THE UNIVERSITY OF CHICAGO

ON THE APPLICATION, THEORY AND COMPUTATION OF OPTIMAL EXPERIMENTAL DESIGN IN THE CONTEXT OF SENSOR PLACEMENT

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF STATISTICS

BY
JING YU

CHICAGO, ILLINOIS
AUGUST 2019

To my parents, Rongzhen Duan and Zhibin Yu

Table of Contents

List of Figures

Acknowledgments

First of all, I would like to express my sincere gratitude to my PhD adviser Mihai Anitescu for his guidance, patience and continuous support during the past four years. I am especially grateful for his encouraging me to be curious and initiative-oriented other than hard-working. Besides, his good sense of humor makes regular meetings both illuminating and entertaining. It is my fortune to spend years of youth working on exciting projects under his supervision.

Without the help of Jonathan Weare and Lek-Heng Lim, the other two committee members, this thesis would not have been possible. I thank them for providing valuable feedback and suggestions on the thesis, and for supporting my application for summer internships and academic jobs. The two courses taught by Lek-Heng Lim led me to the world of optimization and I continue on scientific research in this area. My sincere thanks also goes to Jonathan Weare for introducing interesting projects on stochastic simulation and for holding insightful and stimulating discussions on hard problems.

I would like to thank Mary Sara McPeek for writing letters twice to support my research internships, thank all the faculty members for delivering high-quality courses, and my fellow students for making everyday life colorful and enjoyable. Also, I thank my previous adviser Edwin Perkins for supporting my PhD application without reservation, and thank Yongjin Wang for helping me make the big decision to go abroad and pursue my academic goals.

Finally, I would like to thank and dedicate this thesis to my parents for being supportive to every decision I made throughout my life, for unconditionally believing in me whenever I encounter obstacles and find it hard to overcome, and for sacrificing their own time and even career to provide me the best care. I feel extremely lucky and grateful to be their daughter.

Abstract

Optimal experimental designs are a class of experimental designs that are optimal with respect to some statistical criterion. Sensor placement is a sampling decision on data collection which aims to minimize the uncertainty in parameter estimation. This thesis focuses on two fundamental elements: the selection of sensor locations under statistically optimal conditions, and the computation of sensor placement with an efficient algorithm.

We first present a design of experiments framework for sensor placement in a natural gas pipeline system where the dynamics are described by partial differential equations, and apply *sum-up rounding* strategy as a heuristic to determine the sensor locations. We continue to develop convergence theory on *sum-up rounding* for Bayesian inverse problems, where the direct relationship is described through a discretized integral equation. We show that the integer solution from *sum-up rounding* is asymptotically optimal in the limit of increasingly refined meshes, for different experimental design criteria (A- and D- optimal), and demonstrate its superior performance in comparison with other standard strategies.

We also propose an optimization algorithm to compute the sensor locations, based on sequential quadratic programming and Chebyshev interpolation. By providing gradient and Hessian information on the objective, we solve a sequence of quadratic programs with interior point method and achieve a complexity of $O(n \log^s(n))$, while controlling the error through choosing the number of interpolation points to satisfy a user-defined precision level.

1 Introduction

Design of experiments (DOE) is an important branch of statistics that aims to determine experimental settings and extract the most useful information from data to explain the variation of quantities of interest, which may arise both from the measurement procedure and from the inherent variability of experimental material. In traditional DOE, one selects suitable treatments, assigns the treatments to experimental units, and observes treatment effects by measuring response variables (see [1, 2, 3, 4]). The experimenter also identifies control variables that must be held constant to prevent external factors from affecting the results. Experimental design involves not only the selection of treatment effects, experimental units and control variables, but also the delivery of the experiment under statistically optimal conditions given the constraints of available resources. The optimality of a design depends on the statistical model and is usually related to the variance-matrix of the estimator.

Another branch of DOE attempts to compute the optimal sampling locations given a set of available measurement points (see [5, §7.5] and [6, §9, §12]). For instance, in polynomial regression, where the dependent variable y is modeled as dth degree polynomial in the independent variable x, the goal is to determine the optimal support in $\mathcal{T} = [-1, 1]$ consisting of $d + 1$ different points, such that the information matrix is maximized according to some design criterion. Closed form solutions are available for several design criteria (see [7] and [6, §9]), but in general they are difficult to derive and computational methods are required to approximate the optimal design. It becomes even more challenging when either the dimension of parameters to be inferred or the number of available measurement points increases drastically, and optimal sensor placement is an example of this second-type challenge.

Optimal sensor placement aims to determine the number, locations, and types of sensors that would give the most accurate estimation of parameters or maximize the information about a system. Naturally it arises in many infrastructure networks (oil, water, gas, and electricity) in which large amounts of sensor data need to be processed in real time in order to reconstruct the state of the system or to identify leaks, faults, or attacks. It can also be viewed as a sampling decision of measurements, and there are different formulations in literature (see [8, 9, 10]). In Gaussian process and Kriging (see [11, 12]), the training data $\{(u_k, y(u_k))\}_{k=1}^{N}$ are sampled to predict the response $y(u)$ of a process at some unsampled point u. The observations are modeled as $y(u_k) = f(u_k, \theta) + P(u_k, \omega) + \epsilon_k$ where $f(u, \theta)$ is given and $p(u, \omega)$ denotes a stationary zero-mean random processes with parameters in its covariance function. This formulation makes statistical inference even possible for purely deterministic systems, and it has been successfully applied in many domains of engineering (see [13, 14]). However, few results exist on the optimal sampling of training data, and they are generally assumed to be a collection of identically and independently distributed pairs $(u, y(i))$ (see [15, 16]). The optimal design asymptotically tends to observe everywhere in the design domain, and the points $\{u_k\}_{k=1}^{N}$ are distributed according to a density with respect to Lebesgue measure, but little is known on the analytical form (see [17]).

Sensor placement plays a vital role in the operation of automation systems, and thus has significant impact on our everyday life. In automatic vehicles, the central ingredient is the lidar (light detection and ranging) sensor, a device that maps objects in 3D by bouncing laser beams off its real-world surroundings. Driverless vehicles rely heavily on lidar to locate themselves on the detailed maps they need to get around, and to identify things like pedestrians and other vehicles. Lidar sensors are expensive, costing thousands or even tens of thousands of dollars a piece, and one self-driving vehicle is usually equipped with several lidars. Together with other types of sensors (radar sensors, camera sensors) in automatic vehicles, they collect information for the software system to process, plan and then execute. Different sensors have different mounting positions, and it is of essential importance to study the impact of sensor positioning (the field of view of sensors) on the scenarios that autonomous cars can manage. Another example of remote sensing system where lidars are widely used is weather forecasting. The spinning lidar design, which includes the balance between the scanning frequency and spatial distribution of laser emission, is crucial in collecting measurements to infer the wind speed and humidity for use in weather prediction.

Optimal sensor placement is computationally difficult because given n candidate locations, there are 2^n possible combinations for sensor placement which is exponential in n and makes the computation NP-hard [1]. The goal of this thesis is to efficiently compute an approximation that converges to the optimal sensor placement in the limit of n, in particular settings and under particular technical assumptions. Our approach is to first relax the integrality constraint and then round it off to an integer one. More specifically, we will

- apply a rounding strategy that provides a feasible point to the optimization in formulation, and examine the changes in optimality gap as n gets large;
- prove zero convergence of optimality gap [2] under various design criteria for continuously indexed problems from a class of integral operators;
- propose a scalable [3] optimization algorithm to compute the rounding solution efficiently and study its accuracy as n gets large.

Each will be elaborated in one chapter of the thesis. Now we introduce our estimation framework and formulate our optimization problem, mainly following [8]. Consider a setting with measurements perturbed by additive Gaussian noise,

$$\mathrm{d} = \mathcal{F}(u_0) + \eta, \quad \eta \sim \mathcal{N}(\mathbf{0}, \Gamma_{\mathrm{noise,d}}),$$

[1] "NP-hard" stands for "non-deterministic polynomial acceptable problems". Although it is suspected that there are no polynomial-time algorithms for NP-hard problems, this has not been proven.

[2] The gap between the objective obtained from rounding strategy and the true minimum objective.

[3] complexity less than $O(n \log^s(n))$

where $\Gamma_{\text{noise,d}} \in \mathbb{R}^{n\times n}$ is the measurement noise covariance matrix and f is an operator that maps a parameter vector $u_0 \in \mathbb{R}^m$ to the observation vector $\mathrm{d} \in \mathbb{R}^n$. Formally,

$$\mathcal{F} : u_0 \overset{f}{\longmapsto} u \overset{w}{\longmapsto} \mathrm{d} \tag{1.1}$$

where u_0 are the input parameters we are interested in, f is the parameter-to-observable operator, $u \in \mathbb{R}^n$ is the output vector that can be potentially observed, w is the observable-to-observed operator which relates to the sensor locations, d is our observation from sensors, and it is a subset of the output u. We consider continuously indexed problems, that is, both u_0 and u are function discretizations on an increasingly refined mesh. Note u and d described in (1.1) are not subject to measurement error.

This representation can also be viewed as an inverse problem in the language of mathematical modeling, where the goal is to infer the input from the observed output. Recently, the Bayesian approach has received lots of attention (see [18]) and has been widely applied in many areas (see [8, 19, 20]). It not only allows for the quantification of uncertainity and risk, but also addresses significant modeling issues, such as ill-posedness of inverse problems, in a clear and precise fashion. We adopt the Bayesian framework and follow the formulation in [8]: assume both the parameter prior and the measurements distributions are Gaussian:

$$\begin{aligned} u_0 &\sim \mathcal{N}(u_{\text{prior}}, \Gamma_{\text{prior}})\,, \\ u =& f(u_0) + \eta, \text{ where } \eta \sim \mathcal{N}(0, \Gamma_{\text{noise}}). \end{aligned}$$

Here, Γ_{prior} and Γ_{noise} represent the prior covariance matrix and measurement noise covariance matrix respectively, whereas u_{prior} is the prior mean. We assume the measurement error to be unbiased conditional on the realization of u_0, and thus η has mean 0.

If the mapping f is linear ($u = Fu_0 + \eta$), from Bayes' rule, we know the posterior distribution of u_0 is also Gaussian and has (up to a constant) the following density:

$$\pi_{\text{post}}(u_0|u) \propto \exp\Big\{ -\frac{1}{2}\|u - Fu_0\|_{\Gamma^{-1}_{\text{noise}}} - \frac{1}{2}\|u_0 - u_{\text{prior}}\|_{\Gamma^{-1}_{\text{prior}}} \Big\}. \tag{1.2}$$

Next we quantify the sensor placement effect in the posterior by creating a weight vector $w=(w_1, w_2, .., w_n) \in \{0, 1\}^n$ where the jth component w_j corresponds to candidate location x_j. A sensor is placed at location x_j if $w_j = 1$ and is not placed if $w_j = 0$, so there is a one-to-one mapping between sensor placement and weight vectors. Let W be a diagonal matrix with weight vector w on the diagonal. The w-weighted data likelihood is given by

$$\pi_{\text{like}}(d|u_0, w) \propto \exp\Big\{ -\frac{1}{2}(d - Fu_0)^T W^{1/2}\Gamma^{-1}_{\text{noise}}W^{1/2}(d - Fu_0)\Big\}.$$

One can immediately verify that for any integer-valued vector w, the posterior distribution is exactly the one for Bayesian least squares with data measured for indices of u where $w_i = 1$ for $i = 1, 2, \ldots, n$. One can either think of the data d as weighted output u, i.e. $d = Wu$, or a lower-dimension copy of u, see Appendix A for more details. Under these assumptions and accounting for the prior distribution, we can compute the posterior u_0, which is the normal distribution $\mathcal{N}(u_{\text{post}}, \Gamma_{\text{post}})$, where

$$\begin{aligned} u_{\text{post}} &= \Gamma_{\text{post}}\Big(F^T W^{1/2}\Gamma^{-1}_{\text{noise}}W^{1/2}d + \Gamma^{-1}_{\text{prior}}u_{\text{prior}}\Big), \\ \Gamma_{\text{post}} &= \Big(F^T W^{1/2}\Gamma^{-1}_{\text{noise}}W^{1/2}F + \Gamma^{-1}_{\text{prior}}\Big)^{-1} \end{aligned} \tag{1.3}$$

are the posterior mean and covariance matrix, respectively. We point out in this estimation model the posterior covariance matrix does not depend on data d, and if we minimize certain metrics of this matrix to calculate the optimal sensor placement, it is determined by the linear mapping f and two Γ matrices.

We are ready to formulate our DOE problem that addresses the issue of optimal sensor placement. The objective is to minimize the estimation error of the parameter u_0, which is quantified by its posterior covariance matrix, $\phi(\Gamma_{\text{post}})$. The three most widely used criteria in experimental design (see [6]) to measure the size of this error are

- A-optimal design: $\phi(\Gamma_{post}) = tr(\Gamma_{post})$;
- D-optimal design: $\phi(\Gamma_{post}) = det(\Gamma_{post})$;
- E-optimal design: $\phi(\Gamma_{post}) = \lambda_{max}(\Gamma_{post})$.

The DOE problem is given as follows (ϕ represents one of the three criteria, and we use *logdet* for D-optimal design):

$$\begin{aligned} \min \quad & \phi(\Gamma_{post}(w)) \\ \text{s.t.} \quad & w_i \in \{0,1\},\ \textstyle\sum_{i=1}^{n} w_i = n_0, \end{aligned} \tag{1.4}$$

where n_0 is the number of sensors on budget. All the above design criteria have the property that, when there are more sensors available, there is less uncetainty remaining in the estimation, i.e., the objective value is smaller. As mentioned earlier, this is an NP-hard problem. Regularization methods have been exploited to alleviate the computation burden ([8, 20]) by removing the integrality constraint, and controlling the number of sensors (i.e., the design cost) by using an sparsity-inducing ℓ_0 regularization norm that is in turn approximated by using a smoothing function. This method requires tuning and can be numerically unstable. Instead, we start with the relaxed version:

$$\begin{aligned} \min \quad & \phi(\Gamma_{\text{post}}(w)) \\ \text{s.t.} \quad & 0 \le w_i \le 1, i = 1, 2, \dots, n,\ \textstyle\sum_{i=1}^{n} w_i = n_0, \end{aligned} \tag{1.5}$$

whose solution we denote by w_{rel}. This problem is convex if the covariance matrix Γ_{noise} is chosen appropriately, such as a diagonal matrix. It can be solved in polynomial time using, for example, interior-point algorithms [5] with gradient and Hessian information. The relaxed solution w_{rel} to (1.5) provides a lower bound to the optimal objective of the convex integer program (1.4), whereas any feasible point would provide an upper bound to (1.4). Next we introduce a rounding strategy that gives a desirable upper bound.

The rounding strategy we apply is called *sum-up rounding* (SUR), which was first used in the context of continuous-time mixed-integer optimal control problems (MIOCPs) (see [21]). SUR for binary variables, as we also pursue here, has been shown in temporally indexed problems to have the desirable asymptotic property of being arbitrarily close to an integer solution as long as the discretization mesh is sufficiently fine (see [21, 22]). In [21], the authors not only clarify the role of SUR in MIOCPs but also obtain a guaranteed bound on the performance loss, depending on the size of discretization mesh. In [22], a specific structure in one dimension is considered where the objective is a function of either the Fisher information matrix or its inverse, and the optimality gap converges to zero. Therefore, *Sum-up rounding* sheds light on both theory and computation of MIOCPs. However, new theory needs to be developed to obtain similar results in the infinite-dimensional setting, because all the previous results are for a fixed and finite number of parameters.

The basic SUR strategy to construct a binary vector w_{int} from w_{rel} is given by:

$$w_{int}^i = \begin{cases} 1, & \text{if } \sum_{k=0}^{i} w_{rel}^i - \sum_{k=0}^{i-1} w_{int}^i \geq 0.5 \\ 0, & \text{otherwise.} \end{cases}$$

for $i = 1, ..., n$. The equality constraint $\sum_{i=1}^{n} w_i = n_0$ is satisfied automatically (see Lemma 3.3.2 in §3.3.1). The idea is to process each element sequentially based on the sum of elements that are already processed, and control the sum difference between w_{int} and w_{rel}.

In Chapter 2, we apply *sum-up rounding* as a heuristic to a nonlinear dynamical system - a gas pipeline network, where the dynamics are described by hyperbolic partial differential equations (PDEs), and the parameter is the initial condition of the PDEs. To represent the problem using our formulation, the parameter-to-observable operator f is nonlinear and it maps the initial condition u_0 to $u(t)$ for $t \in [0, T]$ according to the dynamics described by PDEs. The parameter u_0 contains gas pressure and flow discretized from a mesh of the pipeline, whereas u_t is indexed in both space and time. The measurement data are available from sensors at each discrete time point, but contaminated by noise. In this nonlinear setting, calculating posterior variances in closed form is almost impossible, and we would resort to sampling methods, such as Markov Chain Monte Carlo, for computing the posterior density. Linearization is applied to make the calculation more tractable, and we test the performance of SUR by investigating two objectives: the total-flow variance and the A-optimal design criterion. We conclude that *sum-up rounding* approach produces shrinking gaps with increased meshes, and also observe convergence of gap for two noise structures, which motivates us to develop theories for *Sum-up rounding* in the infinite-dimensional setting.

In Chapter 3, we establish the theoretical work on zero convergence of optimality gap. To briefly describe our theoretical setting, the unknown parameter u_0 is a function that belongs to an infinite-dimensional space, which is approximated by discretization on increasingly fine meshes. We aim to understand the asymptotics of the rounding procedure in the limit of the mesh size going to zero. As a result, the posterior covariance matrix we try to minimize (with respect to a given design criterion) increases in size with the number of discretization points, and we are not aware of prior theoretical work on the convergence analysis of discretized design of experiments with a number of sites that can grow unboundedly.

The particular assumption we make is that the parameter-to-observable operator f is based on an integral equation, as opposed to the solution of a PDE in Chapter 2:

$$u(x) = \int_{\Omega_{in}} f(x,y) u_0(y) \, \mathrm{d}y, \quad x \in \Omega_{out},$$

where Ω_{in} and Ω_{out} are input domain and output domain respectively. The techniques we employ to this end are related to the spectral theory of self-adjoint integral operators [23]. Ultimately we show that, under other technical assumptions,

$$\left| \phi\Big(\Gamma(w^n_{rel})\Big) - \phi\Big(\Gamma(w^n_{int})\Big) \right| \to 0 \qquad \text{as } n \to \infty.$$

for any of the design criteria defined earlier. We demonstrate the effectiveness of SUR and compare with another rounding strategy called *thresholding* in a gravity-surveying example.

In Chapter 4, we provide an optimization algorithm to compute the relaxed solution, based on Chebyshev interpolation and sequential quadratic programming. While the relaxed problem is not NP-hard, the interior-point based algorithm has a complexity of $O(n^3)$. Given that f may come from a mathematical model typically described by a system of PDEs, and the parameters to be estimated are initial or boundary conditions, the discretization of an increasingly refined mesh can easily make the problem size explode to thousands and even millions, so a $O(n^3)$ algorithm is intractable in practice. A scalable algorithm is needed to solve it in a fast and accurate fashion.

Several efficient algorithms have been proposed to tackle the computation issue in literature ([8, 19, 20]) for specific design criteria, all of which exploit low-rank structure of the parameter-to-observable mapping in some way. In [8], randomized methods, such as randomized sigular value decomposition (rSVD) and randomized trace estimator, are employed to evaluate the A-optimal design objective function, i.e. trace of posterior covariance matrix, and its gradient, and the error depends on the threshold chosen in rSVD and the sample size in randomized estimation. In [20], similar approaches (truncated spectral decomposition, randomized estimators for determinants) are investigated for the D-optimal design criterion, and error bounds are derived explicitly. In addition to computational work, a general study on the optimal low-rank update from the prior covariance matrix to the posterior covariance matrix, is also given over a broad class of loss functions (see [24]).

We make use of the integral operator assumption and the fact F is continuously indexed i.e., $F(i,j)$ is evaluated from $f(x,y)$ for $x \in \Omega_{out}, y \in \Omega_{in}$, and propose an interpolation-based algorithm to approximate the gradient and Hessian for both A- and D-optimal design criteria. Interpolation theory is well developed in numerical analysis to approximate function values with evaluations only at a subset of points. It is known that polynomial interpolation at Chebyshev points is optimal in the minimax error for continuously differentiable functions (see [25, §8.5]). We solve a sequence of quadratic programs (SQP), and each quadratic program is solved with interior-point method. The advantage of SQP is twofold: in contrast to previous algorithms, we incorporate Hessian information which can accelerate the convergence rate of optimization algorithms, and the objective value is not required for SQP; we are able to prove the zero convergence of approximation error in the objective value as the problem size goes to infinity. Since the relaxed problem is convex, the KKT condition is both sufficient and necessary for a solution to be optimal, and we demonstrate the effectiveness of our approximation by the shrinkage of maximum KKT violation.

In summary, we apply the *sum-up rounding* strategy as a heuristic in a nonlinear dynamic system of a natural gas network, and demonstrate our observation of gap convergence in Chapter 2, and then prove the zero convergence of gap for a class of integral operators under different design criteria in Chapter 3. In Chapter 4, we propose a scalable algorithm based on sequential quadratic programming and Chebyshev interpolation, to solve the relaxation efficiently. Finally, we discuss various directions for future research.

1.1 Previous work

Inverse problems with Bayesian formulation have been extensively explored recently. A comprehensive review on well-posedness and stability from a function space viewpoint can be found in [18] for linear inverse problems with Gaussian prior and Gaussian likelihood.

Following [18], a framework for A-optimal experimental design together with a randomized optimization algorithm are given in [8] for infinite-dimensional Bayesian linear inverse problems governed by partial differential equations. In their papers, the measurement errors from sensors are uncorrelated, and the covariance operator in the prior is specified as the inverse of an elliptic differential operator. The parameters to be estimated are the coefficients of basis functions in a finite-dimensional subspace of the original infinite-dimensional function space, and the function in the subspace is approximated with a finite-element method. Instead of the equality constraint on the number of sensors, the sparsity of sensor configuration is controlled by employing a sequence of penalty functions that successively approximate the l_0 norm, and tuning is required in the regularization term. A low-rank approximation of the parameter-to-observable map, preconditioned with the square root of the prior covariance operator, and a randomized trace estimator for evaluation of the A-optimal design objective and its gradient, are exploited to reduce the computation cost.

The technique of *sum-up rounding* (SUR) was first applied in the context of continuous-time mixed-integer optimal control problems [21]. Sum-up rounding for binary variables has been shown in temporally indexed problems to have the desirable asymptotic property of approximating the solution to a relaxed and convexified problem with arbitrary precision, as long as the discretization mesh is sufficiently fine [21, 22]. A proof of guaranteed bound for applying SUR on the performance loss, depending linearly on the size of discretization mesh, is given in [21]. In [22], a specific structure in one dimension based on information gain is considered where the objective is a function of either the Fisher information matrix or its inverse, and the optimality gap converges to zero. These works use frequentist approaches, and the parameter of interest has a fixed dimension, so does the Fisher information matrix.

1.2 Contributions

We first apply *sum-up rounding* strategy as heuristics in natural gas pipelines where the dynamics are described by a system of partial differential equations on a spatial and temporal domain. We investigate metrics to guide the design of experiments (the total flow variance and the A-optimal design criterion) and analyze the effect of different noise structures. We conclude that the sum-up rounding approach gives the best results and produces shrinking gaps with increasing mesh resolution. We also observe that convergence for the white noise measurement error case is slower than for the colored noise case.

We then extend the *sum-up rounding* approach to multiple dimensions, analyze its accuracy as a function of the discretization mesh size for a rectangular domain, and prove asymptotic optimality of *sup-up rounding* solutions under different design criteria (A- and

D- optimal). More specifically, we consider a statistical setup that consists of a Bayesian framework for linear inverse problems for which the direct relationship is described by a discretized integral equation, and aim to find the optimal sensor placement from a set of candidate locations where data are collected with measurement error. The convex objective function is a measure of the uncertainty, described by the trace or log-determinant of the posterior covariance matrix. The resulting convex integer program is relaxed, producing a lower bound. An upper bound is obtained by extending the sum-up rounding approach to multiple dimensions. *We show the convergence to zero of the gap between the upper and lower bounds as the mesh size goes to zero.* The technique is illustrated on a two-dimensional gravity surveying problem for both A-optimal and D-optimal sensor placement where our designs yield better results compared with thresholding rounding approach.

We also develop an optimization algorithm by taking advantage of the continuously-indexed structure, propose an interpolation-based approximation to the derivative and Hessian of the objective experimental design criterion – the trace or log-determinant of posterior covariance matrix, and study its accuracy by looking into the difference between the approximation and the true minimum in the objective value. *The complexity is reduced to* $\mathcal{O}(n \log^s(n))$*.* A sequential quadratic programming algorithm, with each quadratic program solved by interior point method, is implemented in Julia without using any existing optimization package. This algorithm is more than 100 times faster than using a standard package, such as `Ipopt` in Julia by passing through the exact gradient and Hessian, which makes it possible to solve problems with hundreds of thousands of integer variables on a laptop with only 4 GB memory. We demonstrate the efficiency of this algorithm on a linear inverse problem governed by advection-diffusion equations, in search of optimal sensing directions for lidar to collect data and infer the initial conditions.

We emphasize the application of *sum-up rounding* strategy to a spatial domain, and the zero gap convergence of rounded DOE solutions over increasing design space sizes have not been investigated before.

2 Application in A Natural Gas Pipeline System

In this chapter, we present a scalable design of experiments framework to compute optimal sensor locations for systems described by partial differential equations (PDEs). This is done by minimizing the uncertainty in the state and parameters estimated from Bayesian inverse problems. The resulting problem is a mixed-integer infinite-dimensional optimal control problem. We apply two heuristics that have the potential to be scalable for such problems: a sparsity-inducing approach [8] and a sum-up rounding approach [26]. We investigate two objectives: the total flow variance and the A-optimal design criterion. Using a natural gas pipeline case study, we conclude that the sum-up rounding approach produces shrinking gaps with increased meshes. We also observe that convergence for the white noise measurement error is slower than for the colored noise case. For the A-optimal design the solution is close to the uniform distribution, but for the total flow variance the pattern is noticeably different.

2.1 Introduction

The sensor placement problem seeks to determine the optimal number, locations, and types of sensors that would maximize information about a dynamical system. Because information can often be expressed in terms of the posterior covariance matrix of the states or parameters of the system, the problem can often be cast as an optimal design of experiments problem. Such a problem is computationally challenging, particularly in the infinite-dimensional case, because one must solve a mixed-integer and bilevel optimization problem constrained by differential algebraic equations or by PDEs. This problem has been addressed by using mixed-integer programming techniques, for contaminant detection in water networks [27, 28, 29, 30]. In these studies, an optimal set of sensor locations is selected from a set of candidate locations to minimize a certain engineering metric such as contaminant detection time, population exposure, or likelihood of detection. Likelihoods are assigned based on contamination scenarios, and not on information content of the sensor data recorded, as in a traditional experimental design setting. As a result, these approaches fail to provide statistically meaningful sensor network designs. Moreover, because the formulations capture flow dynamics by using surrogate representations such as transportation delays, they fail to capture physical effects.

Sensor placement problems have also been addressed in a more general control setting where one seeks to optimize a measure of observability such as the covariance matrix, Kalman estimator gain, or so-called observability Grammian matrix. This problem is again a bilevel optimization problem. The covariance matrix approach in [31] bypasses this by assuming that the dynamic model is linear, thus allowing the inner minimization problem to be formulated as a linear matrix inequality. The approach in [32] models the dynamics of the covariance matrix directly as a Riccati differential equation, which implicitly assumes linearity and thus enables the use of semidefinite programming algorithms. This approach, however, is focused on control policy design to extract maximum information, and not on sensor placement design. Consequently, the authors do not consider discrete decisions associated to placement. A rigorous treatment of nonlinear dynamics is presented in [33] by casting the problem as a mixed-integer nonlinear program. The authors use a genetic algorithm to deal with the inner minimization problem that computes the observability metric. A similar approach is used in [34] to address the inner minimization problem. Mixed-integer techniques have also been used in the context of information maximization for Gaussian processes and for designing Latin hypercube samples [35, 36]. These approaches, however, do not use physical models.

Recently, the sensor placement problem for systems described by PDEs has been cast as an A-experimental design problem in which the number of sensors (i.e., the design cost) is controlled by using an sparsity-inducing ℓ_0 regularization norm that is in turn approximated by using a smoothing function [8]. This compressed sensing approach was shown to be scalable and applicable to infinite-dimensional systems, but it requires tuning and can be numerically unstable. One can also formulate and solve the problem as a mixed-integer programming problem directly, but this is computationally intractable because the PDEs are in general nonconvex and because the problem has a bilevel nature.

An important application of optimal sensor location techniques is infrastructure networks (oil, water, gas, and electricity) in which large amounts of sensor data need to be processed in real time in order to reconstruct the state of the system or to identify leaks, faults, or attacks. In this work we focus on natural gas networks, which are used to transport fuel to power generation facilities and urban areas from storage and processing facilities. These networks comprise pipelines that span thousands of miles and exhibit complex dynamics. An interesting property of natural gas networks is that significant amounts of gas can be stored inside the pipelines. The stored gas is distributed spatially along the pipelines and is normally referred to as *line-pack* [37]. Line-pack is used by pipeline operators to modulate variations of gas demands at multiple spatial points in intraday operations. Some of the strongest variations in gas demands are the result of on-demand startup and shutdown of gas-fired power plants [38]. Modulating these variations is challenging because the fast release of line-pack at multiple simultaneous locations can trigger complex spatiotemporal dynamic responses that propagate hundreds to thousands of miles and that can take hours to stabilize. Therefore, line-pack management is performed by using sophisticated optimal control and pipeline simulation tools. To use these automation tools, one must reconstruct spatiotemporal state fields (flows, pressures, temperatures) [39] and natural gas leaks [40]. This task is challenging from a practical stand point given the limited amounts of sensor data (often limited to pressure and flow signals at a finite set of locations), the infinite-dimensional nature of pipeline systems, and the complex physical behavior of these systems. Such challenges are not unique to natural gas networks but also arise in other domains such as geophysics and contaminant source detection in water networks.

2.2 Distributed System Modeling

We illustrate the complexity of the optimal sensor placement problem by focusing on the physical equations describing the dynamics of natural gas networks. Details on the model derivation, nomenclature, and units used in this section can be found in [41].

2.2.1 Problem Physics

The isothermal flow of gas through a horizontal pipeline is described by the conservation and momentum equations:

$$\begin{aligned}\frac{\partial \rho(\tau,x)}{\partial \tau} + \frac{\partial (\rho(\tau,x)\nu(\tau,x))}{\partial x} &= 0\\ \frac{\partial (\rho(\tau,x)\nu(\tau,x))}{\partial \tau} + \frac{\partial p(\tau,x)}{\partial x} &= -\frac{\lambda}{2D}\rho(\tau,x)\nu(\tau,x)|\nu(\tau,x)|.\end{aligned}$$

Here, $\tau \in \mathcal{T} := [0,T]$ is the time dimension with final time T (planning horizon), and $x \in \mathcal{X} := [0,L]$ is the axial dimension with length L. The pipeline diameters are denoted as D, and the friction coefficients are denoted as λ. The states of the link are the gas density $\rho(\tau,x)$, the gas speed $\nu(\tau,x)$, and the gas pressure $p(\tau,x)$. The transversal area A, volumetric flow $q(\tau,x)$, and mass flow $f(\tau,x)$ are given by

$$\begin{aligned}A &= \frac{1}{4}\pi D^2 \qquad (2.2a)\\ q(\tau,x) &= \nu(\tau,x)\,A\\ f(\tau,x) &= \rho(\tau,x)\,\nu(\tau,x)\,A.\end{aligned}$$

For an ideal gas, pressure and density are related as follows:

$$\frac{p(\tau,x)}{\rho(\tau,x)} = c^2. \qquad (2.3)$$

Here, c is the gas speed of sound. The speed (assuming an ideal gas behavior) and the friction factor λ can be computed from

$$\begin{aligned} c^2 &= \frac{\bar{\gamma} Z R T}{M} \\ \lambda &= \left(2 \log_{10}\left(\frac{3.7\,D}{\epsilon}\right)\right)^{-2}, \end{aligned}$$

where Z is the gas compressibility factor, R is the universal gas constant, T is the gas temperature, M is the gas molar mass, ϵ is the pipe rugosity, and $\bar{\gamma}$ is the adiabatic constant. Often one desires to transform (2.1) into a more convenient form in terms of mass flow and pressure by using (2.3) and (2.2):

$$\begin{aligned} \frac{\partial p(\tau,x)}{\partial \tau} + \frac{c^2}{A}\frac{\partial f(\tau,x)}{\partial x} &= 0 \\ \frac{1}{A}\frac{\partial f(\tau,x)}{\partial \tau} + \frac{\partial p(\tau,x)}{\partial x} &= -\frac{\lambda \rho(\tau,x)}{2D}\frac{f(\tau,x)}{\rho(\tau,x)\,A}\left|\frac{f(\tau,x)}{\rho(\tau,x)\,A}\right|. \end{aligned} \tag{2.5a}$$

Substituting (2.3) and (2.2a) in (2.5a) and performing some manipulations, we obtain the more compact form:

$$\begin{aligned} \frac{\partial p(\tau,x)}{\partial \tau} &= -\frac{c^2}{A}\frac{\partial f(\tau,x)}{\partial x} \\ \frac{1}{A}\frac{\partial f(\tau,x)}{\partial \tau} &= -\frac{\partial p(\tau,x)}{\partial x} - \frac{8\,\lambda\, c^2}{\pi^2\, D^5}\frac{f(\tau,x)|f(\tau,x)|}{p(\tau,x)}. \end{aligned}$$

For numerical purposes, we define scaled flows $f(\tau,x) \leftarrow \alpha_f f(\tau,x)$ and pressures $p(\tau,x) \leftarrow \alpha_p p(\tau,x)$, where α_f and α_p are scaling factors. Scaling (2.6) and rearranging, we obtain the final form:

$$\begin{aligned} \frac{\partial p(\tau,x)}{\partial \tau} &= -c_1\frac{\partial f(\tau,x)}{\partial x}, \; \tau \in \mathcal{T}, x \in \mathcal{X} \\ \frac{\partial f(\tau,x)}{\partial \tau} &= -c_2\frac{\partial p(\tau,x)}{\partial x} - c_3\frac{f(\tau,x)|f(\tau,x)|}{p(\tau,x)}, \; \tau \in \mathcal{T}, x \in \mathcal{X}, \end{aligned}$$

where the constants c_1, c_2, and c_3 are given by

$$c_1 = \frac{\nu^2}{A}\frac{\alpha_p}{\alpha_f}, \quad c_2 = A\frac{\alpha_f}{\alpha_p}, \quad c_3 = \frac{8\,\lambda\,\nu^2\,A}{\pi^2\,D^5}\frac{\alpha_p}{\alpha_f}. \tag{2.8}$$

For subsonic flow, one must impose a boundary condition at the inlet point and a boundary condition at the outlet point. For instance one can specify pressure at the inlet and outlet points,

$$\begin{aligned} p(0,\tau) &= \theta^{orig}(\tau) \\ p(L,\tau) &= \theta^{rec}(\tau). \end{aligned}$$

One also can impose boundary conditions for inlet and outlet flows as

$$\begin{aligned} f(0,\tau) &= f^{orig}(\tau) \\ f(L,\tau) &= f^{rec}(\tau). \end{aligned}$$

Alternatively, one can impose a boundary condition for pressure at the inlet point and one for flow at the outlet point, or vice versa.

2.2.2 Discretization of State Equations

For either simulation or optimization we need to discretize equations (2.7). These equations are a particular case of a nonlinear system of equations. To that end, we introduce the vector variable

$$u(t,x) = \begin{pmatrix} p(t,x) \\ f(t,x) \end{pmatrix}, \quad (t,x) \in [0,T] \times [0,L]$$

which consists of pressure $p(t,x)$ and flow $f(t,x)$ in the system defined over the domain: the Cartesian product of $[0,T]$ in time with $[0,L]$ in space. With this notation, the governing equations of a gas pipeline can be written as the following nonlinear system of PDEs:

$$\frac{\partial u}{\partial t} + \begin{bmatrix} 0 & c_1 \\ c_2 & 0 \end{bmatrix} \frac{\partial u}{\partial x} + c_3 \begin{bmatrix} 0 \\ f|f|/p \end{bmatrix} = 0. \tag{2.11}$$

The parameters $c_1, c_2, c_3 > 0$ are defined in (2.8) and play a key role in identifying stable numerical schemes for solving (2.11). The initial conditions are given by $p(0, x) = p_0(x)$ and $f(0, x) = f_0(x)$. We use prescribed and constant pressure boundary conditions: $p(t, 0) = p_1$ and $p(t, L) = p_2$. We note that experimental validation has indicated that constant pressure boundary conditions are appropriate for gas pipeline systems [42].

We now discretize the system of PDEs (2.11). The system (2.11) is not conservative, since the friction term (nonlinear term) results in dissipation of energy. The linear part of the system (formally obtained by setting c_3 to 0) represents a conservative hyperbolic system, since all the eigenvalues of the 2×2 matrix in (2.11) are real and equal to $\pm\sqrt{c_1 c_2}$. At each point, this system has two characteristic directions each having an angle smaller than 90-degrees with one of the boundaries. To maintain stability, we use an upwinding scheme along each of the characteristics [43].

We first consider the linear part of the system:

$$\frac{\partial u}{\partial t} + B\frac{\partial u}{\partial x} = 0, \quad x \in [0, L], t \in [0, T]$$

with $B = \begin{pmatrix} 0 & c_1 \\ c_2 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 2}$. B has eigenvalue decomposition $B = S\Lambda S^{-1}$, where

$$\Lambda := \operatorname{diag}\{\lambda_1, \lambda_2\} = \begin{pmatrix} -\sqrt{c_1 c_2} & 0 \\ 0 & \sqrt{c_1 c_2} \end{pmatrix}, \quad S := \begin{pmatrix} \sqrt{c_1} & \sqrt{c_1} \\ -\sqrt{c_2} & \sqrt{c_2} \end{pmatrix}.$$

We define the characteristic variable $\tilde{u} := S^{-1}u$, which satisfies the decoupled system

$$\frac{\partial \tilde{u}}{\partial t} + \Lambda \frac{\partial \tilde{u}}{\partial x} = 0.$$

The system consists of two independent wave equations traveling in opposite directions. To separate the two waves, we introduce the splitting of the eigenvalues λ_k as

$$\lambda_k = \lambda_k^+ + \lambda_k^-, \quad \lambda_k^+ := \max(\lambda_k, 0), \quad \lambda_k^- := \min(\lambda_k, 0).$$

We can write the upwind scheme for the characteristic variable as

$$\frac{1}{\Delta t}(\tilde{u}_j^{n+1} - \tilde{u}_j^n) + \frac{1}{\Delta x}\Lambda^+(\tilde{u}_j^n - \tilde{u}_{j-1}^n) + \frac{1}{\Delta x}\Lambda^-(\tilde{u}_{j+1}^n - \tilde{u}_j^n) = 0$$

with

$$\Lambda^+ := \operatorname{diag}(\lambda_1^+, \lambda_2^+), \quad \Lambda^- := \operatorname{diag}(\lambda_1^-, \lambda_2^-).$$

Next, we define $B^+ := S\Lambda^+S^{-1}, B^- := S\Lambda^-S^{-1}$, and obtain the upwinding scheme in terms of the original variable $u(\cdot)$ by multiplying the resulting scheme by the matrix S,

$$\frac{1}{\Delta t}(u_j^{n+1} - u_j^n) + \frac{1}{\Delta x}B^+(u_j^n - u_{j-1}^n) + \frac{1}{\Delta x}B^-(u_{j+1}^n - u_j^n) = 0, \tag{2.12}$$

where

$$u_j^n = \begin{pmatrix} p_j^n \\ f_j^n \end{pmatrix} \quad B^+ = \frac{1}{2}\begin{pmatrix} \sqrt{c_1c_2} & c_1 \\ c_2 & \sqrt{c_1c_2} \end{pmatrix} \quad B^- = \frac{1}{2}\begin{pmatrix} -\sqrt{c_1c_2} & c_1 \\ c_2 & -\sqrt{c_1c_2} \end{pmatrix}.$$

Here, the notation u_j^n indicates the jth point in the spatial mesh and the nth point in the temporal mesh. Plugging these terms back into (2.12), we obtain the discretization scheme:

$$\begin{aligned}
p_j^{n+1} &= p_j^n + \frac{\Delta t}{2\Delta x}[\sqrt{c_1 c_2}(p_{j-1}^n - 2p_j^n + p_{j+1}^n)] - \frac{\Delta t}{2\Delta x}[c_1(f_{j+1}^n - f_{j-1}^n)] \\
f_j^{n+1} &= f_j^n + \frac{\Delta t}{2\Delta x}[\sqrt{c_1 c_2}(f_{j-1}^n - 2f_j^n + f_{j+1}^n)] - \frac{\Delta t}{2\Delta x}[c_2(p_{j+1}^n - p_{j-1}^n)].
\end{aligned}$$

Stability of this scheme is ensured if the corresponding scalar upwind schemes for all variables $\tilde{u}_k$ are stable, which gives the Courant, Friedrichs, and Lewy (CFL) stability condition:

$$\max_k \, |\lambda_k| \frac{\Delta t}{\Delta x} = \sqrt{c_1 c_2} \cdot \frac{\Delta t}{\Delta x} \leq 1.$$

Since we anticipate that the friction term will not dominate, we simply consider the upwinding scheme for each characteristic for the linear equation to which we add the friction term explicitly. This procedure results in the following numerical scheme:

$$\begin{aligned}
p_j^{n+1} &= p_j^n + \frac{\Delta t}{2\Delta x}[\sqrt{c_1 c_2}(p_{j-1}^n - 2p_j^n + p_{j+1}^n)] - \frac{\Delta t}{2\Delta x}[c_1(f_{j+1}^n - f_{j-1}^n)] \\
f_j^{n+1} &= f_j^n + \frac{\Delta t}{2\Delta x}[\sqrt{c_1 c_2}(f_{j-1}^n - 2f_j^n + f_{j+1}^n)] - \frac{\Delta t}{2\Delta x}[c_2(p_{j+1}^n - p_{j-1}^n)] - \Delta t \cdot c_3 f_j^n |f_j^n| / p_j^n.
\end{aligned}$$

for $j = 1, 2, ..., N_x - 1$ and $n = 0, 1, ..., N_t$. The friction term can be split among the characteristic equations based on the eigenvectors of the matrix of the linear system and the boundary. To simplify the implementation, we repeat the flux values of the last interior node: $f_0^n = f_1^n,\ f_{N_x}^n = f_{N_x - 1}^n$.

