Numerical analysis for derivative-free optimization

by

Gabriel Jarry-Bolduc

B.Sc. Hons., Université du Québec à Trois-Rivières , 2017
M.Sc., The University of British Columbia, 2019

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE COLLEGE OF GRADUATE STUDIES

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

July 2023

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis/dissertation entitled:

NUMERICAL ANALYSIS FOR DERIVATIVE-FREE OPTIMIZATION

submitted by GABRIEL JARRY-BOLDUC in partial fulfilment of the requirements of the degree of Doctor of Philosophy

Dr. Warren Hare, Irving K. Barber Faculty of Science
Supervisor

Dr. Wayne Broughton, Irving K. Barber Faculty of Science
Supervisory Committee Member

Dr. Solomon Tesfamariam, School of Engineering
Supervisory Committee Member

Dr. Michael Friedlander, Faculty of Science
University Examiner

Dr. Stefan Wild, Lawrence Berkeley Laboratories
External Examiner

Abstract

In many optimization problems, the objective function is available only as the output of a *blackbox*. In this context, *derivative-free optimization* (DFO) methods can be used to solve the optimization problem. Derivative-free optimization methods may be classified into two main categories: *direct search methods* and *model-based methods*. This thesis presents novel theory and algorithms in both categories.

Model-based methods often approximate the gradient or the Hessian of the objective function. The properties of the gradient approximation technique called *generalized simplex gradient* are scrutinized. Second, two Hessian approximation techniques called *generalized simplex Hessian* (GSH) and *generalized centered simplex Hessian* (GCSH) are defined and analyzed. In particular, we investigate how to approximate partial Hessians, and minimize the number of function evaluations when employing the GSH/GCSH.

A useful notion in direct search methods is that of *positive spanning set*. In this thesis, we present the first deterministic algorithm to compute the *cosine measure* of any finite positive spanning set. Then we investigate the structure of *positive bases* with maximal cosine measure. We focus on positive bases of *intermediate sizes*.

Last, the main theoretical concepts discussed in this thesis are utilized in DFO algorithms. The GSH are employed in a *derivative-free trust-region algorithm* and positive bases with high cosine measure are employed in a direct search algorithm. We examine how the theoretical advancements made in this thesis translate to gains in efficiency in DFO algorithms.

Lay Summary

In many optimization problems arising from different fields, such as engineering and artificial intelligence applications, the objective function is available only as the output of a *blackbox*. In this thesis, A blackbox is defined as any process that returns an output whenever the optimizer provides an input, but the inner mechanism of the process is not analytically available to the optimizer. An example of a blackbox is a computer simulation.

In this context, *derivative-free optimization* algorithms can be used to solve the optimization problem. Three topics that can play an important role when designing derivative-free algorithms are: *gradient approximation techniques*, *Hessian approximation techniques* and *positive bases*. In this thesis, we develop new results concerning these three topics. Furthermore, some of the theoretical advancements made are employed in derivative-free optimization algorithms to help design more efficient algorithms.

Preface

This thesis is based on the following manuscripts: [HJB23, HJBP23b, HJBP23a, JB22, HJB20, HJP20, HJBP22]. All of these manuscripts are published in peer-reviewed scientific journals except [HJBP22] which has been accepted June 19, 2023 in IMA Journal of Numerical Analysis.
The following manuscript was solo-authored: [JB22].
The following manuscripts were co-authored by Dr. Warren Hare: [HJB23, HJB20].
The following manuscripts were co-authored by Dr. Warren Hare and Dr. Chayne Planiden: [HJBP23b, HJBP23a, HJP20, HJBP22].

A breakdown of how the manuscripts relate to each chapter is as follows.
Chapter 1: this chapter includes new material. It is partially based on introductions from all manuscripts.
Chapter 2: this chapter includes new material. It is partially based on the introductions from all manuscripts listed above.
Chapter 3: this chapter contains collective definitions and background from all manuscripts.
Chapter 4: this chapter is based on [HJP20, HJBP23a].
Chapter 5: this chapter includes new material in Section 5.5. The remainder of this chapter is based on [HJBP22, JB22].
Chapter 6: this chapter includes new material in Section 6.5. The results were developed while working with Sébastien Kerleau and Dr. Clément Royer in September 2022. The remainder of this chapter is based on [HJB20, HJBP23b].
Chapter 7: this chapter includes new material in Section 7.2. The remainder of this chapter is based on [HJB23].

Table of Contents

List of Tables

List of Figures

Acknowledgements

I wish to thank my supervisor Dr. Warren Hare for his assistance and for sharing all his knowledge about mathematics with me, but also about how to have a successful career as a mathematician. I could not have asked for a better supervisor. I now feel ready to begin the next step of my career. Thank you.

I wish to thank Dr. Chayne Planiden for always being enthusiastic about collaborating on research projects. It is always a pleasure to work with you. Thank you.

I wish to thank Dr. Wayne Broughton for his help while working on developing an algorithm to compute the cosine measure of a positive basis. I truly enjoyed taking your classes, and your outstanding teaching skills have been an inspiration to me.

I wish to thank the committee members to have the courage to read this thesis. I am sure you have many other tasks on your to-do list. Thank you for your time.

This work has been financially supported by UBC and by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Discover Grant #2018-03865.

Dedication

À mes parents, Michèle et René, qui m'ont toujours supporté. Cette idée d'étudier les mathématiques semblait complètement insensée il y a dix ans. Merci d'avoir cru en moi.

À Katrina, pour tous les sacrifices que tu as fait pour que je puisse compléter mes études.

À Jasper, j'espère que tu n'aimeras pas les mathématiques. Il semble exister des métiers plus facile pour gagner sa vie.

Je vous aime et je suis infiniment reconnaissant pour tout ce que vous avez fait pour moi.

Chapter 1

Introduction

One of the fundamental goals of an optimizer, i.e., a person who is interested by an optimization problem, is to find the minimum value, or maximum value, of an *objective function*. A problem seeking for the minimum value and the set of points achieving this value (these points are called *minimizers*) is called a minimization problem. A problem seeking the maximum value and the set of points achieving this value (these points are called *maximizers*) is called a maximization problem. Since a maximization problem can be transformed into a minimization problem without any difficulty, we only consider minimization problems in this thesis. In some situations, the objective function is accompanied by a number of restrictions that must be respected by a potential solution. These restrictions are constraints. A function that quantifies a constraint is called a *constraint function*.

In many optimization problems, the objective function and/or the constraint functions are available only as the output of a *blackbox*. In this thesis, a blackbox is defined as any process that returns an output whenever an input is provided, but the inner mechanism of the process is not analytically available. In this thesis, we assume that the blackbox only provides zeroth-order information as an output. One example of a blackbox is a computer simulation. In this context, the optimizer could use *derivative-free optimization* (DFO) methods to solve the optimization problem.

Derivative-free optimization methods may be classified into two main categories: *direct search methods* and *model-based methods*. A model-based method approximates the objective function by building model functions. Then it utilizes these model functions to guide the optimization. A popular type of model function is a quadratic model. If the objective function is smooth, then a quadratic model can be built by approximating the gradient and the Hessian of the objective function at the incumbent solution. Techniques to approximate gradients are the topic of Chapter 4. Techniques to approximate full Hessians or partial Hessians, are the topic of Chapter 5.

A direct search method is a method that uses only function values and does not, in its heart, approximate the gradient or the Hessian. It works from an incumbent solution and uses a set of directions to explore the space and

possibly find an improvement in the objective function. The set of directions utilized in such methods is generally from a *positive spanning set*. In general, in blackbox optimization, a good set of directions to explore the space is a set of directions that cover the whole space uniformly. To measure this property, the notion of the *cosine measure* was defined in [Tor97]. Cosine measure and positive spanning sets are discussed in Chapter 6.

There are two main goals in this thesis. The first goal is to present theoretical advancements made in the last years related to gradient approximations, Hessian approximations, and positive spanning sets. The second goal is to verify if the theoretical results obtained in the first part of this thesis can be valuable in DFO algorithms. To achieve this goal, some of the theoretical concepts presented in this thesis are implemented in DFO algorithms and it is verified when it translates into significant gains in terms of efficiency.

This thesis is organized as follows.

- Chapter 2 establishes the context. It provides information on when and why DFO methods could be valuable. The second part of the chapter presents 8 categories in which a DFO algorithm may be classified. While presenting these categories, we clarify on which of these categories this thesis focuses.

- Chapter 3 introduces the notation, definitions, and background theory used throughout this thesis.

- Chapter 4 discusses techniques to approximate gradients using a finite set of sample points. Two gradient approximation techniques called the *generalized simplex gradient* (GSG) and the *generalized centered simplex gradient* (GCSG) are defined. Error bounds for both approaches are provided. The limiting behavior of these techniques as the number of sample points approaches infinity in a fixed region is examined. Error bounds *ad infinitum* are proposed.

- Chapter 5 presents a novel matrix algebra approach to approximate Hessians based on simple matrix algebra. Two Hessian approximation techniques called the *generalized simplex Hessian* (GSH) and the *generalized centered simplex Hessian* (GCSH) are defined. General error bounds for both approaches are provided. How to approximate a proper subset of the entries of the Hessian with the GSH or the GCSH is investigated.

– Chapter 6 focuses on positive spanning sets. First, a deterministic algorithm to compute the cosine measure of any finite positive spanning set is introduced. Second, results regarding the structure of positive bases with maximal cosine measure are presented. These results focus on positive bases of *intermediate sizes*.

– Chapter 7 presents a *derivative-free trust-region algorithm*. The algorithm uses the GSH to build model functions. Several numerical experiments are conducted to verify if a calculus-based approach to build the model function can be more efficient than a non-calculus approach. In the second part of Chapter 7, positive bases with high cosine measure are utilized in a direct search algorithm and the performance of the algorithm is compared to a version of the algorithm using positive bases with low cosine measure.

– Chapter 8 summarizes the main achievements of this thesis.

Chapters 4, 5, 6, and 7 end by presenting possible future research directions.

Chapter 2

Context

2.1 Why DFO methods

Derivative-free optimization methods is a class of optimization methods that do not use derivatives. Note that the objective function and the constraint functions may be fully differentiable. However, we note that derivatives are not employed directly in DFO algorithms.

It is undeniable that derivatives contain valuable information needed to find the minimum value of a function. Indeed, for continuously differentiable functions, a necessary condition to be a local minimum is that the first-order derivatives are equal to zero. Note that in the case where derivatives are available, reliable, and may be obtained at a reasonable cost, then DFO algorithms are usually less efficient than gradient-based algorithms [AH17]. However, in many situations, the derivatives of the function of interest may be unavailable or unreliable. As mentioned earlier, an example where the derivatives are not available is when the function is given through a blackbox. This is the case when the output of a function is obtained through a computer simulation, and *automatic differentiation* techniques cannot be applied [GC91]. Another example of a blackbox is a laboratory experiment [AH17, Chapter 1]. Since there is no explicit function associated to the experiment, derivatives are not available to the optimizer.

In blackbox optimization, the classical assumption is to assume that no information is known about the function hidden in the blackbox. In the case where some characteristics of the blackbox is known, researchers sometimes refer to this situation as a *greybox*. Note that some researchers still refer to this situation as a blackbox and refer to the situation where no information is known at all as a *pure blackbox* [LLRV19].

To optimize problems of the types described in the previous paragraph, DFO methods have been developed. Their popularity may be partially explained by the increasing complexity in mathematical modeling and the sophistication of scientific computing. Their popularity may also be explained by the significant amount of codes that have been written in the past, where derivative information was not readily encoded. Rewriting the

code to provide first-order information might be very time consuming and for this reason, it might not be a viable option. The optimization problem could also depend on a code owned by a company and released as closed source software. Hence, legacy or proprietary issues also justify the importance of DFO methods [CSV09b, Chapter 1].

Derivative-free optimization methods have been effectively applied to a broad range of fields. A survey of DFO methods and their applications can be found in [Aud14, AH20, LMW19, HNT13]. It is very likely that DFO methods will remain popular in a near future and that the lists of applications will continue to grow.

In the next section, we present some of the categories that can be used to classify DFO algorithms.

2.2 Categorizing optimization problems and DFO algorithms

In this section, we present some of the major categories of optimization problems and algorithms.

Linear versus non-linear

We say that an optimization problem is linear if the objective function is linear and all constraints are linear (if any). If the objective function is non-linear or one of the constraints is non-linear, then the optimization problem is said to be non-linear. In this thesis, we assume non-linear optimization problems.

Unconstrained versus constrained

An optimization problem is said to be unconstrained if the *feasible region* is the whole space $\mathbb{R}^n$. The optimization problem is said to be constrained if the feasible region is a proper subset of $\mathbb{R}^n$ [CGT00]. There exist several sub-categories to describe constraints such as

- *equality* versus *inequality*,
- *relaxable* versus *unrelaxable*,
- *hidden* versus *explicit*,
- *algebraic* versus *simulation-based*,

- *quantitative* versus *non-quantitative*,
- *linear* versus *non-linear*,
- *active* versus *inactive*,
- *convex* versus *non-convex*,
- *soft* versus *hard*,
- *simple bound* versus *general*.

More details about these types of constraints and DFO methods developed to specifically solve some of these types of constraints can be found in [AH17, CSV09b, LMW19, LW15]. The theoretical concepts discussed in Chapters 4, 5, and 6 can be used to solve either constrained or unconstrained optimization problems. Chapter 7 contains one DFO algorithm to solve a bound constrained optimization problem (Section 7.1) and one DFO algorithm to solve an unconstrained optimization problem (Section 7.2). The bound constraints will be treated as relaxable in Chapter 7.

Continuous versus discrete

Continuous optimization problems are those where the *constraint space* allows a continuous selection of variables [AH17]. In other words, the variables can take any real value. In contrast, the constraint space does not allow a continuous selection of variables in discrete optimization problems. Some researchers refer to the situation where some of the variables are continuous and some are discrete as *mixed variables*. Furthermore, if the discrete variables are integers, then we say it is a *mixed-integer* problem. There exists several categories to describe further the type of discrete variable such as *categorical*, *periodical* and *binary* (see [AH17, Chapter 12] or [LMW19]). Most DFO methods developed over the years have assumed a continuous optimization problem. However, many DFO methods now exist for optimization problems involving discrete variables such as the ones described in [AAD07, AACW09, AD01, LLR12, LLR15, LLR20, NA15, PT17, SCA09]. This thesis considers continuous optimization problems.

A further classification of continuous optimization problems is *smooth* and *non-smooth*. A smooth optimization problem is one where the objective function and the constraint set are represented using twice continuously differentiable functions. Non-smooth means that there is at least one point where the gradient, or Hessian of the objective function, or one of the

constraint functions delimiting the feasible region, is undefined. Most research done on DFO methods have focused on smooth problems. However, there now exist several results and DFO methods for non-smooth optimization problems [ABLD08, BU06, BKS08, BGP09, BCL$^+$20, CDV08, FLLR14, GJV16, HN13, HL14, LMW16, LMZ21, LLR16, LLRV19].

Deterministic versus stochastic

Let f be a real function and $\tilde{f}$ be a stochastic function. A deterministic (minimization) optimization problem takes the form

$$\begin{aligned} \underset{x}{\text{Minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \Omega \subseteq \mathbb{R}^n, \end{aligned} \tag{2.1}$$

and a stochastic (minimization) optimization problem takes the form

$$\begin{aligned} \underset{x}{\text{Minimize}} \quad & \mathbb{E}_\xi[\widetilde{f}(x;\xi)] \\ \text{subject to} \quad & x \in \Omega \subseteq \mathbb{R}^n, \end{aligned}$$

where $\mathbb{E}_\xi$ denotes the expectation with respect to ξ, and where ξ is a random variable. Developing DFO methods for stochastic optimization problems is currently an active area of research and some of the publications on this topic include [CK16, CMS18, DF06, DF09, DW22, KZ10, LB16, PS20, BW23, SAM10, SCA09]. The results and algorithms in this thesis were developed assuming that the optimization problems are deterministic.

Single-objective versus multi-objective

A deterministic single objective function is defined in (2.1). A deterministic multi-objective function is usually defined as

$$\begin{aligned} \underset{x}{\text{Minimize}} \quad & F(x) \\ \text{subject to} \quad & x \in \Omega \subseteq \mathbb{R}^n, \end{aligned}$$

where $F(x) = \big[f_1(x), \ldots, f_q(x)\big], q > 1$, and $f_k : \mathbb{R}^n \to \mathbb{R}$ for all $k \in \{1, \ldots, q\}$. Minimizing a vector means that each each entry in F is minimized simultaneously. One of the main challenges in presence of a multi-objective function is that the functions f_k may be in conflict in the sense that there is not necessarily a single point $x \in \Omega$ that attains the minimum of all functions f_k. The solution to a multi-objective problem can be viewed as

a set of trade-off points and is called the *Pareto set* [AH17, Chapter 14]. Most of the DFO methods developed over the years are for single objective function. However, multi-objective optimization is currently an active area of research [ASZ08, ASZ10, CLPS18, CMVV11, CM18, LLR16, Reg16a, Reg16b, SAHT20]. This thesis assumes that the optimization problem of interest is single objective. However, the results developed in Chapters 4, 5, and 6 could be used in multi-objective optimization problems by applying the results to each objective individually.

To conclude this section, three additional categories that can be used to classify optimization algorithms are presented.

Local versus global

Local means that the algorithm seeks a local solution. That is a point at which the value of the objective function is no larger than the values of the objective function at all other nearby points. On the other hand, a global algorithm seeks a point where the value of the objective function is smaller than all other points in the feasible region. Many derivative-free algorithms developed are designed to find a local solution. Results on global DFO algorithms can be found in [CM15, FK06, HWS09, HN08, LZY15, VDHL17]. The theoretical concepts developed in Chapter 4, 5 and 6 can be used in either local or global algorithms. The DFO algorithms discussed in Chapter 7 are local DFO algorithms.

First-order versus second-order

In this thesis, we call an algorithm a first-order algorithm if it ensures convergence to a *first-order critical point* [CSV09a].[1] A first-order critical point in a smooth unconstrained optimization problem is a point x^* in $\mathbb{R}^n$ such that the gradient of the objective function at x^* is equal to the zero vector. A second-order algorithm is an algorithm that ensures convergence to a *second-order critical point*. A second-order critical point in a smooth unconstrained optimization problem is a point x^* in $\mathbb{R}^n$ where the gradient of the objective function is equal to the zero vector and where the Hessian of the objective function at x^* is *positive semi-definite* [CGT00, Section 3.2.1]. Most DFO algorithms developed over the years are first-order algorithms. Publications on second-order DFO algorithms include

[1] We note that this terminology is not consistent in the literature. For example, see [Bec17].

[Abr05, AA06, AFS14, CGT00, CSV09a, GRV16, Jam21]. The algorithms developed in Chapter 7 are first-order DFO algorithms.

Heuristic versus non-heuristic

A heuristic algorithm is defined as any algorithm that is not guaranteed to find a solution to the optimization problem. It might be supported by some arguments of why it should succeed, but the mathematical convergence results to support the effectiveness of the algorithm are limited. In contrast, non-heuristic algorithms are supported by a convergence theory and contain a stopping criterion that provides some assurance of optimality [AH17]. There exists a wide range of publications on heuristic DFO algorithms, and this type of algorithm is still often used to solve optimization problems in engineering among other fields. Reasons to explain their popularity are that they are easy to understand, easy to implement, and when function evaluations are computationally inexpensive, they can be efficient. Some of the publications on heuristic algorithms and their applications in structural engineering are [Gee09, GC00, Has08, KK12, KGV83, LPK02, LL11, MMW12, SG13, SAHT20]. This thesis focuses on non-heuristic algorithms.

2.3 A brief overview of DFO methods

To conclude this chapter, a brief overview of DFO methods is presented. It mainly covers the DFO methods meant to address the types of problems that can be classified in the same categories as the work in this thesis: non-linear, continuous, deterministic, single-objective, local, first-order, and non-heuristic.

Early works on DFO methods were done in the United Kingdom and Soviet Union [Box66, Fle65, M$^+$65, NM65, Pow64, Ros60, Ras63].

As mentioned earlier, DFO methods are usually classified into two main categories: model-based methods and direct search methods. Essentially, model-based methods use a surrogate of the function of interest to determine a candidate point, and direct search methods compare function evaluations to directly choose a candidate point. Note that it is not always easy to classify a derivative-free method into one and only one of those two categories. There now exist several DFO methods that employ characteristics of both categories [CRV10, CLD13, GLD15]. Next, a brief overview of model-based methods is presented.

Model-based methods

The beginning of model-based methods seems to have occurred in 1970 when Winfield presented his Ph.D. thesis *Function and functional optimization by interpolation in data tables* [Win70]. Most of the research done on model-based methods since 1970 has assumed that the objective function is smooth and consequently, have built smooth models. Model-based methods were generally considered too computationally expensive by the community until, perhaps, the mid 1990's when Powell developed rigorous analysis for a method based on linear interpolation [Pow94].

The type of model in a model-based method may be classified into one of the following categories: *polynomial models* or *non-polynomial models*. Polynomial models are the most popular type of model when interested in a local solution [LMW19]. Essentially, a polynomial model of degree $d \geq 1$ is built by using a finite set of sample points. Typically, the polynomial model is a linear function or a quadratic function [CSV09b]. To develop convergence results for model-based methods employing polynomial models, it is necessary that the geometry of the sample sets used at some of the iterations satisfy a certain level of quality. A thorough investigation of polynomial models and their convergence results can be found in [CSV09b, Chapter 3].

The simplest type of polynomial model is linear interpolation models. This type of model requires $n+1$ sample points *poised for linear interpolation* (see [AH17, Definition 9.3]). Linear interpolation models are intimately linked to *simplex gradients* [Kel99b]. A simplex gradient is the gradient of a linear interpolation model. Simplex gradients are discussed in Chapter 4.

Quadratic polynomial models can be considered the simplest type of nonlinear model. Nonetheless, it is often the most efficient [CSV09b]. There exist three types of quadratic polynomial models: *quadratic interpolation models*, *quadratic regression models*, and *underdetermined quadratic models*. Quadratic interpolation models were used more than 50 years ago by Winfield [Win70]. This type of model require $(n+1)(n+2)/2$ sample points *poised for quadratic interpolation* (see Definition 5.1 herein). Publications on this topic include [CST97a, CST97b, CSV08a, HJB23, Pow01, Pow02, ST10]. When the function is computationally expensive to evaluate, the $(n+1)(n+2)/2$ function evaluations necessary to build a quadratic interpolation is a high price to pay. For this reason, researchers have focused on quadratic models built from fewer than $(n+1)(n+2)/2$ sample points. This type of model is generally called underdetermined quadratic models. Early works on this topic include [Pow03, Pow04a, Pow04b]. This topic is a popu-

lar area of research. Some of the works published on this topic include [CRV10, Pow04c, Pow06, Pow07, Pow08, Pow13, Wil08, Zha14]. Finally, a quadratic regression model is a type of quadratic model that can be obtained by solving a *linear least squares problem*. It is used when the set of sample points is not *affinely independent*. Some of the works published on this topic include [BLG13, CSV08b, HJBP22, VKPS17].

A non-polynomial type of model used in model-based methods is *radial basis function models*. This type of model provides an alternative approach to capture the curvature of the objective function. One advantage of radial basis function models over polynomial models is that they can be less restrictive about the geometry of the interpolation set of points. It also offers flexibility to model a wide range of non-linear behavior [LMW19]. This type of model is usually used in global DFO algorithms (see for example [RS07]). The local behavior of such models have also been studied in [Wil09, WS13].

One important category of model-based methods is *trust-region methods*. A complete analysis of trust-region methods is presented in [CGT00]. Derivative-free trust-region methods will be discussed further in Chapter 7.

A final type of method that (arguably) fits in model-based methods is *line-search-based methods* (see [AH17, CSV09b, LMW19]). This type of DFO method uses an approximate gradient to determine a descent direction of the true function and then perform a line search on that direction. Some of the early works on this topic include [DLGG84, GLL88]. Since then, convergence results have been developed in [DEMR08, GS07, GR15, LS02b] and a line-search-based method for large-scale optimization has been proposed in [NFSL11].

A sub-type of line-search-based method is *implicit filtering* [CSV09b, Chapter 9]. A DFO implicit filtering method is essentially a *grid-search algorithm* (see [AH17, Chapter 3]) combined with a *Newton-like optimization method* [LMW19]. The gradient and the Hessian are approximated using *finite-difference approximations* [BFB16]. An implicit filtering algorithm is presented in [CLPS18, Kel99b, Kel11].

A final type of method that could be classified as model-based methods is *adaptive regularized methods*. This type of DFO method have been introduced in [CGT12]. It is essentially an adaptation of the gradient-based optimization method called *adaptive cubic regularization* where the true gradient is replaced by a finite-difference approximation of the gradient [LMW19].

Note that there exist other types of model-based DFO methods that do not perfectly fit in the previous categories such as the methods in [BBN19, HL14].

Direct search methods

Initial works on direct search methods include [HJ61, SHH62, NM65] published in the 1960's. The methods developed in [SHH62, NM65] use the vertices of a simplex as the points to evaluate the objective function. The simplex may be shrunk, expanded, reflected or changed depending on the incumbent solution. This type of method is sometimes refer as a *simplex method* [LMW19]. To this day, the simplex method originally developed by Nelder and Mead is extremely popular. The simplex method is contained in the book [PTVF07] which has been cited more than 125000 times (adding the citations of all editions). The Nelder-Mead method is the foundation of the command fminsearch in Matlab [LMW19]. In 1988, Mckinnon provided an example of a smooth and convex function where the Nelder-Mead method fails to converge to a local minimum. Since then, a lot of research has been done to update the Nelder-Mead method and a stronger convergence theory has been developed [Ryk80, LRWW98, Kel99a, Tse99, PCB02, NT02, LPW12, GH12].

A second category of direct search methods is *directional methods*. Directional methods generate a finite set of points called the *poll set* near the incumbent solution x^k. The points in the poll set have the form $x^k + \alpha^k d$ where α^k is the step size parameter at iteration k and d is a direction taken from a set of directions D^k. The objective function is evaluated at the poll points and the next incumbent solution x^{k+1} is set to be one of the poll point whenever sufficient decrease in the objective function is obtained. The step size may be increased or decreased depending on the outcome of the *poll step*. One of the most basic choices for the poll directions is to take the *coordinate directions* $\pm e^i \in \mathbb{R}^n$ [LMW19]. This method is usually refer as the *coordinate search algorithm* and was introduced in 1952 in [FM52]. It is believed that the first convergence results for the coordinate search method were published in [BPC66]. Early directional methods such as coordinate search required the objective function to be continuously differentiable. This assumption became unnecessary in *pattern-search methods* developed in 1991 in [Tor91]. Convergence results for modern directional methods usually require that the set of directions used in the poll step is a *positive spanning set* of $\mathbb{R}^n$. Positive spanning sets will be discussed in Chapter 6.

In 2007, Custódio and Vicente suggested various strategies to improve the performance of directional search methods using the simplex gradient [CV07]. A year after, Custódio et al. demonstrated that the efficiency of directional methods can be improved by reordering the poll directions

according to descent indicators built from simplex gradients. Moreover, they define a new stopping criterion for direct search methods involving the simplex gradient [CDV08].

A directional method may also contained a *search step*. The search step gives the opportunity to evaluate the objective function at a finite number of points generated using a variety of strategies. For instance, the points could be randomly generated or a heuristic algorithm could be embedded in the search step to generate the candidate points [AH17]. It is believed that a search step was used for the first time in a generalized pattern search method by Audet and Dennis in [AD04]. Early convergence results of generalized pattern search methods assumed that the step size parameter α^k is a rational number [AD04]. It was later shown that this requirement can be removed by enforcing a *sufficient decrease criterion* rather than a *simple decrease criterion* when updating the incumbent solution x^k. Other works on generalized pattern search methods and their analysis include [AAD04, Abr05, AFS14, DLT03, FS07]. Extensions of generalized pattern search methods can be found in [AADLD09, FS11, GK06, HKT01]. Section 7.2 implements a generalized pattern search algorithm.

One popular extension of generalized pattern search methods was introduced in 2006 and called *mesh adaptive direct search (MADS)* [AA06, AD06]. This type of method extends generalized pattern search methods by incorporating a *mesh* and a *frame* (see [AH17, Chapter 8]). One advantage of MADS compared to a generalized pattern search method is that the sufficient decrease criteria to accept a new incumbent solution can be replaced by a simple decrease criterion. Recent research related to MADS has focused on decreasing the number of function evaluations necessary in the poll step [AAA$^+$18, AILDT14]. MADS is now supported by strong convergence results including results for the non-smooth case and certain types of discontinuous functions [VC12].

