aligned}$$

For all experiments shown in this section, we set $K = 4|\Lambda| = 4N$.

3.6.2 Unaugmented and gradient-augmented problem comparison

In the first set of experiments, the weights are taken to be $w_{\boldsymbol{n}} = (u_{\boldsymbol{n}})^\theta$ for some $\theta \geq 0$, where

$$u_{\boldsymbol{n}} = \sup_{\boldsymbol{y}\in D}\sqrt{\nu(\boldsymbol{y})/\mu(\boldsymbol{y})}|\phi_{\boldsymbol{n}}(\boldsymbol{y})|.$$

The same way of defining weights has also been applied to the unaugmented problem shown in §3.3. When $\theta = 0$, the weighted ℓ^1 minimization problem becomes the unweighted ℓ^1 minimization problem. Moreover, when $\theta = 1$, it corresponds to the optimal weights $\boldsymbol{w} = \boldsymbol{u}$ as shown in Theorem 3.7.1. As mentioned in Theorem 3.3.1, it is also the optimal weights for the unaugmented problem. Here, we apply different weights to show that, for the gradient-augmented case, the weighted ℓ^1 minimization gives better approximation results compared to the unweighted ℓ^1 minimization.

In this section, we consider approximating the following functions

$$\begin{aligned}&\text{Figures 3.1 \& 3.2:} && f_1(\boldsymbol{y}) = \prod_{k=1}^{d}\frac{d/4}{d/4+(y_k-a_k)^2}, \quad a_k = \frac{(-1)^k}{k+1},\\ &\text{Figures 3.3 \& 3.4:} && f_2(\boldsymbol{y}) = \prod_{k=d/2+1}^{d}\cos(16y_k/2^k)/\prod_{k=1}^{d/2}(1-y_k/4^k),\\ &\text{Figures 3.5 \& 3.6:} && f_3(\boldsymbol{y}) = \exp\left(-\sum_{k=1}^{d}y_k/(2d)\right).\end{aligned}$$

The main conclusion of these experiments is the following. In all dimensions and for all functions, we see that, with the same amount of computational cost $\tilde{m}$, a consistently smaller error is obtained by solving the gradient-augmented problem. In other words, applying the gradient sampling is more beneficial than only taking an equivalent number of function samples.

Figures 3.1–3.6 also compare different weighting strategies for the weighted ℓ^1 minimiztion problem. For those functions we have tested, in most cases, the choice $\boldsymbol{w} = \boldsymbol{u}$ gives amongst the smallest, if not the smallest, error. In particular, these weights often give an improvement over the unweighted case. This finding is in agreement with the theoretical result shown in Theorem 3.7.1. Note that larger values of θ can sometimes give a slightly smaller error for some of the functions we have considered, but the improvement is not substantial.

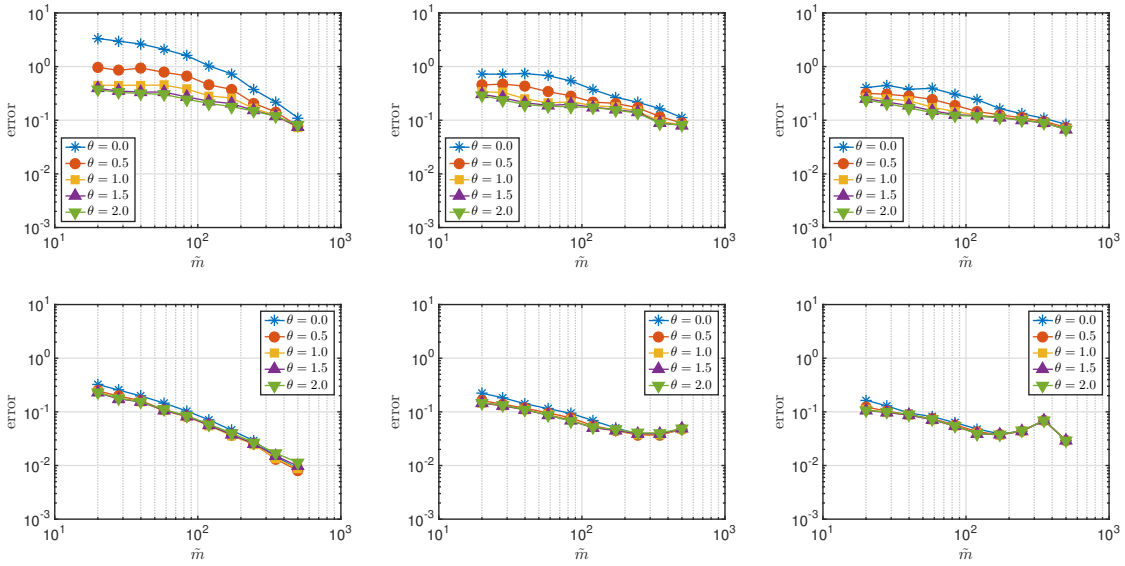


Figure 3.1: The error $\|f_1 - \tilde{f}_1\|_{\tilde{H}^1(D)}$ against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density. From left to right, the values $(d, s) = (4, 72), (8, 23), (12, 14)$ were used. The unaugmented case is shown on the top row and the gradient-augmented case is shown on the bottom row.

3.6.3 With partial sampling of the gradient

In this experiment, we fix the weights as $\boldsymbol{w} = \boldsymbol{u}$ and consider the situation when the gradient is measured at only a fixed percentage of the sample points. A similar set-up has also been considered in [102]. We plot the approximation error of f_3 versus the effective cost $\tilde{m}$ defined in (3.6.1), shown as Figure 3.7. These results show a clear improvement with only 25% of gradient sampling. It can also be clearly seen that, as the percentage of gradient samples increases, the approximation error correspondingly decreases.

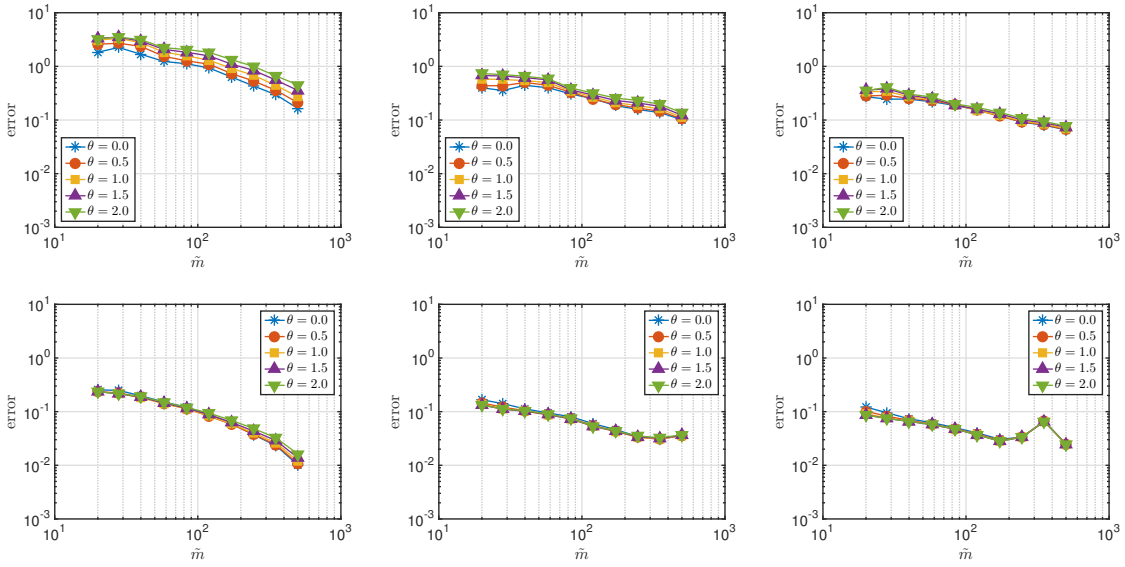


Figure 3.2: The same as Fig. 3.1 but for Chebyshev polynomials with points drawn from the Chebyshev density.

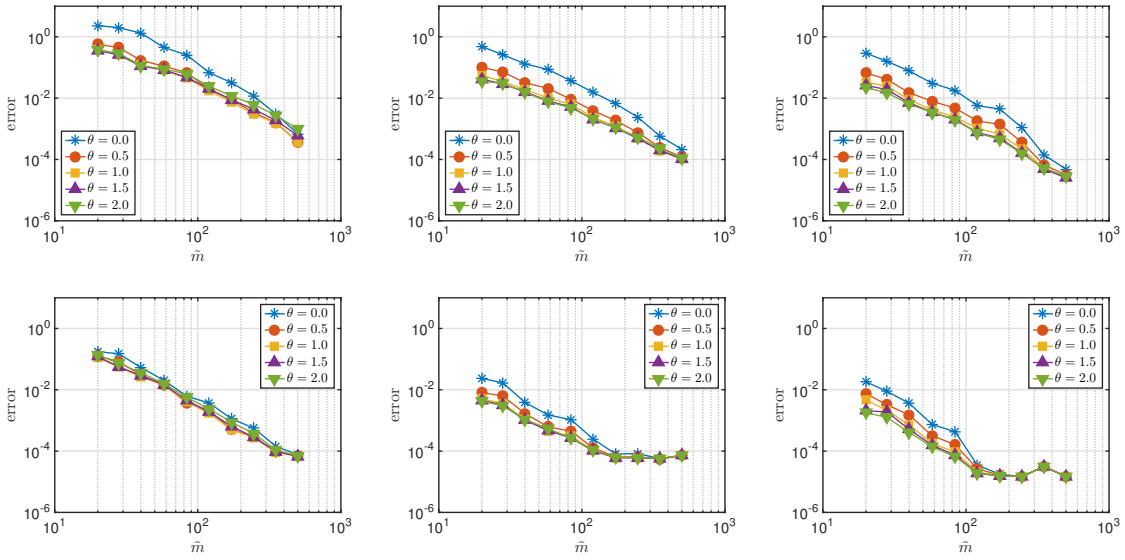


Figure 3.3: The same as Fig. 3.1 but for f_2.

3.6.4 With independent gradient sampling locations

In Figure 3.8, we investigate how the location of the gradient samples affects the approximation error. Specifically, we compare the existing set-up where the gradient of f is sampled at the same points as function f to the case of independent gradient sampling, i.e. where the gradient of f is sampled at different m points $\boldsymbol{y}_{m+1}, \ldots, \boldsymbol{y}_{2m}$ drawn independently and from the same density as $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$. It can be clearly seen that, in all dimensions, independent gradient sampling gives similar approximation results to the original set-up for the same

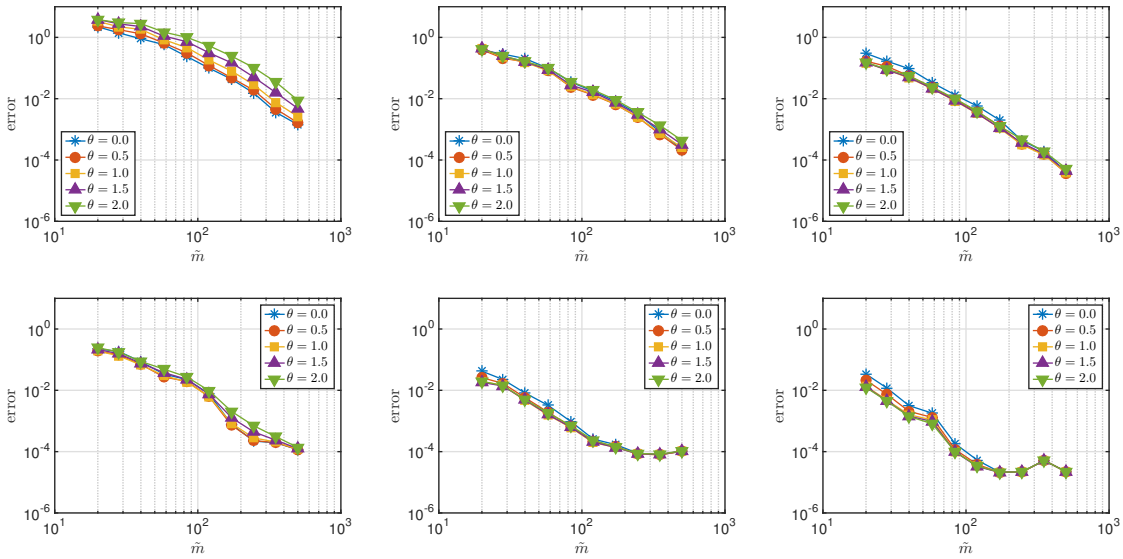


Figure 3.4: The same as Fig. 3.3 but for Chebyshev polynomials with points drawn from the Chebyshev density.

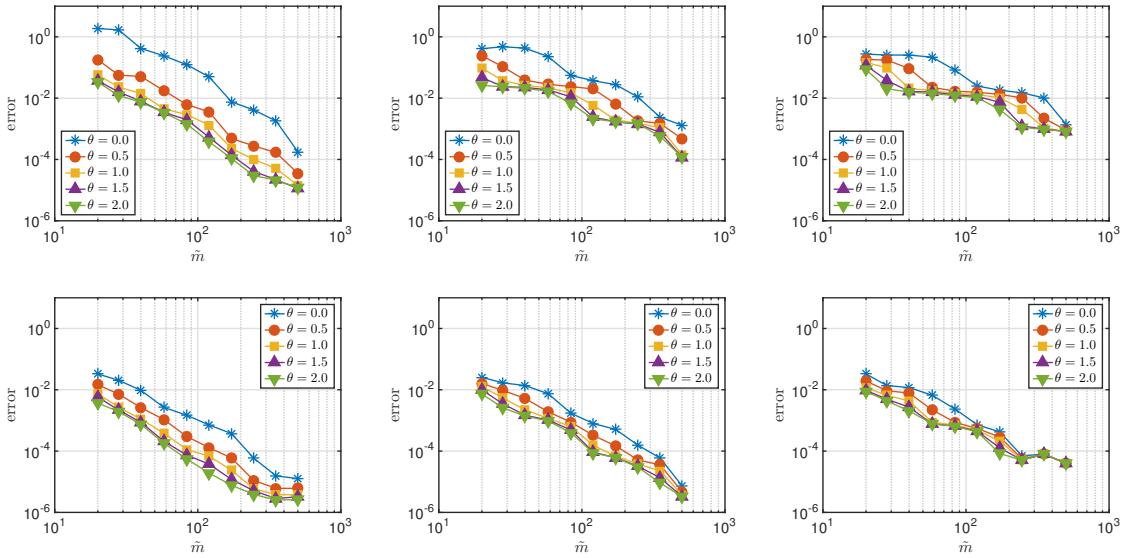


Figure 3.5: The same as Fig. 3.1 but for f_3.

computational cost. As shown in Figure 3.8, there is apparently little benefit to sampling the gradient at a distinct set of sample points. Note that here we do not take into account the fact that, in practice, sampling the gradient of f at distinct points may be more expensive. As shown in §4.3, in order to compute the gradient samples of the QoI with the adjoint sensitivity analysis method, we first need to compute the QoI samples at the same sample points.

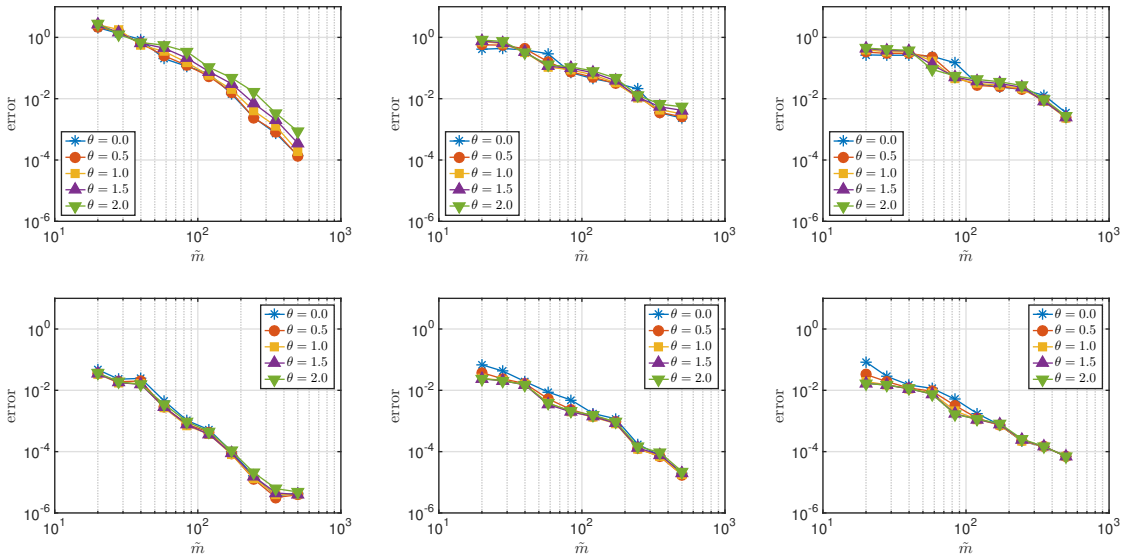


Figure 3.6: The same as Fig. 3.5 but for Chebyshev polynomials with points drawn from the Chebyshev density.

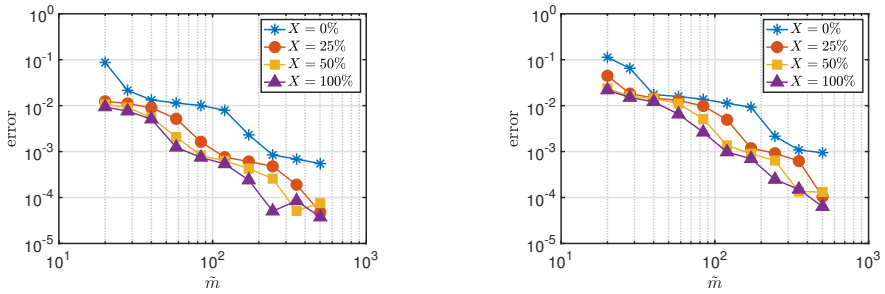


Figure 3.7: The error $\|f_3 - \tilde{f}_3\|_{\tilde{H}^1(D)}$ against $\tilde{m}$ with a different percentage of gradient enhancement. The values $(d, s) = (12, 14)$ were used. The left plot shows the results for Legendre polynomials with uniform sampling and the right plot shows the results for Chebyshev polynomials with Chebyshev sampling.

3.6.5 Comparison in the L^∞ norm error

Finally, in Figure 3.9, we compare the L^∞-norm error for the unaugmented and gradient-augmented cases. Here, the approximation error is computed on a fixed grid of $4|\Lambda|$ uniformly distributed points and averaged over 10 trials. Again, we fix the weights $\boldsymbol{w} = \boldsymbol{u}$. As shown in Figures 3.1–3.6, with the same amount of computational cost, the gradient-augmented sampling leads to a smaller error in the L^∞ norm compared to the non-gradient augmented sampling.

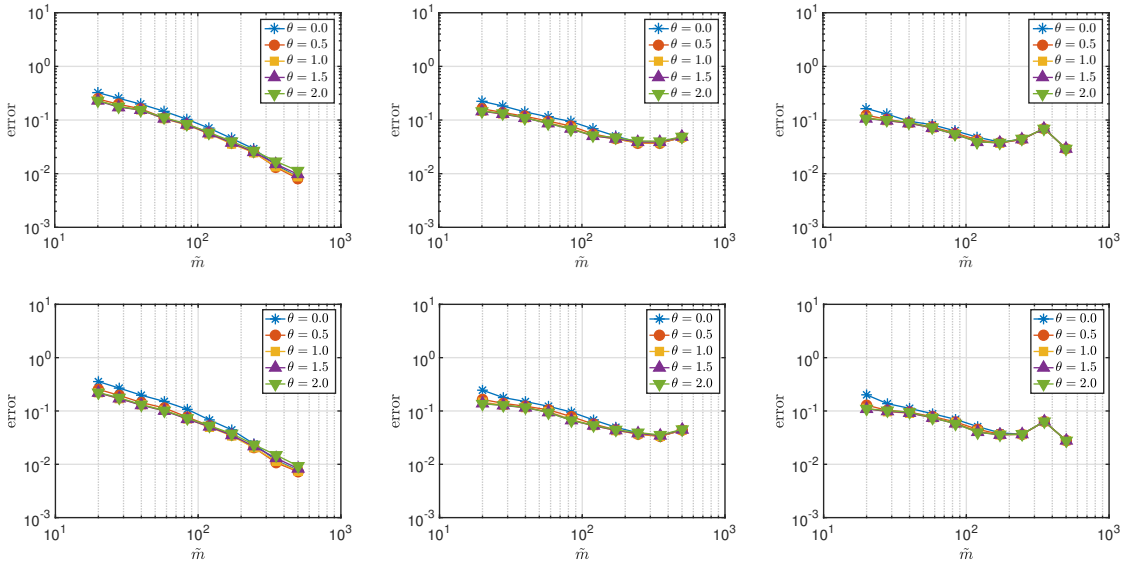


Figure 3.8: The error $\|f_1 - \tilde{f}_1\|_{\tilde{H}^1(D)}$ against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density. From left to right, the values $(d, s) = (4, 72), (8, 23), (12, 14)$ were used. The top row shows the original setup, and the bottom row shows independent gradient sampling.

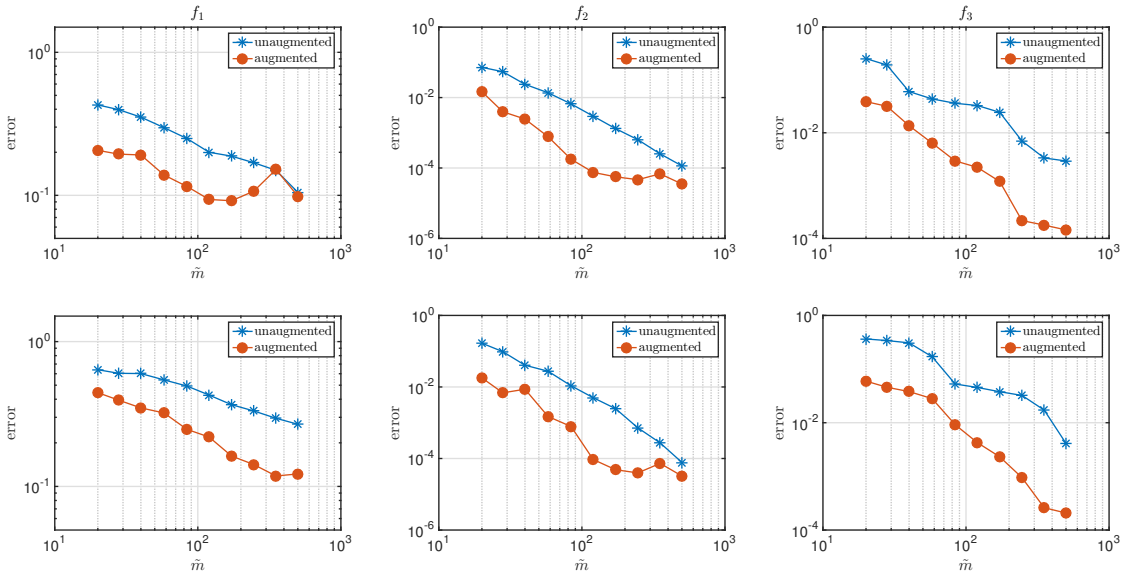


Figure 3.9: The error $\|f - \tilde{f}\|_{L^\infty}$ against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density (top) and Chebyshev polynomials with points drawn from the Chebyshev density (bottom). Function f_1 to f_3 are shown from left to right. The value $(d, s) = (12, 14)$ was used to generate the index set.

3.7 Theoretical results

In §3.6, we have shown various numerical experiments to demonstrate the benefits of gradient-augmented sampling. In this section, we will present theoretical results for the

gradient-augmented problem. It is worth pointing out that all results shown in this section are nonuniform recovery guarantees. Before stating those results, several additional definitions must be introduced. First, recall from §3.3, the weighted cardinality for given positive weights $\boldsymbol{w} = (w_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathbb{N}_0^d}$ and a set $\Delta \subset \mathbb{N}_0^d$ is defined by

$$|\Delta|_{\boldsymbol{w}} = \sum_{\boldsymbol{n}\in\Delta} w_{\boldsymbol{n}}^2.$$

Second, given ν, μ and $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ as in §3.2 and §3.5, we define the intrinsic weights $\boldsymbol{u} = (u_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathbb{N}_0^d}$ as

$$u_{\boldsymbol{n}} = \sup_{\boldsymbol{y}\in D} \sqrt{\nu(\boldsymbol{y})/\mu(\boldsymbol{y})}|\phi_{\boldsymbol{n}}(\boldsymbol{y})|. \tag{3.7.1}$$

Third, we let

$$\kappa_n = u_n^{-2} \sup_{y\in(-1,1)} \frac{\chi(y)}{\mu(y)}|\phi_n'(y)|^2, \qquad n = 0,1,2,\ldots, \tag{3.7.2}$$

where χ is the coefficient from Sturm-Liouville problem (shown as (3.5.1)), and for $\boldsymbol{n} \in \mathbb{N}_0^d$, we set

$$\kappa_{\boldsymbol{n}} = \sum_{k=1}^{d} \kappa_{n_k}. \tag{3.7.3}$$

Finally, given $\boldsymbol{c}_\Lambda$, $\boldsymbol{c}_\Delta \in \mathbb{C}^N$ denotes the vector obtained from $\boldsymbol{c}_\Lambda$ by setting all terms corresponding to indices $\boldsymbol{n} \in \Lambda\backslash\Delta$ to zero.

3.7.1 General recovery guarantees

In this subsection, we will present the general recovery guarantees for the gradient-augmented problem. Here, the first result is given in the following:

Theorem 3.7.1. *Let $\Lambda \subset \mathbb{N}_0^d$ with $|\Lambda| = N \geq 2$, $0 < \epsilon < 1$, $\boldsymbol{w} \in \mathbb{R}^N$ be a vector of weights with $w_{\boldsymbol{n}} \geq 1$, $\forall \boldsymbol{n}$, $\Delta \subset \Lambda$, $|\Delta| \geq 2$ and $f = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}} \in \tilde{H}^1(D)$, where $D = (-1,1)^d$ and $\tilde{H}^1(D)$ is as in (3.5.6). Let*

$$m \gtrsim \max_{\boldsymbol{n}\in\Lambda}\left\{\frac{1+\kappa_{\boldsymbol{n}}}{1+\lambda_{\boldsymbol{n}}}\right\} \cdot \left(|\Delta|_{\boldsymbol{u}} + \max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2}\right\}|\Delta|_{\boldsymbol{w}}\right) \cdot L, \tag{3.7.4}$$

where

$$L = \log(N/\epsilon) + \log(|\Delta|_{\boldsymbol{w}}) \cdot \log(|\Delta|_{\boldsymbol{w}}/\epsilon),$$

draw $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m$ independently according to the density μ, and let $\boldsymbol{A}$, $\boldsymbol{f}$ and η be as in (3.5.13), (3.5.10) and (3.5.11) respectively. Then, if $\hat{\boldsymbol{z}}$ is any minimizer of (3.5.14) and $\hat{\boldsymbol{c}} = \boldsymbol{Q}^{-1}\hat{\boldsymbol{z}}$, the approximation $\hat{f} = \sum_{\boldsymbol{n}\in\Lambda}\hat{c}_{\boldsymbol{n}}\phi_{\boldsymbol{n}}$ satisfies

$$\|f - \hat{f}\|_{\tilde{H}^1(D)} \lesssim \|f - f_\Lambda\|_{\tilde{H}^1(D)} + \|\boldsymbol{c}_\Lambda - \boldsymbol{c}_\Delta\|_{1,\boldsymbol{v}} + \sqrt{|\Delta|_{\boldsymbol{w}}}\eta,$$

with probability at least $1-\epsilon$*, where* $v_{\boldsymbol{n}} = \sqrt{1+\lambda_{\boldsymbol{n}}}w_{\boldsymbol{n}}$*,* $\boldsymbol{n} \in \mathbb{N}_0^d$*.*

Theorem 3.7.1 is understood as follows. For a fixed function f with coefficients $\boldsymbol{c}$ and a fixed set Δ, with a high probability, f can be approximated up to an error (measured in a Sobolev norm) depending on how well $\boldsymbol{c}_\Lambda$ is approximated by its coefficients with indices in Δ (the term $\|\boldsymbol{c}_\Lambda - \boldsymbol{c}_\Delta\|_{1,\boldsymbol{v}}$) by drawing m samples independently according to the probability measure μ, if the sample complexity (3.7.4) holds.