2.3 Design of Experiments Setup

2.3.1 Bayesian Framework

As mentioned in the Introduction chapter, we consider a setting with measurements perturbed by additive Gaussian noise (some dimension notations are different):

$$\mathbf{d} = \mathcal{F}(u_0) + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \Gamma)\,,$$

where Γ is the measurement noise covariance matrix and $\mathcal{F}$ is a nonlinear operator that maps a parameter vector $u_0 \in \mathbb{R}^n$ to the space-time observation vector $\mathbf{d} \in \mathbb{R}^q$. Formally,

$$\mathcal{F} : u_0 \overset{f}{\longmapsto} \boldsymbol{u} \overset{w}{\longmapsto} \mathbf{d} \tag{2.14}$$

where u_0 are the input parameters, f is the discretized PDE solution operator, $\boldsymbol{u}$ is the discretized PDE solution vector, and w is the state-to-observation operator. Note measurement error is not considered in (2.14). In our case, the input parameters u_0 (the inferred variables) consist of the initial pressure and flow at the grid points:

$$u_0 = \left\{ \{p_0(i\Delta x)\}_{i=1,2,\ldots,N_x-1}, \{f_0(i\Delta x)\}_{i=0,1,\ldots,N_x} \right\}.$$

The solution vector consists of the pressure and flow at all nodes and all times:

$$\boldsymbol{u} = \left\{ \left\{ p_j^n \right\}_{j=0,1,2,\ldots,N_x,\; n=0,1,2,\ldots,N_t}, \left\{ f_j^n \right\}_{j=0,1,2,\ldots,N_x,\; n=0,1,2,\ldots,N_t} \right\}.$$

The map f is defined by the numerical scheme (2.13). The observations $\mathbf{d}$ are a subset of entries in the solution vector $\boldsymbol{u}$, and the *space-time observation operator* w is the restriction operator from the components of $\boldsymbol{u}$ to the entries in $\mathbf{d}$. For our experimental design, we assume that the sensors are fixed and interrogated at all times, in which case the observation vector $\mathbf{d}$ and observation operator w are parameterized only by the spatial locations at which we observe the pressure and flow.

The measurement noise $\boldsymbol{\eta}$ is independent of u_0 and thus $u|u_0 \sim \mathcal{N}(\boldsymbol{f}(u_0), \Gamma_{\text{noise}})$. The likelihood is given by

$$\pi_{\text{like}}(u|u_0) \propto \exp\Big(-\frac{1}{2}\|u - \boldsymbol{f}(u_0))\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}}\Big).$$

Stating the consequence of Bayes' theorem $\pi_{\text{post}}(u_0|\mathbf{u}) \propto \pi_{\text{like}}(\mathbf{u}|u_0)\pi_{\text{prior}}(u_0)$ with a Gaussian prior $\pi_{\text{prior}}(u_0) \propto \exp\Big(-\frac{1}{2}\|u_0 - u_{\text{prior}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}\Big)$, we obtain the parameterization of the posterior distribution $\pi_{\text{post}}(u_0|\mathbf{u})$ (up to a constant) as [44]:

$$\pi_{\text{post}}(u_0|\mathbf{u}) \propto \exp\Big(-\frac{1}{2}\|\boldsymbol{f}(u_0) - \mathbf{u})\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} - \frac{1}{2}\|u_0 - u_{\text{prior}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}\Big), \tag{2.15}$$

where u_{prior} is the mean of the prior distribution, $\boldsymbol{\Gamma}_{\text{prior}} \in \mathbb{R}^{n\times n}$ is the covariance matrix for the prior which we assume to be a scalar multiple of the identity matrix. $\boldsymbol{\Gamma}_{\text{noise}} \in \mathbb{R}^{q\times q}$ is the covariance matrix for the noise. We consider two types of noise matrices $\boldsymbol{\Gamma}_{\text{noise}}$. The first one is $\text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_q^2)$ which indicates independent measurements (white noise). The second one is given in (2.16), which assumes independence in space but nonzero and decaying correlation in time (colored noise). The statistical parameters needed to define our model are $\boldsymbol{\Gamma}_{\text{noise}}$, $\boldsymbol{\Gamma}_{\text{prior}}$, and u_{prior}.

By measurement noise we understand here the discrepancy between simulated pressure and flow given exact initial and boundary information and the output of the sensor of a fixed location. For modeling this measurement noise we have several considerations. The intrinsic sensor errors can be assumed to be statistically independent between different sensors. However, some of the discrepancy between sensor indications and computed flow are also due to the numerics and boundary conditions, such as unresolved fluctuations and external perturbations. If the sensors are sufficiently far apart, we can assume that insofar as these are represented as probabilistic errors, they are spatially independent. The situation concerning the temporal features of the noise is more complicated, however. The discrepancy between measurement and simulation can be due to unresolved scales, which typically have nonzero correlation times and cannot be ignored if the measurements are frequent enough. We thus model the measurement noise as a Gaussian random variable that is independent in space but correlated in time. We also assume that its mean is zero. For the intrinsic error of a calibrated sensor, this is a reasonable assumption. Given our definition, measurement noise also includes numerical error. We assume that this and all other biases are small enough compared with the sensor error (and given the optimal variances, this is a reasonable assumption). In summary we assume that the measurement error has zero mean and covariance given by the following function:

$$\operatorname{Cov}\big((t_i, x_i), (t_j, x_j)\big) = \delta(x_i, x_j) \exp\left\{-\frac{|t_i - t_j|}{\tau_i} I_{\{|t_i - t_j| \leq \tau_j\}}\right\}, \tag{2.16}$$

where τ_i, τ_j are parameters with dimension of time that define the shape of the covariance function. Here, $\delta(x_i, x_j)$ is the Kronecker δ symbol, which takes the value 1 if $x_i = x_j$ and 0 otherwise. Γ_{noise} is then computed by evaluating the covariance functions at the position where pressure and flow are measured. In other words, entries on subdiagonals of Γ_{noise} have exponential decay, and Γ_{noise} is a sparse matrix.

We also experiment with white noise in time. This is probably the most common usage in such problems even if it does not make sense in the limit of dense temporal observations. It does, however, has the advantage of needing fewer parameters. Moreover, under proper scaling conditions, this gives a conservative approximation of the variance for target linear forms of the initial state (i.e., it overestimates the posterior variance). We thus consider the case of white noise in time and space as well, which corresponds to a constant diagonal covariance function, that is:

$$\operatorname{Cov}\big((t_i, x_i), (t_j, x_j)\big) = \delta(x_i, x_j)\delta(t_i, t_j).$$

The other element in defining a Bayesian uncertainty framework concerns the prior assumptions about the parameters to be inverted, u_0. Here we use a Gaussian prior, which is a common choice for Bayesian inverse problems [44]. The prior mean describes our best guess about the uncertainty parameter, which can be obtained from historical measurements or from other available information. In addition, because of the lack of a priori information about the parameters, we will use a prior that assumes spatial independence. This setup can be interpreted as making no assumptions about the smoothness of the initial data, which should result in conservative statements.

Despite the choice of Gaussian prior and noise probability distributions, the posterior probability distribution need not be Gaussian, because of the nonlinearity of $\boldsymbol{f}(u_0)$ [44]. If our purpose were estimation, then we would aim to characterize the posterior distribution. The mean of this posterior distribution, $\boldsymbol{m}_{\text{MAP}}$, is the parameter vector maximizing the posterior (2.15), and is known as the *maximum a posteriori* (MAP) point. It can be found by minimizing the negative log posterior, which amounts to solving the following optimization problem:

$$\boldsymbol{m}_{\text{MAP}} = \underset{u_0}{\arg\min}\ \mathcal{J}(u_0) := -\log \pi_{\text{post}}(u_0|\mathbf{d}).$$

Characterizing the posterior uncertainty, however, would require exploring and summarizing this posterior distribution, which in general can be done only with Markov-chain Monte Carlo methods [44]. To simplify computations, we use Laplace's approximation. That is, we make a quadratic approximation of the negative log of the posterior (2.15) around the MAP point. The posterior covariance matrix Γ_{post} is then given by the inverse of the Hessian of $\mathcal{J}$ at u_0. We thus approximate the posterior covariance with a Gaussian distribution, with mean $\boldsymbol{m}_{\text{MAP}}$.

2.3.2 Modeling Sensor Placement Decisions

Our interest, however, is not only in estimation but in optimal sensor placement locations. To this end we allow any of the spatial nodes $x_i,\ i = 0, 1, ..., N_x$ to be candidate sensor locations using the same discretization in x-direction as in §2.2.2. To allow the ability to select the position of the sensors, we associate with each x_i a non-negative binary weight $w_i \in \{0, 1\}$. Our intent is to denote by $w_i = 1$ the situation where a sensor is placed at location x_i and by $w_i = 0$ the situation where no sensor is placed at location x_i. Therefore, the problem of determining the optimal sensor locations becomes an large-scale mixed-integer integer nonlinear program. Our approach will be to perform relaxations of the sensor placement problem, by allowing w_i to have any value in the domain $[0, 1]$.

We model the fact that a sensor has fixed spatial placement, at which we measure both flow and pressure at all times. Nevertheless, we allow grid points on the temporal direction to have the same weight. We thus create a weight diagonal matrix corresponding to each point in $\boldsymbol{u}$:

$$W = \operatorname{diag}(w_0, w_1, ..., w_{N_x}, w_0, w_1, ..., w_{N_x}, ..., w_0, w_1, ..., w_{N_x}) \in \mathbb{R}^{2m \times 2m},$$

where $m = (N_x + 1)(N_t + 1)$ is the total number of discretized points in the domain $[0, T] \times [0, L]$. Because we allow any spatial degree of freedom to be measured, we initially assume that $\boldsymbol{f}(u_0)$ is the entire solution map $\mathcal{S}$, and we use the weights to winnow it down. Since we will end up solving an integer programming problem, we aim to produce a version of the optimal sensor placement that has a convex objective. Therefore, inspired by the workflow from [8], we approximate $\boldsymbol{f}(u_0)$ when used in (2.15) by its linearization around the prior mean u_{prior}. To this end, we denote the Jacobian of $\boldsymbol{f}$ at u_{prior} by F. We then have that

$$\boldsymbol{f}(u_0) \approx \boldsymbol{f}(u_{\text{prior}}) + F(u_0 - u_{\text{prior}}).$$

The last ingredient is to induce a weighted least squares setup to the estimation problem, to allow for a consistent statistical framework when allowing points to come in and out of the measurement set and thus decide on the optimal measurement set and sensor placement. This strategy is equivalent to scaling the variance of the measurement at a certain point x_i by $1/w_i$. For writing down the likelihood we need the inverse of the noise variance, which will now be $W^{1/2}\mathbf{\Gamma}_{\text{noise}}^{-1}W^{1/2}$. In this form, we assume that $\mathbf{\Gamma}_{\text{noise}}$ is the matrix of the noise as if sensors are at *every* grid point and are evaluated based on the covariance kernel (2.16). In this case (2.15) is proportional to the weighted least squares likelihood. That is, the w-weighted likelihood, conditional on the initial conditions $\boldsymbol{u}_0$ and weights w, is [8]

$$\pi_{like}(d|\boldsymbol{u}_0, w) \propto \exp\left\{-\frac{1}{2}(F\boldsymbol{u}_0 - d)^T W^{\frac{1}{2}}\Gamma_{noise}^{-1}W^{\frac{1}{2}}(F\boldsymbol{u}_0 - d)\right\},$$

where d is a potential measurement at all times and space points (or u, see §A). Accounting now for the prior distribution on $\boldsymbol{u}_0$ around its mean u_{prior}, we get the posterior likelihood:

$$\pi_{post}(d|w) \propto \exp\left\{-\frac{1}{2}(F\boldsymbol{u}_0 - d)^T W^{\frac{1}{2}}\Gamma_{noise}^{-1}W^{\frac{1}{2}}(F\boldsymbol{u}_0 - d) - (\boldsymbol{u}_0 - u_{\text{prior}})\Gamma_{prior}^{-1}(\boldsymbol{u}_0 - u_{\text{prior}})\right\}.$$

We note that a maximum likelihood approach would have a similar expression except that it would miss the prior term. In that case the problem would become equivalent to one of least squares. Under the assumptions above, the distribution of the best estimate $\boldsymbol{u}_0$ is normal with covariance matrix:

$$\Gamma_{post}(w) = \left(F^T W^{\frac{1}{2}}\Gamma_{noise}^{-1}W^{\frac{1}{2}}F + \Gamma_{prior}^{-1}\right)^{-1}. \tag{2.17}$$

2.3.3 Optimal Sensor Placement Formulation

The optimal sensor placement problem is cast as a design of experiments formulation. The aim is to minimize a measure of the posterior covariance matrix under the constraint of a fixed number of sensors. In this work we focus on minimizing the trace (corresponding to an A-optimality criterion) and minimizing the variance of the estimated initial flow. We capture these formulations using the following general form:

$$\begin{aligned} \min \quad & \Psi(\Gamma_{post}(w)) \\ \text{subject to} \quad & w_i \in \{0,1\}, i = 0, 1, ..., N_x \text{ and } \sum_{i=1}^{N} w_i = n_0. \end{aligned}$$

Here n_0 is the total number of sensors to be placed. To minimize the posterior covariance of total flow, we use

$$\Psi\left(\Gamma_{post}(w)\right) \equiv a^T \Gamma_{post}(w)\ a, \quad a = (\underbrace{0, \dots, 0}_{N_x - 1}, \underbrace{1, \dots, 1}_{N_x + 1}) \in \mathbb{R}^{2N_x}. \tag{2.19}$$

Here, the vector a has an entry of L/N_x corresponding to any initial flow variable, and 0 otherwise. In other words, the vector a is used to extract the flow variance from the posterior covariance matrix. The trace minimization problem considered in [8] uses

$$\Psi(\Gamma_{post}(w)) \equiv \text{Trace}(\Gamma_{post}(w)). \tag{2.20}$$

We can interpret the minimization of the trace of the posterior covariance as a compromise in aiming to reduce the variances of all possible linear functions of the initial state.

Sparsity-Inducing Approach

Because of the integrality of the sensor placement problem and the complex nonlinear structure of the measures used, direct use of off-the-shelf solvers does not result in scalable solutions. For instance, initial investigation using linearization of the mapping and *mixed-integer linear programming solvers resulted in excessive computational times* once we exceeded a mesh of $N_x = 10$. To enable scalable solutions, we propose to use a sparse (compressed sensing) optimization approach [8] and a sum-up rounding approach [26]. In the compressed sensing approach, we introduce a sparsity-inducing penalty term while relaxing the binary constraints. This results in

$$\begin{aligned} \min \quad & \Psi\left(\Gamma_{post}(w)\right) + \gamma \cdot \Phi(w) \\ \text{subject to} \quad & 0 \leq w_i \leq 1, i = 0, 1, ..., N_x \text{ and } \sum_{i=1}^{N} w_i = n_0. \end{aligned}$$

Here, $\gamma \geq 0$ is a penalty parameter, and $\Phi(\cdot)$ is a penalty function. The ideal penalty function is the so-called 0-norm, which counts the nonzero entries. For γ sufficiently large, such a norm would indeed induce an integer solution. On the other hand, this formulation makes the problem difficult, in effect NP-hard (in N_x). If the number of nonzero entries is small, however, an integer solution can be obtained with high probability by using the 1-norm (which is a continuous and convex metric). This is the basis for the recent advances in the area of compressed sensing [45]. If we insist on the constraint of the sum of weights being prescribed, however, then using $\Phi(w) = \|w\|_1$ has no effect on our problem. We have also tried to use $\Phi(w) = \|w\|_1$ without the total sum constraints $\sum_{i=1}^{N} w_i = n_0$, and chose the penalty parameter γ so that the solution of the problem satisfies $\sum_{i=1}^{N} w_i = n_0$. In the parameter ranges tried, this compressed sensing setup did not produce a sparse solution. An alternative is to use a penalty $\Phi(w)$ that is closer to the 0-norm, although this comes at the cost of abandoning convexity. Such an approach is also used in [8]. In this work we use $\Phi(w) = \|w\|_{1/2}$. We highlight that the cost function $\Psi\left(\Gamma_{post}(w)\right)$ may be nonconvex and the penalty term $\Phi(w) = \|w\|_{1/2}$ would add to nonconvexity.

Sum-up Rounding Approach

The other approach considered is the sum-up rounding (SUR) strategy of Sager. This approach starts with the convex relaxation of the optimization problem (2.18) (and formally represents the problem (2.21) for $\gamma = 0$):

$$\begin{aligned} \min \quad & \Psi\left(\Gamma_{post}(w)\right) \\ \text{subject to} \quad & 0 \le w_i \le 1, i = 0, 1, ..., N_x \text{ and } \sum_{i=1}^{N} w_i = n_0. \end{aligned}$$

An important property is that this problem produces a lower bound for the objective function of (2.18). The key is now to produce an upper bound by finding an integer vector w that satisfies the constraint $\sum w_i = n_0$ and that has an objective only slightly increased from the optimal one of (2.22). In the sum-up rounding approach, an upper bound is produced as follows. If we denote the relaxed solution of (2.22) by $\boldsymbol{w_{rel}} = \{w_{rel}^0, ..., w_{rel}^{N_x}\}$, then an integer-valued solution $\boldsymbol{w_{int}} = \{w_{int}^0, ..., w_{int}^{N_x}\}$ is obtained by:

$$w_{int}^j = \begin{cases} 1 & \text{if } \displaystyle\sum_{k=0}^{j} w_{rel}^j - \sum_{k=0}^{j-1} w_{int}^j \ge 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

for $j = 0, 1, ..., N_x$. It has been shown for optimal control problems with binary controls [46] and for the optimal selection of measurement times in time-dependent initial value problems [26], that this rounded solution will become arbitrarily close to the relaxed one when the underlying grid is chosen fine enough. The problem we apply this strategy to is different, because rounding occurs in space and in the presence of two-point boundary conditions. Hence, it is unclear whether such desirable properties persist. We investigate this situation in the following section.

2.4 Numerical Results

2.4.1 Case I: Total Flow Variance Minimization

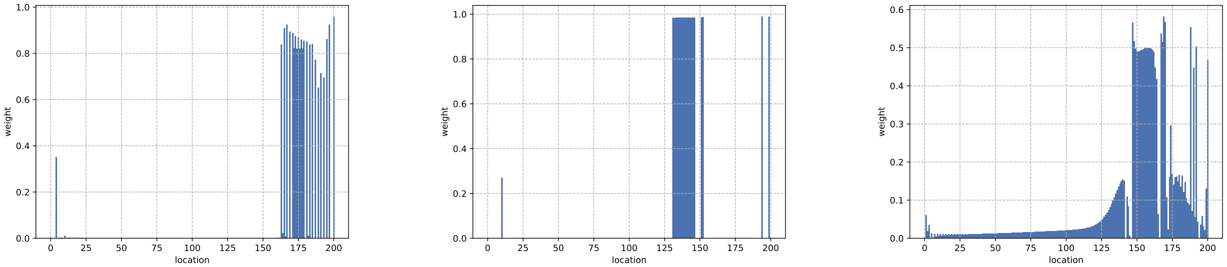


Fig.2.1: Optimal placement for sparsity-inducing (compressed sensing) approach (2.21). Flow variance minimization formulation.

The optimal design of experiments framework is first applied to the total initial flow variance minimization problem given by (2.18) with the objective function (2.19). In this case study we compare the sparsity-inducing optimization approach and the sum-up rounding strategy discussed in §2.3.3.

Sparsity-Inducing Approach

For the variance minimization problem we use $L = 80000$, $T = 60$ based on [41]. Also, using the quantities from [41], we obtain the following parameters. $c_1 = 9.6917$, $c_2 = 14137$, $c_3 = 0.0825$. To obtain a stable discretization, we must obey the CFL restrictions. We thus choose $N_x = N_t = 200$. We then solve the problem (2.21) using the `fmincon` optimization routine in Matlab. We choose as a starting point for the weights $w_i = 0.5$ for all i. We solve the optimization problem (2.21) using a constrained optimization solver. The results for white noise are displayed in Figure 2.1 for increased values of γ. We see that increased γ indeed promotes increased sparsity and a (mostly) integer solution. On the other hand, the changes in the solution for increased γ are large and irregular. We thus suspect that the sparsity-inducing solution may not get close enough to the actual solution, but we do not have enough evidence one way or the other to make a final conclusion.

Sum-Up Rounding Approach

We use $L = 80000, T = 60, c_1 = 9.70, c_2 = 1.41 \times 10^4, c_3 = 8.25 \times 10^{-2}, f_0(x) = 10, p_0(x) = -9x/L + 10$, and $\tau_1 = 2, \tau_2 = 6$ in Figures 2.2, 2.3, and 2.4. The time constant governing the dynamic response of some flowmeters is about 1 second [47]; this governs our choice of τ_1. We chose $\tau_2 = 3.0 \cdot \tau_1$ both to capture enough of the dynamic response of the sensor and to obtain a reasonably sparse covariance matrix to help with computations. We present the gap between the objective function of the relaxation (2.22) and the sum-up rounding integer value (2.23). We present the gap for white noise and colored noise cases and the gap scaled by the mesh size. Scaling here was necessary to compare problems of different sizes and to validate that the gap was shrinking with increasing mesh resolution. While the solution does not change, for the objective function to represent total variance, the vector a in (2.19) must be scaled by the mesh size, which varies with $1/N_x$. The lower and upper bounds in Figures 2.2 and 2.3 are not scaled in order to aid visualization.

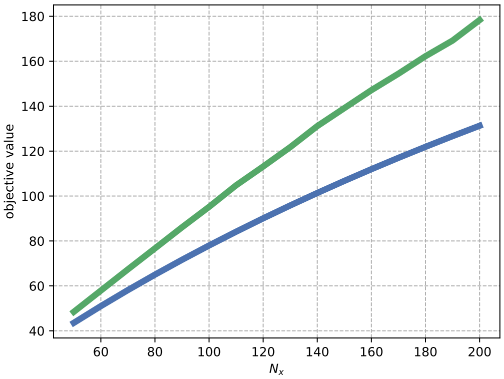


Fig.2.2: Sum-up rounding upper/lower bounds (white noise case).

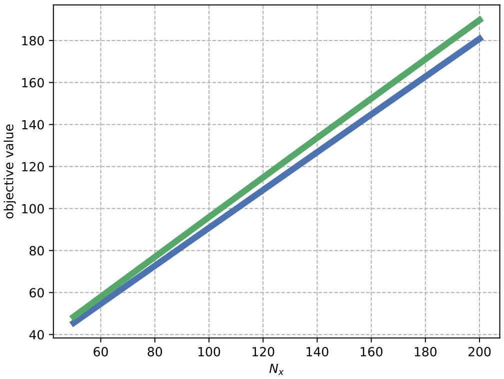


Fig.2.3: Sum-up rounding upper/lower bounds (colored noise case).

In Figure 2.5 (optimal sensor locations), the grid size used is 100 × 100 and $n_0 = 10$, which means that we have 100 candidate locations. These candidate locations are evenly distributed along the pipeline, but only 10 are selected to place sensors. In Figure 2.4, the y

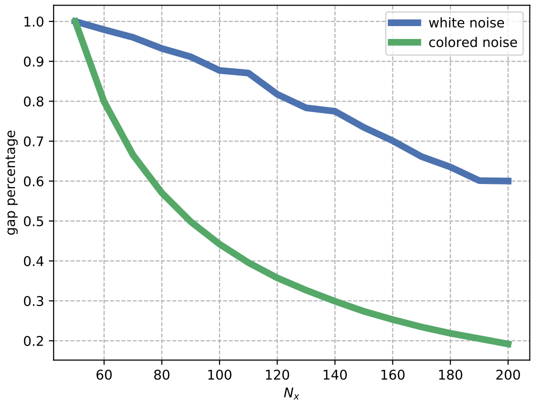


Fig.2.4: Sum-up rounding gap percentage

axis is the percentage of the gap in the objective function when N_x and N_t are determined by x relative to the gap when $x = 1$. In other words, we want to see how the gap decreases as the mesh gets more refined, since this seems to be the message in [46, 26]. The number of sensors is kept to a fixed fraction of the number of grid points, in this case $n_0 = 0.1 \times N_x$.

Given the small magnitude of the gap, we believe that the results for the optimal distribution of the sensors are reasonable. We note that the sensor distributions, while not complex, are certainly not trivial as they show that the spacing increases from inflow to outflow in the white noise case and that it reaches the maximum spacing two thirds away from the inflow in the colored noise case. We also observe that using colored noise against white noise results in a significantly smaller gap.

We also computed the total variance at the solution of the sparsity-inducing approach, and it was indeed worse. For example, for $N_x = 200$, it was worse by about 4.65%. Figure 2.1 looks much worse than that; part of the problem occurs because the total flow variance is not very sensitive to sensor placement if the time horizon is short, here being 60 s. This short time horizon was chosen because of computational time constraints. In any case, we conclude that the sparsity-inducing approach exhibits inferior performance compared with that of the sum-up rounding approach.

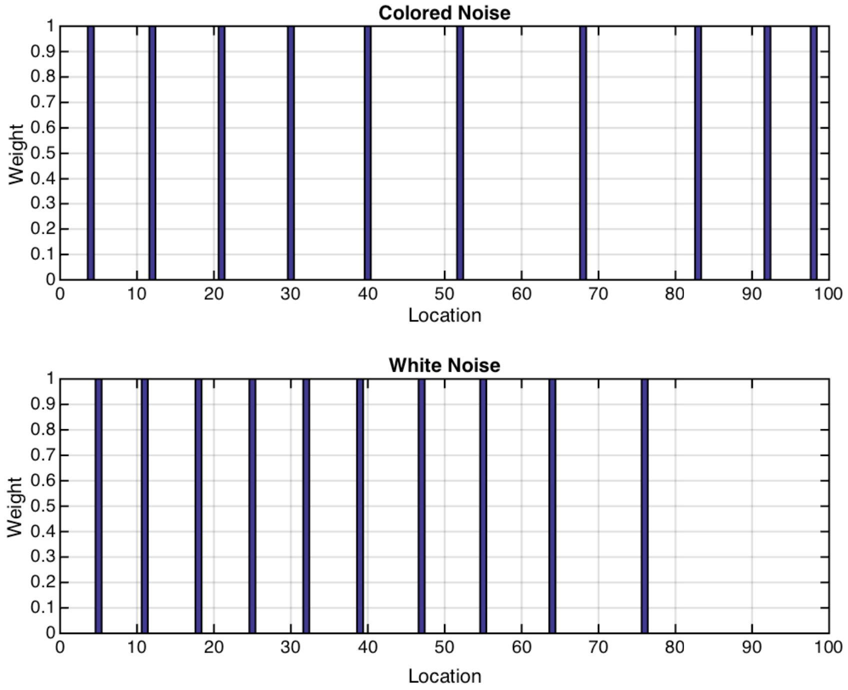


Fig.2.5: Optimal sensor locations using the sum-up rounding approach in the total flow minimization example.

2.4.2 Case II: Trace Minimization

In Figure 2.6, the solution of §2.4.1 in the case of white noise is presented. The conclusions are similar to the ones obtained by minimizing the variance of the total flow. The placement appears to be unstable as γ is varied, and subsequent comparisons with the solution of the sum-up rounding approach convinced us that the solution is far from optimum. For $N_x = 100$, the solution produced is worse by 7.5% in terms of the value of the objective (the trace of the posterior covariance). We observed a similar behavior for the colored noise approach in the few circumstances studied.

We thus focus on applying the sum-up rounding approach to the A-optimal experimental design problem (2.18) that minimizes the trace of the covariance matrix (2.20). That is, the relaxed problem (2.22) is solved followed by the sum-up rounding strategy (2.23) to produce the integer solution. Specifically, suppose w_{rel} is the solution to the relaxation (2.21) and w_{int} is the integer solution obtained from w_{rel} via the rounding-up strategy (2.23). Recall that the posterior covariance matrix is given by (2.17).

We use the notation $C_1 = \Gamma_{post}(w_{rel})$ and $C_2 = \Gamma_{post}(w_{int})$. How well we can certify that the sum-up rounding strategy works depends on how close C1 and C2 are to each other. In keeping with the intuition behind the sum-up rounding method [46], it is expected that the gap decreases as the mesh is refined, provided that the number of sensors is kept at a constant proportion of the number of nodes. One challenge faced here, however, is comparing the optimality gap for different problem instances. In the case of the variance of total flow, described in §2.4.1, this is a discretized linear form of the solution; the weights from the discretization indicate what the scaling should be, so that one can reason whether the gap is large or small from a practical perspective. The trace of the posterior variance does not naturally represent an observable quantity that can be expressed as an integral objective and in this sense this problem does not immediately fit the setting used in [46]. Because of this, the sum-up rounding strategy is used only as a heuristic. An example of the difficulty is the following. Each diagonal entry in C_i, for a fixed number of sensors, would have to converge to a fixed value (the variance of the flow at the given point), but its trace would go to infinity. To this end, the gap $C_1 - C_2$ is mapped to metrics that can be more easily interpreted. One alternative is to compute Trace$(C_1 - C_2)$ scaled by the number of points in the x direction, which is equivalent to multiplying with ΔX. Another option is to compute the largest eigenvalue of $C_1 - C_2$ also scaled by ΔX, we call this quantity the *largest eigenvalue distance*. Another option is to compute the difference between the variances for the total flow, that is, $a^T(C_2 - C_1)a\left(\frac{L}{N_x}\right)^2$, where a is the vector in the objective function of total flow. We call the absolute value of this quantity the *total flow variance distance*. In this last case, we can compare this gap with the gap from the total flow variance minimization problem §2.4.1.

We decided not to present the results for the *white noise* case because this assumption is unrealistic and the resulting figures present behaviors that are similar to those of the *colored noise* case. Moreover, as in the total flow minimization case, they exhibit a slower decrease

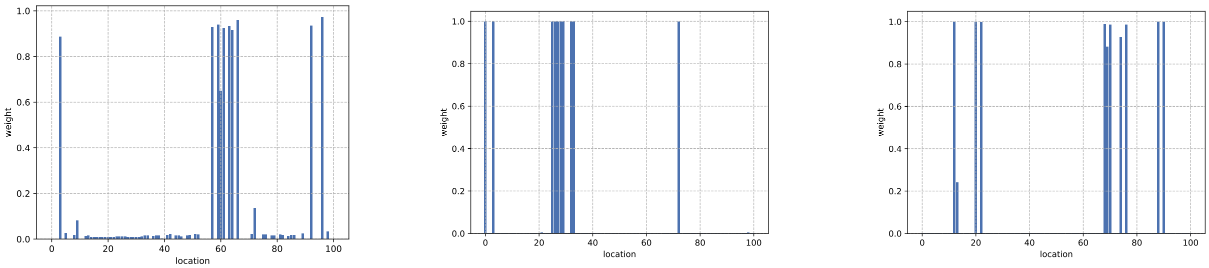


Fig.2.6: Optimal placement for sparsity-inducing (compressed sensing) approach (2.20) for the trace minimization formulation with white noise.

in the gap compared with the colored noise case.

For *colored noise*, with the measurement error covariance kernel described in (2.16), our results are presented in Figures 2.7–2.12.

- The resulting upper and lower bounds are presented in Figure 2.7 (scaled by $\frac{1}{n_p}$ in Figure 2.8) and the gap between them is presented in Figure 2.9. The x axis is $N_x = N_t = n_p$ (the number of parameters), the y axis is Trace($\Gamma_{post}(w_{int})$).

- The best sensor configuration (from the sum-up rounding strategy for $N_x = 100$) is presented in Figure 2.10.

- The largest eigenvalue distance (the eigenvalue with the largest absolute magnitude of $(C_2 - C_1)\Delta X$) is displayed in Figure 2.11. The total flow variance distance is presented in Figure 2.12 (this is equivalent to computing $a^T(C_2 - C_1)a\left(\frac{L}{N_x}\right)^2$ with the notation from (2.19)). On the x axis we display $N_x = N_t$.

From the numerical experiments, we conclude the following. As N_x increases, the gap shrinks. The most convincing evidence we find is the largest eigenvalue discrepancy plots in Figure 2.11, particularly when corroborated with the scaled upper and lower bounds calculations from Figure 2.8. Note that despite the fact that Trace$(C_2 - C_1)\Delta X$ is almost constant, in the spectral norm $(C_2 - C_1)\Delta X$ is decreasing significantly, and the rate is faster than $\frac{1}{\sqrt{N_x}}$ (Figure 2.11). The scaled gap, in particular, does not appear to converge (Figure 2.9). This also shows that, from the comparison with the largest eigenvalue distance

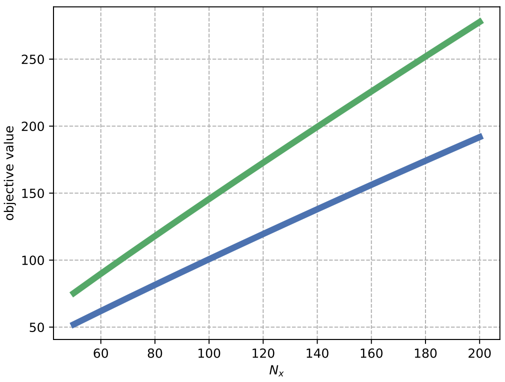


Fig.2.7: Upper/lower bounds for trace minimization case.

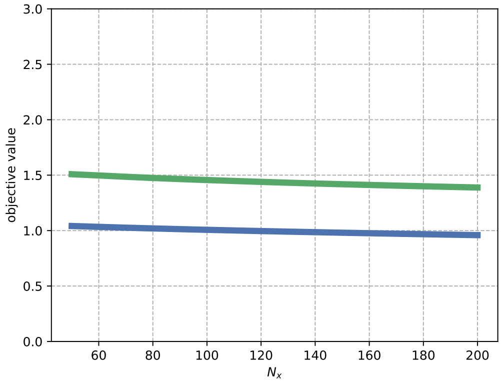


Fig.2.8: Scaled upper/lower bounds for trace minimization case.

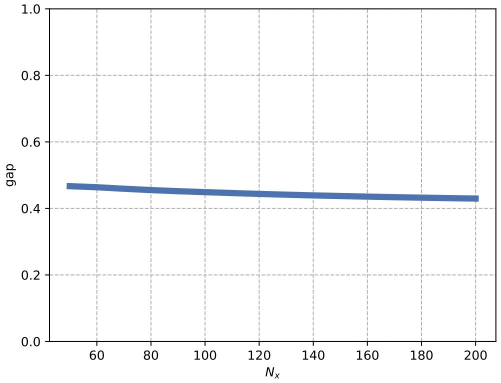


Fig.2.9: Scaled gap for colored noise, trace minimization case.

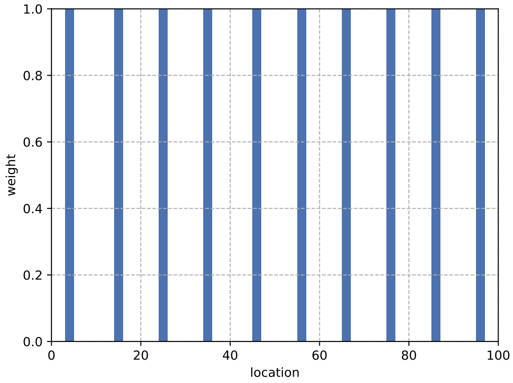


Fig.2.10: Optimal sensor placement for colored noise, trace minimization case.

behavior, scaling the trace by its size is not the right approach. The right one will be the subject of future research and is an intriguing problem in itself. We also display the total flow variance distance in Figure 2.12. In this case, the proper scaling is clear, since it represents differences of statistics of physical random variables. We observe a decrease that is comparable to that in the total flow minimization optimization case. Specifically, we have a decrease by a factor of 5, between the case $N_x = 50$ and $N_x = 200$, whereas in this trace minimization case we have a decrease by a factor of 4.5 between the same mesh sizes in the total flow variance distance.

The plots for the optimal solution, shown in Figure 2.10, indicate that a uniform distri-

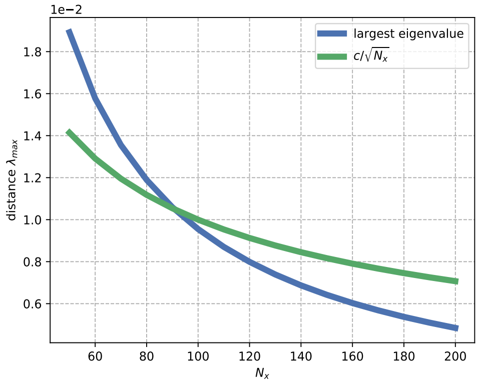


Fig.2.11: Largest eigenvalue distance between relaxed and sum-up rounding covariances for colored noise, trace minimization case.

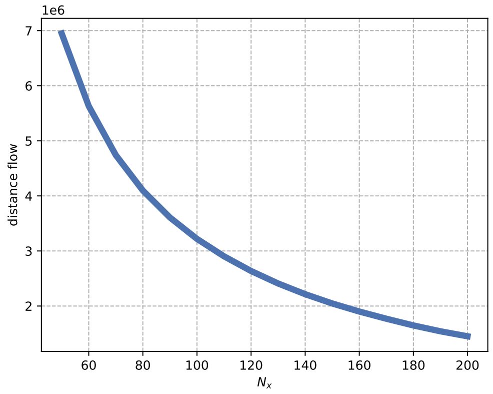


Fig.2.12: Total flow variance distance between relaxed and sum-up rounding covariance for colored noise, trace minimization case.

bution is a reasonable approximation. While the patterns show some spatial variability, it is small compared with the gap to indicate that the exact solution would not be uniform or close to it. In that sense this approach does not bring new insights, with the exception that the approach indicates that the optimal placement will not include samples at the endpoints (again, an otherwise understandable conclusion given such results from approximation theory). On the other hand, the endpoints are places where in practice the industry will most likely have sensors. Thus, the second iteration of this may be to assume that sensors exist at the endpoints and to solve the new problem. Our guess is that the result will be close to uniform distribution again. Our interpretation of such effects comes from the fact that minimizing the trace of the covariance is a bit like minimizing for the variance of all possible linear forms of the initial state (in effect, one can prove this is precisely the average of variances over random uniform choices of a over a sphere). With this interpretation it is perhaps not surprising that a close-to-uniform distribution ensues. As a check suggested by a referee, we tried each best solution we found for the total variance minimization and the trace minimization setups in the other problem, and we found that they had larger objective values, but not by much (less than 1%). Therefore, in the parameter setup we tried, the minima appear to be shallow, and uniform designs appear to be acceptable. We suspect that an issue may be the limited time horizon we can consider because of the intensive nature of the computation (it takes about a day to solve the problem in Matlab for 200 mesh points and a 60-seconds horizon on a laptop). More extensive simulations are needed to elucidate such initial questions.

2.5 Discussion

We concluded that in our setup the sparsity-inducing approach did not produce an integer solution of sufficient quality. Because we do not know the exact solution, we came to this conclusion by examining the stability of the approach with respect to the shrinking parameter and by comparing it with the results of the sum-up rounding strategy. We observed that the latter produced good results particularly in the increasing discretization case, as expected because of the interpretation of the approach as being provably optimal in the limit of increasingly accurate discretizations of a continuous limiting case [46]. For the total flow variance minimization case, the gap in the relaxation and the sum-up rounding integer solution decreases roughly as $\frac{1}{\sqrt{N_x}}$, where N_x is the number of points in the x direction. This indicates indeed an increasingly accurate solution.