Recent convergence results for directional methods often assume that the *cosine measure* (see Definition 6.10 herein) of the set of directions D^k used in the poll step is bounded away from 0 [Vic13]. The notion of cosine measure will be discussed in Chapters 6 and 7. More details about direct search methods (and directional methods) are available in [AH17, CSV09b, KLT03, LMW19, RS13].

Now that we have established the context, fundamental results and notation necessary to understand this thesis are presented in the next chapter.

Chapter 3

Preliminaries

In this chapter, the notation used throughout this thesis is presented. In addition, fundamental results and definitions that will help the reader to understand the results of the subsequent chapters are presented.

Unless stated otherwise, the format of the notation follows [AH17]. The set of all natural numbers, $\{1, 2, \dots\}$ is denoted by $\mathbb{N}$. The set of all real numbers is denoted by $\mathbb{R}$. The set of all positive real numbers, i.e., the set $\{x \in \mathbb{R} : x > 0\}$, is denoted by $\mathbb{R}_+$. The set of all n dimensional real vectors is denoted by $\mathbb{R}^n$. Note that an element of $\mathbb{R}^n$ is represented as a column vector.

The domain of a function f is denoted by $\operatorname{dom} f$. The transpose of a matrix A is denoted by $A^\top$. The inverse of of a real square invertible matrix $A \in \mathbb{R}^{n\times n}$ is denoted by A^{-1}. For an invertible matrix, the inverse transpose is denoted by $A^{-\top} = (A^{-1})^\top$.

We work in finite-dimensional space $\mathbb{R}^n$ with inner product

$$x^\top y = \sum_{i=1}^{n} x_i y_i$$

and induced norm $\|x\| = \sqrt{x^\top x}$. The identity matrix in $\mathbb{R}^{n\times n}$ is denoted by Id_n. We use $e_n^i \in \mathbb{R}^n$ for $i \in \{1, 2, \dots, n\}$ to denote the standard unit basis vectors in $\mathbb{R}^n$, i.e., the i^{th} column of Id_n. The zero vector in $\mathbb{R}^n$ is denoted by $\mathbf{0}_n$ and the zero matrix in $\mathbb{R}^{n\times n}$ is denoted $\mathbf{0}_{n\times n}$. The vector in $\mathbb{R}^n$ with all entries equal to 1 is denoted by $\mathbf{1}_n$. When there is no ambiguity about the dimension of the vector or matrix, we may omit the subscript. The entry in the i^{th} row and j^{th} column of a matrix A is denoted by $A_{i,j}$. If the matrix involves a subscript, say k, then we write $[A_k]_{i,j}$ to denote the entry in the i^{th} row and j^{th} column of a matrix A_k.

We say that $D \in \mathbb{R}^{n\times n}$ is a *diagonal matrix* if $D_{i,j} = 0$ for all $i \neq j$, i and j in $\{1, \dots, n\}$. A diagonal matrix in $\mathbb{R}^{n\times n}$ is denoted by $\operatorname{Diag}[v] = \operatorname{Diag}\begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$ where $v \in \mathbb{R}^n$.

Given a matrix $A \in \mathbb{R}^{n\times m}$, we use the induced matrix norm

$$\|A\| = \|A\|_2 = \max\{\|Ax\|_2 : \|x\|_2 = 1\}$$

and the Frobenius norm

$$\|A\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} A_{i,j}^2 \right)^{\frac{1}{2}}.$$

In general, we use n-by-m matrices in this thesis. A column of a matrix will often represent a direction in $\mathbb{R}^n$ that is added to a point of interest x^0.

The sphere of radius $\Delta > 0$ centered at $x^0 \in \mathbb{R}^n$ is denoted by $S_n(x^0; \Delta)$. That is,

$$S_n(x^0; \Delta) = \left\{ x \in \mathbb{R}^n : \|x - x^0\| = \Delta \right\},$$

We denote by $B_n(x^0; \Delta)$ the open ball centered about $x^0 \in \mathbb{R}^n$ with radius Δ and by $\overline{B}_n(x^0; \Delta)$ the closed ball centered about x^0 with radius Δ. That is

$$B_n(x^0; \Delta) = \left\{ x \in \mathbb{R}^n : \|x - x^0\| < \Delta \right\},$$
$$\overline{B}_n(x^0; \Delta) = \left\{ x \in \mathbb{R}^n : \|x - x^0\| \leq \Delta \right\}.$$

The span of a set $\mathbb{S}$ is denoted by span($\mathbb{S}$).

The *Minkowski sum* of two sets of vectors A and B is denoted by $A \oplus B$. That is

$$A \oplus B = \{a + b : a \in A, b \in B\}.$$

The *rank* of a matrix $A \in \mathbb{R}^{n \times m}$, denoted rank A, is the maximal number of linearly independent columns of A.[2] We say a matrix $A \in \mathbb{R}^{n \times m}$ has *full rank* if and only if its rank equals the largest possible rank for a matrix of the same dimension. Furthermore, we say a matrix $A \in \mathbb{R}^{n \times m}$ is *full column rank* if and only if the columns of A are linearly independent. Similarly, we say a matrix $A \in \mathbb{R}^{n \times m}$ is *full row rank* if and only if the rows of A are linearly independent. Next we introduce fundamental definitions and results that are used in the following chapters.

When A is a real non-square matrix, that is $A \in \mathbb{R}^{n \times m}$ where $n \neq m$, we will use a generalization of the inverse matrix which is called *pseudo-inverse*. The most well-known type of matrix pseudo-inverse is the *Moore-Penrose pseudo-inverse* [HJ90].

[2] In this thesis most applications will involve $A^\top$. In order to have $A^\top \in \mathbb{R}^{m \times n}$, we make the unconventional choice of $A \in \mathbb{R}^{n \times m}$.

Definition 3.1 (Moore-Penrose pseudo-inverse)**.** [Rom07, Chapter 17]

Let $A \in \mathbb{R}^{n\times m}$. A matrix in $\mathbb{R}^{m\times n}$, denoted by $A^\dagger$, is called the Moore-Penrose pseudo-inverse of A and satisfies the following four equations:

(i) $AA^\dagger A = A$

(ii) $A^\dagger AA^\dagger = A^\dagger$

(iii) $(AA^\dagger)^\top = AA^\dagger$

(iv) $(A^\dagger A)^\top = A^\dagger A$.

Note that given $A \in \mathbb{R}^{n\times m}$, there exists a unique Moore–Penrose pseudo-inverse $A^\dagger \in \mathbb{R}^{m\times n}$. The following two properties hold [Gol12, Proposition 19.2].

(i) If $A \in \mathbb{R}^{n\times m}$ is full column rank m, then $A^\dagger$ is a left-inverse of A, that is $A^\dagger A = \mathrm{Id}_m$. In this case,

$$A^\dagger = (A^\top A)^{-1}A^\top. \tag{3.1}$$

(ii) If $A \in \mathbb{R}^{n\times m}$ is full row rank n, then $A^\dagger$ is a right-inverse of A, that is $AA^\dagger = \mathrm{Id}_n$. In this case,

$$A^\dagger = A^\top(AA^\top)^{-1}. \tag{3.2}$$

Note that from the Definition of the Moore-Penrose pseudo-inverse, we have

$$(A^\top)^\dagger = (A^\dagger)^\top.$$

The next lemma presents a result regarding the structure of the Moore-Penrose pseudo-inverse of a specific type of matrix. It will be used in Chapters 4 and 5.

Lemma 3.2. *Let* $A = \begin{bmatrix} S & -S \end{bmatrix} \in \mathbb{R}^{n\times 2m}$ *for some matrix* $S \in \mathbb{R}^{n\times m}$ *. Then*

$$A^\dagger = \frac{1}{2}\begin{bmatrix} S^\dagger \\ -S^\dagger \end{bmatrix}.$$

Proof. We have

$$A^\dagger = \begin{bmatrix} S & -S \end{bmatrix}^\dagger = \left(S \begin{bmatrix} \mathrm{Id} & -\mathrm{Id} \end{bmatrix}\right)^\dagger = \begin{bmatrix} \mathrm{Id} & -\mathrm{Id} \end{bmatrix}^\dagger S^\dagger.$$

Since $\begin{bmatrix}\mathrm{Id} & -\mathrm{Id}\end{bmatrix}$ is full row rank, using (3.2), we obtain

$$\begin{aligned} A^\dagger &= \begin{bmatrix}\mathrm{Id} & -\mathrm{Id}\end{bmatrix}^\top \left(\begin{bmatrix}\mathrm{Id} & -\mathrm{Id}\end{bmatrix}\begin{bmatrix}\mathrm{Id} & -\mathrm{Id}\end{bmatrix}^\top\right)^{-1} S^\dagger \\ &= \begin{bmatrix}\mathrm{Id}\\ -\mathrm{Id}\end{bmatrix}\left(\begin{bmatrix}\mathrm{Id} & -\mathrm{Id}\end{bmatrix}\begin{bmatrix}\mathrm{Id}\\ -\mathrm{Id}\end{bmatrix}\right)^{-1} S^\dagger = \begin{bmatrix}\mathrm{Id}\\ -\mathrm{Id}\end{bmatrix}(2\,\mathrm{Id})^{-1} S^\dagger = \frac{1}{2}\begin{bmatrix}S^\dagger\\ -S^\dagger\end{bmatrix}. \square \end{aligned}$$

Next the Sherman–Morrison–Woodbury formula for calculating matrix inverses is introduced [Bar51]. It will be useful in Sections 4.3, 5.4, and 6.2.

Proposition 3.3 (Sherman–Morrison–Woodbury formula)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a non-singular matrix and $u, v \in \mathbb{R}^n$. If $1 + v^\top A^{-1}u \neq 0$, then*

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u}. \tag{3.3}$$

We now turn our attention to some basic notation and definitions related to functions.

The notation $f \in \mathcal{C}^0$ means that f is *continuous* and $f \in \mathcal{C}^k$ means that f is k times *differentiable* and all *partial derivatives* are continuous up to the kth order. We use the term *smooth* to refer to the situation where $f \in \mathcal{C}^2$. Note that if $f \in \mathcal{C}^k$ where $k \geq 2$, then f is smooth.

We say the function f is *Lipschitz continuous* on the set $\Omega \subseteq \mathbb{R}^n$ if and only if there exists a scalar $L_f \geq 0$ for which

$$\|f(x) - f(y)\| \leq L_f\|x - y\| \quad \text{for all } x, y \in \Omega. \tag{3.4}$$

If it exists, the smallest scalar L_f satisfying Equation (3.4) is called the Lipschitz constant of f on Ω.

We define a quadratic function $Q : \mathbb{R}^n \to \mathbb{R}$ to be a function of the form $Q(x) = \alpha_0 + \alpha^\top x + \frac{1}{2}x^\top Hx$ where $\alpha_0 \in \mathbb{R}, \alpha \in \mathbb{R}^n$ and $H = H^\top \in \mathbb{R}^{n\times n}$. A linear function $L : \mathbb{R}^n \to \mathbb{R}$ is defined to be any function that can be written in the form $L(x) = \alpha_0 + \alpha^\top x$. Note that linear functions and constant functions $C(x) = \alpha_0$ are also considered quadratic functions, with $H = \mathbf{0}_{n\times n}$.

We also use the O (*big oh*) notation to describe the order at which functions converge.

Definition 3.4. [BFB16, Definition 1.19] Let $f : \mathbb{R}_+ \to \mathbb{R}^{n\times m}$ and $g : \mathbb{R}_+ \to \mathbb{R}$. Suppose that $\lim_{\Delta\to 0} g(\Delta) = 0$ and $\lim_{\Delta\to 0} f(\Delta) = L \in \mathbb{R}^{n\times m}$. If there exists a scalar $\kappa \geq 0$ with

$$\|f(\Delta) - L\| \leq \kappa\, g(\Delta) \quad \text{for sufficiently small } \Delta,$$

then we say $f(\Delta)$ is $O(g(\Delta))$, or $f(\Delta)$ is a $O(g(\Delta))$ accurate approximation of L.

In this thesis, $g(\Delta)$ takes the form $g(\Delta) = \Delta^N$, where $N \in \mathbb{N}$. If Definition 3.4 is satisfied, we will sometimes say that $f(\Delta)$ is an *order-N accurate approximation of L* where N is the greatest positive integer satisfying Definition 3.4.

Chapter 4

Gradient approximation techniques

One of the most valuable tools to solve an unconstrained optimization problem is the gradient of the objective function. Many classical optimization methods use gradient information such as *steepest descent methods*, *Newton's method*, *Quasi-Newton methods*, *inexact Newton methods*, and *conjugate gradient methods* [And22]. Many DFO methods have been inspired from classical optimization methods and adapted to the DFO context by accurately approximating gradients using only function evaluations, and then using these approximations within a (modified) classical optimization algorithms. For example, using linear interpolation on function values from $n+1$ well-poised sample points in $\mathbb{R}^n$ creates a linear model of the objective function. The gradient of this linear model is called the *simplex gradient* and it provides an approximation of the true gradient [BK98, Kel99b, Kel99a]. Simplex gradients have been used in DFO algorithms for more than 20 years. It has been used in many direct search methods and a few model-based methods such as the ones in [CV07, CDV08, GR15]. Theoretical properties and situations where they can be used are described in [BK98, CT19, CT21, DH13, HJB18, JB19, Kel99b, Reg15]. The main value of simplex gradients in model-based methods is to determine a descent direction of the true function [AH17, Chapter 10]. Even when the objective function is non-smooth, simplex gradients are well-defined and can help solve the optimization problem [CDV08].

In 2007, Custódio and Vicente suggested various strategies to improve the performance of direct serch methods using the simplex gradient [CV07]. A year after, Custódio et al. demonstrated that the efficiency of direct search methods can be improved by reordering the poll directions according to descent indicators built from simplex gradients [CDV08]. Moreover, they defined a new stopping criterion for direct search methods involving the simplex gradient.

The error bound comparing the simplex gradient and the true gradient dates back to the late 1990s and is known to be $O(\Delta)$, where Δ is the radius

of the sample set of evaluated points [Kel99b]. This error bound is critical in showing convergence of many first-order model-based methods [AH17, Chapters 10 & 11].

Simplex gradients, and their associated error bound, are not limited to the setting where exactly $n+1$ interpolation points are used in $\mathbb{R}^n$. In [CSV08a], the authors study the construction of simplex gradients consisting of $n+1$ interpolation points in $\mathbb{R}^n$, and in [CSV08b], they extend those results to the cases of fewer (underdetermined models) and more (overdetermined models) than $n+1$ points. These cases are analyzed through the use of Lagrange interpolating polynomials. Most notably, they establish error bounds for these cases and find them to be $O(\Delta)$. These results were further elaborated in [Reg15].

Many other methods of approximating gradients exist [BLG13, OB09, Pow04a, RS05, SWJ98, WS11]. One of these techniques is the *centered simplex gradient*, which is created by retaining the original points in the sample set and adding their reflection through the point of interest (see Definition 4.2). This creates an average of two simplex gradients. Interestingly, the accuracy of the centered simplex gradient is $O(\Delta^2)$ [Kel99b]. This error bound is established for the *determined case*, using exactly $2n$ function evaluations (sample points) in $\mathbb{R}^n$. In [CT21], an error bound is also established when $2n+2$ function evaluations (sample points) in $\mathbb{R}^n$ are used.

The main goals of this chapter are the following. First, we present explicit formulae based on simple matrix algebra for the simplex gradient and the centered simplex gradient that can be used for all cases: *nondetermined*, *underdetermined*, *determined* and *overdetermined* (see Definition 4.5). As long as the matrix of directions used to create the sample points is nonempty, the formulae are well-defined. These gradient approximation techniques are called the generalized simplex gradient (GSG) and the generalized centered simplex gradient (GCSG). Note that the results obtained for the GSG are intimately linked to the results found in [CSV08a, CSV08b]. One advantage of the GSG over the approach taken in the previous publications is that it provides a simple explicit formula based on matrix algebra that combines all possible type of models (underdetermined, determined and overdetermined). Hence, the GSG is well-defined regardless of the number of sample points used. Moreover, implementing the GSG in a matrix-based programming language such as Matlab is straightforward and fast.

Second, error bounds for the GSG and the GCSG that cover all possible cases are proposed. These error bounds show that the GSG is an order-1 accurate approximation of a partial gradient in the nondetermined and underdetermined cases, and an order-1 accurate approximation of the

full gradient in the determined and overdetermined cases. Similar the error bounds for the GSG, the error bounds for the GCSG show that the GCSG is an order-2 accurate approximation of a partial gradient in the nondetermined/underdetermined cases, and an order-2 accurate approximation of the full gradient in the determined and overdetermined cases.

Third, we investigate the behavior of the GSG (GCSG) and its associated error bound when the number of sample points tends to infinity in a fixed shape in $\mathbb{R}^n$. This investigation is motivated by the fact that that the number of points m appears in the error bounds and it is not clear at first glance what is the behavior of the error bounds as m tends to infinity.

The structure of this chapter is the following. In Section 4.1, we introduce the GSG and the GCSG. The relation between these two gradient approximation techniques is clarified. In Section 4.2, the GCSG and its error bound is presented. In Section 4.3, the limiting behavior of the GSG is investigated. Last, Section 4.4 summarizes the main results of this chapter and suggests future research directions.

4.1 The Generalized simplex gradient and the generalized centered simplex gradient

In this section, we begin by defining the GSG and the GCSG. Then the relation between these two approximation techniques is clarified. It is shown that the GCSG is a particular case of the GSG.

Throughout this chapter, we define $S \in \mathbb{R}^{n\times m}$ to be a set of directions written in matrix form where each column in S represents a direction in $\mathbb{R}^n$ that is added to the point of interest x^0 to form the sample points. Note that n and m can be any positive integers. The next two definitions present the GSG and the GCSG. Both gradient approximation techniques rely on the Moore-Penrose pseudo-inverse of the matrix $S^\top$ (Definition 3.1).

Definition 4.1 (Generalized simplex gradient)**.** Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$, $x^0 \in \operatorname{dom} f$ be the point of interest and $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n\times m}$. Assume that $x^0 \oplus S$ is contained in $\operatorname{dom} f$. The *generalized simplex gradient of f at x^0 over S* is denoted by $\nabla_s f(x^0; S)$ and defined by

$$\nabla_s f(x^0; S) = \left(S^\top\right)^\dagger \delta_s f(x^0; S) \in \mathbb{R}^n \tag{4.1}$$

where

$$\delta_s f(x^0; S) = \begin{bmatrix} f(x^0 + s^1) - f(x^0) \\ f(x^0 + s^2) - f(x^0) \\ \vdots \\ f(x^0 + s^m) - f(x^0) \end{bmatrix} \in \mathbb{R}^m .$$

Definition 4.2 (Generalized centered simplex gradient)**.** Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$, $x^0 \in \operatorname{dom} f$ be the point of interest and let $S \in \mathbb{R}^{n \times m}$. Assume that $x^0 \oplus (\pm S)$ is contained in $\operatorname{dom} f$. The *generalized centered simplex gradient (GCSG) of f at x^0 over S* is denoted by $\nabla_c f(x^0; S)$ and defined by

$$\nabla_c f(x^0; S) = (S^\top)^\dagger \delta_c f(x^0; S) \in \mathbb{R}^n$$

where

$$\delta_c f(x^0; S) = \frac{1}{2} \begin{bmatrix} f(x^0 + s^1) - f(x^0 - s^1) \\ \vdots \\ f(x^0 + s^m - f(x^0 - s^m) \end{bmatrix} \in \mathbb{R}^m .$$

Note that the subscript "s" in the definition of the GSG is for simplex. The subscript "c" in the definition of the GCSG is for centered. Throughout this thesis, we assume that the matrix of directions S used to compute the GSG or the GCSG is non-null rank. This guarantees that the radius of S, defined as

$$\Delta_S = \max_{j=1,\ldots,m} \|Se^j\| \tag{4.2}$$

is not equal to 0. Hence, the "normalized" matrix $\widehat{S}$, defined as

$$\widehat{S} = \frac{S}{\Delta_S}, \tag{4.3}$$

is always well-defined. This matrix will be used when defining error bounds in Section 4.2.

It is worth mentioning that the order of the columns in the matrix of directions S does not affect the value of the GSG nor the GCSG. Results on this topic can be found in [Reg15]. A simpler proof is the following. A rearrangement of the columns in S can be written as SP where $P \in \mathbb{R}^{m \times m}$ is a permutation matrix, hence an orthonormal matrix. Let us consider the

GSG. The GSG of f at x^0 over SP is

$$\begin{aligned}
&\nabla_s f(x^0; SP) \\
&= \left((SP)^\top\right)^\dagger \delta_s f(x^0; SP) \\
&= (P^\top S^\top)^\dagger P^\top \delta_s f(x^0; S) \\
&= (S^\top)^\dagger P P^\top \delta_s f(x^0; S) && \text{(since } P \text{ is an orthonormal matrix)} \\
&= (S^\top)^\dagger \operatorname{Id}_m \delta_s f(x^0; S) = \nabla_s f(x^0; S).
\end{aligned}$$

The proof for the GCSG can be done using a similar process.

Now we show that the GCSG can be written as the average of two GSGs.

Proposition 4.3. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}, x^0 \in \mathbb{R}^n$ be the point of interest and $S \in \mathbb{R}^{n\times m}$ with $x^0 \oplus (\pm S)$ contained in $\operatorname{dom} f$. Then*

$$\nabla_c f(x^0; S) = \frac{1}{2}\left(\nabla_s f(x^0; S) + \nabla_s f(x^0; -S)\right)$$

Proof. We have

$$\begin{aligned}
\nabla_c f(x^0; S) &= (S^\top)^\dagger \delta_c f(x^0; S) \\
&= (S^\top)^\dagger \frac{1}{2}\left(\delta_s f(x^0; S) - \delta_s f(x^0; -S)\right) \\
&= \frac{1}{2}\left((S^\top)^\dagger \delta_s f(x^0; S) + \left((-S)^\top\right)^\dagger \delta_s f(x^0; -S)\right) \\
&= \frac{1}{2}\left(\nabla_s f(x^0; S) + \nabla_s f(x^0; -S)\right). \qquad \square
\end{aligned}$$

The next proposition shows that the GCSG is simply a specific case of the GSG when the matrix of directions used has a particular structure.

Proposition 4.4. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}, x^0 \in \operatorname{dom} f$ be the point of interest and $S \in \mathbb{R}^{n\times m}$ with $x^0 \oplus (\pm S)$ contained in $\operatorname{dom} f$. Let $A = \begin{bmatrix} S & -S \end{bmatrix} \in \mathbb{R}^{n\times 2m}$. Then*

$$\nabla_s f(x^0; A) = \nabla_c f(x^0; S)).$$

Proof. We have

$$\begin{aligned}
\nabla_s f(x^0; A) &= (A^\top)^\dagger \delta_s f({}^0; A) \\
&= \left(\begin{bmatrix} S & -S \end{bmatrix}^\top \right)^\dagger \begin{bmatrix} \delta_s f(x^0; S) \\ \delta_s f(x^0; -S) \end{bmatrix} \\
&= \left(\begin{bmatrix} S & -S \end{bmatrix}^\dagger \right)^\top \begin{bmatrix} \delta_s f(x^0; S) \\ \delta_s (x^0; -S) \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} S^\dagger \\ -S^\dagger \end{bmatrix}^\top \begin{bmatrix} \delta_s f(x^0; S) \\ \delta_s f(x^0; -S) \end{bmatrix} \qquad \text{(by Lemma 3.2)} \\
&= \frac{1}{2} \begin{bmatrix} (S^\top)^\dagger & -(S^\top)^\dagger \end{bmatrix} \begin{bmatrix} \delta_s f(x^0; S) \\ \delta_s f(x^0; -S) \end{bmatrix} \\
&= \frac{1}{2} \left((S^\top)^\dagger \delta_s f(x^0; S) - (S^\top)^\dagger \delta_s f(x^0; -S) \right) \\
&= \frac{1}{2} \left((S^\top)^\dagger \delta_s f(x^0; S) + (-S^\top)^\dagger \delta_s (x^0; -S) \right) = \nabla_c f(x^0; S)
\end{aligned}$$

by Proposition 4.3. □

In the next section, we introduce general error bounds for the GSG and the GCSG.

4.2 Error bounds

In this section, general error bounds for the GSG and the GCSG are defined. First, we begin by defining the four possible cases to classify a GSG (GCSG).

Definition 4.5. Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ with $x^0 \oplus (\pm S)$ contained in $\operatorname{dom} f$ for all $j \in \{1, \ldots, m\}$. Assume that S is non-null rank. We define the following four cases to characterize the GSG and the GCSG.

- Underdetermined: $\nabla_s f(x^0; S)$, or $\nabla_c f(x^0; S)$, is said to be *underdetermined* if S is non-square and full column rank.
- Determined: $\nabla_s f(x^0; S)$, or $\nabla_c f(x^0; S)$, is said to be *determined* if S is square and full rank.
- Overdetermined: $\nabla_s f(x^0; S)$, or $\nabla_c f(x^0; S)$, is said to be *overdetermined* if S is non-square and full row rank.

– Nondetermined: $\nabla_s f(x^0; S)$, or $\nabla_c f(x^0; S)$, is said to be *nondetermined* if it is not in any of the previous three cases. In other words, the matrix S is neither full column rank nor full row rank.

Note that all GSGs (GSCGs) may be classified into one and only one of the previous four cases.

The definition of an underdetermined GSG (GCSG) implies that

$$\operatorname{span}(S) \neq \mathbb{R}^n,$$

which is true if and only if $SS^\dagger \neq \mathrm{Id}_n$.

When $SS^\dagger \neq \mathrm{Id}_n$, then it is not possible to define an error bound between the GSG (GCSG) and the full true gradient. The following example illustrates this claim.

Example 4.6. Let $f : \mathbb{R}^3 \to \mathbb{R} : y \mapsto ay_1 + (a+1)y_2 + (4-a)y_3$, $a \in \mathbb{R}$. Note that $\nabla f = \begin{bmatrix} a & a+1 & 4-a \end{bmatrix}^\top$. Consider the point of interest $x^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$ and the matrix of directions

$$S = \begin{bmatrix} s^1 & s^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Then regardless of the value of a, we have

$$f(x^0) = 0, \qquad f(x^0 + s^1) = a + 4 - a = 4, \qquad f(x^0 + s^2) = a + 1 + 4 - a = 5.$$

While ∇f depends upon a, the values $f(x^0), f(x^0 + s^1)$, and $f(x^0 + s^2)$ do not involve a, so it is impossible to determine ∇f using x^0 and S.

When $SS^\dagger \neq \mathrm{Id}_n$, we wish to create an approximate gradient that is accurate on the subspace generated by the columns of S. This is done by defining a projection operator.

Given a matrix of directions $S \in \mathbb{R}^{n \times m}$, the projection of $g \in \mathbb{R}^n$ onto $\operatorname{span}(S)$ is denoted by $\operatorname{Proj}_S g$ and defined by

$$\operatorname{Proj}_S g = (S^\top)^\dagger S^\top g.$$

Notice that Proj_S is a linear operator.

Next, it is shown that the projection of the GSG (GCSG) onto $\operatorname{span}(S)$ is equal to the GSG (GCSG). That is, the GSG (GCSG) is already in the subspace $\operatorname{span}(S)$.

Proposition 4.7. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$, $x^0 \in \operatorname{dom} f$ be the point of interest, $S \in \mathbb{R}^{n\times m}$ with $x^0 \oplus (\pm S)$ contained in $\operatorname{dom} f$. Then*

$$\operatorname{Proj}_S \nabla_s f(x^0; S) = \nabla_s f(x^0; S) \quad \textit{and} \quad \operatorname{Proj}_S \nabla_c f(x^0; S) = \nabla_c f(x^0; S).$$

Proof. We have

$$\begin{aligned}\operatorname{Proj}_S \nabla_s f(x^0; S) &= (S^\top)^\dagger S^\top (S^\top)^\dagger \delta_s f(x^0; S) \\ &= (S^\top)^\dagger \delta_s f(x^0; S) \qquad \text{(by Definition 3.1 (ii))} \\ &= \nabla_s f(x^0; S).\end{aligned}$$

Since Proj_S is a linear operator, we obtain

$$\begin{aligned}\operatorname{Proj}_S \nabla_c f(x^0; S) &= \operatorname{Proj}_S \left(\frac{1}{2} \left(\nabla_s f(x^0; S) + \nabla_s f(x^0; -S) \right) \right) \\ &= \frac{1}{2} \left(\operatorname{Proj}_S \nabla_s f(x^0; S) + \operatorname{Proj}_S \nabla_s f(x^0; -S) \right) \\ &= \frac{1}{2} \left(\nabla_s f(x^0; S) + \nabla_s f(x^0; -S) \right) = \nabla_c f(x^0; S). \qquad \square\end{aligned}$$

When S is not full row rank (nondetermined and underdetermined cases), then $\operatorname{Proj}_S \nabla f(x^0) \neq \nabla f(x^0)$. In this case, we may refer to $\operatorname{Proj}_S \nabla f(x^0)$ as a *partial gradient*. When S is full row rank (determined and overdetermined cases), then $(S^\top)^\dagger S^\top = \mathrm{Id}_n$ and $\operatorname{Proj}_S \nabla f(x^0) = \nabla f(x^0)$.