An immediate consequence of Theorem 3.7.1 is that, in order to minimize the right-hand side of (3.7.4), the weights $\boldsymbol{w}$ should be chosen as

$$\boldsymbol{w} = \boldsymbol{u}.$$

In other words, the best weights for the optimization problem are the intrinsic weights defined as in (3.7.1). As pointed out earlier, this is identical to a conclusion obtained for the unaugmented problem. See the discussion after Theorem 3.3.1 for more details about the unaugmented problem. Note that this result makes no assumptions on Δ. If we assume that Δ is lower (recall §3.4), now the theoretical result becomes:

Corollary 3.7.2. *Let* $s \geq 2$*,* $\Lambda = \Lambda_s^{\mathrm{HC}}$ *be the hyperbolic cross index set defined in (3.4.3),* $0 < \epsilon < 1$ *and* $f = \sum_{\boldsymbol{n}\in\mathbb{N}_0^d} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}} \in \tilde{H}^1(D)$*, where* $D = (-1,1)^d$ *and* $\tilde{H}^1(D)$ *is as in (3.5.6). Suppose that*

$$m \gtrsim \max_{\boldsymbol{n}\in\Lambda} \left\{ \frac{1+\kappa_{\boldsymbol{n}}}{1+\lambda_{\boldsymbol{n}}} \right\} \cdot K(s) \cdot L. \tag{3.7.5}$$

where $L = (\min\{d + \log(s/\epsilon), \log(2d)\log(s/\epsilon)\} + \log(K(s)) \cdot \log(K(s)/\epsilon))$*,*

$$K(s) = \max\left\{ |\Delta|_{\boldsymbol{u}} : |\Delta| \leq s \text{ and } \Delta \text{ is lower} \right\}, \tag{3.7.6}$$

and $\boldsymbol{u}$ *are the weights defined in (3.7.1). Draw* $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ *independently according to the density* μ*, let* $\boldsymbol{A}$*,* $\boldsymbol{f}$ *and* η *be as in (3.5.13), (3.5.10) and (3.5.11) respectively and set* $\boldsymbol{w} = \boldsymbol{u}$*. Then, if* $\hat{\boldsymbol{z}}$ *is any minimizer of (3.5.14) and* $\hat{\boldsymbol{c}} = \boldsymbol{Q}^{-1}\hat{\boldsymbol{z}}$*, the approximation* $\hat{f} = \sum_{\boldsymbol{n}\in\Lambda} \hat{c}_{\boldsymbol{n}}\phi_{\boldsymbol{n}}$ *satisfies*

$$\|f - \hat{f}\|_{\tilde{H}^1(D)} \lesssim \|f - f_\Lambda\|_{\tilde{H}^1(D)} + \sigma_{s,L}(\boldsymbol{c}_\Lambda)_{1,\boldsymbol{v}} + \sqrt{K(s)}\eta,$$

with probability at least $1-\epsilon$*, where* $\sigma_{s,L}(\cdot)_{1,\boldsymbol{v}}$ *is as in (3.4.4) and* $v_{\boldsymbol{n}} = \sqrt{1+\lambda_{\boldsymbol{n}}}u_{\boldsymbol{n}}$*,* $\boldsymbol{n} \in \mathbb{N}_0^d$*.*

We see that the error estimate shown in Corollary 3.7.2 is given in terms of the best s-term approximation error in lower sets $\sigma_{s,L}(\cdot)_{1,\boldsymbol{v}}$. However, the sample complexity shown as (3.7.5) is not given explicitly in terms of the sparsity s and dimension d. In order to do this, we need to estimate the quantities $(1+\kappa_{\boldsymbol{n}})/(1+\lambda_{\boldsymbol{n}})$ for $\boldsymbol{n} \in \Lambda$ and $K(s)$, which requires the basis $\{\phi_{\boldsymbol{n}}\}$ and sampling density μ to be specified. We will do the estimations of these quantities in §3.7.2.

3.7.2 The case of Jacobi polynomials with $\mu = \nu$

If we use the Jacobi polynomial basis, defined in §3.2, and take the sampling density $\mu = \nu$, then we have the following corollary:

Corollary 3.7.3. *Consider the set-up of Corollary 3.7.2, where $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ is the tensor-product Jacobi polynomial basis with parameters $\alpha, \beta \geq -1/2$ and sampling density $\mu = \nu$. Suppose that*

$$m \gtrsim K(s) \cdot (\min\{d + \log(s/\epsilon), \log(2d)\log(s/\epsilon)\} + \log(K(s)) \cdot \log(K(s)/\epsilon)), \tag{3.7.7}$$

where

$$K(s) = K^{(\alpha,\beta)}(s) = \max\{|\Delta|_{\boldsymbol{u}} : |\Delta| \leq s \text{ and } \Delta \text{ is lower}\}. \tag{3.7.8}$$

Then, if $\hat{\boldsymbol{z}}$ is any minimizer of (3.5.14) and $\hat{\boldsymbol{c}} = \boldsymbol{Q}^{-1}\hat{\boldsymbol{z}}$, the approximation $\hat{f} = \sum_{\boldsymbol{n}\in\Lambda} \hat{c}_{\boldsymbol{n}}\phi_{\boldsymbol{n}}$ satisfies

$$\|f - \hat{f}\|_{\tilde{H}^1(D)} \lesssim \|f - f_\Lambda\|_{\tilde{H}^1(D)} + \sigma_{s,L}(\boldsymbol{c}_\Lambda)_{1,\boldsymbol{v}} + \sqrt{K(s)}\eta,$$

with probability at least $1-\epsilon$, where $\sigma_{s,L}(\cdot)_{1,\boldsymbol{v}}$ is as in (3.4.4) and $v_{\boldsymbol{n}} = \sqrt{1+\lambda_{\boldsymbol{n}}}u_{\boldsymbol{n}}$, $\boldsymbol{n} \in \mathbb{N}_0^d$.

To prove this corollary, we first need to show that $\kappa_{\boldsymbol{n}} \lesssim \lambda_{\boldsymbol{n}}$, $\forall \boldsymbol{n} \in \mathbb{N}_0^d$, for the Jacobi polynomials. Then, the sample complexity is determined up to the magnitude of $K(s)$, which depends on the parameters α, β of the Jacobi polynomials. We will show the full proof of Corollary 3.7.3 in §3.8.4. Moreover, for certain values of α and β, we have the following results (see [93]):

Theorem 3.7.4. *Let $K(s) = K^{(\alpha,\beta)}(s)$ be as in (3.7.8). Then the following hold:*

(i) if $\alpha, \beta \in \mathbb{N}_0$ then $K(s) \leq s^{2\max\{\alpha,\beta\}+2}$,

(ii) if $\beta = \alpha$ and $2\alpha + 1 \in \mathbb{N}$ then $K(s) \leq s^{2\alpha+2}$,

(iii) if $\alpha = \beta = -1/2$ then $K(s) \leq s^{\log(3)/\log(2)}$.

In particular, $K(s) \leq s^2$ for Legendre polynomials ($\alpha = \beta = 0$) and $K(s) \leq s^{\log(3)/\log(2)}$ for Chebyshev polynomials ($\alpha = \beta = -1/2$).

This result implies that, for values of α, β satisfying Theorem 3.7.4, the sample complexity shown in Corollary 3.7.3 reduces to an estimate of the form

$$m \gtrsim s^\gamma \cdot \log(2d) \cdot \log^2(s/\epsilon), \tag{3.7.9}$$

where $\gamma \geq 1$ depends on the parameters α and β. In other words, now the sample complexity is only a polynomial in s and logarithmic in the dimension d. Hence, the curse of dimensionality is mitigated to a substantial extent.

3.7.3 Legendre polynomials and preconditioning

In the previous subsection, we considered the scenario when sampling density is the same as the orthogonality density ν. However, in practice, we are often in the situation that the sampling density μ is different from the orthogonality density ν. In particular, the case where $\phi_{\boldsymbol{n}}$ are the Legendre polynomials and μ is the Chebyshev density has been studied in [2, 107, 128]. This case is often referred to as *preconditioning* in the literature. For this case, we have the following recovery guarantee:

Corollary 3.7.5. *Let μ be the tensor Chebyshev density, ν be the uniform density, $s \geq 2$, $\Lambda = \Lambda_s^{\mathrm{HC}}$ be the hyperbolic cross index set defined in (3.4.3), $0 < \epsilon < 1$, $\boldsymbol{u}$ be the weights defined in (3.7.1) and $f = \sum_{\boldsymbol{n} \in \mathbb{N}_0^d} c_{\boldsymbol{n}} \phi_{\boldsymbol{n}} \in \tilde{H}^1(D)$, where $D = (-1,1)^d$ and $\tilde{H}^1(D)$ is as in (3.5.6). Suppose that*

$$m \gtrsim \min\left\{2^d s, (\pi/2)^d s^{\log(1+4/\pi)/\log(2)}\right\} \cdot (d + \log(s)) \cdot (d + \log(s/\epsilon)). \tag{3.7.10}$$

Draw $\boldsymbol{y}_1, \dots, \boldsymbol{y}_m$ independently according to the density μ, let $\boldsymbol{A}$, $\boldsymbol{f}$ and η be as in (3.5.13), (3.5.10) and (3.5.11) respectively and set $\boldsymbol{w} = \boldsymbol{u}$. Then, if $\hat{\boldsymbol{z}}$ is any minimizer of (3.5.14) and $\hat{\boldsymbol{c}} = \boldsymbol{Q}^{-1}\hat{\boldsymbol{z}}$, the approximation $\hat{f} = \sum_{\boldsymbol{n} \in \Lambda} \hat{c}_{\boldsymbol{n}} \phi_{\boldsymbol{n}}$ satisfies

$$\|f - \hat{f}\|_{\tilde{H}^1(D)} \lesssim \|f - f_\Lambda\|_{\tilde{H}^1(D)} + \sigma_{s,L}(\boldsymbol{c}_\Lambda)_{1,\boldsymbol{v}} + \sqrt{K(s)}\eta,$$

with probability at least $1-\epsilon$, where $\sigma_{s,L}(\cdot)_{1,\boldsymbol{v}}$ is as in (3.4.4) and $v_{\boldsymbol{n}} = \sqrt{1+\lambda_{\boldsymbol{n}}}u_{\boldsymbol{n}}$, $\boldsymbol{n} \in \mathbb{N}_0^d$.

3.7.4 Discussion

The unaugmented version of Corollary 3.7.3 has been presented in [2]. Using the same set-up and notation, it has been proved in Theorem 6.1 of [2] that if

$$m \gtrsim K(s) \cdot \log(\epsilon^{-1}) \cdot L, \qquad L = \left(\min\{\log(2s) + d, \log(2d)\log(2s)\} + \log(K(s))\right), \tag{3.7.11}$$

where $K(s)$ is as in (3.7.6), then the approximation error satisfies

$$\|f - \hat{f}\|_{L^2(D)} \lesssim \sigma_{s,L}(\boldsymbol{c}_\Lambda)_{1,\boldsymbol{u}} + \|f - f_\Lambda\|_{L^2(D)} + \lambda\sqrt{K(s)}\eta, \tag{3.7.12}$$

with high probability, where $\lambda = 1 + \sqrt{\log(\epsilon^{-1})}/L$. It can be clearly seen the sample complexity (3.7.7) shown in Corollary 3.7.3 and the sample complexity (3.7.10) shown in Corollary 3.7.5 are identical to (3.7.11), up to a slightly increased log factor, which is due to the fact that we have used a slightly different method of proof (see §6.4 of [8]) to remove the factor λ in the error bound. In particular, as shown in Theorem 3.4.2, (3.7.11) reduces to be

$$m \gtrsim s^\gamma \log(\epsilon^{-1}) \min\{d + \log(s), \log(2d)\log(s)\}, \tag{3.7.13}$$

in the case of Legendre and Chebyshev polynomials, which is similar to the sample complexity shown as in (3.7.9). Thus, we can conclude that, for the same sample complexity, the gradient-augmented problem permit an error bounded in the stronger Sobolev norm, as opposed to an $L^2(D)$ norm for the unaugmented problem shown as in (3.7.12).

Recall, as it has been pointed out at the beginning of this section, similar to those results in [2], all theoretical results presented in the section are nonuniform recovery guarantees, which ensure the recovery of a single f from a random draw of sample points. For the unaugmented case, uniform recovery guarantees for Chebyshev and Legendre polynomials with $\mu = \nu$ have been proved in [4,43]. Compared to nonuniform recovery guarantee for the unaugmented problem, as shown in [4,43], the error bound for uniform recovery guarantee is improved by a factor of $1/\sqrt{K(s)}$ when a similar sample complexity, expect with higher log factor, is satisfied. This is typical for uniform recovery guarantees in compressed sensing. We expect a similar uniform recovery guarantee could be attained for the gradient-augmented problem, but we leave this as future work.

3.8 Proofs

In this section, we will give proofs for the theoretical results presented in §3.7. As a first step, we will show the gradient-augmented problem can be reformulated as an instance of the so-call *parallel acquisition* model, which allows us following the approach of [44] to prove the recovery guarantees. For references on parallel acquisition model in compressed sensing, see [20, 24, 44].

3.8.1 The parallel acquisition model reformulation

Before introducing the parallel acquisition model, it is worth mentioning that, different from the standard compressed sensing model, the parallel acquisition model considers the scenario that there are multiple sensors acting in parallel and simultaneously acquiring measurements of a single vector. As pointed out in Chapter 2, parallel Magnetic Resonance Imaging (MRI) can be modelled by the parallel acquisition model. Now we follow the framework described in [44, §II-D] to define the abstract parallel acquisition model,

For some $\mathcal{D} \in \mathbb{N}$, let F be a distribution on a set of $N \times \mathcal{D}$ complex matrices. We assume that F is isotropic in the sense that

$$\mathbb{E}(\boldsymbol{B}\boldsymbol{B}^*) = I, \qquad \boldsymbol{B} \sim F. \tag{3.8.1}$$

Let $\{\boldsymbol{e}_i\}_{i=1}^m$ be the canonical basis of $\mathbb{C}^m$ and let $\boldsymbol{B}_1, \dots, \boldsymbol{B}_m$ be a sequence of independent realizations of matrices from the distribution F. Then we define the sampling matrix

$$\boldsymbol{A} = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} \boldsymbol{e}_i \otimes \boldsymbol{B}_i^* = \frac{1}{\sqrt{m}} \begin{bmatrix} \boldsymbol{B}_1^* \\ \vdots \\ \boldsymbol{B}_m^* \end{bmatrix} \in \mathbb{C}^{\mathcal{D}m \times N}, \tag{3.8.2}$$

where $\otimes$ denoted the Kronecker product. Note that this is an extension of the standard compressed sensing setup, which corresponds to the case $\mathcal{D} = 1$, i.e. matrix $\boldsymbol{A}$ has independent rows. The paper [44] considered solving this parallel acquisition model with a compressed sensing set-up by using the ℓ^1 minimization technique and proved a series of nonuniform recovery guarantees. In this section, we consider to extend the work presented in [44] by considering the weighted ℓ^1 minimization problem

$$\min_{\boldsymbol{z} \in \mathbb{C}^N} \|\boldsymbol{z}\|_{1,\boldsymbol{w}} \text{ subject to } \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{f}\|_2 \leq \eta, \tag{3.8.3}$$

where $\boldsymbol{w} = (w_j)_{j=1}^N \in \mathbb{R}^N$ with $w_j \geq 1$, $\forall j$. Here, $\boldsymbol{f} = \boldsymbol{A}\boldsymbol{c} + \boldsymbol{e}$ are noisy measurements of the unknown vector $\boldsymbol{c}$ (for ease of notation we write this rather than $\boldsymbol{c}_\Lambda$) and $\boldsymbol{e}$ is a vector satisfying $\|\boldsymbol{e}\|_2 \leq \eta$.

Consider the set-up of §3.5.3. We define the random matrix $\boldsymbol{B}$ by

$$\boldsymbol{B} = \left(\sqrt{\tau_k(\boldsymbol{y})} \frac{\overline{\partial_k \phi_{\boldsymbol{n}_j}(\boldsymbol{y})}}{\sqrt{1 + \lambda_{\boldsymbol{n}_j}}} \right)_{j=1,k=0}^{N,d} \in \mathbb{C}^{N \times (d+1)}, \tag{3.8.4}$$

where $\boldsymbol{y}$ is the random variable on D drawn with respect to the probability density μ. This brings about a distribution F on random matrices in $\mathbb{C}^{N \times \mathcal{D}}$, where $\mathcal{D} = (d+1)$. It can be seen that the corresponding matrix (3.8.2) is identical to the matrix defined in (3.5.13) by applying a simple row permutation. Thus, we deduce that the gradient-augmented problem (3.5.14) is an instance of above model, corresponding to choice $\mathcal{D} = (d+1)$ and with F being the distribution of random matrices (3.8.4). In other words, one can think the gradient-augmented problem as a parallel acquisition model where the 1st sensor records the function value and the 2nd to $(d+1)$th sensors record the gradient values.

3.8.2 Parallel acquisition model with weighted ℓ^1 minimization

Now we present the theoretical result for the model described §3.8.1 with the weighted ℓ^1 regularizer (3.8.3) by generalizing the result of [44]. As a first step, we introduce some notation. If $\Delta \subseteq \{1, \dots, N\}$, then we use the notation $\boldsymbol{P}_\Delta$ for the orthogonal projection $\boldsymbol{P}_\Delta \in \mathbb{C}^{N \times N}$ onto $\text{span}\{\boldsymbol{e}_j : j \in \Delta\}$. Note that the vector $\boldsymbol{P}_\Delta \boldsymbol{x} \in \mathbb{C}^N$ is isomorphic to a vector in $\mathbb{C}^{|\Delta|}$. Also, given weights $\boldsymbol{w} \in \mathbb{R}^N$, we write the weighting matrix $\boldsymbol{W} = \text{diag}(\boldsymbol{w})$.

Finally, we note that in this subsection we index over $\mathbb{N}$ where relevant, as opposed to $\mathbb{N}_0^d$ as in the original polynomial approximation problem. Then, as in [44], we can define several notions of local coherence:

Definition 3.8.1. *Let $\Delta \subseteq \{1, \dots, N\}$ and F is a distribution on $\mathbb{C}^{N \times \mathcal{D}}$ satisfying* (3.8.1). *The local coherence of F relative to Δ is the smallest constant $\Upsilon(F, \Delta)$ such that*

$$\|\boldsymbol{P}_\Delta \boldsymbol{B}\boldsymbol{B}^* \boldsymbol{P}_\Delta\|_2 \leq \Upsilon(F, \Delta), \qquad \boldsymbol{B} \sim F,$$

almost surely.

Definition 3.8.2. *Let $\boldsymbol{w} \in \mathbb{R}^N$ be a set of positive weights, $\Delta \subseteq \{1, \dots, N\}$ and F is defined the same as in Definition 3.8.1. The local coherence of F relative to Δ with respect to the weights $\boldsymbol{w}$ is*

$$\Gamma(F, \boldsymbol{w}, \Delta) = \max\left\{\Gamma_1(F, \boldsymbol{w}, \Delta), \Gamma_2(F, \boldsymbol{w}, \Delta)\right\},$$

where $\Gamma_1(F, \boldsymbol{w}, \Delta)$ and $\Gamma_2(F, \boldsymbol{w}, \Delta)$ are the smallest quantities such that

$$\left\|\boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\right\|_\infty \leq \Gamma_1(F, \boldsymbol{w}, \Delta), \qquad \boldsymbol{B} \sim F,$$

almost surely, and

$$\sup_{\|\boldsymbol{z}\|_\infty = 1} \max_{j=1,\dots,N} \mathbb{E}|\langle \boldsymbol{e}_j, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle|^2 \leq \Gamma_2(F, \boldsymbol{w}, \Delta).$$

From Definition 3.8.2, we see that, if $j \in \Delta$, then

$$\Gamma_1(F, \boldsymbol{w}, \Delta) \geq \mathbb{E}|\langle \boldsymbol{e}_j, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{e}_j\rangle| \geq \left|\mathbb{E}\langle \boldsymbol{e}_j, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{e}_j\rangle\right| = |\langle \boldsymbol{e}_j, \boldsymbol{W}^{-1}\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{e}_j\rangle| = 1.$$

Hence we deduce that $\Gamma_1(F, \boldsymbol{w}, \Delta) \geq 1$. Similarly, for $j \in \Delta$, we also have

$$\begin{aligned}\Gamma_2(F, \boldsymbol{w}, \Delta) &\geq \mathbb{E}|\langle \boldsymbol{e}_j, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{e}_j\rangle|^2 \geq \left|\mathbb{E}\langle \boldsymbol{e}_j, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{e}_j\rangle\right|^2 \\ &= |\langle \boldsymbol{e}_j, \boldsymbol{W}^{-1}\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{e}_j\rangle|^2 = 1,\end{aligned}$$

and the same for the unweighted local coherence

$$\Upsilon(F, \Delta) \geq \mathbb{E}\,|\langle e_j, \boldsymbol{P}_\Delta \boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta e_j\rangle| \geq |\mathbb{E}\langle e_j, \boldsymbol{P}_\Delta \boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta e_j\rangle| = 1.$$

The following is the main result for the abstract model shown in §3.8.1:

Theorem 3.8.3. *Let $0 < \epsilon < 1$, $\eta \geq 0$, $N \geq 2$, $\Delta \subseteq \{1, \dots, N\}$ with $|\Delta| \geq 2$ and $\boldsymbol{w} \in \mathbb{R}^N$ be weights with $w_j \geq 1$, $\forall j$. Fix $\boldsymbol{c} \in \mathbb{C}^N$ and construct $\boldsymbol{A} \in \mathbb{C}^{m\mathcal{D} \times N}$ as in (3.8.2). Let*

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{c} + \boldsymbol{e}$, *where* $\|\boldsymbol{e}\|_2 \leq \eta$. *If*

$$m \gtrsim \Upsilon(F, \Delta) \cdot \log(N/\epsilon) + \Gamma(F, \boldsymbol{w}, \Delta) \cdot (\log(N/\epsilon) + \log(|\Delta|_{\boldsymbol{w}}) \cdot \log(|\Delta|_{\boldsymbol{w}}/\epsilon)),$$

where $\Upsilon(F, \Delta)$ *and* $\Gamma(F, \boldsymbol{w}, \Delta)$ *are as in Definitions 3.8.1 and 3.8.2 respectively, then, with probability at least* $1 - \epsilon$, *any minimizer* $\hat{\boldsymbol{c}}$ *of (3.8.3) satisfies*

$$\|\boldsymbol{c} - \hat{\boldsymbol{c}}\|_2 \lesssim \|\boldsymbol{c} - P_\Delta \boldsymbol{c}\|_{1,\boldsymbol{w}} + \sqrt{|\Delta|_{\boldsymbol{w}}}\eta.$$

We omit the proof of Theorem 3.8.3 here. For the full proof, see §6.4 of [8]. Note that, with $\mathcal{D} = 1$, we can get the following corollary by estimating $\Upsilon(F, \Delta)$ and $\Gamma(F, \boldsymbol{w}, \Delta)$.

Corollary 3.8.4. *Let* $\Lambda \in \mathbb{N}_0^d$ *with* $|\Lambda| = N \geq 2$, $0 < \epsilon < 1$, $\Delta \subset \Lambda$ *with* $|\Delta| \geq 2$, $\boldsymbol{w} \in \mathbb{C}^N$ *be a vector of weights with* $w_{\boldsymbol{n}} \geq 1$, $\forall \boldsymbol{n}$, *and* $f = \sum_{\boldsymbol{n} \in \mathbb{N}_0^d} c_{\boldsymbol{n}} \phi_{\boldsymbol{n}} \in L^2_\nu(D)$, *where* $D = (-1, 1)^d$. *Let*

$$m \gtrsim \left(|\Delta|_{\boldsymbol{u}} + \max_{\boldsymbol{n} \in \Lambda} \left\{ \frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2} \right\} |\Delta|_{\boldsymbol{w}} \right) \cdot (\log(N/\epsilon) + \log(|\Delta|_{\boldsymbol{w}}) \cdot \log(|\Delta|_{\boldsymbol{w}}/\epsilon)), \tag{3.8.5}$$

Draw $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ *independently according to the density* μ, *let* $\boldsymbol{A}$, $\boldsymbol{f}$ *and* η *be as in* (3.3.6), (3.3.8) *and* (3.3.9) *respectively. Then, if* $\hat{\boldsymbol{c}}$ *is any minimizer of* (3.3.10) *satisfies*

$$\|\boldsymbol{c}_\Lambda - \hat{\boldsymbol{c}}\|_2 \lesssim \|\boldsymbol{c}_\Lambda - \boldsymbol{c}_\Delta\|_{1,\boldsymbol{u}} + \sqrt{|\Delta|_{\boldsymbol{u}}}\eta.$$

Proof. Let $\boldsymbol{z} \in \mathbb{C}^N$ with $\|\boldsymbol{z}\|_2 = 1$. $\boldsymbol{B} = \left(\sqrt{\frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})}}\, \overline{\phi_{\boldsymbol{n}_j}(\boldsymbol{y})} \right)_{j=1}^N$ is an $N \times 1$ vector. Then

$$\|\boldsymbol{B}^* \boldsymbol{P}_\Delta \boldsymbol{z}\|_2^2 = \left| \sum_{\boldsymbol{n} \in \Delta} \sqrt{\frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})}}\, \phi_{\boldsymbol{n}}(\boldsymbol{y}) z_{\boldsymbol{n}} \right|^2 \leq \sum_{\boldsymbol{n} \in \Delta} \frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})} |\phi_{\boldsymbol{n}}(\boldsymbol{y})|^2 \leq |\Delta|_{\boldsymbol{u}}.$$

Since $\boldsymbol{z}$ was arbitrary, we have $\Upsilon(F, \Delta) \leq |\Delta|_{\boldsymbol{u}}$.

