

ABSTRACT

HERMAN, ELIZABETH ASHLEY. Design of Inverse Problems and Surrogate Modeling in Complex Physical Systems. (Under the direction of Alen Alexanderian and Arvind Krishna Saibaba.)

Continuous advances in mathematical modeling have allowed mathematicians and computational scientists to successfully describe physical phenomena. Specifically, within the areas of science and engineering, there are many applications that require computationally complex models, in the form of partial differential equations, to model the physical phenomena; these complex mathematical models can be expensive to simulate. In addition, tasks, such as design, control, and optimization, require repeated evaluations of the model, making the computational cost of analyzing the models prohibitively expensive. In this thesis, we focus on two aspects: optimal experimental design (OED) for inverse problems and developing surrogate models for an application in materials science.

Majority of the work presented in this thesis focuses on optimal placement of sensors for data acquisition in Bayesian inverse problems. This problem can be formulated in the optimal experimental design framework, where we specifically focus on the A-optimal criterion. We exploit the low-rank structure of the inverse problem to derive efficient randomized estimators for the A-optimal objective function and its gradient, based on the randomized subspace iteration. Motivated by computational costs involving the A-optimal criterion, we also propose a new criterion called the modified A-optimal criterion and derive randomized estimators to efficiently compute this new criterion. For both of these criteria, along with their gradients, we derive error bounds to monitor the accuracy of the proposed estimators. The placement of sensors is determined by the solution of a prohibitively expensive binary optimization problem. Instead, we seek a sparse and binary solution to a continuous optimization problem, where the objective function is the summation of the design criterion and a penalty function. We propose a new penalty function and derive a new reweighted ℓ_1 optimization algorithm, based on the Majorization-Minimization principle. Finally, motivated by applications in which implementations of the adjoint PDE solvers are not readily available, we develop randomized estimators to evaluate the A- and D-optimal criterion and their gradients in this setting. This approach relies on an additional assumption that the prior covariance operator is approximately low-rank. The success of the proposed methods are illustrated on a synthetic model problem in contaminant source identification.

In the last part of the thesis, we consider a phase-field PDE model for physical vapor deposition (PVD). PVD is an experimental procedure for creating binary alloy thin films, where the structure of the thin film is dictated by the process parameters of the experiment. In the thesis, we are interested in understanding the relationship between a subset of the process parameters and a function of the resulting PVD output. Since model evaluations of the PDE model are computationally expensive, we construct a surrogate model using non-intrusive

techniques to efficiently explore the input parameter space. We take advantage of the low-rank structure of the output sample covariance matrix and use principal component analysis (PCA) to compute a low-dimensional approximation of the high-fidelity output. The process parameters are then linked to the low-dimensional approximation by polynomial chaos expansions (PCEs). Through an error study, the surrogate model is demonstrated to be accurate.

© Copyright 2020 by Elizabeth Ashley Herman

All Rights Reserved

Design of Inverse Problems and Surrogate Modeling in Complex Physical Systems

by
Elizabeth Ashley Herman

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2020

APPROVED BY:

Carl Kelley

Murthy Guddati

Alen Alexanderian
Co-chair of Advisory Committee

Arvind Krishna Saibaba
Co-chair of Advisory Committee

DEDICATION

To my family and friends, your love and support has made this journey possible.

BIOGRAPHY

Elizabeth Herman grew up in Black Creek, Wisconsin and later pursued a Bachelor of Science degree in Mathematics at Marian University in Fond du Lac, Wisconsin in 2015. Following the completion of her Bachelor's degree in 2015, she moved to Raleigh to attend North Carolina State University. She received her Masters in 2017 and PhD in 2020. While in the graduate program, she participated in an internship at Sandia National Laboratories.

ACKNOWLEDGEMENTS

First, I would like to thank my advisors, Alen Alexanderian and Arvind Saibaba, for their guidance throughout this research experience; I appreciate the time and effort that was invested in my success. In a similar light, I would like to thank, Rémi Dingreville and James Stewart, for the discussions and support that was provided during (and after) my internship.

I would also like to acknowledge my sources of financial support. The material on optimal experimental design, Chapters 3 - 5, was supported in part by the National Science Foundation (NSF) award DMS-1745654. The work in Chapter 6 was performed, in part, at the Center for Integrated Nanotechnologies, an Office of Science User Facility operated for the U.S. Department of Energy. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy National Nuclear Security Administration under contract DE-NA0003525. The views expressed in Chapter 6 do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Finally, I would like to thank all of my family and friends who have provided me with love, support, and inspiration along this journey.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
Chapter 1 Introduction	1
1.1 Problem statements	2
1.2 Outline of the thesis	5
Chapter 2 Background	7
2.1 Gaussian random fields	7
2.2 Bayesian linear inverse problems	8
2.3 Optimal design of experiments	11
2.4 Design criteria	12
2.4.1 A-optimal criterion	12
2.4.2 D-optimal criterion	18
2.5 Randomized subspace iteration algorithm	18
Chapter 3 Efficient computation of A-optimal criterion and its gradient	20
3.1 Randomized estimators for $\Phi_{\text{aopt}}(\mathbf{w})$ and its gradient	21
3.1.1 Computational procedure	21
3.1.2 Error Analysis	24
3.2 Modified A-optimal criterion	25
3.2.1 Computational procedure	26
3.2.2 Error analysis	26
3.3 Numerical results	27
3.3.1 Model problem and solvers	27
3.3.2 Accuracy of estimators	28
3.4 Proofs of bounds	30
3.4.1 Trace of matrix function	30
3.4.2 Proof of Theorem 2	34
3.4.3 Proof of Theorem 3	35
3.5 Conclusion	35
Chapter 4 An optimization framework for finding binary designs	36
4.1 Penalty functions	37
4.2 MM approach and reweighted ℓ_1 algorithm	38
4.3 Numerical results	39
4.3.1 Performance of the reweighted ℓ_1 algorithm	40
4.3.2 Computing optimal designs	41
4.3.3 Comparing A-optimal and modified A-optimal designs	43
4.4 Conclusion	44
Chapter 5 Efficient computation of the OED problem without adjoint PDE applications	45
5.1 Reformulation of the posterior covariance	46
5.2 Design criteria and their gradients	48

5.3	Computational procedure	50
5.3.1	Low-rank approximation of $\mathbf{\Gamma}_{\text{prior}}$	50
5.4	Numerical results	54
5.4.1	A-optimal designs	54
5.4.2	D-optimal designs	56
5.5	Conclusion	56
Chapter 6 A non-intrusive surrogate model for predicting microstructure mor-		
phologies during physical vapor deposition		58
6.1	High-fidelity model for PVD	59
6.1.1	Phase-field PVD model and numerical solution	60
6.1.2	Statistical representation of the microstructure	63
6.1.3	Computation of the high-fidelity model	64
6.2	Construction of a surrogate model for PVD	67
6.2.1	PCA: Compute a k -dimensional approximation	68
6.2.2	PCE: Linking PCA coefficients with process parameters	70
6.3	Numerical results	72
6.3.1	High-fidelity samples	72
6.3.2	Accuracy of PCA	72
6.3.3	Linking process parameters to PCA expansion	74
6.4	Conclusion	76
Chapter 7 Conclusion		77
BIBLIOGRAPHY		79

LIST OF TABLES

Table 3.1	Computational cost measured in terms of PDE solves for different methods in computing $\Phi_{\text{aopt}}(\mathbf{w})$ and $\nabla\Phi_{\text{aopt}}(\mathbf{w})$. Typically, $q = 1$ in Algorithm 1 is sufficient.	24
Table 4.1	Number of subproblem solves (4.5), function evaluations, and active sensors for varying ℓ with the reweighted ℓ_1 algorithm and $n_s = 109$	40
Table 5.1	Numerical results obtained from computing the A-optimal designs in Figure 5.2.	56
Table 5.2	Numerical results obtained from computing the D-optimal designs in Figure 5.3.	57
Table 6.1	The error associated with using 8 basis functions to approximate the high-fidelity samples of the training and testing set. By training (testing) error, we mean $\text{err}_1(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is in the training (testing) set.	74
Table 6.2	Size of polynomial basis $\ell_i + 1$ for each coefficient $\alpha_i(\boldsymbol{\theta})$ determined by performing 10-fold cross validation on the training set and choosing the truncation level which produced the smallest average error in the validation set.	74
Table 6.3	The error in approximating $\boldsymbol{\alpha}(\boldsymbol{\theta})$ using PCE, $\text{err}_2(\boldsymbol{\theta})$, and the error incurred by using the surrogate model to predict $\mathbf{g}(\boldsymbol{\theta})$, $\text{err}_3(\boldsymbol{\theta})$. The training (testing) error is computed by evaluating $\text{err}_2(\boldsymbol{\theta})$ and $\text{err}_3(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is in the training (testing) set.	76

LIST OF FIGURES

Figure 2.1	Different spaces, their inner products, and the adjoints of linear transformations between them. Here $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and $\langle \cdot, \cdot \rangle_{\mathbf{M}}$ is the mass-weighted inner product.	11
Figure 3.1	Domain with 109 candidate sensor locations.	28
Figure 3.2	Relative error of the randomized estimators for the A-optimal criterion (left), and those corresponding to the modified A-optimal criterion (right) for varying ℓ	29
Figure 3.3	Absolute error bound for $\hat{\Phi}_{\text{aopt}}(\mathbf{w})$ from Theorem 2 (left), and $\hat{\Phi}_{\text{mod}}(\mathbf{w})$ from Theorem 3 (right) for varying ℓ	29
Figure 4.1	Different choices of penalty functions with $n_s = 1$	37
Figure 4.2	Optimal designs as a result of varying γ with the reweighted ℓ_1 algorithm for the A-optimal criterion. We set $\ell = 207$	41
Figure 4.3	The effect of varying γ on the reweighted ℓ_1 algorithm for the A-optimal criterion. We set $\ell = 207$	42
Figure 4.4	Standard deviation computed using the optimal design indicated by the gray circles (left). True initial condition (middle) and initial condition reconstruction (right). The optimal design was computed using $\ell = 207$, the reweighted ℓ_1 algorithm, and $\gamma = 5$	42
Figure 4.5	Comparison of the prior standard deviation field (left) with the posterior standard deviation field (right) computed using the optimal design indicated by the gray circles.	43
Figure 4.6	The true A-optimal criterion computed using the optimal and randomly generated designs. The optimal designs were computed using the A-optimal criterion and the reweighted ℓ_1 algorithm for different values of γ (left). Comparing the computed optimal design when $\gamma = 3$ to 1500 randomly generated designs (right).	43
Figure 4.7	Comparison of designs obtained by minimizing the (approximate) A-optimal and modified A-optimal criteria. For each design, we report the exact trace of the corresponding posterior covariance operator.	44
Figure 5.1	The eigenvalue decay (left) and the cumulative eigenvalue fraction (right) of $\mathbf{\Gamma}_{\text{prior}}$	55
Figure 5.2	Designs computed using $r = 98$ (left), $r = 200$ (middle left), $r = 400$ (middle right), and $r = 1000$ (left).	55
Figure 5.3	Designs computed using $r = 98$ (left), $r = 200$ (middle left), $r = 400$ (middle right), $r = 1000$ (right) with D-optimal criterion.	56
Figure 6.1	Domain used for phase-field model. The horizontal direction is defined as x_1 and the vertical direction is defined as x_2	65
Figure 6.2	Example of microstructures obtained from the PVD process obtained with varying velocity and phase fraction.	66
Figure 6.3	Autocorrelation $\mathbf{R}_{c_\alpha c_\alpha}(\boldsymbol{\theta})$ (left) of a microstructure (right), where phase α is defined as the yellow state and β as the blue state.	67
Figure 6.4	Outline of approach to create a surrogate model for PVD, $\hat{\mathbf{g}}(\boldsymbol{\theta})$	68

Figure 6.5	Performing PCA on the high-fidelity training samples. The computed variance of $\mathbf{g}(\boldsymbol{\theta})$ (left) and the cumulative proportion of total variance (percentage) for the first 50 principal directions (right).	73
Figure 6.6	The resulting average, $\hat{\boldsymbol{\mu}}$, (top, left) and the first 5 basis elements, $\mathbf{u}_1, \dots, \mathbf{u}_5$, computed from PCA on the high-fidelity training set.	73
Figure 6.7	Example prediction using the surrogate model: high-fidelity evaluation $\mathbf{g}(\boldsymbol{\theta})$ (left) and surrogate evaluation $\hat{\mathbf{g}}(\boldsymbol{\theta})$ (right). The relative point-wise error, $\boldsymbol{\delta}(\boldsymbol{\theta})$, as calculated by (6.19) is illustrated in (middle).	75

CHAPTER

1

INTRODUCTION

Mathematical models described by partial differential equations (PDEs) are important in many applications in science and engineering. Examples include epidemiology (e.g., modeling the spread of disease), meteorology (e.g., predicting hurricane paths), geophysics (e.g., modeling the flow of a contaminant in groundwater), and materials science (e.g., radiation damage in metals). In general, models can vary in computational complexity, from simple models that can be run in a few seconds to elaborate ones that can require days to simulate. Furthermore, the computational cost becomes prohibitively high if the model is the governing equation of design, control, or optimization problems, as these tasks require many model evaluations.

In this thesis, we consider two computationally intensive problems: design of inverse problems and forward uncertainty propagation. In each problem, the model in question is described by a system of PDEs, which is assumed to be computationally expensive to evaluate. To enable efficient solution methods, a common theme throughout the thesis is understanding and exploiting problem structure (e.g., low-rank structures in the problem).

The majority of the work presented in the thesis focuses on the design of inverse problems. In particular, we seek to develop efficient methods that address challenges associated with computing optimal experimental designs (OEDs) for infinite-dimensional Bayesian linear inverse problems governed by PDEs. We demonstrate the efficiency of the developed methods on a contaminant source identification problem. The second problem we consider involves a non-intrusive approach for constructing a surrogate model that approximates a computationally expensive high-fidelity model. The specific high-fidelity model we consider is a function of a PDE model output, where the PDE model describes a materials science experimental process called physical vapor deposition (PVD). The surrogate model construction is the first step toward

enabling efficient forward uncertainty quantification.

In Section 1.1, we introduce the problems of interest in the thesis. Then, in Section 1.2, we describe the premise behind each chapter and highlight the contributions of this thesis towards addressing the challenges associated with each of the respective problems.

1.1 Problem statements

Here we motivate the two problems we consider in the thesis: design of experiments for large-scale linear inverse problems governed by PDEs and surrogate model construction for PVD. We discuss the challenges associated with each problem and provide references to works that have made significant advancements to each research area.

OED for large-scale linear inverse problems governed by PDEs

A central problem in scientific computing involves estimating parameters that describe mathematical models, such as initial conditions, boundary conditions, or material parameters. This is often addressed by using experimental measurements and a mathematical model to compute estimates of unknown model parameters. In other words, one can estimate the parameters by solving an inverse problem. Experimental design involves specifying the experimental setup for collecting measurement data, with the goal of accurately recovering the parameters of interest. As such, optimal experimental design (OED) is an important aspect of effective and efficient parameter estimation. Namely, in applications where collecting experimental measurements is expensive (e.g., because of budget, labor, or physical constraints), deploying experimental resources has to be done efficiently and in a parsimonious manner. Even when collecting large amounts of data is feasible, OED is still important; the computational cost of processing all the data may be prohibitive or a poorly designed experiment with many measurements may miss important information about the parameters of interest.

To make matters concrete, we explain the inverse problem and the experimental design in the context of an application. Consider the transport of a contaminant in an urban environment or the subsurface. The forward problem involves forecasting the spread of the contaminant, whereas the inverse problem involves using the measurements of the contaminant concentration at discrete points in space and time to recover the source of the contaminant (i.e., the initial state). In this application, OED involves optimal placement of sensors, at which measurement data is collected, to reconstruct the initial state.

We focus on OED for Bayesian linear inverse problems governed by PDEs. In our formulation of the OED problem, the goal is to find an optimal subset of sensors from a fixed array of n_s candidate sensor sites. The experimental design is parameterized by assigning non-negative weights to each candidate sensor location. Ideally, we seek a binary weight vector \mathbf{w} ; if $w_i = 1$, a sensor will be placed at the i th candidate location, and if $w_i = 0$, no sensor will be placed at the location. However, formulating an optimization problem over binary weight vectors leads to

a binary optimization problem that is prohibitively expensive from a computational standpoint. A common approach to address this issue is to relax the binary requirement on design weights by letting the weights take values in the interval $[0, 1]$. The sparsity of the design will then be controlled using a penalty method; see e.g., [5, 42, 111]. This results in an optimization problem of the following form:

$$\min_{\mathbf{w} \in [0,1]^{n_s}} \Phi(\mathbf{w}) + \gamma P(\mathbf{w}), \quad (1.1)$$

where Φ denotes the design criterion, $\gamma > 0$ is a penalty parameter, and P is a penalty function.

Adopting a Bayesian approach to the inverse problem, the design criterion will be a measure of the uncertainty in the estimated parameters. Throughout this problem, we focus on a popular choice known as the A-optimal criterion [20, 104]. That is, we seek a sensor configuration that results in a minimized average posterior variance. The design criterion, in this case, is given by the trace of the posterior covariance operator.

One major challenge in solving (1.1), specifically for PDE-based inverse problems, is the computational cost of objective function and gradient evaluations. Namely, the posterior covariance operator $\mathbf{\Gamma}_{\text{post}}$ is dense, high-dimensional, and computationally challenging to explicitly form—computing applications of $\mathbf{\Gamma}_{\text{post}}$ to vectors requires solving multiple PDEs. Furthermore, these computations must be performed at each iteration of an optimization algorithm used to solve (1.1). To address this computational challenge, efficient and accurate matrix-free approaches for computing the OED objective and its gradient are needed. Another challenge in solving the OED problem (1.1) is the need for a suitable penalty method that is computationally tractable and results in sparse and binary optimal weight vectors. Majority of the thesis is related to this problem set up and specifically to developing methods that overcome these computational challenges.

The research area of OED is widely studied [3, 5, 6, 8, 9, 11, 20, 24, 26, 40–42, 46, 48, 58–60, 67, 84, 86, 88, 92, 104, 105, 111]. The efforts that focus on developing algorithms for A-optimal designs for ill-posed linear inverse problems include [40, 42], and more specifically [5, 59] focus on infinite-dimensional Bayesian linear inverse problems. In these articles, Monte-Carlo trace estimators are used to approximate the A-optimal design criterion and its gradient. Also, [5, 42] advocate use of low-rank approximations using the Lanczos algorithm or the randomized SVD [43]. We refer to the work, [89], for comparison of Monte Carlo trace estimators and those based on randomized subspace iteration; it was shown that the latter are significantly more accurate than Monte Carlo trace estimators. Regarding sparsity control, various techniques have been used to approximate the ℓ_0 -“norm” to enforce sparse and binary designs. For example, [40–42] use the ℓ_1 -penalty function with an appropriate threshold to post-process the solution. In [5], a continuation approach is proposed that involves solving a sequence of optimization problems with non-convex penalty functions that approximate the ℓ_0 -“norm”. More recently, in [111], a sum-up rounding approach is proposed to obtain binary optimal designs.

Surrogate model construction for PVD

Nanostructured thin films are used as coatings to serve specific mechanical, thermal, magnetic or optical functionalities [21, 27, 69]. The properties of these thin films depend directly on the underlying structure resulting from the synthesis process called physical vapor deposition. PVD consists of solid (metal) materials being converted to a vapor and then deposited to a surface, where the vaporized materials are returned to solid form. This process is complex to model due to a competition between the depositing vapor flux and surface transport effects, which govern the formation of specific underlying structures of the thin film [95, 101, 102]. The complexity and high-dimensionality of the processing input space hinders the ability to design synthesis protocols for fabrication of reliable and functional thin films. In our work, we focus on constructing a surrogate model which approximates the relationship between a subset of process parameters and the output thin film structure of PVD. Building a surrogate model that rapidly links process to structure is central to being able to fabricate thin films that have specific structure. The consequence is possible advancements to applications in aerospace, automotive industry, manufacturing, textiles, and electronics [7, 90].

Numerous high-fidelity simulation methods using atomistic, kinetic Monte Carlo or phase-field methods [23, 31, 68, 95] predict the thin film evolution during PVD quite well. We utilize a phase-field model for simulating PVD of binary-alloy thin films developed in [95] (see brief description of the model in Chapter 6) to simulate the PVD process. While this model accurately predicts experimentally observed thin films across a broad class of immiscible materials systems and deposition conditions, it is computationally expensive. This phase-field model is not suited to rapidly explore the large number of combinations of processing parameters needed to design thin films with specific properties. A large portion of the literature has been centered around the development of surrogate models that efficiently and quickly link process, structure, and properties for a wide variety of materials and fabrication processes [19, 52–54, 62, 74, 83, 93, 96, 107, 109]. Such linkage is constructed using a range of different techniques including, regression models [54, 62, 83, 96], Gaussian processes [52] or deep learning [19, 110]. All of these methods rely on a statistical representation of the thin film to construct a surrogate model that can be used to identify process-structure relations, thus, eliminating the source of randomness in the output of the high-fidelity model. For example, [109] presented a workflow to extract process-structure linkages from phase-field simulations of the structure evolution of ternary eutectics during solidification. This methodology uses two-point statistics, principal component analysis (PCA), and a spectral representation to link the temperature-gradient to the output of their model. As identified by the authors in [109], this methodology would need to be demonstrated for a wide range of solidification rate and eutectic concentration. In our case, this would correspond to deposition rate and phase fraction, the two process parameters we consider in our work. Predicting the underlying thin film structures achievable during PVD as a function of the deposition parameters presents an additional challenge related to the fact that the deposition conditions can yield a much broader range of structures [95].

1.2 Outline of the thesis

The details of this section are reserved for providing an overview of the problems discussed in each chapter of the thesis and the contributions of the author. We mention that the majority of the work presented in this thesis has been submitted for publication. Specifically, the contents of Chapter 3 and Chapter 4 are based on an article, [44], that has been accepted for publication in the SIAM Journal of Scientific Computing. The contents of Chapter 6 are based on a manuscript that has been recently submitted to a peer-reviewed journal [45].

Chapter 2: Background. This chapter includes background details to the problems we consider and sets up notation that will be carried throughout the thesis.

Chapter 3: Efficient computation of A-optimal criterion and its gradient. The A-optimal criterion is the trace of the posterior covariance operator; however, evaluating this criterion is computationally challenging since constructing the posterior covariance is infeasible. Because the objective criterion and the gradient need to be evaluated many times, we require efficient methods for their evaluation. In particular, we need algorithms that are matrix-free, in the sense that, the operators are not formed explicitly, but only through their action on vectors.

Contributions: In this chapter, we develop efficient and accurate randomized estimators of the A-optimal criterion and its gradient, based on randomized subspace iteration. This is accompanied by a detailed algorithm that guides efficient implementations, discussion of computational cost, as well as theoretical error analysis. The estimators are structure exploiting, in that they use the low-rank structure embedded in the posterior covariance operator. To quantify the accuracy of the estimators, a rigorous error analysis is presented, significantly advancing the methods in [89]. A feature of the analysis is that the bounds are independent of the dimension of the discretized inversion parameter. Furthermore, the computational cost (measured in the number of PDE solves) of the A-optimal objective and gradient using the proposed estimators is independent of the discretized parameter dimension. In this chapter, a new criterion is also proposed. Motivated by reducing computational cost, the modified A-optimal criterion is derived by considering a suitably weighted A-optimal criterion. We present randomized estimators for this novel criterion and its gradient and derive error bounds for its accuracy. We demonstrate the methods on a synthetic source identification problem.

Chapter 4: An optimization framework for finding binary designs. We formulate the optimal sensor placement problem as in (1.1), where a penalty term is included to promote sparsity in the solution. The penalty term must enforce a sparse and binary design. However, many convex penalty functions that ensure sparsity of the design, require additional techniques (i.e., thresholding) to obtain a binary design.

Contributions: We propose a new penalty method that enforces sparse and binary designs. Then, we develop a reweighted ℓ_1 algorithm for solving this optimization problem based on the Majorization-Minimization principle. The resulting algorithm requires solving a sequence of convex subproblems and is accelerated by the randomized estimators developed in Chapter 3.

We illustrate the success of the penalty method on the same source identification problem as in Chapter 3.

Chapter 5: Efficient computation of the OED problem without adjoint PDE applications. In this chapter, we consider a slight variation of the problem setup in the previous two chapters. We assume that an implementation of the adjoint PDE solver may not be readily available. For example, the adjoint solver may be complicated to derive and implement due to the coupled multiphysics, the forward model may be implemented with legacy software, or the adjoint solver may be computationally expensive.

Contributions: Under this context, we present adjoint-free estimators for the A- and D-optimal criterion and their respective gradients. The estimators are developed using an alternative expression for the posterior covariance operator and take advantage of the symmetry and low-rankness of the operator. This approach requires an additional assumption; i.e., we assume that the prior covariance operator is approximately low-rank. For OED problems where the added assumption is reasonable, the computational cost is significantly decreased using our methods. The algorithms are demonstrated on a model problem from contaminant source identification.

Chapter 6: A non-intrusive surrogate model for predicting microstructure morphologies during physical vapor deposition. Thin film creation via PVD is a complex physical phenomenon; however, understanding how different process parameters affect the thin film structure is important to be able to design thin films with favorable properties. The existing high-fidelity models that describe PVD are computationally expensive and make it challenging for efficient exploration of the input space. Within this chapter, we consider a high-fidelity model: the input is a finite set of physical parameters and the output is a function of a phase-field model solution.

Contributions: In Chapter 6, we develop a non-intrusive surrogate modeling approach to approximate the high-fidelity model. To construct the surrogate model, we first evaluate the high-fidelity at a set of parameters and utilize PCA to find a reduced-dimensional subspace of the output space, exploiting the structure of the problem [50]. Then the relationship between the process parameters and the PCA coefficients is determined using polynomial chaos expansions (PCEs) [25]. The error of the surrogate model is explored at various stages of the construction process and is shown to accurately approximate the high-fidelity model.

Chapter 7: Conclusion. This chapter summarizes the main contributions and conclusions from the work presented in the thesis. We also discuss some potential future directions.

CHAPTER

2

BACKGROUND

In this section, we recall the background material needed in the remainder of the thesis. In Section 2.1, we provide definitions of a random field and specifically discuss Gaussian random fields. Section 2.2 introduces the basics of Bayesian inverse problems. Then, in Section 2.3, we define the optimal experimental design problem, which is the focus of most of this thesis. Section 2.4 provides a discussion of the design criteria we consider in our work. Finally, Section 2.5 states the randomized algorithms we utilize throughout the thesis.

2.1 Gaussian random fields

In this section, we recall some basics regarding Gaussian random fields. The details of this section are based on the following references [3, 5, 16, 28, 85, 97].

Consider the probability triple $(\Omega, \mathcal{B}(\Omega), P)$, where Ω represents a non-empty sample space, $\mathcal{B}(\Omega)$ is the Borel sigma-algebra on Ω , and the final term is a probability measure $P : \mathcal{B} \rightarrow [0, 1]$. We begin our discussion with defining a random field.

Definition 1. *Let $(\Omega, \mathcal{B}(\Omega), P)$ and $\mathcal{D} \subseteq \mathbb{R}^n$. A random field X is defined as a function*

$$X : \Omega \times \mathcal{D} \rightarrow \mathbb{R}.$$

Intuitively, a random field can be thought of in the following ways:

1. For a fixed point, $\mathbf{d} \in \mathcal{D}$, $X_{\mathbf{d}} : \Omega \rightarrow \mathbb{R}$, is a random variable. Therefore, a random field assigns a probability distribution to each spatial location.

2. Fixing $\omega \in \Omega$, X_ω is a function from \mathcal{D} to \mathbb{R} . Or alternatively, $X_\omega \in \mathcal{V}$, where \mathcal{V} is a function space (for our work, we set $\mathcal{V} = \mathcal{L}^2(\mathcal{D})$). This function is commonly called a sample function [66].

In our work, we take the approach that X is a random variable, which maps $(\Omega, \mathcal{B}(\Omega), P)$ to $(\mathcal{V}, \mathcal{B}(\mathcal{V}))$. Next, we can define the probability law of X .

Defintion 2. *The probability law μ of a random variable, $X : (\Omega, \mathcal{B}(\Omega), P) \rightarrow (\mathcal{V}, \mathcal{B}(\mathcal{V}))$, is defined as $\mu(E) = P(X \in E)$, where $E \in \mathcal{B}(\mathcal{V})$.*

A particular class of random fields that we consider in Chapter 3 – Chapter 5 are Gaussian random fields. Here we define them along with important features that are needed within the thesis.

Defintion 3. *A random field, X , is Gaussian if for any finite set of spatial locations, $\mathbf{d}_1, \dots, \mathbf{d}_n \in \mathcal{D}$, $\mathbf{X} = [X(\mathbf{d}_1), \dots, X(\mathbf{d}_n)]$ is a n -dimensional Gaussian random vector.*

A Gaussian random field is completely defined by its mean and covariance function [28, 97].

Defintion 4. *The mean of a random field X is defined as*

$$\bar{m}(\mathbf{d}) = \mathbb{E}[X(\mathbf{d})] \quad \text{for } \mathbf{d} \in \mathcal{D}.$$

Defintion 5. *The covariance function of a random field X is defined as*

$$c(\mathbf{d}_i, \mathbf{d}_j) = \mathbb{E}[(X(\mathbf{d}_i) - \bar{m}(\mathbf{d}_i))(X(\mathbf{d}_j) - \bar{m}(\mathbf{d}_j))] \quad \text{for } \mathbf{d}_i, \mathbf{d}_j \in \mathcal{D},$$

where $\bar{m}(\mathbf{d})$ is the mean of the random field.

The covariance operator, Γ , for a random field, is defined in terms of the covariance function. Specifically, the covariance operator can be represented as an integral operator, such that for $u \in \mathcal{V}$,

$$[\Gamma u](\mathbf{x}) = \int_{\mathcal{D}} c(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) d\mathbf{y}, \quad (2.1)$$

where $c(\mathbf{x}, \mathbf{y})$ was defined in Defintion 5.

2.2 Bayesian linear inverse problems

Inverse problems are used to estimate a parameter using measurement data and a model. In general, the relationship between the unknown parameter, m , and the measurement data, \mathbf{y} , is described by the model

$$f(m) + \boldsymbol{\eta} = \mathbf{y},$$

where $\boldsymbol{\eta}$ represents the measurement noise and f is a (possibly nonlinear) parameter-to-observable map [99]. The focus of the thesis is on the case where the parameter-to-observable map is linear,

i.e., $f(m) = Fm$, where F is a continuous linear transformation. Then, the corresponding model is of the form,

$$Fm + \boldsymbol{\eta} = \mathbf{y}. \quad (2.2)$$

Additionally, we assume that the inversion parameter m is an element of $\mathcal{V} = L^2(\mathcal{D})$, where \mathcal{D} is a bounded domain and that the measurement noise is Gaussian with $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_{\text{noise}})$.

Standard techniques of computing a solution to the inverse problem, described by (2.2), fall into two categories, deterministic or Bayesian. The deterministic setting yields a single estimate of the unknown parameter, obtained from solving the optimization problem

$$\min_m \|Fm - \mathbf{y}\|_{\boldsymbol{\Gamma}_{\text{noise}}^{-1}}^2 + R(m),$$

where $\|\cdot\|_{\boldsymbol{\Gamma}_{\text{noise}}^{-1}}$ is a weighted norm and $R(m)$ is a regularization term [5, 97]. On the other hand, the solution to the inverse problem in a Bayesian setting is accompanied by knowledge about the uncertainties in the solution, in the form of the posterior probability law of m . In this section, we discuss the Bayesian approach to inverse problems.

The setup of the inverse problem

To fully specify the inverse problem, we need to describe the prior law of m and our choice of data likelihood. For the prior, we choose a Gaussian measure $\mu_{\text{pr}} = \mathcal{N}(m_{\text{pr}}, \boldsymbol{\Gamma}_{\text{pr}})$. We assume the prior mean m_{pr} is a sufficiently regular element of \mathcal{V} and that the covariance operator $\boldsymbol{\Gamma}_{\text{pr}} : \mathcal{V} \rightarrow \mathcal{V}$ is a strictly positive self-adjoint trace-class operator. Following the developments in [16, 29, 97], we use $\boldsymbol{\Gamma}_{\text{pr}} = \mathcal{A}^{-2}$ with \mathcal{A} taken to be a Laplacian-like operator [97]. This ensures that $\boldsymbol{\Gamma}_{\text{pr}}$ is trace-class in two and three space dimensions and allows for use of efficient elliptic PDE solvers to compute matrix-vector products, i.e., the action of $\boldsymbol{\Gamma}_{\text{pr}}^{1/2}$ on $u \in L^2(\mathcal{D})$ is computed by solving the elliptic PDE, $\mathcal{A}v = u$.

We consider the case where F represents a time-dependent PDE and we assume observations are taken at n_s sensor locations at n_t points in time. Thus, the vector of experimental data \mathbf{y} is an element of $\mathbb{R}^{n_s n_t}$. An application of the parameter-to-observable map, $F : \mathcal{V} \rightarrow \mathbb{R}^{n_s n_t}$, involves a PDE solve followed by an application of a spatiotemporal observation operator that extracts solution values at the observation times for each of the sensor locations. Given our choice of the noise model—additive and Gaussian—the likelihood probability density function (pdf) is

$$\pi_{\text{like}}(\mathbf{y} \mid m) \propto \exp \left\{ -\frac{1}{2} (Fm - \mathbf{y})^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1} (Fm - \mathbf{y}) \right\}.$$

For a prior probability law μ_{pr} and noise model, the solution to the infinite dimensional Bayesian inverse problem is the posterior distribution law $\mu_{\text{post}}^{\mathbf{y}}$ that satisfies the infinite dimensional Bayes formula [97, 106]:

$$\frac{d\mu_{\text{post}}^{\mathbf{y}}}{d\mu_{\text{pr}}} \propto \pi_{\text{like}}(\mathbf{y} \mid m),$$

where $\frac{d\mu_{\text{post}}^{\mathbf{y}}}{d\mu_{\text{pr}}}$ is the Radon–Nikodým derivative of the posterior measure with respect to the prior measure and $\pi_{\text{like}}(\mathbf{y}|m)$ is the data likelihood. Since we have assumed a linear forward mapping, an additive and Gaussian noise model, and a Gaussian measure as the prior law, then the solution of the Bayesian inverse problem is the Gaussian measure, $\mu_{\text{post}}^{\mathbf{y}} = \mathcal{N}(m_{\text{post}}^{\mathbf{y}}, \Gamma_{\text{post}})$ with

$$m_{\text{post}}^{\mathbf{y}} = \Gamma_{\text{post}}(F^* \Gamma_{\text{noise}}^{-1} \mathbf{y} + \Gamma_{\text{pr}}^{-1} m_{\text{pr}}), \quad \Gamma_{\text{post}} = (F^* \Gamma_{\text{noise}}^{-1} F + \Gamma_{\text{pr}}^{-1})^{-1}. \quad (2.3)$$

Note that here the posterior mean $m_{\text{post}}^{\mathbf{y}}$ coincides with the maximum a posteriori probability (MAP) estimator [97].