We are now ready to present a general error bound for the GSG and the GCSG. Note that an error bound for the determined case has been proved several times in the literature [Reg15, CSV08a, AH17, CSV09b]. An error bound for the gradient of an underdetermined model and an overdetermined model constructed through Lagrange interpolating polynomials have also been proposed in [CSV08b]. Hence, the main contribution of the error bound presented for the GSG (GCSG) is to provide one error bound that covers all four possible cases. The next theorem and lemma will be used to prove the error bounds. In this thesis, Note that the error bounds assume the function f is $\mathcal{C}^k, k \geq 2$, on an open domain containing all the sample points. This assumption could be changed to $f \in \mathcal{C}^{k-1}$ and $\nabla^{k-1} f$ is Lipschitz continuous on an open ball containing all sample points.

Theorem 4.8. [LT17, Section 4.3] *Suppose $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ is a $\mathcal{C}^3$ function on $B_n(x^0; \overline{\Delta})$. Let $L_{\nabla f} \geq 0$ and $L_{\nabla^2 f} \geq 0$ be the Lipschitz constant of ∇f and $\nabla^2 f$ on $B_n(x^0; \overline{\Delta})$, respectively. Then for $x^0 + d$ in the ball,*

$$f(x^0 + d) = f(x^0) + \nabla f(x^0)^\top d + R_1(x^0; d)$$

and

$$f(x^0+d) = f(x^0) + \nabla f(x^0)^\top d + \frac{1}{2} d^\top \nabla^2 f(x^0) d + R_2(x^0; d),$$

where

$$|R_1(x^0; d)| \leq \frac{L_{\nabla f}}{2} \|d\|^2$$

and

$$|R_2(x^0; d)| \leq \frac{L_{\nabla^2 f}}{6} \|d\|^3.$$

Lemma 4.9. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^3$ on $B_n(x^0; \overline{\Delta})$. Let $L_{\nabla f} \geq 0$ and $L_{\nabla^2 f} \geq 0$ be the Lipschitz constant of ∇f and $\nabla^2 f$ on $B_n(x^0; \overline{\Delta})$, respectively. Then for any $d \in B_n(x^0; \overline{\Delta})$, we have*

$$|f(x^0+d) - f(x^0) - \nabla f(x^0)^\top d| \leq \frac{L_{\nabla f}}{2} \|d\|^2 \tag{4.4}$$

and

$$|f(x^0+d) - f(x^0-d) - 2\nabla f(x^0)^\top d| \leq \frac{L_{\nabla^2 f}}{3} \|d\|^3. \tag{4.5}$$

Proof. Equation (4.4) follows immediately from Theorem 4.8. To show (4.5), we have

$$f(x^0+d) = f(x^0) + \nabla f(x^0)^\top d + \frac{1}{2} d^\top \nabla^2 f(x^0) d + R_2(x^0; d), \tag{4.6}$$

$$f(x^0-d) = f(x^0) - \nabla f(x^0)^\top d + \frac{1}{2} d^\top \nabla^2 f(x^0) d + R_2(x^0; -d). \tag{4.7}$$

Subtracting (4.7) from (4.6), we find

$$\begin{aligned} &f(x^0+d) - f(x^0-d) - 2\nabla f(x^0)^\top d = R_2(x^0; d) - R_2(x^0; -d) \\ \Rightarrow\ &|f(x^0+d) - f(x^0-d) - 2\nabla f(x^0)^\top d| = |R_2(x^0; d) - R_2(x^0; -d)|. \end{aligned}$$

Also from Theorem 4.8, we know that

$$|R_2(x^0; \pm d)| \leq \frac{L_{\nabla^2 f} \|d\|^3}{6}. \tag{4.8}$$

Therefore, by (4.8) and the triangle inequality, we obtain (4.5). □

Theorem 4.10 (Error bound for the GSG)**.** *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^2$ on $B_n(x^0; \overline{\Delta})$ and let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ with radius $0 < \Delta_S < \overline{\Delta}$. Denote by $L_{\nabla f} \geq 0$ the Lipschitz constant of ∇f on $\overline{B}(x^0; \Delta_S)$. Then*

$$\begin{aligned} \| \operatorname{Proj}_S \nabla_s f(x^0; S) - \operatorname{Proj}_S \nabla f(x^0) \| &= \| \nabla_s f(x^0; S) - \operatorname{Proj}_S \nabla f(x^0) \| \\ &\leq \frac{\sqrt{m}}{2} L_{\nabla f} \left\| (\widehat{S}^\top)^\dagger \right\| \Delta_S. \end{aligned} \tag{4.9}$$

Proof. The equality follows from Proposition 4.7. We have

$$\begin{aligned} \| \nabla_s f(x^0; S) - \operatorname{Proj}_S \nabla f(x^0) \| &= \| (S^\top)^\dagger \delta_s f(x^0; S) - (S^\top)^\dagger S^\top \nabla f(x^0) \| \\ &\leq \frac{1}{\Delta_S} \| (\widehat{S}^\top)^\dagger \| \| \delta_s f(x^0; S) - S^\top \nabla f(x^0) \|. \end{aligned}$$

Note that

$$\left[\delta_s f(x^0; S) - S^\top \nabla f(x^0) \right]_j = f(x^0 + s^j) - f(x^0) - (s^j)^\top \nabla f(x^0)$$

for all $j \in \{1, \ldots, m\}$. Using the bound in (4.4), we obtain

$$\begin{aligned} \| \nabla_s f(x^0; S) - \operatorname{Proj}_S \nabla f(x^0) \| &\leq \frac{1}{\Delta_S} \| (\widehat{S}^\top)^\dagger \| \sqrt{\sum_{j=1}^m \left(\frac{L_{\nabla f}}{2} \|s^j\|^2 \right)^2} \\ &\leq \frac{\sqrt{m}}{2} L_{\nabla f} \left\| (\widehat{S}^\top)^\dagger \right\| \Delta_S. \end{aligned}$$

□

The previous theorem shows that the GSG is an order-1 accurate approximation of a partial gradient when S is not full row rank and an order-1 accurate approximation of the full gradient when S is full row rank. The presence of the Lipschitz constant $L_{\nabla f}$ tells us that the GSG is a perfectly accurate approximation of $\operatorname{Proj}_S \nabla f(x^0)$ whenever the function f is linear on $\overline{B}(x^0; \Delta_S)$.

Theorem 4.11 (Error bound for the GCSG)**.** *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^3$ on $B_n(x^0; \overline{\Delta})$ and let $S \in \mathbb{R}^{n \times m}$ with radius $0 < \Delta_S < \overline{\Delta}$. Denote by $L_{\nabla^2 f} \geq 0$ the Lipschitz constant of $\nabla^2 f$ on $\overline{B}(x^0; \Delta_S)$. Then*

$$\begin{aligned} \| \operatorname{Proj}_S \nabla_c f(x^0; S) - \operatorname{Proj}_S \nabla f(x^0) \| &= \| \nabla_c f(x^0; S) - \operatorname{Proj}_S \nabla f(x^0) \| \\ &\leq \frac{\sqrt{m}}{6} L_{\nabla^2 f} \left\| (\widehat{S}^\top)^\dagger \right\| \Delta_S^2. \end{aligned} \tag{4.10}$$

Proof. The equality follows from Proposition 4.7. We have

$$\|\nabla_c f(x^0; S) - \text{Proj}_S \nabla f(x^0)\| \leq \frac{1}{\Delta_S}\|(\widehat{S}^\top)^\dagger\|\|\delta_c f(x^0; S) - S^\top \nabla f(x^0)\|.$$

Note that

$$\left[\delta_c f(x^0; S) - S^\top \nabla f(x^0)\right]_j = \frac{1}{2}\left(f(x^0 + s^j) - f(x^0 - s^j)\right) - (s^j)^\top \nabla f(x^0)$$

for all $j \in \{1, \ldots, m\}$. Using the bound in (4.5), we obtain

$$\begin{aligned}\|\nabla_c f(x^0; S) - \text{Proj}_S \nabla f(x^0)\| &\leq \frac{1}{\Delta_S}\|(\widehat{S}^\top)^\dagger\| \sqrt{\sum_{j=1}^m \left(\frac{L_{\nabla^2 f}}{6}\|s^j\|^3\right)^2} \\ &\leq \frac{\sqrt{m}}{6} L_{\nabla^2 f} \left\|(\widehat{S}^\top)^\dagger\right\| \Delta_S^2. \end{aligned}$$ □

The previous theorem shows that the GCSG is an order-2 accurate approximation of a partial gradient when S is not full row rank and an order-2 accurate approximation of the full gradient when S is full row rank. The presence of the Lipschitz constant $L_{\nabla^2 f}$ tells us that the GCSG is a perfectly accurate approximation of $\text{Proj}_S \nabla^2 f(x^0)$ whenever the function f is a polynomial of degree 2 or less on $\overline{B}(x^0; \Delta_S)$.

The main three differences between the error bound for the GSG and the error bound for the GCSG are

- The error bound for the GSG assumes that $f \in \mathcal{C}^2$ and the error bound for the GCSG assumes $f \in \mathcal{C}^3$.
- The error bound for the GSG involves the Lipschitz constant $L_{\nabla f}$ and the error bound for the GCSG involves the Lipschitz constant $L_{\nabla^2 f}$.
- The error bound for the GSG is $O(\Delta_S)$ and the error bound for the GCSG is $O(\Delta_S^2)$

In the determined case, $n + 1$ function evaluations are required to compute the GSG and obtain an order-1 accurate approximation of the full gradient. The GCSG provides an order-2 accurate approximation of the full gradient, but requires $2n$ function evaluations.

In the next section, we investigate the behavior of the GSG (GCSG) and its associate error bound as the number of sample points tends to infinity in a fixed region.

4.3 Limiting behavior of the GSG

In this chapter, we investigate the asymptotic behavior of the gradient approximation methods called GSG. This method has an error bound that at first glance seem to tend to infinity as the number of sample points increases, but with some careful construction, we show that this is not the case. We present two new error bounds ad infinitum depending on the position of the point of interest. The error bounds are not a function of the number of sample points and thus remain finite.

In Section 4.1, we saw that the GSG relaxes the requirement of evaluating exactly $n+1$ points in $\mathbb{R}^n$, making it possible to obtain an accurate approximation of the full gradient, or a partial gradient, by using $m+1$ points with an arbitrary number m (the positive integer m can be greater than, equal to, or less than n). This method has an error bound that depends on the number of points used m and the sampling radius Δ_S, which means that if one were to increase the number of sample points used and consider the limit as $m \to \infty$, then the error bound does not necessarily remain bounded. Indeed, it is possible to construct examples in which the classical error bound (Theorem 4.10) tends to infinity as m increases [BHJB21]. This is a counter-intuitive event, as it is reasonable to conjecture that more sample points in a fixed region would provide better accuracy of the model function. This problem is investigated in [BHJB21] for functions $f : \mathbb{R} \to \mathbb{R}$, in which new error bounds are developed and shown to have a more desirable behavior at the limit.

In [BHJB21], it is shown that the norm of the difference between the approximate derivative and the true derivative of a one-dimensional function has a limit that is a factor of the Lipschitz constant of ∇f and the sampling radius Δ_S. However, [BHJB21] considers only single-variable functions. In this section, we extend those results to the multivariable setting.

We first consider sampling over a hyperrectangle with the point of interest at a corner point. We explore the limiting behavior of the gradient approximation on $\mathbb{R}^n$ and calculate the limit of the corresponding error bound as the number of sample points tends to infinity. We then repeat the process considering the case where the sampling set is a ball with the point of interest at the center. From these results, it becomes clear how the method can be adapted to other shapes. We discuss the option of other shapes further in Section 4.4.

The remainder of this section is organized as follows. In Section 4.3.1, we investigate the limiting behavior of the GSG in the Cartesian setting when the sample region is a hyperrectangle, and in Section 4.3.1, we present

its error bound ad infinitum. Section 4.3.2 considers a ball as the sample region and provides an error bound ad infinitum. Illustrative examples are included throughout.

4.3.1 The GSG over a dense hyperrectangle

We now find an expression for the GSG on a hyperrectangle R as the number of points in R tends to infinity in such a way that they form a dense grid. The variable m shall be used to denote the number of sample points.

As the case $n = 1$ is covered in [BHJB21], we assume $n \geq 2$.

Preliminaries: Using the rightmost endpoints in each partition

Let $x^0 \in \mathbb{R}^n$ be the point of interest. Consider a hyperrectangular sample region with side lengths $\Delta_1, \Delta_2, \ldots, \Delta_n > 0$ where x^0 is the leftmost vertex of the hyperrectangle (the point with the lowest value for each component of all points in the hyperrectangle). We denote this hyperrectangle by $R(x^0; d)$ where $d = \begin{bmatrix}\Delta_1 & \Delta_2 & \cdots & \Delta_n\end{bmatrix}^\top$, i.e.,

$$R(x^0; d) = \{x : x = x^0 + z,\ 0 \leq z_i \leq \Delta_i\ i = 1, 2, \ldots, n\}.$$

Then $R(x^0; d)$ is partitioned into sub-hyperrectangles with lengths $\overline{\Delta}_i = \Delta_i / m_i$ where $m_i \in \mathbb{N} \setminus \{1\}$ for all $i \in \{1, 2, \ldots, n\}$. We define m to be the product of all m_i: $m = m_1 m_2 \cdots m_n$. Then S_R is defined to be a matrix in $\mathbb{R}^{n \times m}$ that contains all the directions used to form the sample points when the rightmost endpoint of each sub-hyperrectangle is chosen. The process will be generalized later in this section to allow choosing any arbitrary point in each partition of $R(x^0; d)$. Note that $R(x^0; d)$ contains m sample points and one point of interest, x^0.

Let $\overline{D} = \text{Diag}\begin{bmatrix}\overline{\Delta}_1 & \overline{\Delta}_2 & \cdots & \overline{\Delta}_n\end{bmatrix} \in \mathbb{R}^{n \times n}$. Then S_R can be written as a block matrix in the following way:

$$S_R = \begin{bmatrix}B_R^{1,1,\ldots,1,1} & B_R^{1,1,\ldots,1,2} & \cdots & B_R^{m_2, m_3, \ldots, m_{n-1}, m_n}\end{bmatrix} \in \mathbb{R}^{n \times m}, \tag{4.11}$$

where

$$\begin{aligned} B_R^{\vec{z}} &= B_R^{z_2,z_3,\ldots,z_{n-1},z_n} \\ &= \overline{D}\ddot{B}_R^{\vec{z}} \\ &= \overline{D}\begin{bmatrix} 1 & 2 & \cdots & m_1 \\ z_2 & z_2 & \cdots & z_2 \\ z_3 & z_3 & \cdots & z_3 \\ \vdots & \vdots & \ddots & \vdots \\ z_{n-1} & z_{n-1} & \cdots & z_{n-1} \\ z_n & z_n & \cdots & z_n \end{bmatrix} \in \mathbb{R}^{n\times m_1}, \end{aligned} \tag{4.12}$$

for $z_k \in \{1, 2, \ldots m_k\}$ and $k \in \{2, 3, \ldots, n\}$. The matrix S_R contains $m_2 m_3 \cdots m_n$ blocks $B_R^{\vec{z}}$, each of which contains m_1 directions. Thus, S_R contains $m_1 m_2 \cdots m_n = m$ columns in total. Note that a block is identified using a vector $\vec{z}$ containing $n-1$ components labeled $z_2, z_3, \ldots, z_n$. Next, we provide an example in $\mathbb{R}^2$ to get the reader accustomed to the notation.

Example 4.12. Select the point of interest $x^0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$ and the sample region $[0, 12] \times [0, 6]$. Then $d = [12, 6]^\top = [\Delta_1, \Delta_2]^\top$. Suppose the longer side is divided into three equal parts, so $m_1 = 3$, and the shorter side is divided into two equal parts, so $m_2 = 2$. Thus, the side lengths of one partition are $\overline{\Delta}_1 = 12/3 = 4$ and $\overline{\Delta}_2 = 6/2 = 3$. The matrix S_R is given by

$$S_R = \begin{bmatrix} B_R^1 & B_R^2 \end{bmatrix},$$

where

$$\begin{aligned} B_R^1 &= \begin{bmatrix} \frac{12}{3} & 0 \\ 0 & \frac{6}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 12 \\ 3 & 3 & 3 \end{bmatrix}, \\ B_R^2 &= \begin{bmatrix} \frac{12}{3} & 0 \\ 0 & \frac{6}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 12 \\ 6 & 6 & 6 \end{bmatrix}. \end{aligned}$$

The sample points x^j are obtained by setting $x^j = x^0 + S_R e^j$ for all $j \in \{1, 2, \ldots, 6\}$. We see that x^1, x^2, x^3 are associated with B_R^1 and x_4, x_5, x_6 are associated with B_R^2. Figure 4.1 illustrates the sample points built from S_R.

The first goal of this section is to find an expression for the GSG over $R(x_0; d)$ when the set of sample points forms a dense subset of $\mathbb{R}^n$, that is, as $m_i \to \infty$ for each i. As long as each m_i tends to infinity (not necessarily at the same speed), we will show that an expression for the GSG over $R(x^0; d)$ can be found.

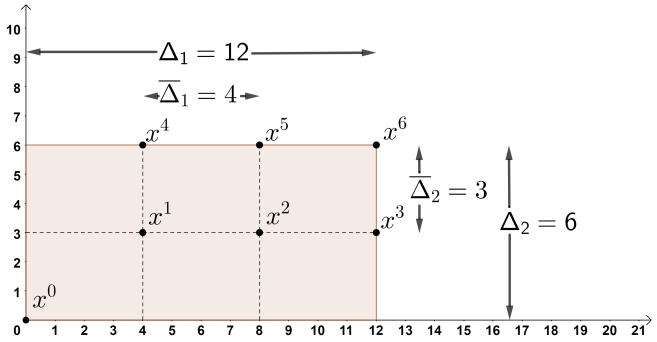


Figure 4.1: An example of a sample set built from a matrix S_R in $\mathbb{R}^2$

When S is full row rank, using (3.2) and recalling that $(S^\top)^\dagger = (S^\dagger)^\top$, the GSG is equal to

$$\nabla_s f(x^0; S) = \left(S^\top (SS^\top)^{-1}\right)^\top \delta_s f(x^0; S) = (SS^\top)^{-\top} S \delta_s f(x^0; S). \quad (4.13)$$

Note that $S_R \in \mathbb{R}^{n \times m}$ as defined in (4.11) is full row rank whenever $m_i \geq 2$ for all $i \in \{1, 2, \ldots, n\}$. For the remainder of this section and in Section 4.3.1, we will assume $m_i \geq 2$ for all i. In the following lemma, we begin our investigation of (4.13) by finding an expression for the matrix $S_R S_R^\top$.

Lemma 4.13. *Let $S_R \in \mathbb{R}^{n \times m}$ be defined as in* (4.11)*. Then*

$$S_R S_R^\top = \sum_{z_2=1}^{m_2} \sum_{z_3=1}^{m_3} \cdots \sum_{z_n=1}^{m_n} B_R^{\vec{z}} \left(B_R^{\vec{z}}\right)^\top = U_n \in \mathbb{R}^{n \times n},$$

where

$$[U_n]_{i,i} = m \frac{(m_i + 1)(2m_i + 1)}{6} \overline{\Delta}_i^2, \quad i \in \{1, 2, \ldots, n\},$$
$$[U_n]_{i,j} = m \frac{(m_i + 1)(m_j + 1)}{4} \overline{\Delta}_i \overline{\Delta}_j, \quad i, j \in \{1, 2, \ldots, n\},\ i \neq j.$$

Proof. The proof is by induction on n. First, we prove the case $n = 2$. We

have

$$\begin{aligned}
&\sum_{z_2=1}^{m_2} B_R^{z_2}\left(B_R^{z_2}\right)^\top \\
&= \begin{bmatrix}\overline{\Delta}_1 & 0\\ 0 & \overline{\Delta}_2\end{bmatrix}\left(\sum_{z_2=1}^{m_2}\begin{bmatrix}1 & 2 & \cdots & m_1\\ z_2 & z_2 & \cdots & z_2\end{bmatrix}\begin{bmatrix}1 & z_2\\ 2 & z_2\\ \vdots & \vdots\\ m_1 & z_2\end{bmatrix}\right)\begin{bmatrix}\overline{\Delta}_1 & 0\\ 0 & \overline{\Delta}_2\end{bmatrix}\\
&= \begin{bmatrix}\overline{\Delta}_1 & 0\\ 0 & \overline{\Delta}_2\end{bmatrix}\left(\sum_{z_2=1}^{m_2}\begin{bmatrix}\frac{m_1(m_1+1)(2m_1+1)}{6} & z_2\frac{m_1(m_1+1)}{2}\\ z_2\frac{m_1(m_1+1)}{2} & z_2^2 m_1\end{bmatrix}\right)\begin{bmatrix}\overline{\Delta}_1 & 0\\ 0 & \overline{\Delta}_2\end{bmatrix}\\
&= \begin{bmatrix}\overline{\Delta}_1 & 0\\ 0 & \overline{\Delta}_2\end{bmatrix}\begin{bmatrix}\frac{m_1 m_2(m_1+1)(2m_1+1)}{6} & \frac{m_1 m_2(m_1+1)(m_2+1)}{4}\\ \frac{m_1 m_2(m_1+1)(m_2+1)}{4} & \frac{m_1 m_2(m_2+1)(2m_2+1)}{6}\end{bmatrix}\begin{bmatrix}\overline{\Delta}_1 & 0\\ 0 & \overline{\Delta}_2\end{bmatrix}\\
&= m_1 m_2\begin{bmatrix}\frac{(m_1+1)(2m_1+1)}{6}\overline{\Delta}_1^2 & \frac{(m_1+1)(m_2+1)}{4}\overline{\Delta}_1\overline{\Delta}_2\\ \frac{(m_1+1)(m_2+1)}{4}\overline{\Delta}_1\overline{\Delta}_2 & \frac{(m_2+1)(2m_2+1)}{6}\overline{\Delta}_2^2\end{bmatrix} = U_2.
\end{aligned}$$

Now, let $n = k$ for some $k \in \mathbb{N}\backslash\{1\}$. We will write $B_R^{\vec{z_k}}$ instead of $B_R^{\vec{z}}$ to make it clear that the last component in the vector of indices $\vec{z}$ is z_k. Suppose that the induction hypothesis is true for $n = k$. We want to show that it is true for $n = k + 1$. We have

$$\sum_{z_2=1}^{m_2}\sum_{z_3=1}^{m_3}\cdots\sum_{z_{k+1}=1}^{m_{k+1}} B_R^{\overrightarrow{z_{k+1}}}\left(B_R^{\overrightarrow{z_{k+1}}}\right)^\top \tag{4.14}$$

$$= \sum_{z_2=1}^{m_2}\sum_{z_3=1}^{m_3}\cdots\sum_{z_{k+1}=1}^{m_{k+1}}\begin{bmatrix}B_R^{\vec{z_k}}\\ \overline{\Delta}_{k+1}z_{k+1}\mathbf{1}_{m_1}^\top\end{bmatrix}\begin{bmatrix}\left(B_R^{\vec{z_k}}\right)^\top & \overline{\Delta}_{k+1}z_{k+1}\mathbf{1}_{m_1}\end{bmatrix}$$

$$= \sum_{z_2=1}^{m_2}\sum_{z_3=1}^{m_3}\cdots\sum_{z_{k+1}=1}^{m_{k+1}}\begin{bmatrix}B_R^{\vec{z_k}}(B_R^{\vec{z_k}})^\top & \overline{\Delta}_{k+1}z_{k+1}B_R^{\vec{z_k}}\mathbf{1}_{m_1}\\ \overline{\Delta}_{k+1}z_{k+1}\mathbf{1}_{m_1}^\top\left(B_R^{\vec{z_k}}\right)^\top & m_1 z_{k+1}^2\overline{\Delta}_{k+1}^2\end{bmatrix}. \tag{4.15}$$

Now, we compute the last sum of the k-tuple sum. We have

$$\begin{aligned}
&\sum_{z_{k+1}=1}^{m_{k+1}}\begin{bmatrix}B_R^{\vec{z_k}}(B_R^{\vec{z_k}})^\top & \overline{\Delta}_{k+1}z_{k+1}B_R^{\vec{z_k}}\mathbf{1}_{m_1}\\ \overline{\Delta}_{k+1}z_{k+1}\mathbf{1}_{m_1}^\top\left(B_R^{\vec{z_k}}\right)^\top & m_1 z_{k+1}^2\overline{\Delta}_{k+1}^2\end{bmatrix}\\
&= \begin{bmatrix}m_{k+1}B_R^{\vec{z_k}}\left(B_R^{\vec{z_k}}\right)^\top & \overline{\Delta}_{k+1}\frac{m_{k+1}(m_{k+1}+1)}{2}B_R^{\vec{z_k}}\mathbf{1}_{m_1}\\ \overline{\Delta}_{k+1}\frac{m_{k+1}(m_{k+1}+1)}{2}\mathbf{1}_{m_1}^\top\left(B_R^{\vec{z_k}}\right)^\top & m_1\overline{\Delta}_{k+1}^2\frac{m_{k+1}(m_{k+1}+1)(2m_{k+1}+1)}{6}\end{bmatrix}.
\end{aligned} \tag{4.16}$$

Substituting (4.16) into (4.15), we obtain

$$\begin{aligned}
&\sum_{z_2=1}^{m_2}\cdots\sum_{z_{k+1}=1}^{m_{k+1}} B_R^{\overrightarrow{z_{k+1}}}\left(B_R^{\overrightarrow{z_{k+1}}}\right)^\top \\
&=\sum_{z_2=1}^{m_2}\cdots\sum_{z_k=1}^{m_k}\begin{bmatrix} m_{k+1}B_R^{\vec{z_k}}\left(B_R^{\vec{z_k}}\right)^\top & \overline{\Delta}_{k+1}\frac{m_{k+1}(m_{k+1}+1)}{2}B_R^{\vec{z_k}}\mathbf{1}_{m_1} \\ \overline{\Delta}_{k+1}\frac{m_{k+1}(m_{k+1}+1)}{2}\mathbf{1}_{m_1}^\top\left(B_R^{\vec{z_k}}\right)^\top & \overline{\Delta}_{k+1}^2 m_1\frac{m_{k+1}(m_{k+1}+1)(2m_{k+1}+1)}{6}\end{bmatrix}.
\end{aligned} \tag{4.17}$$

Let us find an expression for each of the four blocks in (4.17). Using the induction assumption, we have

$$\sum_{z_2=1}^{m_2}\sum_{z_3=1}^{m_3}\cdots\sum_{z_k}^{m_k} m_{k+1}B_R^{\vec{z_k}}\left(B_R^{\vec{z_k}}\right)=m_{k+1}U_k.$$

The off-diagonal blocks are transposes of each other, with

$$\begin{aligned}
&\sum_{z_2=1}^{m_2}\sum_{z_3=1}^{m_3}\cdots\sum_{z_k=1}^{m_k}\overline{\Delta}_{k+1}\frac{m_{k+1}(m_{k+1}+1)}{2}B_R^{\vec{z_k}}\mathbf{1}_{m_1} \qquad (4.18)\\
&=\overline{\Delta}_{k+1}\frac{m_{k+1}(m_{k+1}+1)}{2}\begin{bmatrix}\overline{\Delta}_1 m_1\frac{(m_1+1)}{2}m_2m_3\cdots m_k\\ \overline{\Delta}_2 m_1m_2\frac{(m_2+1)}{2}m_3m_4\cdots m_k\\ \vdots\\ \overline{\Delta}_k m_1m_2\cdots m_{k-1}m_k\frac{(m_k+1)}{2}\end{bmatrix}\\
&=\begin{bmatrix} m\overline{\Delta}_1\overline{\Delta}_{k+1}\frac{(m_1+1)(m_{k+1}+1)}{4}\\ m\overline{\Delta}_2\overline{\Delta}_{k+1}\frac{(m_2+1)(m_{k+1}+1)}{4}\\ \vdots\\ m\overline{\Delta}_k\overline{\Delta}_{k+1}\frac{(m_k+1)(m_{k+1}+1)}{4}\end{bmatrix}.
\end{aligned}$$

The last block is

$$\begin{aligned}
&\sum_{z_2=1}^{m_2}\sum_{z_3=1}^{m_3}\cdots\sum_{z_k=1}^{m_k}\overline{\Delta}_{k+1}^2 m_1\frac{m_{k+1}(m_{k+1}+1)(2m_{k+1}+1)}{6}\\
&=m\overline{\Delta}_{k+1}^2\frac{(m_{k+1}+1)(2m_{k+1}+1)}{6}.
\end{aligned}$$

Hence, (4.17) is equal to U_{k+1}. By the principle of induction, the claim is true for all $n\in\mathbb{N}\setminus\{1\}$. □

The GSG uses the inverse of $(S_R S_R^\top)^\top$. The following lemma finds an expression for this inverse.