Now let $\boldsymbol{z} \in \mathbb{C}^N$ with $\|\boldsymbol{z}\|_\infty = 1$ and $\boldsymbol{n'} \in \Lambda$. Then

$$\left| \langle \boldsymbol{e}_{\boldsymbol{n'}}, \boldsymbol{W}^{-1} \boldsymbol{B} \boldsymbol{B}^* \boldsymbol{P}_\Delta \boldsymbol{W} \boldsymbol{z} \rangle \right| = \frac{1}{w_{\boldsymbol{n'}}} \left| \sqrt{\frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})}} \phi_{\boldsymbol{n'}}(\boldsymbol{y}) \sum_{\boldsymbol{n} \in \Delta} \sqrt{\frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})}}\, \phi_{\boldsymbol{n}}(\boldsymbol{y}) w_{\boldsymbol{n}} z_{\boldsymbol{n}} \right|. \tag{3.8.6}$$

With Cauchy-Schwarz inequality, we get

$$\begin{aligned}
\left|\langle \boldsymbol{e_{n'}}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right| &\leq \frac{1}{w_{\boldsymbol{n'}}}\sqrt{\frac{\nu(\boldsymbol{y})|\phi_{\boldsymbol{n'}}(\boldsymbol{y})|^2}{\mu(\boldsymbol{y})}}\sum_{\boldsymbol{n}\in\Delta}\sqrt{\frac{\nu(\boldsymbol{y})|\phi_{\boldsymbol{n}}(\boldsymbol{y})|^2}{\mu(\boldsymbol{y})}}w_{\boldsymbol{n}} \\
&\leq \frac{1}{w_{\boldsymbol{n'}}}\sqrt{\frac{\nu(\boldsymbol{y})|\phi_{\boldsymbol{n'}}(\boldsymbol{y})|^2}{\mu(\boldsymbol{y})}}\sqrt{\sum_{\boldsymbol{n}\in\Delta}\frac{\nu(\boldsymbol{y})|\phi_{\boldsymbol{n}}(\boldsymbol{y})|^2}{\mu(\boldsymbol{y})}}\sqrt{|\Delta|_{\boldsymbol{w}}}. \\
&\leq \frac{u_{\boldsymbol{n'}}}{w_{\boldsymbol{n'}}}\sqrt{\sum_{\boldsymbol{n}\in\Delta}u_{\boldsymbol{n}}^2}\sqrt{|\Delta|_{\boldsymbol{w}}} \\
&\leq \sqrt{\max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2}\right\}}\sqrt{|\Delta|_{\boldsymbol{u}}|\Delta|_{\boldsymbol{w}}}.
\end{aligned}$$

Since $\boldsymbol{z}$ and $\boldsymbol{n'}$ were arbitrary, after applying the inequality $ab \leq a^2/2 + b^2/2$, we obtain

$$\Gamma_1(F, \boldsymbol{w}, \Delta) \leq \frac{1}{2}|\Delta|_{\boldsymbol{u}} + \frac{1}{2}\max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2}\right\}|\Delta|_{\boldsymbol{w}}.$$

We now consider $\Gamma_2(F, \boldsymbol{w}, \Delta)$. From (3.8.6), we have

$$\begin{aligned}
\mathbb{E}\left|\langle \boldsymbol{e_{n'}}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right|^2 &\leq \frac{1}{w_{\boldsymbol{n'}}^2}\mathbb{E}\left(\frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})}|\phi_{\boldsymbol{n'}}(\boldsymbol{y})|^2\left(\left|\sum_{\boldsymbol{n}\in\Delta}\sqrt{\frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})}}\,\phi_{\boldsymbol{n}}(\boldsymbol{y})w_{\boldsymbol{n}}z_{\boldsymbol{n}}\right|^2\right)\right) \\
&\leq \frac{u_{\boldsymbol{n'}}^2}{w_{\boldsymbol{n'}}^2}\mathbb{E}\left(\frac{\nu(\boldsymbol{y})}{\mu(\boldsymbol{y})}\left|\sum_{\boldsymbol{n}\in\Delta}\phi_{\boldsymbol{n}}(\boldsymbol{y})w_{\boldsymbol{n}}z_{\boldsymbol{n}}\right|^2\right) \\
&= \frac{u_{\boldsymbol{n'}}^2}{w_{\boldsymbol{n'}}^2}\int_D\left|\sum_{\boldsymbol{n}\in\Delta}\phi_{\boldsymbol{n}}(\boldsymbol{y})w_{\boldsymbol{n}}z_{\boldsymbol{n}}\right|^2\nu(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}.
\end{aligned}$$

Recall the functions $\phi_{\boldsymbol{n}}$ are orthonormal with respect to the weight function ν. Thus, by Parseval's identity, we get

$$\mathbb{E}\left|\langle \boldsymbol{e_{n'}}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right|^2 \leq \frac{u_{\boldsymbol{n'}}^2}{w_{\boldsymbol{n'}}^2}\sum_{\boldsymbol{n}\in\Delta}|w_{\boldsymbol{n}}z_{\boldsymbol{n}}|^2 \leq \max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2}\right\}|\Delta|_{\boldsymbol{w}},$$

Since $\boldsymbol{z}$ and $\boldsymbol{n'}$ were arbitrary, we deduce that

$$\Gamma_2(F, \boldsymbol{w}, \Delta) \leq \max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2}\right\}|\Delta|_{\boldsymbol{w}}.$$

Thus, we have

$$\Gamma(F, \boldsymbol{w}, \Delta) \leq |\Delta|_{\boldsymbol{u}} + \max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2}{w_{\boldsymbol{n}}^2}\right\}|\Delta|_{\boldsymbol{w}}.$$

Applying the bound for $\Upsilon(F,\Delta)$ and the bound for $\Gamma(F,\boldsymbol{w},\Delta)$ to Theorem 3.8.3 completes the proof. □

Note that, if taking $\phi_{\boldsymbol{n}}$ as Legendre or Chebyshev polynomials and $\boldsymbol{w} = \boldsymbol{u}$ with the lower sets assumption, the sample complexity (3.8.5) becomes

$$m \gtrsim s^{\gamma} \cdot \log(2d) \cdot \log^2(s/\epsilon),$$

which is identical to the sample complexity obtained in Theorem 3.4.2, up to a minor change in the log factor.

3.8.3 Proofs of Theorem 3.7.1 and Corollary 3.7.2

Now we shall prove Theorem 3.7.1 and Corollary 3.7.2. It can be seen that Theorem 3.7.1 follows as a corollary of Theorem 3.8.3, after estimating the local coherences $\Upsilon(F,\Delta)$ and $\Gamma(F,\boldsymbol{w},\Delta)$ for the gradient-augmented problem. The estimations of the local coherences are done in the following two lemmas. Note that we now revert back to indexing over the multi-index set $\Lambda \subset \mathbb{N}_0^d$ (as was introduced in §3.2), rather than over the integers $\{1,\dots,N\}$.

Lemma 3.8.5. *Let $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ be the orthonormal basis of tensor-product Sturm–Louiville eigenfunctions defined in §3.5.1, F be the distribution of matrices defined in §3.8.1 for the gradient-augmented problem, and suppose that $\Upsilon(F,\Delta)$ is as in Definition 3.8.1, where $\Delta \subset \mathbb{N}_0^d$ is a multi-index set. Then*

$$\Upsilon(F,\Delta) \leq \max_{\boldsymbol{n}\in\Delta} \left\{ \frac{1+\kappa_{\boldsymbol{n}}}{1+\lambda_{\boldsymbol{n}}} \right\} |\Delta|_{\boldsymbol{u}},$$

where $\lambda_{\boldsymbol{n}}$, $\kappa_{\boldsymbol{n}}$ and $\boldsymbol{u}$ are as in (3.5.7), (3.7.3) and (3.7.1) respectively

Proof. Let $\boldsymbol{z} \in \mathbb{C}^N$ with $\|\boldsymbol{z}\|_2 = 1$ and let B be as in (3.8.4). Then

$$\|\boldsymbol{B}^*\boldsymbol{P}_{\Delta}\boldsymbol{z}\|_2^2 = \sum_{k=0}^{d} \left| \sum_{\boldsymbol{n}\in\Delta} \frac{\sqrt{\tau_k(\boldsymbol{y})}\ \partial_k\phi_{\boldsymbol{n}}(\boldsymbol{y}) z_{\boldsymbol{n}}}{\sqrt{1+\lambda_{\boldsymbol{n}}}} \right|^2 \leq \sum_{k=0}^{d} \sum_{\boldsymbol{n}\in\Delta} \frac{\tau_k(\boldsymbol{y})\,|\partial_k\phi_{\boldsymbol{n}}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n}}},$$

by Cauchy-Schwarz inequality. Observe that, when $k \neq 0$,

$$\begin{aligned}\tau_k(\boldsymbol{y})\,|\partial_k\phi_{\boldsymbol{n}}(\boldsymbol{y})|^2 &= \frac{\chi(y_k)\prod_{j=1,j\neq k}^{d}\nu(y_j)}{\mu(\boldsymbol{y})} |\phi'_{n_k}(y_k)|^2 \prod_{\substack{j=1\\ j\neq k}}^{d} |\phi_{n_j}(y_j)|^2 \\ &\leq \frac{\chi(y_k)}{\mu(y_k)} |\phi'_{n_k}(y_k)|^2 \prod_{\substack{j=1\\ j\neq k}}^{d} u_{n_j}^2 \leq \kappa_{n_k} u_{\boldsymbol{n}}^2,\end{aligned}$$

and therefore

$$\sum_{k=0}^{d} \tau_k(\boldsymbol{y}) \left|\partial_k \phi_{\boldsymbol{n}}(\boldsymbol{y})\right|^2 \leq u_{\boldsymbol{n}}^2 \left(1 + \sum_{k=1}^{d} \kappa_{n_k}\right) = u_{\boldsymbol{n}}^2 (1 + \kappa_{\boldsymbol{n}}). \tag{3.8.7}$$

Hence

$$\|\boldsymbol{B}^* \boldsymbol{P}_\Delta \boldsymbol{z}\|_2^2 \leq \sum_{\boldsymbol{n}\in\Delta} \frac{1+\kappa_{\boldsymbol{n}}}{1+\lambda_{\boldsymbol{n}}} u_{\boldsymbol{n}}^2 \leq \max_{\boldsymbol{n}\in\Delta} \left\{\frac{1+\kappa_{\boldsymbol{n}}}{1+\lambda_{\boldsymbol{n}}}\right\} |\Delta|_{\boldsymbol{u}}.$$

Since $\boldsymbol{z}$ was arbitrary, we deduce the result. □

Lemma 3.8.6. *Let $\{\phi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathbb{N}_0^d}$ and F be as in Lemma 3.8.5 and $\Gamma(F, \boldsymbol{w}, \Delta)$ be as in Definition 3.8.2. Then*

$$\Gamma(F, \boldsymbol{w}, \Delta) \leq |\Delta|_{\boldsymbol{u}} + \max_{\boldsymbol{n}\in\Lambda} \left\{\frac{u_{\boldsymbol{n}}^2(1+\kappa_{\boldsymbol{n}})}{w_{\boldsymbol{n}}^2(1+\lambda_{\boldsymbol{n}})}\right\} |\Delta|_{\boldsymbol{w}}.$$

Proof. Let $\boldsymbol{z} \in \mathbb{C}^N$ with $\|\boldsymbol{z}\|_\infty = 1$ and $\boldsymbol{n'} \in \Lambda$. Then

$$\left|\langle \boldsymbol{e}_{\boldsymbol{n'}}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right| = \frac{1}{w_{\boldsymbol{n'}}} \left|\sum_{k=0}^{d} \frac{\sqrt{\tau_k(\boldsymbol{y})}\partial_k \phi_{\boldsymbol{n'}}(\boldsymbol{y})}{\sqrt{1+\lambda_{\boldsymbol{n'}}}} \sum_{\boldsymbol{n}\in\Delta} \frac{\sqrt{\tau_k(\boldsymbol{y})}\ \partial_k \phi_{\boldsymbol{n}}(\boldsymbol{y})}{\sqrt{1+\lambda_{\boldsymbol{n}}}} w_{\boldsymbol{n}} z_{\boldsymbol{n}}\right|. \tag{3.8.8}$$

Hence, by Cauchy-Schwarz inequality,

$$\begin{aligned}
\left|\langle \boldsymbol{e}_{\boldsymbol{n'}}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right| &\leq \frac{1}{w_{\boldsymbol{n'}}} \sum_{k=0}^{d} \sqrt{\frac{\tau_k(\boldsymbol{y})|\partial_k \phi_{\boldsymbol{n'}}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n'}}}} \sum_{\boldsymbol{n}\in\Delta} \sqrt{\frac{\tau_k(\boldsymbol{y})|\partial_k \phi_{\boldsymbol{n}}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n}}}} w_{\boldsymbol{n}} \\
&\leq \frac{1}{w_{\boldsymbol{n'}}} \sqrt{\sum_{k=0}^{d} \frac{\tau_k(\boldsymbol{y})|\partial_k \phi_{\boldsymbol{n'}}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n'}}}} \sqrt{\sum_{k=0}^{d} \left(\sum_{\boldsymbol{n}\in\Delta} \sqrt{\frac{\tau_k(\boldsymbol{y})|\partial_k \phi_{\boldsymbol{n}}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n}}}} w_{\boldsymbol{n}}\right)^2} \\
&\leq \frac{1}{w_{\boldsymbol{n'}}} \sqrt{\sum_{k=0}^{d} \frac{\tau_k(\boldsymbol{y})|\partial_k \phi_{\boldsymbol{n'}}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n'}}}} \sqrt{\sum_{\boldsymbol{n}\in\Delta} \frac{\sum_{k=0}^{d} \tau_k(\boldsymbol{y})|\partial_k \phi_{\boldsymbol{n}}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n}}}} \sqrt{|\Delta|_{\boldsymbol{w}}}.
\end{aligned}$$

We now apply (3.8.7) to get

$$\begin{aligned}
\left|\langle \boldsymbol{e}_{\boldsymbol{n'}}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right| &\leq \frac{u_{\boldsymbol{n'}}}{w_{\boldsymbol{n'}}} \sqrt{\frac{1+\kappa_{\boldsymbol{n'}}}{1+\lambda_{\boldsymbol{n'}}}} \sqrt{\sum_{\boldsymbol{n}\in\Delta} \frac{1+\kappa_{\boldsymbol{n}}}{1+\lambda_{\boldsymbol{n}}} u_{\boldsymbol{n}}^2} \sqrt{|\Delta|_{\boldsymbol{w}}} \\
&\leq \sqrt{\max_{\boldsymbol{n}\in\Lambda} \left\{\frac{u_{\boldsymbol{n}}^2(1+\kappa_{\boldsymbol{n}})}{w_{\boldsymbol{n}}^2(1+\lambda_{\boldsymbol{n}})}\right\}} \sqrt{|\Delta|_{\boldsymbol{u}}|\Delta|_{\boldsymbol{w}}}.
\end{aligned}$$

Since $\boldsymbol{z}$ and $\boldsymbol{n'}$ were arbitrary, after an application of the inequality $ab \leq a^2/2 + b^2/2$, we obtain

$$\Gamma_1(F, \boldsymbol{w}, \Delta) \leq \frac{1}{2}|\Delta|_{\boldsymbol{u}} + \frac{1}{2} \max_{\boldsymbol{n}\in\Lambda} \left\{\frac{u_{\boldsymbol{n}}^2(1+\kappa_{\boldsymbol{n}})}{w_{\boldsymbol{n}}^2(1+\lambda_{\boldsymbol{n}})}\right\} |\Delta|_{\boldsymbol{w}}. \tag{3.8.9}$$

We now consider $\Gamma_2(F, \boldsymbol{w}, \Delta)$. From (3.8.8) and (3.8.7) we have

$$\begin{aligned}\mathbb{E}\left|\langle \boldsymbol{e}_{\boldsymbol{n}'}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right|^2 &\leq \frac{1}{w_{\boldsymbol{n}'}^2}\mathbb{E}\left(\sum_{k=0}^d \frac{\tau_k(\boldsymbol{y})|\partial_k\phi_{\boldsymbol{n}'}(\boldsymbol{y})|^2}{1+\lambda_{\boldsymbol{n}'}}\left(\sum_{k=0}^d \left|\sum_{\boldsymbol{n}\in\Delta}\frac{\sqrt{\tau_k(\boldsymbol{y})}\partial_k\phi_{\boldsymbol{n}}(\boldsymbol{y})}{\sqrt{1+\lambda_{\boldsymbol{n}}}}w_{\boldsymbol{n}}z_{\boldsymbol{n}}\right|^2\right)\right)\\ &\leq \frac{u_{\boldsymbol{n}'}^2(1+\kappa_{\boldsymbol{n}'})}{w_{\boldsymbol{n}'}^2(1+\lambda_{\boldsymbol{n}'})}\mathbb{E}\sum_{k=0}^d \tau_k(\boldsymbol{y})\left|\sum_{\boldsymbol{n}\in\Delta}\frac{\partial_k\phi_{\boldsymbol{n}}(\boldsymbol{y})}{\sqrt{1+\lambda_{\boldsymbol{n}}}}w_{\boldsymbol{n}}z_{\boldsymbol{n}}\right|^2\\ &= \frac{u_{\boldsymbol{n}'}^2(1+\kappa_{\boldsymbol{n}'})}{w_{\boldsymbol{n}'}^2(1+\lambda_{\boldsymbol{n}'})}\sum_{k=0}^d\int_D\left|\sum_{\boldsymbol{n}\in\Delta}\frac{\partial_k\phi_{\boldsymbol{n}}(\boldsymbol{y})}{\sqrt{1+\lambda_{\boldsymbol{n}}}}w_{\boldsymbol{n}}z_{\boldsymbol{n}}\right|^2\nu_k(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}.\end{aligned}$$

Recall that the functions $\partial_k\phi_{\boldsymbol{n}}$ are orthogonal with respect to the weight function ν_k, and that $\int_D |\partial_k\phi_{\boldsymbol{n}}(\boldsymbol{y})|^2\nu_k(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} = \lambda_{n_k}$. Therefore, by Parseval's identity, we get

$$\begin{aligned}\mathbb{E}\left|\langle \boldsymbol{e}_{\boldsymbol{n}'}, \boldsymbol{W}^{-1}\boldsymbol{B}\boldsymbol{B}^*\boldsymbol{P}_\Delta \boldsymbol{W}\boldsymbol{z}\rangle\right|^2 &\leq \frac{u_{\boldsymbol{n}'}^2(1+\kappa_{\boldsymbol{n}'})}{w_{\boldsymbol{n}'}^2(1+\lambda_{\boldsymbol{n}'})}\sum_{k=0}^d\sum_{\boldsymbol{n}\in\Delta}\frac{|w_{\boldsymbol{n}}z_{\boldsymbol{n}}|^2\lambda_{n_k}}{1+\lambda_{\boldsymbol{n}}}\\ &= \frac{u_{\boldsymbol{n}'}^2(1+\kappa_{\boldsymbol{n}'})}{w_{\boldsymbol{n}'}^2(1+\lambda_{\boldsymbol{n}'})}\sum_{\boldsymbol{n}\in\Delta}|w_{\boldsymbol{n}}z_{\boldsymbol{n}}|^2 \leq \max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2(1+\kappa_{\boldsymbol{n}})}{w_{\boldsymbol{n}}^2(1+\lambda_{\boldsymbol{n}})}\right\}|\Delta|_{\boldsymbol{w}},\end{aligned}$$

where in the last step we recall that $\|\boldsymbol{z}\|_\infty = 1$. Since $\boldsymbol{z}$ and $\boldsymbol{n}'$ were arbitrary, we deduce that

$$\Gamma_2(F, \boldsymbol{w}, \Delta) \leq \max_{\boldsymbol{n}\in\Lambda}\left\{\frac{u_{\boldsymbol{n}}^2(1+\kappa_{\boldsymbol{n}})}{w_{\boldsymbol{n}}^2(1+\lambda_{\boldsymbol{n}})}\right\}|\Delta|_{\boldsymbol{w}}.$$

Combining this with (3.8.9) now completes the proof. □

Proof of Theorem 3.7.1. We now apply the previous two lemmas to the sample complexity shown in Theorem 3.8.3 to get the sample complexity (3.7.4). Then, this sample complexity leads to an error bound $\|\boldsymbol{z}_\Lambda - \hat{\boldsymbol{z}}\|_2 \lesssim \|\boldsymbol{z}_\Lambda - \boldsymbol{z}_\Delta\|_{1,\boldsymbol{w}} + \sqrt{|\Delta|_{\boldsymbol{w}}}\eta$. Recall that $\boldsymbol{z}_\Lambda = \boldsymbol{Q}\boldsymbol{c}_\Lambda$, $\hat{\boldsymbol{z}} = \boldsymbol{Q}\hat{\boldsymbol{c}}$ and $\hat{f} = \sum_{\boldsymbol{n}\in\Lambda} c_{\boldsymbol{n}}\phi_{\boldsymbol{n}}$. Hence

$$\|f_\Lambda - \hat{f}\|_{\tilde{H}^1(D)} = \|\boldsymbol{Q}(\boldsymbol{c}_\Lambda - \hat{\boldsymbol{c}})\|_2 = \|\boldsymbol{z}_\Lambda - \hat{\boldsymbol{z}}\|_2 \lesssim \|\boldsymbol{c}_\Lambda - \boldsymbol{c}_\Delta\|_{1,\boldsymbol{v}} + \sqrt{|\Delta|_{\boldsymbol{w}}}\eta,$$

where $v_{\boldsymbol{n}} = \sqrt{1+\lambda_{\boldsymbol{n}}}w_{\boldsymbol{n}}$, $\boldsymbol{n}\in\mathbb{N}_0^d$. Thus, the error estimate of Theorem 3.7.1 follows from the triangle inequality. □

We may now prove Corollary 3.7.2:

Proof of Corollary 3.7.2. Given $s \geq 1$, let Δ be a lower set with $|\Delta| \leq s$ such that $\|\boldsymbol{c}_\Lambda - \boldsymbol{c}_\Delta\|_{1,\boldsymbol{v}} = \sigma_{s,L}(\boldsymbol{c}_\Lambda)_{1,\boldsymbol{v}}$. Since $\Lambda = \Lambda_s^{\mathrm{HC}}$ is the union of all lower sets of size at most s, we know that $N = |\Lambda_s^{\mathrm{HC}}| \leq \min\left\{2s^3 4^d, \mathrm{e}^2 s^{2+\log_2(d)}\right\}$. See, for example, [4, Eqn. (10)]. Thus, we have $\log(N/\epsilon) \lesssim \min\left\{\log(s/\epsilon) + d, \log(s/\epsilon)\log(2d)\right\}$. We now apply Theorem 3.7.1 with $\boldsymbol{w} = \boldsymbol{u}$, noting that $|\Delta|_{\boldsymbol{u}} \leq K(s)$. □

3.8.4 Proofs of Corollary 3.7.3

In this subsection, we will prove the Corollary 3.7.3. At first, we require some further background on Jacobi polynomials. For $\alpha, \beta > -1$ and $n \in \mathbb{N}_0$, let $P_n^{(\alpha,\beta)}$ be the Jacobi polynomial of degree n. These polynomials are orthogonal on $(-1,1)$ with respect to the weight function $\omega^{(\alpha,\beta)}(y) = (1-y)^\alpha(1+y)^\beta$, and satisfy

$$\langle P_n^{(\alpha,\beta)}, P_m^{(\alpha,\beta)} \rangle_{L^2_{\omega^{(\alpha,\beta)}}} = \kappa_n^{(\alpha,\beta)} \delta_{n,m},$$

where

$$\kappa_n^{(\alpha,\beta)} = \frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1} \frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(n+1)\Gamma(n+\alpha+\beta+1)}.$$

These polynomials are normalized so that $P_n^{(\alpha,\beta)}(1) = \begin{pmatrix} n+\alpha \\ n \end{pmatrix}$. Moreover, if $\alpha, \beta \geq -1/2$, then

$$\sup_{y\in(-1,1)} |P_n^{(\alpha,\beta)}(y)| = \begin{pmatrix} n+q \\ n \end{pmatrix} \sim \frac{n^q}{\Gamma(q+1)}, \quad n \to \infty, \tag{3.8.10}$$

where $q = \max\{\alpha, \beta\}$. See, for example, [111, Thm. 7.32.1]. We also note the reflection property

$$P_n^{(\alpha,\beta)}(y) = (-1)^n P_n^{(\beta,\alpha)}(-y). \tag{3.8.11}$$

Let $c^{(\alpha,\beta)} = \int_{-1}^1 \omega^{(\alpha,\beta)}(y)\,\mathrm{d}y$, and define the probability density function $\nu^{(\alpha,\beta)}(y) = \frac{\omega^{(\alpha,\beta)}(y)}{c^{(\alpha,\beta)}}$. Then the corresponding orthonormal polynomials with respect to this density are given by

$$\phi_n(y) = \frac{P_n^{(\alpha,\beta)}(y)}{\sqrt{\kappa_n^{(\alpha,\beta)} c^{(\alpha,\beta)}}}, \qquad n \in \mathbb{N}_0. \tag{3.8.12}$$

Proof of Corollary 3.7.3. In view of Corollary 3.7.2, it suffices to show that $\kappa_{\boldsymbol{n}} \lesssim \lambda_{\boldsymbol{n}}$, $\forall \boldsymbol{n} \in \mathbb{N}_0^d$. Since $\lambda_{\boldsymbol{n}} = \sum_{k=1}^d \lambda_{n_k}$ and $\kappa_{\boldsymbol{n}} = \sum_{k=1}^d \kappa_{n_k}$ (see (3.5.7) and (3.7.3) respectively), it is enough to show $\kappa_n \lesssim \lambda_n$, $\forall n \in \mathbb{N}_0$. Using the definition of κ_n (see (3.7.2)) and the fact that $\chi(y) = \frac{1}{c^{(\alpha,\beta)}}(1-y)^{\alpha+1}(1+y)^{\beta+1}$ and $\nu(y) = (1-y)^\alpha(1+y)^\beta/c^{(\alpha,\beta)}$ for the Jacobi polynomials (see (3.5.2)), this is equivalent to

$$\sup_{y\in(-1,1)} \left\{ \sqrt{1-y^2} |\phi_n'(y)| \right\} \lesssim u_n \sqrt{\lambda_n}.$$

Furthermore, using (3.5.3), (3.8.10) and (3.8.12), we see that it is sufficient to show that

$$\sup_{y\in(-1,1)} \left\{ \sqrt{1-y^2} \left| \left(P_n^{(\alpha,\beta)}(y) \right)' \right| \right\} \lesssim n^{1+q}. \tag{3.8.13}$$

Note that from this equation onwards we allow the constant implied by the expression $\lesssim$ to depend on α and β. The derivatives of the Jacobi polynomials satisfy the following bound:

$$\left|\frac{\mathrm{d}P_n^{(\alpha,\beta)}(y)}{\mathrm{d}y}\bigg|_{y=\cos(\theta)}\right| \lesssim \begin{cases} \theta^{-\alpha-3/2}n^{1/2} & cn^{-1} \leq \theta \leq \pi/2 \\ n^{2+\alpha} & 0 \leq \theta \leq cn^{-1} \end{cases}, \tag{3.8.14}$$

(see [111, Thm. 7.32.4]). Using this and the fact that $\sin(\theta) \leq \theta$ for $0 \leq \theta \leq \pi/2$, we deduce that

$$\begin{aligned}\sup_{0\leq y\leq 1}\sqrt{1-y^2}\left|\left(P_n^{(\alpha,\beta)}(y)\right)'\right| &= \sup_{0\leq\theta\leq\pi/2}\sin(\theta)\left|\frac{\mathrm{d}P_n^{(\alpha,\beta)}(y)}{\mathrm{d}y}\bigg|_{y=\cos(\theta)}\right| \\ &\lesssim \max\left\{\sup_{cn^{-1}\leq\theta\leq\pi/2}\theta^{-\alpha-1/2}n^{1/2}, \sup_{0\leq\theta\leq cn^{-1}}n^{2+\alpha}\theta\right\} \lesssim n^{\alpha+1}.\end{aligned}$$

Now suppose $-1 \leq y \leq 0$. Using (3.8.11) and replacing α with β in the above arguments, we deduce that

$$\sup_{-1\leq y\leq 0}\sqrt{1-y^2}\left|\left(P_n^{(\alpha,\beta)}(y)\right)'\right| \lesssim n^{\beta+1},$$

Therefore, (3.8.13) follows immediately. □

3.8.5 Proofs of Corollary 3.7.5

Finally, we shall prove Corollary 3.7.5.