For the A-optimal design (the trace minimization problem) the comparison is more complicated, because classical design of experiments theory has a framework for increasing the number of observations but not for increasing the dimension of the problem in terms of comparing the objectives of solutions for different problem sizes. We experimented with different comparison metrics, the most satisfying of which we found to be the largest absolute value of the eigenvalue of the difference between the covariance of the relaxed solution and that of the integer-valued problem. We found that in this metric the sum-up rounding strategy also results in decreasing the optimality gaps with problem size in the same $\frac{1}{\sqrt{N_x}}$ fashion.

Comparing the white noise with the colored case, we observed that the solution of the white noise case is harder for both total flow variance minimization and A-optimal design, in the sense of the metrics of convergence being larger and slower to converge, although not significantly so. Since the colored-in-time noise case is more realistic for large data (even if the white noise case is more common in the literature), this is a fortunate occurrence in our opinion. A good outcome for white noise situations is also not conceptually covered by the 1D ansatz in [46], since in this case the embedding problem is closer to a two-dimensional one. But the optimality gap shrinks as the mesh is refined in the white noise case as well.

In the total flow variance case the optimal sensor placement solution departs significantly from the uniform distribution, whereas in the A-optimal design the optimal solution appears close to uniform. While our approach presents the first computational evidence of this fact for our target problem class, it is certainly disappointing, although given the worst-case flavor of the A-optimal design perhaps not entirely unexpected. Of course when experimenting with different regimes this conclusion may change, but this is how the evidence sits at the moment.

While many more experiments need to be run, the behavior of the total flow optimal sensor placement suggests that the optimal solution while focusing on application-specific target functions of the unknown flow signal may result in nontrivial geometrical patterns of sensor placement. This, however, may not be desirable because the objective function may change with usage and redeployment of sensors. Therefore, a conservative approach would be to use an A-optimal design objective. But when desired, it appears to be an interesting direction of investigation in terms of expected outcome.

3 Theoretical Convergence on Optimality Gap

This section provides theoretical results on the zero convergence of integrality gap. In §3.2, we review the Bayesian framework and the mixed-integer nonlinear program formulation, and make a connection to integral operators. The *sum-up rounding* (SUR) procedure is defined in §3.3 based on a two-level meshing decomposition, and we give its properties of approximation in multiple dimensions. In §3.4, we show convergence of the integrality gap based on SUR strategy for different experimental design criteria, with identity covariance matrix in the prior. We provide simulation results in §3.5 on two-dimensional gravity surveying and compare with thresholding designs. In §3.6, we extend the convergence results to a Gaussian prior with Laplacian precision matrix, and give a different formulation where the parameters are the truncated coefficients of basis in a function space in §3.7.

3.1 Introduction

Design of experiments (DOE) aims to determine experimental settings that yield accurate results for statistical model parameters. One important branch of DOE seeks to determine the optimal sampling locations given a set of available measurement points (see [5, §7.5] and [6, §9, §12]). In [5], the goal is to select m regression vectors with replacement from a prescribed set of p regression vectors, so as to obtain best ordinary least squares (OLS) estimates. The optimality criteria are based on the trace, log-determinant, or maximum eigenvalue of the covariance matrix of OLS estimates. This is an integer programming problem, and it is generally NP-hard [48]. One tractable approach is to first solve the convex problem obtained from relaxing the integrality constraints, and then round the solution off to an integer one. In [6], the setting is also linear, where measurements are selected from an infinite set of regression vectors, allowing for repeated measurements. Several efficient rounding-to-integrality procedures are proposed and an analysis of asymptotic performance loss is given. A common feature of all these approaches is that the analysis is done with respect to a fixed number of model parameters.

Our focus of investigation is related to such previous endeavors but takes a different direction. Instead of a linear relationship between response (output) and parameters (input) in fixed and finite dimensions, our measurement of response is determined by the discretization of an integral functional of distributed parameters. The unknown quantity is a function that belongs to an infinite-dimensional space, which is approximated by discretization on increasingly fine meshes. Here, we aim to understand the asymptotics of the rounding procedure in the limit of the mesh size going to zero. As a result, the inverse Fisher information matrix we try to minimize (with respect to a given design criterion, such as its trace) increases in size with the number of discretization points, which makes analysis with common design criteria difficult (§3.2.5). We are not aware of prior theoretical work on the convergence analysis of discretized design of experiments with a number of sites that can grow unboundedly. Moreover, we assume here–as would be the case in many physical settings–that each data site is measured only once, so repeated measurements (as in [5, 6]) are not allowed. This would be the case, for example, if the problem is time dependent and thus a certain point in space cannot be revisited at the same instant in time or if the sensor error is constant in time but has mean zero over the sensor population, as is typical of physical sensors [49, §34.3].

Since we aim to determine the optimal sensor locations starting from a relaxed problem, the construction of an integer solution with appropriate rounding strategies of the relaxed version is a critical endeavor. Numerous rounding heuristics are given in the literature (see [50, 51, 52]), and some specifically aim for binary variables (see [53, 54, 55]). In [50], the author studied the optimal rounding by recording and comparing empirical success rates, defined as the percentage of "roundable relaxation" optima (in the words of [50]), for different types of optimization problems (mixed-integer quadratically constrained program, mixed-integer nonlinear program, etc.) among the existing rounding strategies. Classical mixed-integer techniques have been used specifically for sensor placement aiming at detecting contamination in water networks (see [56, 57, 58]) but focusing mainly on a fixed-sized discretization without investigation of limiting properties. Closer to the continuously indexed (in the limit) framework in this paper, sensor placement for systems governed by partial differential equations has been studied using a Bayesian framework [8]. In that case, the discrete nature of sensor placement problems was recovered by seeking sparsity in the solution of the relaxed problems by means of an l_0 penalty that is approximated by a sequence of smooth functions. This approach can be applied to infinite-dimensional problems, but the numerical results can be unstable, and they depend on the choice of various tuning parameters. All the rounding approaches described in this paragraph have shown good performance for certain classes of problems, including the type studied here, but their asymptotic properties have not been investigated theoretically.

Since we are interested in problems that can be continuously indexed, we investigate an extension of *sum-up rounding* (SUR). Sum-up rounding for binary variables, as we also pursue here, has been shown in temporally indexed problems to have the desirable asymptotic property of being arbitrarily close to an integer solution as long as the discretization mesh is sufficiently fine [22, 21]. In [21], the authors not only clarify the role of SUR in MIOCPs but also obtain a guaranteed bound on the performance loss, depending on the size of discretization mesh. In [22], a specific structure in one dimension is considered where the objective is a function of either the Fisher information matrix or its inverse, and the optimality gap converges to zero. Recently we used SUR as a heuristic for the sensor placement problem in natural gas pipelines governed by systems of nonlinear hyperbolic differential equations. We observed convergence of the integrality gap as the spatial mesh was progressively refined [59]; but since the spatial problem had a different nature from [21], we did not have theory to justify that observation. That was one of the main motivators for this work.

Here, we investigate DOE based on a Bayesian framework for parameter estimation [8], and we minimize functions of the posterior covariance matrix based on common experimental design criteria [6]. Our parameter to the observations map is based on an integral equation, as opposed to the solution of a partial differential equation as in [8], although the two are conceptually equivalent if one considers the Green function resolvent with the prior interpreted as a regularization term [60]. The resulting DOE problem after spatial discretization is a convex mixed-integer program; see §3.2.5. After solving the relaxed problem, we define and employ a multidimensional SUR procedure inspired by the one-dimensional procedure proposed in [22, 21]. Our main objective is to investigate whether the integrality gap between the DOE criteria at the rounded solution and relaxed solution converges to zero in the limit of zero mesh size, as was observed for MIOCPs in [22, 21]. Our contributions consist of proposing an extension of the SUR rounding procedure in multiple dimensions and proving that, for common experimental design criteria, the integrality gap converges to zero as the mesh size shrinks to zero. The techniques we employ to this end are related to the spectral theory of self-adjoint integral operators [23]. We emphasize that questions about the asymptotic quality of DOE solutions over varying design space size have not been investigated in classical DOE theory [6].

While inspired from the idea of SUR in [21] and using it as a building block, this work is different in several respects. First, applying it in a multidimensional setting allows for a larger number of rounding options and our theory covers a fairly general setup based on what we call compatible two-level domain decomposition schemes. Also, while the SUR technique itself works for rectangular domains, (which in effect, we argue in the construction at the end of §3.3.2), the proof in [21] relies on the convergence of one-dimensional integrals which would not directly apply to more than one dimension. While in the end, for implementation simplicity, our examples are for rectangular domains as well, the theoretical framework itself allows in principle a broad set of domain shapes and other rounding techniques, another example of which we give in Appendix B. Second, the functions we optimize here, which define the objective of the experimental design, depend on the posterior covariance matrix, whereas the entries in the precision matrix (the inverse of the covariance matrix) are the ones related to an integral quantity for which the typical SUR analysis applies. To carry out our the gap convergence analysis for experimental design requires the investigation of SUR effects on the eigenvalues of the precision and covariance matrices. Moreover, the sizes of these matrices go to infinity, which poses additional obstacles to the convergence analysis as we discuss in §3.4, whereas results in [22] primarily address a fixed dimensional parameter space, and thus, covariance matrix.

3.2 Estimation Framework

While the contribution of this work concerns primarily the behavior of the SUR-induced integrality gap, some of the assumptions we make stem from the estimation framework itself. In particular, our results are tied to a common but specific choice of the covariance matrices as well as to a limiting interpretation in terms of a certain integral operator. In the latter case, the integer programming relaxation needs to be interpreted in an extended output space. We thus describe the estimation framework that we use to define our DOE problem. The setup is based primarily on [8].

3.2.1 Parameter-to-Observable Map

Consider the input domain $\Omega_{in} \subset \mathbb{R}^Q$ and output domain $\Omega_{out} \subset \mathbb{R}^P$, both of which are compact sets. Suppose the output without measurement error depends on the input through an integral equation:

$$u(x) = \int_{\Omega_{in}} f(x,y) u_0(y) \,\mathrm{d}y, \quad x \in \Omega_{out},$$

where $f(x,y)$ is prescribed by the physical constraints in the setup; we thus assume it is known. The output $u(x)$ can be measured at selected points but is affected by measurement error. Our goal is to infer the parameter vector u_0 from the observation vector u. Equation

(3.56) defines a parameter-to-observable map.

To create a finite-dimensional approximation we now discuss a simple discretization strategy. More advanced discretization approaches as in [61] could easily be incorporated but would complicate the presentation whose focus is on the SUR approximation properties for DOE. We divide $D = \Omega_{in}$ (or an approximation of Ω_{in}) into m subdomains $D_1, D_2, ..., D_m$ with equal size $\mu(D_i) = \Delta_y = \mu(\Omega_{in})/m$ for $i = 1, 2, ..., m$ (as is done, e.g., for versions of Nyström's method in [62]). Then, we select a representation point y_i in each D_i and represent the input function u_0 as the finite-dimensional vector $\hat{u}_0 = \big(u_0(y_1), u_0(y_2), ..., u_0(y_m)\big)$. Similarly we divide $V = \Omega_{out}$ into n subdomains $V_1, V_2, ..., V_n$ with equal size $\mu(V_j) = \Delta_x = \mu(\Omega_{out})/n$ for $j = 1, 2, ..., n$ and select a representation point x_j for each V_j. Then we represent the continuous output u as the vector $\hat{u} = \big(u(x_1), u(x_2), ..., u(x_n)\big)$. These $x_1, x_2, ..., x_n$ points are also the candidate locations to place sensors. We approximate the integral from (3.56) by the Riemann sum:

$$u(x_j) = \int_{\Omega_{in}} f(x_j, y) u_0(y) \, \mathrm{d}y \approx \sum_i f(x_j, y_i) u_0(y_i) \Delta_y.$$

To write it in matrix form, we define $F \in \mathbb{R}^{n \times m}$ with $F(j, i) = f(x_j, y_i)\Delta_y$, and then $\hat{u} = F\hat{u}_0$. Here $\hat{u}$ and $\hat{u}_0$ represent the discretized output and input respectively.

We note that in applications the function $f(x, y)$ in (3.56) may not always be continuous. For example, when the function f encapsulates wave dynamics, it is represented by a Dirac functional $f\big((x, t), y\big) \equiv \delta(y, x - at)$, where a is the wave speed. For the remainder of this work, we assume f to be continuous. Another restriction in (3.56) is that $u(x)$ depends linearly on $u_0(x)$, which is not the case in nonlinear relationships, such as for pipeline gas dynamics [59]. In that case, the target problem can be approximated in the framework of (3.56) by linearization, as was done in [8, 59].

In the rest of this work, we use $\delta(x)$ to denote the Kronecker δ symbol:

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0, \\ 0, & \text{otherwise.} \end{cases}$$

3.2.2 Bayesian Estimation Framework

Our goal is to estimate the parameter vector $\hat{u}_0$ as a proxy for the unknown function u_0. We consider a Bayesian framework where $\hat{u}_0$ is the parameter vector to be estimated and the measurements $\hat{u}$ are data perturbed by noise. Similar to [8, 59], we assume that both the parameter prior and the measurements distributions are Gaussian:

$$\begin{aligned} \hat{u}_0 &\sim N(u_{pri}, \Gamma_{\text{prior}}), \\ \hat{u} &= F\hat{u}_0 + \eta, \text{ where } \eta \sim N(0, \Gamma_{noise}). \end{aligned}$$

Here, Γ_{pri} and Γ_{noise} represent the prior and measurement noise covariance matrices, respectively, whereas u_{pri} is the prior mean. We assume the measurement error to be unbiased conditional on the realization of u_0, and thus η has mean 0. From Bayes' rule, the posterior distribution of $\hat{u}_0$ is also Gaussian and has (up to a constant) the following density:

$$\pi_{post}(\hat{u}_0|\hat{u}) \propto \exp\Big\{ -\frac{1}{2}(\hat{u} - F\hat{u}_0)^T \Gamma_{noise}^{-1}(\hat{u} - F\hat{u}_0) - \frac{1}{2}(\hat{u}_0 - u_{pri})^T \Gamma_{\text{prior}}^{-1}(\hat{u}_0 - u_{pri}) \Big\}.$$

As mentioned in the Introduction and Chapter 2, we quantify the sensor placement effect in the posterior by creating a weight vector $w = (w_1, w_2, .., w_n) \in \{0, 1\}^n$ where the jth component w_j corresponds to candidate location x_j in the output domain. Let W be a diagonal matrix with weight vector w on its diagonal. The w-weighted posterior likelihood, conditional on the data u and weight vector w, is

$$\begin{aligned}\pi_{post}(\hat{u}_0|\hat{u},w) \;\propto\; & \exp\Big\{ -\tfrac{1}{2}(\hat{u}-F\hat{u}_0)^T W^{1/2}\Gamma_{noise}^{-1}W^{1/2}(\hat{u}-F\hat{u}_0) \\ & -\tfrac{1}{2}(\hat{u}_0-u_{pri})^T\Gamma_{\text{prior}}^{-1}(\hat{u}_0-u_{pri})\Big\}.\end{aligned}$$

Under these assumptions and accounting for the prior distribution, we can compute the posterior $\hat{u}_0$, which is the normal distribution $N(u_{post},\Gamma_{post})$, where

$$u_{post}=\Gamma_{post}\Big(F^T\Gamma_{noise}^{-1}\hat{u}+\Gamma^{-1}u_{pri}\Big),\ \Gamma_{post}=\Big(F^TW^{1/2}\Gamma_{noise}^{-1}W^{1/2}F+\Gamma_{\text{prior}}^{-1}\Big)^{-1}$$

are the posterior mean and covariance matrix, respectively. In this estimation model, the posterior covariance matrix does not depend on data $\hat{u}$. In other words, the optimal sensor placement is determined by the parameter-to-observable map and two Γ matrices.

3.2.3 Choice of Covariance Matrices

We assume that, conditional on the true $\hat{u}$, the measurement errors are independent. In most physical processes and sensor systems this is a reasonable assumption [63]. Consequently, the matrix Γ_{noise} is diagonal and commutes with W and all its positive powers, resulting in the expression

$$u_{post}=\Gamma_{post}\Big(F^T\Gamma_{noise}^{-1}\hat{u}+\Gamma^{-1}u_{pri}\Big),\ \Gamma_{post}=\Big(F^T\Gamma_{noise}^{-1}WF+\Gamma_{\text{prior}}^{-1}\Big)^{-1}.$$

In particular, the precision matrix (the inverse of the covariance matrix) becomes *linear* in W, which considerably simplifies our calculations and analysis. We assume identical sensors, and therefore $\Gamma_{noise}=\sigma_{noise}I_n$ for some prescribed sensor noise standard deviation σ_{noise}. The other covariance matrix that needs to be selected is the one corresponding to the prior distribution. Here we use a multiple of the identity $\Gamma_{\text{prior}}=\sigma_{pri}I_m$. This choice can be interpreted as ridge regression [64] or Tikhonov regularization of an inverse problem [65]. While for some setups our choice is not the ideal prior [65] it is one of the most common choices, at least before significant collection of data.

Our analysis is tied significantly to these choices, and particularly so for the prior where other reasonable choices may be available. On the other hand, this is one of the most common choices in statistical analysis of inverse problems [65]; therefore our setup does represent many problems of interest.

3.2.4 Connection to Integral Operators

With the covariance choices specified in §3.2.3, the precision matrix, the inverse of the posterior matrix Γ_{post}, becomes

$$\Gamma_{post}^{-1} = \sigma_{noise}^{-1} F^T W F + \sigma_{pri}^{-1} I_m.$$

Note that the (i, j)th entry in Γ_{post}^{-1} is

$$\Gamma_{post}^{-1}(i, j) = (\Delta_y)^2 \sigma_{noise}^{-1} \sum_{k=1}^{n} f(x_k, y_i) w_k^n f(x_k, y_j) + \sigma_{pri}^{-1} \cdot \delta(x_i - x_j), \tag{3.1}$$

with w_k^n being the weights from the diagonal of W. With reference to the notations from §3.2.1, we denote by $w^n(x)$ the piecewise constant function defined as $w^n(x) = w_k^n$, $x \in D_k$, which is the discretized area corresponding to kth candidate location in Ω_{in}. Assume that there is a measurable function $w(x) : \Omega_{out} \to [0, 1]$ such that $w^n(x) \to w(x)$ in L^1. For purposes of illustration we assume that $w^n(x)$ converges in this subsection; that will not be required in our results in §3.4. Then, if $\Delta_x, \Delta_y \to 0$ with Δ_y/Δ_x constant, the first term in (3.1) will converge to

$$\Delta_y \Big(\frac{\Delta_y}{\Delta_x}\Big) \sigma_{noise}^{-1} \int_{\Omega_{out}} f(x, y_i) w(x) f(x, y_j) \, \mathrm{d}x. \tag{3.2}$$

This quantity relates to the discretization of an integral operator

$$\mathcal{L}u_0(z) = \Big(\frac{\Delta_y}{\Delta_x}\Big)\sigma_{noise}^{-1}\iint_{\Omega_{out}\times\Omega_{in}} f(x,z)w(x)f(x,s)u_0(s)\,\mathrm{d}x\mathrm{d}s, \quad z \in \Omega_{out}. \tag{3.3}$$

Note that if $\Delta_x = \Delta_y$, then (3.2) is one coefficient of the discretization of (3.3) along the input variable s. If $w(x)$ is nonnegative, then the eigenvalues of $\mathcal{L}$ are nonnegative. Because $\mathcal{L}$ is a compact operator [23], it has a countable spectrum with 0 its only accumulation point. Moreover, because of its integral form, its trace is finite [66]. This prompts the hypothesis that the spectrum of Γ_{post}^{-1} is related to the spectrum of $\mathcal{L}$ and σ_{pri}. Specifically, eigenvalues of $\sigma_{noise}^{-1}F^TWF$ approach eigenvalues of $\mathcal{L}$ [66] in the limit of Δ_x, Δ_y going to 0 at a fixed ratio. This indicates that the eigenvalues of Γ_{post} will approximately be $1/(\lambda + \sigma_{pri}^{-1})$, where λ are eigenvalues of $\mathcal{L}$. This insight, with mathematical statements that will be made more rigorous in §3.4, allows the analysis of optimization problems whose objectives are functions of the spectrum of Γ_{post}, as is the case for the DOE problems described in §3.2.5.

3.2.5 Design of Experiments Problems

We are ready to formulate our DOE problem that addresses the issue of optimal sensor placement. We aim to minimize the estimation error of the parameter $\hat{u}_0$, which can be quantified by using its posterior covariance matrix, $\phi(\Gamma_{post})$. The three most widely used criteria in experimental design to measure the size of this error are [6]

- A-optimal design: $\phi(\Gamma_{post}) = tr(\Gamma_{post})$;
- D-optimal design: $\phi(\Gamma_{post}) = det(\Gamma_{post})$;
- E-optimal design: $\phi(\Gamma_{post}) = \lambda_{max}(\Gamma_{post})$.

Lemma 3.2.1. *$tr(\Gamma_{post})$, $\log\det(\Gamma_{post})$ and $\lambda_{max}(\Gamma_{post})$ are convex functions in the weight vector w.*

Proof. The posterior matrix can be written as

$$\Gamma_{post}(w) = \Big(\sigma_{noise}^{-1} \sum_{i=1}^{n} w_i F_i F_i^T + \sigma_{pri}^{-1} I_m\Big)^{-1},$$

where F_i is the ith column of F^T. The desired results follow because $tr(X^{-1})$, $\log\det(X^{-1})$ and $\lambda_{max}(X^{-1})$ are all convex in X [5, Exercise 3.26], and the fact that X is affine in w. □

We formulate the DOE problem as follows (ϕ represents one of the three criteria, and we use *logdet* for D-optimal design):

$$\begin{array}{ll} \min & \phi(\Gamma_{post}(w)) \\ \text{s.t.} & w_i \in \{0,1\},\ \sum_{i=1}^{n} w_i = n_0, \end{array} \tag{3.4}$$

where n_0 is the number of sensors. To avoid the complexity of integer programming, we start by examining the relaxed problem obtained by relaxing the integrality constraint,

$$\begin{array}{ll} \min & \phi(\Gamma_{post}(w)) \\ \text{s.t.} & 0 \leq w_i \leq 1, i = 1, 2, \ldots, n,\ \sum_{i=1}^{n} w_i = n_0, \end{array} \tag{3.5}$$

whose solution we denote by w_{rel}. Problem (3.5) is convex from Lemma 3.2.1. It can be solved, after using some standard semidefinite programming reformulations, by interior-point algorithms [5]. The relaxed solution w_{rel} provides a lower bound to the optimal objective of the convex integer program (3.4).

Our results will apply for any n_0 (and its value could also change with the number of discretization domains n), but they would be most meaningful in certain ranges. An examination of (3.1) indicates that if f is bounded by C, then the trace of the discretization of the integral operator is nonnegative and upper bounded by $n_0 n C^2 \Delta_y^2$. We must have $n\Delta y = O(1)$ since $n\Delta y$ must be the volume of the initial set V. Therefore, for the estimation problem to carry information comparable to the prior, we need to have $n_0 \Delta_y = O(1)$; that is, n_0 must be of comparable order with n. Otherwise the contribution from ϕ would originate in the limit exclusively from the prior. In other words, a meaningful asymptotics is the one where the number of sensors is in a fixed ratio with the number of mesh domains. This is the corresponding constraint to the one in [22] whereby the measurement time is proportional to the considered time range $[0, T]$.

3.3 Sum-up Rounding Strategy in Multiple Dimensions

In this section we describe a sum-up rounding procedure that maps the fractional vector w_{rel} solution of (3.5) into an integer vector w_{SUR} in a way that ensures the spectrum of $\Gamma_{post}(w_{rel})$ and $\Gamma_{post}(w_{SUR})$ are not too far from each other. In turn, this will ensure that the gap $\phi(\Gamma_{post}(w_{SUR})) - \phi(\Gamma_{post}(w_{rel}))$ stays small.

Our procedure is presented here for rectangular domains V (i.e., Ω_{out}, but the same construction can be applied to Ω_{in}), divided into n subdomains $V_1, V_2, ..., V_n$ of equal size $\mu(V_k) = \Delta_x = \frac{\mu(V)}{n}$. Given the function $w^n(x) : V \to [0, 1]$, which is constant on each V_i, we construct a 0-1 valued function $\tilde{w}^n(x)$ that is also constant on each V_i such that the two sums

$$S_1^n = \sum_{k=1}^{n} f(x_k) w^n(x_k) \Delta_x, \qquad S_2^n = \sum_{k=1}^{n} f(x_k) \tilde{w}^n(x_k) \Delta_x \tag{3.6}$$

are arbitrarily close to each other as long as n is large enough. Our analysis is centered around estimating the variation in the entry i, j of Γ_{post}^{-1} following the SUR procedure. The bounding technique will end up being uniform in i, j. To simplify our exposition, we ignore in the rest of the analysis the argument y of f in (3.2) since it has no effect on our approach.

Note that the function f need not be the same as the one defining the integral equation (3.56), and it can be any function defined on Ω_{out} satisfying certain continuity conditions. If $V \subset R$, this is essentially a one-dimensional time domain problem that has already been studied in [21]. In multiple dimensions, we can flatten the multidimensional vector and apply the basic sum-up rounding. However, the integration-by-part technique in the proof of [21, Theorem 2] becomes problematic in multiple dimensions, and this is why we resort to a two-level decomposition which also covers the basic one-dimensional case. It is worth mentioning that depending on the ordering of entries, we can obtain different integer vectors. In this section, we discuss the basic sum-up rounding strategy in §3.3.1 where Lemma 3.3.1 is an analogue to [21, Theorem 3]. The multidimensional strategy and its properties are given in §3.3.2 and §3.3.3 respectively, and Theorem 3.3.4 in §3.3.3 is an extension of [21, Theorem 2].

3.3.1 Properties of Basic Sum-up Rounding Strategy

We denote $\tilde{w}_i^n$ (w_1^n) as the value of $\tilde{w}^n(x)$ ($w^n(x)$) in V_i and construct the binary function $\tilde{w}^n(x)$ from $w^n(x)$ as follows.

(1) Compute $I_1 = w_1^n \cdot \mu(V_1)$, and set $\tilde{w}_1^n$ to

$$\tilde{w}_1^n = \begin{cases} 0, & \text{if } I_1 \le \frac{1}{2}\mu(V_1), \\ 1, & \text{otherwise.} \end{cases}$$

(2) For $i = 2, 3, ..., n$, compute

$$I_i = \sum_{k=1}^{i} w^n(x_k)\mu(V_k) \quad \text{and} \quad \tilde{I}_{i-1} = \sum_{k=1}^{i-1} \tilde{w}^n(x_k)\mu(V_k),$$

where $\tilde{w}_i^n$ is given by

$$\tilde{w}_i^n = \begin{cases} 0, & \text{if } I_i - \tilde{I}_{i-1} \le \frac{1}{2}\mu(V_i), \\ 1, & \text{otherwise.} \end{cases}$$

We call this strategy basic *sum-up rounding*, in reference to the name of the one-dimensional technique introduced in [22, 21] which inspired this approach. The basic idea is that each element is scanned sequentially and is rounded to either 0 or 1 determined by the difference in the accumulated sum of elements that are already processed. The strategy has the property that for large n, $w^n(x)$ and $\tilde{w}^n(x)$ get close to each other for all partial sums, which is stated in the following lemma.

Lemma 3.3.1. *The function $\tilde{w}^n(x)$ has the following property: For any $i = 1, 2, ..., n$,*

$$|I_i - \tilde{I}_i| = \Big| \sum_{k=1}^{i} \Big(w^n(x_k) - \tilde{w}^n(x_k) \Big) \Delta_x \Big| \leq \frac{1}{2n}\mu(V).$$

where V is the rectangular output domain with fixed size.

Proof. We prove this result by induction. For $i = 1$, we have the following.

- When $I_1 \leq \frac{1}{2}\mu(V_1) = \frac{1}{2n}\mu(V)$, we have $\tilde{w}^n_1 = 0$ and $\tilde{I}_1 = 0$, and therefore

$$|I_1 - \tilde{I}_1| = I_1 \leq \frac{1}{2n}\mu(V).$$

- When $I_1 > \frac{1}{2n}\mu(V)$, we have $\tilde{w}^n_1 = 1$. Since $w^n(x) \leq 1$, we get

$$\frac{1}{2n}\mu(V) < I_1 \leq \frac{1}{n}\mu(V), \quad |I_1 - \tilde{I}_1| = \frac{1}{n}\mu(V) - I_1 \leq \frac{1}{2n}\mu(V).$$

By the induction hypothesis, assume $|I_i - \tilde{I}_i| \leq \frac{1}{2n}\mu(V)$ is true for $i = k$. We show it for $i = k+1$ as follows.

- When $0 \leq I_k - \tilde{I}_k \leq \frac{1}{2n}\mu(V)$, note that $I_k \leq I_{k+1}$. We discuss two cases.

  (a) If $0 \leq I_{k+1} - \tilde{I}_k \leq \frac{1}{2n}\mu(V)$, then $\tilde{w}^n_{k+1} = 0$ from the rounding rule, and thus $\tilde{I}_{k+1} = \tilde{I}_k$. Therefore

$$0 \leq I_{k+1} - \tilde{I}_{k+1} \leq \frac{1}{2n}\mu(V),$$

  and the induction hypothesis is satisfied.

  (b) If $\frac{1}{2n}\mu(V) < I_{k+1} - \tilde{I}_k$, which implies

$$\frac{1}{2n}\mu(V) < I_{k+1} - \tilde{I}_k \leq I_k - \tilde{I}_k + \frac{w^n_{k+1}}{n}\mu(V) \leq \frac{1}{2n}\mu(V) + \frac{1}{n}\mu(V), \qquad (3.7)$$

then from the rounding rule we have that $\tilde{w}^n_{k+1}\!=\!1$, and we obtain

$$\tilde{I}_{k+1} - \tilde{I}_k = \mu(V_{i+1}) = \frac{1}{n}\mu(V). \tag{3.8}$$

Subtracting the equality (3.8) from the inequality (3.7) gives the desired result:

$$-\frac{1}{2n}\mu(V) < I_{k+1} - \tilde{I}_{k+1} = I_{k+1} - \tilde{I}_k - \frac{1}{n}\mu(V) \le \frac{1}{2n}\mu(V).$$

- When $-\frac{1}{2n}\mu(V) \le I_k - \tilde{I}_k \le 0$, since $I_{k+1} = I_k + w^n_{k+1}\frac{\mu(V)}{n}$, we also have that $I_{k+1} \ge I_k$, and thus $-\frac{1}{2n}\mu(V) \le I_{k+1} - \tilde{I}_k$. We discuss two cases in a similar way.

  (a) If $-\frac{1}{2n}\mu(V) \le I_{k+1} - \tilde{I}_k \le \frac{1}{2n}\mu(V)$, then $\tilde{w}^n_{k+1}\!=\!0$ from the rounding rule, and thus $\tilde{I}_{k+1} = \tilde{I}_k$. Hence

$$-\frac{1}{2n}\mu(V) \le I_{k+1} - \tilde{I}_{k+1} \le \frac{1}{2n}\mu(V).$$

  (b) If $\frac{1}{2n}\mu(V) < I_{k+1} - \tilde{I}_k$, then

$$\frac{1}{2n}\mu(V) < I_{k+1} - \tilde{I}_k \le I_k - \tilde{I}_k + \frac{w^n_{k+1}}{n}\mu(V) \le 0 + \frac{1}{n}\mu(V). \tag{3.9}$$

  In turn, from the rounding rule this implies that $\tilde{w}^n_{k+1}\!=\!1$. As a result, we have

$$\tilde{I}_{k+1} = \tilde{I}_k + \mu(V_{i+1}) = \tilde{I}_k + \frac{1}{n}\mu(V). \tag{3.10}$$

  Replacing the identity (3.10) in the inequality (3.9), we obtain

$$-\frac{1}{2n}\mu(V) < I_{k+1} - \tilde{I}_{k+1} = I_{k+1} - \tilde{I}_k - \frac{1}{n}\mu(V) \le 0.$$

Inspecting the consequences of these four branches, we have completed the proof for $i\!=\!k\!+\!1$, namely, $|I_{k+1} - \tilde{I}_{k+1}| \le \frac{1}{2n}\mu(V)$. Therefore the statement is true for $i = 1, 2, ..., n$ and the proof is complete. □

We now have a rounding strategy, and before we apply it, it is important to check feasibility of the resulting integer vector. The lemma below states that sum-up rounding always provides a feasible vector for our main optimization problem (3.4).

Lemma 3.3.2. *With the basic sum-up rounding strategy, if* $\sum_{k=1}^{n} w^n(x_k) = n_0$ *is an integer, then*

$$\sum_{k=1}^{n} \tilde{w}^n(x_k) = \sum_{k=1}^{n} w^n(x_k) = n_0.$$

Proof. In Lemma 3.3.1 we have that $\Delta_x = \frac{\mu(V)}{n}$, and the conclusion for $i = n$ can be rewritten as

$$\Big|\sum_{k=1}^{n} w^n(x_k) - \sum_{k=1}^{n} \tilde{w}^n(x_k)\Big| \leq \frac{1}{2}.$$

Since both $\sum_{k=1}^{n} w^n(x_k)$ and $\sum_{k=1}^{n} \tilde{w}^n(x_k)$ are integers, they have to be equal. □

3.3.2 A Two-level Decomposition Scheme

We showed in §3.3.1 that $w^n(x)$ and $\tilde{w}^n(x)$ are close to each other, but our goal is to prove that the two sums in (3.6) are close. Suppose $V = [l_1^1, l_2^1] \times [l_1^2, l_2^2] \times ... \times [l_1^P, l_2^P] \subset \mathbb{R}^P$, and each $[l_1^i, l_2^i]$ is divided into n_i intervals $\mathcal{I}_{i,1}, \mathcal{I}_{i,2}, ..., \mathcal{I}_{i,n_i}$ (script letters represent one-dimensional intervals) of equal length. Then there are $n = n_1 n_2 \cdots n_P$ unit rectangles of the form

$$\prod_{\substack{i=1,2,..,P,\\ j_i \in \{1,2,..,n_i\}}} \mathcal{I}_{i,j_i}.$$

They all have the same size $\mu(V)/n$, and we call them $R_1, R_2, ..., R_n$. In addition, we assume that there exist two positive constants c_1, c_2 such that

$$c_1 \leq \frac{\max_{i=1,2,..,P} n_i}{\min_{i=1,2,..,P} n_i} \leq c_2. \tag{3.11}$$

This implies that $n_i = \mathcal{O}(n^{1/P})$ for any $i \in \{1, 2, ..., P\}$ and that each rectangle R_i is not far from a "unit box."

Definition 1. *We call a compatible two-level decomposition scheme a domain decomposition setup of a compact domain V with the following properties. The rectangles R_i, $i = 1, 2, \dots, n$, are grouped in subdomains V_j, $j = 1, 2, \dots, \tilde{k}(n)$, for which the first $k(n)$ subdomains contain an equal number of rectangles, $r(n)$. The intersections between the interiors of each two subdomains V_j is empty, moreover the subdomains V_j need not cover the entire domain V, and we denote the remainder by $V_{rem} = V - \cup_{j=1}^{\tilde{k}(n)} V_j$. We denote by $\rho(V_j)$ the diameter of the subdomains, $j = 1, 2, \dots, k(n)$. Subsequently, we reindex the rectangles such that their ordering respects the subdomains ordering, that is, $R_{i_1} \in V_{j_1}, R_{i_2} \in V_{j_2}, j_1 \leq j_2 \Rightarrow i_1 \leq i_2$. Our sum-up rounding approach consists of applying the basic method from §3.3.1 to the rectangles R_i in their modified ordering.*

To obtain the approximation properties, it would be sufficient to apply the basic method from §3.3.1 to each subdomain V_i. The extra steps of reordering and the application to the entire rectangle set ensure that we preserve the total sum of the weights, and thus that we satisfy the constraints from (3.4).

To achieve a vanishing integrality gap, we will be interested in compatible two-level decompositions that satisfy in the limit the following properties:

$$\lim_{n \to \infty} \max_{1 \leq j \leq k(n)} \rho(V_j) = 0, \quad k(n), r(n) \overset{n \to \infty}{\longrightarrow} \infty, \quad \frac{r(n)k(n)}{n} \overset{n \to \infty}{\longrightarrow} 1, \quad \mu(V_{rem}) \to 0. \qquad (3.12)$$

For many domains V such compatible two-level decompositions can be easily obtained based on algorithms for hexahedral meshing [67] that are commonly used in spectral element methods [68]. Note that our problem is easier than most in that sense, since the mesh need not be conformal [69], that is, we allow $V_{rem} \neq \emptyset$. Even in that case, however, a rigorous proof of (3.12) for a wide class of domains is non-trivial and significantly beyond the scope of the paper. The theoretical existence of such decompositions, however, seems clear as similar techniques are central to Riemann sums convergence arguments.

We thus demonstrate how to create compatible two-level decompositions for rectangular domains only, as follows.

(i) We divide V into $n = n_1 n_2 \cdots n_P$ small rectangles of the form (3.3.2) as before, and we list them as $R_1, R_2, ..., R_n$.

(ii) We order the unit rectangles $R_1, R_2, ..., R_n$, as follows:

$$\begin{aligned}
&R_1 = \mathcal{I}_{1,1} \times \mathcal{I}_{2,1} \times ... \times \mathcal{I}_{P,1}\\
&R_2 = \mathcal{I}_{1,2} \times \mathcal{I}_{2,1} \times ... \times \mathcal{I}_{P,1}\\
&\vdots\\
&R_{n_1} = \mathcal{I}_{1,n_1} \times \mathcal{I}_{2,1} \times ... \times \mathcal{I}_{P,1}\\
&R_{n_1+1} = \mathcal{I}_{1,1} \times \mathcal{I}_{2,2} \times ... \times \mathcal{I}_{P,1}\\
&R_{n_1+2} = \mathcal{I}_{1,2} \times \mathcal{I}_{2,2} \times ... \times \mathcal{I}_{P,1}\\
&\vdots\\
&R_n = \mathcal{I}_{1,n_1} \times \mathcal{I}_{2,n_2} \times ... \times \mathcal{I}_{P,n_P}.
\end{aligned}$$

They are ordered "line by line" according to the first dimension. Denoting $k_1(n_1) \triangleq \lfloor \sqrt{n_1} \rfloor$, we now build the subdomains V_j as follows.

(a) On $[l_1^1, l_2^1]$ we group the first $k_1(n_1)$ intervals $\{\mathcal{I}_{i,j}\}_{j=1}^{k_1(n_1)}$ as $\mathcal{G}_{1,1}$, group the next $k_1(n_1)$ intervals $\{\mathcal{I}_{1,j}\}_{k_1(n_1)+1}^{2k_1(n_1)}$ as $\mathcal{G}_{1,2}$, and so forth until we get $\mathcal{G}_{1,k_1(n_1)}$. The remaining intervals $\{\mathcal{I}_{1,j}\}_{j=k_1(n_1)^2+1}^{n}$ are grouped as $\mathcal{G}_{1,last}$, and the number of intervals in $\mathcal{G}_{1,last}$ equals $n_1 - \lfloor \sqrt{n_1} \rfloor^2$.

(b) The subdomain V_j has the following form:

$$\mathcal{G}_{1,j_1} \times \mathcal{I}_{2,j_2} \times .. \times \mathcal{I}_{P,j_P},$$

where $j_1 \in \{1, 2, .., k_1(n_1), last\}, j_i \in \{1, 2, .., n_i\}$ for $i \geq 2$.

This decomposition has the following parameters and properties, in reference to Definition 1.

$$k(n) = \lfloor\sqrt{n_1}\rfloor \prod_{i=2}^{P} n_i,\ \tilde{k}(n) = \lceil\sqrt{n_1}\rceil \prod_{i=2}^{P} n_i,\ r(n) = \lfloor\sqrt{n_1}\rfloor \tag{3.13}$$

$$\rho(V_j) = \sqrt{\left(\frac{(l_2^1 - l_1^1)}{\lfloor\sqrt{n_1}\rfloor}\right)^2 + \sum_{i=2}^{P}\left(\frac{(l_2^i - l_1^i)}{n_i}\right)^2}, \quad j = 1, 2, \ldots, k(n) \tag{3.14}$$

With these definitions, sum-up rounding is applied as described in Definition 1. We note that many other compatible two-level decompositions are possible, another one is presented in §B.