Lemma 4.14. *Let $S_R \in \mathbb{R}^{n\times m}$ be defined as in (4.11). Then*

$$(S_R S_R^\top)^{-\top} = \frac{12}{m}\left(E - \frac{3}{1+3s} yy^\top\right),$$

where

$$E = \operatorname{Diag}\left[\frac{1}{(m_1^2-1)\overline{\Delta}_1^2} \quad \cdots \quad \frac{1}{(m_n^2-1)\overline{\Delta}_n^2}\right] \in \mathbb{R}^{n\times n},\ s = \sum_{i=1}^{n} \frac{m_i+1}{m_i-1},$$

and

$$y = \left[\frac{1}{(m_1-1)\overline{\Delta}_1} \quad \cdots \quad \frac{1}{(m_n-1)\overline{\Delta}_n}\right]^\top \in \mathbb{R}^n.$$

Proof. Note that for all $n \in \mathbb{N}\setminus\{1\}$ the symmetric matrix U_n can be written as

$$U_n = m\left(\operatorname{Diag}\left[\frac{(m_1-1)(m_1+1)}{12}\overline{\Delta}_1^2 \quad \cdots \quad \frac{(m_n-1)(m_n+1)}{12}\overline{\Delta}_n^2\right] + \begin{bmatrix} \frac{(m_1+1)}{2}\overline{\Delta}_1 \\ \vdots \\ \frac{(m_n+1)}{2}\overline{\Delta}_n \end{bmatrix} \left[\frac{(m_1+1)}{2}\overline{\Delta}_1 \quad \cdots \quad \frac{(m_n+1)}{2}\overline{\Delta}_n\right]\right).$$

Let $\dot{d} = \left[\frac{(m_1+1)}{2}\overline{\Delta}_1 \quad \cdots \quad \frac{(m_n+1)}{2}\overline{\Delta}_n\right]^\top \in \mathbb{R}^n$ and

$$\widetilde{D} = \operatorname{Diag}\left[\frac{m_1^2-1}{12}\overline{\Delta}_1^2 \quad \cdots \quad \frac{m_n^2-1}{12}\overline{\Delta}_n^2\right] \in \mathbb{R}^{n\times n}.$$

Then

$$U_n = m(\widetilde{D} + \dot{d}\,\dot{d}^\top).$$

Therefore, using (3.3), we obtain

$$\begin{aligned}(U_n)^{-\top} &= (U_n^\top)^{-1} = U_n^{-1} \\ &= \frac{1}{m}\left(\widetilde{D}^{-1} - \frac{\widetilde{D}^{-1}\dot{d}\,\dot{d}^\top\widetilde{D}^{-1}}{1+\dot{d}^\top\widetilde{D}^{-1}\dot{d}}\right). \end{aligned} \tag{4.19}$$

The denominator of the second term in (4.19) is

$$
\begin{aligned}
&1 + \dot{d}^\top \widetilde{D}^{-1} \dot{d} \\
&= 1 + \begin{bmatrix} \frac{(m_1+1)}{2}\overline{\Delta}_1 & \cdots & \frac{(m_n+1)}{2}\overline{\Delta}_n \end{bmatrix} \operatorname{Diag}\begin{bmatrix} \frac{12}{(m_1^2-1)\overline{\Delta}_1^2} & \cdots & \frac{12}{(m_n^2-1)\overline{\Delta}_n^2} \end{bmatrix} \begin{bmatrix} \frac{(m_1+1)}{2}\overline{\Delta}_1 \\ \vdots \\ \frac{(m_n+1)}{2}\overline{\Delta}_n \end{bmatrix} \\
&= 1 + \begin{bmatrix} \frac{6}{(m_1-1)\overline{\Delta}_1} & \cdots & \frac{6}{(m_n-1)\overline{\Delta}_n} \end{bmatrix} \begin{bmatrix} \frac{(m_1+1)}{2}\overline{\Delta}_1 \\ \vdots \\ \frac{(m_n+1)}{2}\overline{\Delta}_n \end{bmatrix} \\
&= 1 + 3\sum_{i=1}^{n} \frac{m_i+1}{m_i-1} = 1 + 3s.
\end{aligned}
$$

The numerator of the second term in (4.19) is

$$
\widetilde{D}^{-1}\dot{d}\,\dot{d}^\top\widetilde{D}^{-1} = \begin{bmatrix} \frac{6}{(m_1-1)\overline{\Delta}_1} \\ \vdots \\ \frac{6}{(m_n-1)\overline{\Delta}_n} \end{bmatrix} \begin{bmatrix} \frac{6}{(m_1-1)\overline{\Delta}_1} & \cdots & \frac{6}{(m_n-1)\overline{\Delta}_n} \end{bmatrix} = 36yy^\top.
$$

Since $\widetilde{D}^{-1} = 12E$, we get

$$
(U_n)^\top = \frac{1}{m}\left(\widetilde{D}^{-1} - \frac{36}{1+3s}yy^\top\right) = \frac{12}{m}\left(E - \frac{3}{1+3s}yy^\top\right). \qquad \square
$$

Using the previous lemma, we can now provide an expression for the transpose of the Moore–Penrose inverse $S_R^\dagger$.

Corollary 4.15 (The matrix $(S_R^\dagger)^\top$)**.** *Let $S_R \in \mathbb{R}^{n\times m}$ be defined as in* (4.11)*. Then*

$$
(S_R^\dagger)^\top = \frac{12}{m}\left(E - \frac{3}{1+3s}yy^\top\right) S_R,
$$

where

$$
E = \operatorname{Diag}\begin{bmatrix} \frac{1}{(m_1^2-1)\overline{\Delta}_1^2} & \cdots & \frac{1}{(m_n^2-1)\overline{\Delta}_n^2} \end{bmatrix} \in \mathbb{R}^{n\times n},
$$

$y = \begin{bmatrix} \frac{1}{(m_1-1)\overline{\Delta}_1} & \cdots & \frac{1}{(m_n-1)\overline{\Delta}_n} \end{bmatrix}^\top \in \mathbb{R}^n$, *and the scalar* $s = \sum_{i=1}^n \frac{m_i+1}{m_i-1}$.

Proof. Since S_R has full row rank, we have

$$
(S_R^\dagger)^\top = \left(S_R^\top (S_R S_R^\top)^{-1}\right)^\top = (S_R S_R)^{-\top} S_R = \frac{12}{m}\left(E - \frac{3}{1+3s}yy^\top\right) S_R,
$$

by Lemma 4.14. $\square$

In the following proposition, we investigate the limit of $m(S_R S_R^\top)^{-\top}$ when all m_i go to infinity. Applying these limits, the sample points contained in $R(x_0; d)$ form a dense grid of $\mathbb{R}^n$. To make the notation compact, we write $\lim_{\vec{m}\to\vec{\infty}}$ to represent the limit as $m_1 \to \infty, m_2 \to \infty, \ldots, m_n \to \infty$. The reason we are interested in the limit of $m(S_R S_R^\top)^{-\top}$ is that, assuming the limits exist, we may write

$$\begin{aligned}\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S_R) &= \lim_{\vec{m}\to\vec{\infty}} (S_R^\dagger)^\top \delta_s f(x^0; S_R) \\ &= \lim_{\vec{m}\to\vec{\infty}} \frac{m}{\Delta} (S_R S_R^\top)^{-\top} \frac{\Delta}{m} S_R \delta_s f(x^0; S_R) \\ &= \frac{1}{\Delta} \left[\lim_{\vec{m}\to\vec{\infty}} m(S_R S_R^\top)^{-\top}\right] \left[\lim_{\vec{m}\to\vec{\infty}} \frac{\Delta}{m} S_R \delta_s f(x^0; S_R)\right], \end{aligned} \tag{4.20}$$

where $\Delta = \Delta_1 \Delta_2 \cdots \Delta_n$. So if we can show that the two limits in (4.20) exist, then we have found the limit of the GSG over a dense hyperrectangle. Note that the term Δ remains inside the second limit, as we will show that the expression in the second limit is a n-tuple Riemann sum.

Proposition 4.16. *Let S_R be defined as in* (4.11)*. Then*

$$\lim_{\vec{m}\to\vec{\infty}} m(S_R S_R^\top)^{-\top} = L_n \in \mathbb{R}^{n\times n}$$

where

$$\begin{aligned}[L_n]_{i,i} &= \frac{12(3n-2)}{\Delta_i^2(3n+1)}, \quad i \in \{1, 2, \ldots, n\}, \\ [L_n]_{i,j} &= \frac{-36}{\Delta_i \Delta_j (3n+1)}, \quad i, j \in \{1, 2, \ldots, n\}, i \neq j.\end{aligned}$$

Proof. By Lemma 4.14, we have

$$\begin{aligned}\lim_{\vec{m}\to\vec{\infty}} m(S_R S_R^\top)^{-\top} &= \lim_{\vec{m}\to\vec{\infty}} m\frac{12}{m}\left(E - \frac{3}{1+3s} yy^\top\right) \\ &= \lim_{\vec{m}\to\vec{\infty}} 12\left(E - \frac{3}{1+3s} yy^\top\right), \end{aligned} \tag{4.21}$$

where

$$\begin{aligned} E &= \operatorname{Diag}\left[\tfrac{1}{(m_1^2-1)\overline{\Delta}_1^2} \quad \cdots \quad \tfrac{1}{(m_n^2-1)\overline{\Delta}_n^2}\right] \in \mathbb{R}^{n\times n}, \\ y &= \left[\tfrac{1}{(m_1-1)\overline{\Delta}_1} \quad \cdots \quad \tfrac{1}{(m_n-1)\overline{\Delta}_n}\right]^\top \in \mathbb{R}^n \\ s &= \sum_{i=1}^n \frac{m_i+1}{m_i-1} \in \mathbb{R}\,. \end{aligned}$$

We show that this converges component-wise to L.

We begin with the diagonal entries. Applying $\overline{\Delta}_i^2 = \Delta_i^2/m_i^2$, note that

$$[yy^\top]_{i,i} = \frac{1}{(m_i-1)^2\overline{\Delta}_i^2} = \frac{m_i^2}{(m_i-1)^2\Delta_i^2}.$$

Substituting this and $\overline{\Delta}_i^2 = \Delta_i^2/m_i^2$ into the definition of E yields

$$\begin{aligned}
&\lim_{\vec{m}\to\vec{\infty}} \left[12\left(E - \frac{3}{1+3s}yy^\top\right)\right]_{i,i} \\
&= 12 \lim_{\vec{m}\to\vec{\infty}} \left(\frac{m_i^2}{(m_i-1)(m_i+1)\Delta_i^2} - \frac{3}{1+3\sum_{i=1}^n \frac{m_i+1}{m_i-1}} \frac{m_i^2}{(m_i-1)^2\Delta_i^2}\right) \\
&= 12\left(\frac{1}{\Delta_i^2} - \frac{3}{(1+3n)\Delta_i^2}\right) \\
&= \frac{12(3n-2)}{\Delta_i^2(3n+1)}, \quad i \in \{1,2,\ldots,n\}.
\end{aligned}$$

Similarly, the off-diagonal entries of the matrix in (4.21) are given by

$$\lim_{\vec{m}\to\vec{\infty}} \left[\frac{-36}{1+3\sum_{i=1}^n \frac{m_i+1}{m_i-1}} \frac{m_i m_j}{(m_i-1)(m_j-1)\Delta_i\Delta_j}\right]_{i,j} = \frac{-36}{(1+3n)\Delta_i\Delta_j},$$

for $i,j \in \{1,2,\ldots,n\}, i \neq j$. □

Generalization: using an arbitrary point in each partition

We now generalize S_R to a matrix that allows choosing an arbitrary point in each partition of the sample region, not necessarily the right endpoint of each partition. The matrix containing all directions used to obtain an arbitrary sample point in each partition will be denoted by S. Let $m = m_1 m_2 \cdots m_n$ and $\overline{D} = \text{Diag}\begin{bmatrix}\overline{\Delta}_1 & \overline{\Delta}_2 & \cdots & \overline{\Delta}_n\end{bmatrix} \in \mathbb{R}^{n\times n}$ where $\overline{\Delta}_i = \Delta_i/m_i$ for all i. The matrix $S \in \mathbb{R}^{n\times m}$ can be written as a block matrix in the following way:

$$S = \begin{bmatrix} B^{1,1,\ldots,1,1} & B^{1,1,\ldots,1,2} & \cdots & B^{m_2,m_3,\ldots,m_{n-1},m_n}\end{bmatrix} \tag{4.22}$$

where

$$B^{\vec{z}} = B_R^{\vec{z}} - B_M^{\vec{z}} \in \mathbb{R}^{n\times m_1}.$$

The block $B_R^{\vec{z}} \in \mathbb{R}^{n \times m_1}$ is defined in (4.12) and the block $B_M^{\vec{z}} \in \mathbb{R}^{n \times m_1}$ is

$$B_M^{\vec{z}} = \overline{D} \ddot{B}_M^{\vec{z}}$$

where all entries of $\ddot{B}_M^{\vec{z}}$ are in $[0, 1]$. Let $S_M \in \mathbb{R}^{n \times m}$ be defined as

$$S_M = \begin{bmatrix} B_M^{1,1,\ldots,1,1} & B_M^{1,1,\ldots,1,2} & \cdots & B_M^{m_2,m_3,\ldots,m_{n-1},m_n} \end{bmatrix}.$$

Then S can be written as $S = S_R - S_M$. Let us provide an example of S in $\mathbb{R}^2$.

Example 4.17. Consider the same sample region as Example 4.12. Select the 'arbitrary' points $[2, 2]^\top$, $[5, 1]^\top$, $[8, 3]^\top$, $[1, 1]^\top$, $[6, 4.5]^\top$, and $[12, 6]^\top$. These are visualized in Figure 4.2.

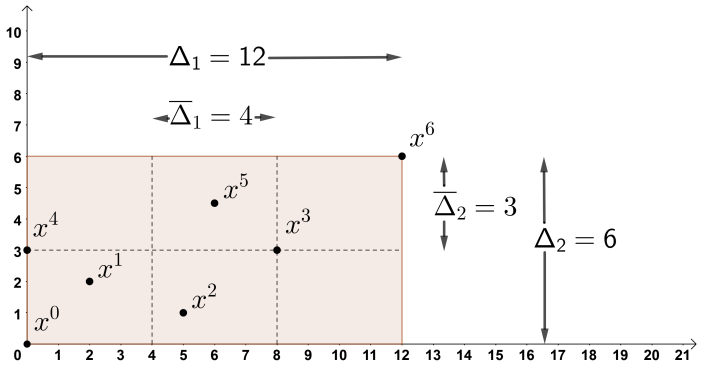


Figure 4.2: An example of sample set built from a matrix S in $\mathbb{R}^2$

The matrix $S \in \mathbb{R}^{2 \times 6}$ is given by

$$S = \begin{bmatrix} B^1 & B^2 \end{bmatrix},$$

where

$$\begin{aligned} B^1 &= B_R^1 - B_M^1 \\ &= \begin{bmatrix} \frac{12}{3} & 0 \\ 0 & \frac{6}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} \frac{12}{3} & 0 \\ 0 & \frac{6}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{3}{4} & 1 \\ \frac{1}{3} & \frac{2}{3} & 0 \end{bmatrix} \\ &= \begin{bmatrix} 4 & 8 & 12 \\ 3 & 3 & 3 \end{bmatrix} - \begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 5 & 8 \\ 2 & 1 & 3 \end{bmatrix}, \end{aligned}$$

and

$$\begin{aligned} B^2 &= B_R^2 - B_M^2 \\ &= \begin{bmatrix} \frac{12}{3} & 0 \\ 0 & \frac{6}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \end{bmatrix} - \begin{bmatrix} \frac{12}{3} & 0 \\ 0 & \frac{6}{2} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \end{bmatrix} \\ &= \begin{bmatrix} 4 & 8 & 12 \\ 6 & 6 & 6 \end{bmatrix} - \begin{bmatrix} 4 & 2 & 0 \\ 3 & \frac{3}{2} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 6 & 12 \\ 3 & \frac{9}{2} & 6 \end{bmatrix}. \end{aligned}$$

The sample points x^j are built by setting $x^j = x^0 + Se^j$ for $j \in \{1, 2, \ldots, 6\}$. We see that x^1, x^2, x^3 are associated with B^1 and x^4, x^5, x^6 are associated with B^2.

The following proposition generalizes Proposition 4.16 by considering S in place of S_R.

Proposition 4.18. *Let $S \in \mathbb{R}^{n \times m}$ be defined as in* (4.22)*. Then*

$$\lim_{\vec{m} \to \vec{\infty}} m(SS^\top)^{-\top} = L_n \in \mathbb{R}^{n \times n},$$

where L_n is defined as in Proposition 4.16.

Proof. We have

$$\lim_{\vec{m} \to \vec{\infty}} m(SS^\top)^{-\top} = \lim_{\vec{m} \to \vec{\infty}} \left(\frac{1}{m} SS^\top \right)^{-\top}.$$

Note that the inverse of $SS^\top$ is well-defined, since S is full row rank whenever $m_i \in \mathbb{N} \setminus \{1\}$ for all i. It follows that $SS^\top$ is full rank, so the inverse of $(SS^\top)^\top$ exists. Since the inverse of $(SS^\top)^\top$ is a continuous function with respect to $\vec{m}$, we may take the limit inside the inverse. We obtain

$$\lim_{\vec{m} \to \vec{\infty}} \left(\frac{1}{m} SS^\top \right)^{-\top} \tag{4.23}$$

$$= \left(\lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} SS^\top \right)^{-\top}$$

$$= \left(\lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} S_R S_R^\top - \lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} S_M S_R^\top - \lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} S_R S_M^\top + \lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} S_M S_M^\top \right)^{-\top}. \tag{4.24}$$

Now, we show $\lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} S_M S_R^\top, \lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} S_R S_M^\top$, and $\lim_{\vec{m} \to \vec{\infty}} \frac{1}{m} S_M S_M^\top$ are equal to the $n \times n$ zero matrix.

We begin with showing $\lim_{\vec{m}\to\vec{\infty}} \frac{1}{m} S_M S_M^\top = \mathbf{0}_{n\times n}$. Let $\overline{D} \in \mathbb{R}^{n\times n}$ be the diagonal matrix with entries $\Delta_1/m_1, \ldots, \Delta_n/m_n$. We have

$$\begin{aligned}\frac{1}{m} S_M S_M^\top &= \frac{1}{m} \sum_{z_2=1}^{m_2} \cdots \sum_{z_n=1}^{m_n} B_M^{\vec{z}} B_M^{\vec{z}} \\ &= \frac{1}{m} \overline{D} \left(\sum_{z_2=1}^{m_2} \cdots \sum_{z_n=1}^{m_n} \ddot{B}_M^{\vec{z}} (\ddot{B}_M^{\vec{z}})^\top \right) \overline{D}. \end{aligned} \tag{4.25}$$

Since all entries in the matrix $\ddot{B}_M^{\vec{z}}$ are contained in $[0,1]$, the $(n-1)$-tuple sum in (4.25) is bounded component-wise below by the matrix $\mathbf{0}_{n\times n}$ and above by

$$\begin{aligned}\sum_{z_2=1}^{m_2} \cdots \sum_{z_n=1}^{m_n} \ddot{B}_M^{\vec{z}} (\ddot{B}_M^{\vec{z}})^\top &\le \sum_{z_2=1}^{m_2} \cdots \sum_{z_n=1}^{m_n} \mathbf{1}_n \mathbf{1}_{m_1}^\top \mathbf{1}_{m_1} \mathbf{1}_n^\top \\ &= m_1 \sum_{z_2=1}^{m_2} \cdots \sum_{z_n=1}^{m_n} \mathbf{1}_n \mathbf{1}_n^\top = m \mathbf{1}_n \mathbf{1}_n^\top. \end{aligned}$$

It follows that component-wise

$$\begin{aligned}&\lim_{\vec{m}\to\vec{\infty}} \frac{1}{m} \overline{D} \left(\sum_{z_2=1}^{m_2} \cdots \sum_{z_n=1}^{m_n} \ddot{B}_M^{\vec{z}} (\ddot{B}_M^{\vec{z}})^\top \right) \overline{D} \\ &\le \lim_{\vec{m}\to\vec{\infty}} \frac{1}{m} \overline{D} m \mathbf{1}_n \mathbf{1}_n^\top \overline{D} = \lim_{\vec{m}\to\vec{\infty}} \begin{bmatrix} \frac{\Delta_1}{m_1} \\ \frac{\Delta_2}{m_2} \\ \vdots \\ \frac{\Delta_n}{m_n} \end{bmatrix} \begin{bmatrix} \frac{\Delta_1}{m_1} & \frac{\Delta_2}{m_2} & \cdots & \frac{\Delta_n}{m_n} \end{bmatrix} = \mathbf{0}_{n\times n}. \end{aligned}$$

By the Squeeze Theorem, $\lim_{\vec{m}\to\vec{\infty}} \frac{1}{m} S_M S_M^\top = \mathbf{0}_{n\times n}$.

Now, we show that $\lim_{\vec{m}\to\vec{\infty}} \frac{1}{m} S_R S_M^\top = \mathbf{0}_{n\times n}$. We have

$$\frac{1}{m} S_R S_M^\top = \frac{1}{m} \overline{D} \left(\sum_{z_2=1}^{m_2} \cdots \sum_{z_n=1}^{m_n} \ddot{B}_R^{\vec{z}} (\ddot{B}_M^{\vec{z}})^\top \right) \overline{D}. \tag{4.26}$$

The $(n-1)$-tuple sum in (4.26) is bounded component-wise below by $\mathbf{0}_{n\times n}$.

A component-wise upper bound for (4.26) is

$$\begin{aligned}
&\frac{1}{m}\overline{D}\left(\sum_{z_2=1}^{m_2}\cdots\sum_{z_n=1}^{m_n}\ddot{B}_R^{\vec{z}}(\ddot{B}_M^{\vec{z}})^\top\right)\overline{D}\\
&\leq \frac{1}{m}\overline{D}\left(\sum_{z_2=1}^{m_2}\cdots\sum_{z_n=1}^{m_n}\ddot{B}_R^{\vec{z}}\mathbf{1}_n\mathbf{1}_{m-1}^\top\right)\overline{D}\\
&= \frac{1}{m}\overline{D}\frac{m}{2}\begin{bmatrix} m_1+1 & m_1+1 & \cdots & m_1+1\\ m_2+1 & m_2+1 & \cdots & m_2+1\\ \vdots & \vdots & \ddots & \vdots\\ m_n+1 & m_n+1 & \cdots & m_n+1\end{bmatrix}\overline{D}\\
&= \frac{1}{2}\begin{bmatrix} \frac{\Delta_1^2(m_1+1)}{m_1^2} & \frac{\Delta_1\Delta_2(m_1+1)}{m_1m_2} & \cdots & \frac{\Delta_1\Delta_n(m_1+1)}{m_1m_n}\\ \frac{\Delta_1\Delta_2(m_2+1)}{m_1m_2} & \frac{\Delta_2^2(m_2+1)}{m_2^2} & \cdots & \frac{\Delta_2\Delta_n(m_2+1)}{m_2m_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\Delta_1\Delta_n(m_n+1)}{m_1m_n} & \frac{\Delta_2\Delta_n(m_n+1)}{m_2m_n} & \cdots & \frac{\Delta_n^2(m_n+1)}{m_n^2}\end{bmatrix}.
\end{aligned}$$

It follows that $\lim_{\vec{m}\to\vec{\infty}} S_R S_M^\top \leq \mathbf{0}_{n\times n}$, and by the Squeeze Theorem,

$$\lim_{\vec{m}\to\vec{\infty}} S_R S_M^\top = \mathbf{0}_{n\times n}.$$

As $S_M S_R^\top = (S_R S_M^\top)^\top$, we also have $\lim_{\vec{m}\to\vec{\infty}} S_M S_R^\top = \mathbf{0}_{n\times n}$. Thus, (4.24) reduces to

$$\begin{aligned}
&\left(\lim_{\vec{m}\to\vec{\infty}}\frac{1}{m}S_RS_R^\top - \lim_{\vec{m}\to\vec{\infty}}\frac{1}{m}S_MS_R^\top - \lim_{\vec{m}\to\vec{\infty}}\frac{1}{m}S_RS_M^\top + \lim_{\vec{m}\to\vec{\infty}}\frac{1}{m}S_MS_M^\top\right)^{-\top}\\
&= \left(\lim_{\vec{m}\to\vec{\infty}}\frac{1}{m}S_RS_R^\top\right)^{-\top}\\
&= \lim_{\vec{m}\to\vec{\infty}} m\left(S_RS_R^\top\right)^{-\top} = L_n
\end{aligned}$$

by Proposition 4.16. □

The previous proposition gives an expression for the first limit in (4.20). The following theorem gives the limit of the product $\frac{\Delta}{m}S\delta_s f(x^0;S)$, the second limit in (4.20), as a multiple integral over $R(\mathbf{0};d)$.

Proposition 4.19. *Let $f\in\mathcal{C}^0$ on an open domain containing $R(x^0;d)\subseteq \operatorname{dom} f$. Let $d=\begin{bmatrix}\Delta_1 & \Delta_2 & \cdots & \Delta_n\end{bmatrix}^\top > 0$, $x=\begin{bmatrix}x_1 & \cdots & x_n\end{bmatrix}^\top$ and let $S\in$*

$\mathbb{R}^{n\times m}$ *be defined as in* (4.22). *Then*

$$\lim_{\vec{m}\to\vec{\infty}} \frac{\Delta}{m} S\, \delta_s f(x^0; S) = T_n \in \mathbb{R}^n,$$

where

$$[T_n]_i = \int_{R(\mathbf{0};d)} x_i \left(f(x^0 + x) - f(x^0)\right)\, dx, \quad i \in \{1, 2, \ldots, n\}.$$

Proof. We have

$$\begin{aligned}
&\lim_{\vec{m}\to\vec{\infty}} \frac{\Delta}{m} S\, \delta_s f(x^0; S) && (4.27)\\
&= \lim_{\vec{m}\to\vec{\infty}} \frac{\Delta}{m} \begin{bmatrix} B^{1,1,\ldots,1} & \cdots B^{m_2,m_3,\ldots,m_n} \end{bmatrix} \begin{bmatrix} \delta_s f(x^0; B^{1,1,\ldots,1}) \\ \vdots \\ \delta_s f(x^0; B^{m_2,m_3,\ldots,m_n}) \end{bmatrix} \\
&= \lim_{\vec{m}\to\vec{\infty}} \sum_{z_2=1}^{m_2} \sum_{z_3=1}^{m_3} \cdots \sum_{z_n=1}^{m_n} \frac{\Delta}{m} B^{\vec{z}} \delta_s f(x^0; B^{\vec{z}}). && (4.28)
\end{aligned}$$

Note that $\frac{\Delta}{m}$ is the volume of one partition of $R(x^0; d)$. Recall that

$$\delta_s f(x^0; B^{\vec{z}}) = \left[f(x^0 + B^{\vec{z}} e^1) - f(x^0) \;\cdots\; f(x^0 + B^{\vec{z}} e^{m_1}) - f(x^0)\right]^\top$$

in $\mathbb{R}^{m_1}$. The matrix $B^{\vec{z}}$ has dimension $n \times m_1$ so (4.28) is the limit of a vector in $\mathbb{R}^n$. Since $f \in \mathcal{C}^0$ on an open domain containing $R(x^0; d)$, (4.28) is a vector of n definite integrals:

$$\begin{aligned}
&\lim_{\vec{m}\to\vec{\infty}} \sum_{z_2=1}^{m_2} \sum_{z_3=1}^{m_3} \cdots \sum_{z_n=1}^{m_n} \frac{\Delta}{m} B^{\vec{z}} \delta_s f(x^0; B^{\vec{z}}) \\
&= \begin{bmatrix} \int_{R(\mathbf{0};d)} x_1 \left(f(x^0 + x) - f(x^0)\right)\, dx \\ \int_{R(\mathbf{0};d)} x_2 \left(f(x^0 + x) - f(x^0)\right)\, dx \\ \vdots \\ \int_{R(\mathbf{0};d)} x_n \left(f(x^0 + x) - f(x^0)\right)\, dx \end{bmatrix} = T_n. \qquad \square
\end{aligned}$$

Now we are ready for our first main result: the limiting behavior of the GSG at x^0 over the dense grid $R(x^0; d)$. The result of Theorem 4.20 below is expressed as a multiple definite integral.