Discretization

We use a continuous Galerkin finite element discretization approach for the governing PDEs, as well as the inverse problem. Specifically, our discretization of the Bayesian inverse problem follows the developments in [16]. The discretized parameter space in the present case is $\mathcal{V}_n = \mathbb{R}^n$ equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathbf{M}}$ and norm $\|\cdot\|_{\mathbf{M}} = \langle \cdot, \cdot \rangle_{\mathbf{M}}^{1/2}$, where \mathbf{M} is the finite element mass matrix. Note that $\langle \cdot, \cdot \rangle_{\mathbf{M}}$ is the discretized $L^2(\mathcal{D})$ inner product. The discretized parameter-to-observable map is a linear transformation $\mathbf{F} : \mathcal{V}_n \rightarrow \mathbb{R}^{n_s n_t}$ with adjoint \mathbf{F}^* discussed below. The discretized prior measure $\mathcal{N}(\mathbf{m}_{\text{pr}}, \Gamma_{\text{prior}})$ is obtained by discretizing the prior mean and covariance operator, and the discretized posterior measure is given by $\mathcal{N}(\mathbf{m}_{\text{post}}, \Gamma_{\text{post}})$, with

$$\Gamma_{\text{post}} = \left(\mathbf{F}^* \Gamma_{\text{noise}}^{-1} \mathbf{F} + \Gamma_{\text{prior}}^{-1} \right)^{-1}, \quad \mathbf{m}_{\text{post}}^{\mathbf{y}} = \Gamma_{\text{post}} \left(\mathbf{F}^* \Gamma_{\text{noise}}^{-1} \mathbf{y} + \Gamma_{\text{prior}}^{-1} \mathbf{m}_{\text{pr}} \right). \quad (2.4)$$

We point out that the operator $\mathbf{F}^* \Gamma_{\text{noise}}^{-1} \mathbf{F}$ is the Hessian of the data-misfit cost functional whose minimizer is the MAP point, and is thus referred to as the data-misfit Hessian; see e.g., [5].

It is important to keep track of the inner products in the domains and ranges of the linear mappings, appearing in the above expressions, when computing the respective adjoint operators. For the readers' convenience, in Figure 2.1, we summarize the different spaces that are important, the respective inner products, and the adjoints of the linear transformations defined between these spaces.

Using the fact that Γ_{prior} is a self-adjoint operator on \mathcal{V}_n and the form of the adjoint operator \mathbf{F}^* (see Figure 2.1), we can rewrite the expression for Γ_{post} as follows:

$$\Gamma_{\text{post}} = \Gamma_{\text{prior}}^{1/2} \mathbf{M}^{-1/2} \left(\mathbf{I} + \mathcal{F}^{\top} \Gamma_{\text{noise}}^{-1} \mathcal{F} \right)^{-1} \mathbf{M}^{1/2} \Gamma_{\text{prior}}^{1/2},$$

with

$$\mathcal{F} = \mathbf{F} \Gamma_{\text{prior}}^{1/2} \mathbf{M}^{-1/2}. \quad (2.5)$$

Note that the operator $\mathcal{H}_{\text{m}} = \mathcal{F}^{\top} \Gamma_{\text{noise}}^{-1} \mathcal{F}$ is a symmetric positive semidefinite matrix, and is a similarity transform of the prior-preconditioned data-misfit Hessian $\Gamma_{\text{prior}}^{1/2} \mathbf{F}^* \Gamma_{\text{noise}}^{-1} \mathbf{F} \Gamma_{\text{prior}}^{1/2}$. In many applications, \mathcal{H}_{m} has rapidly decaying eigenvalues and therefore, it can be approximated

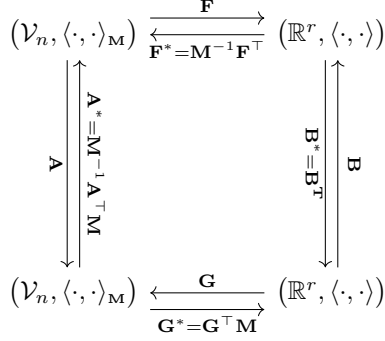


Figure 2.1 Different spaces, their inner products, and the adjoints of linear transformations between them. Here $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product and $\langle \cdot, \cdot \rangle_M$ is the mass-weighted inner product.

by a low-rank matrix. This is a key insight that will be exploited in our estimators for the OED criterion and its gradient in Chapter 3.

2.3 Optimal design of experiments

As mentioned in Chapter 1, in our work, an experimental design refers to a placement of sensors used to collect measurement data for the purposes of parameter inversion. Here we describe the setup of the optimization problem for finding an optimal experimental design.

Experimental design and dependence of the posterior covariance on the design

We seek to find an optimal subset of a network of n_s candidate sensor locations, which collect measurements at n_t points in time. The experimental design is parameterized by a vector of design weights $\mathbf{w} \in [0, 1]^{n_s}$, where each weight value indicates the importance of the respective sensor location.

We follow the same setup as [5]. Namely, the design weights enter the Bayesian inverse problem through the data likelihood, so that the \mathcal{H}_m operator depends on \mathbf{w} . To weight the importance of the spatiotemporal observations, the matrix \mathbf{W} is defined as:

$$\mathbf{W} = \sum_{i=1}^{n_s} w_i \mathbf{E}_i, \quad \text{with} \quad \mathbf{E}_i = \mathbf{I}_{n_t} \otimes \mathbf{e}_i \mathbf{e}_i^\top,$$

where \otimes is the Kronecker product. Therefore, $\mathcal{H}_m(\mathbf{w})$ is expressed as follows:

$$\mathcal{H}_m(\mathbf{w}) = \mathcal{F}^\top \mathbf{W}^{1/2} \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W}^{1/2} \mathcal{F},$$

We refer to [5] for details. This results in the \mathbf{w} dependent posterior covariance operator:

$$\mathbf{\Gamma}_{\text{post}}(\mathbf{w}) = \mathbf{\Gamma}_{\text{prior}}^{1/2} \mathbf{M}^{-1/2} (\mathcal{H}_m(\mathbf{w}) + \mathbf{I})^{-1} \mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}}^{1/2}. \quad (2.6)$$

In our formulation, we assume uncorrelated observations across sensor locations and time which implies $\mathbf{\Gamma}_{\text{noise}}$ is a $n_s n_t \times n_s n_t$ block diagonal matrix, with n_t blocks of the form $\text{diag}(\sigma_1^2, \dots, \sigma_{n_s}^2)$; here σ_i^2 , $i = 1, \dots, n_s$, denote the measurement noise at individual sensor locations. Using this structure for $\mathbf{\Gamma}_{\text{noise}}$, we define

$$\mathbf{W}^{\text{noise}} \equiv \mathbf{W}^{1/2} \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W}^{1/2} = \mathbf{\Gamma}_{\text{noise}}^{-1/2} \mathbf{W} \mathbf{\Gamma}_{\text{noise}}^{-1/2} = \sum_{i=1}^{n_{\text{sens}}} w_i \mathbf{E}_i^{\text{noise}}, \quad (2.7)$$

with $\mathbf{E}_i^{\text{noise}} = \sigma_i^{-2} \mathbf{I}_{n_{\text{time}}} \otimes \mathbf{e}_i \mathbf{e}_i^\top$. Thus, we have $\mathcal{H}_{\text{m}}(\mathbf{w}) = \mathcal{F}^\top \mathbf{W}^{\text{noise}} \mathcal{F}$.

Optimization problem for finding an optimal design

Recall the optimization problem from (1.1). As explained before, to enable efficient solution methods for the above optimization problem we need (i) a numerical method for fast computation of $\Phi(\mathbf{w})$ and its gradient, and (ii) a choice of penalty function that promotes sparse and binary weights. The former is facilitated by the randomized subspace iteration approach outlined in Section 2.5 (see Chapter 3 and 5), and for the latter, we present an approach based on reweighted ℓ_1 minimization (see Chapter 4).

2.4 Design criteria

There are many ways to measure the uncertainty in the solution to the inverse problem [20, 104]. Common design criteria (in finite-dimensions) include:

- A-optimal: $\text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$, the average posterior variance;
- C-optimal: $\mathbf{c}^\top \mathbf{\Gamma}_{\text{post}}(\mathbf{w}) \mathbf{c}$, the variance of a given linear combination of parameters;
- D-optimal: $\log \det(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$, the expected information gain (measured by the Kullback-Leibler divergence); and
- E-optimal: $\lambda_{\max}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$, the worst case uncertainty defined by the largest eigenvalue of the posterior covariance operator.

In our work, we focus on the A- and D-optimal criterion to compute optimal designs. Therefore, we discuss the two criterion and some useful properties that aid us in our discussion regarding solving the optimal experimental design problem in (1.1).

2.4.1 A-optimal criterion

A large part of this thesis focuses on A-optimal designs. Here, we discuss the A-optimal criterion in terms of an infinite dimensional posterior covariance operator, the convexity of the criterion, and the gradient (with respect to \mathbf{w}) of the criterion.

The infinite-dimensional A-optimal criterion

Since the posterior covariance is a discretization of an infinite-dimensional operator, we need to ensure the criterion we use makes sense in the infinite-dimensional limit. Let $m(\mathbf{x})$ be Gaussian random field on the domain \mathcal{D} . And let its covariance operator, Γ_{post} , be positive, self-adjoint, and trace-class. Then the A-optimal criterion, measuring the average posterior variance, in infinite dimensions is

$$\frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} c_{\text{post}}(\mathbf{x}, \mathbf{x}) d\mathbf{x},$$

where $c_{\text{post}}(\mathbf{x}, \mathbf{y})$ is the covariance function of $m(\mathbf{x})$ defined in Definition 5 and $c_{\text{post}}(\mathbf{x}, \mathbf{y})$ is assumed to be a continuous kernel on \mathcal{D} . Then, using Mercer's theorem [38], we provide justification that the A-optimal criterion for an infinite-dimensional operator is equivalent to the finite-dimensional criterion, i.e., the average variance of an infinite-dimensional operator is equivalent to $\frac{1}{|\mathcal{D}|} \text{tr}(\Gamma_{\text{post}})$. Note that for the moment, we suppress the dependency of the posterior covariance operator on the weights \mathbf{w} .

Since Γ_{post} is self-adjoint and trace-class, it has real eigenvalues λ_i for $i = 1, 2, \dots$ and a complete set of orthonormal eigenvectors $\phi_i \in L^2(\mathcal{D})$, $i = 1, 2, \dots$. Also, since Γ_{post} is positive, $\lambda_i \geq 0$ for $i = 1, 2, \dots$.

Recall the integral operator definition of the covariance operator in (2.1). The next step is to apply Mercer's theorem to obtain a spectral representation of $c_{\text{post}}(\mathbf{x}, \mathbf{x})$. Therefore, we have

$$c_{\text{post}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y}),$$

where λ_i for $i = 1, 2, \dots$ are real eigenvalues and $\{\phi_i(\mathbf{x})\}_{i=1}^{\infty}$ is the complete set of orthonormal eigenvectors of Γ_{post} . This series converges absolutely and uniformly.

Using the expansion of c_{post} , we express the average variance as

$$\begin{aligned} \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} c_{\text{post}}(\mathbf{x}, \mathbf{x}) d\mathbf{x} &= \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} \sum_{i=1}^{\infty} \lambda_i \phi_i^2(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} \lim_{k \rightarrow \infty} \sum_{i=1}^k \lambda_i \phi_i^2(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{|\mathcal{D}|} \lim_{k \rightarrow \infty} \sum_{i=1}^k \lambda_i = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{\infty} \lambda_i = \frac{1}{|\mathcal{D}|} \text{tr}(\Gamma_{\text{post}}), \end{aligned}$$

where the limit and integral are interchanged by the Monotone Convergence Theorem [87, Theorem 11.30].

Thus, the A-optimal criterion naturally extends to the infinite-dimensional case and is given in terms of the trace of the posterior covariance operator (the constant $\frac{1}{|\mathcal{D}|}$ is dropped when formulating the OED problem).

Convexity of the criterion

We now verify the convexity of the A-optimal criterion. First, we define some notation. We denote by \mathcal{S}_M the space of self-adjoint operators on $(\mathbb{R}^n, \langle \cdot, \cdot \rangle_M)$ and by \mathcal{S}_M^{++} cone of self-adjoint and positive definite operators on $(\mathbb{R}^n, \langle \cdot, \cdot \rangle_M)$. Note that $\mathbf{A} \in \mathcal{S}_M^{++}$ satisfies

$$\begin{aligned}\langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle_M &= \langle \mathbf{x}, \mathbf{A}\mathbf{y} \rangle_M \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \\ \langle \mathbf{A}\mathbf{x}, \mathbf{x} \rangle_M &> 0, \quad \text{for all } \mathbf{x} \neq \mathbf{0}.\end{aligned}$$

Recall the posterior covariance operator for a Bayesian linear inverse problem with Gaussian prior and additive Gaussian noise model in (2.6). By construction, $\mathbf{\Gamma}_{\text{post}}(\mathbf{w}) \in \mathcal{S}_M^{++}$, for all $\mathbf{w} \in \mathbf{R}_{\geq 0}^n$, where $\mathbf{R}_{\geq 0}^n$ denotes the non-negative orthant in \mathbb{R}^n . Here we seek to prove that the mapping $\mathbf{w} \mapsto \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$, i.e., the A-optimal design criterion, is convex on $\mathbf{R}_{\geq 0}^n$ and strictly convex provided an additional assumption is satisfied.

An important step in showing the convexity of the A-optimal criterion involves showing strict convexity of the function $G(\mathbf{X}) = \text{tr}(\mathbf{X}^{-1})$ on $\text{dom}(G) = \mathcal{S}_M^{++}$ [104, Theorem B.27]. We include the justification for completeness of this section. To show this we recall a result from [13] that a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is (strictly) convex if and only if its domain, $\text{dom}(f)$, is convex and that the function $g : \mathbb{R} \rightarrow \mathbb{R}$, defined by

$$g(t) = f(\mathbf{x} + t\mathbf{v}), \quad \text{dom}(g) := \{t : \mathbf{x} + t\mathbf{v} \in \text{dom}(f)\},$$

is a (strictly) convex function of t , for any $\mathbf{x} \in \text{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$. In the present setting, we consider the function $G : \mathcal{S}_M \rightarrow \mathbb{R}$ with $\text{dom}(G) = \mathcal{S}_M^{++}$.

Lemma 2.4.0.1. *The function $G(\mathbf{X}) = \text{tr}(\mathbf{X}^{-1})$ is strictly convex on $\text{dom}(G) = \mathcal{S}_M^{++}$.*

Proof. Consider the restriction of G to a line, $\mathbf{S} + t\mathbf{B}$, where $\mathbf{S} \in \mathcal{S}_M^{++}$ and \mathbf{B} is self-adjoint; that is, we consider the function

$$g(t) = G(\mathbf{S} + t\mathbf{B}),$$

defined for $t \in \text{dom}(g)$, where $\text{dom}(g)$ is an interval of values of t for which $\mathbf{S} + t\mathbf{B} \in \mathcal{S}_M^{++}$. Consider $\mathbf{V} = \mathbf{S}^{-1/2}\mathbf{B}\mathbf{S}^{-1/2}$ and let $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^*$ be its spectral decomposition; here \mathbf{Q} is an orthogonal matrix with eigenvectors of \mathbf{V} as its columns, $\mathbf{Q} = [\mathbf{q}_1 \ \cdots \ \mathbf{q}_n]$, and $\mathbf{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues on its diagonal. It is straightforward to note that $\mathbf{I} + t\mathbf{V} \in \mathcal{S}_M^{++}$ for

$t \in \text{dom}(g)$ and that

$$\begin{aligned}
g(t) &= \text{tr} \left((\mathbf{S} + t\mathbf{B})^{-1} \right) \\
&= \text{tr} \left(\mathbf{S}^{-1/2} (\mathbf{I} + t\mathbf{S}^{-1/2}\mathbf{B}\mathbf{S}^{-1/2})^{-1} \mathbf{S}^{-1/2} \right) \\
&= \text{tr} \left(\mathbf{S}^{-1} (\mathbf{I} + t\mathbf{V})^{-1} \right) \\
&= \sum_{i=1}^n \langle \mathbf{S}^{-1} (\mathbf{I} + t\mathbf{V})^{-1} \mathbf{q}_i, \mathbf{q}_i \rangle_{\mathbf{M}} \\
&= \sum_{i=1}^n (1 + t\lambda_i)^{-1} \langle \mathbf{S}^{-1} \mathbf{q}_i, \mathbf{q}_i \rangle_{\mathbf{M}}.
\end{aligned}$$

Now, $c_i = \langle \mathbf{S}^{-1} \mathbf{q}_i, \mathbf{q}_i \rangle_{\mathbf{M}} > 0$, for $i = 1, \dots, n$, and $h_i(t) = (1 + t\lambda_i)^{-1}$ is strictly convex for $t \in \text{dom}(g)$. To see the latter, note that $(1 + t\lambda_i)^{-1} > 0$, $i = 1, \dots, n$, for $t \in \text{dom}(g)$, and that

$$h_i''(t) = 2\lambda_i^2(1 + t\lambda_i)^{-3} > 0, \quad t \in \text{dom}(g), i = 1, \dots, n.$$

Hence, $g(t) = \sum_{i=1}^n c_i h_i(t)$ is a positive weighted sum of strictly convex functions, and is thus strictly convex. Therefore, G is strictly convex on $\text{dom}(G) = \mathcal{S}_{\mathbf{M}}^{++}$. \square

We are now ready to prove the following result:

Theorem 1. *The function $\Psi : \mathbf{R}_{\geq 0}^n \rightarrow \mathbb{R}$, given by $\Psi(\mathbf{w}) = \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$ is convex.*

Proof. Note that $\mathbf{\Gamma}_{\text{post}}(\mathbf{w}) = \mathbf{H}(\mathbf{w})^{-1}$, where $\mathbf{H}(\mathbf{w}) = \mathbf{F}^* \mathbf{W}^{\text{noise}} \mathbf{F} + \mathbf{\Gamma}_{\text{prior}}^{-1}$. Clearly $\mathbf{H}(\mathbf{w})$ is an affine function of \mathbf{w} . Moreover, $\mathbf{H}(\mathbf{w}) \in \mathcal{S}_{\mathbf{M}}^{++}$, for every $\mathbf{w} \in \mathbf{R}_{\geq 0}^n$. Let $\mathbf{w}_1, \mathbf{w}_2 \in \mathbf{R}_{\geq 0}^n$ with $\mathbf{w}_1 \neq \mathbf{w}_2$ and let $0 < \alpha < 1$. To simplify the notation, we denote $\mathbf{H}_i = \mathbf{H}(\mathbf{w}_i)$, $i = 1, 2$. Note that $\mathbf{w}_1 \neq \mathbf{w}_2$ does not imply $\mathbf{H}_1 \neq \mathbf{H}_2$ without added assumptions. We have

$$\mathbf{\Gamma}_{\text{post}}(\alpha \mathbf{w}_1 + (1 - \alpha) \mathbf{w}_2) = [\alpha \mathbf{H}_1 + (1 - \alpha) \mathbf{H}_2]^{-1}.$$

If $\mathbf{H}_1 \neq \mathbf{H}_2$, then using Lemma 2.4.0.1, we obtain

$$\begin{aligned}
\Psi(\alpha \mathbf{w}_1 + (1 - \alpha) \mathbf{w}_2) &= \text{tr}(\mathbf{\Gamma}_{\text{post}}(\alpha \mathbf{w}_1 + (1 - \alpha) \mathbf{w}_2)) \\
&= \text{tr}([\alpha \mathbf{H}_1 + (1 - \alpha) \mathbf{H}_2]^{-1}) < \alpha \text{tr}(\mathbf{H}_1^{-1}) + (1 - \alpha) \text{tr}(\mathbf{H}_2^{-1}) \\
&= \alpha \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}_1)) + (1 - \alpha) \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}_2)) = \alpha \Psi(\mathbf{w}_1) + (1 - \alpha) \Psi(\mathbf{w}_2).
\end{aligned}$$

However, if we assume $\mathbf{H}_1 = \mathbf{H}_2$, then the strict inequality in the previous derivation becomes an equality. Thus, we have, $\Psi(\alpha \mathbf{w}_1 + (1 - \alpha) \mathbf{w}_2) \leq \alpha \Psi(\mathbf{w}_1) + (1 - \alpha) \Psi(\mathbf{w}_2)$. \square

Therefore, if $\Phi(\mathbf{w}) = \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$ in (1.1), then we have a convex optimization problem, provided the penalty function is also convex. As briefly mentioned in the proof, more assumptions are required to show strict convexity. Specifically, the added assumptions must ensure that if $\mathbf{w}_1 \neq \mathbf{w}_2$ then $\mathbf{H}_1 \neq \mathbf{H}_2$. The following corollary provides a condition that ensures this property.

Corollary 1.1. Assume that $\text{rank}(\mathbf{F}) = n_s n_t$. The function $\Psi : \mathbf{R}_{\geq 0}^n \rightarrow \mathbb{R}$, given by $\Psi(\mathbf{w}) = \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$ is strictly convex.

Proof. The proof follows Theorem 1; however, we will use the added assumption to show if $\mathbf{w}_1 \neq \mathbf{w}_2$, then $\mathbf{H}_1 \neq \mathbf{H}_2$. We consider proving the contrapositive. If $\mathbf{H}_1 = \mathbf{H}_2$, then $\mathbf{F}^* \widehat{\mathbf{W}}^{\text{noise}} \mathbf{F} = \mathbf{0}$, where $\widehat{\mathbf{W}}^{\text{noise}} = \mathbf{W}_1^{\text{noise}} - \mathbf{W}_2^{\text{noise}}$ and $\mathbf{0} \in \mathbb{R}^{n_s n_t \times n_s n_t}$. We consider the rank of the matrix $\mathbf{F}^* \widehat{\mathbf{W}}^{\text{noise}} \mathbf{F}$ and apply Sylvester's rank inequality [75, Eq. (4.5.3)] twice to the expression to obtain

$$\begin{aligned} 0 = \text{rank}(\mathbf{F}^* \widehat{\mathbf{W}}^{\text{noise}} \mathbf{F}) &\geq \text{rank}(\mathbf{F}^*) + \text{rank}(\widehat{\mathbf{W}}^{\text{noise}} \mathbf{F}) - n_s n_t \\ &\geq 2 \cdot \text{rank}(\mathbf{F}) + \text{rank}(\widehat{\mathbf{W}}^{\text{noise}}) - 2n_s n_t = \text{rank}(\widehat{\mathbf{W}}^{\text{noise}}). \end{aligned}$$

The last equality follows because we have assumed that $\text{rank}(\mathbf{F}) = n_s n_t$. Therefore, $\widehat{\mathbf{W}}^{\text{noise}} = \mathbf{0}$ and thus $\mathbf{w}_1 = \mathbf{w}_2$. \square

Intuitively, since the rows of \mathbf{F} are assumed to be linearly independent, the sensors at each location must gather “new” data, i.e., none of the data gathered can be a linear combination of the other observations.

Other (weaker) assumptions can be made to show that the A-optimal criterion is strictly convex. For instance, if the matrix set, $\{\mathbf{F}^* \mathbf{E}_j^{\text{noise}} \mathbf{F}\}_{j=1}^{n_s}$, is linearly independent, then one can use the definition of $\mathbf{W}^{\text{noise}}$ to show strict convexity of the A-optimal criterion. Since the proof is straightforward, we have omitted it. An intuitive interpretation of this assumption is that each sensor location must contribute “different” information.

A reformulation of the A-optimal criterion

Because the criterion forms part of the objective function of an optimization problem, we consider a slight variation to the traditional definition of the A-optimal criterion. Note that $\text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w})) = \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}) - \mathbf{\Gamma}_{\text{prior}}) + \text{tr}(\mathbf{\Gamma}_{\text{prior}})$ and thus, minimizing the trace of the posterior covariance operator is equivalent to minimizing

$$\Phi_{\text{aopt}}(\mathbf{w}) \equiv \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}) - \mathbf{\Gamma}_{\text{prior}}). \quad (2.8)$$

This is the objective function we seek to minimize for finding A-optimal designs. As seen below, this formulation of the A-optimal criterion is well suited for approximations via randomized matrix methods Section 2.5. Note also that minimizing $\Phi_{\text{aopt}}(\mathbf{w})$ amounts to maximizing $\text{tr}(\mathbf{\Gamma}_{\text{prior}}) - \text{tr}(\mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$, which can be thought of as a measure of uncertainty reduction.

We can also interpret (2.8) using a decision theoretic point of view. It is well known [3, 9, 20] that for Bayesian linear inverse problems with Gaussian prior and additive Gaussian noise

models, $\text{tr}(\mathbf{\Gamma}_{\text{post}})$ coincides with Bayes risk (with respect to the L^2 loss function):

$$\begin{aligned}\text{tr}(\mathbf{\Gamma}_{\text{post}}) &= \mathbb{E}_{\mu_{\text{pr}}} \left(\mathbb{E}_{\pi_{\text{like}}(\mathbf{y}|\mathbf{m})} (\|\mathbf{m}_{\text{post}}^{\mathbf{y}} - \mathbf{m}\|_{\mathbf{M}}^2) \right) \\ &= \int_{\mathcal{V}_n} \int_{\mathbb{R}^{n_s n_t}} \|\mathbf{m}_{\text{post}}^{\mathbf{y}} - \mathbf{m}\|_{\mathbf{M}}^2 \pi_{\text{like}}(\mathbf{y} | \mathbf{m}) d\mathbf{y} \mu_{\text{pr}}(d\mathbf{m}).\end{aligned}$$

Here μ_{pr} denotes the discretized prior measure, $\mu_{\text{pr}} = \mathcal{N}(\mathbf{m}_{\text{pr}}, \mathbf{\Gamma}_{\text{prior}})$. Using

$$\int_{\mathcal{V}_n} \|\mathbf{m}_{\text{pr}} - \mathbf{m}\|_{\mathbf{M}}^2 \mu_{\text{pr}}(d\mathbf{m}) = \text{tr}(\mathbf{\Gamma}_{\text{prior}}),$$

we see that

$$\text{tr}(\mathbf{\Gamma}_{\text{post}} - \mathbf{\Gamma}_{\text{prior}}) = \mathbb{E}_{\mu_{\text{pr}}} \left(\mathbb{E}_{\pi_{\text{like}}(\mathbf{y}|\mathbf{m})} (\|\mathbf{m}_{\text{post}}^{\mathbf{y}} - \mathbf{m}\|_{\mathbf{M}}^2 - \|\mathbf{m}_{\text{pr}} - \mathbf{m}\|_{\mathbf{M}}^2) \right);$$

this provides an alternate interpretation of $\Phi_{\text{aopt}}(\cdot)$ as a Bayes risk with respect to a modified loss function.

The gradient of the A-optimal criterion

We now specialize the optimization problem (1.1) to the case of A-optimal sensor placement for linear inverse problems governed by time-dependent PDEs:

$$\min_{\mathbf{w} \in [0,1]^{n_s}} \Phi_{\text{aopt}}(\mathbf{w}) + \gamma P(\mathbf{w}). \quad (2.9)$$

We utilize the expanded formulation of the posterior covariance operator in (2.6) to express the A-optimal criterion as

$$\begin{aligned}\Phi_{\text{aopt}}(\mathbf{w}) &= \text{tr} \left(\mathbf{\Gamma}_{\text{prior}}^{1/2} \mathbf{M}^{-1/2} [(\mathbf{\mathcal{H}}_{\text{m}}(\mathbf{w}) + \mathbf{I})^{-1} - \mathbf{I}] \mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}}^{1/2} \right) \\ &= \text{tr} \left([(\mathbf{\mathcal{H}}_{\text{m}}(\mathbf{w}) + \mathbf{I})^{-1} - \mathbf{I}] \mathbf{Z} \right),\end{aligned} \quad (2.10)$$

with

$$\mathbf{Z} \equiv \mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1/2}. \quad (2.11)$$

Also, anticipating that we will use a gradient-based solver for solving (2.9), we need the gradient of $\Phi_{\text{aopt}}(\mathbf{w})$. Using Theorems B.17 and B.19 in [104], the partial derivatives of (2.10) with respect to w_j , $j = 1, \dots, n_s$, are

$$\partial_j \Phi_{\text{aopt}}(\mathbf{w}) = -\text{tr} \left((\mathbf{I} + \mathbf{\mathcal{H}}_{\text{m}}(\mathbf{w}))^{-1} \partial_j \mathbf{\mathcal{H}}_{\text{m}}(\mathbf{w}) (\mathbf{I} + \mathbf{\mathcal{H}}_{\text{m}}(\mathbf{w}))^{-1} \mathbf{Z} \right). \quad (2.12)$$

(We have used the notation ∂_j to denote $\frac{\partial}{\partial w_j}$.) Note that using the definition of $\mathbf{\mathcal{H}}_{\text{m}}(\mathbf{w})$, we have $\partial_j \mathbf{\mathcal{H}}_{\text{m}}(\mathbf{w}) = \mathbf{\mathcal{F}}^\top \mathbf{E}_j^{\text{noise}} \mathbf{\mathcal{F}}$, $j = 1, \dots, n_s$.

2.4.2 D-optimal criterion

Here we consider the infinite-dimensional D-optimal criterion and its optimization problem. Since the D-optimal criterion is not the main focus of our work, we simply state and cite the result.

The finite dimensional D-optimal criterion, $\log \det(\mathbf{\Gamma}_{\text{post}})$, is not meaningful in infinite dimensions due to the posterior covariance operator being trace-class, i.e., the eigenvalues have zero as the accumulation point. Instead, we use the definition of the D-optimal criterion derived in [3, 6] in the infinite-dimensional case. The expected information gain is defined as

$$\Phi_{\text{dopt}}(\mathbf{w}) \equiv \frac{1}{2} \log \det(\mathbf{I} + \mathbf{\Gamma}_{\text{prior}}^{1/2} \mathbf{F}^* \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W} \mathbf{F} \mathbf{\Gamma}_{\text{prior}}^{1/2}). \quad (2.13)$$

This quantity remains meaningful in the infinite-dimensional limit. Then, using this definition, we define a D-optimal design to be the solution to the optimization problem,

$$\min_{\mathbf{w} \in [0,1]^{n_s}} -\Phi_{\text{dopt}}(\mathbf{w}) + \gamma P(\mathbf{w}).$$

Since the D-optimal criterion is used as part of the objective function for an optimization problem, for the remainder of the thesis, we remove the factor of $\frac{1}{2}$ that appears in (2.13).

2.5 Randomized subspace iteration algorithm

In this section, we discuss the randomized subspace iteration algorithm. We begin by explaining how to compute a low-rank approximation of a symmetric positive semidefinite matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$. To draw connection with the previous subsection, in our application, \mathbf{A} will stand for \mathcal{H}_{m} . We first draw a random Gaussian matrix $\mathbf{\Omega} \in \mathbb{R}^{n \times \ell}$ (i.e., the entries are independent and identically distributed standard normal random variables). We then perform q steps of subspace iteration on \mathbf{A} with the starting guess $\mathbf{\Omega}$ to obtain the matrix \mathbf{Y} . If, for example, the matrix has rank $k \leq \ell$, or the eigenvalues decay sufficiently, then the range of \mathbf{Y} is a good approximation to the range of \mathbf{A} under these suitable conditions. This is the main insight behind randomized algorithms. We now show how to obtain a low-rank approximation of \mathbf{A} . A thin-QR factorization of \mathbf{Y} is performed to obtain the matrix \mathbf{Q} , which has orthonormal columns. We then form the “projected” matrix $\mathbf{T} = \mathbf{Q}^\top \mathbf{A} \mathbf{Q}$ and obtain the low-rank approximation

$$\mathbf{A} \approx \mathbf{Q} \mathbf{T} \mathbf{Q}^\top. \quad (2.14)$$

An alternative low-rank approximation can be computed using the Nyström approximation, see e.g., [43]. Once the matrix \mathbf{T} is computed, it can be used in various ways. In [89], $\text{tr}(\mathbf{T})$ was used as an estimator for $\text{tr}(\mathbf{A})$, whereas $\log \det(\mathbf{I} + \mathbf{T})$ was used as estimator for $\log \det(\mathbf{I} + \mathbf{A})$. The main idea behind these estimators is that the eigenvalues of \mathbf{T} are good approximations to the eigenvalues of \mathbf{A} , when \mathbf{A} is sufficiently low-rank or has rapidly decaying eigenvalues. Our

estimators for the A-optimal criterion and its gradient utilize the same idea but in a slightly different form.

Algorithm 1 $[\mathbf{Q}, \mathbf{T}] = \text{randSubspaceIteration}(\mathbf{A}, k, p, q)$

Input: $\mathbf{A} \in \mathbb{R}^{n \times n}$ with target rank k , oversampling parameter $p \geq 2$, with $\ell \equiv k + p \leq n$ and $q \geq 1$ (number of subspace iterations).

Output: $\mathbf{Q} \in \mathbb{R}^{n \times \ell}$, $\mathbf{T} \in \mathbb{R}^{\ell \times \ell}$.

- 1: **Draw** a standard Gaussian random matrix $\mathbf{\Omega} \in \mathbb{R}^{n \times \ell}$.
 - 2: **Compute** $\mathbf{Y} = \mathbf{A}^q \mathbf{\Omega}$.
 - 3: **Compute** thin QR decomposition $\mathbf{Y} = \mathbf{Q}\mathbf{R}$.
 - 4: **Compute** $\mathbf{T} = \mathbf{Q}^\top \mathbf{A} \mathbf{Q}$.
-

In our work, we consider an eigenvalue decomposition of the matrix $\mathbf{T} = \mathbf{U}\hat{\mathbf{\Lambda}}\mathbf{U}^\top$ so that (2.14) becomes

$$\mathbf{A} \approx \mathbf{Q}\mathbf{T}\mathbf{Q}^\top = \mathbf{Q}\mathbf{U}\hat{\mathbf{\Lambda}}\mathbf{U}^\top\mathbf{Q}^\top = \mathbf{V}\hat{\mathbf{\Lambda}}\mathbf{V}^\top,$$

where $\mathbf{V} = \mathbf{Q}\mathbf{U}$ and $\hat{\mathbf{\Lambda}}$ are the approximate eigenvalues of \mathbf{A} captured in the matrix \mathbf{T} . We define a procedure for computing this specific low-rank approximation of a matrix \mathbf{A} in Algorithm 2, which will be used throughout the thesis.

Algorithm 2 $[\mathbf{V}, \mathbf{\Lambda}] = \text{randEig}(\mathbf{A}, k, p)$

Input: $\mathbf{A} \in \mathbb{R}^{n \times n}$, target rank k , oversampling parameter $p \geq 2$, with $\ell \equiv k + p \leq n$.