Proof of Corollary 3.7.5. As in the proof of Corollary 3.7.3, we first need to show that $\kappa_n \lesssim \lambda_n,\ \forall n \in \mathbb{N}_0$, which is equivalent to

$$\sup_{y\in(-1,1)}(1-y^2)^{3/4}|\phi_n'(y)| \lesssim u_n\sqrt{\lambda_n}, \tag{3.8.15}$$

where

$$u_n = \sup_{y\in(-1,1)}(\pi/2)^{1/2}\,(1-y^2)^{1/4}|\phi_n(y)|.$$

We first seek a lower bound for u_n. The classical Legendre polynomials $P_n = P_n^{(0,0)}$ satisfy

$$P_n^{(0,0)}(\cos\theta) = 2^{1/2}(\pi n\sin\theta)^{-1/2}\cos\left((n+1/2)\,\theta - \pi/4\right) + \mathcal{O}(n^{-3/2}), \qquad 0 < \theta < \pi.$$

See [111, Thm. 8.21.2]. This formula holds uniformly in the interval $\epsilon \leq \theta \leq \pi - \epsilon$. When n is even, Legendre polynomials have extrema at $\cos(\theta) = 0$, i.e. $\theta = \frac{\pi}{2}$. Then, we have

$$P_n^{(0,0)}(0) = 2^{1/2}(\pi n)^{-1/2}\cos\left(n\pi/2\right) + \mathcal{O}(n^{-3/2}) = (2/\pi)^{1/2}\,n^{-1/2}(-1)^{n/2} + \mathcal{O}(n^{-3/2}).$$

When n is odd, we consider the point $\theta = \frac{\pi}{2} + \epsilon_n$, where $\epsilon_n = \pi/(2n+1)$. Then

$$\begin{aligned} P_n^{(0,0)}\left(\cos\left(\pi/2+\epsilon_n\right)\right) &= 2^{1/2}(\pi n)^{-1/2}(\cos\left(\epsilon_n\right))^{-1/2}\cos\left(n\pi/2+(n+1/2)\,\epsilon_n\right)+\mathcal{O}(n^{-3/2}) \\ &= (2/\pi)^{1/2}n^{-1/2}(-1)^{(n-1)/2}+\mathcal{O}(n^{-3/2}) \end{aligned}$$

Therefore, for both even and odd n, we have

$$\sup_{y\in(-1,1)}(1-y^2)^{1/4}|P_n^{(0,0)}(y)| \gtrsim n^{-1/2},$$

and since $\phi_n(y) = \sqrt{2n+1}P_n^{(0,0)}(y)$, we deduce that $u_n \gtrsim 1$. Since $\lambda_n = n(n+1)$, we then see that (3.8.15) is now implied by

$$\sup_{y\in(-1,1)}(1-y^2)^{3/4}|\phi_n'(y)| \lesssim n. \tag{3.8.16}$$

Using (3.8.14) with $\alpha = \beta = 0$ and arguing as in Corollary 3.7.3, we obtain

$$\sup_{0\le y\le 1}(1-y^2)^{3/4}\left|\frac{\mathrm{d}P_n^{(0,0)}(y)}{\mathrm{d}y}\right| \lesssim \max\left\{\sup_{cn^{-1}\le\theta\le\pi/2} n^{1/2}, \sup_{0\le\theta\le cn^{-1}} n^2\theta^{3/2}\right\} \lesssim n^{1/2}.$$

Using the reflection property and the fact that $\phi_n(y) = \sqrt{2n+1}P_n^{(0,0)}(y)$, we now deduce (3.8.16).

With this in hand, we apply Corollary 3.7.2 to get that the conclusion of Corollary 3.7.5 hold under the condition

$$m \gtrsim K(s)\cdot L,$$

where $L = (\min\{d+\log(s/\epsilon), \log(2d)\log(s/\epsilon)\} + \log(K(s))\cdot\log(K(s)/\epsilon))$. It remains to estimate $K(s)$ and L. In [2, Cor. 7.7], it was shown that $K(s) \lesssim \min\left\{2^d s, (\pi/2)^d s^{\log(1+4/\pi)/\log(2)}\right\}$. From this, we also observe that $\log(K(s)) \lesssim d+\log(s)$ and $\log(K(s)/\epsilon) \lesssim d+\log(s/\epsilon)$. Therefore $L \lesssim (d+\log(s))(d+\log(s/\epsilon))$, which completes the proof. □

Chapter 4

Parametric differential equations

As mentioned in the previous chapter, a motivation of the high-dimensional function approximation problem comes from approximating a quantity of interest of a parametric differential equation (DE) from a set of discrete samples. In this chapter, we will look at different examples of applying the compressed sensing method to approximate quantities of interest of such parametric DEs. In particular, we will work on the gradient-augmented high-dimensional function approximation problem throughout this chapter.

4.1 Preliminaries

We first review some definitions and theorems in functional analysis. For references, see [66, Chpt. 4 & Chpt. 7], [79, Chpt. 2 & Chpt. 3] and [76, Chpt. 5 & Chpt. 6].

Recall that a *normed space* is a vector space with a given norm defined on it. A normed space is called *complete* if every Cauchy sequence is convergent. A complete normed space is called a *Banach space*. Assume Ω is an open measurable subset of $\mathbb{R}^n$. For $1 \leq p < \infty$, the space $L^p(\Omega)$ consists of all measurable functions f on Ω that satisfy

$$\int_\Omega |f(\boldsymbol{x})|^p \, d\boldsymbol{x} < \infty,$$

and the corresponding L^p-norm of f is defined by

$$\|f\|_{L^p(\Omega)} = \left(\int_\Omega |f(\boldsymbol{x})|^p \, d\boldsymbol{x} \right)^{1/p}.$$

With $p = \infty$, the space $L^\infty(\Omega)$ consists of all measurable functions f that are *essentially bounded* on Ω, which means that there exist a number $0 < M < \infty$ with

$$|f(x)| \leq M \quad \text{almost everywhere in } \Omega,$$

and the L^∞-norm is defined by the *essential supremum*:

$$\begin{aligned}\|f\|_{L^\infty(\Omega)} &= \operatorname*{ess\,sup}_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})| \\ &= \inf\{M|\ |f(\boldsymbol{x})| \le M \text{ almost everywhere in } \Omega\}.\end{aligned}$$

It can be shown that the $L^p(\Omega)$ spaces with $1 \le p \le \infty$ are examples of Banach spaces.

Definition 4.1.1. *An inner product on a vector space $\mathcal{V}$ to the field $\mathcal{K}$ (either $\mathbb{R}$ or $\mathbb{C}$) is a map*

$$\langle\cdot,\cdot\rangle : \mathcal{V}\times\mathcal{V}\to\mathcal{K}$$

such that, for all $v_1, v_2, v_3 \in \mathcal{V}$ and $a, b \in \mathcal{K}$:

(i) $\langle v_1, av_2 + bv_3\rangle = a\langle v_1, v_2\rangle + b\langle v_1, v_3\rangle$;

(ii) $\langle v_2, v_1\rangle = \overline{\langle v_1, v_2\rangle}$;

(iii) $\langle v_1, v_1\rangle \ge 0$;

(iv) $\langle v_1, v_1\rangle = 0$ *if and only if* $v_1 = 0$.

A vector space with an inner product is called an *inner product space*. Moreover, a complete inner product space is a *Hilbert space*. For example, the space of square integrable function on Ω, denoted as $L^2(\Omega)$, is a Hilbert space with an inner product

$$\langle f, g\rangle = \int_\Omega fg\ d\boldsymbol{x}.$$

Next, we will define the *dual space* of a normed space:

Definition 4.1.2. *Let $\mathcal{V}$ be a normed space, the field $\mathcal{K}$ is either $\mathbb{R}$ or $\mathbb{C}$. Then, the set of all bounded linear functionals $f : \mathcal{V}\to\mathcal{K}$ constitutes a normed space with norm defined as*

$$\|f\|_{\mathcal{V}'} = \sup_{\substack{v\in\mathcal{V}\\ v\neq 0}} \frac{|(f,v)|}{\|v\|_{\mathcal{V}}},$$

which is called the dual space of $\mathcal{V}$ and is denoted as $\mathcal{V}'$. Here, $(\cdot,\cdot)$ denotes the dual pairing between between an element in $\mathcal{V}'$ and an element in $\mathcal{V}$. That is, for $f \in \mathcal{V}'$ and $v \in \mathcal{V}$, $(f, v) = f(v)$.

For example, the dual space of the Euclidean space $\mathbb{R}^n$ is $\mathbb{R}^n$. See [79, Example 2.10-5] for the proof. Also, the dual space $\mathcal{V}'$ is a Banach space. See [79, Thm. 2.10-4] for details.

Moreover, similar with the inner product, the *dual pairing* $(\cdot,\cdot) : \mathcal{V}'\times\mathcal{V}\to\mathcal{K}$ has the following properties [77, §6.2]:

(i) $(f, av_1 + bv_2) = a\,(f, v_1) + b\,(f, v_2) \quad$ for $f \in \mathcal{V}', v_1, v_2 \in \mathcal{V}, a, b \in \mathcal{K}$
$(af_1 + bf_2, v) = a\,(f_1, v) + b\,(f_2, v) \quad$ for $f_1, f_2 \in \mathcal{V}', v \in \mathcal{V}, a, b \in \mathcal{K}$

(ii) $|(f, v)| \leq \|f\|_{\mathcal{V}'} \|v\|_{\mathcal{V}} \quad$ for $f \in \mathcal{V}', v \in \mathcal{V}$

(iii) If $(f, v) = 0$ for all $f \in \mathcal{V}'$, then $v = 0$.

Since the dual space $\mathcal{V}'$ is also a normed space, then we can form a dual space of $\mathcal{V}'$, which is denoted by $\mathcal{V}''$ and called the *second dual space* or *bidual space* of $\mathcal{V}$. For every $v \in \mathcal{V}$, there is a bounded linear functional $g_v \in \mathcal{V}''$ satisfying

$$g_v(f) = f(v), \qquad \text{for } f \in \mathcal{V}'.$$

This relation between $\mathcal{V}$ and $\mathcal{V}''$ defines a mapping

$$\begin{aligned} C : \mathcal{V} &\to \mathcal{V}'' \\ v &\to g_v, \end{aligned}$$

which is called the *canonical mapping* of $\mathcal{V}$ into $\mathcal{V}''$. It can be shown that the canonical mapping C is injective. If the mapping C is also surjective, we say this normed space $\mathcal{V}$ is *reflexive*. In other words, it says that $\mathcal{V} = \mathcal{V}''$ under an isometric mapping C. For example, every Hilbert space $\mathcal{H}$ is reflexive. For more details on dual space, bidual space, canonical mapping and reflexivity, see [79, §2.8 & §2.10 & §4.6].

Before giving a couple more definitions, more notation is required. Assume there is a non-negative multi-index $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ of order $|\boldsymbol{\alpha}| = \sum_{n=1}^{m} \alpha_n$. The space of infinitely differentiable functions with compact support in Ω is denoted by $C_0^\infty(\Omega)$. For a test function $\psi(\boldsymbol{x}) \in C_0^\infty(\Omega)$, we denote its partial derivative of order $|\alpha|$ as

$$D^{\boldsymbol{\alpha}}\psi = \left(\frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}}\right) \cdots \left(\frac{\partial^{\alpha_m}}{\partial x_m^{\alpha_m}}\right) \psi.$$

Now we are ready to introduce the concept of weak derivative. As explained in [56, §5.2.1] (see also [25, §1.2]), the motivation for defining the weak derivative is to extend idea of differentiability to the space of functions which is only locally integrable.

Definition 4.1.3. *Given a domain Ω, the set of locally integrable functions is denoted by*

$$L^1_{loc}(\Omega) = \left\{ f : f \in L^1(K) \quad \forall \text{ compact } K \subset \Omega \right\}.$$

Definition 4.1.4. *Suppose $u, v \in L^1_{\text{loc}}(\Omega)$ and $\boldsymbol{\alpha}$ is a multi-index. We say the v is the $\boldsymbol{\alpha}^{th}$-weak partial derivative of u, written $D^{\boldsymbol{\alpha}} u = v$, provided*

$$\int_\Omega u D^{\boldsymbol{\alpha}} \psi \; d\boldsymbol{x} = (-1)^{|\boldsymbol{\alpha}|} \int_\Omega v \psi \; d\boldsymbol{x}$$

for all test functions $\psi \in C_0^\infty(\Omega)$.

We then define the *Sobolev space*, the *Sobolev norm* and its dual space the same way as shown in [56, §5.5.2]:

Definition 4.1.5. *Fix* $1 \le p \le \infty$ *and let* k *be a nonnegative integer. The Sobolev space* $W^{k,p}(\Omega)$ *consists of all locally integrable functions* $u : \Omega \to \mathbb{R}$ *such that for each multi-index* $\boldsymbol{\alpha}$ *with* $|\boldsymbol{\alpha}| \le k$, $D^{\boldsymbol{\alpha}}u$ *exists in the weak sense and belongs to* $L^p(\Omega)$.

Definition 4.1.6. *If* $u \in W^{k,p}(\Omega)$, *we define its norm to be*

$$\|u\|_{W^{k,p}(\Omega)} := \begin{cases} \left(\sum_{|\boldsymbol{\alpha}|\le k} \int_\Omega |D^{\boldsymbol{\alpha}}u|^p \, d\boldsymbol{x}\right)^{1/p} & (1 \le p < \infty), \\ \sum_{|\boldsymbol{\alpha}|\le k} \operatorname{ess\,sup}_\Omega |D^{\boldsymbol{\alpha}}u| & (p = \infty). \end{cases}$$

It can be proved that the Sobolev space $W^{k,p}(\Omega)$ is a Banach space. For details, see [25, Thm. 1.3.2]. For $p = 2$, we often write $H^k(\Omega) = W^{k,2}(\Omega)$. Moreover, the space H^k is a Hilbert space, and the corresponding inner product is defined as

$$\langle u, v\rangle = \sum_{|\boldsymbol{\alpha}|\le k} \int_\Omega D^{\boldsymbol{\alpha}}u D^{\boldsymbol{\alpha}}v d\boldsymbol{x},$$

for $u, v \in H^k(\Omega)$. The space of all functions $u \in H^1(\Omega)$ with $u = 0$ on the boundary $\partial\Omega$ is denoted by $H_0^1(\Omega)$. Moreover, the space of all bounded linear functionals on $H_0^1(\Omega)$ is denoted by $H^*(\Omega)$. By definition of the dual space, we see that $H^*(\Omega)$ is the dual space of $H_0^1(\Omega)$. For more topics on Sobolev space, see [56, Chpt. 5].

Next, we will define the *adjoint operator* on the normed space. For references, see [79, §4.5 & §3.9].

Definition 4.1.7. *Let* $F : \mathcal{V} \to \mathcal{Z}$ *be a bounded linear operator, where* $\mathcal{V}$ *and* $\mathcal{Z}$ *are normed spaces. Then, the adjoint operator* $F^* : \mathcal{Z}' \to \mathcal{V}'$ *of* F *is defined by*

$$(F^*f, v) = (f, Fv), \qquad \forall f \in \mathcal{Z}', v \in \mathcal{V}.$$

For an operator $F : \mathcal{V} \to \mathcal{V}'$, *we say it is self-adjoint if* $F = F^*$.

It can be shown that the adjoint operator F^* is uniquely defined. Suppose there is another operator $S : \mathcal{Z}' \to \mathcal{V}'$ satisfying

$$(Sf, v) = (f, Fv), \qquad \forall f \in \mathcal{Z}', v \in \mathcal{V}.$$

Then, we have

$$(Sf, v) - (F^*f, v) = (Sf - F^*f, v) = 0, \qquad \forall v \in \mathcal{V},$$

which implies that $Sf - F^*f = 0$ for all $f \in \mathcal{Z}'$. So, we can conclude that $S = F^*$.

Moreover, assume $\mathcal{V}, \mathcal{W}, \mathcal{Z}$ are Banach spaces. Let $T : \mathcal{V} \to \mathcal{W}$ and $S : \mathcal{W} \to \mathcal{Z}$ be two bounded linear operators. The composition of ST is then defined as a bounded linear operator from $\mathcal{V}$ to $\mathcal{Z}$. By definition of the adjoint operator, we have

$$((ST)^* f, v) = (f, (ST)v) = (S^* f, Tv) = (T^*(S^* f), v) = ((T^* S^*) f, v),$$

for all $f \in \mathcal{Z}'$ and $v \in \mathcal{V}$, which implies that $(ST)^* = T^* S^*$. Here, $(ST)^* : \mathcal{Z}' \to \mathcal{V}'$ denotes the adjoint operator of ST, $S^* : \mathcal{Z}' \to \mathcal{W}'$ denotes the adjoint operator of S and $T^* : \mathcal{W}' \to \mathcal{V}'$ denotes the adjoint operator of T.

The last definition we need before introducing the adjoint sensitivity analysis method is the Fréchet derivative. See [105, Def. 2.5] or [66, Def. 11.5]. The Fréchet derivative generalizes the derivative of a real-valued function to the derivative of an operator on a normed space, and it is often used to formalize the functional derivative [17, 122].

Definition 4.1.8. *Let $\mathcal{V}, \mathcal{Z}$ be normed vector spaces and $\mathcal{U} \subseteq \mathcal{V}$ be an open subset of $\mathcal{V}$. We say an operator $F : \mathcal{U} \to \mathcal{Z}$ is Fréchet differentiable at a point $\bar{u} \in \mathcal{U}$ if there is a bounded linear operator $DF(\bar{u}) : \mathcal{V} \to \mathcal{Z}$ satisfying*

$$\lim_{\|h\|_{\mathcal{V}} \to 0^+} \frac{\|F(\bar{u} + h) - F(\bar{u}) - DF(\bar{u})h\|_{\mathcal{Z}}}{\|h\|_{\mathcal{V}}} = 0.$$

This bounded linear operator $DF(\bar{u})$ is called the Fréchet derivative of F at the point $\bar{u}$.

It can be checked from the definition that, for a bounded linear operator defined on $\mathcal{V}$, the Fréchet derivative is itself [66, §11.1] [105, §2.2]. Furthermore, we say an operator $F : \mathcal{U} \to \mathcal{Z}$ is continuously Fréchet differentiable if F is differentiable at every point $\bar{u} \in \mathcal{U}$ and the operator $DF : \mathcal{U} \to \mathcal{B}(\mathcal{V}, \mathcal{Z});\ \bar{u} \to DF(\bar{u})$ is continuous, where $\mathcal{B}(\mathcal{V}, \mathcal{Z})$ denotes the set of bounded linear operators from $\mathcal{V}$ to $\mathcal{Z}$ [18, Chpt. 3] [124, §2.1]. We denote the space of continuously Fréchet differentiable operators $F : \mathcal{U} \to \mathcal{Z}$ by $C^1(\mathcal{U}, \mathcal{Z})$. Moreover, the chain rule is also valid for the Fréchet derivative. Let $\mathcal{V}, \mathcal{W}, \mathcal{Z}$ be normed vector spaces and $\mathcal{U}, \mathcal{X}$ be open subsets of $\mathcal{V}, \mathcal{W}$ respectively. If the operator $F : \mathcal{U} \to \mathcal{W}$ satisfies $F(\mathcal{U}) \subseteq \mathcal{X}$ and is Fréchet differentiable at a point $\bar{u} \in \mathcal{U}$, and the operator $G : \mathcal{X} \to \mathcal{Z}$ is Fréchet differentiable at a point $F(\bar{u}) \in \mathcal{X}$, then the composition $G \circ F : \mathcal{U} \to \mathcal{Z}$ is Fréchet differentiable at $\bar{u}$ and its Fréchet derivative at $\bar{u}$ is

$$D(G \circ F)(\bar{u}) = DG(F(\bar{u}))DF(\bar{u}).$$

For the proof, see [18, Chpt. 4] or [124, §2.1.4]. Finally, we define the partial Fréchet derivative for operators whose domains are Banach spaces as follows (see [124, §2.4]) :

Definition 4.1.9. *Let $\mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_n$ be a product of Banach spaces, $\mathcal{Z}$ be a Banach space, and $\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_n \subseteq \mathcal{V}$ be a subset of $\mathcal{V}$ with $\mathcal{U}_i$ open in each $\mathcal{V}_i$ for $i = 1, \ldots, n$.*

For a given operator $F : \mathcal{U} \to \mathcal{Z}$, we say it has an i^{th} partial derivative at the point $\bar{u} = (u_1, \ldots, u_n) \in \mathcal{U}$ if the operator $F_i : \mathcal{U}_i \to \mathcal{Z}$ defined by

$$F_i(u) = F(u_1, \ldots, u_{i-1}, u, u_{i+1}, \ldots, u_n)$$

is Fréchet differentiable at the point $u = u_i$. We define the i^{th} partial derivative of F at the point $\bar{u}$ as $\partial_i F(\bar{u}) := DF_i(u_i)$.

It can be immediately seen from the definition that each operator $\partial_i F(\bar{u})$ is a bounded linear operator from $\mathcal{V}_i$ to $\mathcal{Z}$.

Since it will be useful for later sections, we now introduce the Lax-Milgram Theorem, which can be used to prove the existence and uniqueness of the weak solutions of the elliptic PDEs. An example to demonstrate how this theorem is applied will be shown in §4.3.1.

Theorem 4.1.10. *[56, Lax-Milgram Theorem, Thm. 1 of §6.2.1] Let H be a real Hilbert space with norm $\|\cdot\|_H$. Let $(\cdot, \cdot)$ denotes the pairing of H with its dual space. Assume that*

$$B : H \times H \to \mathbb{R}$$

is a bilinear mapping, for which there exists constants $c_1, c_2 > 0$ such that

$$|B[u, v]| \leq c_1 \|u\|_H \|v\|_H \quad \text{for } u, v \in H, \tag{4.1.1}$$

and

$$c_2 \|u\|_H^2 \leq B[u, u] \quad \text{for } u \in H. \tag{4.1.2}$$

Finally, let $T : H \to \mathbb{R}$ be a bounded linear functional on H. Then, there exists a unique element $u \in H$ such that

$$B[u, v] = (T, v) \tag{4.1.3}$$

for all $v \in H$.

Finally, we introduce two inequalities will be useful for §4.3.1.

Theorem 4.1.11. *[26, Hölder's inequality, Thm. 4.6] Assume that $f \in L^p(\Omega)$ and $g \in L^q(\Omega)$ with $1 \leq p, q \leq \infty$ and $1/p + 1/q = 1$. Then, $fg \in L^1(\Omega)$ and*

$$\int_\Omega |fg| \, d\boldsymbol{x} \leq \|f\|_{L^p(\Omega)} \|g\|_{L^q(\Omega)}.$$

The special case when $p = q = 2$ is called the Cauchy-Schwarz inequality.

Proposition 4.1.12. *[26, Poincarè's inequality, Prop. 8.13] Suppose Ω is a bounded interval. Then there exists a constant C (depending on $|\Omega| < \infty$) such that*

$$\|u\|_{W^{1,p}(\Omega)} \leq C\|\nabla u\|_{L^p(\Omega)}, \qquad \forall u \in W_0^{1,p}(\Omega).$$

4.2 Adjoint sensitivity analysis method

For all examples shown in this chapter, we compute the gradient samples of the quantity of interest (QoI) with the adjoint sensitivity analysis method, which is a common technique used in statistics for studying the impact of uncertain parameters [110, §10.2]. In this section, we will introduce the adjoint sensitivity analysis method, which can be seen as an application of the implicit function theorem (IFT) from multivariable calculus. First, let us review the implicit function theorem:

Theorem 4.2.1 (IFT, Thm. 10.7, [110])**.** *Let $\mathcal{X}, \mathcal{V}$ and $\mathcal{Z}$ be Banach spaces, let $\mathcal{W} \subseteq \mathcal{X} \times \mathcal{V}$ be open, and let $F \in C^1(\mathcal{W}; \mathcal{Z})$. Suppose that, at $(\bar{y}, \bar{u}) \in \mathcal{W}$, the partial Fréchet derivative $\frac{\partial F}{\partial u}(\bar{y}, \bar{u}) : \mathcal{V} \to \mathcal{Z}$ is an invertible bounded linear map. Then there exist open sets $\mathcal{Y} \subseteq \mathcal{X}$ about $\bar{y}$, $\mathcal{U} \subseteq \mathcal{V}$ about $\bar{u}$, with $\mathcal{Y} \times \mathcal{U} \subseteq \mathcal{W}$, and a unique $\vartheta \in C^k(\mathcal{Y}; \mathcal{U})$ such that*

$$\{(y, u) \in \mathcal{Y} \times \mathcal{U} |\ F(y, u) = F(\bar{y}, \bar{u})\} = \{(y, u) \in \mathcal{Y} \times \mathcal{U} |\ u = \vartheta(y)\},$$

i.e. the contour of F through $(\bar{y}, \bar{u})$ is locally the graph of ϑ. Furthermore, $\mathcal{Y}$ can be chosen so that $\frac{\partial F}{\partial u}(y, \vartheta(y))$ is boundedly invertible for all $y \in \mathcal{Y}$, and the Fréchet derivative $\frac{d\vartheta}{dy}(y) : \mathcal{X} \to \mathcal{V}$ of ϑ at any $y \in \mathcal{Y}$ is the composition

$$\frac{d\vartheta}{dy}(y) = -\left(\frac{\partial F}{\partial u}(y, \vartheta(y))\right)^{-1} \left(\frac{\partial F}{\partial y}(y, \vartheta(y))\right). \tag{4.2.1}$$

Now we are ready to derive the adjoint equation in the adjoint sensitivity analysis method with the IFT. The derivation of the adjoint equation presented here is based on [110, §10.2]. Assume $\mathcal{Y}$ and $\mathcal{U}$ are open subsets of Banach spaces $\mathcal{X}$ and $\mathcal{V}$. We have a differential equation (DE) which can be represented as an operator $F : \mathcal{Y} \times \mathcal{U} \to \mathcal{Z}$. For any $y \in \mathcal{Y}$, we have $u(y) \in \mathcal{U}$ as the solution of the function $F(y, u(y)) = 0$, where 0 is the zero element of the Banach space $\mathcal{Z}$. We define the quantity of interest (QoI) of the DE as

$$\begin{aligned} q &: \mathcal{Y} \to \mathbb{R}, \\ q(y) &= Q(y, u(y)), \end{aligned}$$

where the functional $Q : \mathcal{Y} \times \mathcal{U} \to \mathbb{R}$. The goal here is to compute the Fréchet derivative of the QoI with respect to y at a given point $\bar{y} \in \mathcal{Y}$ using the adjoint sensitivity analysis method, given that $F(\bar{y}, \bar{u}) = 0$ with $\bar{y} \in \mathcal{Y}$ and $\bar{u} = u(\bar{y}) \in \mathcal{U}$.