Theorem 4.20 (Limiting behavior of the GSG of f at x^0 over S)**.** *Let $f \in \mathcal{C}^0$ on an open domain containing $R(x^0; d) \subseteq \operatorname{dom} f$, $d = \begin{bmatrix}\Delta_1 & \cdots & \Delta_n\end{bmatrix}^\top > \mathbf{0}$. Let $S \in \mathbb{R}^{n\times m}$ be defined as in* (4.22)*. Then*

$$\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) = \Delta^{-1} L_n T_n, \tag{4.29}$$

where the entries of $L_n \in \mathbb{R}^{n\times n}$ are given by

$$[L_n]_{i,i} = \frac{12(3n-2)}{\Delta_i^2(3n+1)}, \quad i \in \{1, 2, \ldots, n\},$$
$$[L_n]_{i,j} = \frac{-36}{\Delta_i \Delta_j (3n+1)}, \quad i, j \in \{1, 2, \ldots, n\}, i \neq j,$$

and the entries of $T_n \in \mathbb{R}^n$ are given by

$$[T_n]_i = \int_{R(\mathbf{0};d)} x_i \left(f(x^0 + x) - f(x^0)\right)\, dx, \quad i \in \{1, 2, \ldots, n\}. \tag{4.30}$$

Proof. We have

$$\begin{aligned}\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) &= \lim_{\vec{m}\to\vec{\infty}} (S^\dagger)^\top \delta_s f(x^0; S) \\ &= \lim_{\vec{m}\to\vec{\infty}} \left(SS^\top\right)^{-\top} S\delta_s f(x^0; S) \\ &= \lim_{\vec{m}\to\vec{\infty}} \frac{m}{\Delta} \left(SS^\top\right)^{-\top} \frac{\Delta}{m} S\delta_s f(x^0; S).\end{aligned}$$

By Proposition 4.18,

$$\lim_{\vec{m}\to\vec{\infty}} m(SS^\top)^{-\top} = L_n \in \mathbb{R}^{n\times n}.$$

By Proposition 4.19,

$$\lim_{\vec{m}\to\vec{\infty}} \frac{\Delta}{m} S\delta_s f(x^0; S) = T_n \in \mathbb{R}^n.$$

Therefore,

$$\begin{aligned}\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) &= \frac{1}{\Delta} \lim_{\vec{m}\to\vec{\infty}} m(SS^\top)^{-\top} \lim_{\vec{m}\to\vec{\infty}} \frac{\Delta}{m} S\delta_s f(x^0; S) \\ &= \Delta^{-1} L_n T_n.\end{aligned}$$

□

Remark 4.21. The previous result agrees with what was found in [BHJB21] for $n = 1$. Indeed, in $\mathbb{R}$ [BHJB21, Theorem 4.1] found that

$$\lim_{m\to\infty} \nabla_s f(x^0; S) = \frac{3}{\Delta^3} \int_0^\Delta x(f(x^0 + x) - f(x^0))dx,$$

which is what (4.29) becomes when $n = 1$.

We end this section with an example of $\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S)$.

Example 4.22. Let $f : \mathbb{R}^2 \to \mathbb{R} : x \mapsto x_1^2 + x_2^2$ and the point of interest $x^0 = \begin{bmatrix}3 & 1\end{bmatrix}^\top$. Consider the sample region $R(x^0; \begin{bmatrix}1 & 1\end{bmatrix}^\top)$, a square of side length 1. By Theorem 4.20, we know that

$$\begin{aligned}
&\lim_{\vec{m}\to\vec{\infty}} \nabla_S f(x^0; S) \\
&= \frac{1}{\Delta} L_2 T_2 \\
&= \frac{1}{(1)(1)} \begin{bmatrix} \frac{48}{7} & \frac{-36}{7} \\ \frac{-36}{7} & \frac{48}{7} \end{bmatrix} \begin{bmatrix} \int_0^1 \int_0^1 x_1 \left((3+x_1)^2 + (1+x_2)^2 - 10\right) dx_1 dx_2 \\ \int_0^1 \int_0^1 x_2 \left((3+x_1)^2 + (1+x_2)^2 - 10\right) dx_1 dx_2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{48}{7} & \frac{-36}{7} \\ \frac{-36}{7} & \frac{48}{7} \end{bmatrix} \begin{bmatrix} \frac{35}{12} \\ \frac{31}{12} \end{bmatrix} \approx \begin{bmatrix} 6.71 \\ 2.71 \end{bmatrix}.
\end{aligned}$$

The absolute error is approximately $\left\| \begin{bmatrix}6.71 & 2.71\end{bmatrix}^\top - \begin{bmatrix}6 & 2\end{bmatrix}^\top \right\| \approx 1.01$.

Now that we have an expression for the GSG as the number of points tends to infinity, the next step is to define an error bound *ad infinitum*. We do this next.

Error bound ad infinitum of the GSG over a hyperrectangle

Now we use the results obtained thus far to formulate an error bound ad infinitum that does not depend on the number of points used in the hyperrectangle $R(x^0; d)$.

To obtain an error bound ad infinitum for the GSG at x^0 over $R(x^0; d)$, we require the function f to be $\mathcal{C}^2$ on an open domain containing $R(x^0; d)$. We will write $f(x^0 + x)$ as the first-order Taylor expansion (Theorem 4.8). By rewriting $f(x^0 + x)$ as a first-order Taylor expansion, the components of

the vector T_n defined in (4.30) can be written as

$$\begin{aligned}[T_n]_i &= \int_{R(\mathbf{0};d)} x_i \left(f(x^0+x) - f(x^0)\right)\, dx \\ &= \int_{R(\mathbf{0};d)} x_i \left(\nabla f(x^0)^\top x + R_1(x^0;x)\right)\, dx \\ &= \int_{R(\mathbf{0};d)} x_i \nabla f(x^0)^\top x\, dx + \int_{R(\mathbf{0};d)} x_i R_1(x^0;x)\, dx.\end{aligned}$$

for $i \in \{1,2,\ldots,n\}$. Let $v \in \mathbb{R}^n$ be the vector defined by

$$v_i = \int_{R(\mathbf{0};d)} x_i \nabla f(x^0)^\top x\, dx, \quad i \in \{1,2,\ldots,n\}, \tag{4.31}$$

and $w \in \mathbb{R}^n$ be the vector defined by

$$w_i = \int_{R(\mathbf{0};d)} x_i R_1(x^0;x)\, dx, \quad i \in \{1,2,\ldots,n\}.$$

Then the expression for $\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0;S)$ given in (4.29) can be expressed as

$$\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0;S) = \frac{1}{\Delta} L_n v + \frac{1}{\Delta} L_n w. \tag{4.32}$$

We begin by showing that the first term in (4.32) is equal to $\nabla f(x^0)$.

Lemma 4.23. *Let $L_n \in \mathbb{R}^{n\times n}$ be defined as in Theorem 4.20,*

$$\begin{aligned}[L_n]_{i,i} &= \frac{12(3n-2)}{\Delta_i^2(3n+1)}, \quad i \in \{1,2,\ldots,n\}, \\ [L_n]_{i,j} &= \frac{-36}{\Delta_i\Delta_j(3n+1)}, \quad i,j \in \{1,2,\ldots,n\}, i \neq j.\end{aligned}$$

Let $v \in \mathbb{R}^n$ be defined by

$$v_i = \int_{R(\mathbf{0};d)} x_i \nabla f(x^0)^\top x\, dx, \quad i \in \{1,2,\ldots,n\}.$$

Then $\frac{1}{\Delta} L_n v = \nabla f(x^0)$.

Proof. First, we find an expression for $v_i, i \in \{1, 2, \ldots, n\}$. To make notation tighter, let $g = \nabla f(x^0)$. We have

$$\begin{aligned}
v_i &= \int_{R(\mathbf{0};d)} x_i g^\top x \, dx \\
&= \int_{R(\mathbf{0};d)} x_i \left(\sum_{j=1}^{n} g_j x_j \right) dx \\
&= \int_{R(\mathbf{0};d)} x_i^2 g_i \, dx + \sum_{j \neq i} \int_{R(\mathbf{0};d)} x_i x_j g_j \, dx \\
&= \frac{\Delta_i^3}{3} \frac{\Delta}{\Delta_i} g_i + \sum_{j \neq i} \frac{\Delta_i^2}{2} \frac{\Delta_j^2}{2} \frac{\Delta}{\Delta_i \Delta_j} g_j \\
&= \frac{\Delta_i^2 \Delta}{3} g_i + \sum_{j \neq i} \frac{\Delta_i \Delta_j \Delta}{4} g_j = \frac{\Delta \Delta_i}{12} \left(4\Delta_i g_i + 3 \sum_{j \neq i} \Delta_j g_j \right).
\end{aligned}$$

Let $s = \sum_{j=1}^{n} \Delta_j g_j$. Then

$$v_i = \frac{\Delta \Delta_i}{12} \left(4\Delta_i g_i + 3 \left(\sum_{j \neq i}^{n} \Delta_j g_j \right) + 3\Delta_i g_i - 3\Delta_i g_i \right) = \frac{\Delta \Delta_i}{12} \left(\Delta_i g_i + 3s \right).$$

Now, let us compute $\frac{1}{\Delta} L_n v$. Let $D \in \mathbb{R}^{n \times n}$ be the diagonal matrix with entries $\Delta_1, \Delta_2, \ldots, \Delta_n$. Note that

$$L_n = \frac{12}{3n+1} D^{-1} L_n' D^{-1},$$

where $[L_n']_{i,i} = 3n - 2$ for all $i \in \{1, 2, \ldots, n\}$ and $[L_n']_{i,j} = -3$ for all $i, j \in \{1, \ldots, n\}, i \neq j$. Let the vector $d = \begin{bmatrix} \Delta_1 & \Delta_2 & \cdots & \Delta_n \end{bmatrix}^\top \in \mathbb{R}^n$. The vector v can be written as

$$v = \frac{\Delta}{12} \left(D^2 g + 3sd \right).$$

We obtain

$$\begin{aligned}
\frac{1}{\Delta} L_n v &= \frac{12}{\Delta(3n+1)} D^{-1} L_n' D^{-1} \frac{\Delta}{12} (D^2 g + 3sd) \\
&= \frac{1}{3n+1} D^{-1} L_n' D^{-1} (D^2 g + 3sd) \\
&= \frac{1}{3n+1} D^{-1} L_n' D g + \frac{3s}{3n+1} D^{-1} L_n' D^{-1} d. \qquad (4.33)
\end{aligned}$$

The first term in (4.33) is equal to

$$\begin{aligned}
&\frac{1}{3n+1}D^{-1}L_n'Dg \\
&= \frac{1}{3n+1}D^{-1}\begin{bmatrix}(3n-2)\Delta_1 g_1 - 3\sum_{j\neq 1}\Delta_j g_j \\ \vdots \\ (3n-2)\Delta_n g_n - 3\sum_{j\neq n}\Delta_j g_j\end{bmatrix} \\
&= \frac{1}{3n+1}D^{-1}\begin{bmatrix}(3n-2)\Delta_1 g_1 - 3\sum_{j\neq 1}\Delta_j g_j - 3\Delta_1 g_1 + 3\Delta_1 g_1 \\ \vdots \\ (3n-2)\Delta_n g_n - 3\sum_{j\neq n}\Delta_j g_j - 3\Delta_n g_n + 3\Delta_n g_n\end{bmatrix} \\
&= \frac{1}{3n+1}D^{-1}\begin{bmatrix}(3n+1)\Delta_1 g_1 - 3s \\ \vdots \\ (3n+1)\Delta_n g_n - 3s\end{bmatrix} \\
&= g - \frac{3s}{3n+1}D^{-1}\mathbf{1}_n.
\end{aligned}$$

The second term in (4.33) is equal to

$$\begin{aligned}
\frac{3s}{3n+1}D^{-1}L_n'D^{-1}d &= \frac{3s}{3n+1}D^{-1}L_n'\mathbf{1}_n \\
&= \frac{3s}{3n+1}D^{-1}\begin{bmatrix}(3n-2)-3(n-1) \\ \vdots \\ (3n-2)-3(n-1)\end{bmatrix} \\
&= \frac{3s}{3n+1}D^{-1}\mathbf{1}_n.
\end{aligned}$$

Hence,

$$\begin{aligned}
\frac{1}{\Delta}L_n &= \frac{1}{3n+1}D^{-1}L_n'Dg + \frac{3s}{3n+1}D^{-1}L_n'D^{-1}d \\
&= g - \frac{3s}{3n+1}D^{-1}\mathbf{1}_n + \frac{3s}{3n+1}D^{-1}\mathbf{1}_n = g.
\end{aligned}$$

□

By the previous lemma, we now know that the error bound ad infinitum of the GSG is

$$\left\|\lim_{\vec{m}\to\vec{\infty}}\nabla_s f(x^0;S) - \nabla f(x^0)\right\| = \left\|\frac{1}{\Delta}L_n v + \frac{1}{\Delta}L_n w - \nabla f(x^0)\right\| = \frac{1}{\Delta}\left\|L_n w\right\|. \tag{4.34}$$

To determine an upper bound for $\frac{1}{\Delta}\|L_n w\|$, recall that L_n can be written as $D^{-1}\ddot{L}_n D^{-1}$, where the diagonal matrix $D = \operatorname{Diag}\begin{bmatrix}\Delta_1 & \Delta_2 & \cdots & \Delta_n\end{bmatrix} \in \mathbb{R}^{n\times n}$ and the entries of $\ddot{L}_n \in \mathbb{R}^{n\times n}$ are given by

$$\begin{aligned}[\ddot{L}_n]_{i,i} &= \frac{12(3n-2)}{3n+1}, \quad i \in \{1,\dots,n\},\\ [\ddot{L}_n]_{i,j} &= \frac{-36}{3n+1}, \quad i,j \in \{1,2,\dots,n\}, i \neq j.\end{aligned} \tag{4.35}$$

Hence, the right-hand side of (4.34) is

$$\frac{1}{\Delta}\,\|L_n w\| = \frac{1}{\Delta}\|D^{-1}\ddot{L}_n D^{-1} w\| \leq \frac{1}{\Delta}\|D^{-1}\|\|\ddot{L}_n\|\|D^{-1}w\|. \tag{4.36}$$

The norm of D^{-1} is simply $1/\Delta_{\texttt{min}}$ where $\Delta_{\texttt{min}} = \min_{i\in\{1,\dots,n\}}\Delta_i$. In the following two lemmas, we find the value of the two other norms that appear in (4.36), $\|\ddot{L}_n\|$ and $\|D^{-1}w\|$.

Lemma 4.24. *Let $\ddot{L}_n \in \mathbb{R}^{n\times n}$ be defined as in* (4.35)*. Then*

$$\left\|\ddot{L}_n\right\| = 12.$$

Proof. We find the norm by finding the largest eigenvalue of $\ddot{L}_n^\top \ddot{L}_n$. We have

$$\left[\ddot{L}_n^\top \ddot{L}_n\right]_{i,j} = \begin{cases} \dfrac{144}{(3n+1)^2}(9n^2-3n-5), & \text{if } i = j,\\ \dfrac{144}{(3n+1)^2}(-9n-6), & \text{if } i \neq j, i,j \in \{1,2,\dots,n\}.\end{cases}$$

Let $t = \frac{144}{(3n+1)^2}$. It follows that

$$\begin{aligned}&|\ddot{L}_n^\top \ddot{L}_n - \lambda \operatorname{Id}|\\ &= t^n \left|\begin{bmatrix} 9n^2-3n-5-\frac{\lambda}{t} & -9n-6 & \cdots & -9n-6 & -9n-6\\ -9n-6 & 9n^2-3n-5-\frac{\lambda}{t} & \cdots & -9n-6 & -9n-6\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ -9n-6 & -9n-6 & \cdots & 9n^2-3n-5-\frac{\lambda}{t} & -9n-6\\ -9n-6 & -9n-6 & \cdots & -9n-6 & 9n^2-3n-5-\frac{\lambda}{t}\end{bmatrix}\right|.\end{aligned} \tag{4.37}$$

Now, we apply elementary row and column operations on the matrix in (4.37) to make it an upper-diagonal matrix. First, let Row i = Row i −

Row 1 for $i \in \{2, 3, \ldots, n\}$. Second, using the new matrix, let Column 1 = Column 1 + $\sum_{i=2}^{n}$ Column i. This generates the matrix

$$\begin{bmatrix} 1-\frac{\lambda}{t} & -9n-6 & -9n-6 & \cdots & -9n-6 & -9n-6 \\ 0 & 9n^2+6n+1-\frac{\lambda}{t} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 9n^2+6n+1-\frac{\lambda}{t} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 9n^2+6n+1-\frac{\lambda}{t} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 9n^2+6n+1-\frac{\lambda}{t} \end{bmatrix}$$

and therefore we have

$$|\ddot{L}_n^\top \ddot{L}_n - \lambda \operatorname{Id}| = t^n \left(1 - \frac{\lambda}{t}\right) \left(9n^2 + 6n + 1 - \frac{\lambda}{t}\right)^{n-1}.$$

Hence, the eigenvalues of $\ddot{L}_n^\top \ddot{L}_n$ are

$$\lambda_1 = t = \frac{144}{(3n+1)^2} \quad \text{and} \quad \lambda_{2,3,\ldots,n} = t(9n^2+6n+1) = 144.$$

We see that the maximum eigenvalue, denoted $\lambda_{\texttt{max}}(\ddot{L}_n^\top \ddot{L}_n)$, is 144. Therefore,

$$\left\|\ddot{L}_n\right\| = \sqrt{\lambda_{\texttt{max}}(\ddot{L}_n^\top \ddot{L}_n)} = 12. \qquad \square$$

Lemma 4.25. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^2$ on an open domain containing $R(x^0; d)$. Let $L_{\nabla f}$ denote the Lipschitz constant of ∇f on $R(x^0; d)$. Let $D = \operatorname{Diag}\begin{bmatrix}\Delta_1 & \cdots & \Delta_n\end{bmatrix} \in \mathbb{R}^{n\times n}$ and let $w \in \mathbb{R}^n$ be defined by*

$$w_i = \int_{R(\mathbf{0};d)} x_i R_1(x^0; x)\, dx, \quad i \in \{1, 2, \ldots, n\},$$

where $R_1(x^0; x)$ is the remainder term of the first-order Taylor expansion of $f(x^0 + x)$ about x^0. Then

$$\|D^{-1}w\| \leq \frac{\sqrt{n}}{8} L_{\nabla f} \|d\|^2.$$

Moreover, if all Δ_i are equal (i.e. the sample region is a hypercube), then

$$\|D^{-1}w\| \leq \frac{\sqrt{n}}{24} \frac{(2n+1)}{n} L_{\nabla f} \Delta \|d\|^2.$$

Proof. First, let us find an upper bound for $\left|\frac{w_i}{\Delta_i}\right|, i \in \{1, 2, \ldots, n\}$. We have

$$\begin{aligned}\left|\frac{w_i}{\Delta_i}\right| &= \left|\int_{R(\mathbf{0};d)} \frac{x_i}{\Delta_i} R_1(x^0; x)\, dx\right| \\ &= \frac{1}{2}\left|\int_{R(\mathbf{0};d)} \frac{x_i}{\Delta_i} x^\top \nabla^2 f(\xi) x\, dx\right| \\ &\leq \frac{1}{2}\int_{R(\mathbf{0};d)} \left|\frac{x_i}{\Delta_i}\right| \|x\|^2 \|\nabla^2 f(\xi)\|\, dx.\end{aligned}$$

Using $\|\nabla^2 f(\xi)\| \leq L_{\nabla f}$ and $R(\mathbf{0}; d) \subseteq \mathbb{R}^n_+$, we obtain

$$\begin{aligned}&\frac{1}{2}\int_{R(\mathbf{0};d)} \left|\frac{x_i}{\Delta_i}\right| \|x\|^2 \|\nabla^2 f(\xi)\|\, dx \\ \leq &\frac{L_{\nabla f}}{2}\int_{R(\mathbf{0};d)} \left(\frac{x_i}{\Delta_i}\right)\left(\sum_{j=1}^n x_j^2\right)\, dx \\ = &\frac{L_{\nabla f}}{2}\left(\int_{R(\mathbf{0};d)} \frac{x_i^3}{\Delta_i}\, dx + \sum_{j\neq i}\int_{R(\mathbf{0};d)} \frac{x_i}{\Delta_i} x_j^2\, dx\right) \\ = &\frac{L_{\nabla f}}{2}\left(\frac{\Delta\Delta_i^2}{4} + \sum_{j\neq i}\frac{\Delta\Delta_j^2}{6}\right) \\ = &\frac{L_{\nabla f}}{24}\Delta\left(3\Delta_i^2 + 2\sum_{j\neq i}\Delta_j^2\right) = \frac{L_{\nabla f}}{24}\Delta\left(\Delta_i^2 + 2\|d\|^2\right). \qquad (4.38)\end{aligned}$$

Therefore,

$$\begin{aligned}\|D^{-1}w\| &\leq \sqrt{\sum_{i=1}^n \left(\frac{L_{\nabla f}}{24}\Delta\left(\Delta_i^2 + 2\|d\|^2\right)\right)^2} \\ &\leq \frac{L_{\nabla f}}{24}\Delta\sqrt{\sum_{i=1}^n (3\|d\|^2)^2} = \frac{\sqrt{n}}{8} L_{\nabla f}\Delta\|d\|^2.\end{aligned}$$

When all Δ_i are equal, then (4.38) becomes

$$\frac{L_{\nabla f}}{24}\Delta\left(\Delta_i^2 + 2\|d\|^2\right) = \frac{L_{\nabla f}}{24}\Delta\left(\frac{\|d\|^2}{n} + 2\|d\|^2\right)$$

and it follows that

$$\|D^{-1}w\| \leq \frac{\sqrt{n}}{24}\frac{(2n+1)}{n}L_{\nabla f}\Delta\|d\|^2. \qquad \square$$

We are now ready to introduce an error bound ad infinitum for the GSG.

Theorem 4.26 (Error bound ad infinitum for the GSG)**.** *Let the function $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^2$ on an open domain containing $R(x^0; d)$ where $d = \begin{bmatrix}\Delta_1 & \Delta_2 & \cdots & \Delta_n\end{bmatrix} > 0$ and x^0 is the point of interest. Let Δ_S be the radius of $S \in \mathbb{R}^{n\times m}$ as defined in (4.2). Let $\Delta_{\texttt{min}} = \min_{i\in\{1,\ldots,n\}} \Delta_i$. Let $L_{\nabla f}$ denote the Lipschitz constant of ∇f on $R(x^0; d)$. Then*

$$\left\| \lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) - \nabla f(x^0) \right\| \leq \frac{3}{2}\sqrt{n}\, L_{\nabla f}\, \frac{\Delta_S^2}{\Delta_{\texttt{min}}}. \tag{4.39}$$

Moreover, if all Δ_i are equal, then

$$\left\| \lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) - \nabla f(x^0) \right\| \leq \frac{1}{2}(2n+1)\, L_{\nabla f}\, \Delta_S.$$

Proof. We have

$$\begin{aligned}
&\left\| \lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) - \nabla f(x^0) \right\| \\
&= \left\| \Delta^{-1} L_n(v+w) - \nabla f(x^0) \right\| \\
&= \left\| \Delta^{-1} L_n v - \nabla f(x^0) + \Delta^{-1} L_n w \right\| \\
&\leq \left\| \Delta^{-1} L_n v - \nabla f(x^0) \right\| + \Delta^{-1} \left\| D^{-1} \right\| \|\ddot{L}_n\| \|D^{-1} w\|.
\end{aligned}$$

By Lemma 4.23, we know $\|\Delta^{-1} L_n v - \nabla f(x^0)\| = 0$. By Lemma 4.24, Lemma 4.25 and since $\|D^{-1}\| = \Delta_{\texttt{min}}$, we obtain

$$\left\| \lim_{\vec{m}\to\vec{\infty}} \nabla f(x^0; S) - \nabla f(x^0) \right\| = \Delta^{-1} \left\| D^{-1} \right\| \|\ddot{L}_n\| \|D^{-1} w\| \leq \frac{3}{2}\sqrt{n}\, L_{\nabla f}\, \Delta_S.$$

When all Δ_i are equal, $\Delta_{\texttt{min}} = \Delta_i = \Delta_S/\sqrt{n}$ for any $i \in \{1, \ldots, n\}$. We obtain

$$\begin{aligned}
\left\| \lim_{\vec{m}\to\vec{\infty}} \nabla f(x^0; S) - \nabla f(x^0) \right\| &\leq \Delta^{-1} \left\| D^{-1} \right\| \|\ddot{L}_n\| \|D^{-1} w\| \\
&\leq \Delta^{-1} \frac{\sqrt{n}}{\Delta_S} (12) \frac{\sqrt{n}}{24} \frac{(2n+1)}{n} L_{\nabla f} \Delta \Delta_S^2 \\
&= \frac{1}{2}(2n+1)\, L_{\nabla f}\, \Delta_S.
\end{aligned}$$

$\square$

In comparison to the classical error bound (Theorem 4.10), notice that the error bound in Theorem 4.26 is not reliant on the number of sample points m. As such, Theorem 4.26 immediately provides a finite error bound ad infinitum, where Theorem 4.10 does not.

In Theorem 4.26, we see that the error bound is $O\left(\frac{\Delta_S^2}{\Delta_{\texttt{min}}}\right)$. As Δ_S is the radius of the sample region and $\Delta_{\texttt{min}}$ is the length of the shortest side of the sample region, the theorem suggests that the more uniform the sample region the smaller the error. In other words, we want the simplex with vertices $x^0, x^0 + \Delta_1, \ldots, x^0 + \Delta_n$ to be 'as uniform as possible'. Analyzing the ratio $\Delta_S/\Delta_{\texttt{min}}$, we see that the minimum value of this ratio is $\sqrt{n}$, which occurs when the hyperrectangle is a hypercube.

4.3.2 The GSG over a dense ball

In this section, we find an error bound ad infinitum for the GSG of f at x^0 over a ball. First, we present some results on integration over a ball. In the next theorem, $P(x)$ denotes a monomial. That is,

$$P(x) = x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}, \tag{4.40}$$

where $\alpha_i \in \mathbb{N} \cup \{0\}$ for all i.

Theorem 4.27 (Integrating over a ball)**.** [Fol01] *Let P be a monomial defined as in* (4.40)*. Let σ be the $(n-1)$-dimensional surface measure on $S_n(\mathbf{0}; r)$. Let $\beta_i = \frac{1}{2}(\alpha_i + 1)$ for all i. Then*

$$\int_{\overline{B}_n(\mathbf{0};r)} P(x)dx = \frac{r^{\alpha_1+\cdots+\alpha_n+n}}{\alpha_1 + \cdots + \alpha_n + n} \int_{S_n(\mathbf{0};r)} P d\sigma,$$

where

$$\int_{S_n(\mathbf{0};r)} P d\sigma = \begin{cases} 0, & \text{if any } \alpha_i \text{ is odd,} \\ \dfrac{2\Gamma(\beta_1)\Gamma(\beta_2)\cdots\Gamma(\beta_n)}{\Gamma(\beta_1 + \beta_2 + \cdots + \beta_n)}, & \text{if all } \alpha_i \text{ are even.} \end{cases}$$

We will also need an expression for the integral of

$$Q(x) = |x_1|^{\alpha_1} |x_2|^{\alpha_2} \cdots |x_n|^{\alpha_n}$$

over the ball $\overline{B}_n(\mathbf{0}; r)$.