Output: $\mathbf{V} \in \mathbb{R}^{n \times \ell}$, $\mathbf{\Lambda} \in \mathbb{R}^{\ell \times \ell}$.

- 1: Compute $[\mathbf{Q}, \mathbf{T}] = \text{randSubspaceIteration}(\mathbf{A}, k, p, 1)$.
 - 2: Compute eigendecomposition $[\mathbf{U}, \mathbf{\Lambda}]$ of \mathbf{T} and form $\mathbf{V} = \mathbf{Q}\mathbf{U}$.
 - 3: **Return** \mathbf{V} and $\mathbf{\Lambda}$.
-

CHAPTER

3

EFFICIENT COMPUTATION OF A-OPTIMAL CRITERION AND ITS GRADIENT

The computational cost of solving (2.9) is dominated by the PDE solves required in OED objective and gradient evaluations; these operations need to be performed repeatedly when using an optimization algorithm for solving (2.9). Therefore, to enable computing A-optimal designs for large-scale applications, efficient methods for objective and gradient computations are needed. In this chapter, we derive efficient and accurate randomized estimators for (2.10) and (2.12). The proposed estimators are matrix-free—they require only applications of the (prior-preconditioned) forward operator \mathcal{F} and its adjoint on vectors. Moreover, the computational cost, measured in the number of PDE solves, does not increase with the discretized parameter dimension. This is due to the fact that our estimators exploit the low-rank structure of $\mathcal{H}_m(\mathbf{w})$, a problem property that is independent of the choice of discretization. We also present a novel modified A-optimal criterion to further reduce the computational cost. Randomized estimators for this criterion are developed and error bounds are also discussed.

We introduce our proposed randomized estimators for the A-optimal design criterion $\Phi_{\text{aopt}}(\mathbf{w})$ and its gradient $\nabla\Phi_{\text{aopt}}(\mathbf{w})$ in Section 3.1. Additionally, in this section, we present a detailed computational procedure and analyze the errors associated with our proposed estimators. Section 3.2 introduces the modified A-optimal criterion. The randomized estimators are derived and the computational procedure is presented. We also discuss the error associated with the

modified A-optimal estimators in this section. In Section 3.3, we present numerical results that indicate the accuracy of our estimators and illustrates the bounds we prove in Section 3.4. Lastly, Section 3.5 includes concluding remarks and future directions of the work discussed in this chapter.

3.1 Randomized estimators for $\Phi_{\text{aopt}}(\mathbf{w})$ and its gradient

Consider the low-rank approximation of $\mathcal{H}_m(\mathbf{w})$ given by $\hat{\mathcal{H}}_m(\mathbf{w}) = \mathbf{Q}(\mathbf{w})\mathbf{T}(\mathbf{w})\mathbf{Q}^\top(\mathbf{w})$, with $\mathbf{Q}(\mathbf{w})$ and $\mathbf{T}(\mathbf{w})$ computed using Algorithm 1. Replacing $\mathcal{H}_m(\mathbf{w})$ by its approximation and using the cyclic property of the trace, we obtain the estimator for the A-optimal criterion (2.10):

$$\hat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell) = \text{tr} \left(\left((\mathbf{I} + \hat{\mathcal{H}}_m(\mathbf{w}))^{-1} - \mathbf{I} \right) \mathbf{Z} \right), \quad (3.1)$$

where \mathbf{Z} is as in (2.11).

To derive an estimator for the gradient, once again, we replace $\mathcal{H}_m(\mathbf{w})$ with its low-rank approximation $\hat{\mathcal{H}}_m(\mathbf{w})$ in (2.12) to obtain

$$\widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}; \ell) = -\text{tr} \left((\mathbf{I} + \hat{\mathcal{H}}_m(\mathbf{w}))^{-1} \mathcal{F}^\top \mathbf{E}_j^{\text{noise}} \mathcal{F} (\mathbf{I} + \hat{\mathcal{H}}_m(\mathbf{w}))^{-1} \mathbf{Z} \right), \quad (3.2)$$

for $j = 1, \dots, n_s$.

3.1.1 Computational procedure

First, we discuss computation of the A-optimal criterion estimator using Algorithm 1. Typically, the algorithm can be used with $q = 1$, due to rapid decay of eigenvalues of $\mathcal{H}_m(\mathbf{w})$. In this case, Algorithm 1 requires 2ℓ applications of $\mathcal{H}_m(\mathbf{w})$. Since each application of $\mathcal{H}_m(\mathbf{w})$ requires one \mathcal{F} apply (forward solve) and one \mathcal{F}^\top apply (adjoint solve), computing $\hat{\mathcal{H}}_m(\mathbf{w})$ requires 4ℓ PDE solves. Letting the spectral decomposition of the $\ell \times \ell$ matrix $\mathbf{T}(\mathbf{w})$ be given by $\mathbf{T}(\mathbf{w}) = \mathbf{U}(\mathbf{w})\mathbf{\Lambda}_T(\mathbf{w})\mathbf{U}(\mathbf{w})^\top$ and denoting $\mathbf{V}(\mathbf{w}) \equiv \mathbf{Q}(\mathbf{w})\mathbf{U}(\mathbf{w})$, we have $\hat{\mathcal{H}}_m(\mathbf{w}) = \mathbf{V}(\mathbf{w})\mathbf{\Lambda}_T(\mathbf{w})\mathbf{V}^\top(\mathbf{w})$. Applying the Sherman–Morrison–Woodbury formula [75] and the cyclic property of the trace to (3.1), we obtain

$$\hat{\Phi}_{\text{aopt}}(\mathbf{w}) = -\text{tr} \left(\mathbf{D}_T(\mathbf{w}) \mathbf{V}^\top(\mathbf{w}) \mathbf{Z} \mathbf{V}(\mathbf{w}) \right), \quad (3.3)$$

where $\mathbf{D}_T(\mathbf{w}) = \mathbf{\Lambda}_T(\mathbf{w})(\mathbf{I} + \mathbf{\Lambda}_T(\mathbf{w}))^{-1}$. To simplify notation, the dependence of \mathbf{w} is suppressed for the operators used in computing the estimators for the remainder of the chapter; however, the notation is retained for $\mathcal{H}_m(\mathbf{w})$ and $\hat{\mathcal{H}}_m(\mathbf{w})$.

Next, we describe computation of the gradient estimator (3.2). Here we assume $n_s n_t \leq n$; the extension to the case $n_s n_t > n$ is straightforward and is omitted. Again, using the Woodbury formula and cyclic property of the trace, we rewrite (3.2) as

$$\widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}; \ell) = -\text{tr} \left(\mathcal{F} (\mathbf{I} - \mathbf{V} \mathbf{D}_T \mathbf{V}^\top) \mathbf{Z} (\mathbf{I} - \mathbf{V} \mathbf{D}_T \mathbf{V}^\top) \mathcal{F}^\top \mathbf{E}_j^{\text{noise}} \right) \quad (3.4)$$

for $j = 1, \dots, n_s$. Expanding this expression, we obtain

$$\begin{aligned} \widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}) = & -\text{tr} \left(\mathbf{Z} \mathbf{F}^\top \mathbf{E}_j^{\text{noise}} \mathbf{F} \right) + 2\text{tr} \left(\mathbf{F} \mathbf{V} \mathbf{D}_T \mathbf{V}^\top \mathbf{Z} \mathbf{F}^\top \mathbf{E}_j^{\text{noise}} \right) - \\ & \text{tr} \left(\mathbf{F} \mathbf{V} \mathbf{D}_T \mathbf{V}^\top \mathbf{Z} \mathbf{V} \mathbf{D}_T \mathbf{V}^\top \mathbf{F}^\top \mathbf{E}_j^{\text{noise}} \right). \end{aligned} \quad (3.5)$$

Note that the first term $s_j = -\text{tr} \left(\mathbf{Z} \mathbf{F}^\top \mathbf{E}_j^{\text{noise}} \mathbf{F} \right)$ in (3.5) does not depend on the design \mathbf{w} , for $j = 1, \dots, n_s$. As a result, this term can be precomputed and used in subsequent function evaluations. We expand s_j to be

$$\begin{aligned} s_j = & -\text{tr} \left(\mathbf{Z} \mathbf{F}^\top \mathbf{E}_j^{\text{noise}} \mathbf{F} \right) \\ = & -\sum_{k=1}^{n_s n_t} \left(\mathbf{F}^\top (\mathbf{E}_j^{\text{noise}})^{1/2} \widehat{\mathbf{e}}_k \right)^\top \mathbf{Z} \left(\mathbf{F}^\top (\mathbf{E}_j^{\text{noise}})^{1/2} \widehat{\mathbf{e}}_k \right), \end{aligned}$$

where $\widehat{\mathbf{e}}_k$ is the k^{th} column of the identity matrix of size $n_s n_t$. Because there are only n_t columns of $\mathbf{E}_j^{\text{noise}}$ with nonzero entries, the total cost to precompute s_j for $j = 1, \dots, n_s$ is $n_s n_t$ PDE solves.

To compute the remaining terms in (3.5), we exploit the fact that \mathbf{V} has ℓ columns. Notice all the other occurrences of \mathbf{F} and \mathbf{F}^\top occur as a combination of $\mathbf{F} \mathbf{V}$ and $\mathbf{F} \mathbf{Z} \mathbf{V}$ (or of their transposes). Both of these terms require ℓ PDE solves to compute. As a result, the total cost to evaluate $\widehat{\Phi}_{\text{aopt}}(\mathbf{w})$ and $\widehat{\nabla} \Phi_{\text{aopt}}(\mathbf{w}; \ell)$ is 4ℓ PDE solves to apply Algorithm 1 and 2ℓ PDE solves to compute $\mathbf{F} \mathbf{V}$ and $\mathbf{F} \mathbf{Z} \mathbf{V}$. We detail the steps for computing our estimators for A-optimal criterion and its gradient in Algorithm 3.

Alternative approaches and summary of computational cost

A closely related variation of Algorithm 3 is obtained by replacing step 1 of the algorithm (i.e., randomized subspace iteration) by the solution of an eigenvalue problem to compute the dominant eigenvalues of $\mathbf{H}_m(\mathbf{w})$ “exactly”. We refer to this method as Eig- k , where k is the target rank of $\mathbf{H}_m(\mathbf{w})$. This idea was explored for computing Bayesian D-optimal designs in [6]. The resulting cost is similar to that of the randomized method: it would cost $\mathcal{O}(k)$ PDE solves per iteration to compute the spectral decomposition of $\mathbf{H}_m(\mathbf{w})$, plus $\min\{n_s n_t, n\}$ PDE solves to precompute s_j for $j = 1, \dots, n_s$. While both the randomized and Eig- k methods provide a viable scheme for computing the A-optimal criteria, our randomized method can exploit parallelism to lower computational costs. Each matrix-vector application with $\mathbf{H}_m(\mathbf{w})$ in Algorithm 1 can be computed in parallel. However, if accurate eigenpairs of $\mathbf{H}_m(\mathbf{w})$ are of importance to the problem, one can choose to use the Eig- k approach at the cost of computing a more challenging problem.

Another possibility suitable for problems where the forward model does not depend on the design (as is the case in the present work), is to precompute a low-rank SVD of \mathbf{F} , which can then be applied as necessary to compute the A-optimal criterion and its gradient. This *frozen*

Algorithm 3 Randomized method for computing $\widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)$ and $\widehat{\nabla}\Phi_{\text{aopt}}(\mathbf{w}; \ell)$.

Input: Target rank k , oversampling parameter $p \geq 0$, design \mathbf{w} , and s_j for $j = 1, \dots, n_s$.

Output: OED objective $\widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)$ and gradient $\widehat{\nabla}\Phi_{\text{aopt}}(\mathbf{w}; \ell)$.

- 1: Apply Algorithm 1 with $\ell = k + p$ and $q = 1$ to obtain $\mathbf{T} \in \mathbb{R}^{\ell \times \ell}$ and $\mathbf{Q} \in \mathbb{R}^{n \times \ell}$.
- 2: Compute eigendecomposition $[\mathbf{U}, \mathbf{\Lambda}_{\mathbf{T}}]$ of \mathbf{T} . Let $\mathbf{D}_{\mathbf{T}} = \mathbf{\Lambda}_{\mathbf{T}}(\mathbf{I} + \mathbf{\Lambda}_{\mathbf{T}})^{-1}$.
- 3: **for** $i = 1$ to ℓ **do**
- 4: Compute $\mathbf{v}_i = \mathbf{Q}\mathbf{u}_i$, where \mathbf{u}_i are the columns of \mathbf{U} .
- 5: **end for**
- 6: Compute

$$\widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell) = - \sum_{i=1}^{\ell} d_i \mathbf{v}_i^{\top} \mathbf{Z} \mathbf{v}_i,$$

where d_i is the i^{th} diagonal of $\mathbf{D}_{\mathbf{T}}$.

- 7: **for** $i = 1$ to ℓ **do**
- 8: Compute $\mathbf{a}_i = \mathcal{F}\mathbf{v}_i$ and $\mathbf{b}_i = \mathcal{F}\mathbf{Z}\mathbf{v}_i$.
- 9: **end for**
- 10: **for** $j = 1$ to n_s **do**
- 11: Compute

$$\widehat{\partial}_j \Phi_{\text{aopt}}(\mathbf{w}; \ell) = s_j + 2 \sum_{i=1}^{\ell} d_i \mathbf{b}_i^{\top} \mathbf{E}_j^{\text{noise}} \mathbf{a}_i - \sum_{i=1}^{\ell} \sum_{k=1}^{\ell} d_i d_k \mathbf{a}_i^{\top} \mathbf{E}_j^{\text{noise}} \mathbf{a}_k \mathbf{v}_k^{\top} \mathbf{Z} \mathbf{v}_i.$$

- 12: **end for**
 - 13: Return $\widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)$ and $\widehat{\nabla}\Phi_{\text{aopt}}(\mathbf{w}; \ell) = [\widehat{\partial}_1 \Phi_{\text{aopt}}(\mathbf{w}; \ell), \dots, \widehat{\partial}_{n_s} \Phi_{\text{aopt}}(\mathbf{w}; \ell)]^{\top}$.
-

forward operator approach has been explored in [5, 42] for the A-optimal criterion and in [6] for the D-optimal criterion. The resulting PDE cost of precomputing a low-rank approximation of \mathcal{F} is $\mathcal{O}(k)$, with k indicating the target rank. The Frozen method is beneficial as no additional PDE solves are required when applying Algorithm 3; however, this approach would not favor problems where \mathcal{F} depends on \mathbf{w} , nor can the modeling errors associated with the PDE be controlled in subsequent evaluations without the construction of another operator.

Finally, if the problem size is not too large (i.e., in small-scale applications), one could explicitly construct the forward operator \mathcal{F} . This enables exact (excluding floating point errors) computation of the objective and its gradient. The total cost for evaluating \mathcal{F} involves an upfront cost of $\min\{n_s n_t, n\}$ PDE solves. We summarize the computational cost of Algorithm 3 along with the other alternatives mentioned above in Table 3.1.

To summarize, the randomized methods for computing OED objective and its gradient present several attractive features; our approach is well suited to large-scale applications; it is matrix free, simple to implement and parallelize, and exploits low-rank structure in the inverse problem. Moreover, as is the case with the Eig-k approach, the performance of the randomized subspace iteration does not degrade as the dimension of the parameter increases due to mesh refinement. This is the case because the randomized subspace iteration relies on

Table 3.1 Computational cost measured in terms of PDE solves for different methods in computing $\Phi_{\text{aopt}}(\mathbf{w})$ and $\nabla\Phi_{\text{aopt}}(\mathbf{w})$. Typically, $q = 1$ in Algorithm 1 is sufficient.

Method	$\Phi_{\text{aopt}}(\mathbf{w})$ and $\nabla\Phi_{\text{aopt}}(\mathbf{w})$	Precompute	Storage Cost
“Exact”	—	$\min\{n_s n_t, n\}$	$nn_s n_t$
Frozen	—	$\mathcal{O}(k)$	$(2n + 1)k$
Eig- k	$\mathcal{O}(k)$	$\min\{n_s n_t, n\}$	n_s
Randomized	$2(q + 2)(k + p)$	$\min\{n_s n_t, n\}$	n_s

spectral properties of the prior-preconditioned data misfit Hessian—a problem structure that is independent of discretization.

3.1.2 Error Analysis

Here we analyze the error associated with our estimators computed using Algorithm 3. For fixed $\mathbf{w} \in [0, 1]^{n_s}$, since $\mathcal{H}_m(\mathbf{w}) \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite, we can order its eigenvalues as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$. Suppose $\lambda_1, \dots, \lambda_k$ are the dominant eigenvalues of $\mathcal{H}_m(\mathbf{w})$, we define $\mathbf{\Lambda}_1 = \text{diag}(\lambda_1, \dots, \lambda_k)$ and $\mathbf{\Lambda}_2 = \text{diag}(\lambda_{k+1}, \dots, \lambda_n)$ and we assume that the eigenvalue ratio satisfies

$$\gamma_k = \|\mathbf{\Lambda}_2\|_2 \|\mathbf{\Lambda}_1^{-1}\|_2 = \frac{\lambda_{k+1}}{\lambda_k} < 1.$$

We now present the error bounds for the objective function and its gradient. To this end, we define the constant C as

$$C \equiv \frac{e^2(k+p)}{(p+1)^2} \left(\frac{1}{2\pi(p+1)} \right)^{\frac{2}{p+1}} (\mu + \sqrt{2})^2 \left(\frac{p+1}{p-1} \right), \quad (3.6)$$

with $r = \text{rank}(\mathcal{H}_m(\mathbf{w}))$ and $\mu \equiv \sqrt{r-k} + \sqrt{k+p}$.

Theorem 2. Let $\widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)$ and $\widehat{\nabla\Phi}_{\text{aopt}}(\mathbf{w}; \ell)$ be approximations of the A -optimal objective function $\Phi_{\text{aopt}}(\mathbf{w})$ and its gradient $\nabla\Phi_{\text{aopt}}(\mathbf{w})$, respectively, computed using Algorithm 3 for fixed $\mathbf{w} \in [0, 1]^{n_s}$. Recall that k is the target rank of $\mathcal{H}_m(\mathbf{w})$, $p \geq 2$ is the oversampling parameter such that $k + p \leq n$, and $q \geq 1$ is the number of subspace iterations. Assume that $\gamma_k < 1$. Then, with $f = x/(1+x)$

$$\mathbb{E} \left[|\Phi_{\text{aopt}}(\mathbf{w}) - \widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)| \right] \leq \|\mathbf{Z}\|_2 \left(\text{tr}(f(\mathbf{\Lambda}_2)) + \text{tr} \left(f(\gamma_k^{2q-1} C \mathbf{\Lambda}_2) \right) \right). \quad (3.7)$$

Furthermore, with $\mathbf{P}_j = \mathcal{F}^\top \mathbf{E}_j^{\text{noise}} \mathcal{F}$, for $j = 1, \dots, n_s$,

$$\mathbb{E} \left[|\partial_j \Phi_{\text{aopt}}(\mathbf{w}) - \partial_j \widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)| \right] \leq 2\|\mathbf{Z}\|_2 \|\mathbf{P}_j\|_2 \left(\text{tr}(f(\mathbf{\Lambda}_2)) + \text{tr} \left(f(\gamma_k^{2q-1} C \mathbf{\Lambda}_2) \right) \right). \quad (3.8)$$

Proof. See Section 3.4.2. □

In Theorem 2, the estimators are unbiased when the target rank equals the rank of $\mathcal{H}_m(\mathbf{w})$. If the eigenvalues decay rapidly, the bounds suggest that the estimators are accurate. Recall that $\text{rank}(\mathcal{H}_m(\mathbf{w})) \leq \min\{n_s n_t, n\}$ is the number of nonzero eigenvalues. Consequently, it is seen that the bounds are independent of the dimension of the discretization n .

3.2 Modified A-optimal criterion

Motivated by reducing the computational cost of computing A-optimal designs, in this section, we introduce a *modified A-optimality* criterion. As mentioned in [20], we can consider a weighted A-optimal criterion $\text{tr}(\mathbf{\Gamma} \mathbf{\Gamma}_{\text{post}}(\mathbf{w}))$, where $\mathbf{\Gamma}$ is a positive semidefinite weighting matrix. Similar to (2.8), we work with $\text{tr}(\mathbf{\Gamma}(\mathbf{\Gamma}_{\text{post}} - \mathbf{\Gamma}_{\text{prior}}))$, since the term $\text{tr}(\mathbf{\Gamma} \mathbf{\Gamma}_{\text{prior}})$ is independent of the weights \mathbf{w} . By choosing $\mathbf{\Gamma} = \mathbf{\Gamma}_{\text{prior}}^{-1}$, we obtain the modified A-optimal criterion

$$\Phi_{\text{mod}}(\mathbf{w}) \equiv \text{tr} \left((\mathbf{I} + \mathcal{H}_m(\mathbf{w}))^{-1} - \mathbf{I} \right). \quad (3.9)$$

Note that the expression for $\Phi_{\text{mod}}(\mathbf{w})$ remains meaningful in the infinite-dimensional limit. This can be seen by noting that

$$\Phi_{\text{mod}}(\mathbf{w}) = \text{tr} \left((\mathbf{I} + \mathcal{H}_m(\mathbf{w}))^{-1} - \mathbf{I} \right) = -\text{tr} \left(\mathcal{H}_m(\mathbf{w}) (\mathbf{I} + \mathcal{H}_m(\mathbf{w}))^{-1} \right),$$

and using the fact that in the infinite-dimensional limit, for every $\mathbf{w} \in [0, 1]^{n_s}$, $\mathcal{H}_m(\mathbf{w})$ is trace class and $(\mathbf{I} + \mathcal{H}_m(\mathbf{w}))^{-1}$ is a bounded linear operator.

We show in our numerical results that the modified A-optimality criterion can be useful in practice, if a cheaper alternative to the A-optimal criterion is desired, and minimizing Φ_{mod} can provide designs that lead to small posterior uncertainty.

Derivation of estimators

Here we seek to improve the efficiency of the modified A-optimal criterion by computing a randomized estimator for the modified A-optimal criterion and its gradient. As in the previous derivation of our estimators, we replace $\mathcal{H}_m(\mathbf{w})$ by its low rank approximation to obtain the randomized estimator for the modified A-optimal criterion

$$\Phi_{\text{mod}}(\mathbf{w}) \approx \text{tr} \left((\mathbf{I} + \hat{\mathcal{H}}_m(\mathbf{w}))^{-1} - \mathbf{I} \right) \equiv \hat{\Phi}_{\text{mod}}(\mathbf{w}; \ell). \quad (3.10)$$

Similarly, we use the same low-rank approximation $\hat{\mathcal{H}}_m(\mathbf{w})$ in the gradient of the modified A-optimal criterion to obtain the randomized estimator

$$\widehat{\partial_j \Phi}_{\text{mod}}(\mathbf{w}; \ell) = -\text{tr} \left((\mathbf{I} + \hat{\mathcal{H}}_m(\mathbf{w}))^{-1} \mathcal{F}^T \mathbf{E}_j^{\text{noise}} \mathcal{F} (\mathbf{I} + \hat{\mathcal{H}}_m(\mathbf{w}))^{-1} \right), \quad (3.11)$$

for $j = 1, \dots, n_s$.

3.2.1 Computational procedure

Using similar techniques as those described in Section 3.1, we can write the estimator for the modified A-optimal criterion in terms of the eigenvalues of \mathbf{T} :

$$\widehat{\Phi}_{\text{mod}}(\mathbf{w}; \ell) = -\text{tr}(\mathbf{D}_{\mathbf{T}}). \quad (3.12)$$

Moreover,

$$\begin{aligned} \widehat{\partial_j \Phi}_{\text{mod}}(\mathbf{w}) = & -\text{tr}(\mathcal{F}^\top \mathbf{E}_j^{\text{noise}} \mathcal{F}) + 2\text{tr}(\mathcal{F} \mathbf{V} \mathbf{D}_{\mathbf{T}} \mathbf{V}^\top \mathcal{F}^\top \mathbf{E}_j^{\text{noise}}) - \\ & \text{tr}(\mathcal{F} \mathbf{V} \mathbf{D}_{\mathbf{T}}^2 \mathbf{V}^\top \mathcal{F}^\top \mathbf{E}_j^{\text{noise}}). \end{aligned} \quad (3.13)$$

with $\mathbf{D}_{\mathbf{T}} = \mathbf{\Lambda}_{\mathbf{T}}(\mathbf{I} + \mathbf{\Lambda}_{\mathbf{T}})^{-1}$ and \mathbf{V} defined as in Section 3.1.

The procedure for computing the estimators for the modified A-optimal criterion follows the steps in Algorithm 3 closely. Instead of presenting an additional algorithm, we provide an overview of the computation of the estimators for the modified A-optimal criterion along with the associated computational cost in terms of the number of PDE solves.

To evaluate the estimators for the modified A-optimal criterion, the only precomputation we perform is to obtain

$$s_j = -\text{tr}(\mathcal{F}^\top \mathbf{E}_j^{\text{noise}} \mathcal{F}), \quad j = 1, \dots, n_s.$$

This term appears in the estimator for the gradient and accumulates a total cost of $n_s n_t$ PDE solves. The remaining terms in the estimators depend on a design \mathbf{w} and, in particular, the eigenvalues and eigenvectors of $\widehat{\mathcal{H}}_{\mathbf{m}}(\mathbf{w})$. As with the estimators for the A-optimal criterion, the eigenvalues and eigenvectors of $\widehat{\mathcal{H}}_{\mathbf{m}}(\mathbf{w})$ are obtained by Algorithm 1. Recall that the cost associated with Algorithm 1, with $q = 1$, is 4ℓ PDE solves. Once the eigenvalues and eigenvectors are computed, (3.12) can be evaluated without any additional PDE solves. The remaining computational effort occurs in the evaluation of the gradient. Because of our modification to the A-optimal criterion, the expression (3.13) is efficiently evaluated by computing $\mathcal{F}\mathbf{V}$. Therefore, the total cost of evaluating the estimators for the modified A-optimal criterion and its gradient is 5ℓ PDE solves. From this computational cost analysis, we see the modified A-optimal estimators require ℓ less PDE solves than the A-optimal estimators.

3.2.2 Error analysis

We now quantify the absolute error of our estimators with the following theorem.

Theorem 3. *Let $\widehat{\Phi}_{\text{mod}}(\mathbf{w}; \ell)$ and $\widehat{\nabla \Phi}_{\text{mod}}(\mathbf{w}; \ell)$ be the randomized estimators approximating the modified A-optimal objective function $\Phi_{\text{mod}}(\mathbf{w})$ and its gradient $\nabla \Phi_{\text{mod}}(\mathbf{w})$, respectively. Using the notation and assumptions of Theorem 2, for fixed $\mathbf{w} \in [0, 1]^{n_s}$*

$$\mathbb{E} \left[|\Phi_{\text{mod}}(\mathbf{w}) - \widehat{\Phi}_{\text{mod}}(\mathbf{w}; \ell)| \right] \leq \text{tr}(f(\mathbf{\Lambda}_2)) + \text{tr}\left(f(\gamma_k^{2q-1} C \mathbf{\Lambda}_2)\right),$$

and for $j = 1, \dots, n_s$,

$$\mathbb{E} \left[\left| \partial_j \Phi_{mod}(\mathbf{w}) - \widehat{\partial_j \Phi_{mod}(\mathbf{w}; \ell)} \right| \right] \leq 2 \|\mathbf{P}_j\|_2 \left(\text{tr} (f(\mathbf{\Lambda}_2)) + \text{tr} \left(f(\gamma_k^{2q-1} C \mathbf{\Lambda}_2) \right) \right),$$

where C is defined in (3.6).

Proof. See Appendix Section 3.4.3. □

Notice the bounds presented in Theorem 2 and 3 differ by a factor of $\|\mathbf{Z}\|_2$. Since the modified A-optimal criterion removes one application of $\mathbf{\Gamma}_{\text{prior}}$ from the computation of the A-optimal criterion, the bounds related to the modified A-optimal criterion no longer have the factor of $\|\mathbf{Z}\|_2$.

3.3 Numerical results

In this section, we present numerical results that test various aspects of the proposed methods. We begin by a brief description of the inverse advection-diffusion problem used to illustrate the proposed OED methods, in Section 3.3.1. The setup of our model problem is adapted from that in [5], where further details about the forward and inverse problem can be found. In Section 3.3.1, we also describe the numerical methods used for solving the forward problem, as well as the optimization solver for the OED problem. In Section 3.3.2, we test the accuracy of our randomized estimators and illustrate our error bounds.

3.3.1 Model problem and solvers

We consider a two-dimensional time-dependent advection-diffusion equation

$$\begin{aligned} u_t - \kappa \Delta u + \mathbf{v} \cdot \nabla u &= 0 && \text{in } \mathcal{D} \times (0, T), \\ u(\cdot, 0) &= m && \text{in } \mathcal{D}, \\ \kappa \nabla u \cdot \mathbf{n} &= 0 && \text{on } \partial \mathcal{D} \times (0, T), \end{aligned}$$

which models the transport of contaminants (e.g., in the atmosphere or the subsurface). Here κ is the diffusion coefficient and is taken to be $\kappa = 0.01$. The velocity field \mathbf{v} is computed by solving a steady-state Navier-Stokes equations, as in [5]. The domain \mathcal{D} , depicted in Figure 3.1, is the unit square in \mathbb{R}^2 with the gray rectangles, modeling obstacles/buildings, removed. The boundary $\partial \mathcal{D}$ is the union of the outer boundary and the boundaries of the obstacles. The PDE is discretized using linear triangular continuous Galerkin finite elements in space and implicit Euler in time. We let the final simulation time be $T = 5$.

The inverse problem involves reconstructing the initial state m from space-time point measurements of $u(\mathbf{x}, t)$. We consider $n_s = 109$ sensor candidate locations distributed throughout the domain which are indicated by hollow squares in Figure 3.1. Measurement data is collected from a subset of these locations at three observation times, $t = 1, 2$, and 3.5 . To simulate noisy

observations, 2% noise is added to the simulated data. We fix the random matrix $\mathbf{\Omega}$ in Algorithm 1 for the numerical experiments performed in this chapter.

Recall, in Section 2.2, we define our prior covariance operator to be a Laplacian-like operator. Following [5], an application of the square root of the prior on $s \in L^2(\mathcal{D})$ is $v = \mathcal{A}^{-1}s$, which satisfies the following weak form

$$\int_{\mathcal{D}} \theta \nabla v \cdot \nabla p + \alpha v p \, d\mathbf{x} = \int_{\mathcal{D}} s p \, d\mathbf{x} \quad \text{for every } p \in H^1(\mathcal{D}). \quad (3.14)$$

Here θ and α control the variance and correlation length and are chosen to be $\alpha = 0.1$ and $\theta = 0.002$, respectively.

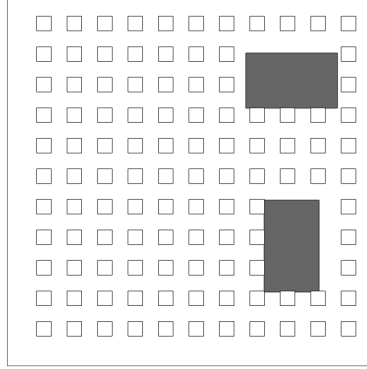


Figure 3.1 Domain with 109 candidate sensor locations.

3.3.2 Accuracy of estimators

Here we examine the accuracy of the randomized estimators with respect to ℓ , the number of columns in the sampling matrix $\mathbf{\Omega}$ in the randomized subspace iteration algorithm. Specifically, we compute

$$\begin{aligned} e_1(\ell) &= \frac{|\Phi_{\text{aopt}}(\mathbf{w}) - \widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)|}{|\Phi_{\text{aopt}}(\mathbf{w})|}, & e_2(\ell) &= \frac{\|\nabla \Phi_{\text{aopt}}(\mathbf{w}) - \widehat{\nabla \Phi}_{\text{aopt}}(\mathbf{w}; \ell)\|_2}{\|\nabla \Phi_{\text{aopt}}(\mathbf{w})\|_2}, \\ e_3(\ell) &= \frac{|\Phi_{\text{mod}}(\mathbf{w}) - \widehat{\Phi}_{\text{mod}}(\mathbf{w}; \ell)|}{|\Phi_{\text{mod}}(\mathbf{w})|}, & e_4(\ell) &= \frac{\|\nabla \Phi_{\text{mod}}(\mathbf{w}) - \widehat{\nabla \Phi}_{\text{mod}}(\mathbf{w}; \ell)\|_2}{\|\nabla \Phi_{\text{mod}}(\mathbf{w})\|_2}, \end{aligned}$$

with \mathbf{w} taken to be a vector of all ones; that is, with all sensors activated. We let ℓ to vary from 17 to 327, because the rank of $\mathbf{H}_m(\mathbf{w})$ is no larger than the number of observations taken, $n_s n_t = 327$. Figure 3.2 illustrates the relative error in the estimators for the A-optimal criterion

and its gradient (left) and the modified A-optimal criterion and its gradient (right), as ℓ is varied. We observe that the error decreases rapidly with increasing ℓ . This illustrates accuracy and efficiency of our estimators.

Next, we consider the absolute error in the estimators for the objective function and compare them with the theoretical bounds derived in Theorems 2 and 3. In Figure 3.3, we compare the absolute error in the estimators with bounds from Theorem 2 and Theorem 3. As before, we take $\mathbf{w} = [1, 1, \dots, 1]^\top \in \mathbb{R}^{n_s}$. We observe that our error bound captures the general trend in the error. Moreover, the error bound for the modified A-optimality is better since it does not have the additional factor of $\|\mathbf{Z}\|_2$.

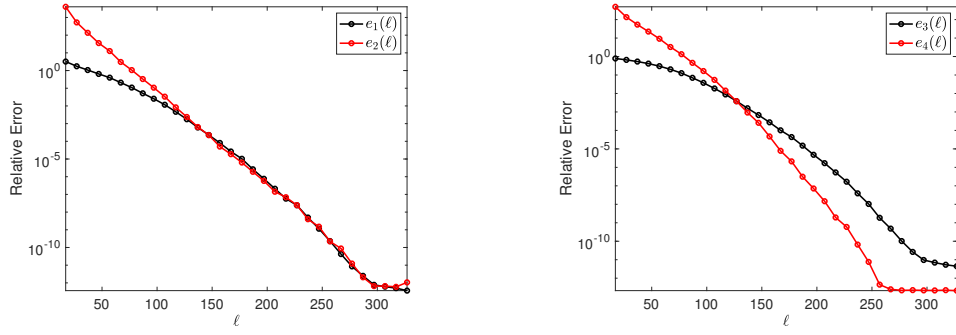


Figure 3.2 Relative error of the randomized estimators for the A-optimal criterion (left), and those corresponding to the modified A-optimal criterion (right) for varying ℓ .