Assume the operator $F \in C^1(\mathcal{Y} \times \mathcal{U}, \mathcal{Z})$ and the partial Fréchet derivative $\frac{\partial F}{\partial u}(\bar{y}, \bar{u}) : \mathcal{V} \to \mathcal{Z}$ is invertible. Since $F(y, u(y)) = 0$, its Fréchet derivative with respect to y vanishes at the point $(\bar{y}, \bar{u})$, that is

$$\frac{d}{dy} F(y, u(y))\Big|_{(y,u)=(\bar{y},\bar{u})} = \frac{\partial F}{\partial u}(\bar{y}, \bar{u}) \frac{\partial u}{\partial y}(\bar{y}, \bar{u}) + \frac{\partial F}{\partial y}(\bar{y}, \bar{u}) = 0,$$

Then, we have the Fréchet derivative $\frac{\partial u}{\partial y}(\bar{y}, \bar{u})$ is the composition:

$$\frac{\partial u}{\partial y}(\bar{y}, \bar{u}) = -\left(\frac{\partial F}{\partial u}(\bar{y}, \bar{u})\right)^{-1} \left(\frac{\partial F}{\partial y}(\bar{y}, \bar{u})\right), \tag{4.2.2}$$

as (4.2.1) in the conclusion of the IFT. By the chain rule, we also have that the Fréchet derivative of $q(y)$ with respect to y at the point $(\bar{y}, \bar{u})$ is

$$\frac{dq}{dy}(\bar{y}) = \frac{\partial Q}{\partial u}(\bar{y}, \bar{u}) \frac{\partial u}{\partial y}(\bar{y}, \bar{u}) + \frac{\partial Q}{\partial y}(\bar{y}, \bar{u}). \tag{4.2.3}$$

By substituting (4.2.2) into (4.2.3), we get

$$\frac{dq}{dy}(\bar{y}) = -\frac{\partial Q}{\partial u}(\bar{y}, \bar{u}) \left(\frac{\partial F}{\partial u}(\bar{y}, \bar{u})\right)^{-1} \frac{\partial F}{\partial y}(\bar{y}, \bar{u}) + \frac{\partial Q}{\partial y}(\bar{y}, \bar{u}). \tag{4.2.4}$$

As a check, we see that $\frac{\partial Q}{\partial u}(\bar{y}, \bar{u})$ defines a bounded linear functional from $\mathcal{V}$ to $\mathbb{R}$, $\left(\frac{\partial F}{\partial u}(\bar{y}, \bar{u})\right)^{-1}$ defines a bounded linear operator from $\mathcal{Z}$ to $\mathcal{V}$, $\frac{\partial F}{\partial y}$ defines a bounded linear operator from $\mathcal{X}$ to $\mathcal{Z}$. So, the composition $\frac{\partial Q}{\partial u}(\bar{y}, \bar{u}) \left(\frac{\partial F}{\partial u}(\bar{y}, \bar{u})\right)^{-1} \frac{\partial F}{\partial y}$ defines a bounded linear functional between $\mathcal{X}$ and $\mathbb{R}$. Moreover, $\frac{\partial Q}{\partial y}$ defines a bounded linear functional from $\mathcal{X}$ to $\mathbb{R}$. It gives that the right-hand side of (4.2.4) defines a bounded linear functional from $\mathcal{X}$ to $\mathbb{R}$, which matches with how the functional $\frac{dq}{dy} : \mathcal{X} \to \mathbb{R}$ is defined on the left-hand side of (4.2.4).

Next, we set

$$\lambda = -\frac{\partial Q}{\partial u}(\bar{y}, \bar{u}) \left(\frac{\partial F}{\partial u}(\bar{y}, \bar{u})\right)^{-1}.$$

Then, we can use λ to rewrite (4.2.4) as

$$\frac{dq}{dy}(\bar{y}) = \lambda \frac{\partial F}{\partial y}(\bar{y}, \bar{u}) + \frac{\partial Q}{\partial y}(\bar{y}, \bar{u}), \tag{4.2.5}$$

where the functional $\lambda \in \mathcal{Z}'$ (the dual space of Z) is the solution to

$$\lambda \frac{\partial F}{\partial u}(\bar{y}, \bar{u}) = -\frac{\partial Q}{\partial u}(\bar{y}, \bar{u}). \tag{4.2.6}$$

Alternatively, we can consider the adjoint of (4.2.6). Since the adjoint operator on Banach space is uniquely defined, (4.2.6) is equivalent to

$$\left(\lambda\frac{\partial F}{\partial u}(\bar{y},\bar{u})\right)^* = -\left(\frac{\partial Q}{\partial u}(\bar{y},\bar{u})\right)^*. \tag{4.2.7}$$

Since it has been shown in §4.1 that

$$\left(\lambda\frac{\partial F}{\partial u}(\bar{y},\bar{u})\right)^* = \left(\frac{\partial F}{\partial u}(\bar{y},\bar{u})\right)^*\lambda^*,$$

then (4.2.7) can be rewritten as

$$\left(\frac{\partial F}{\partial u}(\bar{y},\bar{u})\right)^*\lambda^* = -\left(\frac{\partial Q}{\partial u}(\bar{y},\bar{u})\right)^*, \tag{4.2.8}$$

which is known as the *adjoint equation* in the adjoint sensitivity analysis method. Note that $\left(\frac{\partial F}{\partial u}(\bar{y},\bar{u})\right)^* : \mathcal{Z}' \to \mathcal{V}'$ is a bounded linear operator and $\lambda^* : \mathbb{R} \to \mathcal{Z}'$ is a bounded linear operator since the dual space of $\mathbb{R}$ is $\mathbb{R}$. Thus, we have $\left(\frac{\partial F}{\partial u}(\bar{y},\bar{u})\right)^*\lambda^*$ defines a bounded linear operator from $\mathbb{R}$ to $\mathcal{V}'$, which matches with how the operator $\left(\frac{\partial Q}{\partial u}(\bar{y},\bar{u})\right)^*$ is defined on the right-hand side.

After solving for λ in (4.2.6), then $\frac{dq}{dy}(\bar{y})$ can be directly computed by substituting λ back to (4.2.5). Note that, in practice, we often solve for λ^* in (4.2.8) first and find its adjoint, instead of directly solving for λ via (4.2.6). This approach of computing the derivative of $q(y)$ at $\bar{y}$ by solving the adjoint equation is called the *adjoint sensitivity analysis method*. As pointed out in [110, §10.2], the adjoint sensitivity analysis method provides an easy way to compute the bounded linear functional $\frac{dq}{dy}$ without evaluating the derivative $\frac{\partial u}{\partial y}$ explicitly. To compute $\frac{dq}{dy}$ with the adjoint equation (4.2.8), we only need to know the partial Fréchet derivatives F and Q with respect to y and u. Compared to $\frac{\partial u}{\partial y}$, these derivatives are often much easier to compute.

4.3 Parametric diffusion equation with homogenous Dirichlet boundary conditions

In this section, we will present several examples of approximating the quantities of interest (QoIs) of parametric diffusion equations with homogenous Dirichlet boundary conditions. The parametric diffusion equation with homogenous Dirichlet boundary conditions is defined by

$$-\nabla\cdot(a(\boldsymbol{y},\boldsymbol{x})\nabla u(\boldsymbol{x})) = f(\boldsymbol{x}), \qquad \forall \boldsymbol{x}\in\Omega, \tag{4.3.1}$$

$$u(\boldsymbol{x}) = 0, \qquad \forall \boldsymbol{x}\in\partial\Omega, \tag{4.3.2}$$

where Ω is an open and bounded physical domain in $\mathbb{R}^n$, $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ is the n-dimensional physical variable. The parametric diffusion coefficient satisfies $a(\boldsymbol{y}, \boldsymbol{x}) > 0$, where $\boldsymbol{y} = (y_1, y_2, \ldots, y_d) \in D$ is a d-dimensional parameter vector and D is an open subset of $\mathbb{R}^d$ with a probability measure defined on it. The function $\boldsymbol{y} \to a(\boldsymbol{y}, \cdot)$ is continuously Fréchet differentiable for all $\boldsymbol{y} \in D$. Moreover, the parametric diffusion coefficient $a(\boldsymbol{y}, \boldsymbol{x})$ is bounded, that is

$$M_1 \leq a(\boldsymbol{y}, \boldsymbol{x}) \leq M_2, \qquad \forall \boldsymbol{x}, \boldsymbol{y},$$

where M_1, M_2 are positive real numbers. The forcing term on the right-hand side of (4.3.1) satisfies $f(\boldsymbol{x}) \in L^2(\Omega)$. The QoI is $q(\boldsymbol{y}) = Q(u(\boldsymbol{y}))$, where Q is a bounded linear functional acting on the solution u. Here, the solution u depends on $\boldsymbol{y}$ through (4.3.1). It is clear that u always depends on $\boldsymbol{x}$. For simplicity, we write $u(\boldsymbol{y}, \boldsymbol{x})$ as $u(\boldsymbol{y})$.

4.3.1 The weak problem

As the first step for obtaining one sample of QoI and its gradient sample, we need to solve the weak problem of (4.3.1) and (4.3.2) at a given point $\bar{\boldsymbol{y}} = (\bar{y}_1, \ldots, \bar{y}_d) \in D$. We derive the weak problem to solve with multiplying (4.3.1) by a smooth test function $v \in C_0^\infty(\Omega)$ and integrating over Ω. With integration by parts, we get

$$\begin{aligned} -\int_\Omega \left(\nabla \cdot (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla \bar{u}(\boldsymbol{x}))\right) v(\boldsymbol{x}) d\Omega &= \int_\Omega (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla \bar{u}(\boldsymbol{x})) \cdot \nabla v(\boldsymbol{x}) d\Omega \\ &= \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) d\Omega \quad \text{since } v(\boldsymbol{x}) = 0 \text{ on } \partial\Omega. \end{aligned}$$

Due to the fact that $C_0^\infty(\Omega)$ is dense in $H_0^1(\Omega)$, the smooth test function v can be replaced by $v \in H_0^1(\Omega)$ [37, 75]. Now we have $\bar{u} \in H_0^1(\Omega)$ to be a solution of the weak problem

$$B[\bar{u}, v] = (T, v), \qquad \forall v \in H_0^1(\Omega), \tag{4.3.3}$$

where the bilinear mapping is defined by

$$B[\bar{u}, v] = \int_\Omega (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla \bar{u}(\boldsymbol{x})) \cdot \nabla v(\boldsymbol{x}) d\Omega,$$

for $\bar{u}, v \in H_0^1(\Omega)$. The functional $T : H_0^1(\Omega) \to \mathbb{R}$ on the right-hand side is defined by

$$(T, v) = \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) d\Omega = \langle f, v \rangle,$$

where $\langle f, v \rangle$ is the $L^2(\Omega)$ inner product.

With the Lax-Milgram Theorem, we can show that (4.3.3) has a unique solution. First, by Cauchy-Schwarz inequality, we have

$$|B[\bar{u}, v]| \leq M_2 \|\nabla \bar{u}\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} \leq M_2 \|\bar{u}\|_{H_0^1(\Omega)} \|v\|_{H_0^1(\Omega)}.$$

Since $M_1 \leq a(\bar{\boldsymbol{y}}, \boldsymbol{x})$, we have

$$\frac{M_1}{1+c} \|\bar{u}\|^2_{L^2(\Omega)} \leq B[\bar{u}, \bar{u}],$$

where c is the constant from Poincaré's inequality. Finally, we have that $T : H_0^1(\Omega) \to \mathbb{R}$ defines a bounded linear functional on $H_0^1(\Omega)$, since

$$|(T, v)| \leq \|f\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \leq \|f\|_{L^2(\Omega)} \|v\|_{H_0^1(\Omega)}.$$

In other words, it says that $T \in H^*(\Omega)$, the dual space of $H_0^1(\Omega)$. We have now proved that (4.3.3) has a unique solution $\bar{u} \in H_0^1(\Omega)$. Note that, for simplicity, we will write $H^*(\Omega)$ as H^* from now on.

4.3.2 The adjoint equation

In the previous subsection, we derived the weak problem and proved the existence and uniqueness of the solution. Now we are ready to derive the adjoint equation for generating the gradient samples of the quantity of interest (QoI).

Recall, in §4.2, we have shown that the adjoint sensitivity analysis method is a simple application of the implicit function theorem (IFT). Thus, in order to derive the adjoint equation, we first need to check whether the assumptions needed for the IFT are satisfied. Since the boundary value problem (shown as (4.3.1) and (4.3.2)) can be rewritten as an operator

$$F : D \times (H^2(\Omega) \cap H_0^1(\Omega)) \to L^2.$$

we have that, for any $\boldsymbol{y} \in D$, the strong solution $u \in H^2(\Omega) \cap H_0^1(\Omega)$ solves the function

$$F(\boldsymbol{y}, u) = -\nabla \cdot (a(\boldsymbol{y}, \boldsymbol{x}) \nabla u(\boldsymbol{x})) - f(\boldsymbol{x}) = 0$$

with $u = 0$ on boundaries. Furthermore, we can extend the operator F to be

$$F : D \times H_0^1(\Omega) \to H^*,$$

which is defined by

$$(F(\boldsymbol{y}, u), v) = (-\nabla \cdot (a(\boldsymbol{y}, \boldsymbol{x}) \nabla u), v) - (T, v) = \langle a(\boldsymbol{y}, \boldsymbol{x}) \nabla u, \nabla v \rangle - \langle f, v \rangle, \tag{4.3.4}$$

for $F(\boldsymbol{y}, u) \in H^*$ and for all $v \in H_0^1(\Omega)$. In other words, for $\boldsymbol{y} \in D$, the weak solution $u \in H_0^1(\Omega)$ solves the function

$$(F(\boldsymbol{y}, u), v) = \langle a(\boldsymbol{y}, \boldsymbol{x})\nabla u, \nabla v\rangle - \langle f, v\rangle = 0.$$

Thus, two open subsets of Banach spaces $\mathcal{X}$ and $\mathcal{V}$ considered in the IFT are $\mathcal{Y} = D \subset \mathcal{X} = \mathbb{R}^d$ and $\mathcal{U} = H_0^1(\Omega) = \mathcal{V}$ respectively.

Now we shall show the operator $F \in C^1(\mathcal{Y} \times \mathcal{U}, H^*)$. Note that $F(\boldsymbol{y}, u)$ can be rewritten as $F(\boldsymbol{y}, u) = G(\boldsymbol{y}, u) - f(\boldsymbol{x})$, where the operator $G : \mathcal{Y} \times \mathcal{U} \to H^*$ is defined by

$$(G(\boldsymbol{y}, u), v) = \left(-\nabla \cdot \left(a(\boldsymbol{y}, \boldsymbol{x})\nabla u\right), v\right) = \langle a(\boldsymbol{y}, \boldsymbol{x})\nabla u, \nabla v\rangle.$$

Since $f(\boldsymbol{x})$ only depends on $\boldsymbol{x}$, $f(\boldsymbol{x})$ can be considered as a constant function in $\boldsymbol{y}$ and u. In other words, in order to show $F \in C^1(\mathcal{Y} \times \mathcal{U}, H^*)$, all we need is to show $G \in C^1(\mathcal{Y} \times \mathcal{U}, H^*)$. For a given $\bar{\boldsymbol{y}} \in \mathcal{Y}$, we define $G_{\bar{\boldsymbol{y}}} : \mathcal{U} \to H^*$ by

$$(G_{\bar{\boldsymbol{y}}}(\bar{u}), v) = \left(-\nabla \cdot \left(a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla \bar{u}\right), v\right) = \langle a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla \bar{u}, \nabla v\rangle.$$

The operator $G_{\bar{\boldsymbol{y}}}$ on $\mathcal{U}$ is linear, since

$$\begin{aligned} G_{\bar{\boldsymbol{y}}}(\bar{u}_1 + \bar{u}_2) &= -\nabla \cdot \left(a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla \left(\bar{u}_1(\boldsymbol{x}) + \bar{u}_2(\boldsymbol{x})\right)\right) \\ &= \left(-\nabla \cdot \left(a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla \bar{u}_1(\boldsymbol{x})\right)\right) + \left(-\nabla \cdot \left(a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla \bar{u}_2(\boldsymbol{x})\right)\right) \\ &= G_{\bar{\boldsymbol{y}}}(\bar{u}_1) + G_{\bar{\boldsymbol{y}}}(\bar{u}_2), \end{aligned}$$

for $\bar{u}_1, \bar{u}_2 \in \mathcal{U}$. With the same procedure used in Chapter 1 of [117], we can prove the operator $G_{\bar{\boldsymbol{y}}}$ is also bounded on $\mathcal{U}$. Recall, it has been seen in §4.3.1 that $(G_{\bar{\boldsymbol{y}}}(\bar{u}), v) = B[\bar{u}, v]$ for $\bar{u}, v \in \mathcal{U}$. So, we have

$$\left\|G_{\bar{\boldsymbol{y}}}(\bar{u})\right\|_{H^*} = \sup_{\substack{v \in \mathcal{U} \\ v \neq 0}} \frac{(G_{\bar{\boldsymbol{y}}}(\bar{u}), v)}{\|v\|_{\mathcal{U}}} = \sup_{\substack{v \in \mathcal{U} \\ v \neq 0}} \frac{B[\bar{u}, v]}{\|v\|_{\mathcal{U}}} \leq M_2 \|\bar{u}\|_{\mathcal{U}},$$

which proves that $G_{\bar{\boldsymbol{y}}}$ is a linear bounded operator on $\mathcal{U}$. This implies that the operator $G_{\bar{\boldsymbol{y}}}$ is Fréchet differentiable on $\mathcal{U}$ and its Fréchet derivative is itself. Given the function $\boldsymbol{y} \to a(\boldsymbol{y}, \cdot)$ is continuously Fréchet differentiable for all $\boldsymbol{y} \in \mathcal{Y}$, we have $G \in C^1(\mathcal{Y} \times \mathcal{U}, H^*)$. With this in hand, we can deduce that $F \in C^1(\mathcal{Y} \times \mathcal{U}, H^*)$.

Next, we need to show the partial Fréchet derivative $\frac{\partial F}{\partial u}(\bar{\boldsymbol{y}}, \bar{u}) : \mathcal{U} \to H^*$ is invertible. For the same reason as before, it is equivalent to showing that the partial Fréchet derivative $\frac{\partial G}{\partial u}(\bar{\boldsymbol{y}}, \bar{u}) : \mathcal{U} \to H^*$ is invertible. Since the Fréchet derivative of the operator $G_{\bar{\boldsymbol{y}}}$ on $\mathcal{U}$ is itself, we have that $\frac{\partial G}{\partial u}(\bar{\boldsymbol{y}}, \bar{u}) = G_{\bar{\boldsymbol{y}}}$ for any given $\bar{\boldsymbol{y}} \in \mathcal{Y}$. Moreover, it has been proved in

§4.3.1 that the weak problem

$$(G_{\bar{\boldsymbol{y}}}(\bar{u}), v) = (T, v) = \langle f, v\rangle, \qquad \forall v \in \mathcal{U}$$

has a unique solution $\bar{u} \in \mathcal{U}$. So, we have the operator $G_{\bar{\boldsymbol{y}}}$ is invertible and $\bar{u} = G_{\bar{\boldsymbol{y}}}^{-1}(T)$, where the operator $G_{\bar{\boldsymbol{y}}}^{-1} : H^* \to \mathcal{U}$. This indicates that the partial Fréchet derivative $\frac{\partial F}{\partial u}(\bar{\boldsymbol{y}}, \bar{u}) : \mathcal{U} \to H^*$ is invertible.

Now we can apply the IFT and compute the gradient samples of the QoI with

$$\frac{dq}{d\boldsymbol{y}}(\bar{\boldsymbol{y}}) = \lambda \frac{\partial F}{\partial \boldsymbol{y}}(\bar{\boldsymbol{y}}, \bar{u}), \tag{4.3.5}$$

where the functional $\lambda \in (H^*)'$ is the solution to

$$\lambda \frac{\partial F}{\partial u}(\bar{\boldsymbol{y}}, \bar{u}) = -\frac{\partial Q}{\partial u}(\bar{u}). \tag{4.3.6}$$

Since the operators $\frac{\partial F}{\partial u}(\bar{\boldsymbol{y}}, \bar{u}) = G_{\bar{\boldsymbol{y}}}$ and $\frac{\partial Q}{\partial u}(\bar{u}) = Q$, (4.3.6) can be rewritten as

$$\lambda G_{\bar{\boldsymbol{y}}} = -Q, \tag{4.3.7}$$

where the bounded linear operator $G_{\bar{\boldsymbol{y}}} : \mathcal{U} \to H^*$, the bounded linear functional $\lambda : H^* \to \mathbb{R}$, the composition $\lambda G_{\bar{\boldsymbol{y}}} : \mathcal{U} \to \mathbb{R}$ and the bounded linear operator $Q : \mathcal{U} \to \mathbb{R}$. Next, we will derive the adjoint equation of (4.3.7). As a first step, we need to find the adjoint operator $G_{\bar{\boldsymbol{y}}}^* : (H^*)' \to \mathcal{U}'$ of $G_{\bar{\boldsymbol{y}}}$. Recall, since every Hilbert space is reflexive, we have $(H^*)' = H_0^1(\Omega) = \mathcal{U}$. With $\mathcal{U}' = (H_0^1(\Omega))' = H^*$, we know that the adjoint operator $G_{\bar{\boldsymbol{y}}}^*$ satisfies $G_{\bar{\boldsymbol{y}}}^* : \mathcal{U} \to H^*$. By definition of the adjoint operator (shown in §4.1), we have

$$\left(G_{\bar{\boldsymbol{y}}}^* v, w\right) = \left(v, G_{\bar{\boldsymbol{y}}} w\right), \tag{4.3.8}$$

for all $w, v \in \mathcal{U}$. Applying integration by parts twice on the right-hand side of (4.3.8), we get

$$\begin{aligned}(v, G_{\bar{\boldsymbol{y}}} w) &= \int_\Omega \left(-\nabla \cdot (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla w(\boldsymbol{x}))\right) v(\boldsymbol{x}) d\Omega = \int_\Omega (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla w(\boldsymbol{x})) \cdot \nabla v(\boldsymbol{x}) d\Omega \\ &= \int_\Omega \left(-\nabla \cdot (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla v(\boldsymbol{x}))\right) w(\boldsymbol{x}) d\Omega = \left(G_{\bar{\boldsymbol{y}}} v, w\right),\end{aligned}$$

which implies that the operator $G_{\bar{\boldsymbol{y}}} = G_{\bar{\boldsymbol{y}}}^*$, i.e. $G_{\bar{\boldsymbol{y}}}$ is self-adjoint. Thus, we have the adjoint equation of (4.3.8) as

$$G_{\bar{\boldsymbol{y}}} \lambda^* = -Q^*, \tag{4.3.9}$$

where the bounded linear operators $\lambda^* : \mathbb{R} \to \mathcal{U}$ and $Q^* : \mathbb{R} \to H^*$ are the adjoint operators of λ and Q respectively and the bounded linear operator $G_{\bar{\boldsymbol{y}}} : \mathcal{U} \to H^*$. Finally, after solving (4.3.7) for λ (or (4.3.9) for λ^*) and computing the partial Fréchet derivative $\frac{\partial F}{\partial \boldsymbol{y}}(\bar{\boldsymbol{y}}, \bar{u})$, we can obtain the gradient of the QoI at $\bar{\boldsymbol{y}}$ using (4.3.5).

4.3.3 Galerkin discretization and the discretized adjoint equation

In order to solve the weak problem numerically, we first need to truncate the problem into a finite-dimensional setting. In this subsection, we will discretize the weak problem with the Galerkin method, described in [25, §0.2] and [57], and show how the adjoint sensitivity analysis method works for the discretized setting.

Recall, the weak problem to solve is defined by

$$B[\bar{u}, v] = \langle f, v \rangle, \tag{4.3.10}$$

where the bilinear mapping $B[\bar{u}, v] = \int_\Omega (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla \bar{u}(\boldsymbol{x})) \cdot \nabla v(\boldsymbol{x}) d\Omega$ and the inner product $\langle f, v \rangle = \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) d\Omega$ for all $v \in \mathcal{U}$. As the first step of the Galerkin method, we define a finite-dimensional space, $\mathcal{U}_h$, which is a subspace of $\mathcal{U}$. Instead of (4.3.10), now we solve a discretized problem

$$B[u_h, v] = \langle f, v \rangle, \tag{4.3.11}$$

where $u_h, v \in \mathcal{U}_h$.

Let the set $\{\varphi_j : 1 \le j \le N\}$ be a basis of $\mathcal{U}_h$. We can write an approximated solution of $\bar{u}$ as $u_h(\boldsymbol{x}) = \sum_{j=1}^N u_j \varphi_j(\boldsymbol{x})$ and the test function v as $v(\boldsymbol{x}) = \sum_{i=1}^N v_i \varphi_i(\boldsymbol{x})$ with $v(\boldsymbol{x}) = 0$ on the boundary. Now (4.3.11) can be rewritten as a matrix equation

$$\boldsymbol{U} \bar{\boldsymbol{u}} = \boldsymbol{b}, \tag{4.3.12}$$

where

$$\begin{aligned}
\boldsymbol{U} &= \left(\int_\Omega a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla \varphi_i(\boldsymbol{x}) \nabla \varphi_j(\boldsymbol{x}) d\Omega \right)_{i,j=1}^N \in \mathbb{R}^{N \times N}, \\
\boldsymbol{b} &= \left(\int_\Omega f(\boldsymbol{x}) \varphi_i(\boldsymbol{x}) d\Omega \right)_{i=1}^N \in \mathbb{R}^N, \\
\bar{\boldsymbol{u}} &= (u_1, u_2, \ldots, u_N)^T \in \mathbb{R}^N.
\end{aligned}$$

Then, we solve for $\bar{\boldsymbol{u}}$ in (4.3.12) to get an approximated solution u_h of $\bar{u}$. The matrix U defined above is often referred as the *stiffness matrix* in the finite element literature. Note that the matrix $\boldsymbol{U}$ is symmetric and is also positive definite since $B[w, w] > 0$ for all nonzero $w \in \mathcal{U}_h$ [25, §0.2]. Thus, (4.3.12) has a unique solution $\bar{\boldsymbol{u}} = \boldsymbol{U}^{-1} \boldsymbol{b}$.

Recall, at a given point $\bar{\boldsymbol{y}}$, the quantity of interest (QoI) is defined by

$$q(\bar{\boldsymbol{y}}) = Q(\bar{u}), \tag{4.3.13}$$

where $Q : \mathcal{U} \to \mathbb{R}$ is a bounded linear functional. We approximate $\bar{u}$ with $u_h \in \mathcal{U}_h$, which has coefficients $\bar{\boldsymbol{u}}$ in the $\{\varphi_j\}$ basis. In the discretized setting, (4.3.13) becomes

$$q(\bar{\boldsymbol{y}}) = Q_h(\bar{\boldsymbol{u}}) = \boldsymbol{a}^T \bar{\boldsymbol{u}}, \tag{4.3.14}$$

where $\boldsymbol{a}^T : \mathbb{R}^N \to \mathbb{R}$ represents the operator Q_h – the discretized version of the operator Q. Moreover, the operator $\frac{\partial Q_h}{\partial \boldsymbol{u}}(\bar{\boldsymbol{u}}) : \mathbb{R}^N \to \mathbb{R}$ defines a mapping $\bar{\boldsymbol{u}} \to \boldsymbol{a}^T \bar{\boldsymbol{u}}$.