Proposition 4.28. *Let* $Q(x) = |x_1|^{\alpha_1}|x_2|^{\alpha_2}\cdots|x_n|^{\alpha_n}$, *and* $\beta_i = \frac{1}{2}(\alpha_i + 1)$ *for all* i. *Then*

$$\int_{\overline{B}_n(\mathbf{0};r)} Q(x)dx = \frac{2r^{\alpha_1+\cdots+\alpha_n+n}}{(\alpha_1+\cdots+\alpha_n+n)}\frac{\Gamma(\beta_1)\cdots\Gamma(\beta_n)}{\Gamma(\beta_1+\cdots+\beta_n)}.$$

Proof. Note that $|x_i|^{\alpha_i}$ is an even function for any $\alpha_i \in \mathbb{N}\cup\{0\}$. Using this fact and following the same scheme of the proof for Theorem 4.27 in [Fol01] yields the result. □

Now, let us define the matrix of directions S_R that is used to form the sample points. Recall that in $\mathbb{R}^n$, the conversion from Cartesian coordinates $x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^\top$ to n-spherical coordinates is

$$\begin{aligned}
x_1 &= \rho\cos\phi_1,\\
x_2 &= \rho\sin\phi_1\cos\phi_2,\\
x_3 &= \rho\sin\phi_1\sin\phi_2\cos\phi_3,\\
&\vdots\\
x_{n-2} &= \rho\sin\phi_1\cdots\sin\phi_{n-3}\cos\phi_{n-2},\\
x_{n-1} &= \rho\sin\phi_1\cdots\sin\phi_{n-2}\cos\theta,\\
x_n &= \rho\sin\phi_1\cdots\sin\phi_{n-2}\sin\theta,
\end{aligned}$$

where $\rho = \|x\|$ and $\theta, \phi_1, \dots, \phi_{n-2}$ are the angles that identify the direction of x. The angles have domains $\theta \in [0, 2\pi)$ and $\phi_i \in [0, \pi)$ for all $i \in \{1, 2, \dots, n-2\}$. To keep the same notation as the hyperrectangle, we define

$$\Delta_1 = r, \quad \Delta_2 = 2\pi, \quad \Delta_3 = \Delta_4 = \cdots = \Delta_n = \pi.$$

As before, m_i represents the number of subdivisions used to build the partitions in the ball. Once again, we define $\overline{\Delta}_i = \Delta_i/m_i$ for all $i \in \{1, 2, \dots, n\}$, $\overline{\Delta} = \overline{\Delta}_1\overline{\Delta}_2\cdots\overline{\Delta}_n$, and $m = m_1m_2\cdots m_n$.

Now we build the matrix $S_R \in \mathbb{R}^{n\times m}$. The matrix S_R contains all directions to add to the point of interest x^0 to obtain a sample point in each partition of the ball $\overline{B}_n(x^0; r)$. A polar grid is built, in which each partition is a "bent" hyperrectangle. When using S_R, the sample point chosen in each partition is the rightmost endpoint (the point with the largest values of $\rho, \theta, \phi_1, \dots, \phi_{n-2}$). Let $\vec{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^\top$ be a vector of indices in $\mathbb{R}^n$ (not $\mathbb{R}^{n-1}$ as it is the case for $\vec{z}$). Define

$$s^{\vec{y}} = \frac{\rho y_1}{m_1}\begin{bmatrix}\cos\frac{\pi y_3}{m_3} & \sin\frac{\pi y_3}{m_3}\cos\frac{\pi y_4}{m_4} & \dots & \sin\frac{\pi y_3}{m_3}\cdots\sin\frac{\pi y_n}{m_n}\cos\frac{2\pi y_2}{m_2} & \sin\frac{\pi y_3}{m_3}\cdots\sin\frac{\pi y_n}{m_n}\sin\frac{2\pi y_2}{m_2}\end{bmatrix}^\top$$

in $\mathbb{R}^n$. Then S_R can be written as

$$S_R = \begin{bmatrix} s^{1,1,1,\ldots,1,1} & s^{1,1,1,\ldots,1,2} & \cdots & s^{m_1,m_2,m_3,\cdots,m_{n-1},m_n} \end{bmatrix}.$$

Let us provide an example of a sample set built by using S_R in $\mathbb{R}^2$.

Example 4.29. In this example, the point of interest is $x^0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$. The sample region is a ball with radius $\Delta_1 = r = 30$. Set $m_1 = 3$ and $m_2 = 4$. Hence, $\overline{\Delta}_1 = \frac{30}{3} = 10$ and $\overline{\Delta}_2 = \frac{2\pi}{4} = \frac{\pi}{2}$. The matrix $S_R \in \mathbb{R}^{2\times 12}$ is given by

$$\begin{aligned} S_R &= \begin{bmatrix} s^{1,1} & s^{1,2} & s^{1,3} & s^{1,4} & s^{2,1} & s^{2,2} & s^{2,3} & s^{2,4} & s^{3,1} & s^{3,2} & s^{3,3} & s^{3,4} \end{bmatrix} \\ &= \begin{bmatrix} 0 & -10 & 0 & 10 & 0 & -20 & 0 & 20 & 0 & -30 & 0 & 30 \\ 10 & 0 & -10 & 0 & 20 & 0 & -20 & 0 & 30 & 0 & -30 & 0 \end{bmatrix}. \end{aligned}$$

The sample points x^j are obtained by setting $x^j = x^0 + S_R e^j$ for all $j \in \{1, 2, \ldots, 12\}$. Figure 4.3 illustrates the sample points built from the matrix S_R.

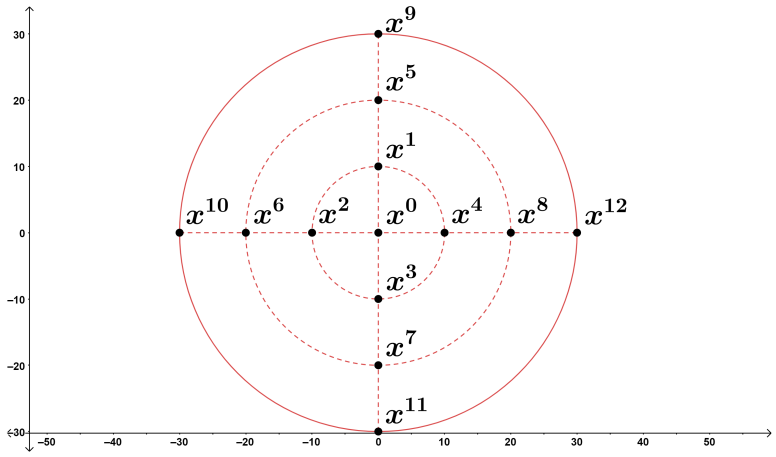


Figure 4.3: An example of a sample set built from S_R in $\mathbb{R}^2$.

Note that S_R is full row rank whenever all $m_i > 2$. For the remainder of this section, assume $m_i > 2$ for all i. Hence, we want to find the limit of the following expression:

$$\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S_R) = \lim_{\vec{m}\to\vec{\infty}} \left(S_R S_R^\top\right)^{-\top} S_R \delta_s f(x^0; S_R). \tag{4.41}$$

Define the determinant of the Jacobian as a function $J : \mathbb{R}^n \to \mathbb{R}$:

$$J(y_1, \ldots, y_n) = \left(\frac{\rho y_1}{m_1}\right)^{n-1} \sin^{n-2} \frac{\pi y_3}{m_3} \sin^{n-3} \frac{\pi y_4}{m_4} \cdots \sin^2 \frac{\pi y_{n-1}}{m_{n-1}} \sin \frac{\pi y_n}{m_n}. \tag{4.42}$$

Let $J \in \mathbb{R}^{m \times m}$ be the matrix defined by

$$J = \text{Diag}\left[j^{1,1,\ldots,1} \quad j^{1,1,\ldots,2} \quad \ldots \quad j^{m_1,m_2,\ldots,m_n}\right]$$

where $j^{\vec{y}} = J(y_1, y_2, \ldots, y_n)$.

Define

$$\begin{aligned} K &= S_R J S_R^\dagger \\ &= S_R J S_R^\top (S_R S_R^\top)^{-1}. \end{aligned}$$

Notice that $K \in \mathbb{R}^{n \times n}$ is an invertible matrix such that

$$S_R J = K S_R. \tag{4.43}$$

Considering (4.41), notice that

$$\begin{aligned} (S_R S_R^\top)^{-\top} S_R \delta_s f(x^0; S_R) &= (S_R S_R^\top)^{-\top} \frac{1}{\overline{\Delta}} K^{-1} K S_R \delta_s f(x^0; S_R) \overline{\Delta} \\ &= (K S_R S_R^\top \overline{\Delta})^{-\top} K S_R \delta_s f(x^0; S_R) \overline{\Delta} \\ &= (S_R J S_R^\top \overline{\Delta})^{-\top} S_R J \delta_s f(x^0; S_R) \overline{\Delta}. \end{aligned}$$

Therefore, (4.41) is equal to

$$\begin{aligned} &\lim_{\vec{m} \to \vec{\infty}} \left(S_R S_R^\top\right)^{-\top} S_R \delta_s f(x^0; S_R) \\ &= \lim_{\vec{m} \to \vec{\infty}} \left[\left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} (s^{\vec{y}})^\top \right)^{-\top} \sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} \delta_s f(x^0; s^{\vec{y}}) \right] \\ &= \lim_{\vec{m} \to \vec{\infty}} \left[\left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} (s^{\vec{y}})^\top J(y_1, \ldots, y_n) \overline{\Delta} \right)^{-\top} \sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} \delta_s f(x^0; s^{\vec{y}}) J(y_1, \ldots, y_n) \overline{\Delta} \right]. \end{aligned}$$

Considering

$$\lim_{\vec{m} \to \vec{\infty}} \left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} (s^{\vec{y}})^\top J(y_1, \ldots, y_n) \overline{\Delta} \right)^{-\top} \tag{4.44}$$

and

$$\lim_{\vec{m}\to\vec{\infty}} \left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} \delta_s f(x^0; s^{\vec{y}}) J(y_1, \ldots, y_n) \overline{\Delta} \right), \tag{4.45}$$

we shall show both limits exist so we can write

$$\begin{aligned} &\lim_{\vec{m}\to\vec{\infty}} \left(S_R S_R^\top \right)^{-\top} S_R \delta_s f(x^0; S_R) \\ &= \lim_{\vec{m}\to\vec{\infty}} \left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} (s^{\vec{y}})^\top J(y_1, \ldots, y_n) \overline{\Delta} \right)^{-\top} \lim_{\vec{m}\to\vec{\infty}} \left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} \delta_s f(x^0; s^{\vec{y}}) J(y_1, \ldots, y_n) \overline{\Delta} \right). \end{aligned}$$

We begin by examining the first limit in (4.44). Recall the following formula for the volume of a ball in $\mathbb{R}^{n+2}$:

$$V_{n+2} = \frac{2\pi^{\frac{n}{2}+1} r^{n+2}}{\Gamma(\frac{n}{2}+1)(n+2)}, \tag{4.46}$$

where Γ is the Gamma function given by

$$\Gamma\left(\frac{n}{2}+1\right) = \begin{cases} (n-1)! & \text{if} \quad \frac{n}{2} \in \mathbb{N}, \\ \left(\frac{n}{2}-1\right)\left(\frac{n}{2}-2\right)\cdots\frac{1}{2}\sqrt{\pi} & \text{if} \quad \frac{n}{2} \notin \mathbb{N}. \end{cases}$$

Let

$$M_n = \lim_{\vec{m}\to\vec{\infty}} \left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} \left(s^{\vec{y}}\right)^\top J(y_1, \ldots, y_n) \overline{\Delta} \right) \in \mathbb{R}^{n\times n}. \tag{4.47}$$

Proposition 4.30 finds the matrix M_n and thereby provides the first limit in (4.44).

Proposition 4.30. *Let $M_n \in \mathbb{R}^{n\times n}$ be defined as in* (4.47)*. Then*

$$[M_n]_{i,j} = \begin{cases} \dfrac{V_{n+2}}{2\pi} & \text{if} \quad i = j, \\ 0 & \text{if} \quad i \neq j. \end{cases}$$

Consequently

$$\lim_{\vec{m}\to\vec{\infty}} \left(\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} (s^{\vec{y}})^\top J(y_1, \ldots, y_n) \overline{\Delta} \right)^{-\top} = \frac{2\pi}{V_{n+2}} \operatorname{Id}_n.$$

Proof. Each entry of M_n is a n-tuple Riemann sum over $\overline{B}_n(\mathbf{0}; r)$. Taking the limit as $\vec{m} \to \vec{\infty}$, each entry of M_n can be written as the following integral (in Cartesian coordinates):

$$[M_n]_{i,j} = \int_{\overline{B}_n(\mathbf{0};r)} x_i x_j dx, \quad i, j \in \{1, 2, \ldots, n\}.$$

The off-diagonal entries of M_n are zero by Theorem 4.27. The diagonal entries are given by

$$\begin{aligned}
[M_n]_{i,i} &= \frac{r^{n+2}}{(n+2)} \frac{2\Gamma\left(\frac{1}{2}\right)^{n-1} \Gamma\left(\frac{3}{2}\right)}{\Gamma\left(\frac{1}{2}(n-1) + \frac{3}{2}\right)} \\
&= \frac{r^{n+2}}{(n+2)} \frac{2\pi^{\frac{n-1}{2}} \pi^{\frac{1}{2}}}{2\Gamma\left(\frac{n}{2}+1\right)} \\
&= \frac{r^{n+2} \pi^{\frac{n}{2}}}{(n+2)\Gamma\left(\frac{n}{2}+1\right)}.
\end{aligned}$$

From (4.46), we see that the diagonal entries are simply

$$[M_n]_{i,i} = \frac{V_{n+2}}{2\pi}.$$

The second equation follows trivially. □

In the next proposition, we give an expression for the second limit in (4.45).

Proposition 4.31. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ with $\overline{B}_n(x^0; r) \subseteq \operatorname{dom} f$. Then the following limit can be written as a vector of integrals (in Cartesian coordinates):*

$$\begin{aligned}
&\lim_{\vec{m} \to \vec{\infty}} \left[\sum_{y_1=1}^{m_1} \cdots \sum_{y_n=1}^{m_n} s^{\vec{y}} \delta_s f(x^0; s^{\vec{y}}) J(y_1, \ldots, y_n) \overline{\Delta} \right] \\
&= \begin{bmatrix} \int_{\overline{B}_n(\mathbf{0};r)} x_1 \left(f(x^0 + x) - f(x^0) \right) dx \\ \int_{\overline{B}_n(\mathbf{0};r)} x_2 \left(f(x^0 + x) - f(x^0) \right) dx \\ \vdots \\ \int_{\overline{B}_n(\mathbf{0};r)} x_n \left(f(x^0 + x) - f(x^0) \right) dx \end{bmatrix} = T_n \in \mathbb{R}^n \,. \qquad (4.48)
\end{aligned}$$

Proof. The n-tuple sum of the left-hand side of (4.48) is a Riemann sum with a finite-sized sample region $\overline{B}_n(x^0; r)$. The result follows by taking the limit as $\vec{m} \to \vec{\infty}$. □

Now we generalize the matrix S_R. Let

$$S = \begin{bmatrix} s^{1,1,1,\ldots,1,1} & s^{1,1,1,\ldots,1,2} & \cdots & s^{m_1,m_2,m_3,\cdots,m_{n-1},m_n} \end{bmatrix}$$

be a matrix in $\mathbb{R}^{n\times m}$ in which each column s is a direction to add to x^0 to form exactly one arbitrary sample point in each partition of $\overline{B}_n(x^0;r)$. Note that Propositions 4.30 and 4.31 still hold by considering S instead of S_R. Indeed, since f is a continuous function, as long as exactly one sample point is used in each partition of the ball, the results of the previous two propositions hold. We are now ready to provide an expression for the GSG ad infinitum of f at x^0 over $\overline{B}_n(x^0;r)$.

Theorem 4.32 (The GSG over a ball)**.** *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ with $\overline{B}_n(x^0;r) \subseteq \operatorname{dom} f$. Let $S \in \mathbb{R}^{n\times m}$ be a matrix such that each sample point $x^0 + Se^j, j \in \{1,2,\ldots,m\}$, is in exactly one partition of the ball $\overline{B}_n(x^0;r)$. Let V_{n+2} be the volume of a ball with radius r in $\mathbb{R}^{n+2}$ and $T_n \in \mathbb{R}^n$ be defined as in* (4.48)*. Then*

$$\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0;S) = \frac{2\pi}{V_{n+2}} T_n. \tag{4.49}$$

Proof. we have

$$\begin{aligned}
&\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0;S) \\
&= \lim_{\vec{m}\to\vec{\infty}} \left(SS^\top\right)^{-\top} S\delta_s f(x^0;S) \\
&= \lim_{\vec{m}\to\vec{\infty}} \left[\left(\sum_{y_1=1}^{m_1}\cdots\sum_{y_n=1}^{m_n} s^{\vec{y}}(s^{\vec{y}})^\top J(y_1,\ldots,y_n)\overline{\Delta}\right)^{-\top}\right] \lim_{\vec{m}\to\vec{\infty}} \left[\sum_{y_1=1}^{m_1}\cdots\sum_{y_n=1}^{m_n} s^{\vec{y}}\delta_s f(x^0;s^{\vec{y}})J(y_1,\ldots,y_n)\overline{\Delta}\right] \\
&= \frac{2\pi}{V_{n+2}} T_n,
\end{aligned}$$

by Propositions 4.30 and 4.31. □

Let us provide an example of the calculations necessary to obtain the limit of the the GSG.

Example 4.33. Let $f : \mathbb{R}^2 \to \mathbb{R} : x \mapsto x_1^2 + x_2^2$. Let the point of interest be $x^0 = \begin{bmatrix} 3 & 1 \end{bmatrix}^\top$. Let the sample region be $\overline{B}_2(x^0;1)$. By Theorem 4.32, we know that

$$\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0;S) = \frac{2\pi}{V_4} \begin{bmatrix} \int_{\overline{B}_2(\mathbf{0};1)} x_1 \left(f(x^0+x) - f(x^0)\right) dx \\ \int_{\overline{B}_2(\mathbf{0};1)} x_2 \left(f(x^0+x) - f(x^0)\right) dx \end{bmatrix}.$$

Writing the vector of integrals in polar coordinates and since $V_4 = \frac{\pi^2}{2}$, we obtain

$$\begin{aligned}
&\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) \\
&= \frac{4}{\pi} \begin{bmatrix} \int_0^{2\pi} \int_0^1 r\cos\theta \left((3 + r\cos\theta)^2 + (1 + r\sin\theta)^2 - 10\right) r dr d\theta \\ \int_0^{2\pi} \int_0^1 r\sin\theta \left((3 + r\cos\theta)^2 + (1 + r\sin\theta)^2 - 10\right) r dr d\theta \end{bmatrix} \\
&= \frac{4}{\pi} \begin{bmatrix} \frac{3}{2}\pi \\ \frac{\pi}{2} \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}.
\end{aligned}$$

Note that for this problem, $\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) = \nabla f(x^0)$.

The reason why the GSG is perfectly accurate in the previous example will be discussed at the end of this section.

Error bound ad infinitum of the GSG over a ball

Now we develop an error bound ad infinitum for the GSG over a ball. To obtain such an error bound, we require f to be $\mathcal{C}^3$ on an open domain containing $\overline{B}_n(x^0; r)$. The function f at $x^0 + x$ is written as the second-order Taylor expansion (Theorem 4.8). By rewriting $f(x^0 + x)$ as a second-order Taylor expansion about x^0, each entry of the vector T_n defined in Proposition 4.31 can be written as

$$\begin{aligned}
[T_n]_i &= \int_{\overline{B}_n(\mathbf{0};r)} x_i \left(f(x^0 + x) - f(x^0)\right) dx \\
&= \int_{\overline{B}_n(\mathbf{0};r)} x_i \left(\nabla f(x^0)^\top x + \frac{1}{2} x^\top \nabla^2 f(x^0) x + R_2(x^0; x)\right) dx \\
&= \int_{\overline{B}_n(\mathbf{0};r)} x_i \nabla f(x^0)^\top x dx + \int_{\overline{B}_n(\mathbf{0};r)} x_i x^\top \nabla^2 f(x^0) x dx \int_{\overline{B}_n(\mathbf{0};r)} x_i R_2(x^0 + x) dx,
\end{aligned}$$

for $i \in \{1, 2, \ldots, n\}$. Let $v \in \mathbb{R}^n$ be the vector defined by

$$v_i = \int_{\overline{B}_n(\mathbf{0};r)} x_i \nabla f(x^0)^\top x dx, \quad i \in \{1, 2, \ldots, n\}, \tag{4.50}$$

$w \in \mathbb{R}^n$ be the vector defined by

$$w_i = \int_{\overline{B}_n(\mathbf{0};r)} x_i x^\top \nabla^2 f(x^0) x dx, \quad i \in \{1, 2, \ldots, n\}, \tag{4.51}$$

and $z \in \mathbb{R}^n$ be the vector defined by

$$z_i = \int_{\overline{B}_n(\mathbf{0};r)} x_i R_2(x^0; x)dx, \quad i \in \{1, 2, \ldots, n\}. \tag{4.52}$$

Then the expression for $\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S)$ given in (4.49) can be written as

$$\lim_{\vec{m}\to\vec{\infty}} \nabla_s f(x^0; S) = \frac{2\pi}{V_{n+2}}(v + w + z) = \frac{2\pi}{V_{n+2}}v + \frac{2\pi}{V_{n+2}}w + \frac{2\pi}{V_{n+2}}z. \tag{4.53}$$

We now find an expression for the first two terms in (4.53).

Lemma 4.34. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^3$ on an open domain containing $\overline{B}_n(x^0; r)$. Let $v \in \mathbb{R}^n$ and $w \in \mathbb{R}^n$ be defined as in* (4.50) *and* (4.51)*. Then*

$$\frac{2\pi}{V_{n+2}}v = \nabla f(x^0) \text{ and } \frac{2\pi}{V_{n+2}}w = 0.$$

Proof. Let $g_i = [\nabla f(x^0)]_i$ for all $i \in \{1, 2, \ldots, n\}$. We have

$$\begin{aligned}\frac{2\pi}{V_{n+2}} \int_{\overline{B}_n(\mathbf{0};r)} x_i \nabla f(x^0)^\top x dx &= \frac{2\pi}{V_{n+2}} \int_{\overline{B}_n(\mathbf{0};r)} x_i^2 g_i dx + \sum_{j\neq i} \int_{\overline{B}_n(\mathbf{0};r)} x_i x_j g_i dx \\ &= \frac{2\pi}{V_{n+2}} \frac{V_n + 2}{2\pi} g_i + 0 = g_i,\end{aligned}$$

by Proposition 4.30. Therefore $\frac{2\pi}{V_{n+2}}v = \nabla f(x^0)$.

Let $\nabla^2 f(x^0) = H \in \mathbb{R}^{n\times n}$. We have

$$\begin{aligned}&\frac{\pi}{V_{n+2}} \int_{\overline{B}_n(\mathbf{0};r)} x_i x^\top \nabla^2 f(x^0) x dx \\ &= \frac{\pi}{V_{n+2}} \left(\sum_{j=1}^n \int_{\overline{B}_n(\mathbf{0};r)} x_i x_j^2 H_{j,j} dx + 2 \sum_{j=1}^n \sum_{\substack{k=1 \\ k\neq j}}^n \int_{\overline{B}_n(\mathbf{0};r)} x_i x_j x_k H_{j,k} dx \right) = 0,\end{aligned}$$

by Theorem 4.27. Therefore, $\frac{2\pi}{V_{n+2}}w = 0$. □

Next, we find an upper bound for the third term in (4.53). In Lemma 4.35 we create the redundant variable $\Delta_S = r$. This allows for easy and immediate comparison to Theorem 4.26.

Lemma 4.35. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^3$ on an open domain containing $\overline{B}_n(x^0; r)$. Let $z \in \mathbb{R}^n$ be defined as in (4.52). Let*

$$\eta = \frac{\Gamma(\frac{n+4}{2})}{\sqrt{\pi}\Gamma(\frac{n+3}{2})}.$$

Also, denote by $L_{\nabla^2 f}$ the Lipschitz constant of the Hessian on $\overline{B}_n(x^0; r)$. Let $\Delta_S = r$. Then

$$\frac{2\pi}{V_n + 2}\|z\| \leq \frac{\sqrt{n}}{3\sqrt{\pi}} L_{\nabla^2 f} \eta \Delta_S^2.$$

Proof. We have

$$\begin{aligned} |z_i| &= \left| \frac{2\pi}{V_{n+2}} \int_{\overline{B}_n(\mathbf{0};r)} x_i R_2(x^0; x) dx \right| \\ &\leq \frac{2\pi}{V_{n+2}} \int_{\overline{B}_n(\mathbf{0};r)} |x_i| \left| R_2(x^0; x) \right| dx \\ &\leq \frac{2\pi}{V_{n+2}} \frac{1}{6} L_H \Delta_S^3 \int_{\overline{B}_n(\mathbf{0};r)} |x_i| dx. \end{aligned} \tag{4.54}$$

By Proposition 4.28, we know that

$$\int_{\overline{B}_n(\mathbf{0};r)} |x_i| dx = 2\frac{r^{n+1}\Gamma(1)\Gamma(\frac{1}{2})^{n-1}}{(n+1)\Gamma\left(1 + \frac{n-1}{2}\right)} = \frac{2r^{n+1}\pi^{\frac{n-1}{2}}}{(n+1)\Gamma\left(\frac{n+1}{2}\right)}. \tag{4.55}$$

Note that

$$\frac{2}{(n+1)\Gamma\left(\frac{n+1}{2}\right)} = \frac{1}{\Gamma\left(\frac{n+3}{2}\right)}.$$

Hence, (4.55) can be written as

$$\int_{\overline{B}_n(\mathbf{0};r)} |x_i| dx = \frac{\pi^{\frac{n-1}{2}} r^{n+1}}{\Gamma\left(\frac{n+3}{2}\right)} = \frac{V_{n+1}}{\pi}. \tag{4.56}$$

Substituting (4.56) in (4.54) gives

$$\left| \frac{2\pi}{V_{n+2}} \int_{\overline{B}_n(\mathbf{0};r)} x_i R_2(x^0; x) dx \right| \leq \frac{1}{3} \frac{V_{n+1}}{V_{n+2}} L_{\nabla^2 f} \Delta_S^3. \tag{4.57}$$

The term V_{n+1}/V_{n+2} in (4.57) is

$$\frac{V_{n+1}}{V_{n+2}} = \frac{\pi^{\frac{n+1}{2}} r^{n+1}}{\Gamma\left(\frac{n+3}{2}\right)} \frac{\Gamma\left(\frac{n+4}{2}\right)}{\pi^{\frac{n+2}{2}} r^{n+2}} = \frac{1}{\sqrt{\pi}\Delta_S} \eta.$$

Thus,

$$|z_i| \leq \frac{1}{3\sqrt{\pi}} \eta L_{\nabla^2 f} \Delta_S^2, \quad \forall i \in \{1, 2, \ldots, n\}.$$

Therefore,

$$\|z\| \leq \frac{\sqrt{n}}{3\sqrt{\pi}} L_{\nabla^2 f} \eta \Delta_S^2. \qquad \square$$

Theorem 4.36 (Error bound ad infinitum for the GSG over a ball)**.** *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^3$ on an open domain containing $\overline{B}_n(x^0; r)$. Let the vectors $v, w, z \in \mathbb{R}^n$ be defined as in* (4.50)*,* (4.51)*,* (4.52)*, respectively. Let*

$$\eta = \frac{\Gamma(\frac{n+4}{2})}{\sqrt{\pi}\Gamma(\frac{n+3}{2})}.$$

Denote by $L_{\nabla^2 f}$ the Lipschitz constant of $\nabla^2 f$ on $\overline{B}_n(x^0; r)$. Let Δ_S be defined as in (4.2)*. Then*

$$\left\| \lim_{\vec{m} \to \vec{\infty}} \nabla_s f(x^0; S) - \nabla f(x^0) \right\| \leq \frac{\sqrt{n}}{3\sqrt{\pi}} L_{\nabla^2 f} \eta \Delta_S^2. \tag{4.58}$$

Proof. We have

$$\begin{aligned} \left\| \lim_{\vec{m} \to \vec{\infty}} \nabla_s f(x^0; S) - \nabla f(x^0) \right\| &= \left\| \frac{2\pi}{V_{n+2}} v + \frac{2\pi}{V_{n+2}} w + \frac{2\pi}{V_{n+2}} z - \nabla f(x^0) \right\| \\ &= \left\| \nabla f(x^0) + 0 + \frac{2\pi}{V_{n+2}} z - \nabla f(x^0) \right\| \\ &= \frac{2\pi}{V_{n+2}} \|z\| \leq \frac{\sqrt{n}}{3\sqrt{\pi}} L_{\nabla^2 f} \eta \Delta_S^2. \end{aligned}$$

$\square$

Like Theorem 4.26, notice that Theorem 4.36 immediately provides a finite error bound ad infinitum, where Theorem 4.10 does not.