The following simple example illustrates the idea of two-level decomposition on a square domain in $\mathbb{R}^2$. There are 10 points evenly spaced on each side, and then unit rectangles R_j; we group 3 of them and form 30 subdomains V_j; the basic sum-up rounding strategy is applied to each V_j. As the construction is repeated for increasing n, the remainder area (yellow in color rendering) will diminish compared to the full domain, and its effect on the difference between the sums in (3.6) and their relationship to the corresponding integral will vanish. The detailed explanation related to the output domain Figure 3.1 is given here:

- domain: $[l_1^1, l_2^1] \times [l_1^2, l_2^2]$;
- discretization parameter: $n_1 = n_2 = 10$, $n = n_1 * n_2 = 100$;
- $k_1(n_1) = \lfloor\sqrt{10}\rfloor = 3$, $k(n) = 30, \tilde{k}(n) = 40$;
- $R_j = I_{1,j_1} \times I_{2,j_2}$ where $j = (j_2 - 1) * n_1 + j_1$;
- $\mathcal{G}_{1,1} = I_{1,1} \cup I_{1,2} \cup I_{1,3}$, $\mathcal{G}_{1,2} = I_{1,4} \cup I_{1,5} \cup I_{1,6}$, $\mathcal{G}_{1,3} = I_{1,7} \cup I_{1,8} \cup I_{1,9}$, $\mathcal{G}_{1,last} = I_{1,last}$;
- subdomain: $V_1 = \mathcal{G}_{1,1} \times I_{2,1}$, $V_2 = \mathcal{G}_{1,2} \times I_{2,1}$, $V_3 = \mathcal{G}_{1,3} \times I_{2,1}$, $\cdots$.

We will characterize essential features of this approach in the next subsection.

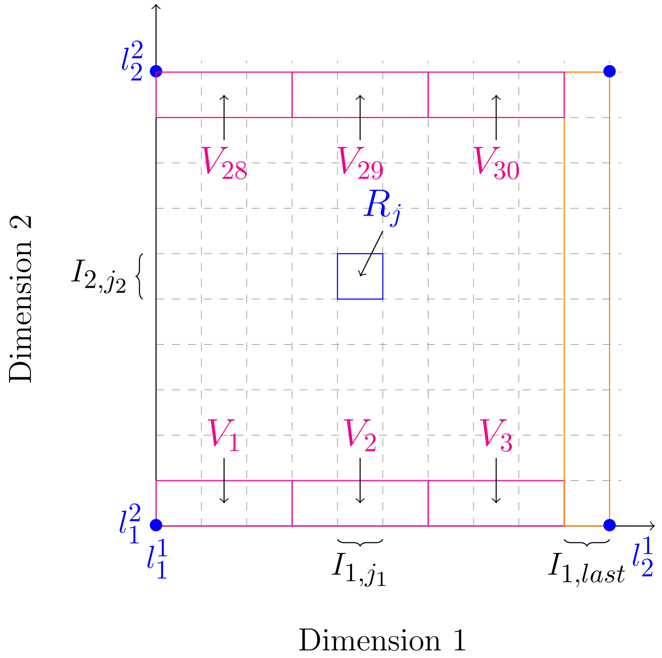


Fig.3.1: An illustration of two-level decomposition of rectangle domain

3.3.3 Properties of Sum-Up Rounding in Multiple Dimensions

For our results, we use the notation $\|x\| = \|x\|_2$ for the norm of a vector $x \in \mathbb{R}^n$.

Theorem 3.3.3. *Assume that V is a compact domain in $\mathbb{R}^P$ and that $f(x)$ is Lipschitz continuous on V with Lipschitz constant L: for any $x, y \in V$,*

$$|f(x) - f(y)| \leq L\|x - y\|.$$

Consider a compatible two-level domain decomposition and let $\tilde{w}^n(x)$ be the binary function from a sum-up rounding algorithm as described in Definition 1. Let x_k be a point in R_k, $k = 1, 2, \ldots, n$. Then we have

$$\Big|\sum_{k=1}^{n} f(x_k)\Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \leq \max_{x\in V}|f(x)|\frac{\mu(V - V_{rem})}{r(n)}\frac{k(n)r(n)}{n} +$$
$$\max_{j=1,2,\ldots,k(n)} \rho(V_j)2L\mu(V - V_{rem})\frac{k(n)r(n)}{n} + 2\max_{x\in V} f(x)\mu(V - V_{rem})\left(1 - \frac{k(n)r(n)}{n}\right).$$

Moreover, if $\sum_{k=1}^{n} w^n(x_k) = n_0$ *is an integer, then* $\sum_{k=1}^{n} \tilde{w}^n(x_k) = n_0$.

Proof. We prove first the result for the case where $k(n) = \tilde{k}(n)$ and $V_{rem} = \emptyset$ (that is, all subdomains V_j have the same size and properties and they exactly cover the domain V). In this case Lemma 3.3.1 gives

$$\Big|\sum_{x_k \in V_1 \cup .. \cup V_j} \Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \leq \frac{1}{2j \cdot r(n)}\mu(V_1 \cup .. \cup V_j) = \frac{1}{2r(n)}\mu(V_j).$$

This implies

$$\begin{aligned}\Big|\sum_{x_k \in V_j} \Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| &\leq \Big|\sum_{x_k \in V_1 \cup .. \cup V_j} \Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \\ &\quad + \Big|\sum_{x_k \in V_1 \cup .. \cup V_{j-1}} \Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \\ &\leq \frac{1}{2r(n)}\mu(V_j) + \frac{1}{2r(n)}\mu(V_j) \\ &= \frac{1}{r(n)}\mu(V_j). \end{aligned} \tag{3.15}$$

Let y_j be any point in subdomain V_j, and define

$$\begin{aligned}\Upsilon &= \sum_{k=1}^{n} f(x_k)\big(w^n(x_k) - \tilde{w}^n(x_k)\big)\Delta_x, \\ \Psi &= \sum_{j=1}^{k(n)} f(y_j) \sum_{x_k \in V_j} \big(w^n(x_k) - \tilde{w}^n(x_k)\big)\Delta_x. \end{aligned}$$

A bound on $|\Psi|$ is given as

$$|\Psi| \leq \sum_{j=1}^{k(n)} |f(y_j)| \Big|\sum_{x_k \in V_j} \Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \overset{(3.15)}{\leq} \max_{x \in V} |f(x)| \sum_{j=1}^{k(n)} \frac{\mu(V_j)}{r(n)} = \frac{\mu(V)}{r(n)} \max_{x \in V} |f(x)|. \tag{3.16}$$

Lipschitz continuity implies $|f(x) - f(y)| \leq L\|x - y\|$ for any $x, y \in V_j$ and

$$\begin{aligned}
|\Upsilon - \Psi| &= \Big|\Delta_x \sum_{j=1}^{k(n)} \sum_{x_k \in V_j} \big(f(x_k) - f(y_j)\big)\big(w^n(x_k) - \tilde{w}^n(x_k)\big)\Big| \\
&\leq \Delta_x \sum_{j=1}^{k(n)} \sum_{x_k \in V_j} \Big|\big(f(x_k) - f(y_j)\big)\big(w^n(x_k) - \tilde{w}^n(x_k)\big)\Big| \\
&\leq \Delta_x \sum_{j=1}^{k(n)} \sum_{x_k \in V_j} 2L\|x_k - y_j\| \\
&\leq \sum_{j=1}^{k(n)} 2L\rho(V_j) \sum_{x_k \in V_j} \Delta_x \\
&= 2L \sum_{j=1}^{k(n)} \rho(V_j)\mu(V_j) \\
&\leq 2L\mu(V) \max_j \rho(V_j). \qquad (3.17)
\end{aligned}$$

Therefore we obtain from (3.16) and (3.17) that

$$\begin{aligned}
\Big|\sum_{k=1}^{n} f(x_k)\Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| &= |\Upsilon| \\
&\leq |\Psi| + |\Upsilon - \Psi| \\
&\leq \max_{x \in V} |f(x)| \frac{\mu(V)}{r(n)} + 2L\mu(V) \max_j \rho(V_j).
\end{aligned}$$

When $\tilde{k}(n) > k(n)$ and $V_{rem} \neq \emptyset$, we divide $V - V_{rem}$ into two disjoint domains $V_{main} = \bigcup_{j=1}^{k(n)} V_j$ and $V_{last} = \bigcup_{j=k(n)+1}^{\tilde{k}(n)} V_j$. We apply the results in the case $k(n) = \tilde{k}(n)$ to V_{main} to obtain

$$\begin{aligned}
\Big|\sum_{x_k \in V_{main}} f(x_k)\Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \leq& \max_{x \in V} |f(x)| \frac{\mu(V_{main})}{r(n)} \\
&+ 2L\mu(V_{main}) \max_j \rho(V_j), \qquad (3.18)
\end{aligned}$$

For the remaining part of the sum, using the fact that the components of w and $\tilde{w}$ are bounded between 0 and 1, we have

$$\Big| \sum_{x_k \in V_{last}} f(x_k)\Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \leq 2 \max_{x \in V} |f(x)| \mu(V_{last}). \tag{3.19}$$

Because each unit rectangle R_k has the same size, we have

$$\begin{aligned} &\mu(V_{main}) = \frac{k(n)r(n)}{n}\mu(V - V_{rem}), \\ &\mu(V_{last}) = \mu(V) - \mu(V_{main}) = \mu(V - V_{rem})\left(1 - \frac{k(n)r(n)}{n}\right). \end{aligned}$$

Applying these identities to the inequalities (3.18) and (3.19), we obtain the inequality claimed in the proof. The equality is a consequence of applying the basic sum-up rounding rule from §3.3.1 to the set of all rectangles as described in Definition 1, in conjunction with Lemma 3.3.2. The proof is complete. □

The preceding result gives us the following immediate corollary.

Corollary 3.3.1. *With the assumptions of Theorem 3.3.3, further assume that a sequence of compatible two-level domain decompositions satisfy* (3.12)*. We then obtain that*

$$\lim_{n\to\infty} \Big| \sum_{k=1}^{n} f(x_k)\Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| = 0.$$

and that, if $\sum_{k=1}^{n} w^n(x_k) = n_0$ is an integer, then $\sum_{k=1}^{n} \tilde{w}^n(x_k) = n_0$. In other words the gap between the relaxation and our sum-up rounded integer solution goes to zero, and the problem is feasible.

As discussed following the definition of compatible two-level domain decompositions, Definition 1, this result can be used to show the vanishing integrality gap of our approach for many types of domains. A complete analysis of when (3.12) holds appear extensive, though cases such as unions of rectangles or polyhedral sets do not seem to require particularly deep analysis. Given our focus on consequences for optimization, we focus exclusively on the rectangular domain case. For that situation, we can strengthen (3.12) and Corollary 3.3.1 by giving a bound on the rate of convergence as $n \to \infty$ (also note that $V_{rem} = \emptyset$ in that case).

Theorem 3.3.4. *Under the assumptions of Theorem 3.3.3, there exists a C such that our sum-up rounding construction satisfies*

$$\Big|\sum_{k=1}^{n} f(x_k)\Big(w^n(x_k) - \tilde{w}^n(x_k)\Big)\Delta_x\Big| \leq \frac{C}{n^{1/2P}}.$$

Proof. We use the inequalities

$$\frac{(\lfloor\sqrt{n}\rfloor)^2}{n} \geq 1 - \frac{2}{\sqrt{n}}, \forall n \in \mathbb{N}, \qquad \frac{(\lfloor\sqrt{n}\rfloor)^2}{n} \geq \frac{1}{2}, \forall n > 3, \tag{3.20}$$

and

$$c_1 n^{\frac{1}{P}} \leq \min_{i=1,2,\dots,P} n_i \leq n^{\frac{1}{P}}, \quad n^{\frac{1}{P}} \leq \max_{i=1,2,\dots,P} n_i \leq c_2 n^{\frac{1}{P}} \tag{3.21}$$

that follow from (3.11).

We use the definitions of the sum-up rounding scheme parameters (3.13)-(3.14) to infer the following inequalities.

$$\frac{1}{\sqrt{n_1}} \leq c_1^{-\frac{1}{2}} n^{-\frac{1}{2P}}; \quad \frac{r(n)k(n)}{n} \leq 1; \quad \frac{1}{r(n)} = \frac{1}{\lfloor\sqrt{n_1}\rfloor} \leq \frac{\sqrt{2}}{\sqrt{n_1}} \tag{3.22}$$

For the maximum diameter of V_j we obtain from (3.14) and (3.20)

$$\max_{j=1,2,\dots,k(n)} \rho(V_j) \leq \sqrt{P}\frac{\max_{i=1,2,..,P}(l_2^i - l_1^i)}{\frac{1}{2}\min_{i=1,2,\dots,P}\sqrt{n_i}} \overset{(3.21)}{\leq} \sqrt{P}\frac{\max_{i=1,2,..,P}(l_2^i - l_1^i)}{\frac{1}{2}\sqrt{c_1}} n^{-\frac{1}{2P}}. \tag{3.23}$$

We also obtain

$$1 - \frac{k(n)r(n)}{n} = 1 - \frac{\lfloor\sqrt{n_1}\rfloor^2}{n_1} \overset{(3.20)}{\leq} 1 - \Big(1 - \frac{2}{\sqrt{n_1}}\Big) \overset{(3.22)}{\leq} 2c_1^{-\frac{1}{2}} n^{-\frac{1}{2P}}. \tag{3.24}$$

We now use Theorem 3.3.3 along with (3.22), (3.23), and (3.24) to obtain the statement of this theorem with the choice

$$C = \max_{x\in V} |f(x)|\mu(V)\sqrt{2}c_1^{-\frac{1}{2}} + 4L\mu(V)\sqrt{P} \max_{i=1,2,..,P}(l_2^i - l_1^i)c_1^{-\frac{1}{2}} + 4\max_{x\in V}|f(x)|\mu(V)c_1^{-\frac{1}{2}}.$$

This completes the proof. □

We note that other compatible two-level relaxations observe similar bounds when used for sum-up rounding; see §B.

3.4 Approximation of Functions of the Covariance Matrix

We rely on the convergence of the sum-up rounding strategy to prove the main results on functions of covariance matrices. We keep the ratio Δ_y/Δ_x (or n/m) constant, say α, in (3.2). We define

$$G_m^n = \Delta_y \cdot \{g^{w^n}(y_i, y_j)\}_{i,j=1}^m \quad \text{and} \quad \tilde{G}_m^n = \Delta_y \cdot \{g^{\tilde{w}^n}(y_i, y_j)\}_{i,j=1}^m,$$

where

$$\begin{aligned} g^{w^n}(y_i, y_j) &= \alpha\sigma_{noise}^{-1}\sum_{k=1}^n f(x_k, y_i)w^n(x_k)f(x_k, y_j)\Delta_x \\ g^{\tilde{w}^n}(y_i, y_j) &= \alpha\sigma_{noise}^{-1}\sum_{k=1}^n f(x_k, y_i)\tilde{w}^n(x_k)f(x_k, y_j)\Delta_x. \end{aligned}$$

Here w^n is the solution to the relaxed optimization problem (3.5) with the discretization parameter n, and we construct $\tilde{w}^n$ from the SUR technique in §3.3. The quantities G_m^n, $\tilde{G}_m^n$, and Γ_{post} satisfy the following relationships

$$\Gamma_{post}(w^n) = \left(G_m^n + \sigma_{pri}^{-1}I_m\right)^{-1}, \quad \Gamma_{post}(\tilde{w}^n) = \left(\tilde{G}_m^n + \sigma_{pri}^{-1}I_m\right)^{-1}. \tag{3.25}$$

The assumption of Lipschitz continuity we make on $f(x, y)$ is

$$\Big| f(x_1, y_1) f(x_1, y_2) - f(x_2, y_1) f(x_2, y_2) \Big| \leq L \|x_1 - x_2\|,$$

where $y_1, y_2 \in \Omega_{in}$ and L is independent of y_1 and y_2. This is not a stringent assumption, since we can let L depend on y_1, y_2 first and then take $L := \max_{y_1, y_2 \in \Omega_{in}} L(y_1, y_2)$ (note that Ω_{in} is bounded and closed, thus ensuring $L < \infty$). Theorem 3.3.4 then implies that

$$\forall i, j = 1, 2, ..., m, \quad |g^{w^n}(y_i, y_j) - g^{\tilde{w}^n}(y_i, y_j)| \leq \tilde{\epsilon}_n \to 0, \quad \text{as } n \to \infty. \tag{3.26}$$

Here $\tilde{\epsilon}_n$ is the bound from Theorem 3.3.4.

By definition of the Frobenius norm,

$$\|G^n_m - \tilde{G}^n_m\|_F \leq \Delta_y \sqrt{m^2 \epsilon_n^2} = \mu(\Omega_{in}) \tilde{\epsilon}_n \to 0. \tag{3.27}$$

Since $\mu(\Omega_{in})$ is constant, we can introduce a new sequence $\{\epsilon_n\} \to 0$, $\epsilon_n = \max\{1, \mu(\Omega)\tilde{\epsilon}_n\}$. With this notation we have

$$|g^{w^n}(y_i, y_j) - g^{\tilde{w}^n}(y_i, y_j)| \leq \epsilon_n \quad \text{and} \quad \|G^n_m - \tilde{G}^n_m\|_F \leq \epsilon_n. \tag{3.28}$$

Denote eigenvalues of G^n_m and $\tilde{G}^n_m$ as

$$\lambda^n_1 \geq \lambda^n_2 \geq ... \geq \lambda^n_m \geq 0$$
$$\tilde{\lambda}^n_1 \geq \tilde{\lambda}^n_2 \geq ... \geq \tilde{\lambda}^n_m \geq 0.$$

Note the number of eigenvalues for both G^n_m and $\tilde{G}^n_m$ is m, which changes and rises up to infinity. We will show the kth eigenvalues of G^n_m and $\tilde{G}^n_m$ are arbitrarily close for any fixed $k \in \mathbb{Z}_+$.

Lemma 3.4.1. *If λ_k^n and $\tilde{\lambda}_k^n$ are the k-th eigenvalues of G_m^n and $\tilde{G}_m^n$, respectively, then*

$$|\lambda_k^n - \tilde{\lambda}_k^n| \leq 2 \cdot \epsilon_n. \tag{3.29}$$

Proof. From the Courant-Fischer theorem for real-valued symmetric matrices [70, Theorem 4.2.11], the kth largest eigenvalue of G_m^n can be computed as

$$\lambda_k^n = \sup_{dim(S)=k} \inf \Big\{ \frac{\|G_m^n \cdot u\|}{\|u\|} : u \in S,\ u \neq 0 \Big\}. \tag{3.30}$$

From this, we know there exists a subspace S of dimension k in $\mathbb{R}^m$ such that

$$\frac{\|G_m^n \cdot u\|}{\|u\|} \geq \lambda_k^n - \epsilon_n$$

for any $u \in S$, $u \neq 0$. We apply (3.28); and using the relationship $\|A\| \leq \|A\|_F$, we obtain

$$\begin{aligned} \inf_{u \in S, u \neq 0} \frac{\|\tilde{G}_m^n \cdot u\|}{\|u\|} &\geq \frac{\|G_m^n \cdot u\| - \|G_m^n - \tilde{G}_m^n\|_F \|u\|}{\|u\|} \\ &\geq \frac{\|G_m^n \cdot u\| - \epsilon_n \|u\|}{\|u\|} \\ &\geq \lambda_k^n - 2\epsilon_n. \end{aligned}$$

Again from (3.30), we get

$$\tilde{\lambda}_k^n \geq \lambda_k^n - 2\epsilon_n.$$

Switching $\tilde{G}_m^n$ and G_m^n and using similar arguments, we obtain the reverse inequality

$$\lambda_k^n \geq \tilde{\lambda}_k^n - 2\epsilon_n.$$

Then (3.29) follows directly. □

Lemma 3.4.1 can directly be used to show convergence of the gap for E-optimality, since in that case, the difference between the objectives is

$$\left|\frac{1}{\sigma+\lambda_n^n}-\frac{1}{\sigma+\tilde{\lambda}_n^n}\right| \leq \frac{|\lambda_n^n-\tilde{\lambda}_n^n|}{\sigma^2} \leq \frac{\epsilon_n}{\sigma^2}.$$

On the other hand, for integral operators with continuous kernels it can be shown that λ_n approaches zero, therefore any design will produce the same result in the limit which makes this criterion uninteresting in our setup. For the A- and D- optimality case, however, the objective function can be seen as the sum of eigenvalues or logarithm of eigenvalues of the covariance matrix, and the number of its terms goes to infinity. In that case, the objective functions may not even be bounded as $n \to \infty$, as we discuss in (3.54) and (3.55). Therefore directly invoking Lemma 3.4.1 would not prove convergence. As a simple example, consider the situation where $\lambda_k^n = 1 + \frac{k}{n\sqrt{n}}$, and $\tilde{\lambda}_k^n = 1$, $k = 1, 2, \ldots, n,$. For any k we have that $|\lambda_k^n - \tilde{\lambda}_k^n| \leq n^{-\frac{1}{2}}$ and thus the two eigenvalue sequences satisfy a relationship as the one in the conclusion of Lemma 3.4.1. On the other hand the difference between the A-optimal criteria would be

$$\sum_{k=1}^{n}\left(\frac{1}{\sigma+1+\frac{k}{n\sqrt{n}}}-\frac{1}{\sigma+1}\right) \leq -\frac{\sum_{k=1}^{n} k}{n\sqrt{n}(\sigma+1)(\sigma+2)} \to -\infty.$$

A proof of a zero gap between function of a matrix and its SUR version will require more results beyond Lemma 3.4.1. In the following two theorems, we provide rigorous proofs on convergence for A- and D-optimal design criteria respectively.

Theorem 3.4.2. *Let $M_m^n = \left(\sigma I_m + G_m^n\right)^{-1}$ and $\tilde{M}_m^n = \left(\sigma I_m + \tilde{G}_m^n\right)^{-1}$, where $\sigma = \sigma_{pri}^{-1}$. Then*

$$tr(M_m^n) - tr(\tilde{M}_m^n) \to 0$$

as $m, n \to \infty$ and with $n/m = \alpha$ constant.

The proof is based on the fact that from Lemma 3.4.1, the spectra of G_m^n and of $\tilde{G}_m^n$ are close to each other. From the definition of M_m^n, its spectra can be inferred from that of

G_m^n through $\lambda_M = 1/(\sigma + \lambda_G)$, where λ_G is an eigenvalue of G_m^n and λ_M is an eigenvalue of M_m^n. The key is to exploit this relationship to show that the spectra of M_m^n and $\tilde{M}_m^n$ are also close, combined with the consequences of the compactness of the integral operator.

Proof. Since w^n and $\tilde{w}^n$ are between 0 and 1, then $g^{w^n}(y,y)$ and $g^{\tilde{w}^n}(y,y)$ are absolutely integrable.

$$\begin{aligned}
0 < \sum_{k=1}^{m} \lambda_k^n = tr(G_m^n) &= \Delta_y \cdot \sum_{i=1}^{m} g^{w^n}(y_i, y_i) \\
&\leq \Delta_y \cdot \sum_{i=1}^{m} |g^{w^n=1}(y_i, y_i)| \to \int_{\Omega_{in}} |g^{w=1}(y,y)| \, \mathrm{d}y \\
0 < \sum_{k=1}^{m} \tilde{\lambda}_k^n = tr(\tilde{G}_m^n) &= \Delta_y \cdot \sum_{i=1}^{m} g^{\tilde{w}^n}(y_i, y_i) \\
&\leq \Delta_y \cdot \sum_{i=1}^{m} |g^{\tilde{w}^n=1}(y_i, y_i)| \to \int_{\Omega_{in}} |g^{w=1}(y,y)| \, \mathrm{d}y
\end{aligned}$$

The inequality holds because $g^{w^n}(y_i, y_j)$ depends linearly on w^n. Since convergent sequences are uniformly bounded, there exists a constant $C>0$ such that for any $n > 0$,

$$0 < \sum_{k=1}^{m} \lambda_k^n \leq C, \qquad 0 < \sum_{k=1}^{m} \tilde{\lambda}_k^n \leq C. \tag{3.31}$$

We also have that

$$\begin{aligned}
\Big| \sum_{k=1}^{m} \lambda_k - \sum_{k=1}^{m} \tilde{\lambda}_k^n \Big| &= \Big| \Delta_y \sum_{i=1}^{m} \Big(g^{w^n}(y_i, y_i) - g^{\tilde{w}^n}(y_i, y_i) \Big) \Big| \\
&\leq \Delta_y \sum_{i=1}^{m} \Big| g^{w^n}(y_i, y_i) - g^{\tilde{w}^n}(y_i, y_i) \Big| \\
&\leq \Delta_y \cdot m \cdot \epsilon_n = \mu(\Omega_{in}) \epsilon_n,
\end{aligned}$$

where the last inequality follows from (3.26). Since $\mu(\Omega_{in})$ does not depend on n, and similar to the way we defined $\{\epsilon_n\}$ in (3.28), we can redefine the sequence $\{\epsilon_n\} \to 0$ (for example as $\epsilon_n \leftarrow \max\{1, \mu(\Omega_{in})\}\epsilon_n$) such that the following inequalities hold simultaneously

$$|g^{w^n}(y_i, y_j) - g^{\tilde{w}^n}(y_i, y_j)| \leq \epsilon_n, \ \|G_m^n - \tilde{G}_m^n\|_F \leq \epsilon_n, \ \Big|\sum_{k=1}^m \lambda_k^n - \sum_{k=1}^m \tilde{\lambda}_k^n\Big| \leq \epsilon_n. \tag{3.32}$$

We now show that for any small $\epsilon > 0$, there exists an integer $N > 0$ such that for any $n > N$, we have

$$\Big|S\Big| \leq D \cdot \epsilon, \quad S \triangleq \sum_{k=1}^m \frac{1}{\sigma + \lambda_k^n} - \sum_{k=1}^m \frac{1}{\sigma + \tilde{\lambda}_k^n} \tag{3.33}$$

with some positive constant D. Note that $n/m = \alpha$, so m is determined by n and they increase at the same rate. We fix $\epsilon > 0$. From the upper bound in (3.31), there are at most $N_0 = \lceil C/\epsilon \rceil$ eigenvalues satisfying $\lambda_k > \epsilon$, or equivalently, when $k > N_0$, $\lambda_k^n < \epsilon$ for any n, and, from similar reasoning, $\tilde{\lambda}_k^n < \epsilon$. From (3.29), there exists $N_1 > 0$ such that for any $n > N_1$, $|\lambda_k^n - \tilde{\lambda}_k^n| < \epsilon^2$ for all $k = 1, 2, ..., n$. We choose $n > \max\{N_0, N_1\}$ and split the sum in (3.33) into two parts:

$$S = \sum_{k \leq N_0} \Big(\frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big) + \sum_{k > N_0} \Big(\frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big).$$

For the first part, we note that

$$\Big|\sum_{k \leq N_0} \Big(\frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big)\Big| \leq \sum_{k \leq N_0} \frac{|\lambda_k^n - \tilde{\lambda}_k^n|}{(\sigma + \lambda_k^n)(\sigma + \tilde{\lambda}_k^n)} \leq N_0 \cdot \frac{\epsilon^2}{\sigma^2} \leq \frac{C}{\sigma^2} \cdot \epsilon. \tag{3.34}$$

For the second part, we know λ_k^n, $\tilde{\lambda}_k^n < \epsilon$, and we discuss two cases.

(1) When $\tilde{\lambda}_k^n > \lambda_k^n$ and $k > N_0$,

$$0 < \frac{1}{(\sigma + \epsilon)^2}(\tilde{\lambda}_k^n - \lambda_k^n) \leq \frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n} = \frac{\tilde{\lambda}_k^n - \lambda_k^n}{(\sigma + \lambda_k^n)(\sigma + \tilde{\lambda}_k^n)} \leq \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma^2}. \tag{3.35}$$

(2) When $\tilde{\lambda}_k^n < \lambda_k^n$ and $k > N_0$,

$$\frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma^2} \leq \frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n} = \frac{\tilde{\lambda}_k^n - \lambda_k^n}{(\sigma + \lambda_k^n)(\sigma + \tilde{\lambda}_k^n)} \leq \frac{\tilde{\lambda}_k^n - \lambda_k^n}{(\sigma + \epsilon)^2} < 0. \tag{3.36}$$

So we have

$$\begin{aligned}\sum_{k>N_0} \Big(\frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big) \leq & \sum_{\tilde{\lambda}_k^n > \lambda_k^n,\ k>N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma^2} + \sum_{\tilde{\lambda}_k^n < \lambda_k^n,\ k>N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{(\sigma + \epsilon)^2} \\ = & \sum_{k>N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma^2} + \sum_{\tilde{\lambda}_k^n < \lambda_k^n,\ k>N_0} \Big(\frac{1}{(\sigma+\epsilon)^2} - \frac{1}{\sigma^2}\Big)(\tilde{\lambda}_k^n - \lambda_k^n) \\ = & \frac{1}{\sigma^2} \sum_{k>N_0} (\tilde{\lambda}_k^n - \lambda_k^n) + \frac{\epsilon(2\sigma + \epsilon)}{\sigma^2(\sigma+\epsilon)^2} \sum_{\tilde{\lambda}_k^n < \lambda_k^n,\ k>N_0} (\lambda_k^n - \tilde{\lambda}_k^n).\end{aligned}$$

With a similar use of (3.35) and (3.36) we obtain

$$\begin{aligned}\sum_{k>N_0} \Big(\frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big) \geq & \sum_{\tilde{\lambda}_k^n > \lambda_k^n,\ k>N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{(\sigma + \epsilon)^2} + \sum_{\tilde{\lambda}_k^n < \lambda_k^n,\ k>N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma^2} \\ = & \sum_{\tilde{\lambda}_k^n > \lambda_k^n,\ k>N_0} \Big(\frac{1}{(\sigma+\epsilon)^2} - \frac{1}{\sigma^2}\Big)(\tilde{\lambda}_k^n - \lambda_k^n) + \sum_{k>N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma^2} \\ = & \frac{1}{\sigma^2} \sum_{k>N_0} (\tilde{\lambda}_k^n - \lambda_k^n) + \frac{\epsilon(2\sigma + \epsilon)}{\sigma^2(\sigma+\epsilon)^2} \sum_{\tilde{\lambda}_k^n > \lambda_k^n,\ k>N_0} (\lambda_k^n - \tilde{\lambda}_k^n).\end{aligned}$$

From the last two inequalities, we obtain

$$\begin{aligned}\Big| \sum_{k>N_0} \Big(\frac{1}{\sigma + \lambda_k^n} - \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big) \Big| \leq & \frac{1}{\sigma^2} \Big| \sum_{k>N_0} (\lambda_k^n - \tilde{\lambda}_k^n) \Big| \\ & + \frac{\epsilon(2\sigma + \epsilon)}{\sigma^2(\sigma+\epsilon)^2} \max\Big\{ \sum_{\tilde{\lambda}_k < \lambda_k,\ k>N_0} (\lambda_k^n - \tilde{\lambda}_k^n), \sum_{\tilde{\lambda}_k > \lambda_k,\ k>N_0} (\tilde{\lambda}_k^n - \lambda_k^n) \Big\}.\end{aligned} \tag{3.37}$$

In order to bound $\sum_{k>N_0} (\tilde{\lambda}_k^n - \lambda_k^n)$, recall (3.32). From it, there exists $N_2 > 0$ such that for any $n > N_2$ we have

$$\Big|\sum_{k=1}^{m}(\lambda_k^n - \tilde{\lambda}_k^n)\Big| < \epsilon. \tag{3.38}$$

Choose $n > \max\{N_0, N_1, N_2\}$. Because $n \geq N_1$, we have

$$\Big|\sum_{k\leq N_0}(\lambda_k^n - \tilde{\lambda}_k^n)\Big| \leq N_0 \cdot \epsilon^2 = C\epsilon, \tag{3.39}$$

and thus from (3.38), (3.39) and the triangle inequality we get

$$\Big|\sum_{k>N_0}(\lambda_k^n - \tilde{\lambda}_k^n)\Big| \leq \Big|\sum_{k=1}^{m}(\lambda_k^n - \tilde{\lambda}_k^n)\Big| + \Big|\sum_{k\leq N_0}(\lambda_k^n - \tilde{\lambda}_k^n)\Big| \leq (C+1)\epsilon. \tag{3.40}$$

Note that if we let $\epsilon < \sigma$ and use (3.31), we obtain

$$0 < \frac{\epsilon(2\sigma+\epsilon)}{\sigma^2(\sigma+\epsilon)^2}\sum_{\tilde{\lambda}_k<\lambda_k,\ k>N_0}(\lambda_k^n - \tilde{\lambda}_k^n) \leq \frac{3\epsilon}{\sigma^3}\sum_{k=1}^{n}\lambda_k^n \leq \frac{3C}{\sigma^3}\epsilon \tag{3.41}$$

$$0 < \frac{\epsilon(2\sigma+\epsilon)}{\sigma^2(\sigma+\epsilon)^2}\sum_{\tilde{\lambda}_k>\lambda_k,\ k>N_0}(\tilde{\lambda}_k^n - \lambda_k^n) \leq \frac{3\epsilon}{\sigma^3}\sum_{k=1}^{n}\tilde{\lambda}_k^n \leq \frac{3C}{\sigma^3}\epsilon. \tag{3.42}$$

Combining (3.37), (3.40), (3.41), and (3.42), we get

$$\Big|\sum_{k>N_0}\Big(\frac{1}{\sigma+\lambda_k^n} - \frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \leq \frac{C+1}{\sigma^2}\epsilon + \frac{3C}{\sigma^3}\epsilon = \Big(\frac{C+1}{\sigma^2} + \frac{3C}{\sigma^3}\Big)\epsilon \tag{3.43}$$

According to (3.34) and (3.43), we get

$$\begin{aligned}\Big|\sum_{k=1}^{m}\Big(\frac{1}{\sigma+\lambda_k^n} - \frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \leq & \Big|\sum_{k\leq N_0}\Big(\frac{1}{\sigma+\lambda_k^n} - \frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \\ & + \Big|\sum_{k>N_0}\Big(\frac{1}{\sigma+\lambda_k^n} - \frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \\ \leq & \frac{C}{\sigma^2}\epsilon + \Big(\frac{C+1}{\sigma^2} + \frac{3C}{\sigma^3}\Big)\epsilon = \Big(\frac{2C+1}{\sigma^2} + \frac{3C}{\sigma^3}\Big)\epsilon.\end{aligned}$$

Let $D = \frac{2C+1}{\sigma^2} + \frac{3C}{\sigma^3}$. Then for any $\epsilon > 0$ smaller than σ, there exists $N = \max\{N_0, N_1, N_2\}$ such that for any $n > N$,

$$\Big|\sum_{k=1}^{m}\Big(\frac{1}{\sigma+\lambda_k^n} - \frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \le D \cdot \epsilon.$$

By definition of limit, as $m, n \to \infty$ and $n/m = \alpha$,

$$\Big|\sum_{k=1}^{m}\Big(\frac{1}{\sigma+\lambda_k^n} - \frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \to 0. \tag{3.44}$$

Given that the first quantity in (3.44) is $tr(M_m^n)$ and the second is $tr(\tilde{M}_m^n)$, the conclusion follows. □

Theorem 3.4.3. *$logdet(M_m^n) - logdet(\tilde{M}_m^n) \to 0$, or equivalently*

$$\sum_{k=1}^{m} \log \frac{1}{\sigma+\lambda_k^n} - \sum_{k=1}^{m} \log \frac{1}{\sigma+\tilde{\lambda}_k^n} \to 0.$$

Here, M_m^n and $\tilde{M}_m^n$ are the matrices from Theorem 3.4.2.

Proof. First note that using the mean value theorem and the monotonicity of the *log* function and its derivative, we have that, if $0 < c_1 < x < y < c_2$, then

$$0 < \frac{1}{c_2}(y-x) \le \log\frac{1}{x} - \log\frac{1}{y} \le \frac{1}{c_1}(y-x). \tag{3.45}$$

Again we show that for any $\epsilon > 0$, there exists an integer $N > 0$ such that for any $n > N$,

$$\Big|\sum_{k=1}^{m} \log \frac{1}{\sigma+\lambda_k^n} - \sum_{k=1}^{m} \log \frac{1}{\sigma+\tilde{\lambda}_k^n}\Big| \le D \cdot \epsilon, \tag{3.46}$$

with some positive constant D. First, from (3.31) we choose N_0 such that when $k > N_0$, $\lambda_k^n < \epsilon$ and $\tilde{\lambda}_k^n < \epsilon$ for any n, using a similar argument in the proof of Theorem 3.4.2. Second, from (3.29), we can find $N_1 > 0$ such that for any $n > N_1$, $|\lambda_k^n - \tilde{\lambda}_k^n| < \epsilon^2$ for all $k = 1, 2, ..., n$.

Third, from (3.32) there exists $N_2 > 0$ such that for any $n > N_2$, $\Big| \sum_{k=1}^{m}(\lambda_k^n - \tilde{\lambda}_k^n)\Big| < \epsilon$. We then split the sum in (3.46) into two parts:

$$\sum_{k \leq N_0} \Big(\log \frac{1}{\sigma + \lambda_k^n} - log \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big) + \sum_{k > N_0} \Big(\log \frac{1}{\sigma + \lambda_k^n} - \log \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big).$$

For the first part, we apply (3.45) to obtain

$$\Big| \sum_{k \leq N_0} \Big(\log \frac{1}{\sigma + \lambda_k^n} - \log \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big)\Big| \leq \sum_{k \leq N_0} \frac{|\lambda_k^n - \tilde{\lambda}_k^n|}{\sigma} \leq N_0 \cdot \frac{\epsilon^2}{\sigma} = \frac{C}{\sigma} \cdot \epsilon. \tag{3.47}$$

For the second part, $0 \leq \lambda_k^n$, $\tilde{\lambda}_k^n < \epsilon$, and we discuss two cases.