At a given point $\bar{\boldsymbol{y}}$, we can write the discretized version of (4.3.4) as

$$F_h(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}) = \boldsymbol{U}\bar{\boldsymbol{u}} - \boldsymbol{b},$$

where the operator $F_h : \mathbb{R}^N \to \mathbb{R}^N$. Moreover, we have

$$\frac{\partial F_h}{\partial \boldsymbol{u}}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}) : \mathbb{R}^N \to \mathbb{R}^N, \quad \bar{\boldsymbol{u}} \to \boldsymbol{U}\bar{\boldsymbol{u}}.$$

Thus, (4.3.7) can be understood as

$$\begin{aligned} \lambda_h(\boldsymbol{U}\bar{\boldsymbol{u}}) &= -\boldsymbol{a}^T(\bar{\boldsymbol{u}}), \\ \Rightarrow \boldsymbol{\lambda}\boldsymbol{U} &= -\boldsymbol{a}^T, \end{aligned} \tag{4.3.15}$$

where $\boldsymbol{\lambda} : \mathbb{R}^N \to \mathbb{R}$ represents the functional λ_h– the discretized version of the functional $\lambda \in (H^*)'$, $\boldsymbol{U} : \mathbb{R}^N \to \mathbb{R}^N$ and the composition $\boldsymbol{\lambda}\boldsymbol{U} : \mathbb{R}^N \to \mathbb{R}$. Since $\boldsymbol{U}$ is symmetric, the adjoint equation of (4.3.15) becomes

$$\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{a}, \tag{4.3.16}$$

where $\boldsymbol{\lambda}^T : \mathbb{R} \to \mathbb{R}^N$ and $\boldsymbol{a} : \mathbb{R} \to \mathbb{R}^N$. Then, we can solve the matrix equation (4.3.16) for $\boldsymbol{\lambda}^T$ and get $\boldsymbol{\lambda}$ by simply taking the transpose.

Furthermore, we have that

$$\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}) = \boldsymbol{U}_k \bar{\boldsymbol{u}}, \tag{4.3.17}$$

where

$$\boldsymbol{U}_k = \left(\int_\Omega a_k(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla\varphi_i(\boldsymbol{x}) \nabla\varphi_j(\boldsymbol{x}) d\Omega \right)_{i,j=1}^{N,N} \in \mathbb{R}^{N\times N} \quad \text{with } a_k = \frac{\partial a(\bar{\boldsymbol{y}}, \boldsymbol{x})}{\partial y_k},\ k = 1, \ldots, d.$$

Thus, the gradient sample of the QoI at $\bar{\boldsymbol{y}}$ can be computed as

$$\frac{dq}{dy_k}(\bar{\boldsymbol{y}}) = \boldsymbol{\lambda}\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}),$$

for $k = 1, \dots, d$.

4.4 Numerical results for homogenous Dirichlet problems

After deriving the discretized weak problem and the discretized adjoint equation in §4.3.3, we are ready to show some examples. In this section, we will present the approximation results of the quantities of interest (QoIs) for three problems with the same set-up as shown in §4.3. Note that, for simplicity, the problems we present here have either one-dimensional or two-dimensional physical domain, i.e. $\Omega = (0,1)$ or $\Omega = (0,1)^2$. However, the higher dimensional problem can be solved in the same way. For details on how these problems are solved numerically, see Appendix A. In this section, §4.6 and §4.7, we use $\boldsymbol{e}_i \in \mathbb{R}^N$ to denote the column vector with 1 at the ith position and 0's elsewhere.

4.4.1 One-dimensional diffusion equation

Let us first look at a simple one-dimensional problem. Suppose that $u : \bar{\Omega} \to \mathbb{R}$ solves the Dirichlet boundary value problem

$$\begin{aligned} -\frac{d}{dx}\left(e^y \frac{d}{dx}u(x)\right) &= f(x), && x \in \Omega, \\ u &= 0, && x \in \partial\Omega, \end{aligned}$$

where the physical domain $\Omega = (0,1)$, the parameter $y \in \mathcal{Y} = (-1,1)$, and the right-hand side function is defined as $f(x) = x(x+1) \in L^2(\Omega)$. Here, we are interested in approximating the QoI $q(y) = u(y, 0.18)$ with gradient-augmented samples.

We discretize this problem by taking 50 equal spaced subregions between 0 and 1, which gives in total $N = 51$ nodes on $[0,1]$. With this discretization, we can explicitly form the matrix equation $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$ at any given point $\bar{y} \in \mathcal{Y}$ and write the QoI at $\bar{y}$ as $q(\bar{y}) = \boldsymbol{e}_{10}^T\bar{\boldsymbol{u}}$. The adjoint equation is given as $\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{e}_{10}$ for $\boldsymbol{\lambda}^T \in \mathbb{R}^N$. In this case, we have $\frac{\partial F_h}{\partial y}(\bar{y}, \bar{\boldsymbol{u}}) = \boldsymbol{A}\bar{\boldsymbol{u}} = \boldsymbol{b}$ and the derivative of the QoI can be computed as $\frac{dq}{dy}(\bar{y}) = \boldsymbol{\lambda}\boldsymbol{b}$. Note that this one-dimensional problem is solved by considering a two-dimensional version of this problem with assuming the solution is constant along the vertical direction. For reference on how to implement one-dimensional problems on FreeFem++, see [10].

Figure 4.1 shows the approximation error $\|q - \tilde{q}\|_{L^\infty}$ against the computational cost $\tilde{m}$ with $s = 2048$ when different approximating polynomial bases are used. With the same amount of computational cost, we see that a smaller approximation error is often obtained when the gradient samples are considered. Figure 4.1 also compares different weighting

strategies. We see that, compared to the unweighted case, an improvement of the recovery is obtained when a weighted ℓ^1 minimization problem is solved.

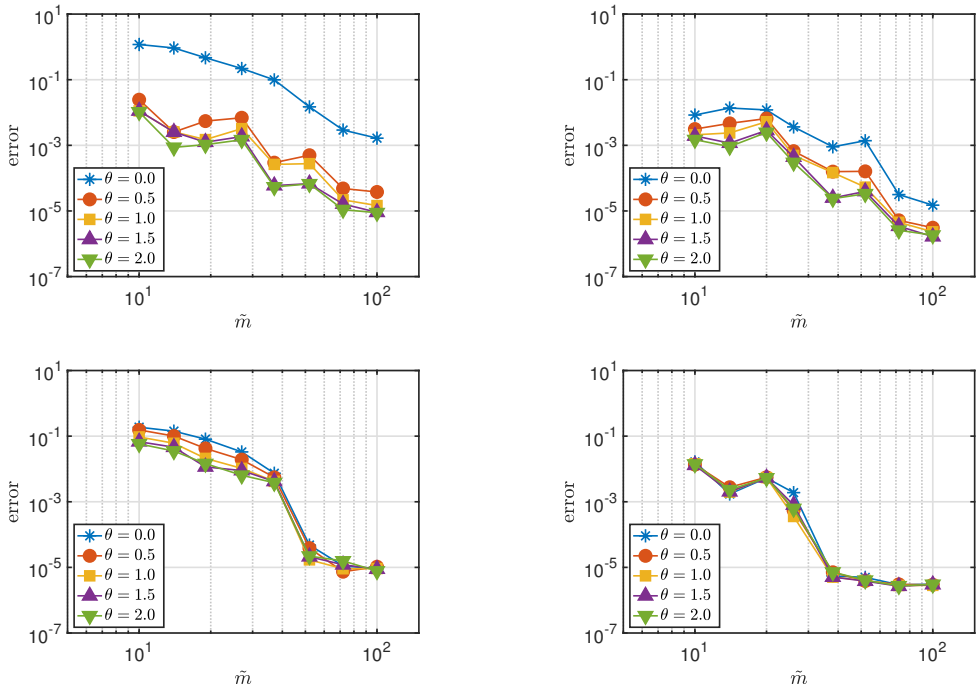


Figure 4.1: The $\|q - \tilde{q}\|_{L^\infty}$ recovery error of the one-dimensional diffusion equation against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density (top) and Chebyshev polynomials with points drawn from the Chebyshev density (bottom). The unaugmented case is shown on the left column and the gradient-augmented case is shown on the right.

4.4.2 Two-dimensional diffusion equation

Next, we consider a two-dimensional problem, which is defined by

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{y}, \boldsymbol{x}) \nabla u(\boldsymbol{x})) &= f(\boldsymbol{x}), && \forall \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) &= 0, && \forall \boldsymbol{x} \in \partial\Omega, \end{aligned}$$

where $\Omega = (0,1)^2$ is the physical domain, $\boldsymbol{x} = (x_1, x_2)$ is the two-dimensional physical variable. The parametric diffusion coefficient is defined by

$$a(\boldsymbol{y}, \boldsymbol{x}) = e^{y_1} \cdot \mathbb{1}_{\Omega_1}(\boldsymbol{x}) + e^{y_2} \cdot \mathbb{1}_{\Omega_2}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \Omega, \ \forall \boldsymbol{y} \in \mathcal{Y},$$

where $\boldsymbol{y} = (y_1, y_2) \in \mathcal{Y}$ is a two-dimensional parameter vector and $\mathcal{Y} = (-1,1)^2$. The indicator function on Ω_1 is denoted by $\mathbb{1}_{\Omega_1}$ and $\mathbb{1}_{\Omega_2}$ is the indicator function on Ω_2, where Ω_1 is defined as the left half of the unit square Ω and Ω_2 is the right half of Ω. The

right-hand side is defined as $f(\boldsymbol{x}) = x_1(x_1+1) \in L^2(\Omega)$. The QoI to approximate here is $q(\boldsymbol{y}) = u(\boldsymbol{y}, (0.7, 0.7))$.

We discretize the domain by taking 20 equal spaced intervals on each boundary of Ω and the vertical line in the middle of Ω. By doing this, we will construct a mesh of 896 triangles with in total $N = 489$ vertices. Figure 4.2 shows the triangular mesh for this two-dimensional problem.

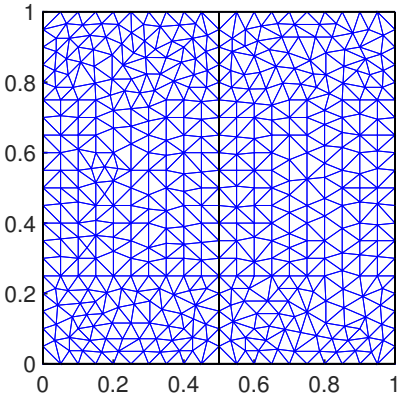


Figure 4.2: Triangular mesh for the two-dimensional diffusion equation.

Now, for any given $\bar{\boldsymbol{y}}$, we can explicitly from a matrix equation $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$. Since $\boldsymbol{x} = (0.7, 0.7)$ corresponds to the 309th vertex of the mesh, we can write the QoI at $\bar{\boldsymbol{y}}$ as $q(\bar{\boldsymbol{y}}) = \boldsymbol{e}_{309}^T \bar{\boldsymbol{u}}$. The corresponding adjoint equation is $\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{e}_{309}$ for $\boldsymbol{\lambda}^T \in \mathbb{R}^N$. Moreover, $\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}})$ can be computed using (4.3.17) with

$$a_k(\bar{\boldsymbol{y}}, \boldsymbol{x}) = e^{\bar{y}_k} \times \mathbb{1}_{\Omega_k}(\boldsymbol{x}), \quad \text{for } k = 1, 2.$$

Then, we have the gradient of the QoI at $\bar{\boldsymbol{y}}$ as

$$\frac{dq}{dy_k}(\bar{\boldsymbol{y}}) = \boldsymbol{\lambda}\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}),$$

for $k = 1, 2$.

Figure 4.3 shows the approximation error $\|q - \tilde{q}\|_{L^\infty}$ against $\tilde{m}$ with $s = 365$ when different kinds of approximating polynomial bases are used. As in the one-dimensional problem, compared to the unaugmented case, a smaller error is often obtained when the gradient samples are considered. Again, compared to the unweighted ℓ^1 minimization, we see an improved approximation result when a weighted ℓ^1 minimization problem is solved.

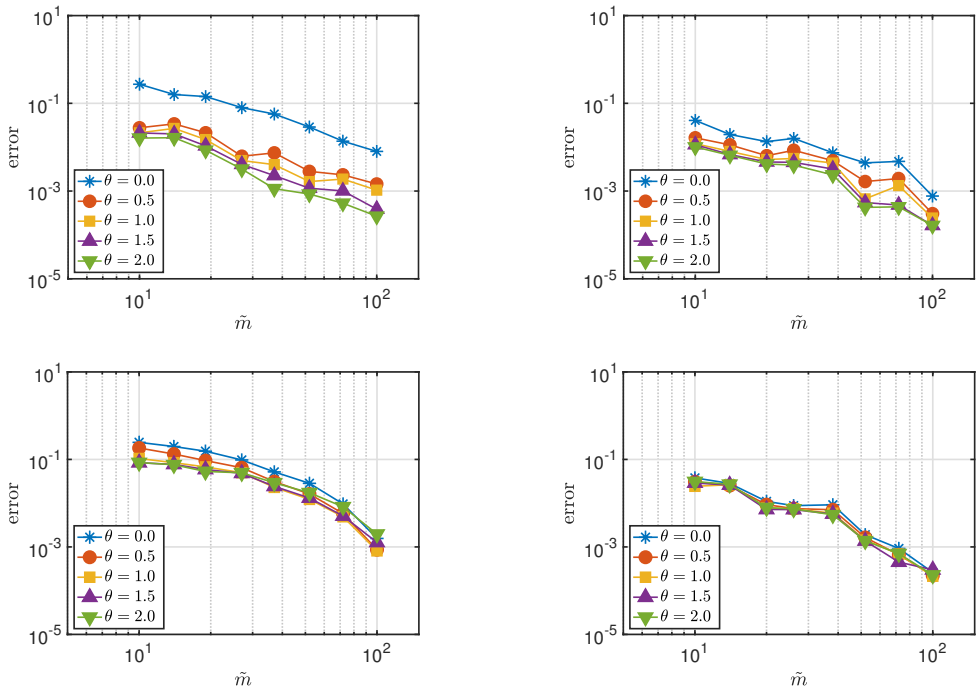


Figure 4.3: The $\|q - \tilde{q}\|_{L^\infty}$ recovery error of the two-dimensional diffusion equation against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density (top) and Chebyshev polynomials with points drawn from the Chebyshev density (bottom). The unaugmented case is shown on the left column and the gradient-augmented case is shown on the right.

4.4.3 The cookie problem

Our final example models how heat distributes inside of an oven with baked goods, which is defined by

$$\begin{aligned} -\nabla \cdot (a(\boldsymbol{y}, \boldsymbol{x}))\nabla u(\boldsymbol{x})) &= f, && \forall \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) &= 0, && \forall \boldsymbol{x} \in \partial\Omega. \end{aligned}$$

Here, the physical domain is $\Omega = (0,1)^2$, $\boldsymbol{x} = (x_1, x_2)$ is the two-dimensional physical variable. The parametric diffusion coefficient is given as

$$a(\boldsymbol{y}, \boldsymbol{x}) = 1 - \sum_{i=1}^{8} \mathbb{1}_{\Omega_i}(\boldsymbol{x})(0.5 + 0.49 y_i), \quad \forall \boldsymbol{x} \in \Omega, \ \forall \boldsymbol{y} \in \mathcal{Y},$$

where $\Omega_1, \ldots \Omega_8$ are circular subregions of Ω of radius 0.14, centered at $(0.5 \pm 0.3, 0.5 \pm 0.3)$, $(0.5, 0.5 \pm 0.3)$, $(0.5 \pm 0.3, 0.5)$. The parameter vector $\boldsymbol{y} = (y_1, \ldots, y_d) \in \mathcal{Y}$, where $\mathcal{Y} = (-1, 1)^8$. The forcing term on the right-hand side is defined as $f = 100 \cdot \mathbb{1}_{\Omega_f}$, where $\Omega_f = [0.4, 0.6]^2$ is a small square in the center of the domain Ω. This problem is referred as the cookie problem in the literature, where the square Ω_f models the central heating source

of an oven and $(\Omega_i)_{i=1}^8$ represents the cookies baking inside of the oven. This problem and analogous versions of it have been studied in various literatures, see [3, 12, 13, 40]. Here, the QoI to approximate is defined as $q(\boldsymbol{y}) = u(\boldsymbol{y}, (0.5, 0.8))$.

We discretize the problem by taking 20 equal spaced intervals along each boundary of Ω and on the eight circles $(\Omega_i)_{i=1}^8$, and taking 5 equal spaced intervals along each boundary of Ω_f. With this discretization, we have in total 1172 triangles with $N = 627$ vertices. Figure 4.4 shows the triangular mesh for the cookie problem.

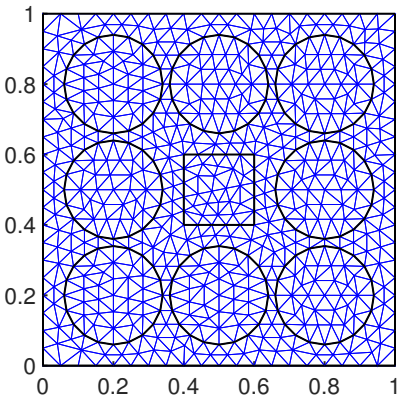


Figure 4.4: Triangular mesh for the cookie problem.

As the next step, we form the matrix equation $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$ to solve. Since $\boldsymbol{x} = (0.5, 0.8)$ corresponds to the 157th vertex, the QoI at $\bar{\boldsymbol{y}}$ can be written as $q(\bar{\boldsymbol{y}}) = \boldsymbol{e}_{157}^T \bar{\boldsymbol{u}}$. It follows that the adjoint equation is $\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{e}_{157}$ for $\boldsymbol{\lambda}^T \in \mathbb{R}^N$. Moreover, it can be seen that, for this problem, $\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}})$ can be simply computed using (4.3.17) with

$$a_k(\bar{\boldsymbol{y}}, \boldsymbol{x}) = -0.49 \times \mathbb{1}_{\Omega_k}(\boldsymbol{x}), \quad \text{for } k = 1, \ldots, 8.$$

Finally, we have the gradient of the QoI at $\bar{\boldsymbol{y}}$ to be

$$\frac{dq}{dy_k}(\bar{\boldsymbol{y}}) = \boldsymbol{\lambda}\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}),$$

for $k = 1, \ldots, 8$.

Figure 4.5 shows the approximation error $\|q - \tilde{q}\|_{L^\infty}$ against $\tilde{m}$ with $s = 23$ when either Legedre or Chebyshev approximating polynomial basis are used. As have been seen in the previous two examples, compared to only sample the QoI, a smaller approximation error is obtained when the gradient samples are also considered. Moreover, compared to the unweighted ℓ^1 minimization, we see a better approximation result when a weighted ℓ^1 minimization problem is solved.

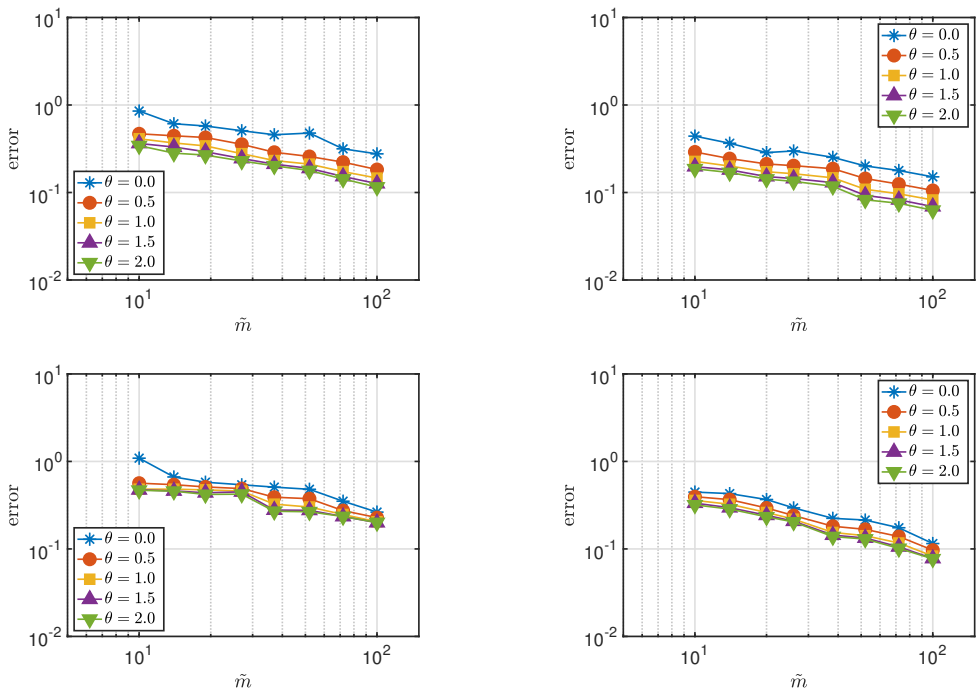


Figure 4.5: The $\|q - \tilde{q}\|_{L^\infty}$ recovery error of the cookie problem against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density (top) and Chebyshev polynomials with points drawn from the Chebyshev density (bottom). The unaugmented case is shown on the left column and the gradient-augmented case is shown on the right.

4.5 Parametric diffusion equation with mixed boundary conditions

In §4.3, we considered the parametric diffusion problem with homogenous Dirichlet boundary conditions. However, in reality, the physical problems we are dealing with often have more complicated boundary conditions. In this section, we will work on a more realistic situation when mixed boundary conditions are applied to the parametric diffusion equation.

The parametric diffusion equation with mixed boundary conditions is defined by

$$-\nabla \cdot (a(\boldsymbol{y}, \boldsymbol{x}) \nabla u(\boldsymbol{x})) = f(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \Omega, \tag{4.5.1}$$

$$u(\boldsymbol{x}) = c, \qquad \forall \boldsymbol{x} \in \Gamma_D, \tag{4.5.2}$$

$$\frac{\partial u(\boldsymbol{x})}{\partial n} = 0, \qquad \forall \boldsymbol{x} \in \Gamma_N, \tag{4.5.3}$$

where Ω is an open and bounded physical domain in $\mathbb{R}^n$ and $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is the n-dimensional physical variable. The parametric diffusion coefficient satisfies $a(\boldsymbol{y}, \boldsymbol{x}) > 0$, where $\boldsymbol{y} = (y_1, y_2, \ldots, y_d) \in D$ is a d-dimensional parameter vector and the parameter domain D is an open subset of $\mathbb{R}^d$ with a probability measure defined on it. The function $\boldsymbol{y} \to a(\boldsymbol{y}, \cdot)$ is continuously Fréchet differentiable for all $\boldsymbol{y} \in D$. Moreover, the parametric

diffusion coefficient $a(\boldsymbol{y}, \boldsymbol{x})$ is bounded, that is

$$M_1 \leq a(\boldsymbol{y}, \boldsymbol{x}) \leq M_2, \quad \forall \boldsymbol{x}, \boldsymbol{y},$$

where M_1, M_2 are positive real numbers. The forcing term on the right-hand side of (4.5.1) satisfies $f(\boldsymbol{x}) \in L^2(\Omega)$. The boundary $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$. On boundary Γ_D, the solution satisfies the non-homogenous Dirichlet boundary condition $u(\boldsymbol{x}) = c$, where c is a non-zero real constant. And, on boundary Γ_N, the homogenous Neumann condition $\frac{\partial u(\boldsymbol{x})}{\partial n} = \nabla u(\boldsymbol{x}) \cdot \boldsymbol{n} = 0$ is applied. The quantity of interest (QoI) is $q(\boldsymbol{y}) = Q(u(\boldsymbol{y}))$, where Q is a bounded linear functional acting on u.

At a given point $\bar{\boldsymbol{y}} \in D$, we have that the weak problem of (4.5.1) with boundary conditions (4.5.2) and (4.5.3) satisfies

$$\int_\Omega (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla \bar{u}(\boldsymbol{x})) \cdot \nabla v(\boldsymbol{x}) d\Omega = \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) d\Omega, \qquad \forall v \in H^1_{\Gamma_D}, \tag{4.5.4}$$

since $\frac{\partial \bar{u}(\boldsymbol{x})}{\partial n} = 0$ on Γ_N. The space $H^1_{\Gamma_D}$ is defined by

$$H^1_{\Gamma_D} = \{w \in H^1(\Omega) : w = 0 \text{ on } \Gamma_D\},$$

and the weak solution satisfies

$$\bar{u} \in \mathcal{U}_c = \{w \in H^1(\Omega) : w = c \text{ on } \Gamma_D\}.$$

However, this weak problem (4.5.4) is hard to be solved directly since $\bar{u}$ and v are defined in different spaces and the space $\mathcal{U}_c$ is not a vector space [105, §3.3.3]. Instead, we write the solution $\bar{u}(\boldsymbol{x})$ of (4.5.4) as

$$\bar{u}(\boldsymbol{x}) = \bar{u}_0(\boldsymbol{x}) + u_c(\boldsymbol{x}) = \bar{u}_0(\boldsymbol{x}) + c,$$

where $u_c(\boldsymbol{x}) = c$ is the lifting function satisfying the boundary condition on Γ_D (shown as (4.5.2)) and $\bar{u}_0 \in H^1_{\Gamma_D}$ is the solution of the weak problem

$$B[\bar{u}_0, v] = (T, v), \quad \forall v \in H^1_{\Gamma_D}, \tag{4.5.5}$$

where the bilinear mapping

$$B[\bar{u}_0, v] = \int_\Omega (a(\bar{\boldsymbol{y}}, \boldsymbol{x}) \nabla \bar{u}_0(\boldsymbol{x})) \cdot \nabla v(\boldsymbol{x}) d\Omega$$

for $\bar{u}_0, v \in H^1_{\Gamma_D}$. The linear functional $T : H^1_{\Gamma_D} \to \mathbb{R}$ is defined by

$$(T, v) = \int_\Omega f(\boldsymbol{x}) v(\boldsymbol{x}) d\Omega = \langle f, v \rangle,$$

where $\langle f, v\rangle$ is the $L^2(\Omega)$ inner product. With this set-up, at a given point $\bar{\boldsymbol{y}}$, the QoI can be written as $q(\bar{\boldsymbol{y}}) = Q(\bar{u}) = Q(\bar{u}_0 + c)$. Moreover, the gradient of the QoI can be written as

$$\frac{\partial q}{\partial \boldsymbol{y}}(\bar{\boldsymbol{y}}) = \frac{\partial q_0}{\partial \boldsymbol{y}}(\bar{\boldsymbol{y}}) \quad \text{with } q_0(\boldsymbol{y}) = Q(u_0(\boldsymbol{y})),$$

since c is a constant.

Next, we will show how to calculate the gradient of $q_0(\boldsymbol{y})$ at $\bar{\boldsymbol{y}}$ with the adjoint sensitivity analysis method. Similar to what we have in §4.3.2, now we have an operator

$$F : D \times H^1_{\Gamma_D} \to (H^1_{\Gamma_D})',$$

which is defined by

$$(F(\boldsymbol{y}, u_0), v) = \left(-\nabla \cdot (a(\boldsymbol{y}, \boldsymbol{x})\nabla u_0), v\right) - (T, v) = \langle a(\boldsymbol{y}, \boldsymbol{x})\nabla u_0, \nabla v\rangle - \langle f, v\rangle,$$

for $F(\boldsymbol{y}, u_0) \in (H^1_{\Gamma_D})'$ and for all $v \in H^1_{\Gamma_D}$. The two open subsets of the Banach spaces in IFT are taken to be $\mathcal{Y} = D \subset \mathcal{X} = \mathbb{R}^d$ and $\mathcal{U} = H^1_{\Gamma_D} = \mathcal{V}$. With the same procedure shown in §4.3.2, we can prove $F \in C^1(\mathcal{Y} \times \mathcal{U}, (H^1_{\Gamma_D})')$ and the partial Fréchet derivative $\frac{\partial F}{\partial u_0}(\bar{\boldsymbol{y}}, \bar{u}_0) : \mathcal{U} \to (H^1_{\Gamma_D})'$ is invertible. Then, we can apply the IFT and get

$$\frac{dq_0}{d\boldsymbol{y}}(\bar{\boldsymbol{y}}) = \lambda \frac{\partial F}{\partial \boldsymbol{y}}(\bar{\boldsymbol{y}}, \bar{u}_0), \tag{4.5.6}$$

where the linear functional $\lambda \in ((H^1_{\Gamma_D})')'$ is the solution to

$$\lambda G_{\bar{\boldsymbol{y}}} = -Q, \tag{4.5.7}$$

since the linear operator $\frac{\partial F}{\partial u_0}(\bar{\boldsymbol{y}}, \bar{u}_0) = G_{\bar{\boldsymbol{y}}}$ and the linear operator $\frac{\partial Q}{\partial u_0}(\bar{\boldsymbol{y}}, \bar{u}_0) = Q$. The linear operator $G_{\bar{\boldsymbol{y}}} : \mathcal{U} \to (H^1_{\Gamma_D})'$ is defined by

$$(G_{\bar{\boldsymbol{y}}}(\bar{u}_0), v) = \left(-\nabla \cdot (a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla \bar{u}_0), v\right) = \langle a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla \bar{u}_0, \nabla v\rangle.$$

Because the operator $G_{\bar{\boldsymbol{y}}}$ is self-adjoint on $\mathcal{U}$, we have the adjoint equation of (4.5.7) as

$$G_{\bar{\boldsymbol{y}}}\lambda^* = -Q^*, \tag{4.5.8}$$

where the bounded linear operators $\lambda^* : \mathbb{R} \to \mathcal{U}$, $G_{\bar{\boldsymbol{y}}} : \mathcal{U} \to (H^1_{\Gamma_D})'$, and $Q^* : \mathbb{R} \to (H^1_{\Gamma_D})'$.