Note that the error bound ad infinitum over a ball is an order-2 accurate approximation of the full gradient, which is not the case for the error bound ad infinitum defined in Section 4.3.1. This is due to the fact that, for each column $s \in S$, its opposite $-s$ is also in S as $\vec{m} \to \vec{\infty}$. Therefore, the limit of the GSG over $\overline{B}_n(x^0; r)$ is equivalent to the limit of the GCSG over a half-ball centered at x^0 of radius r. Notice that the shape of the sample region is not the key point to obtain order-2 accuracy. The position of the

point of interest x^0 is what matters. Indeed, we could get an error bound ad infinitum of accuracy $O(\Delta_S^2)$ by considering a hyperrectangle and letting x^0 be located at the intersection of all diagonals of the hyperrectangular sample region. Finally, note that the error bound in (4.58) involves the Lipschitz constant of the Hessian of f, $L_{\nabla^2 f}$. Therefore, the error bound reduces to zero whenever f is a polynomial of degree at most 2. This explains why the GSG is a perfect approximation in Example 4.33.

We conclude this section with a comparison of classical error bounds (as given by Theorems 4.10 and 4.11) as m gets large to the error bounds ad infinitum derived in Theorems 4.26 and 4.36.

Example 4.37. In this example, we consider the function $f : \mathbb{R}^2 \to \mathbb{R} : x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top \mapsto x_1^3 + x_2^3$. The point of interest is set to $x^0 = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$. Two sample regions are considered: the square $[0,1] \times [0,1]$ and the ball $\overline{B}_2(x^0; 1)$.

We set $m_1 = m_2$, so $m = (m_1)^2$. The classical error bounds are computed using Theorems 4.10 and 4.11. The error bounds ad infinitum are computed using Theorems 4.26 and 4.36. Finally, the GSG is constructed and the true absolute error is computed for both sample regions. Figure 4.4 visualizes the results for $m_1 \in \{2^2, 2^3, \ldots, 2^{10}\}$ (so, $m \in \{2^4, 2^6, \ldots, 2^{20}\}$). Note the bounds ad infinitum are independent of m_1, so constants.

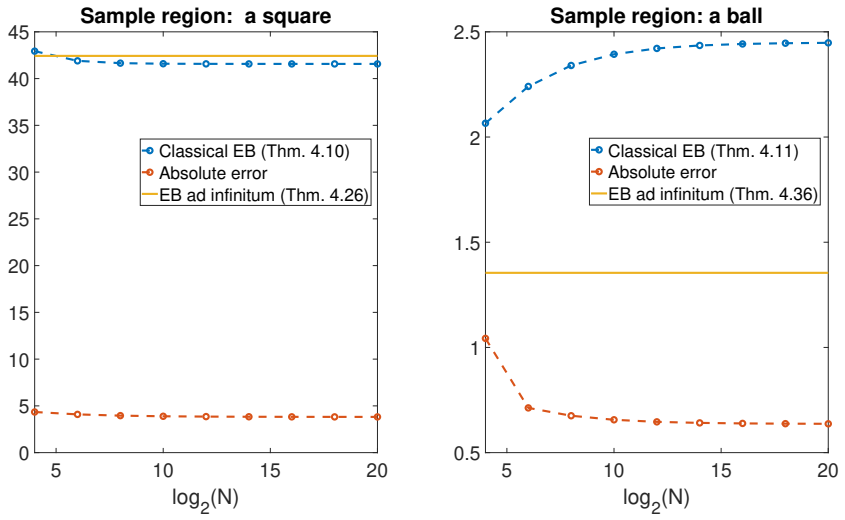


Figure 4.4: The error bound ad infinitum in $\mathbb{R}^2$ for two different sample regions

Based on this example, the error bounds ad infinitum provides an accurate upper bound as low as $m = 16$. It also appears that the error bound ad infinitum over the ball $\overline{B}_2(x^0; 1)$ provides a tighter error bound than the classical error bound for $m \geq 16$. Finally, in this example, it appears that the classical error bounds may converge as m tends to infinity as conjectured.

4.4 Summary and future research directions

In this chapter, we defined the generalized simplex gradient and the generalized centered simplex gradient. These gradient approximation techniques provide a simple explicit formula to approximate gradients. They are well-defined for any number of sample points. Error bounds that cover all four possible cases are presented for both the GSG and GCSG. It shows that the GSG is an order-1 accurate approximation of the full gradient when $SS^\dagger = \mathrm{Id}_n$ (determined and overdetermined cases) and it is an order-1 accurate approximation of a partial gradient $\mathrm{Proj}_S \nabla f(x^0)$ when $SS^\dagger \neq \mathrm{Id}_n$ (underdetermined and nondetermined). The error bound for the GCSG shows that the GCSG is an order-2 accurate approximation. When S is determined, the GCSG requires $2n$ function evaluations compared to $n + 1$ function evaluations for the GSG. The GCSG provides a higher order of accuracy than the GSG, but requires $n - 1$ more function evaluations in the determined case.

In Section 4.3, we have provided an expression for the GSG ad infinitum and an error bound ad infinitum for the GSG over both a hyperrectangle and a ball. In both cases, we note that the error bound is independent of the number of sample points, which is critical in allowing the analysis of the limits. Examining the techniques used in each case, it seems likely that an error bound ad infinitum for the GSG of f at x^0 over any reasonable sample region can be defined. However, repeating the process for every possible region is clearly an unreasonable proposition. A more practical open question is the following: given a set of sample points $\Omega \subseteq \mathbb{R}^n$ and a bijection $\mathcal{T} : \mathbb{R}^n \mapsto \mathbb{R}^n$ such that $\mathcal{T}(\Omega) = R(x^0; d)$, can the bijection be used to determine the GSG ad infinitum and an error bound ad infinitum for the GSG over Ω?

Comparing Theorems 4.26 and 4.36, we see that the position of the point of interest x^0 has an impact on the error bound. Indeed, when x^0 is the center of the sample region, then we obtained an error bound ad infinitum of $O(\Delta_S^2)$. In this case, the error bound can be viewed as an error bound for a GCSG ad infinitum. When x^0 is on the boundary of the sample region,

then we obtained an error bound ad infinitum of $O(\Delta_S)$.

In [BHJB21] it was found that under certain conditions, the limit of the classical error bound for the GSG in $\mathbb{R}$ (Theorem 4.10) could be taken directly. It is possible that this is true in $\mathbb{R}^n$ as well. However, the techniques in [BHJB21] do not adapt directly. Exploring this topic is a potential future research direction.

Chapter 5

Hessian approximation techniques

Many DFO algorithms employ approximate Hessians to solve optimization problems (see, for example, [CRV10, Kel11, Pow04a, Pow98, Pow04b, Pow04c, Pow06, Pow07, Pow08, WRS08]). In contrast to gradients, Hessians are able to capture the curvature of a function.

One of the most known optimization method using (true) Hessians is *Newton's Method* [And22, Section 4.3]. It is well-known that the rate of convergence of Newton's method is quadratic [And22, Theorem 4.3]. In DFO methods, true Hessians are not employed. To develop strong convergence results, a DFO method uses numerical analysis techniques to approximate Hessians (and gradients) in a manner that has controllable error bounds. In Chapter 4, a technique to approximate gradients called the generalized simplex gradient has been discussed. In this chapter, we develop Hessian approximation techniques and show that they have controllable error bounds. Consequently, the Hessian approximation techniques discussed in this chapter are well-suited to be employed in DFO methods.

Hessians have been used in DFO methods since at least 1970 [Win70]. Researchers have previously explored methods to approximate full Hessians or some of the entries of the Hessian. In [CV07], the authors outline an idea for a *simplex Hessian* that is constructed via quadratic interpolation through $(n+1)(n+2)/2$ well-poised sample points. They further suggested that if only a portion of the Hessian were desired (say the diagonal component), then fewer points could be used. These ideas were formalized in [CSV08b] through quadratic interpolation and analyzed through the use of Lagrange polynomials. Obtaining an approximation of the diagonal component of a Hessian is also discussed in [CT21, JB22]. It is shown that the diagonal entries can be obtained for free (in terms of function evaluations) if the gradient has been previously approximated via the centered simplex gradient technique.

In this chapter, we continue the development of these tools by introducing the *generalized simplex Hessian (GSH)* and the *generalized centered*

simplex Hessian (GCSH). The GSH is closely linked to the Hessian of a quadratic interpolation function. We will see that in certain situations, both approaches yield the same result (Section 5.4).

The main achievement of this chapter is to introduce a novel matrix-based method for constructing an approximate Hessian using only function evaluations. The method is well-defined regardless of the number of sample points used. Moreover, the method requires less computational power than interpolation-based methods and is easy to implement in matrix-based programming languages such as Matlab. As only function evaluations are required, the method is suitable for use in DFO algorithms.

The results presented in this chapter can be viewed as an extension of the work related to simplex Hessians introduced in [CSV09b, CV07]. Some advantages of the GSH compared to the approach taken in [CSV08a, CSV08b] are the following. First, the GSH provides an explicit formula to approximate Hessians that is well-defined regardless of the number of sample points utilized. Indeed, as long as the matrices of directions used to build the sample set of points are non-empty, the GSH and the GCSH are well-defined. In particular, the GSH provides an explicit formula to approximate the Hessian even when the quadratic interpolation function does not exist or is not unique. However, we note that when the matrices of directions used in the computation of the GSH have a specific structure, the GSH is equivalent to a *forward-finite-difference approximation* of the Hessian.

A second benefit of the GSH formula is that by employing a matrix structure in the definition, the Hessian approximation is extremely easy to implement in any matrix-based programming language. Moreover, after function evaluations are complete, the matrix structure requires less computational effort (in flops) than quadratic interpolation. Indeed, when S and all T_i are square matrices, then the GSH requires order $O(n^4)$ flops to compute. This is an improvement to the order $O(n^6)$ flops required to find the Hessian of the quadratic interpolation function using a set of sample points that is poised for quadratic interpolation [CSV08a] (see also [CSV09b, §6.2]).

The GSH (GCSH) provides an accurate approximation of the Hessian of an objective function $f : \mathbb{R}^n \to \mathbb{R}$ under reasonable assumptions. The technique uses a point of interest x^0 and sets of directions, stored in a set of matrices $\{S, T_1, \ldots, T_m\}$ whose columns are vectors that are added to the point of interest to determine the sample points. Defining Δ_S as the radius of S, and Δ_T as the maximum radius of the T_i, we prove that if S and all T_i are full row rank and the Hessian $\nabla^2 f$ exists and is Lipschitz continuous, then the GSH (GCSH) is an accurate estimate of $\nabla^2 f$ to within a multiple of $\Delta_u := \max\{\Delta_S, \Delta_T\}$. In terms of order-$n$ accuracy, the GSH is an order-1

accurate approximation of the full Hessian. Furthermore, the GCSH is an order-2 accurate approximation of the full Hessian. Error bounds when less than $(n+1)(n+2)/2$ sample points are used to compute the GSH (GCSH) are presented. Finally, we show that the GSH and its centered version, the GCSH, are order-1 and order-2 accurate approximation of the appropriate *partial Hessian*, respectively.

This chapter is organized as follows. In Section 5.1, fundamental background material to understand this chapter is presented. In Section 5.2, the GSH, and its "centered" version, the GCSH are introduced. The relation between the GSH and the GCSH is clarified. In 5.3, error bounds bounds for the GSH and the GCSH are presented which show that the techniques are order-1 accurate and order-2 accurate of the full Hessian, or a partial Hessian, under the appropriate assumptions. In Section 5.4, we investigate how to choose the matrices of directions involved in the computation of the GSH or the GCSH to minimize the total number of function evaluations and obtain an order-1 or order-2 accurate approximation of the full Hessian. In Section 5.5, it is shown how to choose the matrices S and $T_{1:m}$ when we are only interested by some, or all, diagonal entries of the Hessian. The relation between the GCSH and the *centered simplex Hessian diagonal* (CSHD), a technique to approximate the diagonal entries of the Hessian presented in [JB22], is clarified. It is shown that the CSHD is a special case of the GCSH. Last, how to only approximate the off-diagonal entries of the Hessian or a column (row) of the Hessian, using the GSH or the GCSH are discussed. The properties of the set of sample points that could be employed are discussed. Finally, future research directions related to approximating Hessians are presented in Section 5.6.

5.1 Preliminaries

In this section, we introduce fundamental definitions and notation that will be used throughout this chapter.

Definition 5.1 (Poised for quadratic interpolation)**.** *[AH17]* The set of distinct points $\mathcal{Y} = \{y^0, y^1, \ldots, y^m\} \subset \mathbb{R}^n$ with $m = \frac{1}{2}(n+1)(n+2) - 1$ is *poised for quadratic interpolation* if the system

$$\alpha_0 + \alpha^\top y^j + \frac{1}{2}(y^j)^\top \mathbf{H} y^j = 0, \quad j \in \{0, 1, \ldots, m\}, \tag{5.1}$$

has a unique solution for $\alpha_0 \in \mathbb{R}, \alpha \in \mathbb{R}^n$, and $\mathbf{H} = \mathbf{H}^\top \in \mathbb{R}^{n \times n}$.

When a set of sample points $\mathcal{Y}$ is poised for quadratic interpolation, it means that there exists a unique quadratic model passing through all the sample points in $\mathcal{Y}$. Therefore, the Hessian of the quadratic model defined through $\mathcal{Y}$ is unique. Note that a set of sample points containing $(n+1)(n+2)/2$ distinct sample points is not necessarily poised for quadratic interpolation. The set of sample points is poised for quadratic interpolation if and only the matrix associated to the linear system (5.1) is square and full rank. For instance, the set of sample points

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \right\} \subset \mathbb{R}^2$$

is not poised for quadratic interpolation. Indeed, the linear system associated to $\mathcal{Y}$ is

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 \\ \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \\ \frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ \alpha_1 \\ \alpha_2 \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} f(y^0) \\ f(y^1) \\ f(y^2) \\ f(y^3) \\ f(y^4) \\ f(y^5) \end{bmatrix}$$

where $\mathbf{H} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$, and $\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 \end{bmatrix}$. Since the the rank of the matrix in the previous linear system is $5 < 6$, the sample set $\mathcal{Y}$ is not poised for quadratic interpolation.

On the other hand, the set

$$\left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \right\} \subset \mathbb{R}^2$$

is poised for quadratic interpolation since the rank of the matrix associated to the system (5.1) is 6.

When the number of distinct sample points in $\mathcal{Y}$ is fewer than $(n+1)(n+2)/2$, then the quadratic interpolation model is no longer unique. This case is usually referred as the underdetermined case [CSV09b, Chapter 5]. In this case, a quadratic model may be defined through the minimum Frobenius norm problem [CSV09b, Chapter 5] (see also [CSV08b, Section 5]). When the number of distinct sample points in $\mathcal{Y}$ is greater than $(n+1)(n+2)/2$ and there exists no quadratic function that passes through all points in $\mathcal{Y}$,

the quadratic model may be determined through a least square regression problem [CSV09b, Chapter 4] (see also [CSV08b, Sections 2,3 & 4]).

To draw parallels with Chapter 4, the GSG can be viewed as the gradient of a linear interpolation function in the determined case. When fewer than $n+1$ distinct sample points are used, then the linear interpolation model is no longer unique and we have called this situation the underdetermined case. When the number of sample points is greater than $n+1$ and there exists no linear function passing through all sample points, we have called this situation the overdetermined case, and the GSG is the solution of a least square regression problem.

Definition 5.2 (Quadratic interpolation function)**.** *[AH17, Definition 9.9]* Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $\mathcal{Y} = \{y^0, y^1, \ldots, y^m\} \subset \operatorname{dom} f$ with $m = \frac{1}{2}(n+1)(n+2) - 1$ be poised for quadratic interpolation. Then the *quadratic interpolation function of f over $\mathcal{Y}$* is

$$Q_f(\mathcal{Y})(x) = \alpha_0 + \alpha^\top x + \frac{1}{2} x^\top \mathbf{H} x,$$

where $(\alpha_0, \alpha, \mathbf{H} = \mathbf{H}^\top)$ is the unique solution to the system (5.1).

In the next definition, we introduce key notation used in the construction of the GSH and the GCSH.

Definition 5.3 (GSH notation)**.** Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let

$$S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m} \quad \text{and}$$
$$T_j = \begin{bmatrix} t_j^1 & t_j^2 & \cdots & t_j^k \end{bmatrix} \in \mathbb{R}^{n \times k_j}, j \in \{1, \ldots, m\}$$

be sets of directions contained in $\mathbb{R}^n$, written in matrix form. Define

$$T_{1:m} = \{T_1, \ldots, T_m\}.$$

Assume that $x^0 \oplus T_j, x^0 + s^j, x^0 + s^j \oplus T_j$ are contained in $\operatorname{dom} f$ for all $j \in \{1, \ldots, m\}$. Define

$$\Delta_S = \max_{j \in \{1,\ldots,m\}} \|s^j\|, \quad \Delta_{T_j} = \max_{\ell \in \{1,\ldots,k_j\}} \|t_j^\ell\|, \quad \Delta_T = \max_{j \in \{1,\ldots,m\}} \Delta_{T_j},$$

$$\widehat{S} = \frac{1}{\Delta_S} S, \quad \widehat{T_j} = \frac{1}{\Delta_{T_j}} T_j, j \in \{1, \ldots, m\}, \tag{5.2}$$

and

$$\delta_s f\left(x^0; T_j\right) = \begin{bmatrix} f(x^0 + t_j^1) - f(x^0) \\ f(x^0 + t_j^2) - f(x^0) \\ \vdots \\ f(x^0 + t_j^k) - f(x^0) \end{bmatrix} \in \mathbb{R}^{k_j}.$$

In this chapter, all matrices involved are assumed to be non-null rank. This ensures that the radius of a matrix is non-zero and hence (5.2) is always well-defined. In the previous definition, note that m can be any positive integer. Furthermore, each k_j, where $j \in \{1, \ldots, m\}$, can be any positive integer.

Recall the definition of the generalized simplex gradient discussed in Chapter 4.

Definition 5.4 (Generalized simplex gradient)**.** Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S \in \mathbb{R}^{n \times m}$ with $x^0 \oplus S \subset \operatorname{dom} f$. The *generalized simplex gradient* of f at x^0 over S is denoted by $\nabla_s f(x^0; S)$ and defined by

$$\nabla_s f(x^0; S) = (S^\top)^\dagger \delta_s f(x^0; S).$$

The error bound for the GSG presented in Theorem 4.10 only requires that S is non-null rank. Hence, it covers all cases for the GSG: underdetermined, determined case, overdetermined and nondetermined. In Section 5.3, error bounds with a similar format than the one in Theorem 4.10 are developed for the GSH and the GCSH. It gets more complex as there are now two matrices of directions involved. First, we introduce the definitions of the GSH and the GCSH.

5.2 The generalized simplex Hessian and the generalized centered simplex Hessian

In this section, the GSH and the GCSH are defined. Then the relation between the GSH and GCSH is clarified. In particular, it is shown that the GCSH is equivalent to the GSH when the matrices of directions used to compute the GSH have a specific form.

The GSH requires finite sets of directions and a point of interest at which the Hessian is approximated. Similarly to the generalized simplex gradient, it involves the Moore-Penrose pseudo-inverse and a difference matrix. In

this case, the difference matrix consists of the difference between generalized simplex gradients.

Recall that, for ease of notation, we use the definition

$$T_{1:m} = \{T_1, \ldots, T_m\}$$

where convenient.

Definition 5.5 (Generalized simplex Hessian)**.** Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and $T_j \in \mathbb{R}^{n \times k_j}$ with $x^0 \oplus T_j, x^0 \oplus S, x^0 + s^j \oplus T_j$ contained in $\operatorname{dom} f$ for all $j \in \{1, \ldots, m\}$. The *generalized simplex Hessian* of f at x^0 over S and $T_{1:m}$ is denoted by $\nabla_s^2 f(x^0; S, T_{1:m})$ and defined by

$$\nabla_s^2 f(x^0; S, T_{1:m}) = (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}), \tag{5.3}$$

where

$$\delta_s^2 f(x^0; S, T_{1:m}) = \begin{bmatrix} (\nabla_s f(x^0 + s^1; T_1) - \nabla_s f(x^0; T_1))^\top \\ (\nabla_s f(x^0 + s^2; T_2) - \nabla_s f(x^0; T_2))^\top \\ \vdots \\ (\nabla_s f(x^0 + s^m; T_m) - \nabla_s f(x^0; T_m))^\top \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

In the case where $S \in \mathbb{R}^{n \times m}$ and $T_1 = T_2 = \cdots = T_m$, we define $\overline{T} = T_j$ to simplify notation and write $\nabla_s^2 f(x^0; S, \overline{T})$ to emphasize the special case.

When all matrices $T_j = \overline{T}$, we define the matrix

$$D_s^2 = \begin{bmatrix} \left(\delta_s f(x^0 + s^1; \overline{T}) - \delta_s f(x^0; \overline{T})\right)^\top \\ \vdots \\ \left(\delta_s f(x^0 + s^m; \overline{T}) - \delta_s f(x^0; \overline{T})\right)^\top \end{bmatrix} \in \mathbb{R}^{m \times k}. \tag{5.4}$$

We remark that the s in $\nabla_s^2 f, \delta_s, \delta_s^2$ and D_s^2 is for simplex. This becomes important as we now introduce the generalized centered simplex Hessian. The "square" in δ_s^2 emphasizes the fact that we are taking differences of first-order objects (hence getting a second-order object).

Definition 5.6 (Generalized centered simplex Hessian)**.** Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S \in \mathbb{R}^{n \times m}$ and $T_j \in \mathbb{R}^{n \times k_j}$ with $x^0 \oplus S \oplus T_j, x^0 \oplus (-S) \oplus (-T_j), x^0 \oplus (\pm S)$, and $x^0 \oplus (\pm T_j)$ contained in $\operatorname{dom} f$ for all $j \in \{1, \ldots, m\}$. The *generalized centered simplex Hessian of f at x^0 over S and $T_{1:m}$* is denoted by $\nabla_c^2 f(x^0; S, T_{1:m})$ and defined by

$$\nabla_c^2 f(x^0; S, T_{1:m}) = \frac{1}{2}\left(\nabla_s^2 f(x^0; S, T_{1:m}) + \nabla_s^2 f(x^0; -S, -T_{1:m})\right). \tag{5.5}$$

When all matrices $T_j = \overline{T}$, we define the matrix

$$D_c^2 = \frac{1}{2}\begin{bmatrix} \left(\delta_s f(x^0 + s^1; \overline{T}) + \delta_s f(x^0 - s^1; -\overline{T}) - \delta_s f(x^0; \overline{T}) - \delta_s f(x^0; -\overline{T})\right)^\top \\ \vdots \\ \left(\delta_s f(x^0 + s^m; \overline{T}) + \delta_s f(x^0 - s^m; -\overline{T}) - \delta_s f(x^0; \overline{T}) - \delta_s f(x^0; -\overline{T})\right)^\top \end{bmatrix} \tag{5.6}$$

in $\mathbb{R}^{m\times k}$. Note that the c in $\nabla_c^2 f$ and D_c^2 is for centered. The GCSH can be viewed as a generalization of the centered-finite difference approximation of the Hessian, also called midpoint approximation [BFB16, Section 4.1]. Similarly, The GSH can be viewed as a generalization of the forward-finite-difference approximation of the Hessian [BFB16, Section 4.1]. Indeed, when $S = T_1 = \cdots = T_m = H\,\mathrm{Id}_n$, where $H = \mathrm{Diag}\begin{bmatrix} h_1 & h_2 & \cdots & h_n \end{bmatrix}, h_i > 0$, then the GSH is equivalent to a forward-finite-difference approximation of the second-order derivatives. Moreover, the GSH can be viewed as the DFO analog of the Hessian approximation given by Taylor's Theorem in the derivative-based context, which approximates the gradient by differences of true gradients (see [NW06, Section 8.1]). An advantage of the GSH (GCSH) over forward-finite-difference approximations (centered-finite-difference approximations) is that it is less restrictive. In particular, it is well-defined as long as the matrices S and $T_1, \ldots, T_m$ are non-empty.

Remark 5.7. Consider the case where S and T_i are square matrices, i.e., $m = n$ and $k_j = n$ for all $j \in \{1, \ldots, m\}$. The computation of the GSH begins by the construction of $n + 1$ generalized simplex gradients. Setting aside function evaluations, computing a simplex gradient is dominated by computing a matrix inverse, which is $O(n^3)$ flops. Thus the construction of $\delta_s^2 f(x^0; S, T_{1:m})$ requires $O(n^4)$ flops. The inverse of S is computed ($O(n^3)$ flops) and multiplied by $\delta_s^2 f(x^0; S, T_{1:m})$, $O(n^3)$ flops. Hence, setting aside function evaluations, computing the GSH requires $O(n^4)$ flops.

Conversely, setting aside function evaluations, quadratic interpolation requires $O(n^6)$ flops to compute [CSV09b, Section 6.2].

Next we provide a formula to compute the GCSH in a format similar to the formula for the GSH (5.3).

In Lemma 5.8, we introduce the notation δ_c^2. This is a centered version of δ_s^2; hence, the s for simplex is replaced with c for centered.

Lemma 5.8. *Let $f : \mathrm{dom}\, f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \mathrm{dom}\, f$ be the point of interest. Let $S = \begin{bmatrix} s^1 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n\times m}$ and $T_j \in \mathbb{R}^{n\times k_j}$ with $x^0 \oplus S \oplus T_j, x^0 \oplus (-S) \oplus (-T_j), x^0 \oplus (\pm S), x^0 \oplus (\pm T_j)$ contained in $\mathrm{dom}\, f$ for all $j \in \{1, \ldots, m\}$. Then*

$$\nabla_c^2 f(x^0; S, T_{1:m}) = (S^\top)^\dagger \delta_c^2(x^0; S, T_{1:m})$$

where

$$\begin{aligned}&\delta_c^2 f(x^0; S, T_{1:m})\\ &= \frac{1}{2}\begin{bmatrix} \left(\delta_s f(x^0+s^1;T_1)+\delta_s f(x^0-s^1;-T_1)-\delta_s f(x^0;T_1)-\delta_s f(x^0;-T_1)\right)^\top T_1^\dagger \\ \vdots \\ \left(\delta_s f(x^0+s^m;T_m)+\delta_s f(x^0-s^m;-T_m)-\delta_s f(x^0;T_m)-\delta_s f(x^0;-T_m)\right)^\top T_m^\dagger \end{bmatrix}.\end{aligned}$$

Proof. We have

$$\begin{aligned}&\nabla_c^2 f(x^0; S, T_{1:m})\\ =&\frac{1}{2}\left(\nabla_s^2 f(x^0; S, T_{1:m}) + \nabla_s^2 f(x^0; -S, -T_{1:m})\right)\\ =&\frac{1}{2}(S^\top)^\dagger\left(\delta_s^2 f(x^0; S, T_{1:m})(x^0; S, T_{1;m}) - \delta_s^2 f(x^0; S, T_{1:m})(x^0; -S, -T_{1:m})\right)\\ =&(S^\top)^\dagger\frac{1}{2}\begin{bmatrix} \left(\nabla_s f(x^0+s^1;T_1)-\nabla_s f(x^0;T_1)\right)^\top - \left(\nabla_s f(x^0-s^1;-T_1)-\nabla_s f(x^0;-T_1)\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0+s^m;T_m)-\nabla_s f(x^0;T_m)\right)^\top - \left(\nabla_s f(x^0-s^m;-T_m)-\nabla_s f(x^0;-T_m)\right)^\top \end{bmatrix}\\ =&(S^\top)^\dagger\frac{1}{2}\begin{bmatrix} \left(\delta_s f(x^0+s^1;T_1)-\delta_s f(x^0;T_1)\right)^\top T_1^\dagger - \left(\left(\delta_s f(x^0-s^1;-T_1)-\delta_s f(x^0;-T_1)\right)^\top(-T_1)^\dagger\right) \\ \vdots \\ \left(\delta_s f(x^0+s^m;T_m)-\delta_s f(x^0;T_m)\right)^\top T_m^\dagger - \left(\left(\delta_s f(x^0-s^m;-T_m)-\delta_s f(x^0;-T_m)\right)^\top(-T_m)^\dagger\right) \end{bmatrix}\\ =&(S^\top)^\dagger\frac{1}{2}\begin{bmatrix} \left(\delta_s f(x^0+s^1;T_1)+\delta_s f(x^0-s^1;-T_1)-\delta_s f(x^0;T_1)-\delta_s f(x^0;-T_1)\right)^\top T_1^\dagger \\ \vdots \\ \left(\delta_s f(x^0+s^m;T_m)+\delta_s f(x^0-s^m;-T_m)-\delta_s f(x^0;T_m)-\delta_s f(x^0;-T_m)\right)^\top T_m^\dagger \end{bmatrix}\\ =&(S^\top)^\dagger\delta_c^2 f(x^0; S, T_{1:m}).\end{aligned}$$

□

Notice that

$$\begin{aligned}&\left[\delta_s f(x^0+s^j;T_j)+\delta_s f(x^0-s^j;-T_j)-\delta_s f(x^0;T_i)-\delta_s f(x^0;-T_j)\right]_{\ell_j}\\ =&\frac{1}{2}\big(f(x^0+s^j+t_j^{\ell_j})-f(x^0+s^j)+f(x^0-s^j-t_j^{j\ell_j})-f(x^0-s^j)\\ &-f(x^0+t_j^{\ell_j})+f(x^0)-f(x^0-t_j^{\ell_j})+f(x^0)\big)\\ =&\frac{1}{2}\big(f(x^0+s^j+t_i^{\ell_j})+f(x^0-s^j-t_j^{\ell_j})\\ &-f(x^0+s^j)-f(x^0-s^j)-f(x^0+t_j^{\ell_j})-f(x^0-t_j^{\ell_j})+2f(x^0)\big)\end{aligned}\tag{5.7}$$

for $\ell_j \in \{1,\ldots,k_j\}$ and $j \in \{1,\ldots,m\}$. We will use (5.7) in Theorem 5.14.