(1) When $\tilde{\lambda}_k^n > \lambda_k^n$ and $k > N_0$,

$$0 < \frac{1}{\sigma + \epsilon}(\tilde{\lambda}_k^n - \lambda_k^n) \leq \log \frac{1}{\sigma + \lambda_k^n} - \log \frac{1}{\sigma + \tilde{\lambda}_k^n} \leq \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma}.$$

(2) When $\tilde{\lambda}_k^n < \lambda_k^n$ and $k > N_0$,

$$\frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma} \leq \log \frac{1}{\sigma + \lambda_k^n} - \log \frac{1}{\sigma + \tilde{\lambda}_k^n} \leq \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma + \epsilon} < 0.$$

Therefore, we have

$$\begin{aligned}
\sum_{k > N_0} \Big(\log \frac{1}{\sigma + \lambda_k^n} - \log \frac{1}{\sigma + \tilde{\lambda}_k^n}\Big) \leq & \sum_{\tilde{\lambda}_k^n > \lambda_k^n,\ k > N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma} + \sum_{\tilde{\lambda}_k^n < \lambda_k^n,\ k > N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma + \epsilon} \\
= & \sum_{k > N_0} \frac{\tilde{\lambda}_k^n - \lambda_k^n}{\sigma} + \sum_{\tilde{\lambda}_k^n < \lambda_k^n,\ k > N_0} \Big(\frac{1}{\sigma + \epsilon} - \frac{1}{\sigma}\Big)(\tilde{\lambda}_k^n - \lambda_k^n) \\
= & \frac{1}{\sigma} \sum_{k > N_0} (\tilde{\lambda}_k^n - \lambda_k^n) + \frac{\epsilon}{\sigma(\sigma + \epsilon)} \sum_{\tilde{\lambda}_k^n < \lambda_k^n,\ k > N_0} (\lambda_k^n - \tilde{\lambda}_k^n)
\end{aligned}$$

and similarly

$$\begin{aligned}\sum_{k>N_0}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\geq&\sum_{\tilde{\lambda}_k^n>\lambda_k^n,\ k>N_0}\frac{\tilde{\lambda}_k^n-\lambda_k^n}{\sigma+\epsilon}+\sum_{\tilde{\lambda}_k^n<\lambda_k^n,\ k>N_0}\frac{\tilde{\lambda}_k^n-\lambda_k^n}{\sigma}\\ =&\sum_{\tilde{\lambda}_k^n>\lambda_k^n,\ k>N_0}\Big(\frac{1}{\sigma+\epsilon}-\frac{1}{\sigma}\Big)(\tilde{\lambda}_k^n-\lambda_k^n)+\sum_{k>N_0}\frac{\tilde{\lambda}_k^n-\lambda_k^n}{\sigma}\\ =&\frac{1}{\sigma}\sum_{k>N_0}(\tilde{\lambda}_k^n-\lambda_k^n)+\frac{\epsilon}{\sigma(\sigma+\epsilon)}\sum_{\tilde{\lambda}_k^n>\lambda_k^n,\ k>N_0}(\lambda_k^n-\tilde{\lambda}_k^n).\end{aligned}$$

From these two inequalities, we get

$$\begin{aligned}\Big|\sum_{k>N_0}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big|\leq&\frac{1}{\sigma}\Big|\sum_{k>N_0}(\lambda_k^n-\tilde{\lambda}_k^n)\Big|\\ &+\frac{\epsilon}{\sigma(\sigma+\epsilon)}\max\Big\{\sum_{\tilde{\lambda}_k<\lambda_k,\ k>N_0}(\lambda_k^n-\tilde{\lambda}_k^n),\sum_{\tilde{\lambda}_k>\lambda_k,\ k>N_0}(\tilde{\lambda}_k^n-\lambda_k^n)\Big\}.\end{aligned}\tag{3.48}$$

Using the same rationale that led us to (3.40), we have

$$\Big|\sum_{k>N_0}(\lambda_k^n-\tilde{\lambda}_k^n)\Big|\leq\Big|\sum_{k=1}^{n}(\lambda_k^n-\tilde{\lambda}_k^n)\Big|+\Big|\sum_{k\leq N_0}(\lambda_k^n-\tilde{\lambda}_k^n)\Big|\leq(C+1)\epsilon.\tag{3.49}$$

Moreover, using (3.31) and the nonnegativity of the eigenvalues of M^n and $\tilde{M}^n$, we obtain

$$0<\frac{\epsilon}{\sigma(\sigma+\epsilon)}\sum_{\tilde{\lambda}_k<\lambda_k,\ k>N_0}(\lambda_k^n-\tilde{\lambda}_k^n)\leq\frac{\epsilon}{\sigma^2}\sum_{k=1}^{n}\lambda_k^n\leq\frac{C}{\sigma^2}\epsilon\tag{3.50}$$

$$0<\frac{\epsilon}{\sigma(\sigma+\epsilon)}\sum_{\tilde{\lambda}_k>\lambda_k,\ k>N_0}(\tilde{\lambda}_k^n-\lambda_k^n)\leq\frac{\epsilon}{\sigma^2}\sum_{k=1}^{n}\tilde{\lambda}_k^n\leq\frac{C}{\sigma^2}\epsilon.\tag{3.51}$$

Combining (3.48), (3.49), (3.50) and (3.51), we get

$$\Big|\sum_{k>N_0}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big|\leq\frac{C+1}{\sigma}\epsilon+\frac{C}{\sigma^2}\epsilon=\Big(\frac{C+1}{\sigma}+\frac{C}{\sigma^2}\Big)\epsilon\tag{3.52}$$

Using the bounds (3.47) and (3.52), we get

$$\begin{aligned}
&\Big|\sum_{k=1}^{m}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \\
\leq&\Big|\sum_{k\leq N_0}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big|+\Big|\sum_{k>N_0}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big| \\
\leq&\frac{C}{\sigma}\epsilon+\Big(\frac{C+1}{\sigma}+\frac{C}{\sigma^2}\Big)\epsilon=\Big(\frac{2C+1}{\sigma}+\frac{C}{\sigma^2}\Big)\epsilon.
\end{aligned}$$

Let $D=\frac{2C+1}{\sigma}+\frac{C}{\sigma^2}$. We conclude that for any $\epsilon>0$, there exists $N=\max\{N_0,N_1,N_2\}$ such that for any $n>N$,

$$\Big|\sum_{k=1}^{m}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big|\leq D\cdot\epsilon.$$

By definition of limit, as $m,n\to\infty$ and $n/m=\alpha$,

$$\Big|\sum_{k=1}^{m}\Big(\log\frac{1}{\sigma+\lambda_k^n}-\log\frac{1}{\sigma+\tilde{\lambda}_k^n}\Big)\Big|\to 0.$$

Given that the first quantity is the logarithm of the determinant of M_m^n and the second is the logarithm of the determinant of $\tilde{M}_m^n$, this proves the claim. □

Given the relation of Γ_{post} and G_m^n in (3.59), Theorem 3.4.2 proves that the lower bound of the A-optimal design, which is given by the relaxed optimization problem (3.5), can be achieved by using the sum-up rounding strategy. Theorem 3.4.3 does the same for the D-optimal design. The E-optimal design, where we aim to minimize the largest eigenvalue of Γ_{post}, is actually trivial in this framework because the smallest eigenvalue of G_m^n goes to 0 and the largest eigenvalue of $\Gamma_{post}(w^n)$ converges to σ_{pri}, which is also true for $\Gamma_{post}(\tilde{w}^n)$. This argument also shows that the E-optimal result is trivial for this case since virtually any design will then be E-optimal; hence we do not emphasize it in this paper. To conclude, with the sum-up rounding strategy described in §3.4, we are able to find sensor locations that are asymptotically optimal for A and D design criteria.

While our proofs include several restrictions, they can be extended in several ways. To include more general domains or sum-up rounding patterns would require proving results such as Theorem 3.3.4 and, subsequently, the critical property (3.26) needed to show the shrinking gap for a given design strategy. General domains are not difficult to include, but the resulting proofs would be extensive, involving computational geometry technicalities. However, the two-level strategy presented in Definition 1 resembles the spectral element philosophy [68] that is widely used for quite complex domains. Moreover, the within-subdomain ordering in Definition 1 is entirely open, which would allow experimentation with various strategies such as space-filling curves. While our results are proved for linear operators only, we note that as a first step to extending our results to the case where the nonlinear parameter-to-observable map F is nonlinear, one could use the Laplace approximation as was done in [71, 59].

3.5 Numerical Experiments

We now present numerical experiments based on the model problem of gravity surveying (see Example 1.5 in [72]) in our simulation. Suppose mass is distributed at depth d below the surface where sensors can be deployed, in a unit square $[0,1] \times [0,1]$ indexed by the two-dimensional variable y, and we want to estimate the mass density function $g_0(y)$. Measurements are carried out on a unit square in a plane indexed by the two-dimensional variable x, and we can measure the vertical component of gravitational field $g(x)$ but with error. By Newton's law of universal gravitation, the integral equation of $g(x)$ for $x \in [0,1] \times [0,1]$ is

$$g(x) = \int_{[0,1]\times[0,1]} f(x,y) g_0(y)\,\mathrm{d}y, \qquad f(x,y) = \frac{d}{(d^2 + \|x-y\|^2)^{3/2}},$$

where $\|x-y\|$ is the Euclidean distance between points x and y. In this problem, $\Omega_{in} = \Omega_{out}$, and we use the same discretization for the two domains. We divide $[0,1]\times[0,1]$ into n^2 small squares with equal size $1/n^2$. On each side, there are n points $0 < x_1 < x_2 < \dots < x_n < 1$ ($x_i = i/n + 0.5$) and $\Delta_x = 1/n$. We have n^2 candidate locations, and $w = (w_1, w_2, \dots, w_{n^2})$ is the corresponding weight vector. Let $F \in \mathbb{R}^{n^2 * n^2}$ be the discretization of the above integral operator, and order the candidate locations as $z_1, z_2, .., z_{n^2}$. Then

$$F(i,j) = \frac{d}{(d^2 + \|z_i - z_j\|^2)^{3/2}} \cdot (\Delta_x)^2,$$

for $i, j = 1, 2, .., n^2$. Let $W = \text{diag}(w)$. The relaxed problem is

$$\begin{aligned} \min_{w} \quad & \phi\left(\left(F^T W F + \sigma I_{n^2}\right)^{-1}\right) \\ s.t. \quad & 0 \leq w_i \leq 1, \quad \sum_i w_i = \left\lfloor rn^2 \right\rfloor \quad (0 < r < 1), \end{aligned} \tag{3.53}$$

where σ is not a variance but the ratio of σ_{noise} and σ_{pri}. We keep the number of sensors in a proportion r to the number of candidate locations, as discussed at the end of §3.2.5.

Using the solver `Ipopt` in Julia, we compute w_{rel} and then construct a feasible integer vector w_{int} via the sum-up round approach we developed in this paper. Our experiments are run on a recent MacBook Air laptop with 4GB of memory, and we provide the Hessian of the objective and the relevant objective and constraint gradients. By far the most expensive part of the computation is the Hessian. For example, for the case where $\phi(\cdot) = tr(\cdot)$, the entry ij in the Hessian is proportional to $tr(\Gamma^{-1} f_i f_i^T \Gamma^{-1} f_j f_j^T \Gamma^{-1})$, where $\Gamma = \left(F^T W F + \sigma I_{n^2}\right)$ and f_i is the ith column of F^T. Here $1 \leq i < j \leq n^2$, and for the rest of discussion in this parameter, n represents the size of Γ. Note that Γ is a dense matrix. While the computation can be streamlined to carry out the factorization of Γ once per iteration, followed by solving n linear systems of equations with f_i, then computing $\approx \frac{n^2}{2}$ inner products, each of these operations is $O(n^3)$. The largest problem we solve has $n = 3600$ (a 60×60 two-dimensional grid) and `Ipopt` takes about 3 hours to produce a solution for it, though our code is far from optimized. Interestingly, note that computing even one entry in the gradient, whose ith entry is $-tr(\Gamma^{-1} f_i f_i^T \Gamma^{-1})$ would still take $O(n^3)$ as at least one linear system with Γ needs to be solved. For this reason it is doubtful one can do much better, as most convex integer programming solvers need gradients of the objective. In any case, we had difficulties comparing with other approaches, as most of the ones we had reasonably easy access to required the function to be expressible in a modeling environment such as JuMP or AMPL. This does not occur for matrix functions, as they cannot atomically be expressed in terms of standard libraries. An alternative was to reformulate the problem (3.53) as a semidefinite program with integer variables, which we aimed to do with `Pajarito`. However, solving the $n = 50$ case (in one dimension) took one hour to achieve a gap of less than 1%. Therefore this did not appear to be an easy way to go either. Solving larger problems will probably require reaching towards other ideas, such as perhaps exploiting the (approximate) hierarchical off diagonal low rank structure, as we recently proposed in [73].

In any case, results for D-optimal and A-optimal designs using `Ipopt` as described above are demonstrated below. The E-optimal design is not considered because the largest eigenvalue is extremely close to $1/\sigma$ irrespective of w and there is not much difference in objective values for different designs.

We compare our sum-up rounding design with a thresholding heuristic: let

$$w = (w_1, w_2, \cdots, w_{n^2})$$

be the relaxed solution and its components are ordered by $w_{i_1} \geq w_{i_2} \geq \cdots \geq w_{i_{n^2}}$. The thresholding integer solution $\tilde{w}$ is given by

$$\tilde{w}_j = \begin{cases} 1, \text{ if } j \in \{i_1, i_2, \cdots, i_{\lfloor rn^2 \rfloor}\}; \\ 0, \text{ otherwise.} \end{cases}$$

In other words, we set elements to 1 if they have the largest values in the relaxation, up to the available budget of sensors. We will compare the performance of two strategies by measuring integrality gap.

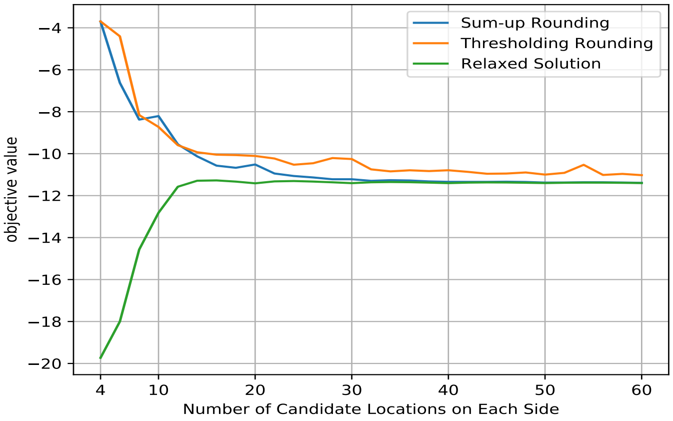


Fig.3.2: Objective value, D-optimal design

3.5.1 D-optimal Design

The parameters we choose are $\sigma = 1, d = 0.1, r = 0.1$. Figure 3.2 shows the objective value (i.e. log determinant) with the continuous relaxation, sum-up rounding and thresholding strategy as n increases from 4 to 60. For the thresholding heuristic, it does not seem to converge at $n = 40$, or at least its gap decreases more slowly than sum-up rounding. We note that this validates the result of Theorem 3.4.2. One point we want to add is the objective value in Figure 3.2 converges to a fixed number (around -11.3), which is related to our choice $\sigma = 1$. Notice, when $\sigma = 1$, that

$$logdet(\Gamma_{post}) = \sum_{k=1}^{n^2} \log \frac{1}{\sigma + \lambda_k} \approx \sum_{k=1}^{n^2} (-\lambda_k) \tag{3.54}$$

and $\sum \lambda_k$ is finite, see (3.31). For other values of σ, the objective value will approach infinity, but the gap will still converge to zero as proved by our theorem.

We also plot the absolute and relative gaps for the two rounding strategies in Figure 3.3, in logarithmic scale. The relative gap is defined as the ratio of absolute gap and the lower bound from the relaxation. We observe that sum-up rounding has a relative gap below 1% at $n = 40$, compared with 5% for the thresholding heuristic.

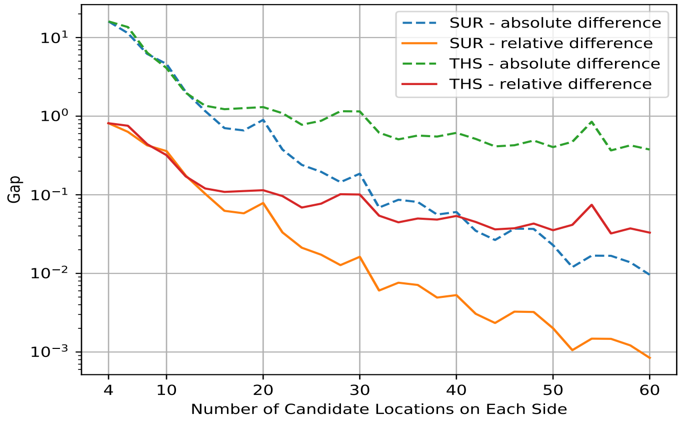


Fig.3.3: Integrality gap, D-optimal design (SUR = sum-up rounding; THS = thresholding rounding)

Figures 3.4, 3.5 and 3.6 give the relaxed solution, the sum-up rounding solution and thresholding solution, respectively, when $n = 40$ (there are 1600 variables). The design is symmetric since both $f(x, y)$ and the output domain $[0, 1] \times [0, 1]$ are symmetric. Sensors are placed toward the boundary and also in the interior. We note that the design highly depends on d: When d goes to zero or infinity, the relaxed solution tends to be uniform. Therefore, if we hope to observe interesting designs, d should be neither too big nor too small. For the thresholding heuristic, a common feature is that sensors tend to be placed together when values in the relaxation change smoothly, and we do not see sensors placed near the center. Sum-up rounding, however, has the property that the 0 or 1 value in the relaxation will remain the same in the integer solution, and the sensor placement is less concentrated than for the thresholding heuristic.

3.5.2 A-optimal Design

We investigate the A-optimal design with the same setting and parameters as in the D-optimal design case: $\sigma = 1, d = 0.1, r = 0.1$, and n starting at 4 and ending at 50. We observe in Figure 3.7 a similar decaying trend as in the D-optimal design case, which validates the

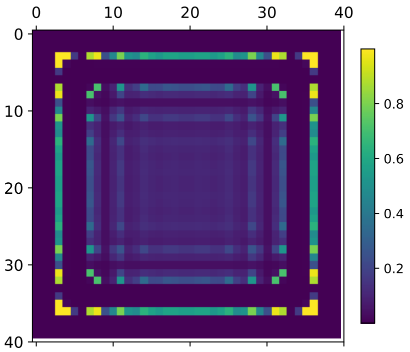


Fig.3.4: Relaxation, D-optimal design

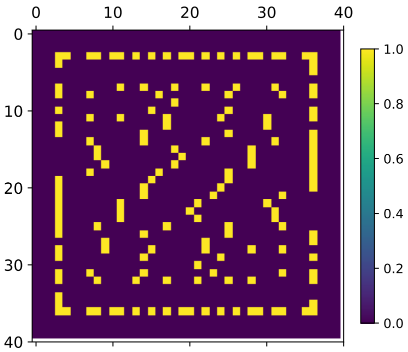


Fig.3.5: SUR solution, D-optimal design

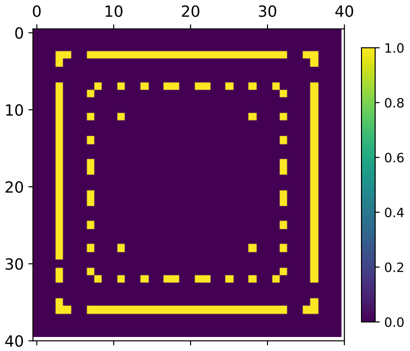


Fig.3.6: THS solution, D-optimal design

finding of Theorem 3.4.3. We would like to mention that in the trace case,

$$tr(\Gamma_{post}) = \sum_{k=1}^{n^2} \frac{1}{\sigma + \lambda_k} = O(n^2), \tag{3.55}$$

so the optimal objective value increases about linearly with respect to the number of candidate locations. However, both the absolute and relative gaps between the upper bound induced by sum-up rounding and the lower bound obtained from the relaxation approach zero for large n, as shown in Figure 3.7 and as claimed in §3.4.

The designs in Figure 3.8, 3.9 and 3.10 also have patterns similar to those in Figure 3.4, 3.5 and 3.6, although they are slightly more centered. It is worth mentioning that, as indicated

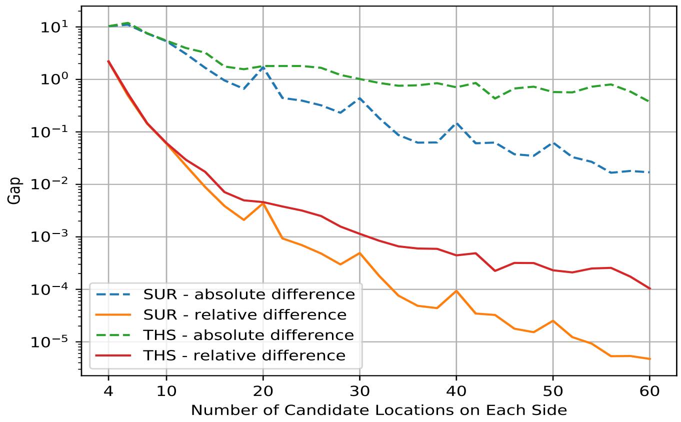


Fig.3.7: Integrality gap, A-optimal design (SUR = sum-up rounding; THS = thresholding rounding)

by Figure 3.3 and 3.7, monotonicity with n is unlikely. Indeed, kinks at $n = 20, 30, \ldots$ are related to the particularities of sum-up rounding design. When n reaches those values, there is a change in shape which induces a small increase in the gap, but the gap will be under control and eventually go to zero.

3.5.3 Discussion

In practice, we normally do not wish to see clusters of sensors because data are usually informative of other data nearby, while sum-up rounding tends to place sensors close to each other because of smoothness in the relaxed solutions. One can request the sensor density not to exceed a given value in any region. An alternative is to use a space-filling curve approach for the sum-up rounding path to "randomize" the choices of 1. For this initial study, we note the significant improvement in the objective, and we leave such issues to further research.

3.6 Extension to Non-Identity Covariance Matrix

Since components in the input are likely to be spatially correlated, it would be unreasonable to assume a Gaussian prior with identity covariance matrix, and in this section, we

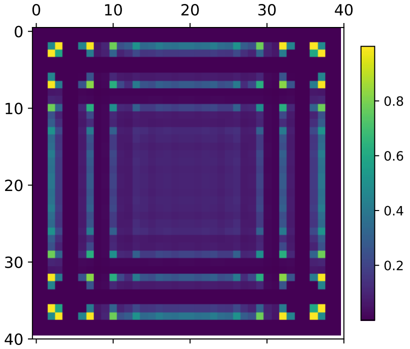


Fig.3.8: Relaxation, A-optimal design

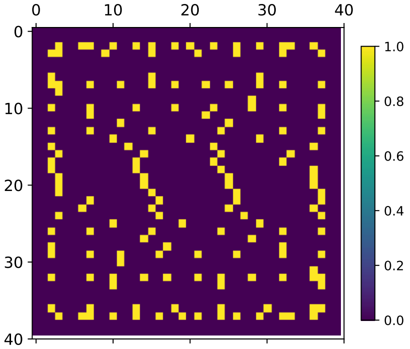


Fig.3.9: SUR solution, A-optimal design

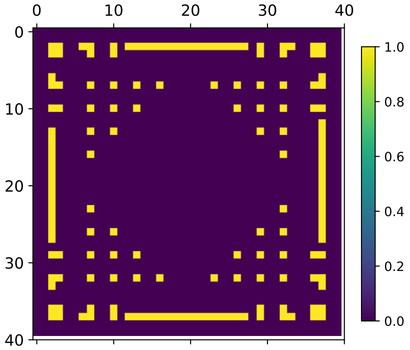


Fig.3.10: THS solution, A-optimal design

extend the previous results to a Gaussian prior with Laplacian precision matrix, which is widely used in image processing, and it is equivalent to a regularized least square problem. We first focus on the one-dimensional case, and then generalize to the multi-dimensional case using tensor product.

Recall that the output without measurement error depends on the input through an integral equation:

$$u(x) = \int_{\Omega_{in}} f(x,y) u_0(y) \, \mathrm{d}y, \quad x \in \Omega_{out}, \tag{3.56}$$

and we approximate the integral (3.56) by Riemann sum:

$$u(x_j) = \int_{\Omega_{in}} f(x_j, y) u_0(y) \, \mathrm{d}y \approx \sum_{i=1}^{m} f(x_j, y_i) u_0(y_i) \Delta_y.$$

Define a matrix $F_m^n \in \mathbb{R}^{n \times m}$ to be the discretization of integral operator

$$F_m^n(j, i) = f(x_j, y_i) \Delta_y,$$

and the prior of our parameter with size m is $u_0^m \sim \mathcal{N}(u_{\text{prior}}^m, L_m)$ where L_m is the discrete Laplacian operator with periodic boundary conditions

$$L_m = \frac{1}{(\Delta_y)^2} \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2. \end{pmatrix} \tag{3.57}$$

The posterior matrix with the relaxed and integer weights are respectively

$$\Gamma_{post}(w^n) = \Big((F_m^n)^T W (F_m^n) + L_m \Big)^{-1}, \quad \Gamma_{post}(\tilde{w}^n) = \Big((F_m^n)^T \tilde{W} (F_m^n) + L_m \Big)^{-1}.$$

We will use F as abbreviation for F_m^n. With some modification ($F \to F_s$), we aim to show

$$\Big| tr\Big(\Gamma_{post}(w^n)\Big) - tr\Big(\Gamma_{post}(\tilde{w}^n)\Big) \Big| \to 0, \quad \text{as } m, n \to \infty \text{ with } n/m \text{ constant.} \tag{3.58}$$

Here w^n is the solution to the relaxed optimization problem and $\tilde{w}^n$ is constructed from the SUR technique. Define $G_m^n = F^T W F$ and $\tilde{G}_m^n = F^T \tilde{W} F$. More specifically,

$$G_m^n = \Delta_y \cdot \{ g^{w^n}(y_i, y_j) \}_{i,j=1}^m \quad \text{and} \quad \tilde{G}_m^n = \Delta_y \cdot \{ g^{\tilde{w}^n}(y_i, y_j) \}_{i,j=1}^m,$$

where

$$g^{w^n}(y_i, y_j) = \alpha \sum_{k=1}^{n} f(x_k, y_i) w^n(x_k) f(x_k, y_j) \Delta_x$$
$$g^{\tilde{w}^n}(y_i, y_j) = \alpha \sum_{k=1}^{n} f(x_k, y_i) \tilde{w}^n(x_k) f(x_k, y_j) \Delta_x.$$

The quantities G_m^n, $\tilde{G}_m^n$, and Γ_{post} satisfy the following relationships

$$\Gamma_{post}(w^n) = \left(G_m^n + L_m\right)^{-1}, \quad \Gamma_{post}(\tilde{w}^n) = \left(\tilde{G}_m^n + L_m\right)^{-1}. \tag{3.59}$$

Denote eigenvalues of G_m^n and $\tilde{G}_m^n$ as

$$\lambda_1^n \geq \lambda_2^n \geq ... \geq \lambda_m^n \geq 0 \tag{3.60}$$
$$\tilde{\lambda}_1^n \geq \tilde{\lambda}_2^n \geq ... \geq \tilde{\lambda}_m^n \geq 0. \tag{3.61}$$

In [74], we showed for any $\epsilon > 0$, there exists a positive integer $M(\epsilon)$ such that when $n > M(\epsilon)$

$$\|G_m^n - \tilde{G}_m^n\|_F < \epsilon, \quad \left|\lambda_k^n - \tilde{\lambda}_k^n\right| < \epsilon, \quad k = 1, 2, ..., m. \tag{3.62}$$

From Theorem 2 in [74], ϵ decays at the rate $n^{-1/2}$ (in one dimension), so $M(\epsilon) \approx \epsilon^{-2}$. In practise, we cannot afford to compute and store the full matrix $F \in \mathbb{R}^{n\times m}$, instead we use interpolatin methods to approximate F, $F \approx F_s$ with $rank(F_s) \leq n_s$, where n_s is the number of interpolation nodes. To be more precise, we apply Chebyshev polynomial approxmation and choose $n_s = O(\log(n))$. From [25], if $f(x, \cdot)$ and $f(\cdot, y)$ is analytical in a compact domain, then

$$\frac{1}{\Delta_y}|F(j,i) - F_s(j,i)| \overset{n\to\infty}{=} O(c^{n_s}) = O(n^{-s}) \tag{3.63}$$

for some $0 < c < 1$ and $s > 0$.For any $\epsilon_s > 0$, we can choose a positive integer $M(\epsilon_s)$ such that when $n > M(\epsilon)$, $\frac{1}{\Delta_y}|F(j,i) - F_s(j,i)| < \epsilon_s$, which implies

$$\|F - F_s\|_F \leq \Delta_y * \sqrt{m * n * \epsilon_s^2} = C\epsilon_s \tag{3.64}$$

where C is a positive constant determined by Ω_{in}, Ω_{out} and α. Note that ϵ_s decays at the rate of n^{-s} which we will use in later proof, so $M(\epsilon_s) \approx \epsilon_s^{-1/s}$. We will show convergence in (3.58) with F replaced by F_s. To do that, we derive the proof into two stages.

Notation: $\|\cdot\|$ is Frobenius norm in the following subsections.

3.6.1 Stage 1: Lower Bound of the Spectrum

In this subsection, we will show for any $m, n > 0$ with n/m constant,

$$\lambda_{min}\Big(G_m^n + L_m\Big) \geq c_0 > 0, \quad \lambda_{min}\Big(\tilde{G}_m^n + L_m\Big) \geq c_0 > 0. \tag{3.65}$$

Or equivalently, $\lambda_m^n \geq c_0$ and $\tilde{\lambda}_m^n > c_0$ in (3.73) and (3.74) respectively. It is known the eigenvalues and eigenvectors of Laplacian matrix in (3.57) are

$$\mu_k^m = \frac{4}{\Delta_y^2}\sin^2\big(\frac{(k-1)\pi}{m}\big), \quad v_{j,k}^m = \frac{1}{\sqrt{m}}\exp\{\frac{i(j-1)(k-1)2\pi}{m}\}$$

where $v_{l,k}^m$ is the j^{th} element in the k^{th} eigenvector of L_m, and i here is the imaginary unit. Because $m\Delta_y = \Omega_{in}$ is a consant, we have asymptotically $\mu_k \to c_k(k-1)^2$. This is also part of Assumption 2.9 in [18] to construct Gaussian priors and obtain well-posedness of Bayesian inverse problems.

The second smallest eigenvalue will exceed some constant $c_2 > 0$ for large n. The eigenvector corresponding to the minimum eigenvalue $\mu_1^m = 0$ is a constant vector $v_1^m = \frac{1}{\sqrt{m}}(1,1,..,1)^T \in \mathbb{R}^m$ and we will show it is not in the null space of F^TWF,

$$(v_1^m)^T F^T W F(v_1^m) = \frac{1}{m}\sum_j w_j\Big(\Delta_y \sum_i f(x_j, y_i)\Big)^2 \approx \frac{C_1}{m}\sum_j w_j\Big(\int_{\Omega_{in}} f(x_j, y)\,\mathrm{d}y\Big)^2$$

for some constant $C_1 > 0$. if $\left(\int_{\Omega_{in}} f(x,y)\,\mathrm{d}y\right)^2$ is bounded below by $C_2 > 0$ for $x \in \Omega_{out}$, then

$$(v_1^m)^T F^T W F(v_1^m) \geq C_1 C_2 \cdot \frac{1}{m}\sum_j w_j$$

As $\frac{1}{n}\sum_j w_j = r$ is a constraint in our optimization, which implies $\frac{1}{m}\sum_j w_j$ is also a constant, there exists an integer $N > 0$ that when $n > N$,

$$(v_1^m)^T F^T W F(v_1^m) > C' := C_1 C_2 \cdot rm/n > 0. \tag{3.66}$$

The same bound applies to $(v_1^m)^T F^T \tilde{W} F(v_1^m)$.

Lemma 3.6.1. *For large enough m, n, there exists $c_0 > 0$ such that* (3.65) *holds.*

Proof. Let $\{v_k^m\}_{k=1}^m$ be eigenvectors of L_m, and they form a basis in $\mathbb{R}^m$. Let $v = \sum_{k=1}^m a_k v_k^m \in \mathbb{R}^m$ be any vector with $\|v\|^2 = \sum_k a_k^2 = 1$. Denote $v = a_1 v_1^m + \tilde{v}_1^m$, where $\tilde{v}_1^m = \sum_{k>1} a_k v_k^m$.

$$\begin{aligned} v^T\Big(G_m^n + L_m\Big)v &= v^T\Big(F^T W F + L_m\Big)v \\ &= a_1^2 \cdot v_1^m F^T W F v_1^m + 2a_1 v_1^m F^T W F \tilde{v}_1^m + (\tilde{v}_1^m)^T F^T W F(\tilde{v}_1^m) + v^T L_m F v \\ &\geq \max\{a_1^2 \cdot v_1^m F^T W F v_1^m + 2a_1 v_1^m F^T W F \tilde{v}_1^m, v^T L_m v\}. \end{aligned} \tag{3.67}$$

Since $v^T L_m v = \sum_{k>1} a_k^2 \lambda_k(L_m) > c_2 \sum_{k>1} a_k^2 = c_2(1 - a_1^2)$ where c_2 is a lower bound of $\lambda_2(L_m)$ for large n, we have $v^T\Big(G_m^n + L_m\Big)v \geq c_2(1 - a_1^2)$ for any $a_1^2 \in [0,1]$. We will find a new lower bound when $|a_1|$ is close to 1. Note

$$\|F\| \leq \Delta_y \sqrt{mnK^2} = \mu(\Omega_{in}) K \sqrt{\frac{m}{n}} =: \tilde{K}$$

where K is an upper bound of $f(x,y)$ (this assumption can be relaxed by $\|Fv\| \leq K\|v\|$). Because W is diagonal and $W_{ii} \in [0,1]$, the same bound applies to $\|WF\|$. Choose a small constant $\epsilon_1 > 0$ and let $a_1^2 = 1 - \epsilon_1^2$, $\sum_{k>1} a_k^2 = \epsilon_1^2$ (so $\|v_1^m\| = \sqrt{1 - \epsilon_1^2}$ and $\|\tilde{v}_1^m\| = \epsilon_1$), we have

$$|a_1 v_1^m F^T W F \tilde{v}_1^m| \leq |a_1| \cdot \|v_1^m\| \cdot \|F\| \cdot \|WF\| \cdot \|\tilde{v}_1^m\| \leq \epsilon_1 \tilde{K}^2. \tag{3.68}$$

From (3.67), (3.66) and (3.68),

$$\begin{aligned} v^T\Big(G_m^n + L_m\Big)v &\geq a_1^2 \cdot v_1^m F^T W F v_1^m + 2a_1 v_1^m F^T W F \tilde{v}_1^m \\ &\geq (1-\epsilon_1^2)C' - 2\epsilon_1 \tilde{K}^2. \end{aligned}$$

When ϵ_1 is small enough, this lower bound is positive. For such ϵ_1, we have

$$v^T\Big(G_m^n + L_m\Big)v \geq \begin{cases} c_2\epsilon_1^2, \text{ when } a_1^2 \leq 1-\epsilon_1^2 \\ (1-\epsilon_1^2)C' - 2\epsilon_1\tilde{K}^2, \text{ when } a_1^2 > 1-\epsilon_1^2. \end{cases}$$

Define c_0 to be $\min\{\epsilon_1^2, (1-\epsilon_1^2)C' - 2\epsilon_1\tilde{K}^2\}$ which is positive, and then for any unit vector $v \in \mathbb{R}^m$,

$$v^T\Big(G_m^n + L_m\Big)v \geq c_0 \quad \Rightarrow \quad \lambda_{min}\Big(G_m^n + L_m\Big) \geq c_0.$$

If we replace W with $\tilde{W}$ in the proof, everything still holds. □

3.6.2 Stage 2: Gap Convergence

In this subsection, we would like to show as $m, n \to \infty$ and keep n/m constant,

$$\left| tr\Big(F_s^T W F_s + L_m\Big)^{-1} - tr\Big(F_s^T \tilde{W} F_s + L_m\Big)^{-1} \right| \to 0.$$

Before we get to the proof, there is some preparation work. We gather what we have so far. From (3.62) and (3.64), we have for any $\epsilon, \epsilon_s > 0$, there are positive integers $M(\epsilon)$ and $M_s(\epsilon_s)$ such that when $n > \max\{M(\epsilon), M_s(\epsilon_s)\}$,

$$\|F^T W F - F^T \tilde{W} F\|_F < \epsilon, \quad \|F - F_s\|_F < \epsilon_s. \tag{3.69}$$

Next we want to show $\|F_s^T W F_s - F_s \tilde{W} F_s\|$ is also small.

Lemma 3.6.2. *For large enough n,*

$$\|F_s^T W F_s - F_s \tilde{W} F_s\| < \epsilon + 2\epsilon_s.$$

Proof. We will prove it in two steps.

- First, we derive an upper bound for $\|F^T W F - F_s^T W F_s\|$ and $\|F^T W F - F_s^T W F_s\|$. Let $E_s = F - F_s$ and note that

  $$F^T W F - F_s^T W F_s = (F_s + E_s)^T W (F_s + E_s) - F_s^T W F_s = E_s^T W F + F^T W E_s - E_s^T W E_s$$

  Because $\|E_s\| < \epsilon_s$ and $\|WF\| < \tilde{K}$ from (3.69), we have

  $$\|F^T W F - F_s^T W F_s\| \leq \|E_s^T W F\| + \|F^T W E_s\| + \|E_s^T W E_s\| < 2\tilde{K}\epsilon_s + \epsilon_s^2$$

  The same bound applies to $\|F^T \tilde{W} F - F_s^T \tilde{W} F_s\|$ as well. Hence we can find another integer $M_s(\epsilon) > 0$ (also $\approx \epsilon_s^{-1/s}$), such that

  $$\|F^T W F - F_s^T W F_s\| < \epsilon_s, \quad \|F^T \tilde{W} F - F_s^T \tilde{W} F_s\| < \epsilon_s. \tag{3.70}$$

- Second, we derive an upper bound for $\|F_s^T W F_s - F_s^T \tilde{W} F_s\|$. From (3.69) and (3.70),

  $$\begin{aligned}
  &\left\|F_s^T W F_s - F_s^T \tilde{W} F_s\right\| \\
  &= \left\|\left(F^T W F - F^T \tilde{W} F\right) + \left(F_s^T W F_s - F^T W F\right) - \left(F_s^T \tilde{W} F_s - F^T \tilde{W} F\right)\right\| \\
  &\leq \left\|F^T W F - F^T \tilde{W} F\right\| + \left\|F_s^T W F_s - F^T W F\right\| + \left\|F_s^T \tilde{W} F_s - F^T \tilde{W} F\right\| \\
  &< \epsilon + 2\epsilon_s.
  \end{aligned}$$

□

We will use it in Theorem 3.6.5 later.

Lemma 3.6.3. *For any $m, n > 0$ with n/m constant,*

$$\lambda_{min}\Big(F_s^T W F_s + L_m\Big) \geq \frac{1}{2}c_0 > 0, \quad \lambda_{min}\Big(F_s^T \tilde{W} F_s + L_m\Big) \geq \frac{1}{2}c_0 > 0.$$

Proof. In Lemma 3.6.1, we already showed $\lambda_{min}\Big(F^T W F + L_m\Big) \geq c_0 > 0$. Together with (3.6.2) and choose $0 < \epsilon_s < \frac{1}{2}c_0$,

$$\lambda_{min}\Big(F_s^T \tilde{W} F_s + L_m\Big) \geq \lambda_{min}\Big(F^T W F + L_m\Big) - \|F^T W F - F_s^T W F_s\| \geq c_0 - \epsilon_s > \frac{1}{2}c_0 > 0.$$

and similarly, we can show $\lambda_{min}\Big(F_s^T \tilde{W} F_s + L_m\Big) > \frac{1}{2}c_0 > 0$. □

Another important component is Lidskii's Theorem (see [75]) and we will state it as follows

Theorem 3.6.4 (Lidskii's Theorem)**.** *Let $A, B \in \mathbb{R}^{n\times n}$ be Hermitian matrices. Then for any choice of indices $1 \leq i_1 < .. < i_k \leq n$,*

$$\sum_{j=1}^{k} \lambda_{i_j}^{\downarrow}(A) - \sum_{j=1}^{k} \lambda_{i_j}^{\downarrow}(B) \leq \sum_{j=1}^{k} \lambda_{j}^{\downarrow}(A - B),$$

where $\lambda^{\downarrow}$ means eigenvalues are in descending order.

Corollary 3.6.1. *With same notation in Lidskii's Theorem, we also have*

$$\sum_{j=1}^{k} \lambda_{i_j}^{\downarrow}(A) - \sum_{j=1}^{k} \lambda_{i_j}^{\downarrow}(B) \geq \sum_{j=1}^{k} \lambda_{n-j}^{\downarrow}(A - B).$$

Theorem 3.6.5. *Denote eigenvalues of $F_s^T W F_s + L_m$ and $F_s^T \tilde{W} F_s + L_m$ as*

$$\nu_1^n \geq \nu_2^n \geq ... \geq \nu_m^n \geq 0 \tag{3.71}$$

$$\tilde{\nu}_1^n \geq \tilde{\nu}_2^n \geq ... \geq \tilde{\nu}_m^n \geq 0. \tag{3.72}$$

then

$$\left| tr\Big(F_s^T W F_s + L_m\Big)^{-1} - tr\Big(F_s^T \tilde{W} F_s + L_m\Big)^{-1} \right| = \Big| \sum_{k=1}^{m} \frac{1}{\nu_k^n} - \sum_{k=1}^{m} \frac{1}{\tilde{\nu}_k^n} \Big| \to 0$$

as $m, n \to \infty$ with n/m constant.

Proof. From Lemma 3.6.3, we know for any $m, n \in \mathbb{Z}^+$,

$$\nu_1^n \geq \nu_2^n \geq ... \geq \nu_m^n \geq \frac{1}{2}c_0 > 0 \tag{3.73}$$

$$\tilde{\nu}_1^n \geq \tilde{\nu}_2^n \geq ... \geq \tilde{\nu}_m^n \geq \frac{1}{2}c_0 > 0. \tag{3.74}$$

To apply Lidskii's Theorem, let $A = F_s^T W F_s + L_m$, $B = F_s^T \tilde{W} F_s + L_m$ and $E = A - B$ which we already have a bound in Lemma 3.6.2 that $\|E\| < \epsilon + 2\epsilon_s$. Note both $F_s^T W F_s$ and $F_s^T \tilde{W} F_s$ have rank $n_s = O(log(n))$ or less, where n_s is the number of interpolation nodes. This implies $rank(E) < 2n_s$ and there are at most $2n_s$ non-zero eigenvalues. Together with the fact $|\lambda_j(E)| \leq \|E\| < \epsilon + 2\epsilon_s$ for any $j \in \{1, 2, .., m\}$, we have for any $k \in \{1, 2, .., m\}$,

$$|\sum_{j=1}^{k} \lambda^{\downarrow}(E)| < \sum_{j=1}^{k} |\lambda^{\downarrow}(E)| < 2n_s(\epsilon + 2\epsilon_s).$$

Because

$$\sum_{k=1}^{m} \frac{1}{\nu_k^n} - \sum_{k=1}^{m} \frac{1}{\tilde{\nu}_k^n} = \sum_{k=1}^{m} \frac{\tilde{\nu}_k^n - \nu_k^n}{\tilde{\nu}_k^n \nu_k^n} = \sum_{\tilde{\nu}_k^n > \nu_k^n} \frac{\tilde{\nu}_k^n - \nu_k^n}{\tilde{\nu}_k^n \nu_k^n} + \sum_{\tilde{\nu}_k^n < \nu_k^n} \frac{\tilde{\nu}_k^n - \nu_k^n}{\tilde{\nu}_k^n \nu_k^n}$$

and $\tilde{\nu}_k^n - \nu_k^n = \lambda_k^{\downarrow}(A) - \lambda_k^{\downarrow}(B)$, we apply Lidskii's Theorem to get

$$\sum_{k=1}^{m} \frac{1}{\nu_k^n} - \sum_{k=1}^{m} \frac{1}{\tilde{\nu}_k^n} < \frac{4}{c_0^2} \sum_{\tilde{\nu}_k^n > \nu_k^n} \frac{\tilde{\nu}_k^n - \nu_k^n}{\tilde{\nu}_k^n \nu_k^n} \leq \frac{4}{c_0^2} \sum_{\tilde{\nu}_k^n > \nu_k^n} \lambda_k^{\downarrow}(E) < \frac{4}{c_0^2} \cdot 2n_s(\epsilon + 2\epsilon_s).$$

With similar reason, we have

$$\sum_{k=1}^{m} \frac{1}{\nu_k^n} - \sum_{k=1}^{m} \frac{1}{\tilde{\nu}_k^n} > \frac{4}{c_0^2} \sum_{\tilde{\nu}_k^n < \nu_k^n} \frac{\tilde{\nu}_k^n - \nu_k^n}{\tilde{\nu}_k^n \nu_k^n} \geq \frac{4}{c_0^2} \sum_{\tilde{\nu}_k^n > \nu_k^n} \lambda_{m-k}^{\downarrow}(E) > -\frac{4}{c_0^2} \cdot 2n_s(\epsilon + 2\epsilon_s).$$

Therefore,

$$\Big| \sum_{k=1}^{m} \frac{1}{\nu_k^n} - \sum_{k=1}^{m} \frac{1}{\tilde{\nu}_k^n} \Big| < \frac{8}{c_0^2} n_s(\epsilon + 2\epsilon_s).$$

It is important to know the behavior of $n_s\epsilon$ and $n_s\epsilon_s$, since even though ϵ and ϵ_s are small, n_s can be large, and it is not obvious whether $n_s(\epsilon + 2\epsilon_s)$ is large or small.