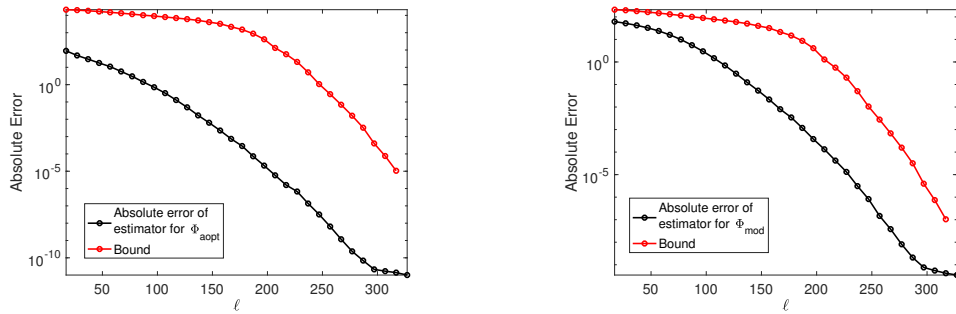


Figure 3.3 Absolute error bound for $\hat{\Phi}_{\text{aopt}}(\mathbf{w})$ from Theorem 2 (left), and $\hat{\Phi}_{\text{mod}}(\mathbf{w})$ from Theorem 3 (right) for varying ℓ .

3.4 Proofs of bounds

3.4.1 Trace of matrix function

In the proofs below, we use the Loewner partial ordering [47, Chapter 7.7]; we briefly recapitulate some main results that will be useful in our proof. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ be symmetric positive definite; then $\mathbf{A} \preceq \mathbf{B}$ means that $\mathbf{B} - \mathbf{A}$ is positive semidefinite. For any $\mathbf{S} \in \mathbb{R}^{n \times m}$, it also follows that $\mathbf{S}^\top \mathbf{A} \mathbf{S} \preceq \mathbf{S}^\top \mathbf{B} \mathbf{S}$. Let $\mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$ be the eigendecomposition of \mathbf{A} . Then, $f(\mathbf{A}) = \mathbf{U} f(\mathbf{\Lambda}) \mathbf{U}^\top$ and $\text{tr}(f(\mathbf{A})) = \sum_{i=1}^n f(\lambda_i)$. If f is monotonically increasing then $\text{tr}(f(\mathbf{A})) \leq \text{tr}(f(\mathbf{B}))$ since $\mathbf{A} \preceq \mathbf{B}$ implies $\lambda_i(\mathbf{A}) \leq \lambda_i(\mathbf{B})$ for $i = 1, \dots, n$.

The following bound allows us to bound the trace of a matrix function in terms of its diagonal subblocks.

Lemma 3.4.0.1. *Let*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^\top & \mathbf{A}_{22} \end{bmatrix}$$

be a symmetric positive definite matrix. Let f be a nonnegative concave function on $[0, \infty)$. Then

$$\text{tr}(f(\mathbf{A})) \leq \text{tr}(f(\mathbf{A}_{11})) + \text{tr}(f(\mathbf{A}_{22})).$$

Proof. See Theorem 2.1 and Remark 2.4 in [64]. □

We are ready to state and prove our main result of this section, which is the key in proving Theorem 2 and Theorem 3.

Theorem 4. *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix with eigendecomposition*

$$\mathbf{A} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1 & \\ & \mathbf{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^\top \\ \mathbf{U}_2^\top \end{bmatrix},$$

where $\mathbf{\Lambda}_1 = \text{diag}(\lambda_1, \dots, \lambda_k)$ and $\mathbf{\Lambda}_2 = \text{diag}(\lambda_{k+1}, \dots, \lambda_n)$ contain the eigenvalues arranged in descending order. Assume that the eigenvalue ratio $\gamma_k \equiv \frac{\lambda_{k+1}}{\lambda_k} < 1$. Let k be the target rank, $p \geq 2$ be the oversampling parameter such that $\ell \equiv k + p \leq n$, and let $q \geq 1$ be the number of subspace iterations. Furthermore, assume that $\mathbf{Q} \in \mathbb{R}^{n \times \ell}$ and $\mathbf{T} \in \mathbb{R}^{\ell \times \ell}$ are computed using Algorithm 1 and define $\hat{\mathbf{A}} \equiv \mathbf{Q} \mathbf{T} \mathbf{Q}^\top$. Then

$$0 \leq \mathbb{E} \left[\text{tr} \left((\mathbf{I} + \hat{\mathbf{A}})^{-1} \right) - \text{tr} \left((\mathbf{I} + \mathbf{A})^{-1} \right) \right] \leq \text{tr}(f(\mathbf{\Lambda}_2)) + \text{tr} \left(f(\gamma_k^{2q-1} C \mathbf{\Lambda}_2) \right),$$

where $f = x/(1+x)$, and the constant C is defined in (3.6).

Proof. Suppose $\text{rank}(\mathbf{A}) = r$. Then, \mathbf{A} has at most r nonzero eigenvalues, and thus, we can

define $\mathbf{\Lambda}_{r-k} = \text{diag}(\lambda_{k+1}, \dots, \lambda_r)$ so that

$$\mathbf{\Lambda}_2 = \begin{bmatrix} \mathbf{\Lambda}_{r-k} & \\ & \mathbf{0}_{n-r-k} \end{bmatrix}. \quad (3.15)$$

We split this proof into several steps.

Step 0: Lower bound

Let $\tilde{\lambda}_1 \geq \dots \geq \tilde{\lambda}_\ell \geq 0$ be the eigenvalues of \mathbf{T} (and also $\widehat{\mathbf{A}}$). By the Cauchy interlacing theorem (see [89, Lemma 1] for the specific version of the argument), $\lambda_i \geq \tilde{\lambda}_i$ for $i = 1, \dots, \ell$. Using properties of the trace operator

$$\begin{aligned} \text{tr} \left((\mathbf{I} + \widehat{\mathbf{A}})^{-1} \right) - \text{tr} \left((\mathbf{I} + \mathbf{A})^{-1} \right) &= \sum_{i=1}^{\ell} \frac{1}{1 + \tilde{\lambda}_i} + (n - \ell) - \sum_{i=1}^n \frac{1}{1 + \lambda_i} \\ &= \sum_{i=1}^{\ell} \frac{\lambda_i - \tilde{\lambda}_i}{(1 + \lambda_i)(1 + \tilde{\lambda}_i)} + \sum_{i=\ell+1}^n \frac{\lambda_i}{1 + \lambda_i}. \end{aligned}$$

Since each term in the summation is nonnegative, the lower bound follows.

Step 1. Trace of matrix function

We first write $\widehat{\mathbf{A}} = \mathbf{Q}\mathbf{T}\mathbf{Q}^\top = \mathbf{Q}\mathbf{Q}^\top \mathbf{A}\mathbf{Q}\mathbf{Q}^\top = \mathbf{P}_\mathbf{Q}\mathbf{A}\mathbf{P}_\mathbf{Q}$, where $\mathbf{P}_\mathbf{Q} = \mathbf{Q}\mathbf{Q}^\top$ is an orthogonal projection matrix onto the range of \mathbf{Q} . Since $\widehat{\mathbf{A}}$ has the same eigenvalues as $\mathbf{R} \equiv \mathbf{A}^{1/2}\mathbf{P}_\mathbf{Q}\mathbf{A}^{1/2}$ [47, Theorem 1.3.22],

$$\text{tr} \left((\mathbf{I} + \widehat{\mathbf{A}})^{-1} \right) = \text{tr} \left((\mathbf{I} + \mathbf{A}^{1/2}\mathbf{P}_\mathbf{Q}\mathbf{A}^{1/2})^{-1} \right). \quad (3.16)$$

Also, since $\mathbf{P}_\mathbf{Q} \preceq \mathbf{I}$, it follows that $\mathbf{R} = \mathbf{A}^{1/2}\mathbf{P}_\mathbf{Q}\mathbf{A}^{1/2} \preceq \mathbf{A}$ and $\mathbf{0} \preceq \mathbf{A} - \mathbf{R}$. Therefore, from the proof of [12, Lemma X.1.4] and (3.16),

$$\text{tr} \left((\mathbf{I} + \widehat{\mathbf{A}})^{-1} \right) - \text{tr} \left((\mathbf{I} + \mathbf{A})^{-1} \right) \leq \text{tr} \left((\mathbf{I} - (\mathbf{I} + \mathbf{A} - \mathbf{R})^{-1}) \right) = \text{tr} (f(\mathbf{A} - \mathbf{R})),$$

where $f(x)$ was defined in the statement of the theorem.

Step 2. Reducing the dimensionality

Let $\mathbf{F}_\mathbf{S} \equiv \mathbf{\Lambda}_2^q \mathbf{\Omega}_2 \mathbf{\Omega}_1^\dagger \mathbf{\Lambda}_1^{-q}$. In Algorithm 1, we compute $\mathbf{Y} = \mathbf{A}^q \mathbf{\Omega}$ and let $\mathbf{Y} = \mathbf{Q}\mathbf{R}_\mathbf{Y}$ be the thin QR factorization of \mathbf{Y} . Let $\mathbf{W}_\mathbf{Q} = \mathbf{R}_\mathbf{Y} \mathbf{\Omega}_1^\dagger \mathbf{\Lambda}_1^{-q} (\mathbf{I} + \mathbf{F}_\mathbf{S}^\top \mathbf{F}_\mathbf{S})^{-1/2} \in \mathbb{R}^{\ell \times k}$ be defined as in the proof of [89, Theorem 6]. It was also shown that $\mathbf{W}_\mathbf{Q}$ has orthonormal columns, so that

$$\mathbf{Q}\mathbf{W}_\mathbf{Q} = \mathbf{U} \begin{bmatrix} \mathbf{I} \\ \mathbf{F}_\mathbf{S} \end{bmatrix} (\mathbf{I} + \mathbf{F}_\mathbf{S}^\top \mathbf{F}_\mathbf{S})^{-1/2} \in \mathbb{R}^{n \times k},$$

has orthonormal columns. The following sequence of identities also hold: $\mathbf{W}_\mathbf{Q}\mathbf{W}_\mathbf{Q}^\top \preceq \mathbf{I}$,

$$\mathbf{Q}\mathbf{W}_\mathbf{Q}\mathbf{W}_\mathbf{Q}^\top\mathbf{Q}^\top \preceq \mathbf{Q}\mathbf{Q}^\top,$$

and

$$\mathbf{A} - \mathbf{R} \preceq \mathbf{A} - \mathbf{A}^{1/2}\mathbf{Q}\mathbf{W}_\mathbf{Q}\mathbf{W}_\mathbf{Q}^\top\mathbf{Q}^\top\mathbf{A}^{1/2} \equiv \mathbf{S}.$$

Since $f(x)$ is a monotonic increasing function, $\text{tr}(f(\mathbf{A} - \mathbf{R})) \leq \text{tr}(f(\mathbf{S}))$.

Step 3. Split into the diagonal blocks

We can rewrite \mathbf{S} as

$$\mathbf{S} = \mathbf{U} \begin{bmatrix} \mathbf{S}_1 & * \\ * & \mathbf{S}_2 \end{bmatrix} \mathbf{U}^\top,$$

where $*$ represents blocks that can be ignored and

$$\mathbf{S}_1 \equiv \mathbf{\Lambda}_1^{1/2}(\mathbf{I} - (\mathbf{I} + \mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S})^{-1})\mathbf{\Lambda}_1^{1/2}, \quad \mathbf{S}_2 = \mathbf{\Lambda}_2^{1/2}(\mathbf{I} - \mathbf{F}_\mathbf{S}(\mathbf{I} + \mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S})^{-1}\mathbf{F}_\mathbf{S}^\top)\mathbf{\Lambda}_2^{1/2}.$$

We can invoke Lemma 3.4.0.1, since $f(x) = x/(1+x)$ is concave and nonnegative on $[0, \infty)$. Therefore, we have

$$\text{tr}(f(\mathbf{S})) \leq \text{tr}(f(\mathbf{S}_1)) + \text{tr}(f(\mathbf{S}_2)).$$

Note that the matrix \mathbf{U} disappears, because the trace is unitarily invariant.

Step 4. Completing the structural bound

Using an SVD based argument it can be shown that $\mathbf{I} - (\mathbf{I} + \mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S})^{-1} \preceq \mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S}$, so that

$$\mathbf{S}_1 = \mathbf{\Lambda}_1^{1/2}(\mathbf{I} - (\mathbf{I} + \mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S})^{-1})\mathbf{\Lambda}_1^{1/2} \preceq \mathbf{\Lambda}_1^{1/2}\mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S}\mathbf{\Lambda}_1^{1/2}.$$

Therefore, since f is monotonically increasing

$$\text{tr}(f(\mathbf{S}_1)) \leq \text{tr}\left(f(\mathbf{\Lambda}_1^{1/2}\mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S}\mathbf{\Lambda}_1^{1/2})\right) = \sum_{j=1}^k f\left(\lambda_j\left[\mathbf{\Lambda}_1^{1/2}\mathbf{F}_\mathbf{S}^\top\mathbf{F}_\mathbf{S}\mathbf{\Lambda}_1^{1/2}\right]\right). \quad (3.17)$$

Note that $\mathbf{F}_\mathbf{S}\mathbf{\Lambda}_1^{1/2}$ is $(n-k) \times k$ has at most $\min\{n-k, k\}$ nonzero singular values. Looking more into the structure of $\mathbf{F}_\mathbf{S}\mathbf{\Lambda}_1^{1/2}$, and using (3.15), we can write

$$\begin{aligned} \mathbf{F}_\mathbf{S}\mathbf{\Lambda}_1^{1/2} &= \mathbf{\Lambda}_2^{1/2} \begin{bmatrix} \mathbf{\Lambda}_{r-k}^{q-1/2} & \\ & \mathbf{0} \end{bmatrix} \mathbf{\Omega}_2\mathbf{\Omega}_1^\dagger\mathbf{\Lambda}_1^{-q+1/2} \\ &= \mathbf{\Lambda}_2^{1/2} \begin{bmatrix} \mathbf{\Lambda}_{r-k}^{q-1/2} \widehat{\mathbf{\Omega}}_2\mathbf{\Omega}_1^\dagger\mathbf{\Lambda}_1^{-q+1/2} & \\ & \mathbf{0} \end{bmatrix}, \end{aligned}$$

where $\widehat{\mathbf{\Omega}}_2 \in \mathbb{R}^{(r-k) \times (k+p)}$ such that $\mathbf{\Omega}_2 = \begin{bmatrix} \widehat{\mathbf{\Omega}}_2 \\ * \end{bmatrix}$. Using the multiplicative singular value inequalities [47, Equation (7.3.14)] and repeated use of the submultiplicative inequality gives

$$\sigma_j(\mathbf{F}_S \mathbf{\Lambda}_1^{1/2}) \leq \gamma_k^{q-1/2} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2 \sigma_j(\mathbf{\Lambda}_2^{1/2}), \quad j = 1, \dots, \min\{k, n-k\}. \quad (3.18)$$

The analysis splits into two cases:

Case 1: $k \leq n - k$. Since f is monotonically increasing, using (3.17) and (3.18)

$$\begin{aligned} \text{tr}(f(\mathbf{S}_1)) &\leq \sum_{j=1}^k f\left(\lambda_j \left[\mathbf{\Lambda}_1^{1/2} \mathbf{F}_S^\top \mathbf{F}_S \mathbf{\Lambda}_1^{1/2}\right]\right) \leq \sum_{j=1}^k f\left(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \sigma_j^2(\mathbf{\Lambda}_2^{1/2})\right) \\ &\leq \sum_{j=1}^{n-k} f\left(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \sigma_j^2(\mathbf{\Lambda}_2^{1/2})\right) = \text{tr}\left(f(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \mathbf{\Lambda}_2)\right). \end{aligned}$$

Case 2: $k > n - k$. Since $\mathbf{\Lambda}_1^{1/2} \mathbf{F}_S^\top \mathbf{F}_S \mathbf{\Lambda}_1^{1/2}$ has at most $n - k$ nonzero eigenvalues, use the fact that $f(0) = 0$, along with (3.17) and (3.18) to obtain

$$\begin{aligned} \text{tr}(f(\mathbf{S}_1)) &\leq \sum_{j=1}^k f\left(\lambda_j \left[\mathbf{\Lambda}_1^{1/2} \mathbf{F}_S^\top \mathbf{F}_S \mathbf{\Lambda}_1^{1/2}\right]\right) = \sum_{j=1}^{n-k} f\left(\lambda_j \left[\mathbf{\Lambda}_1^{1/2} \mathbf{F}_S^\top \mathbf{F}_S \mathbf{\Lambda}_1^{1/2}\right]\right) \\ &\leq \sum_{j=1}^{n-k} f\left(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \sigma_j^2(\mathbf{\Lambda}_2^{1/2})\right) = \text{tr}\left(f(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \mathbf{\Lambda}_2)\right). \end{aligned}$$

To summarize, in both cases $\text{tr}(f(\mathbf{S}_1)) \leq \text{tr}\left(f(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \mathbf{\Lambda}_2)\right)$. Similarly, since $\mathbf{0} \preceq \mathbf{F}_S(\mathbf{I} + \mathbf{F}_S^\top \mathbf{F}_S)^{-1} \mathbf{F}_S^\top$, we can show

$$\mathbf{S}_2 = \mathbf{\Lambda}_2^{1/2} (\mathbf{I} - \mathbf{F}_S (\mathbf{I} + \mathbf{F}_S^\top \mathbf{F}_S)^{-1} \mathbf{F}_S^\top) \mathbf{\Lambda}_2^{1/2} \preceq \mathbf{\Lambda}_2,$$

so that $\text{tr}(f(\mathbf{S}_2)) \leq \text{tr}(f(\mathbf{\Lambda}_2))$. Combine with step 3 to obtain

$$\text{tr}(f(\mathbf{S})) \leq \text{tr}(f(\mathbf{\Lambda}_2)) + \text{tr}\left(f(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \mathbf{\Lambda}_2)\right).$$

Combine this with the results of steps 1 and 2, to obtain the structural bound

$$\text{tr}\left((\mathbf{I} + \widehat{\mathbf{A}})^{-1}\right) - \text{tr}\left((\mathbf{I} + \mathbf{A})^{-1}\right) \leq \text{tr}\left(f(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \mathbf{\Lambda}_2)\right) + \text{tr}(f(\mathbf{\Lambda}_2)).$$

Step 5. The expectation bound

Note that $\widehat{\mathbf{\Omega}}_2 \in \mathbb{R}^{(r-k) \times (k+p)}$ and $\mathbf{\Omega}_1 \in \mathbb{R}^{k \times (k+p)}$. From the proof of [89, Theorem 1], we have $\mathbb{E}[\|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2] \leq C$, where C was defined in (3.6). By Jensen's inequality, using the fact that

$f(x) = x/(1+x)$ is concave on $[0, \infty)$ we have

$$\begin{aligned} \mathbb{E} \left[\text{tr} \left((\mathbf{I} + \widehat{\mathbf{A}})^{-1} \right) - \text{tr} \left((\mathbf{I} + \mathbf{A})^{-1} \right) \right] &\leq \text{tr} (f(\mathbf{\Lambda}_2)) + \mathbb{E} \left[\text{tr} \left(f(\gamma_k^{2q-1} \|\widehat{\mathbf{\Omega}}_2 \mathbf{\Omega}_1^\dagger\|_2^2 \mathbf{\Lambda}_2) \right) \right] \\ &\leq \text{tr} (f(\mathbf{\Lambda}_2)) + \text{tr} \left(f(\gamma_k^{2q-1} C \mathbf{\Lambda}_2) \right). \end{aligned}$$

Combining this with the lower bound (step 0) completes the proof. \square

3.4.2 Proof of Theorem 2

For the remaining discussion, we use the notation

$$\mathbf{P}_j = \mathcal{F}^\top \mathbf{E}_j^{\text{noise}} \mathcal{F}, \quad (3.19)$$

where \mathcal{F} and $\mathbf{E}_j^{\text{noise}}$ are defined in (2.5) and (2.7), respectively. We will also need

Lemma 3.4.0.2 (See [6]). *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ and let \mathbf{B} be a symmetric positive semidefinite matrix. Then, we have $|\text{tr}(\mathbf{AB})| \leq \|\mathbf{A}\|_2 \text{tr}(\mathbf{B})$.*

Proof. Recall our estimator $\widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)$ from (3.1). For fixed ℓ , using Lemma 3.4.0.2 we have

$$\begin{aligned} \mathbb{E} |\Phi_{\text{aopt}}(\mathbf{w}) - \widehat{\Phi}_{\text{aopt}}(\mathbf{w}; \ell)| &= \mathbb{E} \left| \text{tr} \left((\mathbf{I} + \mathbf{H}_m(\mathbf{w}))^{-1} \mathbf{Z} - (\mathbf{I} + \widehat{\mathbf{H}}_m(\mathbf{w}))^{-1} \mathbf{Z} \right) \right| \\ &\leq \|\mathbf{Z}\|_2 \mathbb{E} |\text{tr} \left((\mathbf{I} + \mathbf{H}_m(\mathbf{w}))^{-1} - (\mathbf{I} + \widehat{\mathbf{H}}_m(\mathbf{w}))^{-1} \right)|. \end{aligned}$$

Applying Theorem 4 establishes (3.7).

Next, we consider (3.8). Recall the estimator $\widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}; \ell)$ from (3.2). We can write the absolute error as

$$\begin{aligned} |\partial_j \Phi_{\text{aopt}}(\mathbf{w}) - \widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}; \ell)| &= |\text{tr} \left(((\mathbf{I} + \mathbf{H}_m)^{-1} \mathbf{P}_j (\mathbf{I} + \mathbf{H}_m)^{-1} - (\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1} \mathbf{P}_j (\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1}) \mathbf{Z} \right)|, \end{aligned}$$

where $\widehat{\mathbf{H}}_m = \mathbf{Q} \mathbf{T} \mathbf{Q}^\top$. We use the decomposition

$$\begin{aligned} ((\mathbf{I} + \mathbf{H}_m)^{-1} \mathbf{P}_j (\mathbf{I} + \mathbf{H}_m)^{-1} - (\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1} \mathbf{P}_j (\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1}) \mathbf{Z} &= (\mathbf{D} \mathbf{P}_j (\mathbf{I} + \mathbf{H}_m)^{-1} + (\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1} \mathbf{P}_j \mathbf{D}) \mathbf{Z}, \end{aligned}$$

where $\mathbf{D} \equiv (\mathbf{I} + \mathbf{H}_m)^{-1} - (\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1}$. Repeated application of Lemma 3.4.0.2 gives

$$|\partial_j \Phi_{\text{aopt}}(\mathbf{w}) - \widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}; \ell)| \leq \|\mathbf{P}_j\|_2 \|\mathbf{Z}\|_2 \left(\|(\mathbf{I} + \mathbf{H}_m)^{-1}\|_2 + \|(\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1}\|_2 \right) |\text{tr}(\mathbf{D})|.$$

Since $\mathbf{I} + \mathbf{H}_m$ and $\mathbf{I} + \widehat{\mathbf{H}}_m$ have eigenvalues greater than or equal to one, $\|(\mathbf{I} + \mathbf{H}_m)^{-1}\|_2 + \|(\mathbf{I} + \widehat{\mathbf{H}}_m)^{-1}\|_2 \leq 2$. Finally, taking the expectation and applying Theorem 4, we have the desired

result. □

3.4.3 Proof of Theorem 3

Proof. The proof follows in similar lines as the proof of Theorem 2 except the fact that in Theorem 3 we do not have \mathbf{Z} in the expressions. □

3.5 Conclusion

We have established an efficient and flexible computational framework for A-optimal design of experiments in large-scale Bayesian linear inverse problems. The proposed randomized estimators for the OED objective and its gradient are accurate, efficient, simple to implement and parallelize. Specifically, the randomized estimators exploit the low-rank structure in the inverse problem; namely, the low-rank structure of the prior-preconditioned data misfit Hessian—a common feature of ill-posed inverse problems. We also presented the modified A-optimal criterion, which is more computationally efficient to compute and can provide designs that, while sub-optimal if the goal is to compute A-optimal designs, provide a systematic means for obtaining sensor placements with small posterior uncertainty levels.

Open questions that we seek to explore in our future work include adaptive determination of the target rank k within the optimization algorithm, to further reduce computational costs, while ensuring sufficiently accurate estimates of the OED objective and gradient. Another possible line of inquiry is to use different low-rank approximations, such as Nyström’s method, and extending the randomized estimators to approximate trace of matrix functions. We also seek to incorporate the randomized estimators in a suitable optimization framework for Bayesian nonlinear inverse problems, in our future work.

CHAPTER

4

AN OPTIMIZATION FRAMEWORK FOR FINDING BINARY DESIGNS

We seek A-optimal designs by solving an optimization problem of the form,

$$\min_{\mathbf{w} \in \mathcal{W}} \Phi(\mathbf{w}) + \gamma P(\mathbf{w}), \quad \mathcal{W} = [0, 1]^{n_s}, \quad (4.1)$$

where the design criterion Φ is either the A-optimal criterion $\Phi_{\text{aopt}}(\mathbf{w})$ or the modified A-optimal criterion $\Phi_{\text{mod}}(\mathbf{w})$ (see Section 3.2). In the previous chapter, we laid out an efficient framework for computing accurate approximations to the A-optimal criterion and its gradient. We now discuss the choice of the penalty term P and the algorithm for solving the optimization problem.

The choice of the penalty term must satisfy two conditions: enforcing sparsity, measured by the number of nonzeros of the design vector, and binary designs, i.e., design vectors whose entries are either 1 or 0. One possibility for the penalty function is the ℓ_0 -“norm”, $P_{\ell_0}(\mathbf{w}) = \|\mathbf{w}\|_0$, which measures the number of nonzero entries in the design. However, the resulting optimization problem is challenging to solve due to its combinatorial complexity. A common practice is to replace the ℓ_0 -“norm” penalty by the ℓ_1 -norm, $P_{\ell_1}(\mathbf{w}) = \|\mathbf{w}\|_1$. The penalty function P_{ℓ_1} has desirable features: it is a convex penalty function that promotes sparsity of the optimal design vector \mathbf{w} . However, the resulting design is sparse but not necessarily binary and additional post-processing in the form of thresholding is necessary to enforce binary designs.

In what follows, we introduce a suitable penalty function that enforces both sparsity and binary designs and an algorithm for solving the OED optimization problem based on the MM approach. The resulting algorithm takes the form of a reweighted ℓ_1 -minimization algorithm [18].

The chapter is laid out as follows. In Section 4.1, we present our novel penalty function we use to solve (4.1). Section 4.2 provides a discussion of the MM approach and how this approach transforms the non-convex optimization problem into a sequence of convex ones. This algorithm is tested on our source identification problem from the previous chapter using the randomized algorithms of Chapter 3, where the numerical results are presented in Section 4.3. The last section of this chapter, Section 4.4 summarizes our findings.

4.1 Penalty functions

We propose the following penalty function

$$P_\epsilon(\mathbf{w}) = \sum_{i=1}^{n_s} \frac{|w_i|}{|w_i| + \epsilon}, \quad \mathbf{w} \in \mathbb{R}^{n_s}, \quad (4.2)$$

for a user-defined parameter $\epsilon > 0$. This penalty function approximates P_{ℓ_0} for small values of ϵ ; however, as ϵ becomes smaller the corresponding optimization problem becomes harder. To illustrate the choice of penalty functions, in Figure 4.1, we plot $P_{0.05}$ along with P_{ℓ_0} and P_{ℓ_1} , with $n_s = 1$. Using P_ϵ in the OED problem leads to the optimization problem,

$$\min_{\mathbf{w} \in \mathcal{W}} \Phi(\mathbf{w}) + \gamma P_\epsilon(\mathbf{w}). \quad (4.3)$$

In (4.3), the absolute values in definition of P_ϵ can be dropped since we limit the search for optimal solutions in \mathcal{W} . Since $P_\epsilon(\mathbf{w})$ is concave, (4.3) is a non-convex optimization problem. To tackle this, we adopt the majorization-minimization (MM) approach.

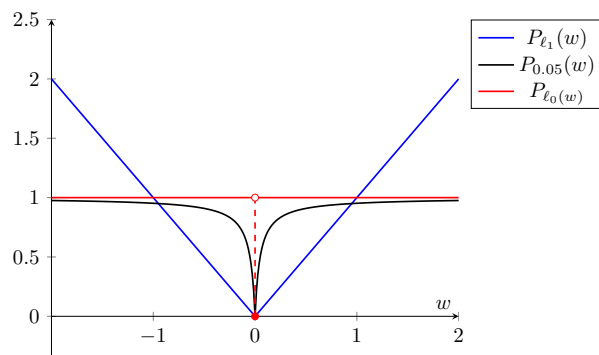


Figure 4.1 Different choices of penalty functions with $n_s = 1$.

4.2 MM approach and reweighted ℓ_1 algorithm

The idea behind the MM approach is to solve a sequence of optimization problems whose solutions converge to that of the original problem [49, 61]. This sequence is generated by a carefully constructed surrogate that satisfies two properties—the surrogate must majorize the objective function for all values, and the surrogate must match the objective function at the current iterate. More specifically, suppose

$$J(\mathbf{w}) = \Phi(\mathbf{w}) + \gamma \sum_{i=1}^{n_s} \frac{w_i}{w_i + \epsilon}.$$

Then the surrogate function $g(\mathbf{w}|\mathbf{w}^{(m)})$ at the current iterate $\mathbf{w}^{(m)}$ must satisfy

$$\begin{aligned} g(\mathbf{w}|\mathbf{w}^{(m)}) &\geq J(\mathbf{w}) \quad \forall \mathbf{w} \in \mathcal{W}, \\ g(\mathbf{w}^{(m)}|\mathbf{w}^{(m)}) &= J(\mathbf{w}^{(m)}). \end{aligned}$$

Granted the existence of this surrogate function, to find the next iterate $\mathbf{w}^{(m+1)}$ we solve the optimization problem

$$\mathbf{w}^{(m+1)} = \arg \min_{\mathbf{w} \in \mathcal{W}} g(\mathbf{w}|\mathbf{w}^{(m)}). \quad (4.4)$$

To show the objective function decreases at the next iterate, observe that the next iterate $\mathbf{w}^{(m+1)}$ stays within the feasible region and use the two properties of the surrogate function

$$J(\mathbf{w}^{(m+1)}) \leq g(\mathbf{w}^{(m+1)}|\mathbf{w}^{(m)}) \leq g(\mathbf{w}^{(m)}|\mathbf{w}^{(m)}) = J(\mathbf{w}^{(m)}).$$

To construct this surrogate function, we use the fact that a concave function is below its tangent [61, Equation (4.7)]. Applying this to our concave penalty $P_\epsilon(\mathbf{w})$, we have

$$P_\epsilon(\mathbf{w}) \leq P_\epsilon(\mathbf{w}^{(m)}) + (\mathbf{w} - \mathbf{w}^{(m)})^\top \nabla_{\mathbf{w}} P_\epsilon(\mathbf{w}^{(m)}), \quad \text{for all } \mathbf{w} \in \mathcal{W}.$$

With this majorization relation, we define the surrogate function to be

$$g(\mathbf{w}|\mathbf{w}^{(m)}) = \Phi(\mathbf{w}) + \gamma \left(P_\epsilon(\mathbf{w}^{(m)}) + (\mathbf{w} - \mathbf{w}^{(m)})^\top \nabla_{\mathbf{w}} P_\epsilon(\mathbf{w}^{(m)}) \right)$$

By dropping the terms that do not depend on \mathbf{w} , it can be readily verified that (4.4) can be replaced by the equivalent problem

$$\begin{aligned} \mathbf{w}^{(m+1)} &= \arg \min_{\mathbf{w} \in \mathcal{W}} \Phi(\mathbf{w}) + \gamma \sum_{i=1}^{n_s} \frac{\epsilon w_i}{(w_i^{(m)} + \epsilon)^2} \\ &= \arg \min_{\mathbf{w} \in \mathcal{W}} \Phi(\mathbf{w}) + \gamma \|\mathbf{R}(\mathbf{w}^{(m)})\mathbf{w}\|_1, \end{aligned} \quad (4.5)$$

where $\mathbf{R}(\mathbf{w}) = \text{diag}\left(\frac{\epsilon}{(w_1+\epsilon)^2}, \dots, \frac{\epsilon}{(w_{n_s}+\epsilon)^2}\right)$. We see that (4.5) is of the form of a reweighted ℓ_1 -optimization problem. The details of the optimization procedure are given in Algorithm 4. (We remark that, in Algorithm 4, other metrics measuring the difference between the successive weight vectors can be used in step 3.)

Algorithm 4 Reweighted ℓ_1 Algorithm.

Input: Initial guess $\mathbf{w}^{(0)} \in \mathbb{R}^{n_s}$, stopping tolerance tol, penalty parameters $\gamma, \epsilon \geq 0$.

Output: Optimal design $\mathbf{w}^* \in \mathbb{R}^{n_s}$.

1: Initialize $m = 1$.

2: Compute

$$\mathbf{w}^{(1)} = \arg \min_{\mathbf{w} \in \mathcal{W}} \Phi(\mathbf{w}) + \gamma \|\mathbf{w}\|_1.$$

3: **while** $m < m_{max}$ and $\|\mathbf{w}^{(m)} - \mathbf{w}^{(m-1)}\|_2 > \text{tol}$ **do**

4: Update $m = m + 1$

5: Compute

$$\mathbf{w}^{(m)} = \arg \min_{\mathbf{w} \in \mathcal{W}} \Phi(\mathbf{w}) + \gamma \sum_{i=1}^{n_s} r_i \cdot w_i,$$

6: Update

$$r_i = \frac{\epsilon}{(|w_i^{(m-1)}| + \epsilon)^2}, \quad i = 1, \dots, n_s.$$

7: **end while**

8: Return $\mathbf{w}^{(m)} = \mathbf{w}^*$.

We conclude this section with a few remarks regarding this algorithm. In our application, $\Phi(\mathbf{w})$ is convex; therefore, each subproblem to update the design weights is also convex. To initialize the reweighted ℓ_1 algorithm, we start with the weights $r_i = 1$, $i = 1, \dots, n_s$. This ensures that, in the first step, we are computing the solution of the ℓ_1 -penalized optimization problem. The subsequent reweighted ℓ_1 iterations further promote binary designs. To solve the subproblems in each reweighted ℓ_1 iteration we use an interior point algorithm; however, any solver for appropriate convex optimization may be used. In Section 4.3, we will provide a discussion of our choice of ϵ for our application. It is also worth mentioning that besides the penalty function P_ϵ used above, another possible choice is $\sum_{i=1}^{n_s} \arctan(|w_i|/\epsilon)$ which yields the weights $\frac{1}{|w_i|^2 + \epsilon^2}$.

4.3 Numerical results

We now consider solving (2.9) with Algorithm 3 and Algorithm 4. The optimization solver used for the OED problem is a quasi-Newton interior point method. Specifically, to solve each subproblem of Algorithm 4, we use MATLAB's interior point solver provided by the `fmincon` function; BFGS approximation to the Hessian is used for line search. We use a vector of all ones,

$\mathbf{1} \in \mathbb{R}^{n_s}$, as the initial guess for the optimization solver.