With the Galerkin method, described in §4.3.3, we can discretize the weak problem (4.5.5) into a matrix equation

$$\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}, \tag{4.5.9}$$

where

$$\boldsymbol{U} = \left(\int_\Omega a(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla\varphi_i(\boldsymbol{x})\nabla\varphi_j(\boldsymbol{x})d\Omega\right)_{i,j=1}^{N} \in \mathbb{R}^{N\times N},$$
$$\boldsymbol{b} = \left(\int_\Omega f(\boldsymbol{x})\varphi_i(\boldsymbol{x})d\Omega\right)_{i=1}^{N} \in \mathbb{R}^{N},$$
$$\bar{\boldsymbol{u}} = (u_1, u_2, \ldots, u_N)^T \in \mathbb{R}^N.$$

Note that the matrix equation (4.5.9) has a unique solution $\bar{\boldsymbol{u}} = \boldsymbol{U}^{-1}\boldsymbol{b}$. Now we can write the QoI at $\bar{\boldsymbol{y}}$ as $q(\bar{\boldsymbol{y}}) = Q_h(\bar{\boldsymbol{u}} + \boldsymbol{c}) = \boldsymbol{a}^T(\bar{\boldsymbol{u}} + \boldsymbol{c})$ for some fixed column vector $\boldsymbol{a} \in \mathbb{R}^N$. The column vector $\boldsymbol{c} \in \mathbb{R}^N$ has value c at each position. Following the same steps as shown in §4.3.3, we have the discretized version of (4.5.7) as

$$\boldsymbol{\lambda}\boldsymbol{U} = -\boldsymbol{a}^T,$$

where $\boldsymbol{\lambda} : \mathbb{R}^N \to \mathbb{R}$ is the discretized version of the functional $\lambda \in ((H^1_{\Gamma_D})')'$. The discretized adjoint equation (4.5.8) becomes

$$\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{a},$$

where $\boldsymbol{\lambda}^T : \mathbb{R} \to \mathbb{R}^N$. Furthermore, we have that

$$\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}) = \boldsymbol{U}_k\bar{\boldsymbol{u}}, \tag{4.5.10}$$

where

$$\boldsymbol{U}_k = \left(\int_\Omega a_k(\bar{\boldsymbol{y}}, \boldsymbol{x})\nabla\varphi_i(\boldsymbol{x})\nabla\varphi_j(\boldsymbol{x})d\Omega\right)_{i,j=1}^{N,N} \in \mathbb{R}^{N\times N} \quad \text{with } a_k = \frac{\partial a(\bar{\boldsymbol{y}}, \boldsymbol{x})}{\partial y_k},\ k = 1, \ldots, d.$$

Thus, the gradient sample of the QoI at $\bar{\boldsymbol{y}}$ can be computed as

$$\frac{dq}{dy_k}(\bar{\boldsymbol{y}}) = \frac{dq_0}{dy_k}(\bar{\boldsymbol{y}}) = \boldsymbol{\lambda}\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}),$$

for $k = 1, \ldots, d$.

4.6 Numerical results for mixed boundary problems

In this section, we show the approximation results for the quantities of interest (QoIs) of two parametric diffusion equations with mixed boundary conditions. For details on how these problems are solved numerically, see Appendix A.

4.6.1 One-dimensional diffusion equation

Let us first look at a one-dimensional parametric diffusion equation defined as

$$-\frac{d}{dx}\left(e^y \frac{d}{dx}u(x)\right) = f(x), \quad x \in \Omega = (0,1),$$
$$u(0) = 1,\ u'(1) = 0.$$

where the parameter $y \in \mathcal{Y} = (-1,1)$ and the right-hand side $f(x) = x(x+1) \in L^2(\Omega)$. The QoI is defined by $q(y) = u(y, 0.18)$.

We denote the solution of this one-dimensional problem at $\bar{y}$ by $\bar{u}(x) = \bar{u}_0(x) + 1$, where $\bar{u}_0 \in \mathcal{U}$ is the solution of the weak problem

$$B[\bar{u}_0, v] = \langle f, v\rangle, \quad \forall v \in \mathcal{U}. \tag{4.6.1}$$

Here, the bilinear mapping $B[\bar{u}_0, v] = \int_0^1 \left(e^{\bar{y}} \frac{d}{dx}\bar{u}_0(x)\right) \frac{d}{dx}v(x)$ and $\langle f, v\rangle$ is the standard L^2 inner product. We discretize the physical domain with 50 equal spaced subregions, then (4.6.1) can be written as a matrix equation $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$ and the QoI at the given point $\bar{y}$ is $q(\bar{y}) = \boldsymbol{e}_{10}^T(\bar{\boldsymbol{u}} + \boldsymbol{1})$. Given the adjoint equation $\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{e}_{10}$ and $\frac{\partial F_h}{\partial y}(\bar{y}, \bar{\boldsymbol{u}}) = \boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$, we have the derivative of the QoI at $\bar{y}$ as $\frac{dq}{dy}(\bar{y}) = \frac{dq_0}{dy}(\bar{y}) = \boldsymbol{\lambda}\frac{\partial F_d}{\partial y}(\bar{y}, \bar{\boldsymbol{u}}) = \boldsymbol{\lambda}\boldsymbol{b}$.

Figure 4.6 shows the approximation error $\|q - \tilde{q}\|_{L^\infty}$ against $\tilde{m}$ with $s = 2048$ when different kinds of approximating polynomial bases are used. As what we have expected, the approximation results are similar to the example shown in §4.4.1. With the same amount of computational cost, a smaller approximation error is obtained when additional gradient samples are also considered. Moreover, we see that, compared to the unweighted case, an improvement of approximation is often obtained when a weighted ℓ^1 minimization problem is solved.

4.6.2 The Darcy flow problem

Next, we will present an example of the Darcy flow problem, which is often used to model the porous media flow. Porous medium is a material containing a solid portion along with a network of pores. Porous media are commonly seen in nature. For instance, soil, gravel, sandstone and limestone are examples of porous media [62, §7.1] [74, Chpt. 1 & Chpt. 2].

The standard model for steady state porous media flow consists of Darcy's equation and the continuity equation for incompressible fluids, which are given as

$$\boldsymbol{q}(\boldsymbol{x}) + a(\boldsymbol{x})\nabla u(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}), \tag{4.6.2}$$
$$\nabla \cdot \boldsymbol{q}(\boldsymbol{x}) = 0, \tag{4.6.3}$$

with some boundary conditions. Here, the physical variable $\boldsymbol{x} \in \Omega \subset \mathbb{R}^n$, u denotes the pressure, coefficient a measures the permeability of the material, $\boldsymbol{q}$ is the velocity and $\boldsymbol{g}$ is

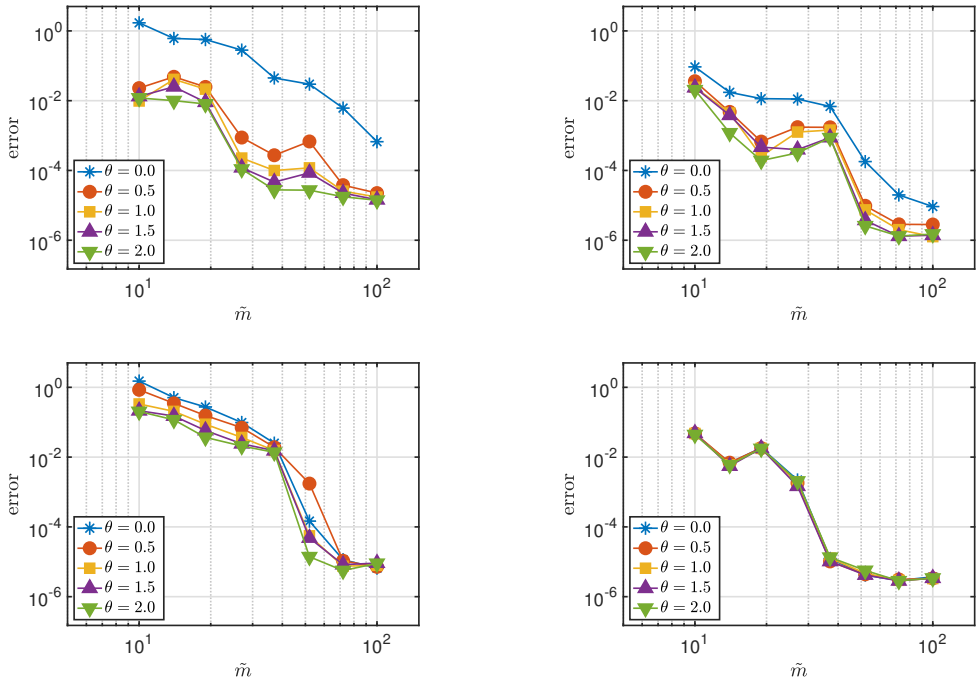


Figure 4.6: The $\|q - \tilde{q}\|_{L^\infty}$ recovery error of the one-dimensional diffusion equation with mixed boundary conditions against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density (top) and Chebyshev polynomials with points drawn from the Chebyshev density (bottom). The unaugmented case is shown on the left column and the gradient-augmented case is shown on the right.

the external source term [46]. The permeability mentioned here is an important property of porous media, which measures the ability for fluids to pass through the material [74, Chpt. 1]. If we combine (4.6.2) and (4.6.3) with some boundary conditions, then we get a Darcy flow problem, defined by

$$-\nabla \cdot (a(\boldsymbol{y}, \boldsymbol{x}) \nabla u(\boldsymbol{x})) = f(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \Omega, \tag{4.6.4}$$

$$u(\boldsymbol{x}) = 2, \qquad \forall \boldsymbol{x} \in \Gamma_D, \tag{4.6.5}$$

$$\frac{\partial u(\boldsymbol{x})}{\partial n} = 0, \qquad \forall \boldsymbol{x} \in \Gamma_N, \tag{4.6.6}$$

where $\Omega = (0, 1)^2$ is the physical domain and $\boldsymbol{x} = (x_1, x_2)$ is the two-dimensional physical variable. The parametric permeability coefficient is defined by

$$a(\boldsymbol{y}, \boldsymbol{x}) = \begin{cases} \exp(3y_k), & \boldsymbol{x} \in \Omega_k \text{ for } k = 1, \ldots, 5, \\ 10^{-4}, & \boldsymbol{x} \in \Omega \backslash \Omega_L, \end{cases}$$

where $\Omega_1, \ldots, \Omega_5$ are five circles with radius 0.15 centered at $(0.5, 0.5)$ and $(0.5 \pm 0.3, 0.5 \pm 0.3)$, and $\Omega_L = \cup_{k=1}^5 \Omega_k$. Here, these circles are constructed to model the pores inside the

medium. The parameter vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_5) \in \mathcal{Y} = (-1, 1)^5$. By defining $a(\boldsymbol{y}, \boldsymbol{x})$ this way, we see the permeability coefficient may jump more than four orders of magnitude from inside of the circles to outside of the circles, which is a typical behavior seen in porous media due to the variation of the material property inside of the domain [46] [74, Chpt. 1] [92, §4.3.3] [95]. We assume that there is randomness in the permeability, which takes into account the fact that permeability of the material may not be accurately known [92,95]. The right-hand side source term is defined by $f(\boldsymbol{x}) = -\nabla \cdot \boldsymbol{g}(\boldsymbol{x}) = 0.1 \in L^2(\Omega)$. Moreover, we label four edges of Ω counterclockwise as $\partial\Omega_1$, $\partial\Omega_2$, $\partial\Omega_3$, $\partial\Omega_4$ starting with the bottom edge. We have non-homogenous Dirichlet boundary condition $u(\boldsymbol{x}) = 2$ on edges $\Gamma_D = \partial\Omega_2 \cup \partial\Omega_4$ and homogenous Neumann boundary condition on edges $\Gamma_N = \partial\Omega_1 \cup \partial\Omega_3$. By defining boundary conditions this way, we ensure a steady state flow from left to right [92, 95]. Analogous examples of this one have been studied in various literatures, see [46, 92, 95], for instance. The QoI is $q(\boldsymbol{y}) = u(\boldsymbol{y}, (0.6, 0.2))$.

For a given $\bar{\boldsymbol{y}}$, we write the solution of this problem as $\bar{u}(\boldsymbol{x}) = \bar{u}_0(\boldsymbol{x}) + 2$, where $\bar{u}_0(\boldsymbol{x}) \in \mathcal{U}$ is the solution of the weak problem (4.5.5). To solve this weak problem, we discretize the domain by taking 20 equal spaced intervals on each edge of Ω and on those five circles $(\Omega_i)_{i=1}^5$. Then, we get in total 1020 triangles with $N = 551$ vertices. Figure 4.7 shows the triangular mesh for this Darcy flow problem.

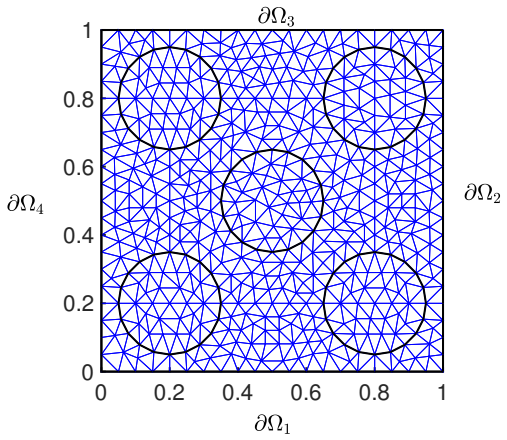


Figure 4.7: Triangular mesh for the Darcy flow problem.

With this discretization, we can form a matrix equation $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$ of the weak problem (4.5.5) and the QoI at $\bar{\boldsymbol{y}}$ can be written as $q(\bar{\boldsymbol{y}}) = \boldsymbol{e}_{89}^T(\bar{\boldsymbol{u}} + \boldsymbol{2})$. Moreover, the discretized adjoint equation can be written as $\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{e}_{89}$ for $\boldsymbol{\lambda}^T \in \mathbb{R}^N$. For this problem, we also have

$$a_k(\bar{\boldsymbol{y}}, \boldsymbol{x}) = 3 \times \exp(3y_k) \times \mathbb{1}_{\Omega_k}(\boldsymbol{x}),$$

for $k = 1, \ldots, 5$. Then, along with (4.5.10), we have the gradient of the QoI at $\bar{\boldsymbol{y}}$ as

$$\frac{dq}{dy_k}(\bar{\boldsymbol{y}}) = \frac{dq_0}{dy_k}(\bar{\boldsymbol{y}}) = \boldsymbol{\lambda}\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}),$$

for $k = 1, \ldots, 5$.

Figure 4.8 shows the approximation error $\|q - \tilde{q}\|_{L^\infty}$ against computational cost $\tilde{m}$ with $s = 58$, when different kinds of approximating polynomial bases are used. As in the previous example, a smaller approximation error is obtained for the gradient-augmented case than the unaugmented case when the same amount of computational cost is used. Moreover, we see that, compared to the unweighted case, an improvement of approximation result is achieved when a weighted ℓ^1 minimization problem is solved.

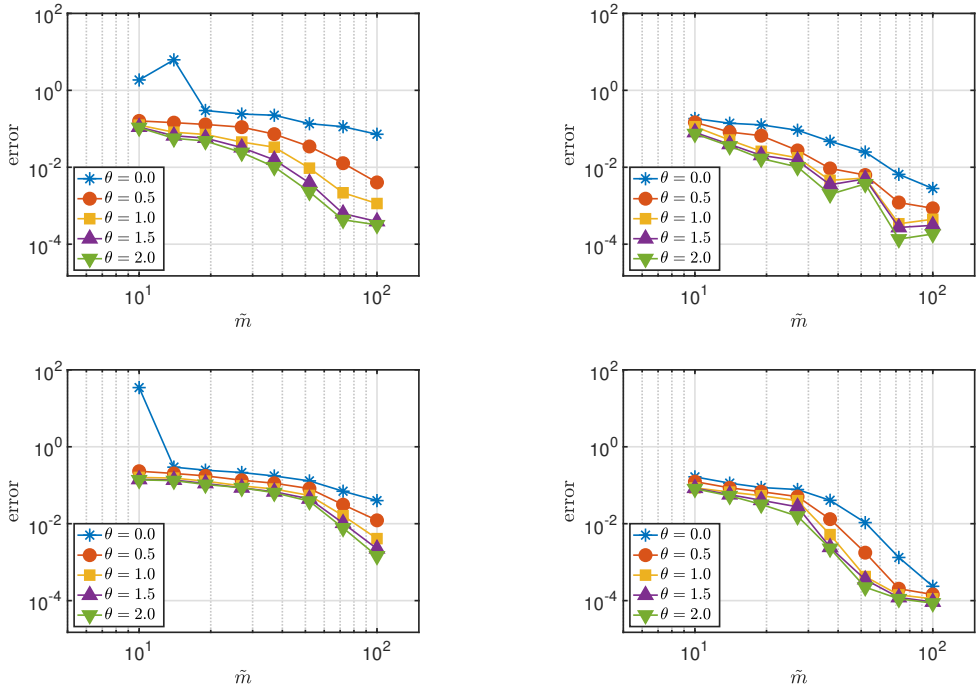


Figure 4.8: The $\|q - \tilde{q}\|_{L^\infty}$ recovery error of the Darcy flow problem against $\tilde{m}$ for Legendre polynomials with points drawn from the uniform density (top) and Chebyshev polynomials with points drawn from the Chebyshev density (bottom). The unaugmented case is shown on the left column and the gradient-augmented case is shown on the right.

4.7 Computational cost

When calculating the computational cost in §4.4 and §4.6, we simply assumed that it takes the same amount of time to generate samples of the quantity of interest (QoI) and the gradient samples of the QoI with the adjoint sensitivity analysis method, which was based on what is indicated in [102]. As what we have seen in §4.3.3 and §4.5, in order to generate

one sample of the QoI, the most cost intensive step is to solve the matrix equation $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$. To generate one sample of the gradient of the QoI with the adjoint sensitivity analysis method, in addition to $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$, we need to solve another matrix equation $\boldsymbol{U}\boldsymbol{\lambda}^T = -\boldsymbol{a}$. In other words, theoretically, we should expect to see that it takes about double amount of the time to generate one sample of the QoI along with one sample of the gradient of the QoI, compared to the time used to generate one sample of the QoI only, when the adjoint sensitivity analysis method is applied. In the section, we will perform further investigation on whether the computational cost for getting the gradient samples is indeed about the same as the cost for getting the QoI samples. We also want to show that the computational cost for generating the gradient samples changes mildly with the dimension of the parameter space d.

The investigation is performed by considering the one-dimensional homogenous Dirichlet boundary value problem:

$$\begin{aligned} -\frac{d}{dx}\left(a(\boldsymbol{y},x)\frac{d}{dx}u(x)\right) &= f(x), && x \in \Omega \\ u &= 0 && x \in \partial\Omega, \end{aligned}$$

where the physical domain $\Omega = (0,1)$ and the right-hand side source term $f(x) = x(x+1) \in L^2(\Omega)$. We define the random diffusion coefficient as a truncated Karhunen-Loève (KL) expansion [63, §2.3], given by

$$a(\boldsymbol{y},x) = e^{\gamma_d(\boldsymbol{y},x)}, \qquad \gamma_d(\boldsymbol{y},x) = \sum_{k=1}^{d} \xi_k \tau_k(x) y_k, \quad \forall x, \boldsymbol{y}.$$

For comparison purpose, we consider two different cases of ξ_k:

$$\begin{aligned} C1: &\quad \xi_k = \sqrt{3}\exp(-k), \\ C2: &\quad \xi_k = \frac{\sqrt{3}}{k}. \end{aligned}$$

Here, $\boldsymbol{y} = (y_1, y_2, \ldots, y_d) \in D$ and $D = (-1,1)^d$. The trigonometric functions τ_k are defined by

$$\tau_k(x) = \begin{cases} \sin\left(\frac{k}{2}\pi x\right) & \text{if } k \text{ is even}, \\ \cos\left(\frac{k-1}{2}\pi x\right) & \text{if } k \text{ is odd}. \end{cases}$$

Elliptic problems with parametric diffusive coefficients, which are defined as a truncated KL expansion, have been widely studied. See [11, 46, 72], for instance. Here, the QoI to approximate is $q(\boldsymbol{y}) = u(\boldsymbol{y}, 0.18)$. We generate samples of $\boldsymbol{y}$ independently and identically with respect to the uniform measure.

As with the example shown in §4.4.1, we discretize the physical domain with $N = 51$, and form the matrix equation $\boldsymbol{U}\bar{\boldsymbol{u}} = \boldsymbol{b}$. In this case, we have $q(\bar{\boldsymbol{y}}) = \boldsymbol{e}_{10}^T\bar{\boldsymbol{u}} = \boldsymbol{e}_{10}^T(\boldsymbol{A}^{-1}\boldsymbol{b})$. The discretized adjoint equation becomes $\boldsymbol{A}\boldsymbol{\lambda}^T = -\boldsymbol{e}_{10}$ for $\boldsymbol{\lambda}^T \in \mathbb{R}^N$. For both cases, we can compute $\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}})$ using (4.3.17) with $a_k(\bar{\boldsymbol{y}}, x)$ defined by

$$C1: \quad a_k(\bar{\boldsymbol{y}}, x) = \left(\sqrt{3}\exp(-k)\tau_k(x)\right) \times e^{\gamma_d(\boldsymbol{y},x)},$$
$$C2: \quad a_k(\bar{\boldsymbol{y}}, x) = \left(\frac{\sqrt{3}}{k}\tau_k(x)\right) \times e^{\gamma_d(\boldsymbol{y},x)},$$

for $k = 1, \ldots, d$. Then, the gradient is computed as

$$\frac{dq}{dy_k}(\bar{\boldsymbol{y}}) = \boldsymbol{\lambda}\frac{\partial F_h}{\partial y_k}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{u}}),$$

for $k = 1, \ldots, d$.

We record the computational time for generating the QoI samples and the computational time for generating both the QoI and the gradient of the QoI samples for various values of the dimension d. We compute the computational time ratio as

$$\text{ratio} = \frac{\text{time used to generate the QoI and the gradient of the QoI samples}}{\text{time used to generate the QoI samples}}.$$

Figure 4.9 shows the box plot for the computational ratio over various values of d, when 500 samples of the QoI and 500 samples of the gradient of the QoI are generated on FreeFem++ (version 3.61) [73]. Here, bottom and top edges of the box indicate the 25th and 75th percentiles respectively and the central red mark in the box is the median. The results for C_1 are shown on the left and C_2 on the right. In table 4.1, we present the average computational time ratio for generating those 500 samples when different values of d are considered. The results for C_1 are shown on the top and C_2 on the bottom. Regardless of the number of dimension d, we see that, most of the time, the computational time ratio is less than 3 for both cases. Moreover, for both cases, the average computational time ratio is always around 2.7 when the dimension of the parameter space d is varying. These indicate that it takes roughly about the same amount of effort to generate the QoI samples and the gradient samples of the QoI with the adjoint sensitivity analysis method. Moreover, the computational cost for generating these samples changes very mildly with the dimension of the parameter space d.

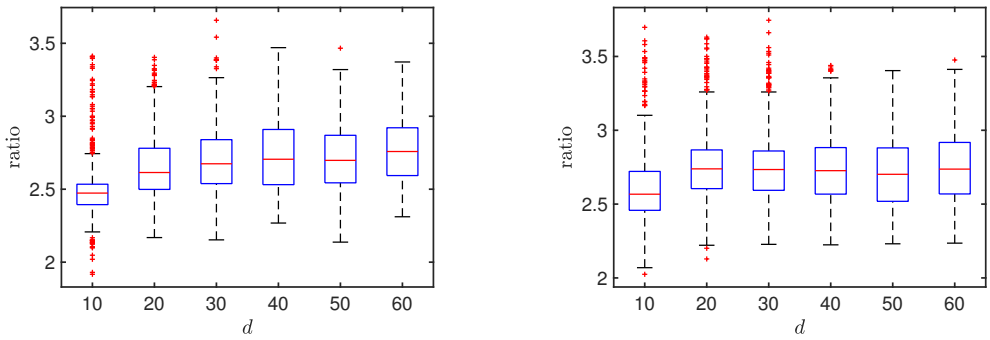


Figure 4.9: The box plot of the computational time ratio against d. The result for $C1$ shows on the left and $C2$ on the right.

d	10	20	30	40	50	60
average time ratio	2.50	2.65	2.71	2.74	2.71	2.77

d	10	20	30	40	50	60
average time ratio	2.61	2.76	2.76	2.75	2.71	2.75

Table 4.1: The table of the average computational time ratio for various values of d. The result for $C1$ shows on the top and $C2$ on the bottom.

Chapter 5

Conclusions and future work

In this thesis, we applied the weighted ℓ^1 minimization techniques in CS to the multiple measurement vector (MMV) problem and the high-dimensional function approximation problem. In Chapter 2, we introduced the variance-based joint sparse (VBJS) method based on weighed ℓ^1 minimization to solve the MMV problem. Unlike the standard $\ell^{2,1}$ minimization approach commonly used to solve the MMV problem, this VBJS method is easily parallelizable. Thus, we expect the VBJS method to be much more efficient compared to the $\ell^{2,1}$ minimization method, particularly for problems with a large number of vectors to recover. As demonstrated through various synthetic numerical experiments on randomly generated sparse vectors, one application to one-dimensional signal recovery and one application to parallel Magnetic Resonance Imaging (MRI), we see that the VBJS method often achieves the same accuracy as the standard $\ell^{2,1}$ minimization approach with fewer measurements. However, it has been seen that the VBJS method is not a panacea for all types of image recovery problems. We have seen that the VBJS method gives a less accurate result compared to the $\ell^{2,1}$ minimization method for recovering the colored Shepp-Logan phantom image. However, the VBJS method has some other uses where it shows promise. For an application to edge detection, see [6].

In Chapter 3, we studied the problem of high-dimensional function approximation with sparse polynomial expansions. In particular, we worked on the approximation problem when both function values and its gradient are sampled. By assuming the computational cost for generating the gradient samples equals to the computational cost for generating function samples, we see numerically that the approximation result from gradient-augmented measurements often gives a smaller error than the case of function samples only. Various nonuniform recovery guarantees are also presented in Chapter 3. We have proved that, with the same sample complexity as the unaugmented case, gradient-augmented measurements permit an error bound in a stronger Sobolev norm as opposed to an L^2 norm. Moreover, for tensor Jacobi polynomials, if we solve a gradient-augmented weighted ℓ^1 minimization problem under the lower sets assumption, then the sample complexity is only a polynomial in s and logarithmic in the dimension d. Thus, the curse of dimensionality is mitigated.