Recall in (3.62) and (3.63) that $\epsilon \sim O(n^{-1/2P})$ with sum-up rounding and $\epsilon_s \sim O(n^{-s})$ with Chebyshev approximation, so $\epsilon + 2\epsilon_s$ has polynomial decay. Since n_s is $O(log(n))$ and $\lim_{n \to \infty} log^{t_1}(n) \cdot n^{-t_2} = 0$ for any $t_1, t_2 > 0$, we conclude that $n_s(\epsilon + 2\epsilon_s)$ goes to zero as $n \to \infty$.

Equivalently, for any fixed $\epsilon' > 0$, there exist $N(\epsilon') > 0$ such that when $n > N(\epsilon')$,

$$\Big| \sum_{k=1}^{m} \frac{1}{\nu_k^n} - \sum_{k=1}^{m} \frac{1}{\tilde{\nu}_k^n} \Big| < \frac{8}{c_0^2} \epsilon',$$

which gives the desired convergence to 0. □

For the D-optimal design where the objective value is $\log\det(\Gamma_{post})$, the proof is quite similar after realizing that the log determinant equals the sum of the logarithm of eigenvalues, and that for any $x, y \geq c > 0$,

$$\Big| \log(\frac{1}{x}) - \log(\frac{1}{y}) \Big| \leq \frac{1}{c} \Big| x - y \Big|.$$

3.6.3 Laplacian Matrix and Convergence in Multiple Dimensions

The main difference in the proof for multiple dimensions is the Laplacian matrix. If we can show its eigenvalues and eigenvectors has a “similar” structure as in one dimension, then we can prove the convergence to zero in the same way. Suppose $\Omega_{in} \in \mathbb{R}^Q$ is a compact domain, and we approximate

$$\nabla u(y_1, y_2, .., y_Q) = \sum_{i=1}^{Q} \frac{\partial^2}{\partial y_i^2}$$

using the tensor products of one dimensional differences. Let D_{y_i} and I_{y_i} be the one dimensional difference and the identity matrix acting on a one dimensional mesh respectively in the y_i direction.

The finite difference Laplacian operator L can be expressed as a sum of Q tensor products

$$Lu = \Big(\sum_{i=1}^{Q} I_{y_1} \otimes \cdots \otimes I_{y_{i-1}} \otimes D_{y_i} \otimes I_{y_{i+1}} \cdots \otimes I_{y_Q} \Big) u.$$

The eigenvectors are also given by tensor products of the one dimensional eigenvectors, and the eigenvalue is given by $\nu = \nu_{y_1} + \nu_{y_2} + \cdots + \nu_{y_Q}$. Based on the discussion of L in one dimension, we know the smallest eigenvector of L is still zero, and the corresponding eigenvector is a constant vector. Further, the second eigenvalue is bounded below by a positive constant for any $n, m > 0$ with n/m constant. Theorem 3.6.5 is also true in multiple dimensions.

3.7 A Different Formulation on Function Space

So far, our parameters are discretized input vector in $\mathbb{R}^n$, and actually we can move from $\mathbb{R}^n$ to $L^2(\Omega_{in})$. Suppose $\{\phi_k\}_{k=1}^{\infty}$ is a basis in $L^2(\Omega_{in})$, and every function $L^2(\Omega_{in})$ can be represented as an infinite sequence in l^2, see (3.75). We transfer the randomness from the input function to its truncated coefficient vector $m^N = (m_1, .., m_N)$,

$$m \in l^2 \xrightarrow{\{\phi_k\}} u_0 \in \Omega_{in} \xrightarrow{\mathcal{F}} u \in \Omega_{out}. \tag{3.75}$$

The relationship between a Gaussian measure on $L^2(\Omega_{in})$ (a Hilbert space) and the distribution of coefficient vector m is given by the following theorem (Theorem 6.19. in [18])

Theorem 3.7.1. *Let $\mathcal{C}$ be a self-adjoint, positive semi-definite, nuclear operator in a Hilbert space $\mathcal{H}$ and let $m \in \mathcal{H}$. Let $\{\phi_k, \gamma_k\}_{k=1}^{\infty}$ be an orthonormal set of eigenvectors/eigenvalues for $\mathcal{C}$ ordered so that*

$$\gamma_1 \geq \gamma_2 \geq \cdots .$$

Take $\{\xi_k\}_{k=1}^{\infty}$ to be an i.i.d. sequence with $\xi_k \sim N(0,1)$. Then the random variable $x \in \mathcal{H}$ given by the Kalhunen-Loève expansion

$$x = x_c + \sum_{k=1}^{\infty} \sqrt{\gamma_k} \xi_k \phi_k$$

is distributed according to $\mu = \mathcal{N}(x_c, \mathcal{C})$.

In Assumption 2.9 of [18], $\gamma_k \sim O(k^{-2})$ is “Laplacian” like. We can construct a matrix $\Phi \in \mathbb{R}^{n \times N}$ with $\Phi_{ij} = \phi_j(x_i)$, where x_i is mesh point in Ω_{in}. The density for the prior is characterized by

$$\pi(m) \propto \exp\{-\frac{1}{2}\|m - m_0\|^2_{\Gamma^{-1}_{prior}}\}$$

where $\Gamma_{prior} = diag(\gamma_1, .., \gamma_N)$. The likelihood is given by

$$\pi(u|m) \propto \exp\{-\frac{1}{2}\|u - F\Phi m\|^2_{\Gamma^{-1}_{noise}}\},$$

where $F \in \mathbb{R}^{n \times n}$ is the discretized parameter-to-observable mapping. The posterior distribution is also Gaussian with covariance function

$$\left(\Phi^T F^T \Gamma^{-1}_{noise} F \Phi + \Gamma^{-1}_{prior}\right)^{-1} \in \mathbb{R}^{N \times N}.$$

Notice that each column of $F\Phi \in \mathbb{R}^{n\times N}$ is a discretization of $\mathcal{F}(\phi_k) \in L^2(\Omega_{out})$. If $\Gamma_{noise} \approx \Delta x^{-1} I_n$ and N is fixed, by looking into each entry of Γ_{post}^{-1} and apply the same technique in the previous proof of Theorem 3.4.2, we can show (without proof) that

$$\left| tr\Big(\frac{\Delta x}{\sigma_{noise}^2}\Phi^T F^T W_n F\Phi + \Gamma_{prior}^{-1}\Big)^{-1} - tr\Big(\frac{\Delta x}{\sigma_{noise}^2}\Phi^T F^T \tilde{W}_n \Phi + \Gamma_{prior}^{-1}\Big)^{-1} \right| \to 0, \quad \text{as } n \to \infty.$$

It's worth mentioning that we do not require $\mathcal{F}$ to be an integral operator here, but require $\mathcal{F}(\phi_k)$ is Lipschitz continuous, which is a weaker assumption. One potential issue is that when the size of Γ_{post} is fixed ($\mathbb{R}^{N\times N}$), the relaxed solution w tends to have N "clusters" and its component tends to be binary already as n increases, so sum-up rounding might not be needed (see [26]).

4 A Scalable Algorithm to Solve the Relaxation

In this section, we provide a fast algorithm to solve the relaxation (3.5) for A-optimal design, and reduce the complexity from $O(n^3)$ with interior point method in §3.5 to $O(n\log^s(n))$. In §4.1, we review the target optimization problem, and in §4.2, we explain the gradient and Hessian approximations with Chebyshev interpolation, and then propose an interpolation-based SQP algorithm in §4.3. An error analysis on the objective gap, together with the choice of interpolation points, are given in §4.4. Finally, we apply the algorithm on the so-called LIDAR problem: selecting sensing directions to infer the initial condition of an advection-diffusion equation in §4.5. The algorithm and error analysis also apply to D-optimal design, but with a different gradient and Hessian than the A-optimal design, and more details can be found in Appendix C.

4.1 Computational Goal

We focus on the criterion of $\phi(M) = trace(M)$ (A-optimal) and solve the following convex optimization problem:

$$\begin{array}{ll} \min & \phi(\Gamma_{post}(w)) \\ \text{s.t.} & 0 \le w_i \le 1,\ \sum_{i=1}^n w_i = n_0 \end{array} \tag{4.1}$$

where $\Gamma_{post} = \left(F^T W F + \sigma^2 I_n\right)^{-1}$ and $\sigma^2 = \sigma^2_{niose}/\sigma^2_{prior}$. For the purpose of easy explanation, we assume for now that $\Omega_{in} = \Omega_{out}$ and they have the same discretizations, but this is not necessary and can be easily generalized. Without loss of generality, we assume $\sigma^2 = 1$ in the discussion of this chapter except the numerical example section. $F \in \mathbb{R}^{n\times n}$ is a discretization of the parameter-to-observable map $\mathcal{F}$ (an integral equation). For instance, in the one-dimensional gravity surveying example, $\mathcal{F}$ maps $C[-1,1]$ to $C[-1,1]$ by

$$u(x) = \mathcal{F}(u_0) = \int_{[-1,1]} f(x,y)u_0(y)\,\mathrm{d}y, \qquad f(x,y) = \frac{d}{(d^2 + \|x-y\|^2)^{3/2}} \tag{4.2}$$

from the example of gravity surveying. We discretize it on a regular n-grid and get

$$F(i,j) = \frac{d \cdot \Delta x}{(d^2 + \|x_i - x_j\|^2)^{3/2}},$$

where $x_i(x_j)$ is the center of the $i^{th}(j^{th})$ interval of length Δx. $W = diag\{w_1, w_2, .., w_n\}$ is the weight matrix and w_i is the wight associated with the i^{th} candidate location.

4.2 Chebyshev Interpolation Method

The complexity of computing the gradient and Hessian of the trace objective in (4.1) is $O(n^3)$ where n is the mesh size or the number of candidate sensor locations, and in practice it can easily go to thousands or millions. By exploiting the continuously indexed structure of out problem, we find that the interpolation method can give accurate approximations.

4.2.1 Gradient and Hessian for A-optimal Design

Both gradient and Hessian can be easily calculated by taking partial derivative of $tr(\Gamma_{post}(w))$ with respect to w_i, and they are provided as follows.

- **Gradient** Denote f_i as the ith column of F^T and we have

$$F^TWF = \sum_{i=1}^n w_i f_i f_i^T \quad \Rightarrow \quad \frac{\partial F^TWF}{\partial w_i} = f_i f_i^T.$$

Therefore the i^{th} component in the gradient is:

$$\frac{\partial tr(\Gamma_{\text{post}})}{\partial w_i} = -tr\Big((F^TWF + I_n)^{-1} f_i f_i^T (F^TWF + I_n)^{-1}\Big) = -\|(F^TWF + I_n)^{-1} f_i\|^2. \tag{4.3}$$

- **Hessian** Following the previous steps, the $(i,j)^{th}$ entry of Hessian matrix is:

$$H_{ij} = \frac{\partial^2 tr(\Gamma_{\text{post}})}{\partial w_i \partial w_j} = 2\Big(f_i^T(F^TWF+I_n)^{-1}f_j\Big)\Big(f_i^T(F^TWF+I_n)^{-2}f_j\Big). \tag{4.4}$$

Note that f_i is discretized from a smooth function $f(x_i,\cdot)$, so both f_i and F are continuously indexed, and next we explain an approximation of the gradient and Hessian with Chebyshev interpolation.

4.2.2 Chebyshev Interpolation in 1D

For starters, assume our domain is $[-1,1]$, the N Chebyshev interpolation points are

$$\tilde{x}_i = \cos\Big(\frac{\pi(i-1)}{N-1}\Big), \quad i = 1,2,..,N.$$

Suppose we want to interpolate a smooth function $h:[-1,1]\to\mathbb{R}$, and function evaluations are available at the set of interpolation points $\{(\tilde{x}_1,h(\tilde{x}_1)),(\tilde{x}_2,h(\tilde{x}_2)),..,(\tilde{x}_N,h(\tilde{x}_N))\}$. Then for any $x\in[-1,1]$, $h(x)$ can be approximated using Lagrange basis polynomials:

$$h(x) = \sum_{i=1}^{N}\big(\prod_{j\neq i}\frac{x-\tilde{x}_j}{\tilde{x}_i-\tilde{x}_j}\big)h(\tilde{x}_i).$$

The number of interpolation points is chosen to be $N = O(log(n))$ for both computational and accuracy purposes which we will see in §4.4. The coefficient vector associated with x is

$$c(x) = \Big(\prod_{j\neq 1}\frac{x-\tilde{x}_1}{\tilde{x}_i-\tilde{x}_1}, \prod_{j\neq 2}\frac{x-\tilde{x}_2}{\tilde{x}_i-\tilde{x}_2}, ..., \prod_{j\neq N}\frac{x-\tilde{x}_N}{\tilde{x}_i-\tilde{x}_N}\Big)^T \in \mathbb{R}^N.$$

To apply it in our problem, we construct a square matrix $\tilde{F}\in\mathbb{R}^{N\times N}$ with

$$\tilde{F}(i,j) = f(x_i,y_j)\Delta y \quad \Big(e.g.\ \tilde{F}(i,j) = \frac{d\cdot\Delta y}{(d^2+\|\tilde{x}_i-\tilde{y}_j\|^2)^{3/2}}\Big)$$

where $\tilde{x}_i$ and $\tilde{y}_j$ are interpolation points in Ω_{out} and Ω_{in} respectively. We calculate the coefficient vector $c(x_i)$ for each mesh point $x_i \in \Omega_{in}$ and create $C_x = \big(c_x(x_1), c_x(x_2), ..., c_x(x_n)\big) \in \mathbb{R}^{N\times n}$. Similarly we construct C_y and then approximate F by

$$F_s := C_x^T * \tilde{F} * C_y \in \mathbb{R}^{n\times n}. \tag{4.5}$$

To approximate the gradient and Hessian in (4.3) and (4.4), we construct $M \in \mathbb{R}^{n\times N}$ with its i^{th} column m_i given by

$$\Big(F_s^T W F_s + I_n\Big)^{-1} \tilde{f}_i. \tag{4.6}$$

where $\tilde{f}_i$ is the i^{th} column of $C_y^T \tilde{F}^T$, as an approximation of the column in F^T evaluated at $\tilde{x}_i$ (note $\tilde{x}_i$ is an interpolation point which may not be a mesh point). We apply the conjugate gradient algorithm (see [76, §5.1]) to solve the linear system (4.6) using matrix-vector product only. Note F is from an integral operator which is of trace class, and the sum of all the eigenvalues of the positive semi-definite matrix $F^T W F$ is finite, so the condition number of $F^T W F + I_n$ is $\mathcal{O}(1)$, and thus we do not need a preconditioner for the conjugate gradient algorithm (see [60]). We then define $M_1, M_2 \in \mathbb{R}^{N\times N}$ where the $(i,j)^{th}$ entry is $\langle \tilde{f}_i, m_j\rangle$ and $\langle m_i, m_j\rangle$ respectively. More specifically, for $i, j = 1, 2, ..., N$,

$$M_1(i,j) = \tilde{f}_i^T \Big(F_s^T W F_s + I_n\Big)^{-1} \tilde{f}_j, \quad M_2(i,j) = \tilde{f}_i^T \Big(F_s^T W F_s + I_n\Big)^{-2} \tilde{f}_j.$$

- **Approximate gradient in** (4.3)**.** Let $g \in \mathbb{R}^n$ be the true gradient, i.e.

  $$g = (\frac{\partial\phi}{\partial w_1}, \frac{\partial\phi}{\partial w_2}, ..., \frac{\partial\phi}{\partial w_n})^T.$$

  and we approximate each component by $g_i \approx -c_x(x_i)^T * M_2 * c_x(x_i)$.

- **Approximate Hessian in** (4.4)**.** We construct another matrix $\tilde{H} \in \mathbb{R}^{N\times N}$ where $\tilde{H}(i,j) = 2 * M_1(i,j) * M_2(i,j)$, then $H(i,j)$ is approximated by $c_x(x_i)^T * \tilde{H} * c_x(x_j)$.

Equivalently,

$$H \approx H_s = C_x^T * \tilde{H} * C_x.$$

The above interpolation-based approximation can be generalized for any interval domain $[a, b]$ by defining a one-to-one mapping between $[a, b]$ and $[-1, 1]$.

4.2.3 Chebyshev Interpolation in 2D

After we understand how the interpolation method works in one dimension, it is not difficult to extend it to multiple dimensions using tensor product, though the notation would be slightly more complicated.

Consider the same input and output domain $\Omega = [-1, 1] \times [-1, 1]$, and let n_{each} and N_{each} be the number of mesh points and interpolation points respectively on each side of the domain. We have $n = n_{each}^2$ mesh points and $N = N_{each}^2$ interpolation points in Ω and they are related by

$$N = N_{each}^2 = O\big(log(n_{each}^2)\big) = O(log(n)).$$

In the example of two dimensional gravity surveying, the integrand in (4.2) becomes

$$f\big((x, y), (x', y')\big) = \frac{d}{(d^2 + \|(x, y) - (x', y')\|^2)^{3/2}} = \frac{d}{(d^2 + (x - x')^2 + (y - y')^2)^{3/2}}.$$

Suppose $\{(x_i, y_j)\}_{i,j=1}^{n_{each}}$ are mesh points, and $\{(\tilde{x}_i, \tilde{y}_j)\}_{i,j=1}^{N_{each}}$ are interpolation points, and we construct matrices $F \in \mathbb{R}^{n \times n}$ and $\tilde{F} \in \mathbb{R}^{N \times N}$ in a similar fashion as in one dimension. The n mesh points are ordered as follows: for an index $k \in \{1, 2, .., n\}$, we write

$$k = (i - 1) * n_{each} + (j - 1),$$

and it is associated with the mesh point (x_i, y_j) in the domain. In other words, we arrange these mesh points "column by column", and the index k is associated with (x_i, y_j). We apply the same ordering to interpolation points. Next we find the coefficient vector $c(x_i, y_j) \in \mathbb{R}^N$, i.e. how a general function $f(x_i, y_j)$ depends on the values at interpolation points. Based on results from one dimension, let $c(x_i), c(y_j) \in \mathbb{R}^{N_{each}}$ be the one-dimensional coefficient vector for x_i and y_j, and $k \in \{1, 2, .., N\}$ with

$$k = (k_1 - 1) * N_{each} + (k_2 - 1),$$

then the k^{th} coefficient for $f(x_i, y_j)$ is given by

$$c(x_i, y_j)_k = c(x_i)_{k_1} * c(y_j)_{k_2}.$$

The k^{th} component in $c(x_i, y_j) \in \mathbb{R}^N$ is the product of k_1^{th} component in $c(x_i)$ and k_2^{th} component in $c(y_j)$. We can calculate the coefficient vector for each mesh point, and create matrices C, M, $M1$ and $M2$ in a similar fashion (details omitted). Gradient and Hessian are approximated in the same way as one dimension.

4.3 Sequential Quadratic Programming (SQP)

Given the (approximated) gradient and Hessian to our optimization program (4.1), we solve a sequence of quadratic program until convergence, where at each step, we approximate the objective by a quadratic Taylor polynomial at the current iterate. We adopt the algorithm from [76, §18.1] and each quadratic program is solved with the interior point method.

Before we get to the program (4.1), we instead solve a slightly different version:

$$\begin{aligned} \min \quad & \phi(\Gamma_{post}(w)) \\ \text{s.t.} \quad & 0 \le w_i \le 1, \ \textstyle\sum_{i=1}^n w_i \le n_0. \end{aligned} \tag{4.7}$$

Claim 4.3.1. *This program and the original program has the same minimal point.*

Proof: Note that if $w \preceq w'$ ($w_i \le w_i'$ for each i), then $\Big(F^T W F + I_n\Big)^{-1} \succeq \Big(F^T W' F + I_n\Big)^{-1}$.
■

4.3.1 A Framework for SQP

Suppose at the k^{th} iteration, (w^k, λ^k) (note w^k is not k^{th} component of w, but k^{th} iterate of w) are respectively the primal and dual variable, we solve the following quadratic program

$$\begin{array}{ll} \min & \phi_k + \nabla \phi_k^T \cdot p^k + \frac{1}{2}(p^k)^T \cdot \nabla^2_{ww}\mathcal{L}_k \cdot p^k \\ \text{s.t.} & -w_i^k \leq p_i^k \leq 1 - w_i^k, \ , i = 1, 2, ..., n, \\ & \sum_{i=1}^n p_i^k \leq n_0 - \sum_{i=1}^n w_i^k \end{array} \tag{4.8}$$

where $\nabla \phi_k$ is the gradient of the objective ϕ and $\nabla^2_{ww}\mathcal{L}_k$ is the Hessian of the Lagrangian, evaluated at the current iterate w^k. As there are only linear constraints, we have $\nabla^2_{ww}\mathcal{L}_k = \nabla^2_{ww}\phi_k = H_k$. The problem (4.8) can be simplified as:

$$\begin{array}{ll} \min & g^T \cdot p^k + \frac{1}{2}(p^k)^T \cdot H \cdot p^k \\ \text{s.t.} & A \cdot p^k \geq b. \end{array} \tag{4.9}$$

where $g = \nabla \phi_k$, $H = \nabla^2_{ww}\phi_k$, $A = \begin{pmatrix} I_n \\ -I_n \\ -\mathbf{1}^T \end{pmatrix} \in \mathbb{R}^{(2n+1)\times n}$, $b = \begin{pmatrix} -w^k \\ w^k - \mathbf{1} \\ \mathbf{1}^T w^k - n_0 \end{pmatrix} \in \mathbb{R}^{2n+1}$.

Both g and H are from Chebyshev approximations. The new iterate w^{k+1} is updated by $w^k + \alpha_k p^k$ where p^k is the solution to the quadratic program (4.9), and α_k is the step length determined by backtracking line search (see [76, Algorithm 3.1]). We discuss details on solving the program (4.9) and getting its Lagrangian multipliers in the next subsection, but provide the SQP framework now in Algorithm 1.

In the backtracking line search step, we need to evaluate $\phi(\Gamma_{post}(w))$ which involves the trace of an inverse matrix of size $n \times n$, and we propose a SVD-based method with complexity $\mathcal{O}(n \log^2(n))$ for the evaluation. Recall that $F_s = C_x^T \tilde{F} C_y$ in (4.5) where $C_x, C_y \in \mathbb{R}^{N\times n}$,

$\tilde{F} \in \mathbb{R}^{N\times N}$ and $N = \mathcal{O}(\log(n))$, then

$$F_s^T W F_s = C_y^T \tilde{F}^T C_x W C_x^T \tilde{F} C_y.$$

To evaluate $\phi(\Gamma_{post}(w))$, we compute the eigenvalues $\{\lambda_i\}_{i=1}^n$ of $F_s^T W F_s$ because

$$\phi(\Gamma_{post}(w)) = tr\big((F_s^T W F_s + I_n)^{-1}\big) = \sum_{i=1}^{n} \frac{1}{1+\lambda_i}.$$

SVD decompositions are applied to $C_y^T \tilde{F} \in \mathbb{R}^{n\times N}$ and $C_x W^{1/2} \in \mathbb{R}^{N\times n}$ respectively, and the complexity is $\mathcal{O}(n\log^2(n))$. After that, we get

$$C_y^T \tilde{F} = U_1\Sigma_1 V_1^T,\ C_x W^{1/2} = U_2\Sigma_2 V_2^T \quad \Rightarrow \quad F_s^T W F_s = U_1\Big(\Sigma_1 V_1^T U_2 \Sigma_2^2 U_2^T V_1 \Sigma_1\Big)U_1^T.$$

Because $\Sigma_1 V_1^T U_2 \Sigma_2^2 U_2^T V_1 \Sigma_1 \in \mathbb{R}^{N\times N}$ is of smaller size, another SVD decomposition (equivalently eigenvalue decomposition) of this matrix directly gives us the eigenvalues of $F_s^T W F_s$, and then the value of $\phi(\Gamma_{post}(w))$.

Algorithm 1 SQP with line search for Solving (4.7)
($c = 0.5, \xi = 10^{-3}, \epsilon_1 = 10^{-5}, \epsilon_2 = 10^{-8}$)

```
*choose an initial state* $(w^0, \lambda^0)$*; set* $k \leftarrow 0$
**repeat** until KKT optimality violation $< \epsilon_1$
    evaluate $\nabla\phi_k, \nabla^2_{ww}\phi_k, A_k, b_k$;
    solve the quadratic program (4.9) to obtain $(p^k, \lambda^{k+1})$;
    $\alpha_k = 1$
    **while** $\phi(\Gamma_{post}(w^k + \alpha p^k)) > \phi(\Gamma_{post}(x^k)) + \xi\alpha_k\nabla\phi_k^T p^k$
        $\alpha_k = c * \alpha_k$
    **end (while)**
    **if** $\alpha_k \cdot \|p^k\|_\infty < \epsilon_2$, **stop** (either $p^k$ is small or $p^k$ is not a descent direction)
    **if** $\|\alpha_k \cdot p^k + \alpha_{k-1} \cdot p^{k-1}\|_\infty < \epsilon_1$, **stop** (moving back and forth between iterates)
    set $w^{k+1} \leftarrow w^k + \alpha_k \cdot p^k$, $\lambda^{k+1} \leftarrow \lambda^k + \alpha_k(\lambda^{k+1} - \lambda^k)$;
**end (repeat)**;
```

Ideally the KKT optimality in the stopping criterion is with respect to the program (4.7), and its Lagrangian is

$$\mathcal{L}(w,\mu,\lambda,\tilde{\lambda}) = \phi\big(\Gamma_{post}(w)\big) + \mu\big(\sum_{i=1}^{n} w_i - n_0\big) + \sum_{i=1}^{n} \lambda_i(-w_i) + \sum_{i=1}^{n} \tilde{\lambda}_i(w_i - 1),$$

but computing the derivative of $\mathcal{L}$ is as expensive as computing $\nabla\phi$, so it is approximated with Chebyshev interpolation. Given a feasible primal variable w and feasible dual variable $(\mu, \lambda, \tilde{\lambda})$, we define the KKT optimality violation as

$$\max\{\big\|\nabla\phi\big(\Gamma_{post}(w)\big) + \mu - \lambda + \tilde{\lambda}\big\|_\infty, \big|\mu(\sum_{i=1}^{n} w_i - n_0)\big|, \big|\lambda_i(-w_i)\big|, \big|\tilde{\lambda}_i(w_i-1)\big|, i = 1,2,..,n.\}. \tag{4.10}$$

Note that the program (4.7) is convex, so the KKT condition is both necessary and sufficient for a solution to be optimal, and we can examine the performance of a solution by looking at its KKT optimality violation (4.10).

The two additional stopping criteria on Line 9 and 10 are to account for the approximation errors in the gradient and Hessian, because it is a question as to whether $\{w^k\}$ converges to the true solution. We further show in §4.4 that, when the KKT condition on Line 2 is satisfied, the SQP solution converges to the true solution as the problem size goes to infinity. In addition, we would like to mention that with other stopping criteria, such as the decrease in the objective is less than ϵ, the algorithm will produce similar results.

4.3.2 Solve QP with Interior Point Method

In this subsection, we focus on solving the program (4.9) with an interior point method, following the procedure in [76, §16.6]. We introduce slack variable $s \succeq 0$ and write down the

KKT condition for (4.9):

$$
\begin{aligned}
&H \cdot p^k + g - A^T \lambda = 0\\
&A \cdot p^k - s - b = 0\\
&s_i \cdot \lambda_i = 0, \quad i = 1, 2, ..., 2n+1\\
&(s,\ \lambda) \succeq 0.
\end{aligned}
$$

Define a complementarity measure $\mu = s^T \cdot \lambda/(2n+1)$, and solve a linear system:

$$
\begin{pmatrix} H & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & S \end{pmatrix} \begin{pmatrix} \Delta p^k \\ \Delta s \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -r_d \\ -r_p \\ -\Lambda \cdot S\mathbf{1} + \sigma \cdot \mu \mathbf{1} \end{pmatrix} \tag{4.11}
$$

where

$$
r_d = H \cdot p^k - A^T \lambda + g, \quad r_p = A \cdot p^k - s - b
$$

and

$$
\Lambda = \mathrm{diag}(\lambda_1, .., \lambda_{2n+1}), \quad S = \mathrm{diag}(s_1, .., s_{2n+1}), \quad \mathbf{1} = (1, 1, ..,)^T.
$$

A more compact "normal equation" form of the system (4.11) is

$$
\Big(H + A^T S^{-1} \Lambda A\Big) \Delta p^k = -r_d + A^T S^{-1} \Lambda \big(- r_p - s + \sigma \mu \Lambda^{-1} \mathbf{1}\big) \tag{4.12}
$$

Next we solve the linear system (4.12). Note that once Δp^k is known, Δs and $\Delta \lambda$ can be derived without much effort. Let $S^{-1}\Lambda = \mathrm{diag}(d_1, d_2, .., d_{2n+1})$, we have from (4.9) that

$$
A^T S^{-1} \Lambda A = D + d_{2n+1} \cdot \mathbf{1} \cdot \mathbf{1}^T
$$

where $D = \text{diag}(d_1+d_{n+1}, d_2+d_{n+2}, .., d_n+d_{2n})$. Then we apply Sherman-Morrison formula to calculate $(H + D + d_{2n+1} \cdot \mathbf{1} \cdot \mathbf{1}^T)^{-1}$. Since $H \approx H_{approx} = C^T \tilde{H} C$, we have

$$\Big(H_{approx}+D\Big)^{-1} = \Big(C^T\tilde{H}C+D\Big)^{-1} = D^{-1}-D^{-1}C^T\Big(\tilde{H}^{-1}+CD^{-1}C^T\Big)^{-1}CD^{-1} \quad (4.13)$$

because $\tilde{H}$ is of much lower dimension $\mathcal{O}(\log(n))$ than H, and it is less expensive to find its inverse. Let $X := H_{approx} + D$, and we apply (4.13) to get

$$(H+D+d_{2n+1}\cdot\mathbf{1}\cdot\mathbf{1}^T)^{-1} = \Big(X+d_{2n+1}\mathbf{1}\cdot\mathbf{1}^T\Big)^{-1} = X^{-1}-X^{-1}\mathbf{1}\mathbf{1}^TX^{-1}/\Big(\mathbf{1}^TX^{-1}\mathbf{1}+d_{2n+1}^{-1}\Big).$$

To solve for Δp^k in (4.12), we only need $\tilde{H}^{-1}$ and matrix vector products with complexity $\mathcal{O}(\log^3(n))$ and $\mathcal{O}(n\log(n))$ respectively, and both are affordable to compute. In particular, Algorithm 16.4 in [76] is implemented to solve (4.9). Together with the fact that the number of iterations with increasing variable dimensions is usually stable for interior point algorithms, our SQP algorithm has an overall complexity of $O(n\log^s(n))$ for some positive $s \leq 3$.

4.4 Error Analysis - Convergence in Optimality Gap

In this section, we determine the number of Chebyshev interpolation points N to approximately achieve an accuracy level ϵ. Specifically, let w^N be the solution from SQP (Algorithm 1), and w^n be the solution to (4.7), we want to choose N based on n such that

$$\Big|\phi\big(\Gamma_{post}(w^n)\big) - \phi\big(\Gamma_{post}(w^N)\big)\Big| < \epsilon$$

where ϵ is a preassigned threshold. Basically we show that if we solve the optimization with the low-rank approximation matrix F_s (see the program (4.14) in the next subsection), then its objective value converges to the true minimum as $n \to \infty$. We address this problem in two steps.

4.4.1 Connection Between Two Optimization Problems

Note that the KKT optimality in Algorithm 1 corresponds to the following program:

$$\begin{aligned} \min \quad & \phi_s(\Gamma_{post}(w)) \\ \text{s.t.} \quad & 0 \leq w_i \leq 1, \ \textstyle\sum_{i=1}^n w_i \leq n_0. \end{aligned} \tag{4.14}$$

where $\phi_s(\Gamma_{post}(w)) = \phi\big((F_s^T W F_s + I)^{-1}\big)$. The program differs from (4.7) only in F, and for simplicity, we use the abbreviation $\phi_s(w)$ for $\phi_s(\Gamma_{post}(w))$, and $\phi(w)$ for $\phi(\Gamma_{post}(w))$.

Claim 4.4.1. *Let w^N, w^n be the solution to* (4.14) *and* (4.7) *respectively. If* $|\phi(w) - \phi_s(w)| < \epsilon$ *for any $w \in \mathbb{R}^n$, then*

$$|\phi(w^N) - \phi(w^n)| < 2\epsilon.$$

It tells us if ϕ_s is close to ϕ for any w, then the objective value with the interpolation solution is close to the true minimum.

Proof: Because w^N and w^n minimizes $\phi_s(w)$ and $\phi(w)$ respectively, we have

$$\phi_s(w^N) \leq \phi_s(w^n), \quad \phi(w^n) \leq \phi(w^N). \tag{4.15}$$

From the assumption we know $|\phi(w^N) - \phi_s(w^N)| < \epsilon$ and $|\phi(w^n) - \phi_s(w^n)| < \epsilon$, and together with (4.15), we get

$$\phi(w^N) \leq \phi_s(w^N) + \epsilon \leq \phi_s(w^n) + \epsilon < \phi(w^n) + 2\epsilon. \tag{4.16}$$

The result follows directly from (4.15) and (4.16). ■

It remains to show $|\phi(w) - \phi_s(w)|$ is small for any $w \in \mathbb{R}^n$. Because each entry in $\frac{1}{\Delta y} F_s$ is an approximation of $\frac{1}{\Delta y} F$ (which is equal to $f(x_i, y_j)$, and Δy is the size of a unit rectangle in Ω_{in}), their difference is small because of the Chebyshev polynomial approximtaion ([25]). We now quantify $|\phi(w) - \phi_s(w)|$.

We use the notation $\|X\| = \|X\|_F$ (Frobenius norm) for any matrix X in this chapter.

Claim 4.4.2. *If $\frac{1}{\Delta y}|F(i,j) - F_s(i,j)| < \epsilon$, then for any $w \in \mathbb{R}^n$,*

$$|\phi(w) - \phi_s(w)| \leq C \cdot N \cdot \epsilon,$$

for some positive constant C independent of n and N.

Note F_s is defined in (4.5) with N interpolation points, and ϵ represents the interpolation error which we will quantify in the next subsection.

Proof: Because $|F(i,j) - F_s(i,j)| < \epsilon\Delta_y$ for $i, j = 1, 2, .., n$, we have

$$\|F - F_s\| < \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} \epsilon^2 \Delta^2 y} = n\Delta y \cdot \epsilon = \mu(\Omega_{in}) \cdot \epsilon. \tag{4.17}$$

Similarly because $|\frac{1}{\Delta y}F(i,j)| = |f(x_i, y_j)| \leq \max f(x,y)$,

$$\|F\| = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} F(i,j)^2(\Delta y)^2} \leq n\Delta y \cdot \max f(x,y) = \mu(\Omega_{in}) \cdot \max f(x,y). \tag{4.18}$$

Moreover, we can show $\|F_s\|$ is also bounded

$$\|F_s\| = \|F_s + F - F\| \leq \|F\| + \|F - F_s\| \leq \mu(\Omega_{in}) \cdot \max f(x,y) + \mu(\Omega_{in}) \cdot \epsilon.$$

When ϵ is small (e.g. $\epsilon < \max f(x,y)$), we get

$$\|F_s\| \leq 2\mu(\Omega_{in}) \cdot \max f(x,y). \tag{4.19}$$

Because W is a diagonal matrix with each component between 0 and 1, the matrix product WF results in multiplying the i^{th} row of F by w_i and thus $\|WF\| \leq \|F\|$. For similar reasons, we have $\|FW\| \leq \|F\|$ and get

$$\begin{aligned}\left\|F^TWF - F_s^TWF_s\right\| &\leq \left\|F^TW(F-F_s)\right\| + \left\|(F-F_s)^TWF_s\right\| \\ &\leq \|F^TW\| \cdot \|(F-F_s)\| + \|F-F_s\| \cdot \|WF_s\| \\ &\leq \|F\| \cdot \|(F-F_s)\| + \|F-F_s\| \cdot \|F_s\| \\ &< c\epsilon\end{aligned}$$

where the postive constant $c = 3 \cdot \mu^2(\Omega_{in}) \cdot \max f(x,y)$ from (4.17), (4.18) and (4.19). Let

$$\lambda_1^n \geq \lambda_2^n \geq \cdots \geq \lambda_n^n, \quad \lambda_1^{n,s} \geq \lambda_2^{n,s} \geq \cdots \geq \lambda_n^{n,s}$$

be the eigenvalues of F^TWF and $F_s^TWF_s$ respectively. In [74], it has been proved

$$|\lambda_i^n - \lambda_i^{n,s}| < \|F^TWF - F_s^TWF_s\| < c\epsilon. \tag{4.20}$$

Because the rank of F_s is at most N, we have $\lambda_{N+1}^{n,s} = .. = \lambda_n^{n,s} = 0$. In the trace case,

$$\begin{aligned}|\phi(w) - \phi_s(w)| &= |\sum_{i=1}^{n} \frac{1}{1+\lambda_i^n} - \sum_{i=1}^{n} \frac{1}{1+\lambda_i^{n,s}}| \\ &\leq |\sum_{i=1}^{N} (\frac{1}{1+\lambda_i^n} - \frac{1}{1+\lambda_i^{n,s}})| + |\sum_{i=N+1}^{n} (\frac{1}{1+\lambda_i^n} - \frac{1}{1+\lambda_i^{n,s}})| \\ &= \sum_{i=1}^{N} \frac{|\lambda_i^n - \lambda_i^{n,s}|}{(1+\lambda_i^n)(1+\lambda_i^{n,s})} + \sum_{i=N+1}^{N} \frac{\lambda_i^n}{1+\lambda_i^n} \\ &\leq \sum_{i=1}^{N} |\lambda_i^n - \lambda_i^{n,s}| + \sum_{i=N+1}^{n} \lambda_i^n.\end{aligned}$$

We control the two terms separately. The first term $\sum_{i=1}^{N} |\lambda_i^n - \lambda_i^{n,s}|$ is bounded by $c \cdot N\epsilon$

from (4.20), and in order to bound $\sum_{i=N+1}^{n} \lambda_i^n$, note that

$$\begin{aligned}\big|tr(F^TWF) - tr(F_s^TWF_s)\big| &= \big|\Delta_y^2 \sum_{i,j} w_i f^2(x_i, y_j) - \Delta_y^2 \sum_{i,j} w_i f_s^2(x_i, y_j)\big| \\ &= \big|\Delta_y^2 \sum_{i,j} w_i \big(f(x_i, y_j) + f_s(x_i, y_j)\big)\big(f(x_i, y_j) - f_s(x_i, y_j)\big)\big| \\ &\leq \Delta_y^2 \sum_{i,j} (2c_f \cdot \epsilon) \\ &= 2(n\Delta y)^2 c_f \cdot \epsilon =: \tilde{c}\epsilon\end{aligned}$$

where c_f is the uniform bound for both $|f(x_i, y_j)|$ and $|f_s(x_i, y_j)|$ because $f(x_i, y_j)$ is smooth and $f_s(x_i, y_j)$ is the Chebyshev interpolation approximation on compact domains, and the constant $\tilde{c}$ depends on c_F and $\mu(\Omega_{in})$. The last by two step is due to the claim assumption and $f(x_i, y_j) - f_s(x_i, y_j) = \frac{1}{\Delta y}(F(i,j) - F_s(i,j))$. Because the trace function can be expressed as sum of eigenvalues, we have

$$\begin{aligned}\big|tr(F^TWF) - tr(F_s^TWF_s)\big| &= \big|\sum_{i=1}^{n} \lambda_i^n - \sum_{i=1}^{N} \lambda_i^{n,s}\big| \\ &= \big|\sum_{i=1}^{N} (\lambda_i^n - \lambda_i^{n,s}) + \sum_{i>N} \lambda_i^n\big| \\ &\geq -\sum_{i=1}^{N} |\lambda_i^n - \lambda_i^{n,s}| + \sum_{i>N} \lambda_i^n\end{aligned}$$

which implies

$$\sum_{i>N} \lambda_s^n \leq \big|tr(F^TWF) - tr(F_s^TWF_s)\big| + \sum_{i=1}^{N} |\lambda_i^n - \lambda_i^{n,s}| \leq \tilde{c}\epsilon + c \cdot N\epsilon.$$

Therefore,

$$|\phi(w) - \phi_s(w)| \leq (\tilde{c} + 2cN)\epsilon, \tag{4.21}$$

where $\tilde{c}$ and c are constants free of n and N. When $N > \tilde{c}$, we get for any $w \in \mathbb{R}^n$,

$$|\phi(w) - \phi_s(w)| \leq (2c+1)N \cdot \epsilon =: C \cdot N \cdot \epsilon.$$

which completes the proof. ■

4.4.2 Determine The Number of Interpolation Points N

Claim 4.4.2 suggests in order to get accurate approximation, we should make $N\epsilon$ small where ϵ is the error in Chebyshev polynomial approximation, and it depends on the number of interpolation points N and the smoothness of $f(x, y)$. We quantify now how ϵ depends on the two factors. Classical theory on Chebyshev interpolation error is well developed (e.g. see [25]): let f be a continuous function on $[-1, 1]$, h_n be its degree n polynomial interpolant at the Chebyshev points, $\varepsilon = \|f - h_n\|_\infty$, we have

- if f has a k^{th} derivative of bounded variation for some $k \geq 1$, then $\varepsilon = \mathcal{O}(N^{-k})$;
- if f is analytical in a neighborhood of $[-1, 1]$, then $\varepsilon = \mathcal{O}(\rho^N)$ for some $0 < \rho < 1$.