For the numerical experiments presented in this section, the random matrix $\mathbf{\Omega}$ in Algorithm 1 is fixed during the optimization process. However, because of the randomness, the accuracy of the estimators, and the optimal design thus obtained, may vary with different realizations of $\mathbf{\Omega}$. By conducting additional numerical experiments (not reported here) the stochastic nature of the estimators resulted in minor variability (at most one or two sensor locations) in the optimal experimental designs for a modest value of ℓ . On the other hand, if ℓ is sufficiently large, we observed that the same design was obtained with different realizations of $\mathbf{\Omega}$.

We now consider the choice of the value of ϵ in (4.5). The user-defined parameter ϵ controls the steepness of the penalty function at the origin; see Figure 4.1. However, we observed that if the penalty function is too steep, the optimization solvers took more iterations without substantially altering the optimal designs. We found $\epsilon = 1/2^8$ to be sufficiently small in our numerical experiments, and we keep this fixed for the remainder of the numerical experiments.

In Section 4.3.1, we investigate the performance of our proposed reweighted ℓ_1 -optimization approach. Next, we utilize the proposed optimization framework to compute A-optimal designs in Section 4.3.2. Finally, in Section 4.3.3, we compare A-optimal sensor placements with those computed by minimizing the modified A-optimal criterion.

4.3.1 Performance of the reweighted ℓ_1 algorithm

We perform two numerical experiments examining the impact of changing ℓ and the penalty parameter γ (in (2.9)). In the first experiment, we fix the penalty coefficient at an experimentally determined value of $\gamma = 3$, and vary ℓ ; the results are recorded in Table 4.1. Notice when $\ell \geq 127$, the objective function value evaluated with the optimal solution, the number of active sensors, and number of subproblem solves do not change. This suggests that the randomized estimators are sufficiently accurate with $\ell = 127$ and this yields a substantial reduction in computational cost.

Table 4.1 Number of subproblem solves (4.5), function evaluations, and active sensors for varying ℓ with the reweighted ℓ_1 algorithm and $n_s = 109$.

ℓ	subproblem solves	function count	function value	$n_{s_{\text{active}}}$
57	10	395	39.0176	38
67	9	610	44.7219	30
77	9	790	44.5484	29
87	9	756	44.1142	30
127	9	1015	44.1139	30
207	9	978	44.1139	30
307	9	970	44.1139	30

The second experiment involves varying γ , which indirectly controls the number of sensors in computed designs, with ℓ kept fixed. Here we fix $\ell = 207$, which corresponds to an accuracy on the order of 10^{-7} for the A-optimal criterion (cf. Figure 3.2). In Figure 4.2, we report the design weights sorted in descending order, as γ varies. This shows that the reweighted ℓ_1 algorithm indeed produces binary designs for a range of penalty parameters. We also notice in Figure 4.3 that as γ increases the sparsity increases (i.e., the number of active sensors ns_{active} decreases) and the number of function evaluations increases. The right panel compares the cost of solving an ℓ_1 -penalized problem for the corresponding penalty parameter γ . Since this problem is the first iterate of the reweighted ℓ_1 algorithm we see that an additional cost is required to obtain binary designs and this cost increases with increasing γ (i.e., more sparse designs).

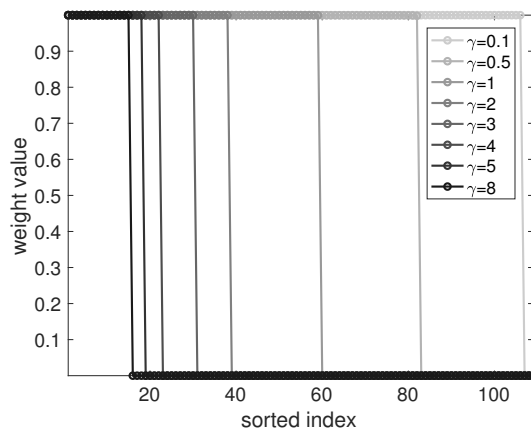


Figure 4.2 Optimal designs as a result of varying γ with the reweighted ℓ_1 algorithm for the A-optimal criterion. We set $\ell = 207$.

4.3.2 Computing optimal designs

In Figure 4.4 (left), we report an A-optimal sensor placement obtained using our optimization framework, with $\ell = 207$ and $\gamma = 5$; the resulting optimal sensor locations, with 18 active sensors, are superimposed on the posterior standard deviation field. While the design is computed to yield a minimal average variance of the posterior distribution, it is also important to consider the mean of this distribution. For completeness, in Figure 4.4 we show the “true” initial condition (middle panel), used to generate synthetic data, and the mean of the resulting posterior distribution for the 18 active sensor design (right panel). With the 18 sensor design, we also illustrate the resulting uncertainty reduction by looking at the prior and posterior standard deviation fields; see Figure 4.5.

We now compare the designs obtained using the reweighted ℓ_1 algorithm and our estimators

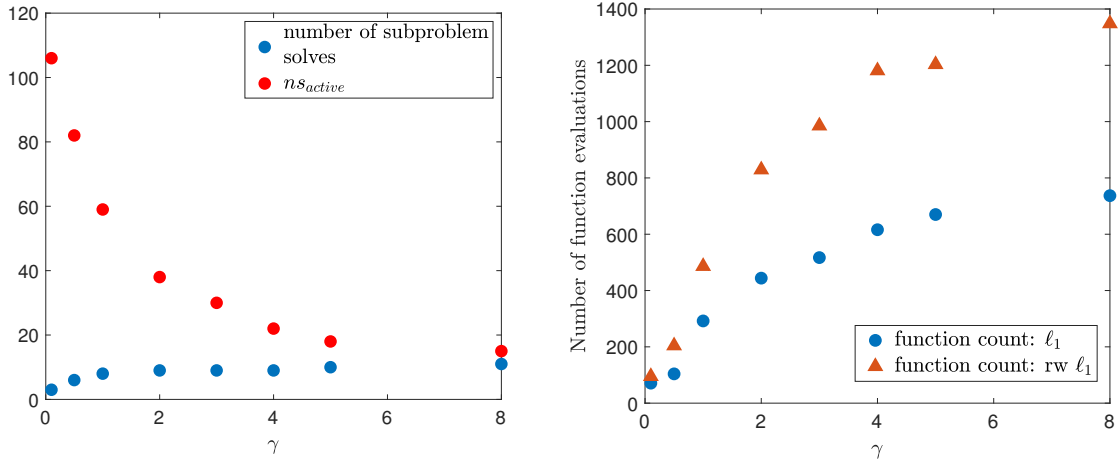


Figure 4.3 The effect of varying γ on the reweighted ℓ_1 algorithm for the A-optimal criterion. We set $\ell = 207$.

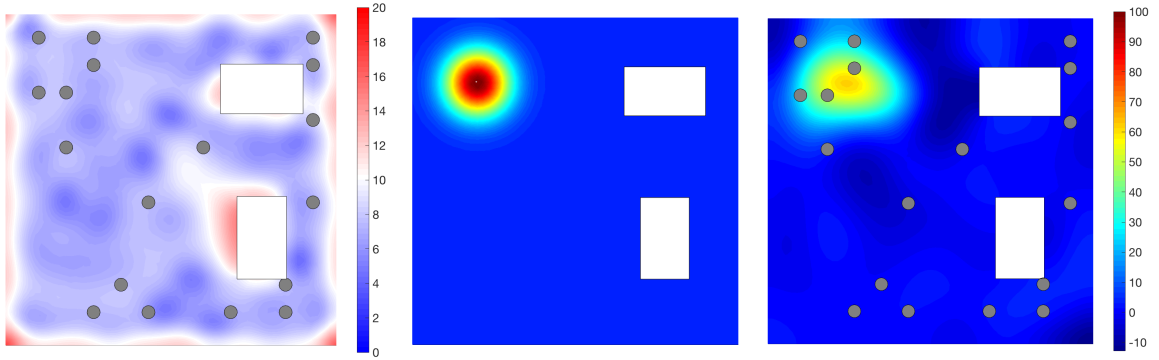


Figure 4.4 Standard deviation computed using the optimal design indicated by the gray circles (left). True initial condition (middle) and initial condition reconstruction (right). The optimal design was computed using $\ell = 207$, the reweighted ℓ_1 algorithm, and $\gamma = 5$.

against designs chosen at random, illustrating the effectiveness of the proposed A-optimal design strategy. Recall that varying γ allows us to obtain optimal designs with different numbers of active sensors. For each value of γ , we use Algorithm 3 and Algorithm 4 to compute an optimal design. We then draw 15 random designs with the same number of active sensors as the optimal design obtained using our algorithms. To enable a consistent comparison, we evaluate the exact A-optimal criterion $\Phi_{\text{aopt}}(\mathbf{w})$ at the computed optimal designs and the random ones; the results are reported in the left panel of Figure 4.6. The values corresponding to the computed optimal designs are indicated as dots on the black solid line. The values obtained from the random designs are indicated by the squares. We note that the designs computed with the reweighted ℓ_1 algorithm consistently beat the random designs, as expected. This observation is emphasized in the right panel of Figure 4.6. Here, we compared the computed optimal design (when $\gamma = 3$)

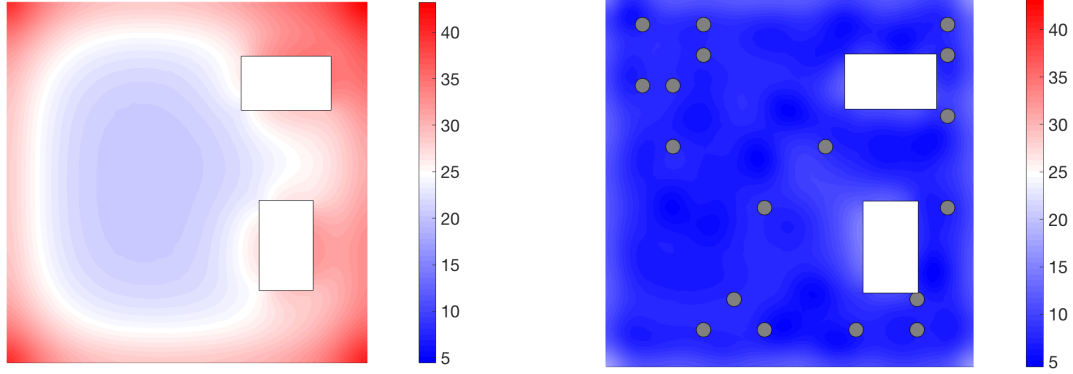


Figure 4.5 Comparison of the prior standard deviation field (left) with the posterior standard deviation field (right) computed using the optimal design indicated by the gray circles.

with 1500 randomly generated designs using the exact trace of the posterior covariance. Again, the computed design results in a lower true A-optimal value than the random designs.

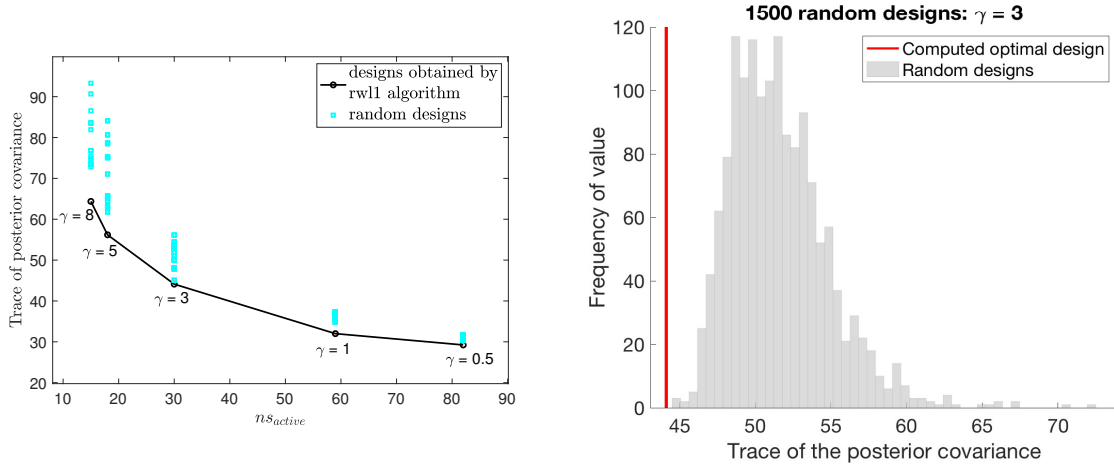


Figure 4.6 The true A-optimal criterion computed using the optimal and randomly generated designs. The optimal designs were computed using the A-optimal criterion and the reweighted ℓ_1 algorithm for different values of γ (left). Comparing the computed optimal design when $\gamma = 3$ to 1500 randomly generated designs (right).

4.3.3 Comparing A-optimal and modified A-optimal designs

Here we provide a quantitative comparison of sensor placements obtained by minimizing A-optimal and modified A-optimal criteria using our proposed algorithms. Specifically, for various

values of γ , we solve (2.9) with both the A-optimal and modified A-optimal estimators to obtain two sets of designs. By varying γ , the resulting designs obtained with the A-optimal and modified A-optimal estimators have different number of active sensors. Using both sets of designs, we evaluate the exact A-optimal criterion $\Phi_{\text{aopt}}(\mathbf{w})$; these are displayed in Figure 4.7. Observe that in all cases the computed A-optimal and modified A-optimal designs lead to similar levels of average posterior variance. This suggests that the modified A-optimal criterion could be used as a surrogate for the A-optimal criterion. Using the modified A-optimal criterion decreases the overall number of PDE solves and yields designs that result in values of the average posterior variance close to those produced by A-optimal designs.

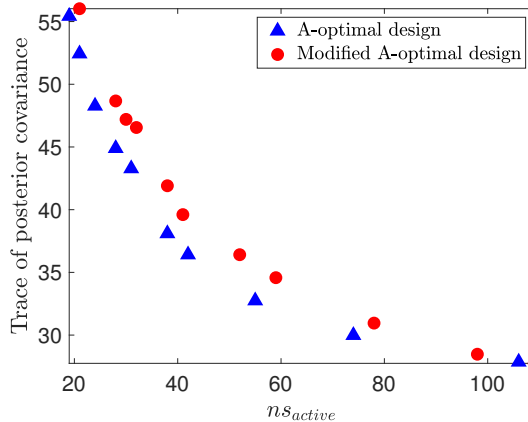


Figure 4.7 Comparison of designs obtained by minimizing the (approximate) A-optimal and modified A-optimal criteria. For each design, we report the exact trace of the corresponding posterior covariance operator.

4.4 Conclusion

In this chapter, we have presented a reweighted ℓ_1 algorithm to solve (4.1), which also produces sparse and binary designs. This method is easy to implement and is applicable to both the A-optimal and modified A-optimal design problems. Numerical experiments suggest that the optimization approach produces reasonable designs that outperform randomly chosen designs. Beyond optimal sensor placement, this optimization approach may find applications in other binary optimization problems requiring sparse solutions.

CHAPTER

5

EFFICIENT COMPUTATION OF THE OED PROBLEM WITHOUT ADJOINT PDE APPLICATIONS

The A-optimal methods discussed in Chapter 3 help reduce the number of forward and adjoint PDE solves for computing an optimal design. However, these methods assume the availability of accurate and efficient solvers for both the forward and adjoint problems. It may be desirable to reduce or completely eliminate the need for adjoint PDE solvers for computing optimal designs for three reasons: (1) for complex PDE models with coupled multiphysics, deriving and implementing the adjoint may be time consuming; (2) many forward PDE implementations are based on legacy software implementations and do not have adjoint capabilities; and (3) the adjoint solves might be computationally expensive. The goal of this chapter is to present an efficient method to solve (4.1) using the A-optimal (2.8) and D-optimal (2.13) criteria when an adjoint solver is not available. The adjoint-free methods presented in this chapter are also advantageous because they can be parallelized and implemented efficiently to decrease the number of PDE computations within the optimization scheme.

As with the approach in Chapter 3, the adjoint-free methods rely on randomized subspace iteration for computing low-rank approximations. We assume the prior covariance operator $\mathbf{\Gamma}_{\text{prior}}$ has rapidly decreasing eigenvalues. Then, we use Algorithm 2 to compute a low rank approximation of the prior covariance operator. Combining this approximation with a different, but equivalent, formulation of the posterior covariance operator, we eliminate the need for an

adjoint solver. We use this idea to compute A- and D-optimal designs.

The chapter is organized as follows. Section 5.1 includes the details regarding the reformulated expressions of the posterior mean and covariance operator. In Section 5.2, we describe how each criterion (A- and D-optimal) is simplified to eliminate the need for the adjoint applies. Section 5.3 includes the computational procedure for the adjoint-free randomized estimators. Section 5.4 includes the numerical results using the reweighted ℓ_1 algorithm from Chapter 4 and the adjoint-free algorithms developed in this chapter on a contaminant source identification problem. Finally, Section 5.5 provides closing remarks and future work regarding the adjoint-free methods.

5.1 Reformulation of the posterior covariance

As discussed in Chapter 2, the posterior covariance operator provides a meaningful description of the uncertainty in linear inverse problem solutions. However, the expressions defined in (2.4) require the application of \mathbf{F}^* to appropriately sized vectors. Since we have assumed that this computation is infeasible, we present an approach to eliminate the applications of the adjoint operator. We begin by reformulating the expressions in (2.4). Notice that in (2.4), the discretized posterior covariance operator contains the operator $\mathbf{F}^* \mathbf{\Gamma}_{\text{noise}} \mathbf{F} : (\mathcal{V}_n, \langle \cdot, \cdot \rangle_{\mathbf{M}}) \rightarrow (\mathcal{V}_n, \langle \cdot, \cdot \rangle_{\mathbf{M}})$. In our reformulation, we want to have an expression for the mean and covariance matrix that includes $\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^* : (\mathbb{R}^{n_s n_t}, \langle \cdot, \cdot \rangle) \rightarrow (\mathbb{R}^{n_s n_t}, \langle \cdot, \cdot \rangle)$, instead of operators on the parameter space. By using the spectral decomposition of the prior covariance matrix, $\mathbf{\Gamma}_{\text{prior}} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^*$, we express $\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*$ as

$$\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^* = \mathbf{F} \mathbf{V} \mathbf{\Lambda} \mathbf{V}^* \mathbf{F}^* = (\mathbf{F} \mathbf{V}) \mathbf{\Lambda} (\mathbf{F} \mathbf{V})^*, \quad (5.1)$$

where $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is the diagonal matrix of eigenvalues of $\mathbf{\Gamma}_{\text{prior}}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ contains the eigenvectors of $\mathbf{\Gamma}_{\text{prior}}$ as columns. Note that $\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*$ can be constructed by forming $\mathbf{F} \mathbf{V}$ without using \mathbf{F}^* . However, this is still computationally expensive, since forming $\mathbf{F} \mathbf{V}$ requires n applications of the forward operator. Recall that the problems we target with our methods assume the discretized parameter dimension n is very large. To reduce the computational cost, we work with a low-rank approximation of the prior covariance operator, i.e., $\mathbf{\Gamma}_{\text{prior}} \approx \mathbf{V}_r \mathbf{\Lambda}_r \mathbf{V}_r^*$ with $r \ll n$. The details of this analysis are discussed in Section 5.3.

The idea of using a low-rank approximation of the prior covariance operator to eliminate adjoint PDE applications is not new. It has been explored within inverse problem theory [97] and has been successfully implemented in geophysical applications to solve inverse problems in an adjoint-free manner [56]. Here we obtain optimal designs, in addition to solving the inverse problem.

To begin the construction of the adjoint-free methods, we consider an alternate (but equivalent) expression of the covariance operator. Proposition 5.1.1 states the reformulated expressions for the posterior covariance and mean, which define the solution for linear inverse problems with a Gaussian prior and additive Gaussian noise model. We start by recording a known technical lemma.

Lemma 5.1.0.1. *If \mathbf{A} and \mathbf{B} are symmetric positive semi-definite $n \times n$ matrices, then \mathbf{AB} has non-negative eigenvalues.*

Proof. See [10, Chapter 3, Exercise 18]. □

Next, recall from Chapter 2 that

$$\mathbf{\Gamma}_{\text{post}}(\mathbf{w}) = (\mathbf{\Gamma}_{\text{prior}}^{-1} + \mathbf{F}^* \mathbf{W}^{1/2} \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W}^{1/2} \mathbf{F})^{-1} \quad (5.2)$$

is defined as the weight-dependent posterior covariance operator. The corresponding expression for the weight-dependent posterior mean (MAP point) is

$$\mathbf{m}_{\text{post}}(\mathbf{w}) = \mathbf{\Gamma}_{\text{post}}(\mathbf{w})(\mathbf{F}^* \mathbf{W}^{1/2} \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W}^{1/2} \mathbf{y} + \mathbf{\Gamma}_{\text{prior}}^{-1} \mathbf{m}_{\text{prior}}). \quad (5.3)$$

We state and justify the equivalent expressions for the weight-dependent posterior covariance and mean in Proposition 5.1.1 [59, 97]. Here we assume that $\mathbf{\Gamma}_{\text{noise}}$ is diagonal, which indicates uncorrelated observations.

Proposition 5.1.1. *Assuming $\mathbf{\Gamma}_{\text{noise}}$ is diagonal, then*

$$\mathbf{\Gamma}_{\text{post}}(\mathbf{w}) = \mathbf{\Gamma}_{\text{prior}} - \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^* (\mathbf{\Gamma}_{\text{noise}} + \mathbf{W} \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*)^{-1} \mathbf{W} \mathbf{F} \mathbf{\Gamma}_{\text{prior}}. \quad (5.4)$$

Furthermore,

$$\mathbf{m}_{\text{post}}(\mathbf{w}) = \mathbf{m}_{\text{prior}} + \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^* (\mathbf{\Gamma}_{\text{noise}} + \mathbf{W} \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*)^{-1} \mathbf{W} (\mathbf{y} - \mathbf{F} \mathbf{m}_{\text{prior}}). \quad (5.5)$$

Proof. Since we assume $\mathbf{\Gamma}_{\text{noise}}$ and \mathbf{W} are both diagonal, in (5.2), $\mathbf{W}^{1/2} \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W}^{1/2}$ can be written as $\mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W}$. We next verify that $\mathbf{\Gamma}_{\text{noise}} + \mathbf{W} \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*$ is invertible. We do this by showing that (i) $\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*$ is symmetric positive semi-definite (with respect to the Euclidean inner product) and (ii) applying Lemma 5.1.0.1 to show that the eigenvalues of $\mathbf{\Gamma}_{\text{noise}} + \mathbf{W} \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*$ are positive.

To show $\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*$ is symmetric positive semi-definite, we proceed as follows. To show symmetry, we use the definition of \mathbf{F}^* and the fact that $\mathbf{\Gamma}_{\text{prior}}$ is symmetric with respect to the \mathbf{M} -weighted inner product (see Figure 2.1) to write

$$(\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*)^\top = (\mathbf{F} \mathbf{M}^{-1} \mathbf{\Gamma}_{\text{prior}}^\top \mathbf{F})^\top = \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1} \mathbf{F}^\top = \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*.$$

Next we show that $\mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*$ is positive semi-definite. Recall that the prior covariance operator is positive definite with respect to the \mathbf{M} -weighted inner product, $\langle \cdot, \cdot \rangle_{\mathbf{M}}$. Then, for any $\mathbf{x} \in \mathbb{R}^{n_s}$,

$$\begin{aligned} \mathbf{x}^\top \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^* \mathbf{x} &= \mathbf{x}^\top \mathbf{F} \mathbf{M}^{-1} \mathbf{\Gamma}_{\text{prior}}^\top \mathbf{F}^\top \mathbf{x} \\ &= (\mathbf{x}^\top \mathbf{F} \mathbf{M}^{-1}) \mathbf{\Gamma}_{\text{prior}}^\top \mathbf{M} (\mathbf{M}^{-1} \mathbf{F}^\top \mathbf{x}) \\ &= \mathbf{z}^\top \mathbf{M} \mathbf{\Gamma}_{\text{prior}} \mathbf{z} = \langle \mathbf{z}, \mathbf{\Gamma}_{\text{prior}} \mathbf{z} \rangle_{\mathbf{M}} \geq 0, \end{aligned}$$

where $\mathbf{z} = \mathbf{M}^{-1}\mathbf{F}^\top \mathbf{x}$.

Finally, we can show the eigenvalues of $\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*$ are positive. Since both \mathbf{W} and $\mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*$ are positive semi-definite, then the product $\mathbf{W}\mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*$ has non-negative eigenvalues by Lemma 5.1.0.1. Furthermore, by adding $\mathbf{\Gamma}_{\text{noise}}$, a symmetric positive definite matrix, to $\mathbf{W}\mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*$, the eigenvalues of the sum become positive and therefore, $\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*$ is invertible.

Applying the Sherman-Morrison-Woodbury formula [75] to (5.2) yields (5.4), the equivalent expression of the posterior covariance matrix.

The final step of the proposition is showing (5.5) is an equivalent expression for the mean of the posterior. The expression (5.5) is obtained by a straight-forward calculation; we substitute (5.4) into (5.3) and rearrange the resulting expression. □

Using (5.4) as the expression for the posterior covariance operator, our approach is two-fold: we exploit the symmetry in $\mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*$ to eliminate the adjoint PDE operators in the posterior covariance operator and replace $\mathbf{\Gamma}_{\text{prior}}$ with a low-rank approximation. This ensures applications of the prior covariance matrix are computationally efficient. In the next section, we formalize our strategy for the A-optimal and D-optimal design criteria.

5.2 Design criteria and their gradients

In the previous section, we discussed Proposition 5.1.1 as the first step to eliminate the need for adjoint applies. In this section, we show how the reformulation of the posterior covariance operator is used to simplify the expressions for the A- and D-optimal criteria, along with their respective gradients, in terms of operators on the measurement space.

Representing the prior covariance by its spectral decomposition, $\mathbf{\Gamma}_{\text{prior}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^*$, we define

$$\mathbf{B} \equiv \mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^* = (\mathbf{F}\mathbf{V})\mathbf{\Lambda}(\mathbf{F}\mathbf{V})^* \quad \text{and} \quad \mathbf{C} \equiv \mathbf{F}\mathbf{\Gamma}_{\text{prior}}^2\mathbf{F}^* = (\mathbf{F}\mathbf{V})\mathbf{\Lambda}^2(\mathbf{F}\mathbf{V})^*. \quad (5.6)$$

The precise implementation details for computing the decomposition $\mathbf{\Gamma}_{\text{prior}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^*$ is discussed in Section 5.3.1. Using \mathbf{B} and \mathbf{C} , we simplify the A- and D-optimality criteria to devise a convenient strategy for their computations. Note that computations of \mathbf{B} and \mathbf{C} only require $\mathbf{F}\mathbf{V}$ and therefore can be done in an adjoint-free manner.

A-optimal criterion and its gradient

We begin with the A-optimal criterion defined in (2.8). Using (5.4) from Proposition 5.1.1, we obtain

$$\Phi_{\text{aopt}}(\mathbf{w}) = -\text{tr} \left(\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*(\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\mathbf{F}\mathbf{\Gamma}_{\text{prior}}\mathbf{F}^*)^{-1}\mathbf{W}\mathbf{F}\mathbf{\Gamma}_{\text{prior}} \right).$$

Then, by the cyclic property of the trace function and the definitions in (6.9), we write the A-optimal criterion as follows:

$$\Phi_{\text{aopt}}(\mathbf{w}) = -\text{tr} \left((\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})^{-1} \mathbf{WC} \right). \quad (5.7)$$

Next, we consider the reformulation of the gradient of the criterion. To aid in the gradient computation, we define

$$\frac{\partial \mathbf{W}}{\partial w_j} = \mathbf{E}_j = \mathbf{I}_{n_t} \otimes \mathbf{e}_j \mathbf{e}_j^\top, \quad (5.8)$$

where \mathbf{e}_j is the j^{th} column of \mathbf{I}_{n_s} . Using Theorems B.17 and B.19 from [104], the derivative of (5.7) with respect to each w_j is

$$\frac{\partial \Phi_{\text{aopt}}(\mathbf{w})}{\partial w_j} = \text{tr} \left((\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})^{-1} \mathbf{E}_j \mathbf{B} (\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})^{-1} \mathbf{WC} - (\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})^{-1} \mathbf{E}_j \mathbf{C} \right),$$

for $j = 1, \dots, n_s$. A straightforward simplification yields,

$$\frac{\partial \Phi_{\text{aopt}}(\mathbf{w})}{\partial w_j} = \text{tr} \left(\mathbf{E}_j [\mathbf{B} (\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})^{-1} \mathbf{W} - \mathbf{I}_{n_s n_t}] \mathbf{C} (\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})^{-1} \right), \quad (5.9)$$

for $j = 1, \dots, n_s$. The derived expressions of the A-optimal criterion (5.7) and its gradient (5.9) are in terms of \mathbf{B} and \mathbf{C} , which only require computation of \mathbf{FV} and are therefore, adjoint-free.

D-optimal criterion and its gradient

Recall the definition of the D-optimal criterion in (2.13). Using Sylvester's determinant identity [82, Corollary 2.1] and by assuming that $\mathbf{\Gamma}_{\text{noise}}$ is diagonal, we write the D-optimal criterion as

$$\Phi_{\text{dopt}}(\mathbf{w}) = \log \det(\mathbf{I} + \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{W} \mathbf{F} \mathbf{\Gamma}_{\text{prior}} \mathbf{F}^*).$$

Then, through the use of (6.9) and basic properties of the log-determinant, we obtain

$$\begin{aligned} \Phi_{\text{dopt}}(\mathbf{w}) &= \log \det(\mathbf{\Gamma}_{\text{noise}}^{-1} (\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})) \\ &= \log \det(\mathbf{\Gamma}_{\text{noise}}^{-1}) + \log \det(\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB}). \end{aligned}$$

Further simplification occurs because $\mathbf{\Gamma}_{\text{noise}}^{-1}$ does not depend on \mathbf{w} , resulting in the expression of the D-optimal criterion,

$$\Phi_{\text{dopt}}(\mathbf{w}) = \log \det(\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB}). \quad (5.10)$$

The gradient computation of the D-optimal objective function again utilizes Theorems B.17 and B.19 from [104], which results in the gradient

$$\partial_j \Phi_{\text{dopt}}(\mathbf{w}) = \text{tr} \left(\mathbf{E}_j \mathbf{B} (\mathbf{\Gamma}_{\text{noise}} + \mathbf{WB})^{-1} \right), \quad (5.11)$$

for $j = 1, \dots, n_s$.

Recall that we have defined our criteria and their respective gradients in terms of the matrices \mathbf{B} and \mathbf{C} from (6.9). The computation of these matrices require efficient evaluation of \mathbf{FV} . However, \mathbf{FV} is the same size as \mathbf{F} ($n_s n_t \times n$), suggesting that the cost of computing \mathbf{FV} is equivalent to building \mathbf{F} . To ensure that our method is computationally efficient, we seek a low-rank approximation of $\mathbf{\Gamma}_{\text{prior}}$ to decrease the number forward PDE solves required to compute \mathbf{FV} .

5.3 Computational procedure

In this section, we begin by describing the computation of a low-rank approximation of $\mathbf{\Gamma}_{\text{prior}}$ and then using the approximation, we discuss the efficient evaluation of the OED criteria. We mention that our approach is only suitable for problems where $\mathbf{\Gamma}_{\text{prior}}$ can be approximated with a low-rank approximation with sufficient accuracy.

5.3.1 Low-rank approximation of $\mathbf{\Gamma}_{\text{prior}}$

Since the prior covariance is symmetric with respect to the \mathbf{M} -weighted inner product, we use the similarity transformation, $\mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1/2}$, to obtain an operator that is symmetric with respect to the Euclidean inner product. As a result, Algorithm 2 can be applied to $\mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1/2}$. Let $\mathbf{\Gamma}_{\text{prior}} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^*$ be the spectral decomposition of the prior covariance. Note that

$$\begin{aligned} \mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1/2} &= \mathbf{M}^{1/2} \mathbf{V} \mathbf{\Lambda} \mathbf{V}^* \mathbf{M}^{-1/2} \\ &= \mathbf{M}^{1/2} \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top \mathbf{M} \mathbf{M}^{-1/2} \\ &= \widehat{\mathbf{V}} \mathbf{\Lambda} \widehat{\mathbf{V}}^\top, \end{aligned}$$

with $\widehat{\mathbf{V}} = \mathbf{M}^{1/2} \mathbf{V}$ and $\widehat{\mathbf{V}}^\top \widehat{\mathbf{V}} = \mathbf{I}$. From this, we obtain the relationship $\mathbf{V} = \mathbf{M}^{-1/2} \widehat{\mathbf{V}}$. This relationship allows us to compute a low-rank approximation of $\mathbf{\Gamma}_{\text{prior}}$ by obtaining the spectral decomposition of $\mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1/2}$.

In our work, we apply Algorithm 2 to $\mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1/2}$ with parameters k and p . Recall that k is the target rank of $\mathbf{\Gamma}_{\text{prior}}$ and p is the oversampling parameter and define $r = k + p$. Then we obtain

$$\mathbf{M}^{1/2} \mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1/2} \approx \widehat{\mathbf{V}}_r \widetilde{\mathbf{\Lambda}}_r \widehat{\mathbf{V}}_r^\top.$$

The result is a low-rank approximation of $\mathbf{\Gamma}_{\text{prior}}$ which satisfies

$$\mathbf{\Gamma}_{\text{prior}} \mathbf{M}^{-1} \approx \widetilde{\mathbf{V}}_r \widetilde{\mathbf{\Lambda}}_r \widetilde{\mathbf{V}}_r^\top, \quad (5.12)$$

where $\widetilde{\mathbf{V}}_r = \mathbf{M}^{-1/2} \widehat{\mathbf{V}}_r$. Assuming $\mathbf{M}^{1/2}$ and $\mathbf{M}^{-1/2}$ are easy to compute, the algorithm is straightforward to implement. Additionally, the implementation of Algorithm 2 decreases the computational cost because the algorithm can be applied in parallel and in a matrix-free manner.

Now let $\widehat{\mathbf{F}}_r = \mathbf{F}\widetilde{\mathbf{V}}_r$. Then using the approximation (5.12), we estimate \mathbf{B} and \mathbf{C} defined in (6.9) by

$$\widehat{\mathbf{B}} = \widehat{\mathbf{F}}_r \widetilde{\mathbf{\Lambda}}_r \widehat{\mathbf{F}}_r^\top \quad \text{and} \quad \widehat{\mathbf{C}} = \widehat{\mathbf{F}}_r \widetilde{\mathbf{\Lambda}}_r^2 \widehat{\mathbf{F}}_r^\top, \quad (5.13)$$

respectively. Recall that the matrix $\widetilde{\mathbf{V}}_r \in \mathbb{R}^{n \times r}$, where n is the discretized parameter dimension and r is the target rank for the prior covariance matrix. Therefore, the computation of $\widehat{\mathbf{F}}_r \in \mathbb{R}^{n_s n_t \times r}$ requires r PDE solves. And since $\widehat{\mathbf{B}} \in \mathbb{R}^{n_s n_t \times n_s n_t}$ and $\widehat{\mathbf{C}} \in \mathbb{R}^{n_s n_t \times n_s n_t}$ do not depend on the weight vector \mathbf{w} , these operators can be precomputed. We mention that if the number of observations are too large, i.e., $n_s n_t \gg r$, then the storage of $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{C}}$ may not be feasible and instead, one could store $\widehat{\mathbf{F}}_r \in \mathbb{R}^{n_s n_t \times r}$ and $\widetilde{\mathbf{\Lambda}}_r$ to efficiently compute the OED.