A further demonstration of the benefit of adding the gradient samples for approximating the quantities of interest (QoIs) of parametric differential equations was presented in Chapter 4. In all numerical experiments, we see that, with additional gradient samples, the QoI can be more accurately approximated than with function samples only when the same amount of computational cost is used. Moreover, we see the weighted ℓ^1 minimization problem gives an improved approximation result compared to the unweighted ℓ^1 minimization problem for both gradient-augmented and unaugmented problems. A simple comparison of the computational cost for generating the QoI samples and the computational cost for generating its gradient samples was also conducted in Chapter 4. We see that, by using the adjoint sensitivity analysis method, it takes about the same amount of computational time to generate samples of the QoI as to generate the gradient samples of the QoI. Moreover, computational time for generating the gradient samples is independent of the dimension of the parameter space d.

Beyond what has been completed in this thesis, there are numerous topics left as future work. There are two potential topics related to the MMV problem. The first one is to explore other weighting strategies. As seen from the experiment of color image recovery, the VBJS method with reciprocal weights gives a worse result than the $\ell^{2,1}$ minimization method. It suggests us to explore other weighting strategy so that an improved performance of the VBJS method can be obtained. The second one is to apply the VBJS method to other problems. For instance, hyperspectral image recovery can be a potential application for the VBJS method.

There are also several topics related to the high-dimensional function approximation problem left as future work. As mentioned before, the recovery guarantees obtained in Chapter 3 for the high-dimensional function approximation problem with the gradient-augmented weighted ℓ^1 minimization are all nonuniform recovery guarantees. For first future topic, we should extend those results to uniform recovery guarantees. In [43], based on the concept of lower restricted isometry property (RIP), uniform recovery guarantees for the unaugmented weighted ℓ^1 minimization problem on lower sets were proved. We expect that a similar procedure could be applied to obtain uniform recovery guarantees for the gradient-augmented problem. The second future topic is to explore other ways to exploit the gradient information. For example, in [112] Tang used a basis pursuit-type technique to approximate the high-dimensional function with the gradient-augmented measurements. Moreover, in [112], Tang only considered the case of recovering with Legendre polynomial expansion. Extending his work to other types of orthogonal polynomial expansion, such as Chebyshev polynomial expansion, also left as future work. The third future topic is to examine the gradient-augmented weighted ℓ^1 minimization method on more complicated parametric partial differential equation (PDE) problems. Recall that those examples we have worked on in Chapter 4 are all simple elliptic problems. As a next step, we should also perform experiments on more realistic problems. Problems in computational fluid dynamics

and shape optimization could be potential directions to work on, for instance. Finally, as a way to connect the MMV problem with the high-dimensional function approximation problem, a topic for future work is that of simultaneously approximating solutions of a high-dimensional parametric PDE with the VBJS method. As shown in Chapter 3, if the solution of a parametric PDE is analytic, then it can be expressed as a polynomial chaos expansion. Thus, to approximate the solution, the task is to reconstruct the corresponding coefficient vector of the polynomial expansion. As what has been shown in [51], we can then reformulate this coefficient vector reconstruction problem as a MMV problem, which could be solved with the VBJS method introduced in Chapter 2.

Bibliography

[1] B. Adcock. Infinite-dimensional ℓ^1 minimization and function approximation from pointwise data. *Constr. Approx.*, 45(3):345–390, 2017.

[2] B. Adcock. Infinite-dimensional compressed sensing and function interpolation. *Found. Comput. Math.*, 18(3):661–701, 2018.

[3] B. Adcock, A. Bao, and S. Brugiapaglia. Correcting for unknown errors in sparse high-dimensional function approximation. *Numer. Math.*, 142(3):667–711, 2019.

[4] B. Adcock, S. Brugiapaglia, and C. G. Webster. Compressed sensing approaches for polynomial approximation of high-dimensional functions. In H. Boche, G. Caire, R. Calderbank, M. März, G. Kutyniok, and R. Mathar, editors, *Compressed Sensing and Its Applications*, Applied and Numerical Harmonic Analysis, pages 93–124. Birkhäuser, Cham, 2017.

[5] B. Adcock and N. Dexter. The gap between theory and practice in function approximation with deep neural networks. *arXiv:2001.07523*, 2020.

[6] B. Adcock, A. Gelb, G. Song, and Y. Sui. Joint sparse recovery based on variances. *SIAM J. Sci. Comput.*, 41(1):A246–268, 2019.

[7] B. Adcock and A. C. Hansen. *Compressive Imaging: Structure, Sampling, Learning.* In preparation, 2020.

[8] B. Adcock and Y. Sui. Compressive hermite interpolation: sparse, high-dimensional approximation from gradient-augmented measurements. *Constr. Approx.*, 50(1):167 – 207, 2019.

[9] A. K. Aleseev, I. M. Navon, and M. E. Zelentsov. The estimation of functional uncertainty using polynomial chaos and adjoint equations. *Int. J. Numer. Meth. Fluids*, 67:328–341, 2011.

[10] S. Auliac. One-dimensional problems with freefem++, 2018. URL: `https://www.um.es/freefem/ff++/pmwiki.php?n=Main.ExampleBySilvainAuliac`. Last visited on 2019/12/22.

[11] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Rev.*, 52(2):317–355, 2010.

[12] J. Bäck, F. Nobile, L. Tamellini, and R. Tempone. Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison. In J. S. Hesthaven and E. M. Rønquist, editors, *Spectral and High Order Methods for Partial Differential Equations: Selected papers from the ICOSAHOM '09 conference, June 22-26, Trondheim, Norway*, pages 43–62. Springer Berlin Heidelberg, 2011.

[13] J. Ballani and L. Grasedyck. Hierarchical tensor approximation of output quantities of parameter-dependent PDEs. *SIAM/ASA J. Uncertain. Quantif.*, 3(1):852–872, 2015.

[14] R. Baraniuk, H. Choi, F. Fernandes, B. Hendricks, R. Neelamani, V. Ribeiro, J. Romberg, R. Gopinath, H. Guo, M. Lang, J. Odegard, and D. Wei. Rice Wavelet Toolbox. `https://www.ece.rice.edu/dsp/software/rwt.shtml`, December 2002.

[15] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk. Distributed compressive sensing. *IEEE Trans. Inform. Theory*, 52:5406–5425, 2006.

[16] J. Beck, F. Nobile, L. Tamellini, and R. Tampone. Convergence of quasi-optimal Stochastic Galerkin methods for a class of PDES with random coefficients. *Comput. Math. Appl.*, 67:732–751, 2014.

[17] D. Behmardi and E. D. Nayeri. Introduction of Fréchet and Gâteaux derivative. *Appl. Math. Sci.*, 2(20):975–980, 2008.

[18] J. Bell. Fréchet derivatives and Gâteaux derivatives, 2014. URL: `http://individual.utoronto.ca/jordanbell/notes/frechetderivatives.pdf`. Last visited on 2019/11/14.

[19] R. E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[20] J. Bigot, C. Boyer, and P. Weiss. An analysis of block sampling strategies in compressed sensing. *IEEE Trans. Inform. Theory*, 62(4):2125–2139, 2016.

[21] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing. Greedy algorithms for joint sparse recovery. *IEEE Trans. Signal Process.*, 62(7):1694–1704, 2014.

[22] L. Borcea and I. Kocyigit. A multiple measurement vector approach to synthetic aperture radar imaging. *SIAM J. Imag. Sci.*, 11(1):770–801, 2018.

[23] R. V. Borries, C. Miosso, and C. Potes. Compressed sensing using prior information. In *2nd IEEE Int, Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMPSAP 2007*, pages 121–124, 2007.

[24] C. Boyer, J. Bigot, and P. Weiss. Compressed sensing with structured sparsity and structured acquisition. *Appl. Comput. Harmon. Anal.*, 46:312–350, 2019.

[25] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, New York, NY, third edition, 2008.

[26] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Universitext. Springer New York, New York, NY, 2010.

[27] S. Brugiapaglia and B. Adcock. Robustness to unknown error in sparse regularization. *IEEE Trans. Inform. Theory*, 64(10):6638–6661, 2018.

[28] D. G. Cacuci. *Sensitivity and Uncertainty Analysis. Vol. I: Theory.* Chapman & Hall / CRC., Boca Raton, FL, 2003.

[29] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *C. R. Acad. Sci. Paris*, Ser. I 346:589–592, 2008.

[30] E. J. Candès and Y. Plan. A probabilistic and ripless theory of compressed sensing. *IEEE Trans. Inform. Theory*, 57(11):7235–7254, 2011.

[31] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59:1207–1223, 2005.

[32] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006.

[33] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, 2005.

[34] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans. Inform. Theory*, 52(12):5406–5425, 2006.

[35] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier. Anal. Appl.*, 14:877–905, 2008.

[36] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains.* Springer-Verlag Berlin Heidelberg, 2006.

[37] L. Chen. Sobolev spaces and elliptic equations, 2017. URL: `https://www.math.uci.edu/~chenlong/226/Ch1Space.pdf`. Last visited on 2019/10/08.

[38] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[39] Y. Chen, G. Li, and Q. Zhang. Modified multiple-measurement vector model for SAR imaging. In *IEEE Radar Conference*, pages 1–4, 2016.

[40] A. Chkifa, A. Cohen, G. Migliorati, and R. Tempone. Discrete least squares polynomial approximation with random evaluations-application to parametric and stochastic elliptic PDEs. *ESAIM: Math. Model. Numer. Anal.*, 49(3):815–837, 2015.

[41] A. Chkifa, A. Cohen, and C. Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Found. Comput. Math.*, 14:601–633, 2014.

[42] A. Chkifa, G. Cohen, and C. Schwab. Breaking the curse of dimensionality in sparse polynomial interpolation and applications to parametric PDEs. *J. Math. Pures Appl.*, 103:400–428, 2015.

[43] A. Chkifa, N. Dexter, H. Tran, and C. G. Webster. Polynomial approximation via compressed sensing of high-dimensional functions on lower sets. *Math. Comp.*, 87(311):1415–1450, 2018.

[44] I.-Y. Chun and B. Adcock. Compressed sensing and parallel acquisition. *IEEE Trans. Inform. Theory*, 63(8):4860–4882, 2017.

[45] I.-Y. Chun, B. Adcock, and T. Talavage. Efficient compressed sensing SENSE pMRI reconstruction with joint sparsity promotion. *IEEE Trans. Med. Imag.*, 31(1):354–368, 2016.

[46] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel monte carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Visual Sci.*, 14(3):3–15, 2011.

[47] A. Cohen, M. A. Davenport, and D. Leviatan. On the stability and accuracy of least squares approximations. *Found. Comput. Math.*, 13:819–834, 2013.

[48] A. Cohen and G. Migliorati. Optimal weighted least-squares methods. *SMAI J. Comput. Math.*, 3:181 – 203, 2017.

[49] A. Cohen and G. Migliorati. Multivariate approximation in downward closed polynomial spaces. In J. Dick, F. Y. Kuo, and H. Woźniakowski, editors, *Contemporary Computational Mathematics - A Celebration of the 80th Birthday of Ian Sloan*, pages 233–282. Springer International Publishing, Cham, 2018.

[50] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Trans. Signal Process.*, 55(7):2477–2488, 2005.

[51] N. Dexter, H. Tran, and C. Webster. On the strong convergence of forward-backward splitting in the reconstructing jointly sparse signals. *arXiv:1711.02591*, 2017.

[52] D. Donoho and J. Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philos. Trans. Royal Soc. A*, 367(1906):4273–4293, 2009.

[53] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparest solution. *Comm. Pure Appl. Math.*, 59(6):797–829, 2006.

[54] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inform. Theory*, 55:5302–5316, 2009.

[55] Y. C. Eldar and H. Rauhut. Average case analysis of multichannel sparse recovery using convex relaxation. *IEEE Trans. Inform. Theory*, 55:505–519, 2009.

[56] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, RI, second edition, 2010.

[57] P. Fischer. Introduction to Galerkin method, 2016. URL: `http://fischerp.cs.illinois.edu/tam470/refs/galerkin2.pdf`. Last visited on 2019/10/21.

[58] S. Foucart. Recovery jointly sparse vectors via hard thresholding pursuit. In *Proceedings of the 9th International Conference on Sampling Theory and Applications*, 2011.

[59] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, New York, NY, USA, 2013.

[60] M. P. Friedlander, H. Mansour, R. Saab, and Ö. Yilmaz. Recovering compressively sampled signals using partial support information. *IEEE Trans. Inform. Theory*, 58:1122–1134, 2012.

[61] Y. Fu, H. Li, Q. Zhang, and J. Zou. Block-sparse recovery via redundant block OMP. *Signal Process.*, 97:162–171, 2014.

[62] D. D. Ganji and S. H. H. Kachapi. *Application of Nonlinear Systems in Nanomechanics and Nanofluids: analytical methods and applications*. Micro and Nano Technologies. Elsevier Science, 2015.

[63] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer, New York, USA, revised edition, 1991.

[64] M. S. Gockenbach. *Understanding and Implementing the Finite Element Method*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[65] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, 2014.

[66] D. H. Griffel. *Applied Functional Analysis*. Ellis Horwood, Chichester, UK, 1981.

[67] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, E. Kiefer, and A. Haase. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magn. Reson. Med.*, 47:1202–1210, 2002.

[68] M. Guerquin-Kern, L. Lejeune, K. P. Pruessmann, and M. Unser. Realistic analytical phantoms for parallel Magnetic Resonance Imaging. *IEEE Trans. Med. Imag.*, 31(3):626–636, 2012.

[69] L. Guo, A. Narayan, and T. Zhou. A gradient enhanced ℓ^1 mininization for sparse approximation of polynomial chaos expansions. *J. Comput. Phys.*, 367:49–64, 2018.

[70] L. Guo, A. Narayan, T. Zhou, and Y. Chen. Stochastic collocation methods via ℓ_1 minimization using randomized quadratures. *SIAM J. Sci. Comput.*, 39(1):A333–A359, 2017.

[71] M. Hadigol and A. Doostan. Least squares polynomial chaos expansion: a review of sampling strategies. *Comput. Methods Appl. Mech. Eng.*, 332:382 – 407, 2018.

[72] A. Haji-Ali, F. Nobile, L. Tamellini, and R. Tempone. Multi-index stochastic collocation for random PDEs. *Comput. Methods. Appl. Mech. Engrg.*, 306:95–122, 2016.

[73] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.

[74] F. Hellman. *Numerical methods for Darcy flow problems with rough and uncertain data*. PhD thesis, Uppsala University, 2017.

[75] J. K. Hunter. Notes on partial differential equations, 2014. URL: `https://www.math.ucdavis.edu/~hunter/pdes/pde_notes.pdf`. Last visited on 2019/10/08.

[76] J. K. Hunter and B. Nachtergaele. *Applied Analysis*. World Scientific, River Edge, NJ, USA, 2001.

[77] V. Hutson, J. S. Pym, and M. J. Cloud. *Applications of Functional Analysis and Operator Theory*. Mathematics in science and engineering Volume 200. Elsevier, 2nd edition, 2005.

[78] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi. Weighted ℓ_1 minimization for sparse recovery with prior information. In *IEEE Int. Symp. Information Theory, ISIT 2009*, pages 483–487, 2009.

[79] E. Kreyszig. *Introductory Functional Analysis with Applications*. Wiley classics library. John Wiley & Sons, 1989.

[80] Y. Li, M. Anitescu, O. Roderick, and F. Hickernell. Orthogonal bases for polynomial regression with derivative information in uncertainty quantification. *Int. J. Uncertain. Quantif.*, 1(4):297–320, 2011.

[81] B. Lockwood and D. Mavriplis. Gradient-based methods for uncertainty quantification in hypersonic flows. *Comput. & Fluids*, 85:27–38, 2013.

[82] M. Lustig and J. M. Pauly. SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space. *Magn. Reson. Med.*, 64:457–471, 2010.

[83] A. Majumdar and R. K. Ward. Compressed sensing of color images. *Signal Process.*, 90:3122–3127, 2010.

[84] A. Majumdar and R. K. Ward. Joint reconstruction of multiecho MR images using correlated sparsity. *Magn. Reson. Imaging*, 29:899–906, 2011.

[85] A. Majumdar and R. K. Ward. Calibration-less multi-coil MR image reconstruction. *Magn. Reson. Imaging*, 7:1032–1045, 2012.

[86] A. Majumdar and R. K. Ward. Face recognition from video: An MMV recovery approach. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2221–2224, 2012.

[87] A. Majumdar and R. K. Ward. Rank awareness in group-sparse recovery of multi-echo MR images. *Sensors*, 13:3902–3921, 2013.

[88] A. Makhzani and S. Valaee. Reconstruction of jointly sparse signals using iterative hard thresholding. In *2012 IEEE International Conference on Communications (ICC)*, pages 3564–3568, 2012.

[89] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, New York, 2008.

[90] H. Mansour and R. Saab. Recovery analysis for weighted ℓ_1-minimization using the null space property. *Appl. Comput. Harmon. Anal.*, 43(1):23–38, 2017.

[91] MATLAB. *9.6.0.1099231 (R2019a)*. The MathWorks Inc., Natick, Massachusetts, 2019.

[92] G. Migliorati. *Polynomial approximation by means of the random discrete L^2 projection and application to inverse problems for PDEs with stochastic data*. PhD thesis, Ecole Polytechnique, 2013.

[93] G. Migliorati. Multivariate Markov-type and Nikolskii-type inequalities for polynomials associated with downward closed multi-index sets. *J. Approx. Theory*, 189:137–159, 2015.

[94] G. Migliorati and F. Nobile. Analysis of discrete least squares on multivariate polynomial spaces with evaluations at low-discrepancy point sets. *J. Complexity*, 31:517–542, 2015.

[95] G. Migliorati, F. Nobile, E. von Schwerin, and R. Tempone. Approximation of quantities of interest in stochastic PDEs by the random discrete L^2 projection on polynomial spaces. *SIAM J. Sci. Comput.*, 35(3):A1440–A1460, 2013.

[96] G. Migliorati, F. Nobile, E. von Schwerin, and R. Tempone. Analysis of the discrete L^2 projection on polynomial spaces with random evaluations. *Found. Comput. Math.*, 14:419–456, 2014.

[97] C. J. Miosso, R. von Borries, and J. H. Pierluissi. Compressive sensing with prior information: Requirements and probabilities of reconstruction in ℓ_1-minimization. *IEEE Trans. Signal Process.*, 61(9):2150–2164, 2013.

[98] H. Monajemi, S. Jafarpour, M. Gavish, Stat 330/CME 362 Collaboration, and D. L. Donoho. Deterministic matrices matching the compressed sensing phase transitions of Gaussian random matrices. *Proc. Natl. A. Sci. USA*, 110(4):1181–1186, Jan. 2013.

[99] D. Needell, R. Saab, and T. Woolf. Weighted ℓ_1-minimization for sparse recovery under arbitrary prior information. *Inf inference*, 6(3):284–309, 2017.

[100] D Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.*, 26(3):301–321, 2008.

[101] J. Peng, J. Hampton, and A. Doostan. A weighted ℓ_1-minimization approach for sparse polynomial chaos expansions. *J. Comput. Phys.*, 267:92–111, 2014.

[102] J. Peng, J. Hampton, and A. Doostan. On polynomial chaos expansion via gradient-enhanced ℓ_1-minimization. *J. Comput. Phys.*, 310:440–458, 2016.

[103] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger. SENSE: Sensitivity encoding for fast MRI. *Magn. Reson. Med.*, 42:952–962, 1999.

[104] D. Qu and C. Xu. Generalized polynomial chaos decomposition and spectral methods for the stochastic Stokes equations. *Computers & Fluids*, 71:250–260, 2013.

[105] A. Quarteroni. *Numerical Models for Differential Problems*, volume 16 of *MS&A*. Springer International Publishing, Cham, 2017.

[106] H. Rauhut. Random sampling of sparse trigonometric polynomials. *Appl. Comput. Harmon. Anal*, 22:16–42, 2007.

[107] H. Rauhut and R. Ward. Sparse Legendre expansions via ℓ_1-minimization. *J. Approx. Theory*, 164(5):517–533, 2012.

[108] H. Rauhut and R. Ward. Interpolation via weighted ℓ_1 minimization. *Appl. Comput. Harmon. Anal.*, 40(2):321–351, 2016.

[109] P. Seshadri, A. Narayan, and S. Mahadevan. Effectively subsampled quadratures for least squares polynomials approximations. *SIAM /ASA J. Uncertain. Quantif.*, 5:1003–1023, 2017.

[110] T. J. Sullivan. *Introduction to Uncertainty Quantification*. Springer International Publishing, Switzerland, 2015.

[111] G. Szegö. *Orthogonal Polynomials*. American Mathematical Society, Providence, RI, 1975.

[112] G Tang. *Methods for high dimensional uncertainty quantification: regularization, sensitivity analysis, and derivative enhancement*. PhD thesis, Stanford University, 2013.

[113] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. R. Statist. Soc. B*, 58(1):267–288, 1996.

[114] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, Philadelphia, PA, USA, 2013.

[115] J. A. Tropp. Algorithms for simultaneous sparse approximation. Part II: Convex relaxation. *Signal Process.*, 86:589–602, 2006.

[116] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. *Signal Process.*, 86:572–588, 2006.

[117] G. Tsogtgerel. Structure of the Laplacian on bounded domains, 2018. URL: `http://www.math.mcgill.ca/gantumur/math580f18/laplacenotes.pdf`. Last visited on 2019/10/31.

[118] E. van den Berg and M. P. Friedlander. SPGL1: A solver for large-scale sparse reconstruction. `http://www.cs.ubc.ca/~mpf/spgl1/`, June 2007.

[119] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput*, 31(2):890–912, 2008.

[120] E. van den Berg and M. P. Friedlander. Theoretical and empirical results for recovery from multiple measurements. *IEEE Trans. Inform. Theory*, 56(5):2516–2527, 2010.

[121] N. Vaswani and W. Lu. Modified-CS: Modifying compressive sensing for problems with partially known support. *IEEE Trans. Signal Process.*, 58(9):4595–4607, 2010.

[122] E. R. Vrscay. AMATH 731: Applied functional analysis, additional notes on fréchet derivatives, 2014. URL: `http://links.uwaterloo.ca/amath731docs/frechet.pdf`. Last visited on 2019/04/08.

[123] C. Wang and J. Peng. Exact recovery of sparse multiple measurement vectors by $\ell_{2,p}$-minimizaation. *J Inequal Appl*, pages 1–18, 2018.

[124] C. L. Wang. Banach calculus, 2012. URL: `http://www.math.ntu.edu.tw/~dragon/Lecture%20Notes/Banach%20Calculus%202012.pdf`. Last visited on 2019/11/27.

[125] E. Wegert. *Visual Complex Functions: An Introduction with Phase Portraits*. Birkhäuser, Basel, 2012.

[126] N. Wiener. The homogeneous chaos. *Am. J. Math*, 60(4):897–936, 1938.

[127] Z. Xu and T. Zhou. A gradient enhanced ℓ^1 recovery for sparse Fourier expansions. *Commun. Comput. Phys.*, 24(1):286 – 308, 2018.

[128] L. Yan, L. Guo, and D. Xiu. Stochastic collocation algorithms using ℓ_1-minimization. *Int. J. Uncertain. Quantif.*, 2(3):279–293, 2012.

[129] J. Yang and J. Zhang. Alternating direction algorithms for L1-problems in compressive sensing. *SIAM J. Sci. Comput.*, 33(1–2):250–278, 2011.

[130] C. Zheng, G. Li, Y. Liu, and X. Wang. Subspace weighted $\ell_{2,1}$ minimization for sparse signal recovery. *EURASIP J. Adv. Signal*, pages 1–11, 2012.

[131] J. Zou, Y. Fu, Q. Zhang, and H. Li. Split Bregman algorithms for multiple measurement vector problem. *Multidim Syst Sign Process*, 26:207–224, 2015.

Appendix A

Numerical experiments set-up for Chapter 4

All problems presented in §4.4 and §4.6 are solved numerically with the finite element method. As a first step, we discretize the physical domain with a triangular mesh. We simply define the finite approximation space $\mathcal{U}_h$ on the triangular mesh to be the space of continuous piecewise linear functions. It can be shown that $\mathcal{U}_h$ is a subset of the solution space $\mathcal{U}$. See [64, §4.1.1] for the proof. If the triangular mesh has in total N vertices, then the approximation space $\mathcal{U}_h$ has a dimension of N. We denote these vertices by $(\boldsymbol{x}_i)_{i=1}^N$. Let the set $\{\varphi_j : 1 \leq j \leq N\}$ be a basis of $\mathcal{U}_h$. Following what has been done in [25, §0.4], we assume this basis satisfying $\varphi_j(\boldsymbol{x}_i) = \delta_{ij} =$ the Kroneacker delta function. This set $\{\varphi_j\}$ is so-called the *nodal basis* of $\mathcal{U}_h$. By defining the basis like this, we will have

$$u_h(\boldsymbol{x}_i) = \sum_{j=1}^{N} u_j \varphi_j(\boldsymbol{x}_i) = u_j, \quad \text{for } i, j = 1, \ldots, N,$$

which are called the *nodal values* of the function u_h [57]. After the basis $\{\varphi_j\}$ has been defined, now we can explicit form the matrix equation $\boldsymbol{U}\boldsymbol{u} = \boldsymbol{b}$. Then, we solve this matrix equation and compute samples of the quantity of interest (QoI) with their gradient samples on FreeFem++ (version 3.61) [73]. Note that the given sampling point $\bar{\boldsymbol{y}} \in \mathcal{Y}$ is generated independent and identically distributed (i.i.d) with respect to the uniform measure when Legendre approximating polynomials are used and with respect to the Chebyshev measure when Chebyshev approximating polynomials are used.

After the QoI samples and the gradient samples are generated, the QoI is approximated by solving the gradient-augmented weighted ℓ^1 minimization problem

$$\min_{\boldsymbol{z} \in \mathbb{C}^N} \|\boldsymbol{z}\|_{1,\boldsymbol{w}} \text{ subject to } \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{q}\|_2 \leq \eta,$$

where sampling matrix $\boldsymbol{A}$ is defined the same way as in Chapter 3. The vector $\boldsymbol{q}$ contains samples of the QoI and the gradient samples of the QoI obtained through FreeFem++. Same as in Chapter 3, this weighted ℓ^1 minimization problem is solved using the SPGL1 package [118, 119] with a maximum number of 10,000 iterations and $\eta = 10^{-12}$ on Matlab

R2019a [91]. We choose the truncated index set Λ as the hyperbolic cross index set of degree s, where the degree s will be specified for each problem. The weights are defined as $w_{\boldsymbol{n}} = (u_{\boldsymbol{n}})^{\theta}$ for some $\theta \geq 0$ and $u_{\boldsymbol{n}}$ is defined the same way as in Chapter 3. Again, we model the total cost of computing the gradient-augment measurements by

$$\tilde{m} = m_o + m_g,$$

where m_o is the number of function samples and m_g is the number of the gradient samples. For the unaugmented problem, the computational cost is just $\tilde{m} = m_o$. For all problems shown in §4.4 and §4.6, we generate the gradient samples of the QoI at the same points as the QoI samples. The approximation error $\|q - \tilde{q}\|_{L^\infty}$, where q is the QoI value obtained with finite element method and $\tilde{q}$ denotes the approximated value of QoI, is computed on a grid of $4\,|\Lambda|$ uniformly distributed points and averaged over 10 trials.