As the second step, we need to bound $\frac{1}{\Delta y}|F(i,j) - F_s(i,j)|$ to satisfy the condition in Claim 4.4.2. Note that $\frac{1}{\Delta y}F(i,j) = f(x_i, y_j)$ and

$$\frac{1}{\Delta y}F_s(i,j) = \sum_{p=1}^{N}\sum_{q=1}^{N} l_p(x_i) l_q(y_j) f(\tilde{x}_p, \tilde{y}_q)$$

where

$$l_p(x) = \prod_{\substack{k=1\\k\neq p}}^{N} \frac{x - \tilde{x}_k}{\tilde{x}_p - \tilde{x}_k}, \quad l_q(y) = \prod_{\substack{k=1\\k\neq q}}^{N} \frac{y - \tilde{y}_k}{\tilde{y}_q - \tilde{y}_k}$$

and $\{\tilde{x}_p\}_{i=1}^N$ and $\{\tilde{y}_q\}_{i=1}^N$ are interpolation points in Ω_{out} and Ω_{in} respectively. Let's first look at the one-dimensional case. If for $\forall x \in \Omega_{out}, \forall y \in \Omega_{in}$,

$$\Big|f(x,y) - \sum_{p=1}^{N} l_p(x) f(\tilde{x}_p, y)\Big| \le \epsilon_0, \quad \Big|f(x,y) - \sum_{q=1}^{N} l_q(y) f(x, \tilde{y}_q)\Big| \le \epsilon_0,$$

then we obtain

$$\begin{aligned}
&\frac{1}{\Delta y}|F(i,j) - F_s(i,j)| \\
&= \Big|f(x_i,y_j) - \sum_{p=1}^{N}\sum_{q=1}^{N} l_p(x_i) l_q(y_j) f(\tilde{x}_p, \tilde{y}_q)\Big| \\
&= \Big|f(x_i,y_j) - \sum_{p=1}^{N} l_p(x_i) f(\tilde{x}_p, y_j) + \sum_{p=1}^{N} l_p(x_i)\Big(f(\tilde{x}_p, y_j) - \sum_{q=1}^{N} l_q(y_j) f(\tilde{x}_p, \tilde{y}_q)\Big)\Big| \\
&\le \Big|f(x_i,y_j) - \sum_{p=1}^{N} l_p(x_i) f(\tilde{x}_p, y_j)\Big| + \sum_{p=1}^{N} |l_p(x_i)| \Big|f(\tilde{x}_p, y_j) - \sum_{q=1}^{N} l_q(y_j) f(\tilde{x}_p, \tilde{y}_q)\Big| \\
&\le \epsilon_0 + \epsilon_0 \sum_{p=1}^{N} |l_p(x_i)| \\
&\le \epsilon_0 + \Lambda_N * \epsilon_0 \qquad (4.22)
\end{aligned}$$

where Λ_N is called the Lebesgue constant and it is the opeator norm of Lagragian interpolation polynomial projection at Chebyshev nodes. It is known (see [77]) that

$$\frac{2}{\pi}\log(N) + a < \Lambda_N < \frac{2}{\pi}\log(N) + 1, \quad a = 0.9625....$$

- If $f(x,y)$ is k^{th} order continuously differentiable, in the one-dimensional case, we have

$$\begin{aligned}
\epsilon_0 = \mathcal{O}(N^{-k}) \quad &\overset{(4.22)}{\Longrightarrow} \quad \frac{1}{\Delta y}|F(i,j) - F_s(i,j)| = \mathcal{O}(N^{-k}\log(N)) \\
&\overset{(4.21)}{\Longrightarrow} \quad |\phi(w) - \phi_s(w)| = \mathcal{O}(N^{1-k}\log(N)).
\end{aligned}$$

We conclude that when $f(x,y)$ is at least 2nd order differentiable, $|\phi(w) - \phi_s(w)|$ will diminish as $n \to \infty$. The decay gets slower in multiple dimensions intuitively because if $\Omega_{in}, \Omega_{out} \subset R^d$, there are only $N^{1/d}$ interpolation points on each dimension. For the sake of clear presentation, let's still assume there are N interpolation points on each dimension. To derive an error bound, we define the Chebyshev interpolation operator $\mathcal{I}_N$ that maps a function $g \in \mathbb{C}([-1,1])$ to a degree-N polynomial:

$$\mathcal{I}_N g(x) = \sum_{p=1}^{N} l_p(x) g(\tilde{x}_p).$$

Since Λ_N is the operator norm, we have $\|\mathcal{I}_N g\|_\infty \le \Lambda_N \|g\|_\infty$. For $x, y \in \mathbb{R}^d$, we now define the double N-th order tensor product interpolation operator:

$$\mathcal{I}_N f(x,y) = \mathcal{I}_{N,x}^1 \times \cdots \times \mathcal{I}_{N,x}^d \times \mathcal{I}_{N,y}^1 \times \cdots \times \mathcal{I}_{N,y}^d f(x,y),$$

where $\mathcal{I}_{N,x}^i$ denotes the single interpolation operator on the i-th dimension in Ω_{out}, and $\mathcal{I}_{N,y}^j$ denotes the single interpolation operator on the j-th dimension in Ω_{in}.

$$\begin{aligned}
|f(x,y) - \mathcal{I}_N f(x,y)| &= |f(x,y) - \mathcal{I}_{N,x}^1 f(x,y) + \mathcal{I}_{N,x}^1 f(x,y) - \mathcal{I}_N f(x,y)| \\
&\le |f(x,y) - \mathcal{I}_{N,x}^1 f(x,y)| + |\mathcal{I}_{N,x}^1 f(x,y) - \mathcal{I}_N f(x,y)| \\
&\le |f(x,y) - \mathcal{I}_{N,x}^1 f(x,y)| + |\mathcal{I}_{N,x}^1 f(x,y) - \mathcal{I}_{N,x}^1 \mathcal{I}_{N,x}^2 f(x,y)| \\
&\quad + |\mathcal{I}_{N,x}^1 \mathcal{I}_{N,x}^2 f(x,y) - \mathcal{I}_N f(x,y)| \\
&\le |f(x,y) - \mathcal{I}_{N,x}^1 f(x,y)| + \Lambda_N |f(x,y) - \mathcal{I}_{N,x}^2 f(x,y)| \\
&\quad + \cdots + \Lambda_N^{2d-1} |f(x,y) - \mathcal{I}_{N,y}^d f(x,y)| \\
&\le \epsilon_0 (1 + \Lambda_N + \Lambda_N^2 + \cdots + \Lambda_N^{2d-1}) \\
&= \epsilon_0 \frac{\Lambda_N^{2d} - 1}{\Lambda_N - 1} \le \epsilon_0 \cdot \Lambda_N^{2d} \qquad (4.23)
\end{aligned}$$

for any $\Lambda_N > 2$. We implicitly assume here ϵ_0 is the uniform bound of the interpolation error on any single dimension in Ω_{in} and Ω_{out}. Now we go back to the case where there are N interpolation points in total, so there are $N^{1/d}$ interpolations on each side and if $f(x, y)$ is k-th order continuously differentiable,

$$\begin{aligned}\epsilon_0 = \mathcal{O}(N^{-k/d}) \quad &\overset{(4.23)}{\Longrightarrow} \quad \frac{1}{\Delta y}|F(i,j) - F_s(i,j)| = \mathcal{O}(N^{-k/d}\log^{2d}(N)) \\ &\overset{(4.21)}{\Longrightarrow} \quad |\phi(w) - \phi_s(w)| = \mathcal{O}(N^{1-k/d}\log^{2d}(N)).\end{aligned}$$

In order to guarantee of the convergence of $|\phi(w) - \phi_s(w)|$ to zero, $f(x, y)$ should be at least $(d+1) - th$ continuously differentiable.

- If $f(x, y)$ is analytical, then $\epsilon_0 = \mathcal{O}(\rho^{N^{1/d}})$ for some $0 < \rho < 1$ and

$$|\phi(w) - \phi_s(w)| = \mathcal{O}\big(N \log^{2d}(N)\rho^{N^{1/d}}\big).$$

which converges to zero for any dimension d.

In Section §4.2, we choose $N = c\log(n)$ to achieve the computational complexity $\mathcal{O}(n\log^s(n))$, but an important question is how to choose the constant c. One practical suggestion is to solve for problems with moderate sizes and get the exact solution (true minimum), and then adjust the constant c by doubling it until all the errors are below the preassigned threshold. As we will see in the numerical experiment, even though the zero convergence of optimality gap is not proved to be monotone, its fluctuation is small, and it goes to zero eventually.

4.5 Temporal and Two-Dimensional LIDAR Problem

In this section, we apply the sequential quadratic programming in §4.3 to solve a Bayesian inverse problem driven by partial differential equations. Specifically, our goal is to infer the initial condition of an advection-diffusion equation on a spatial and temporal domain, where the observable can be expressed as a truncated sum of integral equations so that all the convergence results in Chapter §3 and this chapter would apply.

4.5.1 Extend Convergence Results to Space-time Models

Because we are adding an extra time domain, theorems in Chapter 3 need to be extended for time-dependent measurements as in the gas pipeline system. In addition, we require that the measurements be taken at a fixed frequency for this extension.

Parameter-to-observable Map

Consider a compact domain V in $\mathbb{R}^P$ and a time interval $[0, T]$. Suppose the measurement without noise has the following form: for $x \in \Omega_{out}$

$$u(x,t) = \int_{\Omega_{in}} f(x,y,t) u_0(y) \, \mathrm{d}y. \tag{4.24}$$

In our example, $u(x,t)$ is the solution to partial differential equations describing a dynamical system, where $f(x,y,t)$ is derived from solving the equations. We discretize the integral equation (4.24), construct a matrix F from $f(x,y,t)$, and divide the domain Ω_{out} (Ω_{in} and $[0,T]$) into n_x (n_y and n_t) equally spaced intervals ($\Delta x = \mu(\Omega_{out})/n_x, \Delta y = \mu(\Omega_{out})/n_y, \Delta t = T/n_t$). Then, $\hat{u} = F\hat{u}_0 \in \mathbb{R}^{n_x n_t \times 1}, F \in \mathbb{R}^{n_x n_t \times n_y}$, where

$$\hat{u} = \big(u(x_1,t_1), u(x_1,t_2), ..., u(x_1,t_{n_t}), u(x_2,t_1), ..., u(x_2,t_{n_t}), ..., u(x_{n_x},t_1), u(x_{n_x},t_{n_t})\big)^T$$

$$F = \begin{pmatrix} f(x_1,y_1,t_1) & f(x_1,y_2,t_1) & \cdots & f(x_1,y_{n_y},t_1) \\ f(x_1,y_1,t_2) & f(x_1,y_2,t_2) & \cdots & f(x_1,y_{n_y},t_2) \\ \vdots & \vdots & & \vdots \\ f(x_1,y_1,t_{n_t}) & f(x_1,y_2,t_{n_t}) & \cdots & f(x_1,y_{n_y},t_{n_t}) \\ f(x_2,y_1,t_1) & f(x_2,y_2,t_1) & \cdots & f(x_2,y_{n_y},t_1) \\ \vdots & \vdots & & \vdots \\ f(x_{n_x},y_1,t_{n_t}) & f(x_{n_x},y_2,t_{n_t}) & \cdots & f(x_{n_x},y_{n_y},t_{n_t}) \end{pmatrix} \Delta y$$

and $\hat{u}_0 \in \mathbb{R}^{n_y}$ is a discretization of $u_0(x)$ with $\hat{u}_{0,j} = u_0(y_j)$ $(j = 1, 2, ..., n_y)$. To figure out the $(i, j)^{th}$ entry of F, let $i = (i_1 - 1) * n_t + i_2$ $(i_1 \in \{1, 2, ..., n_x\}, i_2 \in \{1, 2, ..., n_t\})$ and $j = 1, 2, ..., n_y$, we get

$$F(i, j) = f(x_{i_1}, y_j, t_{i_2})\Delta y.$$

If $\Omega_{in} = \Omega_{out}$, we use the same discretization, i.e. $x_j = y_j$ $(j = 1, 2..., n_x)$ and $n_x = n_y$.

Remark 1. *$f(x, y, t)$ in (4.24) is not always continuous as a solution to PDEs, for example, $f(x, y, t)$ in a one-wave system is a delta function $\delta(x-at, y)$ where a is the wave speed.*

Convexity of the Objective Function

In our Bayesian framework, the posterior covariance matrix is given by

$$\Gamma_{\text{post}} = \left(F^T W^{1/2} \Gamma_{noise}^{-1} W^{1/2} F + \Gamma_{\text{prior}}^{-1}\right)^{-1}.$$

where Γ_{noise} is the noise covariance matrix among measurements. We assume the measurement noise is only correlated in time, not in space. Under this assumption, Γ_{noise} is a block diagonal matrix and the number of blocks is equal to the number of discrete points on the spacial domain Ω_{out}.

Lemma 4.5.1. *$tr(\Gamma_{post})$, $\log\det(\Gamma_{post})$ and $\sigma_1(\Gamma_{post})$ are convex functions in the weight vector.*

<u>Proof:</u> We construct the matrix W from the weight vector $w = (w_0, w_1, .., w_{n_x})$ as follows:

$$W = \text{diag}\{w_1, w_1, ..., w_1, w_2, w_2, .., w_2, .., w_{n_x}, w_{n_x}, ..., w_{n_x}\} \in \mathbb{R}^{n_x n_t \times n_x n_t}.$$

Since Γ_{noise} is block diagonal, Γ_{noise}^{-1} is also block diagonal. $\Gamma_{\text{noise}}^{-1}$, F and W can be written as

$$\Gamma_{\text{noise}}^{-1} = \begin{pmatrix} P_1 & \cdots & \cdots & \cdots \\ \cdots & P_2 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & P_{n_x} \end{pmatrix} \qquad F = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_{n_x} \end{pmatrix} \qquad W = \begin{pmatrix} w_1 I_{n_t} & \cdots & \cdots & \cdots \\ \cdots & w_2 I_{n_t} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & w_{n_x} I_{n_t} \end{pmatrix}$$

where $P_k \equiv P \in \mathbb{R}^{n_t \times n_t}$ and $F_k \in \mathbb{R}^{n_t \times n_y}$. Therefore,

$$\Gamma_{\text{post}} = \Big(\sum_{k=1}^{n_x} w_k F_k^T P_k F_k + \Gamma_{\text{prior}}^{-1} \Big)^{-1}$$

The desired results follow because $tr(X^{-1})$, $\log\det(X^{-1})$ and $\sigma_1(X^{-1})$ are convex in X. ■

Extend the Convergence Theory

Note that

$$\Gamma_{\text{post}}^{-1} = \sum_{k=1}^{n_x} w_k F_k^T P F_k + \Gamma_{\text{prior}}^{-1}$$

and denote $f_{i,j,s} = f(x_i, y_j, t_s)$, then $F_k^T P F_k$ can be written as the following

$$\begin{pmatrix} \sum_{s_1,s_2=1}^{n_t} f_{k,1,s_1} P_{s_1,s_2} f_{k,1,s_2} & \cdots & \sum_{s_1,s_2=1}^{n_t} f_{k,1,s_1} P_{s_1,s_2} f_{k,n_y,s_2} \\ \vdots & \ddots & \vdots \\ \sum_{s_1,s_2=1}^{n_t} f_{k,n_y,s_1} P_{s_1,s_2} f_{k,1,s_2} & \cdots & \sum_{s_1,s_2=1}^{n_t} f_{k,n_y,s_1} P_{s_1,s_2} f_{k,n_y,s_2} \end{pmatrix} (\Delta y)^2$$

Therefore, the $(i,j)^{th}$ entry in $\Gamma_{\text{post}}^{-1}$ is

$$\Gamma_{\text{post}}^{-1}(i,j) = \Delta y \sum_{k=1}^{n_x} \sum_{s_1=1}^{n_t} \sum_{s_2=1}^{n_t} w_k f(x_k, t_{s_1}, y_i) P_{s_1,s_2} f(x_k, t_{s_2}, y_j) \Delta x.$$

If measurements are collected every few seconds or minutes within a time range, i.e. n_t is a fixed integer, then for any precision matrix P, we are in the same setting of Chapter 3, and all the convergence proofs can be extended trivially.

4.5.2 2D Advection-diffusion Equation

The advection-diffusion equation is a combination of diffusion and advection equations, and we will first look at the solution to the diffusion equation (or heat equation), which lays the foundation to solving the advection-diffusion equation.

Analytical Solution to 2D Heat Equation

Consider a heat equation on a two-dimensional domain $[-1,1]\times[-1,1]$, with homogeneous Dirichlet boundary conditions:

$$\begin{cases} u_t = \mu\nabla u = \mu\big(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\big) \\ u(x,y,t) = 0, \text{ for } (x,y) \text{ on the boundary.} \\ u(x,y,0) = u_0(x,y), \text{ initial condition.} \end{cases} \tag{4.25}$$

Using separation of variables, we let $u(x,y,t) = T(t)X(x,y)$ and get

$$\frac{T_t}{T} = \mu\frac{\nabla X}{X} \equiv constant.$$

The solution to (4.25) can be written as

$$u(x,y,t) = \sum_{k=1}^{\infty} c_k e^{\mu\lambda_k t} h_k$$

where (λ_k, h_k) are the eigenvalue and eigenvector respectively of the following problem:

$$\begin{cases} \nabla X = \frac{\partial X^2}{\partial^2 x} + \frac{\partial X^2}{\partial^2 y} = \lambda X \\ X(x,y) = 0 \text{ for } (x,y) \text{ on the boundary.} \end{cases}$$

Since $\Big(k^2\pi^2/4, \sin\big(\frac{k\pi x}{2}\big)\Big)$ and $(k^2\pi^2/4, \cos\big(\frac{k\pi x}{2}\big))$ are eigen-pairs in one dimension, we can apply tensor product to obtain eigen-pairs in two dimensions, and further get the solution:

$$\begin{aligned} u(x,y,t) = \sum_{k_1\geq 0}\sum_{k_2\geq 0} & \exp\{-\mu t(k_1^2+k_2^2)\pi^2/4\}\Big(A_{k_1,k_2}\sin\big(\frac{k_1\pi x}{2}\big)\sin\big(\frac{k_2\pi y}{2}\big) \\ & + B_{k_1,k_2}\sin\big(\frac{k_1\pi x}{2}\big)\cos\big(\frac{k_2\pi y}{2}\big) + C_{k_1,k_2}\cos\big(\frac{k_1\pi x}{2}\big)\sin\big(\frac{k_2\pi y}{2}\big) \\ & + D_{k_1,k_2}\cos\big(\frac{k_1\pi x}{2}\big)\cos\big(\frac{k_2\pi y}{2}\big)\Big). \end{aligned} \tag{4.26}$$

In combination with boundary condition and initial condition, we can calculate the coefficients which are given below:

$$\begin{aligned} A_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x,y)\sin\big(\frac{k_1\pi x}{2}\big)\sin\big(\frac{k_2\pi y}{2}\big)dxdy, \quad k_1 \textit{ even and } k_2 \textit{ even;} \\ B_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x,y)\sin\big(\frac{k_1\pi x}{2}\big)\cos\big(\frac{k_2\pi y}{2}\big)dxdy, \quad k_1 \textit{ even and } k_2 \textit{ odd;} \\ C_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x,y)\cos\big(\frac{k_1\pi x}{2}\big)\sin\big(\frac{k_2\pi y}{2}\big)dxdy, \quad k_1 \textit{ odd and } k_2 \textit{ even;} \\ D_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x,y)\cos\big(\frac{k_1\pi x}{2}\big)\cos\big(\frac{k_2\pi y}{2}\big)dxdy, \quad k_1 \textit{ odd and } k_2 \textit{ odd.} \end{aligned}$$

Because the uniqueness of solution to heat equation on bounded domains is already established in the classical theory of partial differential equations (see [78, §2.3]), $u(x,y,t)$ in (4.26) is indeed the solution to (4.25).

From Heat Equation to Advection Diffusion Equation

The two-dimensional advection-diffusion equation we consider here is

$$\frac{\partial u}{\partial t} + c\big(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\big) = \mu\big(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\big), \quad -1 < x,y < 1, \quad t \in [0,T].$$

where c is the velocity constant and μ is the diffusivity. After a change of variables $u(x, y, t) = v(x, y, t)e^{\alpha(x+y)+\beta t}$, we get

$$\frac{\partial v}{\partial t} + (c - 2\mu\alpha)\big(\frac{\partial v}{\partial x} + \frac{\partial v}{\partial x}\big) + (\beta + 2c\alpha - 2\mu\alpha^2)v = \mu\big(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial x^2}\big).$$

Set the coefficients of v and $v'_x + v'_y$ to zero, we have the following

$$\begin{cases} c - 2\mu\alpha = 0 \\ \beta + 2c\alpha - 2\mu\alpha^2 = 0 \end{cases} \Rightarrow \begin{cases} \alpha = c/2\mu \\ \beta = -c^2/2\mu. \end{cases}$$

We can see $v(x, y, t)$ satisfies a heat equation, and its relation with $u(x, y, t)$ is given by

$$u(x, y, t) = \exp\{-c^2 t/2\mu + c(x + y)/2\mu\}v(x, y, t).$$

The solution $u(x, y, t)$ given $v(x, y, t)$ derived from the previous subsection is

$$\begin{aligned} u(x, y, t) = &\exp\{-c^2 t/2\mu + c(x + y)/2\mu\} \sum_{k_1 \geq 0} \sum_{k_2 \geq 0} \exp\{-\mu t(k_1^2 + k_2^2)\pi^2/4\} \\ &\Big(A_{k_1,k_2} \sin\big(\frac{k_1\pi x}{2}\big) \sin\big(\frac{k_2\pi y}{2}\big) + B_{k_1,k_2} \sin\big(\frac{k_1\pi x}{2}\big) \cos\big(\frac{k_2\pi y}{2}\big) \\ &+ C_{k_1,k_2} \cos\big(\frac{k_1\pi x}{2}\big) \sin\big(\frac{k_2\pi y}{2}\big) + D_{k_1,k_2} \cos\big(\frac{k_1\pi x}{2}\big) \cos\big(\frac{k_2\pi y}{2}\big)\Big). \end{aligned}$$

Again in combination with boundary condition and initial condition, the coefficients are:

$$\begin{aligned} A_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x, y) \sin\big(\frac{k_1\pi x}{2}\big) \sin\big(\frac{k_2\pi y}{2}\big) \exp\{-\frac{c}{2\mu}(x + y)\}dxdy, \quad k_1 \textit{ even and } k_2 \textit{ even}; \\ B_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x, y) \sin\big(\frac{k_1\pi x}{2}\big) \cos\big(\frac{k_2\pi y}{2}\big) \exp\{-\frac{c}{2\mu}(x + y)\}dxdy, \quad k_1 \textit{ even and } k_2 \textit{ odd}; \\ C_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x, y) \cos\big(\frac{k_1\pi x}{2}\big) \sin\big(\frac{k_2\pi y}{2}\big) \exp\{-\frac{c}{2\mu}(x + y)\}dxdy, \quad k_1 \textit{ odd and } k_2 \textit{ even}; \\ D_{k_1,k_2} &= \int_{-1}^{1}\int_{-1}^{1} u_0(x, y) \cos\big(\frac{k_1\pi x}{2}\big) \cos\big(\frac{k_2\pi y}{2}\big) \exp\{-\frac{c}{2\mu}(x + y)\}dxdy, \quad k_1 \textit{ odd and } k_2 \textit{ odd.} \end{aligned}$$

In other words, $u(x,y,t)$ is an integral equation of $u_0(x,y)$:

$$u(x,y,t) = \iint_{\Omega_{in}} f(\tilde{x},\tilde{y},x,y,t)u_0(\tilde{x},\tilde{y})d\tilde{x}d\tilde{y}$$

where the function $f(\tilde{x},\tilde{y},x,y,t)$ is an infinite sum of Fourier series.

4.5.3 2D Advection-diffusion Equation with External Source

In the previous subsection, we have considered a stationary process meaning that eventually, the state becomes zero everywhere because of the homogeneous boundary conditions. Now we extend it to a non-stationary process by adding an external force to the equation. Again we work on the heat equation and then generalize to advection-diffusion equation.

2D Heat Equation with External Source

Consider the heat equation on a two-dimensional domain $[-1,1]\times[-1,1]$ with homogeneous Dirichlet boundary conditions and an external force $f(x,y,t)$:

$$u_t - \mu\nabla u = u_t - \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = f(x,y,t)$$

and $u(x,y,t) = 0$ for (x,y) on the boundary. The initial condition is $u(x,y,0) = u_0(x,y)$. Using a variant of separation of variables, we assume the solution $u(x,y,t)$ has the following form

$$u(x,y,t) = \sum_n T_n(t)X_n(x,y)$$

and then apply the tensor product of one-dimensional Fourier basis as before, we get

$$\begin{aligned} u(x,y,t) = \sum_{k_1\geq 0}\sum_{k_2\geq 0}\Big\{&T^{(1)}_{k_1,k_2}(t)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}) + T^{(2)}_{k_1,k_2}(t)\sin(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2}) \\ &+ T^{(3)}_{k_1,k_2}(t)\cos(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}) + T^{(4)}_{k_1,k_2}(t)\cos(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2})\Big\}. \end{aligned}$$

We treat each of the above four terms separately, and will use the Fourier basis $\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2})$ as an example. Results for the other three terms can be derived similarly. Let

$$u^{(1)}(x,y,t) = \sum_{k_1\geq 0}\sum_{k_2\geq 0} T^{(1)}_{k_1,k_2}(t)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}),$$

and then

$$\frac{\partial u^{(1)}}{\partial t} - \mu(\frac{\partial^2 u^{(1)}}{\partial x^2} + \frac{\partial^2 u^{(1)}}{\partial y^2}) = \sum_{k_1\geq 0}\sum_{k_2\geq 0}\Big(\frac{\partial T^{(1)}_{k_1,k_2}(t)}{\partial t} + \mu\frac{k_1^2+k_2^2}{4}\pi^2 T^{(1)}_{k_1,k_2}(t)\Big)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}).$$

We solve the following equation:

$$\begin{cases} \frac{\partial T^{(1)}_{k_1,k_2}(t)}{\partial t} + \mu\frac{k_1^2+k_2^2}{4}\pi^2 T^{(1)}_{k_1,k_2}(t) = f_{k_1,k_2}(t) \\ T^{(1)}_{k_1,k_2}(0) = c_{k_1,k_2} \end{cases} \tag{4.27}$$

where $f_{k_1,k_2}(t)$ and c_{k_1,k_2} are the Fourier coefficients with respect to the basis $\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2})$, for the external force $f(x,y,t)$ and the initial condition $u_0(x,y)$ respectively:

$$\begin{aligned} f_{k_1,k_2}(t) &= \iint_{[-1,1]\times[-1,1]} f(x,y,t)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2})\,\mathrm{d}x\mathrm{d}y, \\ c_{k_1,k_2} &= \iint_{[-1,1]\times[-1,1]} u_0(x,y)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2})\,\mathrm{d}x\mathrm{d}y \end{aligned}$$

and get the solution to (4.27)

$$T^{(1)}_{k_1,k_2}(t) = \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2 t\}c_{k_1,k_2} + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2(t-s)\}f_{k_1,k_2}(s)\,\mathrm{d}s.$$

After working out the other three terms, we combine them together and get:

$$
\begin{aligned}
u(t,x,y) = \sum_{k_1\geq 0}\sum_{k_2\geq 0} & A_{k_1,k_2}(t)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}) + B_{k_1,k_2}(t)\sin(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2}) \\
& +C_{k_1,k_2}(t)\cos(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}) + D_{k_1,k_2}(t)\cos(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2}) \qquad (4.28)
\end{aligned}
$$

where

$$
\begin{aligned}
A_{k_1,k_2}(t) &= \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2 t\}\iint_{\Omega_{in}} u_0(x,y)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2})\,\mathrm{d}x\mathrm{d}y \\
&\quad + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2(t-s)\}f^{(1)}_{k_1,k_2}(s)\,\mathrm{d}s \\
B_{k_1,k_2}(t) &= \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2 t\}\iint_{\Omega_{in}} u_0(x,y)\sin(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2})\,\mathrm{d}x\mathrm{d}y \\
&\quad + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2(t-s)\}f^{(2)}_{k_1,k_2}(s)\,\mathrm{d}s \\
C_{k_1,k_2}(t) &= \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2 t\}\iint_{\Omega_{in}} u_0(x,y)\cos(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2})\,\mathrm{d}x\mathrm{d}y \\
&\quad + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2(t-s)\}f^{(3)}_{k_1,k_2}(s)\,\mathrm{d}s \\
D_{k_1,k_2}(t) &= \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2 t\}\iint_{\Omega_{in}} u_0(x,y)\cos(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2})\,\mathrm{d}x\mathrm{d}y \\
&\quad + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{4}\pi^2(t-s)\}f^{(4)}_{k_1,k_2}(s)\,\mathrm{d}s.
\end{aligned}
$$

Again from boundary conditions, A_{k_1,k_2} is for k_1 even and k_2 even, B_{k_1,k_2} is for k_1 even and k_2 odd, C_{k_1,k_2} is for k_1 odd and k_2 even, D_{k_1,k_2} is for k_1 odd and k_2 odd.

Advection-diffusion Equation with External Source

The two-dimensional advection-diffusion equation with external source is

$$
\frac{\partial u}{\partial t} + c_1\frac{\partial u}{\partial x} + c_2\frac{\partial u}{\partial y} - \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = f(x,y,t), \quad -1 < x, y < 1, \quad t \in [0,T]. \qquad (4.29)
$$

where $c = (c_1, c_2)$ is the velocity constant and μ is the diffusivity. After a change of variables $u(x, y, t) = v(x, y, t)e^{\alpha x+\beta y+\gamma t}$, we get

$$\frac{\partial v}{\partial t} + (c_1 - 2\mu\alpha)\frac{\partial v}{\partial x} + (c_2 - 2\mu\beta)\frac{\partial v}{\partial y} + (\gamma + c_1\alpha + c_2\beta - \mu\alpha^2 - \mu\beta^2)v - \mu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial x^2})$$
$$= f(x, y, t)\exp\{-\alpha x - \beta y - \gamma t\}.$$

Again we set some coefficients to zero and get

$$\begin{cases} c_1 - 2\mu\alpha = 0 \\ c_2 - 2\mu\beta = 0 \\ \gamma + c_1\alpha + c_2\beta - \mu\alpha^2 - \mu\beta^2 = 0 \end{cases} \Rightarrow \begin{cases} \alpha = c_1/2\mu \\ \beta = c_2/2\mu \\ \gamma = -(c_1^2 + c_2^2)/4\mu. \end{cases}$$

Note $v(x, y, t)$ satisfies the heat equation with homogeneous Dirichlet conditions:

$$\begin{cases} v_t - \mu(v_{xx} + v_{yy}) = \tilde{f}(x, y, t) \\ v(x, y, t) = 0, \text{ for } (x, y) \text{ on the boundary.} \\ v_0(x, y) = \exp\{-c_1x/2\mu - c_2y/2\mu\}u_0(x, y), \end{cases}$$

where $\tilde{f}(x, y, t) = \exp\{(c_1^2 + c_2^2)t/4\mu - c_1x/2\mu - c_2y/2\mu\}f(x, y, t)$. Its relation to $u(x, y, t)$ is given by

$$u(x, y, t) = \exp\{-(c_1^2 + c_2^2)t/4\mu + c_1x/2\mu + c_2y/2\mu\}v(x, y, t).$$

Based on the result on heat equation, the solution $u(x, y, t)$ is:

$$\begin{aligned} u(x, y, t) = &\exp\{-(c_1^2 + c_2^2)t/4\mu + c_1x/2\mu + c_2y/2\mu\} \sum_{k_1\geq 0}\sum_{k_2\geq 0} \\ &\Big\{A_{k_1,k_2}(t)\sin(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}) + B_{k_1,k_2}(t)\sin(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2}) \\ &+ C_{k_1,k_2}(t)\cos(\frac{k_1\pi x}{2})\sin(\frac{k_2\pi y}{2}) + D_{k_1,k_2}(t)\cos(\frac{k_1\pi x}{2})\cos(\frac{k_2\pi y}{2})\Big\}. \end{aligned} \tag{4.30}$$

where

$$\begin{aligned}
A_{k_1,k_2}(t) &= \phi^{(1)}_{k_1,k_2} + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{2}\pi^2(t-s)\}\tilde{f}^{(1)}_{k_1,k_2}(s)\,\mathrm{d}s, \quad \text{for } k_1 \text{ even, } k_2 \text{ even;}\\
B_{k_1,k_2}(t) &= \phi^{(2)}_{k_1,k_2} + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{2}\pi^2(t-s)\}\tilde{f}^{(2)}_{k_1,k_2}(s)\,\mathrm{d}s, \quad \text{for } k_1 \text{ even, } k_2 \text{ odd;}\\
C_{k_1,k_2}(t) &= \phi^{(3)}_{k_1,k_2} + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{2}\pi^2(t-s)\}\tilde{f}^{(3)}_{k_1,k_2}(s)\,\mathrm{d}s, \quad \text{for } k_1 \text{ odd, } k_2 \text{ even;}\\
D_{k_1,k_2}(t) &= \phi^{(4)}_{k_1,k_2} + \int_0^t \exp\{-\mu\frac{k_1^2+k_2^2}{2}\pi^2(t-s)\}\tilde{f}^{(4)}_{k_1,k_2}(s)\,\mathrm{d}s, \quad \text{for } k_1 \text{ odd, } k_2 \text{ odd;}
\end{aligned}$$

$\{\phi^{(i)}_{k_1,k_2}\}$ is related to the Fourier coefficients of the initial condition v_0 (or equivalently, u_0) as in (4.28). We see that the solution $u(x,y,t)$ is an additive sum of two components: one is from the initial condition $u_0(x,y)$, the other is from the external source $f(x,y,t)$, and the two sources act independently on the solution. Therefore, the external source does not play a role in the selection of sensor locations, if we use a Bayesian framework of Gaussian distributions to infer the initial condition from time-space measurements.

4.5.4 Numerical Results

In this subsection, we provide numerical results on selecting the optimal sensing directions to estimate the initial condition of a two-dimensional advection-diffusion equation. Here is the problem description: suppose a lidar is sitting at the origin of a unit circle (Ω_{out}), and it collects data $u(x,y,t)$ by sending out laser beams and detecting reflections; we need to determine the optimal directions for releasing the beams to collect data long those directions. Our parameter is the initial condition $u_0(x,y)$ of the advection-diffusion equations (4.29), and the parameter-to-observable mapping is from the solution to (4.29), which is an integral equation

$$u(x,y,t) = \mathcal{F}(u_0) = \iint_{[-1,1]\times[-1,1]} \mathbb{F}(x,y,t)u_0(x,y)\,dxdy$$

where $\mathbb{F}$ is directly from the solution in (4.30). For discretizations, we divide the angle of 2π into n_d parts so that the circle has n_d sectors with the same area, and each beam goes across the center of each sector. We also discretize the radius into nr parts with equal length. We attach a weight variable to each sector (or radius), and points on the same radius have the same weight. The goal is to select a proportion of sectors to measure data along the radius, and best infer u_0 defined on the slightly larger square domain $[-1,1] \times [-1,1]$ (Ω_{in}), which is discretized by regular grid of size $n_x \times n_x$.

The constants we choose in the equations (4.29) are $c_1 = 0.1, c_2 = 0, \mu = 1.0, T = 1, n_t = 5, r = 0.2$, and the noise ratio is 0.01. Here is a reminder on the meaning of these constants: c_1, c_2 are the velocities along the x and y axis respectively, μ is the diffusivity constant, n_t is a fixed integer denoting the number of measurements in time, and the noise ratio is $\sigma^2_{noise}/\sigma^2_{prior}$. The covariance matrix in time is set to be identity at the moment. For the results below, $nd = nr = n_x = 30$, and the velocity (c_1, c_2) is pointing from the origin to its right (the advection term can be thought of as air movement or wind when studying the concentration of a substance), so the placement is symmetric to the x axis (see Fig. 4.1).