Computing the A-optimal criterion

Using the estimators $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{C}}$ from (5.13), the adjoint-free estimator of the A-optimal criterion is defined as

$$\widehat{\Phi}_{\text{aopt}}(\mathbf{w}) = -\text{tr} \left((\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}})^{-1} \mathbf{W}\widehat{\mathbf{C}} \right). \quad (5.14)$$

And the adjoint-free estimator of the gradient is

$$\widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}) = \text{tr} \left(\mathbf{E}_j (\widehat{\mathbf{B}}(\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}})^{-1} \mathbf{W} - \mathbf{I}_{n_s n_t}) \widehat{\mathbf{C}} (\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}})^{-1} \right), \quad (5.15)$$

for $j = 1, \dots, n_s$. We present the algorithm for computing the A-optimal criterion and its gradient defined in Algorithm 5.

The algorithm consists of two parts: precomputation and evaluation of the A-optimal estimator and its gradient. The precomputation is done once per OED problem and consists of r forward PDE solves to compute and explicitly store $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{C}}$. Note that the evaluation of the A-optimal criterion and its gradient require no additional PDE solves.

As a part of the algorithms presented in this chapter, we require computation of the expression $\text{tr}(\mathbf{E}_j \mathbf{X})$, for $j = 1, \dots, n_s$, where $\mathbf{X} \in \mathbb{R}^{n_s n_t \times n_s n_t}$ is context-dependent. Instead of computing $\text{tr}(\mathbf{E}_j \mathbf{X})$ for each j , we suggest a vectorized MATLAB implementation to compute all of the terms simultaneously by taking advantage of the definition of \mathbf{E}_j in (5.8). The computation consists of taking the diagonal of \mathbf{X} , reshaping it to be size $n_s \times n_t$, and then summing along the rows of the new matrix. This observation is justified since for $j = 1, \dots, n_s$,

$$\text{tr}(\mathbf{E}_j \mathbf{X}) = \sum_{i=1}^{n_t} x_{(i-1)n_s+j, (i-1)n_s+j}.$$

Algorithm 5 Randomized method for computing $\widehat{\Phi}_{\text{aopt}}(\mathbf{w})$ and $\widehat{\nabla}\Phi_{\text{aopt}}(\mathbf{w})$.

Input: Target rank k , oversampling parameter $p \geq 2$, design \mathbf{w} , mass matrix \mathbf{M} and $\mathbf{\Gamma}_{\text{prior}}$.

Output: OED objective $\widehat{\Phi}_{\text{aopt}}(\mathbf{w})$ and gradient $\widehat{\nabla}\Phi_{\text{aopt}}(\mathbf{w})$.

Precomputation

- 1: Compute $[\widehat{\mathbf{V}}, \mathbf{\Lambda}_r] = \text{randEig}(\mathbf{M}^{1/2}\mathbf{\Gamma}_{\text{prior}}\mathbf{M}^{-1/2}, k, p)$ and compute $\mathbf{V} = \mathbf{M}^{-1/2}\widehat{\mathbf{V}}$.
 - 2: **for** $i = 1$ to $r = k + p$ **do**
 - 3: Compute $\mathbf{f}_i = \mathbf{F}\mathbf{v}_i$, where \mathbf{v}_i are the columns of \mathbf{V} .
 - 4: **end for**
 - 5: Let $\widehat{\mathbf{F}}_r = [\mathbf{f}_1, \dots, \mathbf{f}_r]$ and compute $\widehat{\mathbf{F}}_r \mathbf{\Lambda}_r \widehat{\mathbf{F}}_r^\top = \widehat{\mathbf{B}} \in \mathbb{R}^{n_s n_t \times n_s n_t}$ and $\widehat{\mathbf{F}}_r \mathbf{\Lambda}_r^2 \widehat{\mathbf{F}}_r^\top = \widehat{\mathbf{C}} \in \mathbb{R}^{n_s n_t \times n_s n_t}$.
-

Evaluating $\widehat{\Phi}_{\text{aopt}}(\mathbf{w})$ and $\widehat{\nabla}\Phi_{\text{aopt}}(\mathbf{w})$

- 6: Compute $\mathbf{S} = (\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}})^{-1}$ and $\mathbf{P} = (\widehat{\mathbf{B}}\mathbf{S}\mathbf{W} - \mathbf{I}_{n_s n_t}) \widehat{\mathbf{C}}\mathbf{S}$.

- 7: Compute

$$\widehat{\Phi}_{\text{aopt}}(\mathbf{w}; r) = -\text{tr}(\mathbf{S}\mathbf{W}\widehat{\mathbf{C}}).$$

- 8: **for** $j = 1$ to n_s **do**

- 9: Compute

$$\widehat{\partial_j \Phi}_{\text{aopt}}(\mathbf{w}) = \text{tr}(\mathbf{E}_j \mathbf{P}).$$

- 10: **end for**

- 11: **Return** $\widehat{\Phi}_{\text{aopt}}(\mathbf{w})$ and $\widehat{\nabla}\Phi_{\text{aopt}}(\mathbf{w}; r)$.
-

So, written in MATLAB notation, we have

$$\begin{bmatrix} \text{tr}(\mathbf{E}_1 \mathbf{X}) \\ \text{tr}(\mathbf{E}_2 \mathbf{X}) \\ \vdots \\ \text{tr}(\mathbf{E}_{n_s} \mathbf{X}) \end{bmatrix} = \text{reshape}(\text{diag}(\mathbf{X}), n_s, n_t) \mathbf{1},$$

where $\mathbf{1} = [1, 1, \dots, 1]^\top \in \mathbb{R}^{n_t}$.

Computing the D-optimal objective function

The D-optimal estimator is computed in a similar way as the A-optimal criterion. We utilize the same definition of $\widehat{\mathbf{B}}$ as in (5.13) to obtain the adjoint-free estimator of the D-optimal objective function

$$\widehat{\Phi}_{\text{dopt}}(\mathbf{w}) = \log \det(\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}}). \quad (5.16)$$

The corresponding gradient of the estimator is

$$\widehat{\partial_j \Phi}_{\text{dopt}}(\mathbf{w}) = \text{tr}(\mathbf{E}_j \mathbf{B}(\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}})^{-1}), \quad (5.17)$$

for $j = 1, \dots, n_s$.

Algorithm 6 describes the procedure for computing the estimators of the D-optimal objective function and its gradient from (5.16) and (5.17).

Algorithm 6 Randomized method for computing $\hat{\Phi}_{\text{dopt}}(\mathbf{w})$ and $\widehat{\nabla\Phi}_{\text{dopt}}(\mathbf{w})$.

Input: Target rank k , oversampling parameter $p \geq 2$, design \mathbf{w} , mass matrix \mathbf{M} and $\mathbf{\Gamma}_{\text{prior}}$.

Output: OED objective $\hat{\Phi}_{\text{dopt}}(\mathbf{w})$ and gradient $\widehat{\nabla\Phi}_{\text{dopt}}(\mathbf{w})$.

Precomputation

- 1: Compute $[\widehat{\mathbf{V}}, \mathbf{\Lambda}_r] = \text{randEig}(\mathbf{M}^{1/2}\mathbf{\Gamma}_{\text{prior}}\mathbf{M}^{-1/2}, k, p)$ and compute $\mathbf{V} = \mathbf{M}^{-1/2}\widehat{\mathbf{V}}$.
 - 2: **for** $i = 1$ to r **do**
 - 3: Compute $\mathbf{f}_i = \mathbf{F}\mathbf{v}_i$, where \mathbf{v}_i are the columns of \mathbf{V} .
 - 4: **end for**
 - 5: Let $\widehat{\mathbf{F}}_r = [\mathbf{f}_1, \dots, \mathbf{f}_r]$ and compute $\widehat{\mathbf{F}}_r\mathbf{\Lambda}_r\widehat{\mathbf{F}}_r^\top = \widehat{\mathbf{B}} \in \mathbb{R}^{n_s \times r}$.
-

Evaluating $\hat{\Phi}_{\text{dopt}}(\mathbf{w})$ and $\widehat{\nabla\Phi}_{\text{dopt}}(\mathbf{w})$

- 6: Compute $\mathbf{S} = (\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}})^{-1}$ and $\mathbf{P} = \widehat{\mathbf{B}}\mathbf{S}$.
- 7: Compute

$$\hat{\Phi}_{\text{dopt}}(\mathbf{w}; r) = \log \det(\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}}).$$

- 8: **for** $j = 1$ to n_s **do**
- 9: Compute

$$\widehat{\partial_j\Phi}_{\text{dopt}}(\mathbf{w}) = \text{tr}(\mathbf{E}_j\mathbf{P}).$$

- 10: **end for**
 - 11: **Return** $\hat{\Phi}_{\text{dopt}}(\mathbf{w})$ and $\widehat{\nabla\Phi}_{\text{dopt}}(\mathbf{w})$.
-

Notice that the precomputation for Algorithm 5 and Algorithm 6 are similar. The difference is that the A-optimal estimators require both $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{C}}$, while the D-optimal estimators only require $\widehat{\mathbf{B}}$. Therefore, the D-optimal estimators require less storage than Algorithm 5. However, both criteria require r forward PDE solves to compute an optimal experimental design.

Computing the MAP point

The adjoint-free framework can be used to compute the MAP point. While the OED problem requires no knowledge about the MAP point, in application problems, it is common to utilize the MAP point to represent the reconstruction of the unknown parameter. However, since we have assumed that the adjoint of the forward PDE operator is unavailable, using (5.3) to compute the posterior mean is infeasible. Instead we approximate the MAP point by using (5.5) along with the low-rank approximation of the prior covariance operator. The resulting adjoint-free

estimator of the MAP point is

$$\mathbf{m}_{\text{post}} \approx \mathbf{m}_{\text{prior}} + \mathbf{V}\mathbf{D}\widehat{\mathbf{F}}_r^\top (\mathbf{\Gamma}_{\text{noise}} + \mathbf{W}\widehat{\mathbf{B}})^{-1}(\mathbf{y} - \mathbf{W}\mathbf{F}\mathbf{m}_{\text{prior}}),$$

where $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{F}}_r$ are defined in (5.13) and \mathbf{F} is the forward PDE operator. Notice that the computation of the MAP point requires $r + 1$ forward PDE solves: r forward PDE solves are needed to compute $\widehat{\mathbf{F}}_r$ and one forward PDE solve is needed to compute $\mathbf{F}\mathbf{m}_{\text{prior}}$.

5.4 Numerical results

In this section, we illustrate the usefulness of the adjoint-free estimators to compute optimal designs for the model source identification problem described in Section 3.3.1. Recall that for this model problem, the prior covariance is the square of a Laplacian-like operator \mathcal{A}^{-2} . To utilize the adjoint-free estimators for the A- and D-optimal criterion, the prior covariance operator must be approximately low-rank. Therefore, we begin by numerically investigating if our choice of $\mathbf{\Gamma}_{\text{prior}}$ satisfies the assumption that the prior covariance operator is approximately low-rank.

We do this by looking at both the decay of eigenvalues and the resulting cumulative eigenvalue fraction. Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ be the eigenvalues of $\mathbf{\Gamma}_{\text{prior}}$. We define the cumulative eigenvalue fraction (or ratio of variances) for the first i eigenvalues by

$$\xi(i) = \frac{\sum_{j=1}^i \lambda_j}{\sum_{j=1}^n \lambda_j}, \quad \text{for } 1 \leq i \leq n.$$

Figure 5.1 shows the eigenvalue decay and the cumulative eigenvalue fraction $\xi(i)$ for our choice of the prior covariance operator, where the discretized parameter dimension is $n = 2605$. The cumulative eigenvalue fraction indicates it is reasonable to assume that the prior covariance is approximately low-rank. Specifically, Figure 5.1 indicates the first 73 eigenvalues of $\mathbf{\Gamma}_{\text{prior}}$ result in 97% of the cumulative eigenvalue fraction. In a similar way, 99% of the cumulative eigenvalue fraction of $\mathbf{\Gamma}_{\text{prior}}$ is obtained with 175 eigenvalues.

Next, we solve (4.1) using our adjoint-free estimators to estimate the OED criterion. In the remainder of the section, we consider values of $r \geq 98$ in Algorithm 5 and Algorithm 6 to ensure a sufficiently accurate low-rank representation of $\mathbf{\Gamma}_{\text{prior}}$. Additionally, we use the reweighted ℓ_1 framework described in Algorithm 4 as means to ensure a sparse and binary solution to (4.1), with the user-defined parameter in this algorithm chosen to be $\epsilon = 1/2^8$. We also fix the penalty coefficient in (4.1) to be $\gamma = 9$ for the entirety of this section.

5.4.1 A-optimal designs

In this experiment, we specifically focus on understanding how the choice of r in Algorithm 5 affects the resulting designs. Using the prescribed parameters, we compute designs for $r = 98, 200, 400$, and 1000. The resulting designs are shown in Figure 5.2. We see from Figure 5.2 that

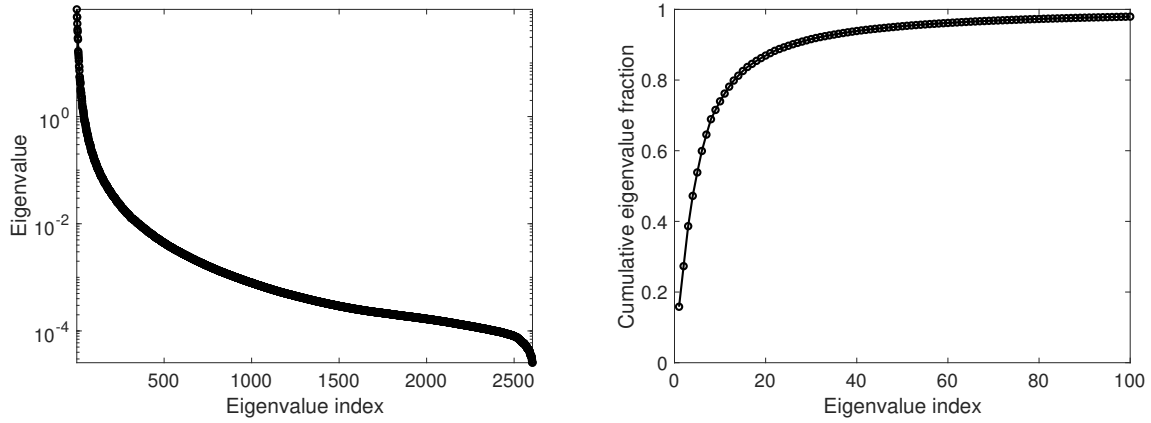


Figure 5.1 The eigenvalue decay (left) and the cumulative eigenvalue fraction (right) of Γ_{prior} .

the design computed with $r = 98$ has only one sensor location which differs from the designs computed with other target ranks. This indicates that beyond a certain target rank, the designs show very mild dependence on the target rank suggesting that the low-rank approximation is sufficiently accurate. Furthermore, the adjoint-free designs obtained are in good agreement with the designs obtained using the method of Chapter 3.

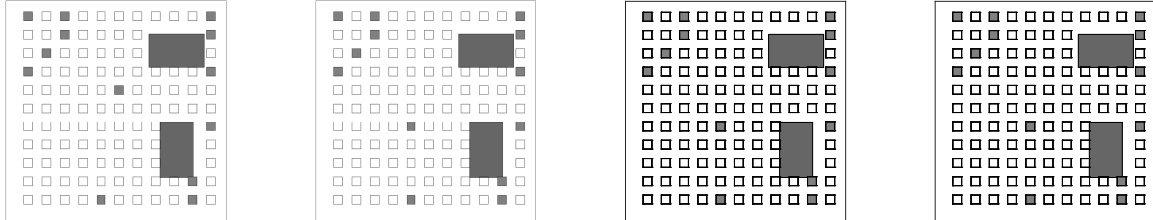


Figure 5.2 Designs computed using $r = 98$ (left), $r = 200$ (middle left), $r = 400$ (middle right), and $r = 1000$ (left).

We also make some remarks regarding the computational cost of the method in Table 5.1. Specifically, we record both the time needed to precompute (5.13) and to solve (4.1). The precomputation time increases as r increases; however, the time to solve the optimization problem remains approximately constant. The computational cost of computing a solution to the optimization problem also remains approximately constant in terms of the number of function evaluations and outer iterations to terminate the reweighted ℓ_1 algorithm.

Table 5.1 Numerical results obtained from computing the A-optimal designs in Figure 5.2.

r	98	200	400	1000
Precomputation (seconds)	9.61	12.73	18.95	38.69
Optimization (seconds)	17.31	16.82	19.01	17.88
Number of function evaluations	1322	1283	1312	1238
Outer iterations	14	12	12	12
ns_{active}	13	13	13	13

5.4.2 D-optimal designs

Using the D-optimal estimator described by Algorithm 6 with $r = 98, 200, 400$, and 1000, we solve (4.1) to obtain 4 optimal designs. These designs are shown in Figure 5.3. In each subfigure, we see that the designs have many sensors in common and that the designs using $r = 400$ and $r = 1000$ are identical.

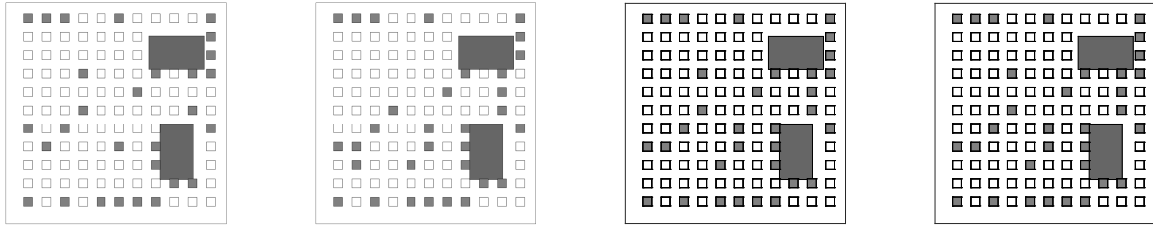


Figure 5.3 Designs computed using $r = 98$ (left), $r = 200$ (middle left), $r = 400$ (middle right), $r = 1000$ (right) with D-optimal criterion.

We record observations about the optimization procedure in Table 5.2. As with the A-optimal criterion problem, the precomputation time increases while the amount of time to solve the optimization problem remains relatively constant. We mention that here we use the same value for the penalty coefficient γ in (4.1) for both the A- and D-optimal designs. As shown in Table 5.1 and Table 5.2, the same penalty coefficient yields different number of active sensors, ns_{active} , for the A- and D-optimal designs. Therefore, a comparison between the A- and D-optimal is not performed at this time.

5.5 Conclusion

The randomized adjoint-free methods presented in this chapter are a convenient choice when solving the OED problem for Bayesian linear inverse problems in which an implementation of

Table 5.2 Numerical results obtained from computing the D-optimal designs in Figure 5.3.

r	98	200	400	1000
Precomputation (seconds)	9.63	12.72	18.93	38.53
Optimization (seconds)	6.47	7.01	8.85	7.97
Number of function evaluations	683	734	837	765
Outer iterations	6	6	7	7
ns_{active}	29	32	33	33

the adjoint PDE solver is either unavailable or undesirable. These methods are constructed with the assumption that the Gaussian prior covariance operator can be chosen to be approximately low-rank. Therefore, problems where this assumption is reasonable benefit from a decreased computational cost through use of the adjoint-free estimators. Specifically, the algorithms developed require fewer forward solves in comparison to the methods in Chapter 3 for evaluating the objective function. In addition, the benefits of the Algorithm 1 are also inherited by the methods presented in this chapter. That is, our algorithms are easy to implement, parallelizable, and can be applied in a matrix-free fashion. The numerical results confirm that our proposed adjoint-free method is efficient when the prior covariance is approximately low-rank. However, this may not be true under all circumstances and a different approach may be needed.

CHAPTER

6

A NON-INTRUSIVE SURROGATE MODEL FOR PREDICTING MICROSTRUCTURE MORPHOLOGIES DURING PHYSICAL VAPOR DEPOSITION

While our focus, thus far, has been on efficient solution methods for OED problems, this chapter's goal is to develop a surrogate model for physical vapor deposition (PVD). PVD is a broad category of experimental processes, where thin films are applied to a substrate of choice [7, 70, 72]. The process consists of solid (metal) materials being converted to a vapor and deposited to a surface, where the vaporized materials are returned to solid form. This process allows very thin coatings to be applied to various substrates. Because the choice of materials being used to create the thin film react and form structures at the microscopic scale (microns to centimeters), the resulting thin film may exhibit favorable properties such as conductivity, strength, and corrosion resistance [7]. Consequently, PVD has been successfully utilized as a process to apply thin films to surfaces with applications in aerospace, automotive industry, manufacturing, textiles, and electronics [7, 90]. Moreover, PVD is used by scientists to fabricate new types of materials, which may lead to advancements in these application areas [33].

The PVD process can be described using a mathematical model developed in [95]. This

model involves a set of coupled PDEs and is solved with a numerical scheme (finite difference in space and explicit Euler in time) for a set of uncertain parameters after appropriate non-dimensionalization, $\boldsymbol{\theta} \in [-1, 1]^{n_p}$. Although the PDE model provides a systematic framework to simulate the PVD process, the computational cost of the model hinders efficient exploration of the parameter space. This is the motivation for our work; a surrogate model results in the ability to explore the parameter space with smaller computational cost than using experiments or detailed numerical simulations. Moreover, in the framework we use, global sensitivity analysis can be performed on the surrogate model to determine the importance of model parameters using the theory of polynomial chaos. Finally, a surrogate model can be used to inform experimentalist as to which parameters may yield a favorable output and therefore, physical costs associated with running experiments can be controlled.

More specifically, we assume that the random field of interest is the output of the high-fidelity PDE model of the form $\mathbf{g} : [-1, 1]^{n_p} \rightarrow \mathbb{R}^N$, where n_p is the number of input parameters and N is the grid size of the discretized system. The high-fidelity model for PVD is computationally expensive so a surrogate model $\hat{\mathbf{g}}$ is desirable. Our strategy is to construct a surrogate model for the random vector \mathbf{g} using a combination of principal component analysis (PCA) and polynomial chaos expansion (PCE). Both of these mathematical tools allow a surrogate model to be constructed in a non-intrusive manner. That is, we use samples from the high-fidelity model \mathbf{g} to create an approximate model, which describes the input-output relationship between the model parameters and the resulting output of the high-fidelity model. Specifically, PCA is used to find a lower-dimensional subspace that yields approximate realizations of $\mathbf{g}(\boldsymbol{\theta})$. Then a linkage between the coefficients of the PCA expansion and the model parameters $\boldsymbol{\theta}$ is formed using PCE. The result of these two steps is a low-fidelity model, $\hat{\mathbf{g}}$. The strategy of combining PCA and PCE to create a surrogate model has been explored both from a theoretical point of view and applied to different applications, [4, 25, 39, 62, 65, 73, 77, 109]. To our knowledge, this has not been applied to PVD.

This chapter is structured as follows. Section 6.1 includes a description of the high-fidelity model: (1) the physics-based PDE model describing the process of PVD and the corresponding computational procedure and (2) computation of the two-point statistics. Section 6.2 provides an overview of the non-intrusive methods that are used to construct the surrogate model: (1) PCA and (2) PCE. Section 6.3 describes numerical results associated with constructing the surrogate model and the corresponding approximation error. And Section 6.4 summarizes the work presented in this chapter and future work needed to be done with respect to applying the mathematical theory to the PVD process.

6.1 High-fidelity model for PVD

As mentioned earlier, PVD is an experimental processes to apply thin film coatings to surfaces. The properties of the thin films are typically determined based on the structure of the material

at a microscopic scale. The term “microstructure” of a material is defined as the physical composition of the material at the micro scale over a physical domain. The microstructure consists of phases which are separated by boundaries, where a phase is a section of the material which exhibits a unique chemical makeup. For our work, we consider a binary alloy thin film as the output of the PVD process; therefore, we have two phases (one for each species) that define the microstructure of the thin film.

Our goal is to develop a surrogate model $\hat{\mathbf{g}}$ to describe the structure of the microstructure for a given set of uncertain model parameters $\boldsymbol{\theta} \in [-1, 1]^{n_p}$. Within the materials science community, microstructures are commonly analyzed in terms of the two-point statistics of their phases within the microstructure [19, 35, 52, 54, 62, 83, 109]. The reason we consider a microstructure’s two-point statistics is because it provides details into each realization of the spatial structure of the composition. Therefore, we seek to create a surrogate model describing the two-point statistics of each microstructure obtained for a variety of process parameters. The discretized autocorrelation field is what we consider as the output of our high-fidelity model \mathbf{g} . In this section, we discuss the PDE model we use to describe the PVD process and the computation of high-fidelity model.

6.1.1 Phase-field PVD model and numerical solution

We generate samples of our high-fidelity model \mathbf{g} by solving a physics-based model developed in [95]. The model was constructed to account for the major aspects of the PVD process and has been shown to capture experimentally observed thin film microstructures. Specifically, the model accounts for the transport of the vapor to the top of the thin film, the transition of vapor to solid thin film, and the separating dynamics of the two alloys within the thin film. The authors of [95] use phase-field modeling techniques to derive the model. We devote this section to explaining the mathematical model for the PVD process of a binary alloy thin film and describing the computational procedure used to solve the phase-field model.

Phase-field model

Phase-field models are commonly used to describe the evolution of microstructures and their resulting properties [22, 76]. The models describe the evolution of materials (e.g., phase α and phase β during phase separation), by taking into account the amount of energy needed to convert between the two phases. Phase-field models are advantageous because they assume a diffuse interface instead of a sharp transition between phase α and phase β . This means that the transition between the two phases is assumed to be continuous over a small finite width. As a result, the evolution of the interactions between two phases can be described with one continuous field instead of tracking the individual interfaces between the two phases. The continuous fields are often called phase-fields or order-parameters.

In [95], the phase-fields are defined in terms of the difference between the fractions of each

phase. A phase-field $h(\mathbf{x}, t)$ is defined as

$$h(\mathbf{x}, t) = \frac{a(\mathbf{x}, t) - b(\mathbf{x}, t)}{a(\mathbf{x}, t) + b(\mathbf{x}, t)},$$

where $a(\mathbf{x}, t)$ represents the mass of phase α and $b(\mathbf{x}, t)$ represents the mass of phase β at location \mathbf{x} and time t . As a result, when $h(\mathbf{x}, t) = 1$, the location \mathbf{x} is classified as being phase α while $h(\mathbf{x}, t) = -1$ indicates the location is classified as phase β .

The PVD model we consider uses two phase-fields, $\phi(\mathbf{x}, t)$ and $c(\mathbf{x}, t)$, which depend on the location \mathbf{x} in the domain and the time t within the PVD process. Each phase-field models the separation of different phases that occur during the PVD process: the field variable $\phi(\mathbf{x}, t)$ distinguishes between the solid and vapor phases, while the field variable $c(\mathbf{x}, t)$ distinguishes between the equilibrium components of the binary alloy species.

Depending on the quantities being represented by the phase-fields, phase-fields are classified as being either conserved or non-conserved. A phase-field is classified as being conserved if the integral of the field over the entire space domain is a constant [81]. If a phase-field is conserved, then the time evolution is described by the Cahn-Hilliard equations, and if a phase-field is non-conserved, then a time-dependent Allen-Cahn equation describes the time evolution [76, 81]. In the PVD process, the phase-fields $c(\mathbf{x}, t)$ and $\phi(\mathbf{x}, t)$ are conserved quantities. Therefore, we use the Cahn-Hilliard equations to model the time evolution of the phase-fields we consider [17, 55, 63, 81]. Before we discuss how the Cahn-Hilliard equations are used to model PVD, we discuss the free energy of the physical system.

Free energy is a thermodynamic term that describes the internal energy of a system [63]. In our problem, neglecting any elastic effects, the total free-energy of the PVD process is taken as a functional of the conserved field variables $\phi(\mathbf{x}, t)$ and $c(\mathbf{x}, t)$ and their gradients, and has the following form:

$$\mathcal{E}(\phi, c) = \int \left\{ f(\phi) + \frac{\kappa_\phi^2}{2} (\nabla \phi)^2 + s(\phi) \left(f(c) + \frac{\kappa_c^2}{2} (\nabla c)^2 \right) \right\} d\mathcal{D}. \quad (6.1)$$

Here, κ_ϕ and κ_c are the gradient energy coefficients. The interpolation function, $s(\phi) = (1 + \phi)^2/4$, ensures that compositional contributions are eliminated within the vapor phase. The first two terms in (6.1) describe free energy between the solid and the vapor, while the terms in parentheses describe the free energy within the growing thin film. The double-well function, f , describes the equilibrium free-energy contributions of the two phases associated with the phase-field, and is taken to be

$$f(v) = \omega_v \left(\frac{v^4}{4} - \frac{v^2}{2} \right),$$

where ω_v is the energetic barriers between the equilibrium phases. In other words, the function f describe how much energy is required to transition between phase α and phase β . We refer the reader to [63, 81] for further details on the derivation of the free energy of a physical system.

Because physical systems seek to minimize their free energy over time, the equations of motion for thin film growth and composition evolution are determined according to the Cahn-Hilliard equations.

The governing equations for the composition evolution are

$$\begin{aligned} \frac{\partial c}{\partial t} - \nabla \cdot \left[M_c(\phi, c) \nabla \frac{\delta \mathcal{E}(\phi, c)}{\delta c} \right] &= 0 && \text{in } \mathcal{D} \times (0, T), \\ c(\mathbf{x}, 0) &= c_0(\mathbf{x}) && \text{in } \mathcal{D}, \\ \mathcal{B}_c(c, \mathbf{x}, t) &= 0 && \text{on } \partial \mathcal{D} \times (0, T), \end{aligned} \quad (6.2)$$

where $\frac{\delta \mathcal{E}(\phi, c)}{\delta c}$ is the variational derivative of (6.1) with respect to $c(\mathbf{x}, t)$ and $M_c(\phi, c)$ is the structurally and compositionally dependent mobility function. This mobility function captures the relative difference between bulk and surface mobilities and allows for a smooth transition between phase α and phase β , and their respective bulk or surface mobilities. In (6.2), $\mathcal{B}_c(c, \mathbf{x}, t)$ is a generic notation to describe the boundary condition; the specific form is given in Section 6.1.3.

To describe thin film growth, it is necessary to incorporate a source term in the Cahn-Hilliard equation which represents the amount of vapor reaching the thin film. Evolution of the solid-vapor field, $\phi(\mathbf{x}, t)$, is then given as

$$\begin{aligned} \frac{\partial \phi}{\partial t} - \nabla \cdot \left[M_\phi(\phi) \nabla \frac{\delta \mathcal{E}}{\delta \phi}(\phi, c) \right] - \nabla \phi \cdot (\rho \mathbf{v}) &= 0 && \text{in } \mathcal{D} \times (0, T), \\ \phi(\mathbf{x}, 0) &= \phi_0(\mathbf{x}) && \text{in } \mathcal{D}, \\ \mathcal{B}_\phi(\phi, \mathbf{x}, t) &= 0 && \text{on } \partial \mathcal{D} \times (0, T), \end{aligned} \quad (6.3)$$

The first term in (6.3) captures arbitrary surface morphology formation and surface diffusion effects (i.e., surface smoothening). The second term couples the thin film evolution to the incident vapor flux and acts as the source term for interfacial growth and surface roughening. For our work, we assume that the velocity is of the form $\mathbf{v} = (0, -v_2)$ with $v_2 > 0$. Additionally, we use $\mathcal{B}_\phi(\phi, \mathbf{x}, t)$ to represent a generic boundary condition.

Transport of the incident vapor flux towards the thin film surface is modeled via the convection-diffusion equation,

$$\begin{aligned} \frac{\partial \rho}{\partial t} - \nabla \cdot [D_\rho \nabla \rho] + \nabla \cdot (\rho \mathbf{v}) + \nabla \phi \cdot (\rho \mathbf{v}) &= 0 && \text{in } \mathcal{D} \times (0, T), \\ \rho(\mathbf{x}, 0) &= \rho_0(\mathbf{x}) && \text{in } \mathcal{D}, \\ \mathcal{B}_\rho(\rho, \mathbf{x}, t) &= 0 && \text{on } \partial \mathcal{D} \times (0, T), \end{aligned} \quad (6.4)$$

where D_ρ is the mass diffusivity of the vapor. The first two terms in (6.4) describe the diffusive and ballistic transport (strength and direction) of the depositing vapor towards the thin film surface. The last term in (6.4) couples thin film growth dynamics to the incident vapor dynamics and acts as a sink that removes vapor that has been converted to solid. The term, $\mathcal{B}_\rho(\rho, \mathbf{x}, t)$, is

a generic boundary condition.

Within the PVD model, there are many parameters, that influence the output of the PDE model. In our work, we restrict ourselves to considering a subset of these parameters, while fixing all remaining parameters to nominal values. We define the varying parameters to be $\boldsymbol{\theta} = [\theta_1, \theta_2]$, where $\theta_i \in [-1, 1]$ for $i = 1, 2$. The physical parameter, $\tilde{\theta}_i$ defined on the interval $[\theta_i^{\min}, \theta_i^{\max}]$, is mapped to its value, θ_i on the interval $[-1, 1]$, by the following transformation:

$$\theta_i = \frac{2\tilde{\theta}_i - \theta_i^{\min} - \theta_i^{\max}}{\theta_i^{\max} - \theta_i^{\min}} \quad i = 1, 2. \quad (6.5)$$

In our work, we define θ_1 to represent the velocity of the vapor to the substrate in the vertical direction and θ_2 to represent the phase fraction. Within the coupled system (6.2) – (6.4), the velocity is controlled by v_2 , where $\mathbf{v} = (0, -v_2)$, and the phase fraction is implicitly controlled by $c_0(\mathbf{x})$ (we discuss how the phase fraction relates to the initial condition of the microstructure in Section 6.1.3).

In this work, we assume that $\boldsymbol{\theta} : \Omega \rightarrow \mathbb{R}^2$ is a random vector, where Ω is the sample space of the probability triple, $(\Omega, \mathcal{B}(\Omega), P)$. Furthermore, assume that each element in $\boldsymbol{\theta}$ is an independent and uniformly distributed random variable such that $\theta_i \sim \mathcal{U}(-1, 1)$ for $i = 1, 2$. Then, the sample space of parameters is defined as $\Omega = [-1, 1]^2$, $\mathcal{B}(\Omega)$ is the Borel sigma-algebra on Ω , and the probability measure P is defined by $P(d\boldsymbol{\theta}) = \frac{1}{4}d\boldsymbol{\theta}$.

Since we are interested in constructing a surrogate model to approximate the relationship between the process parameters, $\boldsymbol{\theta}$, and the random field, $c(\mathbf{x}, t)$ we denote the realizations of the microstructure by $c(\mathbf{x}, t, \boldsymbol{\theta})$.

6.1.2 Statistical representation of the microstructure

So far, we have discussed the PDE model that generates realizations of the microstructure, $c(\mathbf{x}, t, \boldsymbol{\theta})$. The next step in our high-fidelity model is to compute the two-point statistics of $c(\mathbf{x}, t, \boldsymbol{\theta})$. We mention that throughout the rest of the chapter, the terms “autocorrelation” and “cross-correlation” refer to the two-point statistics and not the correlation functions of a random field with respect to the probability space Ω . These terms are common in materials science literature [1, 2, 36, 80, 100].

We are interested in the structural composition of two phases (α and β). To analyze the relationship, we split the information of the microstructure into realizations of two random fields $c_\alpha(\mathbf{x}, t, \boldsymbol{\theta})$ and $c_\beta(\mathbf{x}, t, \boldsymbol{\theta})$. We consider $c_\alpha(\mathbf{x}, t, \boldsymbol{\theta})$ and $c_\beta(\mathbf{x}, t, \boldsymbol{\theta})$ as indicator functions defining the locations of phase α and β , respectively:

$$c_\alpha(\mathbf{x}, t, \boldsymbol{\theta}) = \begin{cases} 1, & c(\mathbf{x}, t, \boldsymbol{\theta}) > 0 \\ 0, & c(\mathbf{x}, t, \boldsymbol{\theta}) \leq 0 \end{cases} \quad \text{and} \quad c_\beta(\mathbf{x}, t, \boldsymbol{\theta}) = \begin{cases} 1, & c(\mathbf{x}, t, \boldsymbol{\theta}) \leq 0 \\ 0, & c(\mathbf{x}, t, \boldsymbol{\theta}) > 0 \end{cases}. \quad (6.6)$$

Then, we compute the two-point statistics for the realizations of the two random fields,

$c_\alpha(\mathbf{x}, t, \boldsymbol{\theta})$ and $c_\beta(\mathbf{x}, t, \boldsymbol{\theta})$. For fixed time, t , and model parameters, $\boldsymbol{\theta}$, the two-point statistic (autocorrelation) with respect to α is defined as [2, 35, 36, 80, 103, 109]

$$R_{c_\alpha c_\alpha}(\mathbf{r}, t, \boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} c_\alpha(\mathbf{x}, t, \boldsymbol{\theta}) c_\alpha(\mathbf{x} + \mathbf{r}, t, \boldsymbol{\theta}) d\mathbf{x}. \quad (6.7)$$

Since (6.7) is a convolutional operator, we use the properties of Fourier transforms to write the autocorrelation [14, 109] as

$$R_{c_\alpha c_\alpha}(\mathbf{r}, t, \boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|} \mathcal{F}^{-1} [\mathcal{F} [c_\alpha(\mathbf{x}, t, \boldsymbol{\theta})] \mathcal{F} [c_\alpha(\mathbf{x}, t, \boldsymbol{\theta})]],$$

where \mathcal{F} and \mathcal{F}^{-1} represent the Fourier and inverse Fourier transforms, respectively.

We mention that other two-point statistics can be defined: $R_{c_\alpha c_\beta}(\mathbf{r}, t, \boldsymbol{\theta})$, $R_{c_\alpha c_\beta}(\mathbf{r}, t, \boldsymbol{\theta})$, and $R_{c_\beta c_\alpha}(\mathbf{r}, t, \boldsymbol{\theta})$; however, it has been shown that due to properties of the Fourier transform and the structure of the random fields, $c_\alpha(\mathbf{x}, t, \boldsymbol{\theta})$ and $c_\beta(\mathbf{x}, t, \boldsymbol{\theta})$, only one of the correlations are independent when considering two phases [34, 79].

6.1.3 Computation of the high-fidelity model

Here, we discuss how each part of the high-fidelity model is computed numerically and then define the high-fidelity model we consider. For simplicity, we consider all numerical and physical parameters as dimensionless quantities.

Phase-field model computation

For the phase-field model, (6.2) – (6.4), we consider the 2D domain illustrated in Figure 6.1. Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ be the domain and $\partial\mathcal{D} = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ be the boundary of the domain. Here, \mathcal{D}_1 is the area where the vapor is being transported (from top to bottom) to the thin film, and \mathcal{D}_2 is the thin film. Since the thin film is assumed to grow in the positive vertical direction, \mathcal{D}_2 increases in the vertical direction as the thin film grows through time. The spatial domain is discretized into a uniform mesh of 256×256 grid points where $\Delta x_1 = \Delta x_2 = 1$, and the temporal domain is discretized with $\Delta t = 10^{-2}$.

We compute the PVD simulation using fixed model parameters with $M_\phi = \omega_\phi = \kappa_\phi = \omega_c = \kappa_c = 1$, $D_\rho = 5 \times 10^{-3}$, and define M_c as in [95, Eq. 8]. The initial conditions are defined in terms of \mathcal{D}_1 and \mathcal{D}_2 in Figure 6.1. Specifically, the simulation begins with \mathcal{D}_2 being two grid-points thick and spans the entire length of Γ_1 . For the initial conditions, we define

$$c_0(\mathbf{x}) \sim \mathcal{N}(\bar{c}, 0.3)|_{(-1,1)}, \quad \phi_0(\mathbf{x}) = \begin{cases} -1, & \mathbf{x} \in \mathcal{D}_1 \\ 1, & \mathbf{x} \in \mathcal{D}_2 \end{cases}, \quad \rho_0(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \mathcal{D}_1 \\ 0, & \mathbf{x} \in \mathcal{D}_2 \end{cases}, \quad (6.8)$$

where $\mathcal{N}(\bar{c}, 0.35)|_{(-1,1)}$ is the normal distribution truncated to the interval $(-1, 1)$. The mean, \bar{c} , of the truncated distribution implicitly defines the phase fraction of phases α and β . For

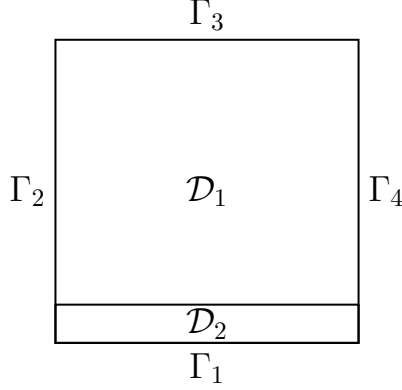


Figure 6.1 Domain used for phase-field model. The horizontal direction is defined as x_1 and the vertical direction is defined as x_2 .

instance, the mean values of 0.00, 0.10, 0.20, 0.30 and 0.45 correspond to generating 50/50, 55/45, 60/40, 65/35, and 73/27 phase fractions. To define the boundary conditions for our PDE model, let L be the distance between Γ_2 and Γ_4 . Then,

$$\begin{aligned}
\mathcal{B}_\phi(\phi, \mathbf{x}, t) &= \begin{cases} \phi(\mathbf{x}, t) - 1, & \mathbf{x} \in \Gamma_1 \\ \phi(\mathbf{x}, t) + 1, & \mathbf{x} \in \Gamma_3 \\ \phi((x_1, x_2), t) - \phi((x_1, x_2 + L), t), & \mathbf{x} \in \Gamma_2 \\ \nabla \phi((x_1, x_2), t) \cdot \mathbf{n} + \nabla \phi((x_1, x_2 + L), t) \cdot \mathbf{n}, & \mathbf{x} \in \Gamma_2 \end{cases}, \\
\mathcal{B}_c(c, \mathbf{x}, t) &= \begin{cases} \nabla c(\mathbf{x}, t) \cdot \mathbf{n}, & \mathbf{x} \in \Gamma_1 \cup \Gamma_3 \\ c((x_1, x_2), t) - c((x_1, x_2 + L), t), & \mathbf{x} \in \Gamma_2 \\ \nabla c((x_1, x_2), t) \cdot \mathbf{n} + \nabla c((x_1, x_2 + L), t) \cdot \mathbf{n}, & \mathbf{x} \in \Gamma_2 \end{cases}, \quad (6.9) \\
\mathcal{B}_\rho(\rho, \mathbf{x}, t) &= \begin{cases} \rho(\mathbf{x}, t), & \mathbf{x} \in \Gamma_1 \\ \rho(\mathbf{x}, t) - 1, & \mathbf{x} \in \Gamma_3 \\ \rho((x_1, x_2), t) - \rho((x_1, x_2 + L), t), & \mathbf{x} \in \Gamma_2 \\ \nabla \rho((x_1, x_2), t) \cdot \mathbf{n} + \nabla \rho((x_1, x_2 + L), t) \cdot \mathbf{n}, & \mathbf{x} \in \Gamma_2 \end{cases}.
\end{aligned}$$

Following [95], the model defined by (6.2) – (6.4), (6.8), and (6.9) is solved numerically using a finite-difference method with second-order central difference stencils for all spatial derivatives. Time integration of the equations is performed using the explicit Euler method. The computational scheme is implemented using Sandia’s in-house multi-physics phase-field modeling code: Mesoscale Multiphysics Phase-Field Simulator (MEMPHIS).

By using a numerical computational scheme to solve the coupled PDEs, we obtain a model to generate a discretization of the phase-fields. The phase-fields in our model are time dependent; however, we are only interested in understanding the compositional structure of the thin film

once it has reached a prescribed height. The time that the thin film reaches the prescribed height t_v can be calculated from the speed at which the vapor is moving toward the top of the thin film. Therefore, for the remainder of this chapter, we use $\mathbf{c}(\boldsymbol{\theta}) \in \mathbb{R}^N$ as the numerical approximation of the compositional phase-field at the time snapshot t_v , $c(\mathbf{x}, t_v, \boldsymbol{\theta})$. An example of four realization of the microstructure $\mathbf{c}(\boldsymbol{\theta})$ are shown in Figure 6.2. In the figure, the yellow color shows the locations where $c_j(\boldsymbol{\theta}) = 1$ and the blue color indicates the locations where $c_j(\boldsymbol{\theta}) = -1$, for $j = 1, \dots, N$. Physically, the two colors represent the locations of each species α and β .

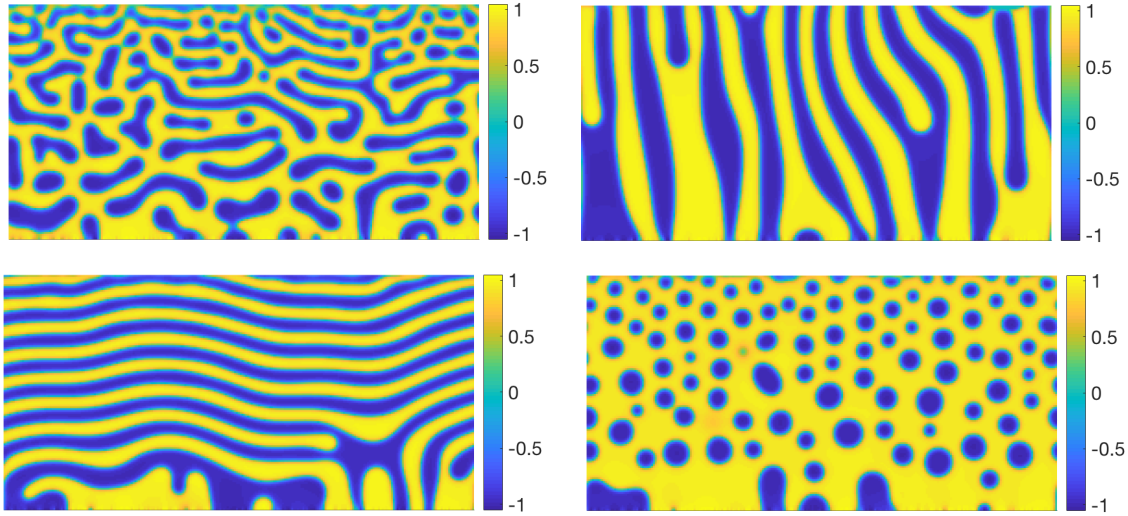


Figure 6.2 Example of microstructures obtained from the PVD process obtained with varying velocity and phase fraction.

Autocorrelation computation

We use the uniform discretization of the microstructure, $\mathbf{c}(\boldsymbol{\theta}) \in \mathbb{R}^N$ to compute the discretized random fields $\mathbf{c}_\alpha(\boldsymbol{\theta}) \in \mathbb{R}^N$ and $\mathbf{c}_\beta(\boldsymbol{\theta}) \in \mathbb{R}^N$ as defined in (6.6). And compute the autocorrelation with respect to α with a two-dimensional Fast Fourier Transform (FFT) [15, 109]. Figure 6.3 illustrates the transformation of a microstructure to the autocorrelation representation, $\mathbf{R}_{c_\alpha c_\alpha}(\boldsymbol{\theta})$. As with the microstructure thin film, we illustrate the autocorrelation $\mathbf{R}_{c_\alpha c_\alpha}(\boldsymbol{\theta})$ as a surface plot, determined according to our numerical mesh.

This completes our discussion of the computation of the high-fidelity model. To summarize, for a realization of the process parameters $\boldsymbol{\theta}$, we discussed the numerical computation of (6.2) – (6.4) and how to use the microstructure to compute realizations of the autocorrelation with

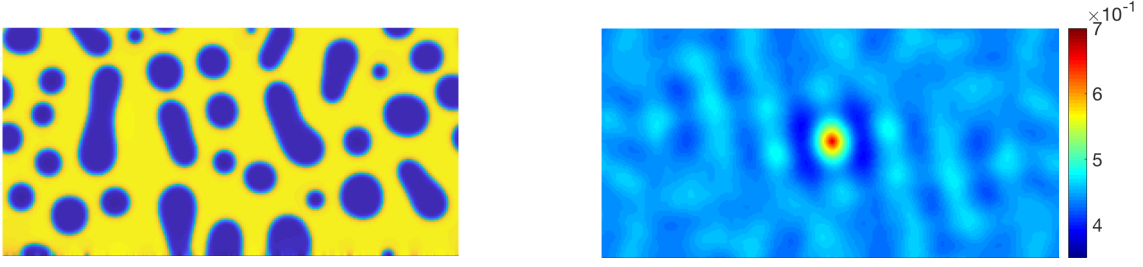


Figure 6.3 Autocorrelation $\mathbf{R}_{c_\alpha c_\alpha}(\boldsymbol{\theta})$ (left) of a microstructure (right), where phase α is defined as the yellow state and β as the blue state.

respect to phase α , $\mathbf{R}_{c_\alpha c_\alpha}(\boldsymbol{\theta})$. We define the high-fidelity model to be

$$\mathbf{g}(\boldsymbol{\theta}) \equiv \mathbf{R}_{c_\alpha c_\alpha}(\boldsymbol{\theta}) : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^N. \quad (6.10)$$

As defined in (6.10), we consider the high-fidelity model as random vector, not a random field. The consequence of treating the random field as a finite dimensional object is that the results we present are mesh-dependent. The next section consists of the construction of the surrogate model for $\mathbf{g}(\boldsymbol{\theta})$.

6.2 Construction of a surrogate model for PVD

The surrogate model $\hat{\mathbf{g}}(\boldsymbol{\theta})$ is constructed using realizations of the high-fidelity model (6.10). We assume that n_d samples are generated from the high-fidelity model such that our data consists of $\{(\boldsymbol{\theta}_m, \mathbf{g}_m)\}_{m=1}^{n_d}$ with $\mathbf{g}_m = \mathbf{g}(\boldsymbol{\theta}_m)$. Using the data, we construct the surrogate model in two steps: (1) utilize PCA to create a k -dimensional approximation of the realizations of the random vector $\mathbf{g}(\boldsymbol{\theta})$, and (2) use PCE to construct a linkage between the process parameters and the projection of the samples to the k -dimensional space [4, 25, 39, 62, 65, 73, 77, 109]. By implementing the methods of PCA and PCE, we obtain a surrogate model $\hat{\mathbf{g}} : [-1, 1]^{n_p} \rightarrow \mathbb{R}^N$ of the form

$$\hat{\mathbf{g}}(\boldsymbol{\theta}) = \hat{\boldsymbol{\mu}} + \sum_{i=1}^k \sum_{j=0}^{\ell_i} a_j^{(i)} \Psi_j(\boldsymbol{\theta}) \mathbf{u}_i. \quad (6.11)$$

The basis vectors, $\mathbf{u}_i \in \mathbb{R}^N$ for $i = 1, \dots, k$ and mean of the random vector, $\hat{\boldsymbol{\mu}} \in \mathbb{R}^N$, are determined from the PCA algorithm, while the theory of PCE defines the family of orthogonal polynomials, Ψ_j for $j = 0, \dots, \ell_i$, and coefficients, $a_j^{(i)}$ for $i = 1, \dots, k$ and $j = 0, \dots, \ell_i$. A summary of the steps used to construct the surrogate model is shown in Figure 6.4. In the remainder of this section, we describe the methods of PCA and PCE and how they are applied to our PVD simulation data $\{(\boldsymbol{\theta}_m, \mathbf{g}_m)\}_{m=1}^{n_d}$.

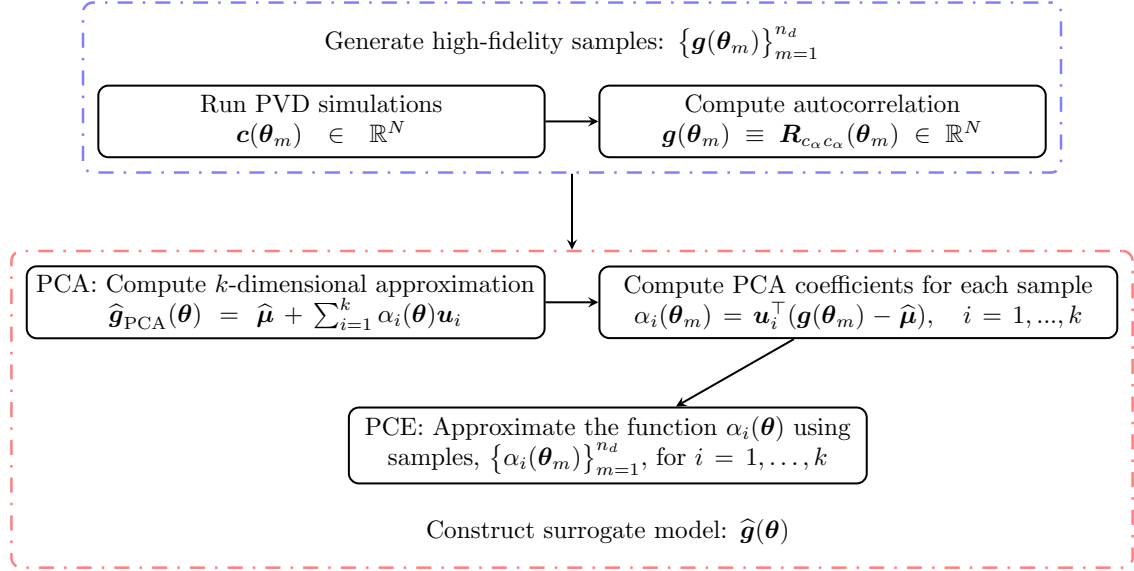


Figure 6.4 Outline of approach to create a surrogate model for PVD, $\hat{g}(\theta)$.

6.2.1 PCA: Compute a k -dimensional approximation

PCA [32, 50, 51, 91] is a common tool for dimensionality reduction of a dataset. Assume that each datapoint in the dataset is a realization of a random variable, $\mathbf{x} \in \mathbb{R}$. Then, PCA computes a set of vectors, $\mathbf{u}_1, \dots, \mathbf{u}_k$, such that the variance of the projection of \mathbf{x} to the space spanned by \mathbf{u}_i is maximized. The choice of the k basis vectors is dictated by the PCA algorithm and ensures that the reduced space captures most of the variance of the process. In this portion of the chapter, we discuss the mathematical formulation of PCA and its application to the problem we consider.

Since the derivation of PCA is standard, we discuss the intuition behind PCA by the mathematical formulation when $k = 1$ and refer the reader to [32, 51] for a complete justification of PCA. Suppose $\mathbf{x} \in \mathbb{R}^n$ is a random vector with mean $\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}]$ and covariance $\boldsymbol{\Sigma} = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top]$. We seek a vector \mathbf{u}_1 such that

$$\mathbf{u}_1 = \arg \max_{\|\mathbf{u}\|_2=1} \text{var}(\mathbf{u}^\top \mathbf{x}) = \arg \max_{\|\mathbf{u}\|_2=1} \mathbf{u}^\top \boldsymbol{\Sigma} \mathbf{u}, \quad (6.12)$$

i.e., \mathbf{u}_1 is the best direction to compress the information contained in \mathbf{x} so that the variance of the projection to the space spanned by \mathbf{u}_1 is maximized. The solution to this problem is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{\Sigma}$. As justified in [32], the subsequent vectors, $\mathbf{u}_2, \dots, \mathbf{u}_n$, are the eigenvectors of $\boldsymbol{\Sigma}$ ordered according to the magnitude of their corresponding eigenvalues.

To approximate the realizations of \mathbf{x} , we project the realizations to the space spanned by the eigenvectors of $\boldsymbol{\Sigma}$ corresponding to the k -largest eigenvalues. The error of $\hat{\mathbf{x}}$, obtained from

projecting \mathbf{x} to the k -dimensional subspace, is

$$\mathbb{E} [\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2] = \sum_{i=k+1}^N \lambda_i,$$

the sum of the $N - k$ smallest eigenvalues of $\mathbf{\Sigma}$. If the magnitude of the $N - k$ smallest eigenvalues of $\mathbf{\Sigma}$ is small, then $\hat{\mathbf{x}}$ is a good approximation to \mathbf{x} in the mean-square sense. A common indicator used to determine a value of k is the proportion of total variance [51]:

$$\xi(i) = \frac{\lambda_i}{\sum_{j=1}^N \lambda_j}. \quad (6.13)$$

The sum of $\xi(i)$ for $i = 1, \dots, k$ indicates the proportion of total variance explained by the first $\mathbf{u}_1, \dots, \mathbf{u}_k$ eigenvectors. We pick the smallest k such that $\sum_{i=1}^k \xi(i)$ surpasses a user-specified threshold.

Dimensionality reduction using our dataset

In our problem, we wish to obtain a k -dimensional approximation of the random vector $\mathbf{g}(\boldsymbol{\theta})$ using PCA. However, the information on the exact mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ of the random vector is not available. Instead, the samples $\{(\boldsymbol{\theta}_m, \mathbf{g}_m)\}_{m=1}^{n_d}$ are used to estimate the mean and covariance of the random vector. The mean and covariance estimates are

$$\hat{\boldsymbol{\mu}} = \frac{1}{n_d} \sum_{i=1}^{n_d} \mathbf{g}_i \quad (6.14)$$

and

$$\hat{\mathbf{\Sigma}} = \frac{1}{n_d - 1} (\mathbf{G} - \hat{\boldsymbol{\mu}} \mathbf{1}^\top) (\mathbf{G} - \hat{\boldsymbol{\mu}} \mathbf{1}^\top)^\top, \quad (6.15)$$

where $\mathbf{1}$ is the vector of all ones and $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_{n_d}] \in \mathbb{R}^{N \times n_d}$.

The PCA algorithm in Algorithm 7 consists of computing the eigenvalue decomposition of (6.15) and returning $\mathbf{U}_k = [\mathbf{u}_1, \dots, \mathbf{u}_k]$, the eigenvectors which correspond to the k largest eigenvalues.

Algorithm 7 Computing a k -dimensional subspace using PCA.

Input: Size of desired subspace k , samples from high-fidelity model \mathbf{g}_i for $i = 1, \dots, n_d$.

Output: k -dimensional basis \mathbf{u}_j for $j = 1, \dots, k$.

- 1: Use (6.14) to compute the sample average $\hat{\boldsymbol{\mu}}$ of \mathbf{g}_i for $i = 1, \dots, n_d$.
 - 2: Compute the sample covariance matrix $\hat{\mathbf{\Sigma}}$ by using (6.15).
 - 3: Compute eigendecomposition $[\mathbf{U}_k, \mathbf{\Lambda}_k]$ of the sample covariance $\hat{\mathbf{\Sigma}}$, corresponding to the k largest eigenvalues.
-

The approximation of $\mathbf{g}(\boldsymbol{\theta})$ in the k -dimensional subspace, $\text{span}\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$, is defined as

$$\mathbf{g}(\boldsymbol{\theta}) \approx \hat{\mathbf{g}}_{\text{PCA}}(\boldsymbol{\theta}) = \hat{\boldsymbol{\mu}} + \sum_{i=1}^k \alpha_i(\boldsymbol{\theta}) \mathbf{u}_i, \quad (6.16)$$

where

$$\alpha_i(\boldsymbol{\theta}) = \mathbf{u}_i^\top (\mathbf{g}(\boldsymbol{\theta}) - \hat{\boldsymbol{\mu}}), \quad i = 1, \dots, k.$$

We mention that the projection of the data to the space spanned by the k eigenvectors of $\hat{\boldsymbol{\Sigma}}$ depend on the process parameters $\boldsymbol{\theta}$ while the eigenvectors are deterministic.

Through PCA, we have approximated the high-fidelity model and the dependency of $\boldsymbol{\theta}$ on $\mathbf{g}(\boldsymbol{\theta})$ by reducing the number of coefficients from N to k . In our application, $k \ll N$ because the eigenvalues of the sample covariance matrix are observed to decay rapidly. The next section is dedicated to discussing how the process parameters are linked to the coefficients of the approximated representation, $\boldsymbol{\alpha}(\boldsymbol{\theta}) = [\alpha_1(\boldsymbol{\theta}), \dots, \alpha_k(\boldsymbol{\theta})]^\top$.

6.2.2 PCE: Linking PCA coefficients with process parameters

To complete the construction of the surrogate model for PVD, we link the random inputs $\boldsymbol{\theta}$ to the coefficients of the PCA approximation, $\boldsymbol{\alpha}(\boldsymbol{\theta})$. We use PCE to accomplish this.

PCE [25, 30, 37, 57, 71, 78, 94, 108] approximates a random variable by a finite linear combination of orthogonal multivariate polynomials. In our problem, we wish to approximate the coordinates $\alpha_i(\boldsymbol{\theta})$ for $i = 1, \dots, k$ in (6.16). Using PCE, the resulting approximation is

$$\alpha_i(\boldsymbol{\theta}) \approx \hat{\alpha}_i(\boldsymbol{\theta}) = \sum_{j=0}^{\ell_i} a_j^{(i)} \Psi_j(\boldsymbol{\theta}), \quad (6.17)$$

for some set of orthogonal polynomial basis functions $\{\Psi_j\}_{j=0}^{\ell_i}$ and deterministic coefficients $\{a_j^{(i)}\}_{j=0}^{\ell_i}$. Next, we discuss the choice of orthogonal polynomial family, the number of terms in the expansion ℓ_i and the determination of the PCA coefficients.

Choosing orthogonal multivariate polynomials

The choice of multivariate orthogonal polynomials depend on the probability density function of $\boldsymbol{\theta}$. For many of the finite dimensional distributions commonly chosen to represent a random variable, the associated probability function corresponds to the weighting function of a family of orthogonal polynomials [30, 37, 57, 94, 108]. Recall that, in our problem, the components of the random vector $\boldsymbol{\theta}$ are assumed to be mutually independent random variables with uniform distributions. The associated probability density function of the probability space we work in corresponds to the weighting function of the two-dimensional Legendre polynomials defined on the domain $[-1, 1]^2$ [57, 94, 108].

The two-dimensional basis is formed by tensorizing the one-dimensional Legendre polynomials

of each parameter, θ_i , for $i = 1, 2$, and is denoted by $\{\Psi_j(\boldsymbol{\theta})\}_{j=0}^{\ell_i}$. The number of polynomials we use in our expansion (6.17) is determined by the total truncation level. We first define a multi-index, $\mathbf{l} = (l_1, l_2)$, where the order of the one-dimensional Legendre polynomial for θ_i is indicated by $l_i \in \mathbb{Z}_{\geq 0}$. A bivariate Legendre basis of total truncation level L consists of all polynomials such that $l_1 + l_2 \leq L$, where the total number of polynomials we use in the expansion is computed by the formula $\ell + 1 = \frac{(L+n_p)!}{L!n_p!}$. Since we compute a PCE for each of the coefficients of the PCA approximation, $\alpha_i(\boldsymbol{\theta})$, we allow each PCE to have a different total truncation level L and, therefore, denote the total number of polynomials used in each expansion as ℓ_i , for $i = 1, \dots, k$. For more information and examples of the construction of the multi-dimensional basis, see [94].

Computing the deterministic coefficients

To complete the surrogate model as described, the remaining task is to compute the coefficients of the expansion (6.17). We accomplish this by first projecting the data, $\{(\boldsymbol{\theta}_m, \mathbf{g}_m)\}_{m=1}^{n_d}$, to the k -dimensional subspace spanned by the vectors \mathbf{u}_i for $i = 1, \dots, k$, transforming the data points to $\{(\boldsymbol{\theta}_m, \boldsymbol{\alpha}(\boldsymbol{\theta}_m))\}_{m=1}^{n_d}$. Then define k vectors, $\boldsymbol{\alpha}_i \in \mathbb{R}^{n_d \times 1}$, $i = 1, \dots, k$, such that each entry in the vector corresponds to a different sample, $\boldsymbol{\theta}_m$, $m = 1, \dots, n_d$, i.e.,

$$\boldsymbol{\alpha}_i = \begin{bmatrix} \alpha_i(\boldsymbol{\theta}_1) & \alpha_i(\boldsymbol{\theta}_2) & \cdots & \alpha_i(\boldsymbol{\theta}_{n_d}) \end{bmatrix}^\top.$$

We also define $\mathbf{A} \in \mathbb{R}^{n_d \times \ell_i}$ to be a matrix with each row corresponding to the set of basis vectors evaluated at a sampled process parameter, $\boldsymbol{\theta}_m$, such that

$$\boldsymbol{\Psi} = \begin{bmatrix} \Psi_0(\boldsymbol{\theta}_1) & \Psi_1(\boldsymbol{\theta}_1) & \cdots & \Psi_{\ell_i}(\boldsymbol{\theta}_1) \\ \Psi_0(\boldsymbol{\theta}_2) & \Psi_1(\boldsymbol{\theta}_2) & \cdots & \Psi_{\ell_i}(\boldsymbol{\theta}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\boldsymbol{\theta}_{n_d}) & \Psi_1(\boldsymbol{\theta}_{n_d}) & \cdots & \Psi_{\ell_i}(\boldsymbol{\theta}_{n_d}) \end{bmatrix}.$$

Then for a fixed i , the coefficients $a_j^{(i)}$, $j = 1, \dots, \ell_i$, are computed by solving the least squares problem

$$\mathbf{a}^{(i)} = \arg \min_{\mathbf{a} \in \mathbb{R}^{\ell_i}} \|\boldsymbol{\alpha}_i - \boldsymbol{\Psi} \mathbf{a}\|_2^2, \quad (6.18)$$

where

$$\mathbf{a}^{(i)} = \begin{bmatrix} a_1^{(i)} & a_2^{(i)} & \cdots & a_{\ell_i}^{(i)} \end{bmatrix}^\top.$$

We compute the solution to the k least squares problems in (6.18) by solving the normal equations.

Using the PCEs for the coefficients, $\alpha_i(\boldsymbol{\theta})$, we substitute (6.17) in (6.16). The sequential computation of the deterministic basis function \mathbf{u}_i for $i = 1, \dots, k$ using PCA and the deterministic coefficients $a_j^{(i)}$ complete the discussion of the surrogate model, (6.11), for $\mathbf{g}(\boldsymbol{\theta})$. We mention

that there are two major sources of error in the construction of the surrogate model. The first source of error is projecting $\mathbf{g}(\boldsymbol{\theta})$ into a k -dimensional subspace using PCA. The second source of error comes from using PCE to approximate the relationship between the process parameters and the coefficients of the PCA expansion. Specifically, there is error associated with choosing a total truncation level for the bivariate Legendre polynomial basis and solving the least squares problem in (6.18).

6.3 Numerical results

In this section, we compute the surrogate model (6.11) and investigate its accuracy. We begin with a description of how the data was collected. Then we discuss the computation of the k -dimensional subspace obtained through PCA and the error obtained from our choice of k . Finally, we link the process parameters and data to complete the construction of the surrogate model. Multiple error metrics are presented to illustrate that the surrogate model yields accurate predictions of $\mathbf{g}(\boldsymbol{\theta})$.

6.3.1 High-fidelity samples

Recall we consider two parameters θ_1 and θ_2 , representing the velocity that the vapor moves (in the vertical direction) towards the thin film and the phase fraction of the alloy α , respectively. We model these random inputs as independent uniform random variables. The physical range of velocity and phase fraction is $[0.08, 2.0]$ and $[0, 0.5]$ respectively.

To sample the high-fidelity model, we create a tensorized grid of the physical process parameters. Specifically, we choose the values of velocity to be equally spaced intervals of 0.03 from 0.08 to 2.0 and the values of phase fraction are chosen to be 0.1, 0.2, 0.3, 0.4, 0.5. These physical parameters are mapped to the interval $[-1, 1]$ by the linear transformation in (6.5) to obtain 325 samples of $\boldsymbol{\theta}$. Using these samples of $\boldsymbol{\theta}$, the high-fidelity model, described in Section 6.1, is evaluated 325 times. The model evaluations are then randomly split into a training set, consisting of 90% of the data, and a testing set, containing the remaining 10% of the data.

6.3.2 Accuracy of PCA

We consider a physical domain of size 256×256 dimensionless units where the numerical grid is 256×256 pixels and run the simulation until the thin film has grown to be 128 pixels. Each high-fidelity sample is then defined as $\mathbf{g}(\boldsymbol{\theta}) \in \mathbb{R}^{32768}$ and our surrogate model is constructed to approximate these 32768 values.

PCA as described in Algorithm 7 is performed using the training dataset to estimate the random variable $\mathbf{g}(\boldsymbol{\theta})$. Figure 6.5 shows the variance λ_i for $i = 1, \dots, 325$ and the total explained variability defined as $\xi_{\text{tot}}(i) = \sum_{k=1}^i \xi(k)$, where $\xi(i)$ is defined in (6.13). The figure illustrates 8 basis vectors provide an accurate low-rank approximation of the sample covariance matrix,

explaining 97.75% of the variability in the estimated covariance matrix (6.15). Letting $k = 8$, we reduce the number of coefficients which depend on the process parameters θ from 32768 to 8 as explained in (6.17). Figure 6.6 depicts the sample average $\hat{\mu}$ and the first five basis vectors u_i for $i = 1, \dots, 5$ that were computed as using Algorithm 7.