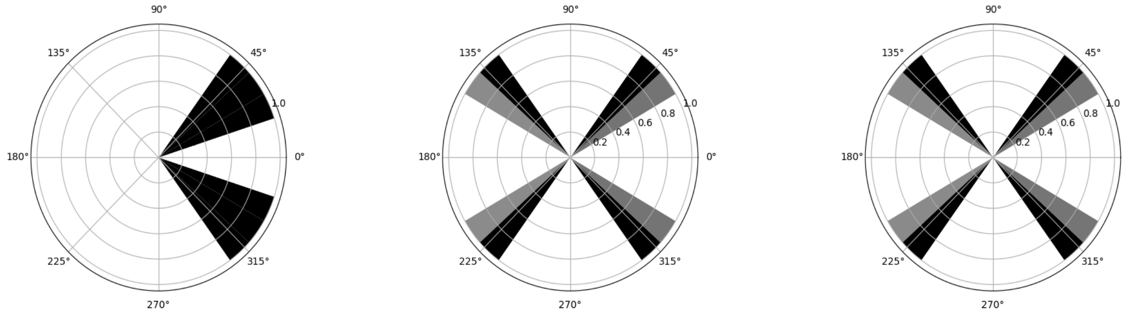


Fig.4.1: Relaxed solution for $p = 1, 2, 3$ respectively

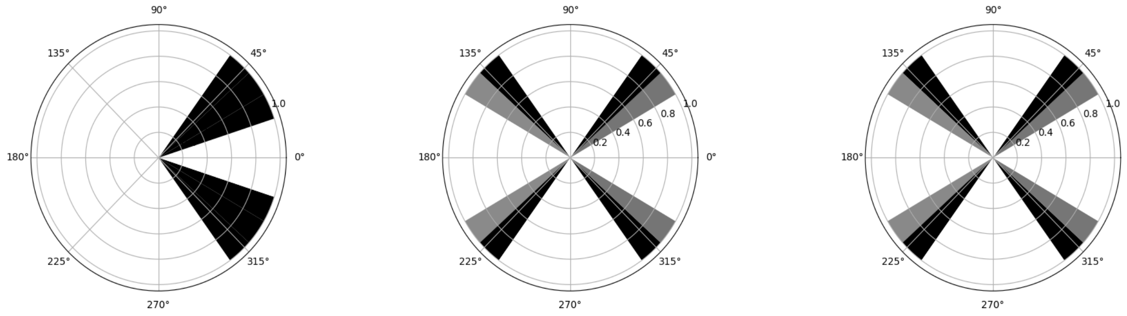


Fig.4.2: SQP solution for $p = 1, 2, 3$ respectively

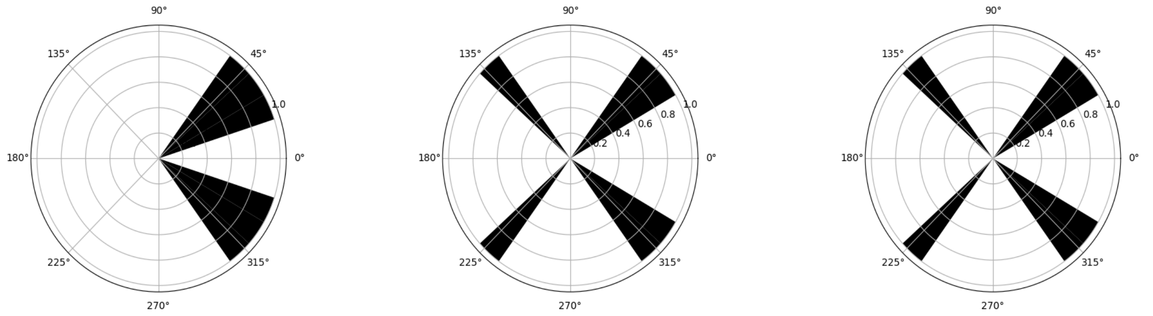
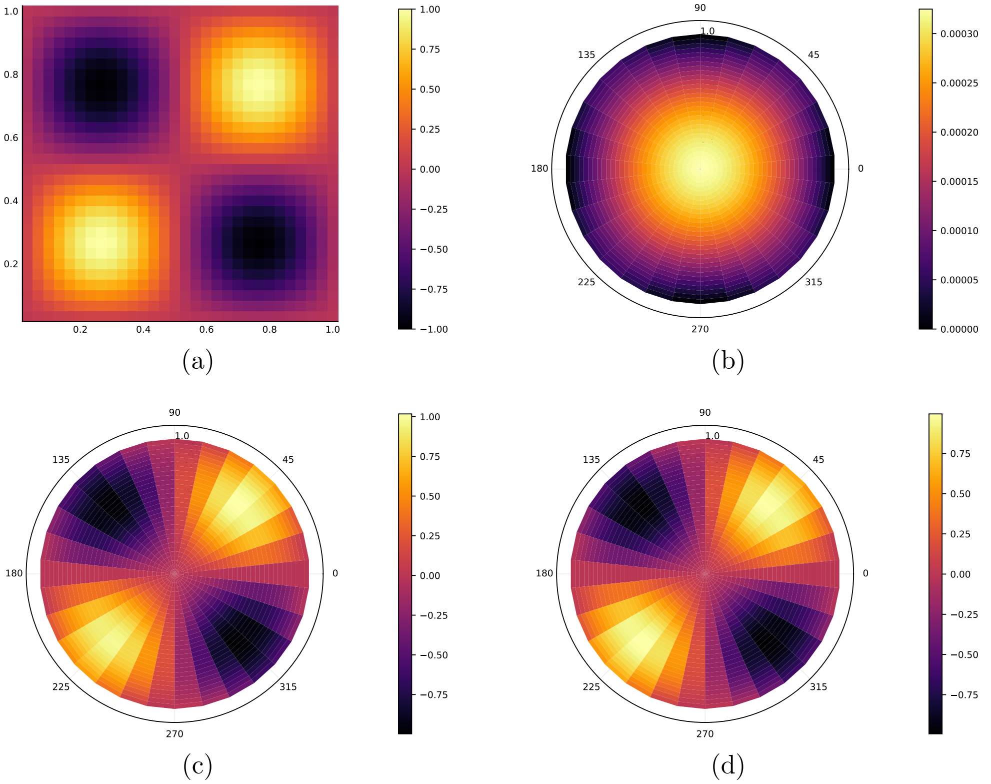


Fig.4.3: Sum-up Rounding for $p = 1, 2, 3$ respectively (based on relaxed solutions)

Because the solution to advection-diffusion equation is an integral of an infinite sum, we truncate and take the dominating terms where $k_1, k_2 \leq p$ in (4.30), and we determine the hyper parameter p from a sanity check, where we try to recover the initial state $\sin(\pi x)\sin(\pi y)$ by looking at the truncated solution at $t = 0$. We observe that when $p = 3$, the values do not change further, which suggests $p = 3$ is sufficient. Note this choice of p is subject to the choice of c_1, c_2 and μ in the PDE, especially when μ is small, a larger value of p is required.

Next we examine the performance of SQP by looking at the computation time in comparison with the *Ipopt* package in Julia, its optimality gap in the objective, and its maximum KKT violation defined in (4.10) (without aprroximation) to measure the closeness to the true minimal point. We use different numbers of interpolation points $c \cdot \log(n)$ by choosing the constant $c = 1, 2, 4, 8$, and let $nd = nr = n_x$, that is, the number of angles equals the number of discretization points on each radius. When $nr = 30$, it takes *Ipopt* about 1.5 hours to find the solution, while SQP only needs a few minutes to get a sufficiently good approximation. When $c = 8, nr = 30$, the SQP solution actually gives a lower objective value than the "true" minimum possibly due to the tolerance level (10^{-6}) specified in *Ipopt*.



(a) (b)

(c) (d)

Fig.4.4: (a) Input function: $z = \sin(\pi x)(\pi y)$ on $[-1,1]\times[-1,1]$; (b) Recover the initial state using dominant terms with $k_1, k_2 \leq 1$; (c) Recover the initial state using dominant terms with $k_1, k_2 \leq 2$; (d) Recover the initial state using dominant terms with $k_1, k_2 \leq 3$.

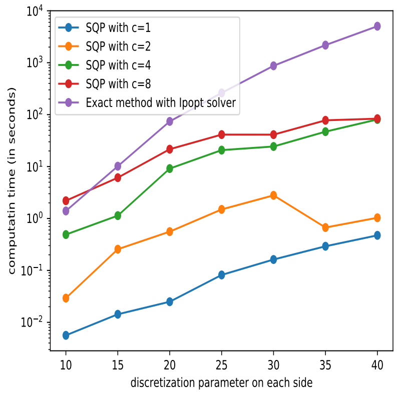


Fig.4.5: Computation time

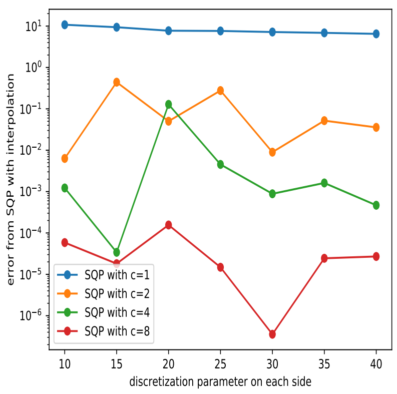


Fig.4.6: Error in the objective

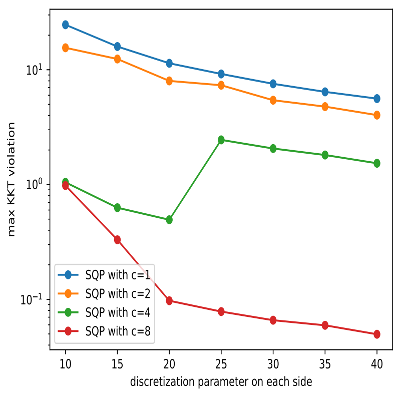


Fig.4.7: KKT violation

We would also like to examine the effect of c_1, c_2 and μ on the optimal sensing directions, and we conduct a few more experiments. The following gives the exact relaxed solution for varying values of c_1 and μ, but fixed $c_2 = 0$ and $p = 3$. Again, $nd = nr = 30$.

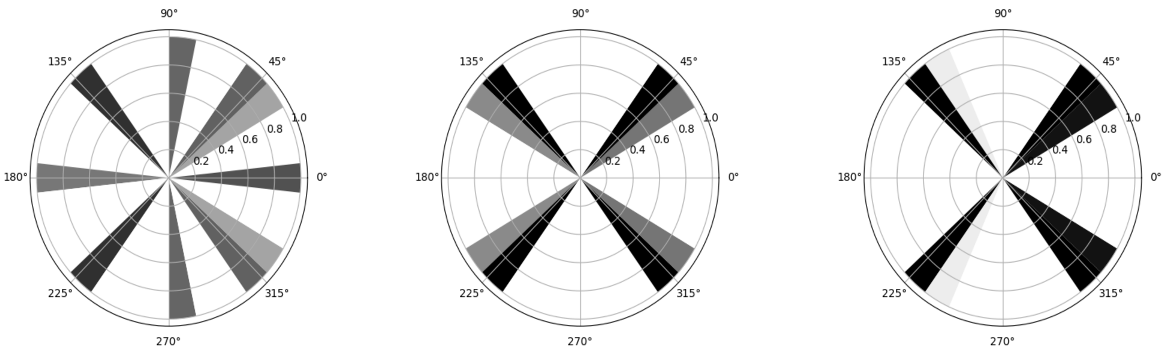


Fig.4.8: Dependence of sensing direction on c_1 and μ when wind blows →. From left to right: (1) $c_1 = 0.1, \mu = 0.1$; (2) $c_1 = 0.1, \mu = 1.0$; (3) $c_1 = 1.0, \mu = 1.0$.

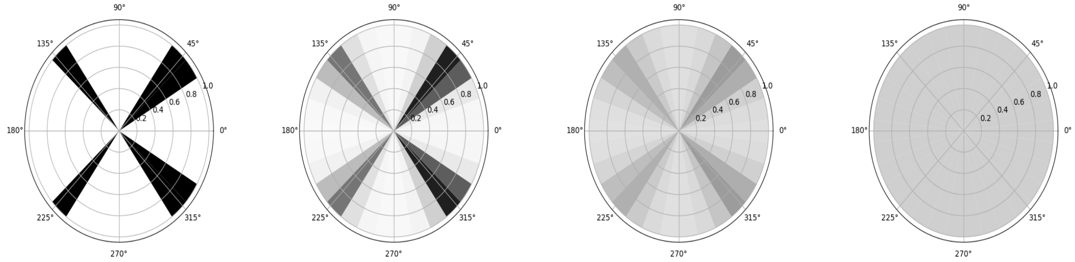


Fig.4.9: Dependence of sensing direction on μ when $c_1 = 0.1$ and wind blows →. From left to right: $\mu = 5.0, 7.0, 8.0, 10.0$.

According to Fig. 4.8 and Fig. 4.9, we find that

- when the wind moves faster, more sensing directions are chosen towards the wind;
- when it is less diffusive (small values of μ), the sensing directions spread out more;
- when the diffusivity μ is large, the relaxed sensing directions gets blurred.

Since SQP is much faster than the exact method, we can run larger size problems ($nd = nr = 80$) and change the wind direction form → to ↗. The relaxed sensing directions are given below.

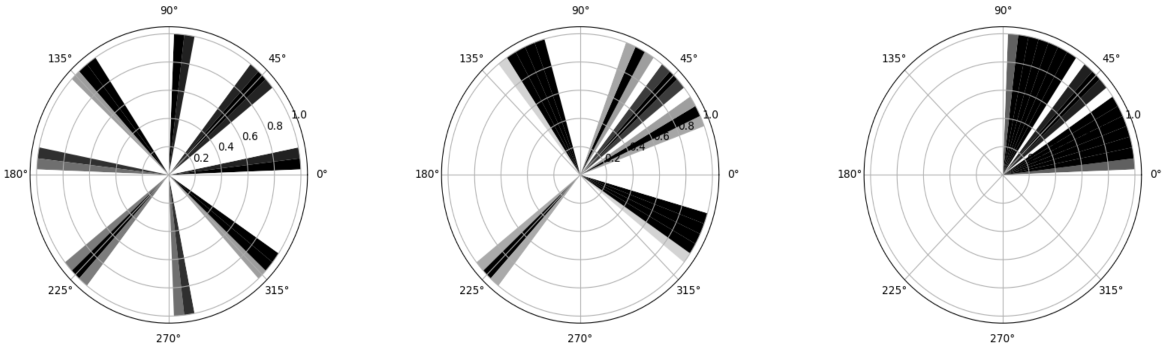


Fig.4.10: Sensing direction for increasing wind speed with $\mu = 0.1$, wind direction$\nearrow$. From left to right: (1) $c_1 = c_2 = 0.1$; (2) $c_1 = c_2 = 0.5$; (3) $c_1 = c_2 = 1.0$.

From the solution to the advection-diffusion equation, for a larger value of μ, p imposes less effect on the sensing directions. But when μ is small, such as 0.1, the design is likely to depend on p, and adding p makes the design more "diffusive", although the configuration is roughly the same, see Figure 4.10 and Figure 4.11.

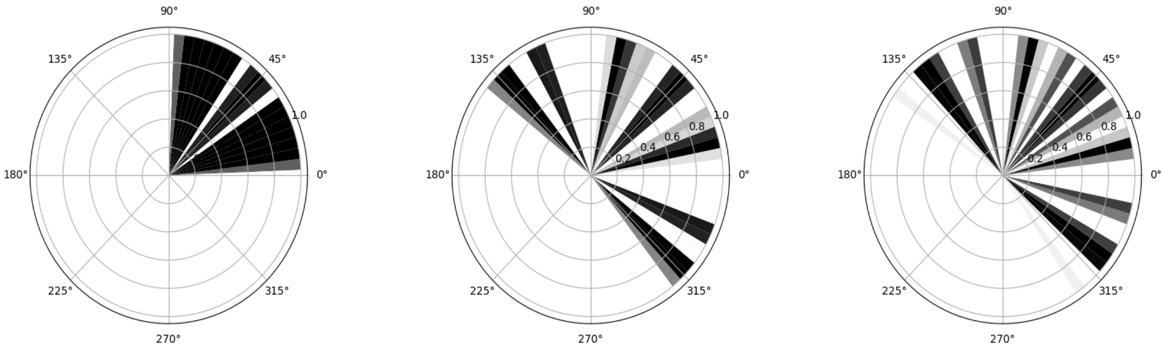


Fig.4.11: Sensing direction for increasing p with $c_1 = c_2 = 1.0$ and $\mu = 0.1$, wind direction$\nearrow$. From left to right: $p = 3, 5, 10$.

5 Future Work

The work discussed in the thesis provides the initial results on the zero gap convergence of optimal sensor placement, and there are many directions for future research. For example, an immediate extension is to consider a general covariance function in the prior; the integral equation seems restrictive in describing the relationship between the parameter and observation. We give the following five prospective research directions to pursue.

(1) Build the connection between the limit of discretized problem and the continuous problem on function spaces. The parameter in our formulation is a vector where each element is associated with evaluation at a mesh point, and we increase the dimensions to infinity. Whether the vector converges to a function-space-valued limit as the finite-dimensional computation is refined deserves further study. There are known results in both Gaussian and non-Gaussian settings (see discussions in [79], [18, §2.5]), but not for the particular formulation incorporating weight matrix for sensor placement. It is also of theoretical interest to explore if the relaxed weight vector w_{rel} converges to a continuous density with respect to Lebesgue measure on Ω_{out}, although this is not required in our theorems.

(2) Generalize parameter-to-observable mapping: integral operator → linear operator → nonlinear operator. When the parameter is from a Hilbert space with the inner product defined as an integral over a compact domain, it is known by Riesz representation theorem any bounded linear operator is an inner product with a representative element, and thus an integral operator. The theory of singular value decomposition for compact operators on Hilbert space is also well established (see [80] and [81, §3.5]), so a low-rank approximation by integral operators is possible by truncation. However for

nonlinear operators, it remains difficult even though linearization methods have been studied extensively (see [82, §4], [83, 84]).

(3) *Sum-up rounding* can potentially be used in any continuously-indexed sampling problems and approximate the density of optimal sampling locations. We can apply the *sum-up rounding* idea on other formulations like Gaussian process/Kriging. A spatio-temporal field estimation based on a kriged Kalman filter was published (see [9]) lately, where the measurement consists of a stationary component capturing the non-dynamic spatial effects, a non-stationary component modeling the physical properties of environmental fields (represented by state space models), and an uncorrelated measurement error. Similarly, the authors discretize the entire service area into small units, construct a weight vector for each unit, formulate a convex optimization and solve the relaxation. In the paper, the thresholding and weight-based multinomial rounding strategies are discussed, but *sum-up rounding* seems a better candidate in this setting.

(4) Consider general domains and general prior information. In the current work, both input and output domains are rectangles, and we have mentioned extensions to domains that can be well approximated by rectangles. However, in many practical problems, domains do not possess regular shapes, such as autonomous vehicles and robots. Reparametrization or transformation may be required to deal with complicated domains, and thus a theory on general domains is one direction to pursue. One advantage of Bayesian formulation is to incorporate prior belief on the parameter, and a common choice is Gaussian prior partially because a lot of theoretical results are available in literature, but prior generalization is considered as a significant challenge. Even in the Gaussian setting, it is unclear how to choose and interpret the covariance function.

(5) Extend *sum-up rounding* strategy for multiple states. The current strategy is to construct a binary vector, and only two states are allowed. There is a general version of the strategy that applies to multiple states (see [46, Theorem 5]). If there are different types

of sensors, for example, self-driving vehicles have lidar sensors, radar sensors and camera sensors, the multi-state *sum-up rounding* would be useful in allocating different sensors in different locations under statistical optimal conditions. It is of interest to prove similar gap convergence given that the general strategy has a wider range of applications.

A Likelihood for Sensor Data

The actual measurements d can be thought of in several ways, and the data likelihood

$$\pi_{\text{like}}(d|u_0, w) \propto \exp\Big\{ -\frac{1}{2}(d - Fu_0)^T W^{1/2} \Gamma_{\text{noise}}^{-1} W^{1/2} (d - Fu_0) \Big\}.$$

shall all agree in the case when the weight matrix in $\mathbb{R}^{n\times n}$ is diagonal with entries either 0 or 1, and Γ_{noise} is diagonal. Suppose the full measurement is $u \in \mathbb{R}^n$, and we observe n_0 of them ($n_0 \leq n$). The most general consideration is that $\tilde{d} \in \mathbb{R}^{n_0}$, the weight matrix $\tilde{W} \in \mathbb{R}^{n_0 \times n}$, and n_0 out of its n columns form an identity matrix I_{n_0}. The restricted covariance matrix is $\tilde{W}\Gamma_{\text{noise}}\tilde{W}^T$, and the restricted mean vector is $\tilde{W}Fu_0$. The data likelihood becomes

$$\pi_{\text{like}}(\tilde{d}|u_0, w) \propto \exp\Big\{ -\frac{1}{2}(\tilde{d} - \tilde{W}Fu_0)^T \Big(\tilde{W}\Gamma_{\text{noise}}\tilde{W}^T\Big)^{-1} (\tilde{d} - \tilde{W}Fu_0) \Big\}.$$

Together with the prior information $u_0 \sim \mathcal{N}(u_{\text{prior}}, \Gamma_{\text{prior}})$, the posterior distribution can be easily computed as $\mathcal{N}(u_{\text{post}}, \Gamma_{\text{post}})$ where

$$\begin{aligned} \Gamma_{\text{post}} &= \Big(F^T \tilde{W}^T \big(\tilde{W}\Gamma_{\text{noise}}\tilde{W}^T\big)^{-1} \tilde{W}F + \Gamma_{\text{prior}}\Big)^{-1} \\ u_{\text{post}} &= \Gamma_{\text{post}}\Big(F^T \tilde{W}^T \big(\tilde{W}\Gamma_{\text{noise}}\tilde{W}^T\big)^{-1} \tilde{d} + \Gamma_{\text{prior}}^{-1} u_{\text{prior}}\Big). \end{aligned} \tag{1}$$

Because data error (or noise) from different sensors are assumed to be uncorrelated, Γ_{noise} is diagonal or block diagonal, i.e.

$$\big(W\Gamma_{\text{noise}}W^T\big)^{-1} = W\Gamma_{\text{noise}}^{-1}W^T.$$

In the case of homogeneous noise, that is, $\Gamma_{\text{noise}} = \sigma^2 I_n$, we have

$$\tilde{W}^T \big(\tilde{W}\Gamma_{\text{noise}}\tilde{W}^T\big)^{-1} \tilde{W} = \tilde{W}^T \Gamma_{\text{noise}}^{-1} \tilde{W} = \sigma^{-2}\tilde{W}\tilde{W}^T = \sigma^{-2}W.$$

Moreover, the following are all equal

$$\begin{aligned}\pi_{\text{like}}(d|u_0,w) &\propto \exp\Big\{-\frac{1}{2}(d-Fu_0)^T W^{1/2}\Gamma_{\text{noise}}^{-1}W^{1/2}(d-Fu_0)\Big\}\\ &= \exp\Big\{-\frac{1}{2}(Wd-WFu_0)^T W^{1/2}\Gamma_{\text{noise}}^{-1}W^{1/2}(Wd-WFu_0)\Big\}\\ &= \exp\Big\{-\frac{1}{2}(\tilde{d}-\tilde{W}Fu_0)^T\Big(\tilde{W}\Gamma_{\text{noise}}\tilde{W}^T\Big)^{-1}(\tilde{d}-\tilde{W}Fu_0)\Big\}\end{aligned}$$

W has only binary values on the diagonal and in the paper, we introduce a weight for each candidate sensor location so that W is diagonal with each entry between 0 and 1. For the purpose of estimation, after the locations are selected, d can be viewed as the potential measurement u in (1.2) with missing values in locations where there is no sensor and apply (1.3), or can be viewed as a low-dimensional copy of u and apply (1).

B Another Sum-Up Rounding Procedure for Rectangular Domains

We present the *sum-up rounding algorithm slowromancapii@* based on the following compatible two-level decomposition, with concepts defined in Definition 1. We use the notation

$$k_1(n_i) = \lfloor\sqrt{n_i}\rfloor, \quad \text{and} \quad k(n) = k_1(n_1)k_1(n_2)..k_1(n_P).$$

(i) On $[l_1^i, l_2^i]$ for $i=1,2,...,P$, group the first $k_1(n_i)$ intervals $\{\mathcal{I}_{i,j}\}_{j=1}^{k_1(n_i)}$ as $\mathcal{G}_{i,1}$, group the next $k_1(n_i)$ intervals $\{\mathcal{I}_{i,j}\}_{j=k_1(n_i)+1}^{2k_1(n_i)}$ as $\mathcal{G}_{i,2}$, and so forth until we get $\mathcal{G}_{i,k_1(n_i)}$. The remaining intervals $\{\mathcal{I}_{i,j}\}_{j=k_1(n_i)^2+1}^{n}$ are grouped as $\mathcal{G}_{i,last}$, and the number of intervals in $\mathcal{G}_{i,last}$ equals $n_i - k_1(n_i)^2$. Note that

$$\sqrt{n_i} - 1 < k_1(n_i) = \lfloor\sqrt{n_i}\rfloor \leq \sqrt{n_i}.$$

We can bound the number of intervals in the last group by

$$n_i - (\sqrt{n_i})^2 \le n_i - k_1(n_i)^2 < n_i - (\sqrt{n_i} - 1)^2$$
$$0 \le n_i - k_1(n_i)^2 < 2\sqrt{n_i},$$

so the cardinality of $\mathcal{G}_{i,\cdot}$ is $\mathcal{O}(\sqrt{n_i})$, and its size is $\mathcal{O}(1/\sqrt{n_i})$.

(ii) Consider a subdomain V_j of the form

$$\prod_{\substack{i=1,2,..,P \\ j_i \in \{1,2,..,k_1(n_i),last\}}} \mathcal{G}_{i,j_i}.$$

This decomposition has the following parameters and properties, in reference to Definition 1.

$$k(n) = \prod_{i=1}^{P} \lfloor \sqrt{n_i} \rfloor,\ \tilde{k}(n) = \prod_{i=1}^{P} \lceil \sqrt{n_i} \rceil,\ r(n) = k(n), \tag{2}$$

$$\rho(V_j) = \sqrt{\sum_{i=1}^{P} \left(\frac{(l_2^i - l_1^i)}{\lfloor \sqrt{n_i} \rfloor} \right)^2}, \quad j = 1, 2, \ldots, k(n) \tag{3}$$

Theorem B.1. *Under the assumptions of Theorem 3.3.3, there exists a C such that the sum-up rounding algorithm slowromancapii@ construction satisfies*

$$\Big| \sum_{k=1}^{n} f(x_k) \Big(w^n(x_k) - \tilde{w}^n(x_k) \Big) \Delta_x \Big| \le \frac{C}{n^{1/2P}}.$$

Proof. We use the definitions of the sum-up rounding procedure parameters (2)–(3), and the inequalities (3.20)-(3.21) to infer the following inequalities:

$$\frac{1}{\sqrt{n_i}} \le c_1^{-\frac{1}{2}} n^{-\frac{1}{2P}},\ i = 1, 2, \ldots, P; \quad \frac{1}{r(n)} = \prod_{i=1}^{P} \frac{1}{\lfloor \sqrt{n_i} \rfloor} \overset{(3.20)}{\le} \frac{2^{\frac{P}{2}}}{\sqrt{n}}. \tag{4}$$

For the maximum diameter of V_j we obtain from (3) and (3.20)

$$\max_{j=1,2,\dots,k(n)} \rho(V_j) \leq \sqrt{P}\frac{\max_{i=1,2,..,P}(l_2^i - l_1^i)}{\frac{1}{2}\min_{i=1,2,\dots,P}\sqrt{n_i}} \overset{(3.21)}{\leq} \sqrt{P}\frac{\max_{i=1,2,..,P}(l_2^i - l_1^i)}{\frac{1}{2}\sqrt{c_1}} n^{-\frac{1}{2P}}. \quad (5)$$

We also obtain

$$1 - \frac{k(n)r(n)}{n} = 1 - \prod_{i=1}^{P} \frac{\lfloor\sqrt{n_i}\rfloor^2}{n_i} \leq 1 - \prod_{i=1}^{P}\Big(1 - \frac{1}{\sqrt{n_i}}\Big) \overset{(3.21)}{\leq} 1 - \Big(1 - c_1^{-\frac{1}{2}} n^{-\frac{1}{2P}}\Big)^P.$$

In turn, from the mean value theorem applied to $(1-x)^P$ for $x \in [0,1]$ and the last inequality, we have

$$1 - (1-x)^P \leq Px,\ \forall x \in [0,1] \Rightarrow 1 - \frac{k(n)r(n)}{n} \leq Pc_1^{-\frac{1}{2}} n^{-\frac{1}{2P}}. \quad (6)$$

We now use Theorem 3.3.3 along with (3.20)-(3.21) , (4), (5), and (6) to obtain the statement of this theorem for the *sum-up rounding algorithm* slowromancapii@ with the choice

$$C = \max_{x\in V} |f(x)|\mu(V) 2^{\frac{P}{2}} + 2L\mu(V)\sqrt{P}\frac{\max_{i=1,2,..,P}(l_2^i - l_1^i)}{\frac{1}{2}\sqrt{c_1}} + 2\max_{x\in V} f(x)\mu(V)Pc_1^{-\frac{1}{2}}.$$

□

C The SQP Algorithm for D-optimal Design

The algorithm is very similar to the one for A-optimal design, except that the gradient and Hessian for D-optimal design objective function are different.

Gradient of logdet objective

First we find the derivatives to the logdet of Γ_{post}:

$$\frac{\partial\ logdet(\Gamma_{\text{post}})}{\partial w_i} = -tr\Big((F^TWF + I_n)^{-1} f_i f_i^T\Big) = -f_i^T(F^TWF + I_n)^{-1} f_i$$

Hessian of logdet objective

The $(i, j)^{th}$ entry of the Hessian matrix is

$$H_{ij} = \frac{\partial^2 logdet(\Gamma_{\text{post}})}{\partial w_i \partial w_j} = \left(f_i^T (F^T W F + I_n)^{-1} f_j\right)^2.$$

Approximation of gradient and Hessian

We will give details for the one-dimensional case, and the procedure can be extended trivially to rectangle domains in multiple dimensions using tensor product. For the input domain, let $\{\bar{x}_i\}_{i=1}^N$ be the N Chebyshev interpolation points, $\{x_i\}_{i=1}^n$ be the n discretization points on the mesh and note $N = O(\log(n))$, and $C_x \in \mathbb{R}^{n\times N}$ be the matrix of interpolation coefficients (see §4.2). Similarly, we can construct C_y for the output domain. We approximate F by

$$F_s = C_x^T \bar{F} C_y$$

where $\bar{F} \in \mathbb{R}^{N\times N}$ is the matrix of $f(\bar{x}_i, \bar{x}_j)$ evaluated at interpolation points. Next, we construct $M \in \mathbb{R}^{n\times N}$ by setting its $(i, j)^{th}$ entry to be

$$\bar{f}_i^T (F_s^T W F_s + I_n)^{-1} \bar{f}_i$$

where $\bar{f}_i$ is the i^{th} column of $C_y^T F_s^T$. Then we approximate the i^{th} gradient by $c_x(x_i)^T M(i,i) c_x(x_i$. To approximate the Hessian H, let $\bar{H} \in \mathbb{R}^{N\times N}$ and $\bar{H}(i,j) = M(i,j)^2$, and then

$$H \approx C_x^T \bar{H} C_x.$$

Once we figure out the gradient and Hessian approximations, it should be clear on the implementation of the sequential quadratic programming algorithm 1 in §4.3.

Error analysis

All the error analysis in §4.4 applies to the log-determinant case, and we only need to modify one step in Claim 4.4.2:

$$\begin{aligned}
|\phi(w) - \phi_s(w)| &= \big|\sum_{i=1}^{n} \log \frac{1}{1+\lambda_i^n} - \sum_{i=1}^{n} \log \frac{1}{1+\lambda_i^{n,s}}\big| \\
&\leq \big|\sum_{i=1}^{N} \big(\log \frac{1}{1+\lambda_i^n} - \log \frac{1}{1+\lambda_i^{n,s}}\big)\big| + \big|\sum_{i=N+1}^{n} \big(\log \frac{1}{1+\lambda_i^n} - \log \frac{1}{1+\lambda_i^{n,s}}\big)\big| \\
&= \sum_{i=1}^{N} \big|\log(1+\lambda_i^{n,s}) - \log(1+\lambda_i^n)\big| + \sum_{i=N+1}^{N} \log(1+\lambda_i^n) \\
&\leq \sum_{i=1}^{N} |\lambda_i^n - \lambda_i^{n,s}| + \sum_{i=N+1}^{n} \lambda_i^n .
\end{aligned}$$

References

[1] D.R. Cox and N. Reid. *The theory of the design of experiments*. Chapman & Hall/CRC, 2000.

[2] Andrew Mead Roger Mead, Steven Gilmour. *Statistical Principles for the Design of Experiments*. Cambridge University Press, 2012.

[3] A. N. Donev A. C. Atkinson. *Optimum Experimental Designs*. Oxford University Press, 1992.

[4] Peter Hackl Valerii V. Fedorov. *Model-Oriented Design of Experiments*. Springer, 1997.

[5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[6] F. Pukelsheim. *Optimal design of experiments*. Classics in Applied Mathematics, SIAM, 2006.

[7] Stephen M Stigler. Optimal experimental design for polynomial regression. *Journal of the American Statistical Association*, 66(334):311–318, 1971.

[8] Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas. A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized l_0-sparsification. *SIAM Journal on Scientific Computing*, 36(5):A2122–A2148, 2014.

[9] Venkat Roy, Andrea Simonetto, and Geert Leus. Spatio-temporal field estimation using kriged kalman filter (kkf) with sparsity-enforcing sensor placement. *Sensors*, 18(6):1778, 2018.

[10] Krithika Manohar, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86, 2018.

[11] Daniel G Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

[12] Georges Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

[13] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.

[14] W. I. Notz T. Santner, B. J. Williams. *The design and analysis of computer experiments.* Heidelberg: Springer, 2003.

[15] Steve Smale Felipe Cucker. On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39(1):1–49, 2002.

[16] Peter Bartlett. Prediction algorithms: complexity, concentration and convexity.

[17] Luc Pronzato. Optimal experimental design and some related control problems. *Automatica*, 44(2):303–325, 2008.

[18] Andrew M Stuart. Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559, 2010.

[19] Noemi Petra, James Martin, Georg Stadler, and Omar Ghattas. A computational framework for infinite-dimensional Bayesian inverse problems: Part II. Stochastic Newton MCMC with application to ice sheet inverse problems. *SIAM Journal on Scientific Computing*, 36(4):A1525–A1555, 2014.

[20] Alen Alexanderian and Arvind K Saibaba. Efficient d-optimal design of experiments for infinite-dimensional bayesian linear inverse problems. *SIAM Journal on Scientific Computing*, 40(5):A2956–A2985, 2018.

[21] S. Sager, H.G. Bock, and M. Diehl. The integer approximation error in mixed-integer optimal control. *Math. Program., Ser. A*, 2012.

[22] S. Sager. Sampling decision in optimum experimental design in the light of Pontryagin's maximum principle. *SIAM J. Control Optim.*, 2013.

[23] K. Atkinson and W. Han. *Theoretical numerical analysis*, volume 39. Springer, 2005.

[24] Alessio Spantini, Antti Solonen, Tiangang Cui, James Martin, Luis Tenorio, and Youssef Marzouk. Optimal low-rank approximations of bayesian linear inverse problems. *SIAM Journal on Scientific Computing*, 37(6):A2451–A2487, 2015.

[25] Anne Greenbaum and Timothy P Chartier. *Numerical methods: design, analysis, and computer implementation of algorithms*. Princeton University Press, 2012.

[26] Sebastian Sager. Sampling decisions in optimum experimental design in the light of Pontryagin's maximum principle. *SIAM Journal on Control and Optimization*, 51(4):3181–3207, 2013.

[27] Jonathan W Berry, Lisa Fleischer, William E Hart, Cynthia A Phillips, and Jean-Paul Watson. Sensor placement in municipal water networks. *Journal of Water Resources Planning and Management*, 131(3):237–243, 2005.

[28] Jonathan Berry, William E Hart, Cynthia A Phillips, James G Uber, and Jean-Paul Watson. Sensor placement in municipal water networks with temporal integer programming models. *Journal of Water Resources Planning and Management*, 132(4):218–224, 2006.

[29] Jean-Paul Watson, Harvey J Greenberg, and William E Hart. A multiple-objective analysis of sensor placement optimization in water networks. In *Proceedings of the World Water and Environment Resources Congress. American Society of Civil Engineers*, 2004.

[30] Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6):516–526, 2008.

[31] Donald J Chmielewski, Tasha Palmer, and Vasilios Manousiouthakis. On the theory of optimal sensor placement. *AIChE Journal*, 48(5):1001–1012, 2002.

[32] Dries Telen, Filip Logist, Rien Quirynen, Boris Houska, Moritz Diehl, and Jan Impe. Optimal experiment design for nonlinear dynamic (bio) chemical systems using sequential semidefinite programming. *AIChE Journal*, 60(5):1728–1739, 2014.

[33] Abhay K Singh and Juergen Hahn. Sensor location for stable nonlinear dynamic systems: Multiple sensor case. *Industrial & Engineering Chemistry Research*, 45(10):3615–3623, 2006.

[34] Estanislao Musulin, Chouaib Benqlilou, Miguel J Bagajewicz, and Luis Puigjaner. Instrumentation design based on optimal Kalman filtering. *Journal of Process Control*, 15(6):629–638, 2005.

[35] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[36] Carl M Harris, Karla L Hoffman, and Leslie-Ann Yarrow. Using integer programming techniques for the solution of an experimental design problem. *Annals of Operations Research*, 58(3):243–260, 1995.

[37] Richard G Carter, Todd F Dupont, and Henry H Rachford Jr. Pack management and transient optimization of natural gas transmission lines. In *IGRC Conference, Vancouver*, 2004.

[38] Henry H Rachford Jr, Richard G Carter Advantica, and Todd F Dupont Advantica. Using optimization in transient gas transmission. In *PSIG Annual Meeting*. Pipeline Simulation Interest Group, 2009.

[39] H Prashanth Reddy, Shankar Narasimhan, and S Murty Bhallamudi. Simulation and state estimation of transient flow in gas pipeline networks using a transfer function model. *Industrial & Engineering Chemistry Research*, 45(11):3853–3863, 2006.

[40] L Billmann and Rolf Isermann. Leak detection methods for pipelines. *Automatica*, 23(3):381–385, 1987.

[41] Victor M Zavala. Stochastic optimal control model for natural gas networks. *Computers & Chemical Engineering*, 64:103–113, 2014.

[42] JK Van Deen and SR Reintsema. Modelling of high-pressure gas transmission lines. *Applied Mathematical Modelling*, 7(4):268–273, 1983.

[43] Robert MM Mattheij, Sjoerd W Rienstra, and Jan HM ten Thije Boonkkamp. *Partial Differential Equations: Modeling, Analysis, Computation*. SIAM, 2005.

[44] Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*, volume 160 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2005.

[45] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[46] Sebastian Sager, Hans Georg Bock, and Moritz Diehl. The integer approximation error in mixed-integer optimal control. *Mathematical Programming*, 133(1-2):1–23, 2012.

[47] David Wiklund and Marcos Peluso. Quantifying and specifying the dynamic response of flowmeters. *TECHNICAL PAPERS-ISA*, 422:463–476, 2002.

[48] W.J. Welch. Algorithmic complexity: three np-hard problems in computational statistics. *Journal of Statistical Computation and Simulation*, 15(1):17–25, 1982.

[49] S.S. Iyengar and R.R. Brooks. *Distributed Sensor Networks, Second Edition: Sensor Networking and Applications*. Chapman and Hall/CRC, October 26, 2016.

[50] T. Berthold. RENS – the optimal rounding. *Math. Prog. Comp.*, 2014.

[51] A. Lodi, P. Bonami, G. Cornujols, and F. Margot. A feasibility pump for mixed integer nonlinear programs. *Math. Program.*, 2009.

[52] G. Nannicini and P. Belotti. Rounding-based heuristics for nonconvexminlps. *Math. Program. Comput.*, 2012.

[53] S. Balas, E. andalex Ceria, M. Dawande, F. Margot, and G. Pataki. Octane: a new heuristic for pure 01 programs. *Operat. Res.*, 2001.

[54] P. L. Hammer, E.L. Johnson, and U.N. Peled. A feasibility pump for mixed integer nonlinear programs. *Math. Program.*, 2009.

[55] L.A. Wolsey. Faces for a linear inequality in $0-1$ variables. *Math. Program.*, 1975.

[56] J. Berry, W.E. Hart, C.A. Phillips, J.G. Uber, and J.P. Watson. Sensor placement in municipal water networks with temporal integer programming models. *Journal of Water Resources Planning and Management*, 2006.

[57] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 2008.

[58] J.P. Watson, H.J. Greenberg, and W.E. Hart. A multiple-objective analysis of sensor placement optimization in water networks. In *Proceedings of the World Water and Environment Resources Congress*.

[59] J. Yu, V.M. Zavala, and M. Anitescu. A scalable design of experiments framework for optimal sensor placement. *Journal of Process Control*, 2017.

[60] A. Drăgănescu. Multigrid preconditioning of linear systems for semi-smooth newton methods applied to optimization problems constrained by smoothing operators. *Optimization Methods and Software*, 29(4):786–818, 2014.

[61] E.A. Kendall. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, Jun 28, 1997.

[62] Y. Wang, A.G. Yagola, and C. Yang. *Computational Methods for Applied Inverse Problems*. Higher Education Press, October 2012.

[63] N. Cressie and C.K. Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2015.

[64] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer Series in Statistics, New York, 2001.

[65] J. Kaipio and E. Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.

[66] H. Wielandt. Error bounds for eigenvalues of symmetric integral equations. In *Proc. Sympos. Appl. Math*, volume 6, pages 261–282, 1956.

[67] Yongjie Zhang and Chandrajit Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer methods in applied mechanics and engineering*, 195(9-12):942–960, 2006.

[68] A.T. Patera. *A spectral element method for fluid dynamics: laminar flow in a channel expansion*, volume 54. Elsevier, 1984.

[69] T. Blacker. Meeting the challenge for automated conformal hexahedral meshing. In *9th international meshing roundtable*, pages 11–20, 2000.

[70] R.A. Horn and C.R. Johnson. *Matrix analysis.* Cambridge University Press, 1990.

[71] A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas. A fast and scalable method for a-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 38(1):A243–A272, 2016.

[72] A. Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems.* Springer, 2011.

[73] C.J. Geoga, M. Anitescu, and M.L. Stein. Scalable gaussian process computations using hierarchical matrices. *arXiv preprint arXiv:1808.03215*, 2018.

[74] Jing Yu and Mihai Anitescu. Multidimensional sum-up rounding for integer programming in optimal experimental design (preprint). *Mathematical Programming*, 2017.

[75] Rajendra Bhatia. *Matrix Analysis*, volume 169. Springer Science & Business Media, 1996.

[76] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, New York, 2nd edition, 2006.

[77] L. Brutman. *Lebesgue functions for polynomial interpolation - a survey.* Annals of Numerical Mathematics, 1996.

[78] Lawrence C. Evans. *Partial Differential Equations.* American Mathematical Soc., 2010.

[79] AF Bennett and WP Budgell. Ocean data assimilation and the moan filter: spatial regularity. *Journal of physical oceanography*, 17(10):1583–1601, 1987.

[80] Jordan Bell. The singular value decomposition of compact operators on hilbert spaces, 2014.

[81] Barry Simon. *Operator Theory: A Comprehensive Course in Analysis, Part 4*. American Mathematical Society, 2015.

[82] Jerrold E Marsden and Thomas JR Hughes. *Mathematical foundations of elasticity*. Courier Corporation, 1994.

[83] LR Hunt and Renjeng Su. Linear approximations of nonlinear systems. In *The 22nd IEEE Conference on Decision and Control*, pages 122–125. IEEE, 1983.

[84] R Pintelon and J Schoukens. The best linear approximation of nonlinear systems operating in feedback. In *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 2092–2097. IEEE, 2012.