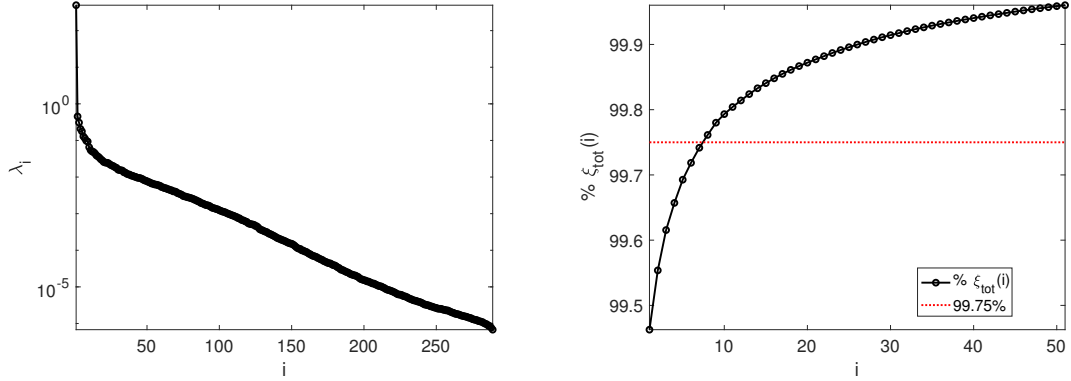


Figure 6.5 Performing PCA on the high-fidelity training samples. The computed variance of $g(\theta)$ (left) and the cumulative proportion of total variance (percentage) for the first 50 principal directions (right).

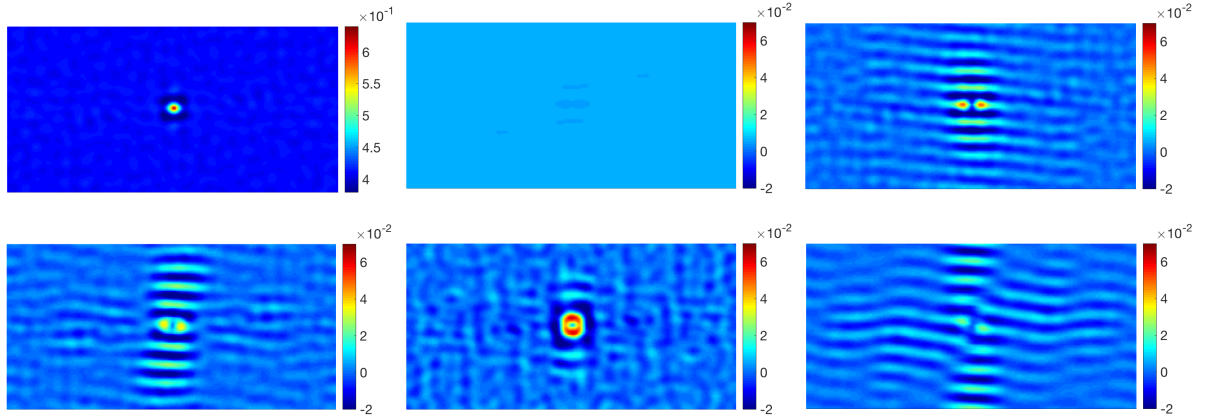


Figure 6.6 The resulting average, $\hat{\mu}$, (top, left) and the first 5 basis elements, u_1, \dots, u_5 , computed from PCA on the high-fidelity training set.

After using the training set to compute u_i for $i = 1, \dots, 8$, we project each realization of the dataset into the k -dimensional space and obtain the corresponding coefficients $\alpha(\theta) \in \mathbb{R}^8$. The error in the approximation of $g(\theta)$ using the PCA expansion in (6.16) is evaluated by the

following error metric

$$\text{err}_1(\boldsymbol{\theta}) = \frac{\|\mathbf{g}(\boldsymbol{\theta}) - \widehat{\mathbf{g}}_{\text{PCA}}(\boldsymbol{\theta})\|_2}{\|\mathbf{g}(\boldsymbol{\theta})\|_2}$$

Table 6.1 shows the minimum, maximum, mean, and standard deviation of the error associated with $\text{err}_1(\boldsymbol{\theta})$ for both the training and testing set. We see that on average the error of using this smaller space as a representation of the data results in less than a 2% error in the output of the high-fidelity model.

Table 6.1 The error associated with using 8 basis functions to approximate the high-fidelity samples of the training and testing set. By training (testing) error, we mean $\text{err}_1(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is in the training (testing) set.

$\text{err}_1(\boldsymbol{\theta})$	min	max	mean	standard deviation
Training error	0.0030	0.0690	0.0146	0.0123
Testing error	0.0030	0.0441	0.0123	0.0091

6.3.3 Linking process parameters to PCA expansion

For $i = 1, \dots, 8$, we approximate the relationship between the process parameters and $\alpha_i(\boldsymbol{\theta})$ by solving the linear least squares problem in (6.18). To determine the total truncation level, L , for each PCA coefficient, $\alpha_i(\boldsymbol{\theta})$, we performed 10-fold cross-validation on the training dataset and chose the total truncation level that produced the smallest average error in the validation set. For every $\alpha_i(\boldsymbol{\theta})$, $i = 1, \dots, 8$, the choice of $\ell_i + 1$ is shown in Table 6.2.

Table 6.2 Size of polynomial basis $\ell_i + 1$ for each coefficient $\alpha_i(\boldsymbol{\theta})$ determined by performing 10-fold cross validation on the training set and choosing the truncation level which produced the smallest average error in the validation set.

PCA coefficient, $\alpha_i(\boldsymbol{\theta})$	$\ell_i + 1$
$\alpha_1(\boldsymbol{\theta})$	15
$\alpha_2(\boldsymbol{\theta})$	6
$\alpha_3(\boldsymbol{\theta})$	15
$\alpha_4(\boldsymbol{\theta})$	10
$\alpha_5(\boldsymbol{\theta})$	1
$\alpha_6(\boldsymbol{\theta})$	15
$\alpha_7(\boldsymbol{\theta})$	1
$\alpha_8(\boldsymbol{\theta})$	3

Using these coefficients, we illustrate the success of the surrogate model by providing an example of the prediction from the set of testing parameters in Figure 6.7. The high-fidelity outputs we consider is shown in left panel in Figure 6.7 and the surrogate's prediction is the right panel. In the middle panel of Figure 6.7, a relative point-wise error is illustrated. This error is defined as a vector, $\delta(\theta)$, where each entry is computed by

$$\delta_i(\theta) = \frac{|g_i(\theta) - \hat{g}_i(\theta)|}{g_i(\theta)}, \quad i = 1, \dots, N. \quad (6.19)$$

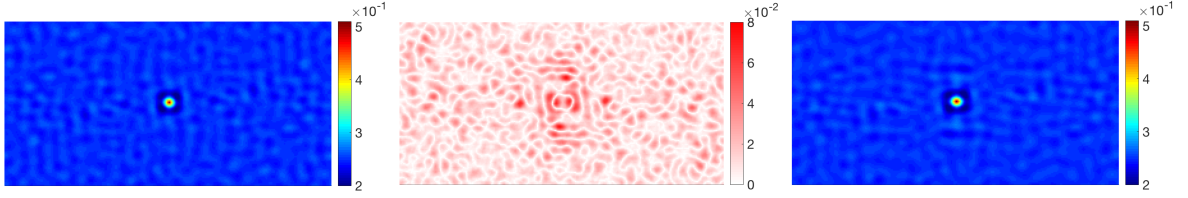


Figure 6.7 Example prediction using the surrogate model: high-fidelity evaluation $\mathbf{g}(\theta)$ (left) and surrogate evaluation $\hat{\mathbf{g}}(\theta)$ (right). The relative point-wise error, $\delta(\theta)$, as calculated by (6.19) is illustrated in (middle).

To summarize the error associated with using PCE to approximate the relationship between θ and $\alpha(\theta)$, we define the following error metric,

$$\text{err}_2(\theta) = \frac{\|\alpha(\theta) - \hat{\alpha}(\theta)\|_2}{\|\alpha(\theta)\|_2}.$$

Table 6.3 illustrates the error for the training and testing sets.

We also consider the total error of the surrogate model defined as

$$\text{err}_3(\theta) = \frac{\|\mathbf{g}(\theta) - \hat{\mathbf{g}}(\theta)\|_2}{\|\mathbf{g}(\theta)\|_2}.$$

The total surrogate error on both the testing and training set is summarized in Table 6.3. On average the surrogate's error is about 2% for both the training and testing sets. This suggest our surrogate estimates the high-fidelity model accurately.

It is clear that the the surrogate model's accuracy can be improved by decreasing the error in the PCE step. Higher degree polynomials may be required for to obtain more accurate approximations of the relationship between the process parameters and the PCA space. Other possible methods that could describe this relationship include using a Gaussian process [52] or machine learning techniques (deep neural networks or convolutional neural networks) to represent the output of the high-fidelity model [19, 110].

Table 6.3 The error in approximating $\alpha(\boldsymbol{\theta})$ using PCE, $\text{err}_2(\boldsymbol{\theta})$, and the error incurred by using the surrogate model to predict $\mathbf{g}(\boldsymbol{\theta})$, $\text{err}_3(\boldsymbol{\theta})$. The training (testing) error is computed by evaluating $\text{err}_2(\boldsymbol{\theta})$ and $\text{err}_3(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is in the training (testing) set.

$\text{err}_2(\boldsymbol{\theta})$	min	max	mean	standard deviation
Training error	0.0024	3.5730	0.1034	0.2762
Testing error	0.0034	0.5734	0.0865	0.1399
$\text{err}_3(\boldsymbol{\theta})$				
Training error	0.0032	0.1197	0.0206	0.0223
Testing error	0.0035	0.0771	0.0154	0.0139

6.4 Conclusion

We have implemented an efficient method to create a surrogate model for the autocorrelation of a realization from PVD. This random field is of interest as it can aid experimentalist in determining the process parameters which yield microstructures with favorable properties. The construction of surrogate model only requires samples from the high-fidelity model; therefore, there is a high initial computational cost to construct the model. Using the samples, we use PCA to find a k -dimensional subspace to approximate the output of the high-fidelity model. Then the relationship between the process parameters and the coefficients in the PCA expansion are approximated using PCEs. As illustrated by the error analysis, the combination of PCA and PCE produces an accurate surrogate model.

Open directions that can be pursued include computing a surrogate model for the time dependent process of PVD (not just one snapshot) and one which accounts for three-dimensional thin films. The resulting surrogate model from these alterations would better represent the physical process, but may require implementing more efficient methods to overcome the added computational cost of the time-dependent process and a three-dimensional thin film. In addition, creating a surrogate model that accounts for more process parameters would allow sensitivity analysis to be performed. A benefit of using PCE is that the coefficients in the linear expansion can be used to estimate statistical properties of the random variable along with having a systematic way to perform sensitivity analysis on the inputs [94, 98]. If we construct a surrogate model that incorporates all of the model parameters pertinent to PVD, then sensitivity analysis could prove useful in guiding resources to ensure important parameters are accurately chosen.

CHAPTER

7

CONCLUSION

In this thesis, two computationally intensive problems are considered: design of linear inverse problems and construction of a surrogate model. With both of these problems, the physical phenomena are modeled by partial differential equations (PDEs) and consequently result in a high computational cost for each model evaluation. To solve each problem we considered, multiple model evaluations are necessary. Therefore, we developed and utilized efficient techniques to address the computational challenges associated with each problem.

Chapter 3 – 5 focused on methods that improve upon existing computational techniques for solving optimal experimental design (OED) problems governed by infinite-dimensional Bayesian linear inverse problems. In Chapter 3, we developed randomized estimators to efficiently compute the A-optimal criterion and its gradient. The randomized estimators were constructed to exploit the low-rank structure in the governing inverse problem. We also presented a novel modified A-optimal design criterion, and developed efficient randomized estimators to estimate the criterion and its gradient. We provided discussions of computational cost and implementation issues, and theoretical error analysis for these estimators. In Chapter 4, we presented a new optimization framework for sensor placement based on reweighted ℓ_1 minimization. The algorithm was further accelerated by the randomized estimators developed in Chapter 3. Since the A-optimal criterion is a convex function, applying the penalty method to the OED problem resulted in a sequence of convex optimization problems. The effectiveness of this, along with the randomized estimators from Chapter 3, was demonstrated on a contaminant source identification problem. In Chapter 5, we developed computational methods for OED problems where an adjoint PDE solver is not available. These estimators required an added assumption that the prior covariance operator is approximately low-rank. Using this approach, the randomized estimators for the A-optimal and

D-optimal objective functions and their derivatives can be evaluated without adjoint capabilities. We also illustrated the efficiency and effectiveness of these methods numerically.

In Chapter 6, we developed a surrogate model using non-intrusive techniques that approximated the relationship between the process parameters of physical vapor deposition (PVD) and a function of its output. We used principal component analysis (PCA) to reduce the dimensionality of the output of our high-fidelity model through exploitation of the low rank structure of the output sample covariance matrix. Then, we approximated the relationship between the low-rank approximation and the process parameters through polynomial chaos expansion (PCE). The error study provided in that chapter suggests that the computed surrogate model for a subset of the process parameters is sufficiently accurate.

Potential future directions:

Future directions for furthering our methods in Chapter 3 – Chapter 5 include incorporating a mechanism within the optimization algorithm to adaptively determine the target rank k , comparing the accuracy of our estimators with those of other low-rank approximations techniques, extending the randomized estimators to approximate trace of matrix functions, and validating our numerical results using real-world data. Another track of investigation would be to incorporate the randomized estimators in optimization problems of nonlinear inverse problems.

In terms of the surrogate model for PVD, open directions include: developing a surrogate for a more physically realistic dataset, i.e., one that accounts for three-dimensional thin film growth as a result of PVD and including more process parameters. Once the surrogate model is constructed, sensitivity analysis can be performed efficiently to determine the importance of process parameters on the output of PVD.

BIBLIOGRAPHY

- [1] Adams, B. L., Gao, X. C. & Kalidindi, S. R. “Finite approximations to the second-order properties closure in single phase polycrystals”. *Acta Materialia* **53**.13 (2005), pp. 3563–3577.
- [2] Adams, B. L., Kalidindi, S. R. & Fullwood, D. T. *Microstructure sensitive design for performance optimization*. 1st. Newton, MA, USA: Butterworth-Heinemann, 2012.
- [3] Alexanderian, A., Gloor, P. J. & Ghattas, O. “On Bayesian A-and D-optimal experimental designs in infinite dimensions”. *Bayesian Analysis* **11**.3 (2016), pp. 671–695.
- [4] Alexanderian, A., Gremaud, P. A. & Smith, R. C. “Variance-based sensitivity analysis for time-dependent processes”. *Reliability Engineering & System Safety* **196** (2020).
- [5] Alexanderian, A., Petra, N., Stadler, G. & Ghattas, O. “A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized ℓ_0 -sparsification”. *SIAM Journal on Scientific Computing* **36**.5 (2014), A2122–A2148.
- [6] Alexanderian, A. & Saibaba, A. K. “Efficient D-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems”. *SIAM Journal on Scientific Computing* **40**.5 (2018), A2956–A2985.
- [7] Aliofkhazraei, M. & Ali, N. “7.04 - PVD technology in fabrication of micro- and nanostructured coatings”. *Comprehensive materials processing*. Ed. by Hashmi, S., Batalha, G. F., Tyne, C. J. V. & Yilbas, B. Oxford: Elsevier, 2014, pp. 49–84.
- [8] Atkinson, A., Donev, A. & Tobias, R. *Optimum experimental designs, with SAS*. Vol. 34. Oxford University Press, 2007.
- [9] Attia, A., Alexanderian, A. & Saibaba, A. K. “Goal-oriented optimal design of experiments for large-scale Bayesian linear inverse problems”. *Inverse Problems* **34**.9 (2018).
- [10] Bapat, R. *Linear algebra and linear models*. Universitext. Springer, London, 2012.
- [11] Beck, J., Dia, B. M., Espath, L. F., Long, Q. & Tempone, R. “Fast Bayesian experimental design: Laplace-based importance sampling for the expected information gain”. *Computer Methods in Applied Mechanics and Engineering* **334** (2018), pp. 523–553.
- [12] Bhatia, R. *Matrix analysis*. Vol. 169. Graduate Texts in Mathematics. Springer-Verlag, New York, 1997.
- [13] Boyd, S. & Vandenberghe, L. *Convex Optimization*. Cambridge: Cambridge University Press, 2004.
- [14] Bracewell, R. *The Fourier transform and its applications*. 3rd ed. Electrical engineering series. New York : McGraw Hill, 2000.

- [15] Brigham, E. O. *The fast Fourier transform*. Prentice-Hall signal processing series. Englewood Cliffs, N.J. : Prentice-Hall, 1974.
- [16] Bui-Thanh, T., Ghattas, O., Martin, J. & Stadler, G. “A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion”. *SIAM Journal on Scientific Computing* **35.6** (2013), A2494–A2523.
- [17] Cahn, J. W. & Hilliard, J. E. “Free energy of a nonuniform system. I. Interfacial free energy”. *Journal of Chemical Physics* **28** (1958), pp. 258–267.
- [18] Candès, E. J., Wakin, M. B. & Boyd, S. P. “Enhancing Sparsity by Reweighted ℓ_1 Minimization”. *Journal of Fourier Analysis and Applications* **14** (2008), pp. 877–905.
- [19] Cecen, A., Dai, H., Yabansu, Y. C., Kalidindi, S. R. & Song, L. “Material structure-property linkages using three-dimensional convolutional neural networks”. *Acta Materialia* **146** (2018), pp. 76–84.
- [20] Chaloner, K. & Verdinelli, I. “Bayesian experimental design: A review”. *Statistical Science* **10.3** (1995), pp. 273–304.
- [21] Chen, A., Bi, Z., Tsai, C., Lee, J., Su, Q., Zhang, X., Jia, Q., MacManus-Driscoll, J. L. & Wang, H. “Tunable low-field magnetoresistance in $(\text{La}_{0.7}\text{Sr}_{0.3}\text{MnO}_3)_{0.5}:(\text{ZnO})_{0.5}$ self-assembled vertically aligned nanocomposite thin films”. *Advanced Functional Materials* **21.13** (2011), pp. 2423–2429.
- [22] Chen, L.-Q. “Phase-field models for microstructure evolution”. *Annual Review of Materials Research* **32.1** (2002), pp. 113–140.
- [23] Cho, J., Terry, S., LeSar, R. & Levi, C. “A kinetic Monte Carlo simulation of film growth by physical vapor deposition on rotating substrates”. *Materials Science and Engineering: A* **391.1** (2005), pp. 390–401.
- [24] Chung, M. & Haber, E. “Experimental design for biological systems”. *SIAM Journal on Control and Optimization* **50.1** (2012), pp. 471–489.
- [25] Crestaux, T., Maître, O. L. & Martinez, J.-M. “Polynomial chaos expansion for sensitivity analysis”. *Reliability Engineering & System Safety* **94.7** (2009). Special Issue on Sensitivity Analysis, pp. 1161–1172.
- [26] Crestel, B., Alexanderian, A., Stadler, G. & Ghattas, O. “A-optimal encoding weights for nonlinear inverse problems, with application to the Helmholtz inverse problem”. *Inverse Problems* **33.7** (2017).
- [27] Cui, Y., Derby, B., Li, N. & Misra, A. “Design of bicontinuous metallic nanocomposites for high-strength and plasticity”. *Materials & Design* **166** (2019).
- [28] Da Prato, G. *An introduction to infinite-dimensional analysis*. Springer-Verlag, Berlin, Heidelberg, 2006.

- [29] Dashti, M. & Stuart, A. M. “The Bayesian approach to inverse problems”. *Handbook of Uncertainty Quantification*. Ed. by Ghanem, R., Higdon, D. & Owhadi, H. Springer, Cham, 2017.
- [30] Debusschere, B., Najm, H. N., Pébay, P. P., Knio, O. M., Ghanem, R. G. & Maître, O. P. L. “Numerical challenges in the use of polynomial chaos representations for stochastic processes”. *SIAM Journal on Scientific Computing* **26.2** (2004), pp. 698–719.
- [31] Dong, L., Smith, R. W. & Srolovitz, D. J. “A two-dimensional molecular dynamics simulation of thin film growth by oblique deposition”. *Journal of Applied Physics* **80.10** (1996), pp. 5682–5690.
- [32] Embree, M. “The singular value decomposition”. Virginia Tech CMDA:3606 lecture notes. 2019.
- [33] Fayomi, O. S. I., Akande, I. G., Abioye, O. P. & Fakehinde, O. B. “New trend in thin film composite coating deposition: A mini review”. *Procedia Manufacturing* **35** (2019). The 2nd International Conference on Sustainable Materials Processing and Manufacturing, SMPM 2019, 8-10 March 2019, Sun City, South Africa, pp. 1007–1012.
- [34] Frisch, H. L. & Stillinger, F. H. “Contribution to the statistical geometric basis of radiation scattering”. *Journal of Chemical Physics* **38** (1963), pp. 2200–2207.
- [35] Fullwood, D. T., Kalidindi, S. R. & Adams, B. L. “Second-order microstructure sensitive design using 2-point spatial correlations”. 2009, pp. 177–188.
- [36] Fullwood, D. T., Niezgoda, S. R. & Kalidindi, S. R. “Microstructure reconstructions from 2-point statistics using phase-recovery algorithms”. *Acta Materialia* **56.5** (2008), pp. 942–948.
- [37] Gerritsma, M., Vos, P. & Steen, J.-B. van der. “Time-dependent polynomial chaos”. *AIP Conference Proceedings* **1048.1** (2008), pp. 221–244.
- [38] Gohberg, I., Goldberg, S. & Kaashoek, M. A. *Basic classes of linear operators*. Birkhäuser, Basel, 2003.
- [39] Gupta, A., Cecen, A., Goyal, S., Singh, A. K. & Kalidindi, S. R. “Structure-property linkages using a data science approach: Application to a non-metallic inclusion/steel composite system”. *Acta Materialia* **91.1** (2015), pp. 239–254.
- [40] Haber, E., Horesh, L. & Tenorio, L. “Numerical methods for experimental design of large-scale linear ill-posed inverse problems”. *Inverse Problems* **24.5** (2008).
- [41] Haber, E., Horesh, L. & Tenorio, L. “Numerical methods for the design of large-scale nonlinear discrete ill-posed inverse problems”. *Inverse Problems* **26.2** (2009).
- [42] Haber, E., Magnant, Z., Lucero, C. & Tenorio, L. “Numerical methods for A-optimal designs with a sparsity constraint for ill-posed inverse problems”. *Computational Optimization and Applications* **52** (2012), pp. 293–314.

- [43] Halko, N., Martinsson, P. G. & Tropp, J. A. “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”. *SIAM Review* **53.2** (2011), pp. 217–288.
- [44] Herman, E., Alexanderian, A. & Saibaba, A. K. “Randomization and reweighted ℓ_1 -minimization for A-optimal design of linear inverse problems”. *arXiv:1906.03791* (2019). Accepted, *SIAM Journal on Scientific Computing*.
- [45] Herman, E., Stewart, J. A. & Dingreville, R. “A data-driven surrogate model to rapidly predict microstructure morphology during physical vapor deposition” (2020). Under review.
- [46] Horesh, L., Haber, E. & Tenorio, L. “Optimal experimental design for the large-scale nonlinear ill-posed problem of impedance imaging”. *Large-Scale Inverse Problems and Quantification of Uncertainty*. John Wiley & Sons, Ltd, 2010, pp. 273–290.
- [47] Horn, R. A. & Johnson, C. R. *Matrix analysis*. Second. Cambridge: Cambridge University Press, 2012.
- [48] Huan, X. & Marzouk, Y. M. “Simulation-based optimal Bayesian experimental design for nonlinear systems”. *Journal of Computational Physics* **232.1** (2013), pp. 288–317.
- [49] Hunter, D. R. & Lange, K. “A tutorial on MM algorithms”. *The American Statistician* **58.1** (2004), pp. 30–37.
- [50] Jolliffe, I. T. *Principal component analysis*. 2nd ed. Springer Series in Statistics. Springer-Verlag New York, 2002.
- [51] Jolliffe, I. T. & Cadima, J. “Principal component analysis: A review and recent developments”. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**.2065 (2016).
- [52] Jung, J., Yoon, J. I., Park, H. K., Kim, J. Y. & Kim, H. S. “An efficient machine learning approach to establish structure-property linkages”. *Computational Materials Science* **156** (2019), pp. 17–25.
- [53] Kalidindi, S. R., Niezgoda, S. R. & Salem, A. A. “Microstructure informatics using higher-order statistics and efficient data-mining protocols”. *JOM* **63.4** (2011), pp. 34–41.
- [54] Khosravani, A., Cecen, A. & Kalidindi, S. R. “Development of high throughput assays for establishing process-structure-property linkages in multiphase polycrystalline metals: Application to dual-phase steels”. *Acta Materialia* **123** (2017), pp. 55–69.
- [55] Kim, J., Lee, S., Choi, Y., Lee, S.-M. & Jeong, D. “Basic principles and practical applications of the Cahn-Hilliard equation”. *Mathematical Problems in Engineering* **2016** (2016).
- [56] Kitanidis, P. K. & Lee, J. “Principal component geostatistical approach for large-dimensional inverse problems”. *Water Resources Research* **50** (2014), pp. 5428–5443.

- [57] Knio, O. & Maître, O. L. “Uncertainty propagation in CFD using polynomial chaos decomposition”. *Fluid Dynamics Research* **38.9** (2006), pp. 616–640.
- [58] Körkel, S., Kostina, E., Bock, H. G. & Schlöder, J. P. “Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes”. *Optimization Methods and Software* **19.3–4** (2004), pp. 327–338.
- [59] Koval, K., Alexanderian, A. & Stadler, G. “Optimal experimental design under irreducible uncertainty for inverse problems governed by PDEs”. *arXiv:1912.08915* (2019).
- [60] Lahmer, T. “Optimal experimental design for nonlinear ill-posed problems applied to gravity dams”. *Inverse Problems* **27.12** (2011).
- [61] Lange, K. *MM optimization algorithms*. Vol. 147. Society for Industrial and Applied Mathematics, 2016.
- [62] Latypov, M. I. & Kalidindi, S. R. “Data-driven reduced order models for effective yield strength and partitioning of strain in multiphase materials”. *Journal of Computational Physics* **346** (2017), pp. 242–261.
- [63] Lee, D., Huh, J.-Y., Jeong, D., Shin, J., Yun, A. & Kim, J. “Physical, mathematical, and numerical derivations of the Cahn-Hilliard equation”. *Computational Materials Science* **81** (2014), pp. 216–225.
- [64] Lee, E.-Y. “Extension of Rotfel’d theorem”. *Linear Algebra and its Applications* **435.4** (2011), pp. 735–741.
- [65] Li, G., Iskandarani, M., Hénaff, M. L., Winokur, J., Maître, O. L. & Knio, O. “Quantifying initial and wind forcing uncertainties in the Gulf of Mexico”. *Computational Geosciences* **20.5** (2016), pp. 1133–1153.
- [66] Lifshits, M. A. *Gaussian random functions*. Mathematics and Its Applications. Springer Netherlands, 1995.
- [67] Long, Q., Motamed, M. & Tempone, R. “Fast Bayesian optimal experimental design for seismic source inversion”. *Computer Methods in Applied Mechanics and Engineering* **291** (2015), pp. 123–145.
- [68] Lu, Y., Wang, C., Gao, Y., Shi, R., Liu, X. & Wang, Y. “Microstructure map for self-organized phase separation during film deposition”. *Physical Review Letters* **109.8** (2012).
- [69] Ma, J., Hu, J., Li, Z. & Nan, C.-W. “Recent progress in multiferroic magnetoelectric composites: From bulk to thin films”. *Advanced Materials* **23.9** (2011), pp. 1062–1087.
- [70] Mahan, J. E. *Physical vapor deposition of thin films*. John Wiley & Sons, Ltd, 2000.
- [71] Maître, O. L. & Knio, O. *Spectral methods for uncertainty quantification: With applications to computational fluid dynamics*. Scientific Computation. Springer Netherlands, 2010.

- [72] Makhlouf, A. S. H. *Nanocoatings and ultra-thin films*. Woodhead Publishing Series in Metals and Surface Engineering. Woodhead Publishing, 2011.
- [73] Marzouk, Y. M. & Najm, H. N. “Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems”. *Journal of Computational Physics* **228.6** (2009), pp. 1862–1902.
- [74] McDowell, D. L., Ghosh, S. & Kalidindi, S. R. “Representation and computational structure-property relations of random media”. *JOM* **63.3** (2011), pp. 45–51.
- [75] Meyer, C. D. *Matrix analysis and applied linear algebra*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2000.
- [76] Moelans, N., Blanpain, B. & Wollants, P. “An introduction to phase-field modeling of microstructure evolution”. *Calphad* **32.2** (2008), pp. 268–294.
- [77] Nagel, J. B., Rieckermann, J. & Sudret, B. “Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration: Application to urban drainage simulation”. *Reliability Engineering & System Safety* **195** (2020).
- [78] Najm, H. N. “Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics”. *Annual Review of Fluid Mechanics* **41.1** (2009), pp. 35–52.
- [79] Niezgoda, S. R., Fullwood, D. T. & Kalidindi, S. R. “Delineation of the space of 2-point correlations in a composite material system”. *Acta Materialia* **56.18** (2008), pp. 5285–5292.
- [80] Niezgoda, S. R., Yabansu, Y. C. & Kalidindi, S. R. “Understanding and visualizing microstructure and microstructure variance as a stochastic process”. *Acta Materialia* **59.16** (2011), pp. 6387–6400.
- [81] Nikolas Provatas, K. E. *Phase-field methods in materials science and engineering*. John Wiley & Sons, Ltd, 2010.
- [82] Ouellette, D. V. “Schur complements and statistics”. *Linear Algebra and its Applications* **36** (1981), pp. 187–295.
- [83] Paulson, N. H., Priddy, M. W., McDowell, D. L. & Kalidindi, S. R. “Reduced-order structure-property linkages for polycrystalline microstructures based on 2-point statistics”. *Acta Materialia* **129** (2017), pp. 428–438.
- [84] Pázman, A. *Foundations of optimum experimental design*. Mathematics and its Applications. Springer Netherlands, 1986.
- [85] Prato, G. D. & Zabczyk, J. *Second order partial differential equations in Hilbert spaces*. Cambridge: Cambridge University Press, 2002.
- [86] Pukelsheim, F. *Optimal design of experiments*. Society for Industrial and Applied Mathematics, 2006.

- [87] Rudin, W. *Principles of mathematical analysis*. 3rd ed. International series in pure and applied mathematics. New York: McGraw-Hill, 1976.
- [88] Ruthotto, L., Chung, J. & Chung, M. “Optimal experimental design for inverse problems with state constraints”. *SIAM Journal on Scientific Computing* **40.4** (2018), B1080–B1100.
- [89] Saibaba, A. K., Alexanderian, A. & Ipsen, I. C. F. “Randomized matrix-free trace and log-determinant estimators”. *Numerische Mathematik* **137.2** (2017), pp. 353–395.
- [90] Shahidi, S., Moazzenchi, B. & Ghoranneviss, M. “A review-application of physical vapor deposition (PVD) and related methods in the textile industry”. *Eur. Phys. J. Appl. Phys.* **71.3** (2015).
- [91] Shlens, J. “A tutorial on principal component analysis”. *arXiv:1404.1100* (2014).
- [92] Sinkoe, A. & Hahn, J. “Optimal experimental design for parameter estimation of an IL-6 signaling model”. *Processes* **5** (2017).
- [93] Smith, J., Xiong, W., Yan, W., Lin, S., Cheng, P., Kafka, O. L., Wagner, G. J., Cao, J. & Liu, W. K. “Linking process, structure, property, and performance for metal-based additive manufacturing: Computational approaches with experimental support”. *Computational Mechanics* **57.4** (2016), pp. 583–610.
- [94] Smith, R. C. *Uncertainty quantification: Theory, implementation, and applications*. Society for Industrial and Applied Mathematics, 2013.
- [95] Stewart, J. A. & Dingreville, R. “Microstructure morphology and concentration modulation of nanocomposite thin-films during simulated physical vapor deposition”. *Acta Materialia* **188** (2020), pp. 181–191.
- [96] Stewart, J. A., Kohnert, A. A., Capolungo, L. & Dingreville, R. “Design and analysis of forward and reverse models for predicting defect accumulation, defect energetics, and irradiation conditions”. *Computational Materials Science* **148** (2018), pp. 272–285.
- [97] Stuart, A. M. “Inverse problems: A Bayesian perspective”. *Acta Numerica* **19** (2010), pp. 451–559.
- [98] Sudret, B. “Global sensitivity analysis using polynomial chaos expansions”. *Reliability Engineering & System Safety* **93.7** (2008). Bayesian Networks in Dependability, pp. 964–979.
- [99] Tarantola, A. *Inverse problem theory and methods for model parameter estimation*. Society for Industrial and Applied Mathematics, 2005.
- [100] Tashkinov, M. “Statistical methods for mechanical characterization of randomly reinforced media”. *Mechanics of Advanced Materials and Modern Processes* **3.1** (2017).
- [101] Thompson, C. V. “Structure evolution during processing of polycrystalline films”. *Annual Review of Materials Science* **30.1** (2000), pp. 159–190.

- [102] Thornton, J. A. “The microstructure of sputter-deposited coatings”. *Journal of Vacuum Science and Technology A* **4** (1986), pp. 3059–3065.
- [103] Torquato, S. *Random heterogeneous materials: Microstructure and macroscopic properties*. Interdisciplinary Applied Mathematics. Springer-Verlag New York, 2002.
- [104] Uciński, D. *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2005.
- [105] Walsh, S. N., Wildey, T. M. & Jakeman, J. D. “Optimal experimental design using a consistent Bayesian approach”. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* **4.1** (2017).
- [106] Williams, D. *Probability with martingales*. Cambridge: Cambridge University Press, 1991.
- [107] Xia, L., Raghavan, B. & Breitenkopf, P. “Towards surrogate modeling of material microstructures through the processing variables”. *Applied Mathematics and Computation* **294** (2017), pp. 157–168.
- [108] Xiu, D. & Karniadakis, G. “The Wiener-Askey polynomial chaos for stochastic differential equations”. *SIAM Journal on Scientific Computing* **24.2** (2002), pp. 619–644.
- [109] Yabansu, Y. C., Steinmetz, P., Hötzer, J., Kalidindi, S. R. & Nestler, B. “Extraction of reduced-order process-structure linkages from phase-field simulations”. *Acta Materialia* **124** (2017), pp. 182–194.
- [110] Yang, Z., Li, X., Catherine Brinson, L., Choudhary, A. N., Chen, W. & Agrawal, A. “Microstructural materials design via deep adversarial learning methodology”. *Journal of Mechanical Design* **140.11** (2018).
- [111] Yu, J., Zavala, V. M. & Anitescu, M. “A scalable design of experiments framework for optimal sensor placement”. *Journal of Process Control* **67** (2018). Big Data: Data Science for Process Control and Operations, pp. 44–55.