Next, we show that the GCSH is a special case of the GSH where the matrices of directions have a specific form. This result is similar to Proposition 4.4, which showed that the generalized centered gradient is equal to the generalized simplex gradient when the matrix of directions used to compute the generalized simplex gradient has the form $A = \begin{bmatrix} S & -S \end{bmatrix}$ for some matrix $S \in \mathbb{R}^{n \times m}$.

Proposition 5.9. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $A = \begin{bmatrix} a^1 & a^2 & \cdots & a^{2m} \end{bmatrix} = \begin{bmatrix} S & -S \end{bmatrix} \in \mathbb{R}^{n \times 2m}$ for some $S = \begin{bmatrix} s^1 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and let $B_j = T_j \in \mathbb{R}^{n \times k_j}$, and $B_{m+j} = -T_j$ for all $j \in \{1, \ldots, m\}$. Suppose that $x^0 \oplus S \oplus T_j, x^0 \oplus (-S) \oplus (-T_j), x^0 \oplus (\pm S)$, and $x^0 \oplus (\pm T_j)$ are contained in $\operatorname{dom} f$ for all $j \in \{1, \ldots, m\}$. Then*

$$\nabla_s^2 f(x^0; A, B_{1:2m}) = \nabla_c^2 f(x^0; S, T_{1:m}).$$

Proof. We have

$$\begin{aligned}
&\nabla_s^2 f(x^0; A, B_{1:2m}) \\
&= \begin{bmatrix} S^\top \\ -S^\top \end{bmatrix}^\dagger \begin{bmatrix} \left(\nabla_s f(x^0 + a^1; B_1) - \nabla_s f(x^0; B_1)\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0 + a^{2m}; B_{2m}) - \nabla_s f(x^0; B_{2m})\right)^\top \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} (S^\top)^\dagger & -(S^\top)^\dagger \end{bmatrix} \begin{bmatrix} \left(\nabla_s f(x^0 + a^1; B_1) - \nabla_s f(x^0; B_1)\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0 + a^{2m}; B_{2m}) - \nabla_s f(x^0; B_{2m})\right)^\top \end{bmatrix} \\
&= \frac{1}{2} (S^\top)^\dagger \begin{bmatrix} \left(\nabla_s f(x^0 + s^1; T_1) - \nabla_s f(x^0; T_1)\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0 + s^m; T_m) - \nabla_s f(x^0; T_m)\right)^\top \end{bmatrix} \\
&\qquad + \frac{1}{2} (-S^\top)^\dagger \begin{bmatrix} \left(\nabla_s f(x^0 - s^1; -T_1) - \nabla_s f(x^0; -T_1)\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0 - s^m; -T_m) - \nabla_s f(x^0; -T_m)\right)^\top \end{bmatrix} \\
&= \frac{1}{2} \left(\nabla_s^2 f(x^0; S, T_{1:m}) + \nabla_s^2 f(x^0; -S, -T_{1:m})\right) \\
&= \nabla_c^2 f(x^0; S, T_{1:m}).
\end{aligned}$$

□

We conclude this section by defining the possible cases to classify a GSH or a GCSH. Since there are more than one matrix of directions involved in the computation of the GSH (GCSH), the matrix of directions S or the set of matrices $T_{1:m}$ is specified when defining the different cases.

Definition 5.10. Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and $T_j \in \mathbb{R}^{n \times k_j}$ with $x^0 \oplus S \oplus T_j, x^0 \oplus (-S) \oplus (-T_j), x^0 \oplus (\pm S)$, and $x^0 \oplus (\pm T_j)$ contained in $\operatorname{dom} f$ for all $j \in \{1, \ldots, m\}$. Assume that all matrices are non-null rank. We define the following four cases to characterize the matrix $S \in \mathbb{R}^{n \times m}$ and the set of matrices $T_{1:m} = \{T_1, \ldots, T_m\}$.

- Underdetermined: the GSH (GCSH) is said to be *S-underdetermined* if S is non-square and full column rank. We say that it is *$T_{1:m}$-underdetermined* if all matrices in the set $T_{1:m}$ are full column rank and at least one matrix is non-square.
- Determined: the GSH (GCSH) is said to be *S-determined* if S is square and full rank. It is *$T_{1:m}$-determined* if all matrices in the set $T_{1:m}$ are square and full rank.
- Overdetermined: the GSH (GCSH) is said to be *S-overdetermined* if S is non-square and full row rank. It is *$T_{1:m}$-overdetermined* if all matrices in the set $T_{1:m}$ are full row rank and at least one is non-square.
- Nondetermined: the GSH (GCSH) is said to be *S-nondetermined* if it is not in any of the previous three cases. It is *$T_{1:m}$-nondetermined* if the set $T_{1:m}$ is not in any of the previous three cases.

In the special case where all matrices T_i are equal, we may write $\overline{T}$- instead of $T_{1:m}$-. Note that the definition of an S-underdetermined GSH (GCSH) implies that $\operatorname{span}(S) \neq \mathbb{R}^n$, which is true if and only if $SS^\dagger \neq \operatorname{Id}_n$. Similarly, the definition of a $T_{1:m}$-underdetermined GSH (GCSH) implies that $\operatorname{span}(T_j) \neq \mathbb{R}^n$ for some $j \in \{1, \ldots, m\}$, which is true if and only if $T_j T_j^\dagger \neq \operatorname{Id}_n$ for some j.

Defining underdetermined, determined, overdetermined and nondetermined as above creates 16 different cases to classify a GSH (GCSH), all of which are investigated in Section 5.3

When $SS^\dagger \neq \operatorname{Id}_n$ or $T_j T_j^\dagger \neq \operatorname{Id}_n$ for some $j \in \{1, 2, \ldots, m\}$, then it is not possible to define an error bound between the GSH (GCSH) and the full true Hessian. The following example illustrates this claim.

Example 5.11 (An $T_{1:m}$-underdetermined GSH)**.** Let

$$f(x, y) = x^2 + y^2 + \alpha xy$$

where $\alpha \in \mathbb{R}$ and let $x^0 = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$. Let $S = h \cdot \begin{bmatrix} e^1 & e^2 \end{bmatrix}$ where $h \neq 0$, and $T_1 = he^1, T_2 = he^2$. Regardless of the value of h and α, $\nabla_s^2 f(x^0; S, T_{1:2})$ is

equal to $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$. The error bound is

$$\left\|\nabla_s^2 f(x^0; S, T_{1:2}) - \nabla^2 f(x^0)\right\| = \left\| \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} - \begin{bmatrix} 2 & \alpha \\ \alpha & 2 \end{bmatrix} \right\| = \alpha.$$

If we let $\alpha \to \infty$, then the previous error bound tends to infinity.

In the previous example, even though the off-diagonal entries of the GSH are inaccurate, the diagonal entries of the GSH are perfectly accurate. This shows that a GSH may contain accurate approximations of some of the entries of the true Hessian. It turns out that error bounds may be defined between the GSH (GCSH) and some of the entries of the true Hessian. The appropriate entries of the true Hessian are obtained via a projection operator. The projection operator involves all matrices of directions utilized to compute the GSH (GCSH).

Given matrices $S \in \mathbb{R}^{n\times m}$ and $T_j \in \mathbb{R}^{n\times k_j}$, the projection of the matrix $H \in \mathbb{R}^{n\times n}$ onto S and $T_{1:m}$ is denoted by $\text{Proj}_{S,T_{1:m}} H$ and defined by

$$\text{Proj}_{S,T_{1:m}} H = \sum_{j=1}^{m} (S^\top)^\dagger e_m^j (e_m^j)^\top S^\top H T_j T_j^\dagger.$$

In the case where $T_1 = T_2 = \cdots = T_m = \overline{T}$, the projection of H onto S and $\overline{T}$ is denoted by $\text{Proj}_{S,\overline{T}} H$, and reduces to

$$\begin{aligned} \text{Proj}_{S,\overline{T}} H &= \sum_{j=1}^{m} (S^\top)^\dagger e_m^j (e_m^j)^\top S^\top H \overline{T}\overline{T}^\dagger \\ &= (S^\top)^\dagger \left(\sum_{j=1}^{m} e_m^j (e_m^j)^\top \right) S^\top H \overline{T}\overline{T}^\dagger \\ &= (S^\top)^\dagger \text{Id}_m S^\top H \overline{T}\overline{T}^\dagger \\ &= (S^\top)^\dagger S^\top H \overline{T}\overline{T}^\dagger. \end{aligned}$$

Note that $\text{Proj}_{S,T_{1:m}}$ is a linear operator. We are now ready to discuss the error bounds for the GSH and the GCSH.

5.3 Error bounds

In this section, we provide error bounds for the GSH and the GCSH. These error bounds are separated in two categories: the general case where

all matrices T_j are not necessarily equal, and the special case where all matrices $T_j = \overline{T}$ are equal.

When defining error bounds, the matrix L in Definition 3.4 will be equal to $\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)$. Note that when S is full row rank and T_j is full row rank for all $j \in \{1, \ldots, m\}$, then $\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)$ is equal to the true Hessian $\nabla^2 f(x^0)$. When

$$\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0) \neq \nabla^2 f(x^0),$$

we sometimes refer to $\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)$ as a *partial Hessian*. We will see that we can obtain order-1 accuracy, or order-2 accuracy, of the partial Hessian $\text{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m})$ under the appropriate assumptions.

The error bounds presented in this chapter share similarities to those presented in [CSV08a, CSV08b]. In those papers, the authors provide error bounds for the Hessian of a fully quadratic model [CSV08a, Theorem 3], the Hessian of a quadratic regression model [CSV08b, Theorem 3.2], and the Hessian of a quadratic undetermined model [CSV08b, Theorem 5.12].

Before introducing the error bounds, we note that in certain situations, the projection of the GSH (GCSH) onto S and $T_{1:m}$ is equal to the GSH (GCSH).

Proposition 5.12. *Let $f : \text{dom}\, f \subseteq \mathbb{R}^n \to \mathbb{R}$ and $x^0 \in \text{dom}\, f$. Let $S = \begin{bmatrix} s^1 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$, and $T_j \in \mathbb{R}^{n \times k_j}$. Assume that $x^0 \oplus (\pm S), x^0 \oplus (\pm T_j)$, and $x^0 \oplus (\pm(S \oplus T_j))$ are contained in $\text{dom}\, f$ for all j. Then the following hold.*

(i) Let C be defined as $C = S^\top (S^\top)^\dagger \in \mathbb{R}^{m \times m}$. Let j be any index in $\in \{1, \ldots, m\}$, Let L_j be the index set containing all indices $\ell_j \in \{1, \ldots, m\}$ such that $C_{j,\ell_j} \neq 0$. If $T_j T_j^\dagger = T_{\ell_j} T_{\ell_j}^\dagger$ for all $\ell_j \in L_j$, then

$$\text{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) = \nabla_s^2 f(x^0; S, T_{1:m}) \tag{5.8}$$

and

$$\text{Proj}_{S,T_{1:m}} \nabla_c^2 f(x^0; S, T_{1:m}) = \nabla_c^2 f(x^0; S, T_{1:m}). \tag{5.9}$$

(ii) If S is full column rank **or** *T_j is full row rank for all $j \in \{1, \ldots, m\}$, then*

$$\text{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) = \nabla_s^2 f(x^0; S, T_{1:m}) \tag{5.10}$$

and

$$\text{Proj}_{S,T_{1:m}} \nabla_c^2 f(x^0; S, T_{1:m}) = \nabla_c^2 f(x^0; S, T_{1:m}). \tag{5.11}$$

(iii) If $T_1 = T_2 = \cdots = T_m = \overline{T}$, *then*

$$\mathrm{Proj}_{S,\overline{T}} \nabla_s^2 f(x^0; S, \overline{T}) = \nabla_s^2 f(x^0; S, \overline{T}) \tag{5.12}$$

and

$$\mathrm{Proj}_{S,\overline{T}} \nabla_c^2 f(x^0; S, \overline{T}) = \nabla_c^2 f(x^0; S, \overline{T}). \tag{5.13}$$

Proof. We have

$$\mathrm{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) = \sum_{j=1}^{m} (S^\top)^\dagger e_m^j e_m^j S^\top (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) T_j T_j^\dagger.$$

Let $C = \begin{bmatrix} c^1 & \cdots & c^m \end{bmatrix} = S^\top (S^\top)^\dagger \in \mathbb{R}^{m \times m}$. Note that C is symmetric. We have

$$\begin{aligned} \mathrm{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) &= (S^\top)^\dagger \sum_{j=1}^{m} \mathrm{Diag}(e_m^j) C \delta_s^2 f(x^0; S, T_{1:m}) T_j T_j^\dagger \\ &= (S^\top)^\dagger \sum_{j=1}^{m} \begin{bmatrix} \mathbf{0}_{j-1 \times n} \\ (c^j)^\top \delta_s^2 f(x^0; S, T_{1:m}) T_j T_j^\dagger \\ \mathbf{0}_{m-j \times n} \end{bmatrix}. \end{aligned}$$

Notice that

$$\begin{aligned} &(c^j)^\top \delta_s^2 f(x^0; S, T_{1:m}) (x^0; S, T_{1:m}) T_j T_j^\dagger \\ &= (c^j)^\top \begin{bmatrix} \left((\delta_s f(x^0 + s^1; T_1) - \delta_s f(x^0; T_1) \right)^\top T_1^\dagger T_j T_j^\dagger \\ \vdots \\ \left((\delta_s f(x^0 + s^m; T_m) - \delta_s f(x^0; T_m) \right)^\top T_m^\dagger T_j T_j^\dagger \end{bmatrix}. \end{aligned}$$

For all $\ell_j \in L_j$, we have that $T_{\ell_j} T_{\ell_j}^\dagger = T_j T_j^\dagger$. Hence, for all $\ell_j \in L_j$, we have

$$\begin{aligned} T_{\ell_j}^\dagger T_j T_j^\dagger &= T_{\ell_j}^\dagger \left(T_j T_j^\dagger \right) \\ &= T_{\ell_j}^\dagger (T_{\ell_j} T_{\ell_j}^\dagger) \\ &= T_{\ell_j}^\dagger, \end{aligned}$$

by Property (ii) of the Moore-Penrose pseudo-inverse (Definition 3.1). It follows that

$$\begin{aligned}
&\mathrm{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) \\
&= (S^\top)^\dagger \sum_{j=1}^{m} \begin{bmatrix} \mathbf{0}_{j-1\times n} \\ (c^j)^\top \delta_s^2 f(x^0; S, T_{1:m}) f(x^0; S, T_{1:m}) \\ \mathbf{0}_{m-j\times n} \end{bmatrix} \\
&= (S^\top)^\dagger C \delta_s^2 f(x^0; S, T_{1:m}) \\
&= (S^\top)^\dagger S^\top (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) \\
&= (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) = \nabla_s^2 f(x^0; S, T_{1:m}).
\end{aligned}$$

Equation (5.9) follows from the fact that

$$\nabla_c^2 f(x^0; S, T_{1:m}) = \frac{1}{2}\left(\nabla_s^2 f(x^0; S, T_{1:m}) + \nabla_s^2 f(x^0; -S, -T_{1:m}\right)$$

and $\mathrm{Proj}_{S,T_{1:m}}$ is a linear operator.

Now, suppose S is full column rank. Then $C = S^\top (S^\top)^\dagger = \mathrm{Id}_m$. Hence, the result follows from Item (i). Equation (5.11) follows analogously.

Now, suppose that T_j is full row rank for all $\in \{1, \ldots, m\}$. Then we know $T_j T_j^\dagger = \mathrm{Id}_n$ for all $j \in \{1, \ldots, m\}$. We obtain

$$\begin{aligned}
\mathrm{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) &= \sum_{j=1}^{m} (S^\top)^\dagger e_m^j (e_m^j)^\top S^\top \nabla_s^2 f(x^0; S, T_{1:m}) T_j T_j^\dagger \\
&= \sum_{j=1}^{m} (S^\top)^\dagger e_m^j (e_m^j)^\top S^\top \nabla_s^2 f(x^0; S, T_{1:m}) \\
&= (S^\top)^\dagger \left(\sum_{j=1}^{m} e_m^j (e_m^j)^\top \right) S^\top \nabla_s^2 f(x^0; S, T_{1:m}) \\
&= (S^\top)^\dagger S^\top (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) \\
&= (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) = \nabla_s^2 f(x^0; S, T_{1:m}).
\end{aligned}$$

Equation (5.11) follows analogously.

Finally, suppose $T_1 = T_2 = \cdots = T_m = \overline{T}$. We have

$$
\begin{aligned}
&\mathrm{Proj}_{S,\overline{T}} \nabla_s^2 f(x^0; S, \overline{T}) \\
&= (S^\top)^\dagger S^\top \nabla_s^2 f(x^0; S, \overline{T}) \overline{T}\overline{T}^\dagger \\
&= (S^\top)^\dagger S^\top (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) \overline{T}\overline{T}^\dagger \\
&= (S^\top)^\dagger S^\top (S^\top)^\dagger \begin{bmatrix} \left(\nabla_s f(x^0 + s^1; \overline{T}) - \nabla_s f(x^0; \overline{T})\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0 + s^m; \overline{T}) - \nabla_s f(x^0; \overline{T})\right)^\top \end{bmatrix} \overline{T}\overline{T}^\dagger \\
&= (S^\top)^\dagger S^\top (S^\top)^\dagger \begin{bmatrix} \left((\overline{T}^\top)^\dagger \delta_s f(x^0 + s^1; \overline{T}) - (\overline{T}^\top)^\dagger \delta_s f(x^0; \overline{T})\right)^\top \\ \vdots \\ \left((\overline{T}^\top)^\dagger \delta_s f(x^0 + s^m; \overline{T}) - (\overline{T}^\top)^\dagger \delta_s f(x^0; \overline{T})\right)^\top \end{bmatrix} \overline{T}\overline{T}^\dagger \\
&= (S^\top)^\dagger S^\top (S^\top)^\dagger \begin{bmatrix} \left(\delta_s f(x^0 + s^1; \overline{T}) - \delta_s f(x^0; \overline{T})\right)^\top \\ \vdots \\ \left(\delta_s f(x^0 + s^m; \overline{T}) - \delta_s f(x^0; \overline{T})\right)^\top \end{bmatrix} \overline{T}^\dagger \overline{T}\overline{T}^\dagger \\
&= (S^\top)^\dagger \begin{bmatrix} \left(\delta_s f(x^0 + s^1; \overline{T}) - \delta_s f(x^0; \overline{T})\right)^\top \\ \vdots \\ \left(\delta_s f(x^0 + s^m; \overline{T}) - \delta_s f(x^0; \overline{T})\right)^\top \end{bmatrix} \overline{T}^\dagger
\end{aligned}
$$

by Property (ii) of the Moore-Penrose pseudo-inverse. Therefore, we get $\mathrm{Proj}_{S,\overline{T}} \nabla_s^2 f(x^0; S, \overline{T}) = \nabla_s^2 f(x^0; S, \overline{T})$. Equation (5.13) follows analogously. □

In the previous proposition, Item (ii) fully covers 12 out of the 16 possible cases defined to classify the GSG (GCSG) (Definition 5.10). The four cases not covered by Item (ii) are S-nondetermined/overdetermined and $T_{1:m}$-nondetermined/underdetermined. These cases are (partially) covered in Item (i). Indeed, Proposition 5.12 Item (i) provides a sufficient condition for a GSH (GCSH) which is S-nondetermined/ overdetermined and $T_{1:m}$-nondetermined/ underdetermined to be equal to its projection. In the simpler situation where all matrices T_i are equal, Item (iii) fully covers all possible 16 cases for the GSH (GCSH).

In the following theorem and the remainder of this chapter, we use the

notation

$$\begin{aligned}
\Delta_u &= \max\{\Delta_S, \Delta_{T_1}, \ldots, \Delta_{T_m}\},\\
\Delta_l &= \min\{\Delta_S, \Delta_{T_1}, \ldots, \Delta_{T_m}\},\\
\Delta_T &= \max\{\Delta_{T_1}, \ldots, \Delta_{T_m}\}\\
\widehat{T} &= \widehat{T}_j \quad \text{such that} \quad \left\|\widehat{T}_j^{\dagger}\right\| \quad \text{is maximal,} \quad j \in \{1, \ldots, m\},\\
k &= \max\{k_1, \ldots, k_m\},\\
H &= \nabla^2 f(x^0).
\end{aligned}$$

Theorem 5.13 (Error bounds for the GSH)**.** *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^3$ on $B_n(x^0; \overline{\Delta})$ where $x^0 \in \operatorname{dom} f$ is the point of interest and $\overline{\Delta} > 0$. Denote by $L_{\nabla^2 f} \geq 0$ the Lipschitz constant of $\nabla^2 f$ on $\overline{B}_n(x^0; \overline{\Delta})$. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and $T_j = \begin{bmatrix} t_j^1 & t_j^2 & \cdots & t_j^{k_j} \end{bmatrix} \in \mathbb{R}^{n \times k_j}$ for all $j \in \{1, \ldots, m\}$. Assume that $B_n(x^0 + s^j; \Delta_{T_j}) \subset B_n(x^0; \overline{\Delta})$ for all $j \in \{1, \ldots m\}$. Then*

(i)

$$\begin{aligned}
&\|\nabla_s^2 f(x^0; S, T_{1:m}) - \operatorname{Proj}_{S, T_{1:m}} \nabla^2 f(x^0)\|\\
&\qquad \leq 4m\sqrt{k} L_{\nabla^2 f} \|(\widehat{S}^{\top})^{\dagger}\| \|\widehat{T}^{\dagger}\| \left(\frac{\Delta_u}{\Delta_l}\right)^2 \Delta_u. \qquad (5.14)
\end{aligned}$$

(ii) If the conditions in Proposition 5.12 Item (i) *or* (ii) *hold, then*

$$\begin{aligned}
&\|\operatorname{Proj}_{S, T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) - \operatorname{Proj}_{S, T_{1:m}} \nabla^2 f(x^0)\|\\
&= \|\nabla_s^2 f(x^0; S, T_{1:m}) - \operatorname{Proj}_{S, T_{1:m}} \nabla^2 f(x^0)\|\\
&\leq 4m\sqrt{k} L_{\nabla^2 f} \|(\widehat{S}^{\top})^{\dagger}\| \|\widehat{T}^{\dagger}\| \left(\frac{\Delta_u}{\Delta_l}\right)^2 \Delta_u. \qquad (5.15)
\end{aligned}$$

(iii) If $T_1 = T_2 = \cdots = T_m = \overline{T}$, then

$$\begin{aligned}
&\|\operatorname{Proj}_{S, \overline{T}} \nabla_s^2 f(x^0; S, \overline{T}) - \operatorname{Proj}_{S, \overline{T}} \nabla^2 f(x^0)\|\\
&\qquad = \|\nabla_s^2 f(x^0; S, \overline{T}) - \operatorname{Proj}_{S, \overline{T}} \nabla^2 f(x^0)\|\\
&\qquad \leq 4\sqrt{mk} L_{\nabla^2 f} \frac{\Delta_u}{\Delta_l} \left\|(\widehat{S}^{\top})^{\dagger}\right\| \left\|\widehat{T}^{\dagger}\right\| \Delta_u. \qquad (5.16)
\end{aligned}$$

Proof. We have

$$
\begin{aligned}
&\|\nabla_s^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} H\| \\
&= \left\| (S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) - \sum_{j=1}^m (S^\top)^\dagger e_m^j (e_m^j)^\top S^\top H T_j T_j^\dagger \right\| \\
&\leq \frac{1}{\Delta_S} \left\| (\widehat{S}^\top)^\dagger \right\| \left\| \delta_s^2 f(x^0; S, T_{1:m}) - \sum_{j=1}^m (s^j)^\top H T_j T_j^\dagger \right\| \\
&= \frac{1}{\Delta_S} \left\| (\widehat{S}^\top)^\dagger \right\| \left\| \sum_{j=1}^m (e_m^j)^\top \delta_s^2 f(x^0; S, T_{1:m}) - (s^j)^\top H T_j T_j^\dagger \right\| \\
&\leq \sum_{j=1}^m \frac{1}{\Delta_S \Delta_{T_j}} \left\| (\widehat{S}^\top)^\dagger \right\| \left\| \widehat{T}_j^{\,\dagger} \right\| \left\| \big(\delta_s f(x^0 + s^j; T_j) - \delta_s f(x^0; T_j)\big)^\top - (s^j)^\top H T_j \right\|.
\end{aligned}
\tag{5.17}
$$

Let $u_j = \big(\delta_s f(x^0 + s^j; T_j) - \delta_s f(x^0; T_j)\big)^\top \in \mathbb{R}^{k_j}$ for all $j \in \{1, \ldots, m\}$. Next we find a bound for each $\left[u_j - (s^j)^\top H T_j\right]_{\ell_j}, \ell_j \in \{1, \ldots, k_j\}$. Note that

$$
\begin{aligned}
&\left[u_j - (s^j)^\top H T_j\right]_{\ell_j} \\
&\qquad = f(x^0 + s^j + t_j^{\ell_j}) - f(x^0 + s^j) - f(x^0 + t_j^{\ell_j}) + f(x^0) - (s^j)^\top H t_j^{\ell_j}.
\end{aligned}
$$

Each function value $f(x^0 + s^j + t_j^{\ell_j}), f(x^0 + s^j)$, and $f(x^0 + t_j^{\ell_j})$, can be written as a second-order Taylor expansion about x^0 plus a remainder term $R_2(x^0; \cdot)$. Hence, we obtain

$$
\begin{aligned}
&\left| f(x^0 + s^j + t_i^{\ell_j}) - f(x^0 + s^j) - f(x^0 + t_j^{\ell_j} + f(x^0) - (s^j)^\top H T_j^{\ell_j} \right| \\
&\leq |R_2(x^0; s^j + t_j^{\ell_j})| + |R_2(x^0; s^j)| + |R_2(x^0; t_j^{\ell_j})| \\
&\leq \frac{1}{6} L_{\nabla^2 f} \|s^j + t_j^{\ell_j}\|^3 + \frac{1}{6} L_{\nabla^2 f} \|s^j\|^3 + \frac{1}{6} L_{\nabla^2 f} \|t_j^{\ell_j}\|^3 \\
&\leq \frac{1}{2} L_{\nabla^2 f} (\Delta_S + \Delta_{T_j})^3.
\end{aligned}
\tag{5.18}
$$

Plugging (5.18) in (5.17), we obtain

$$\begin{aligned}
&\|\nabla_s^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} H\| \\
&\leq \sum_{j=1}^{m} \frac{1}{\Delta_S \Delta_{T_j}} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}_j^{\,\dagger}\right\| \frac{\sqrt{k_j}}{2} L_{\nabla^2 f} (\Delta_S + \Delta_{T_j})^3 \\
&\leq \sum_{j=1}^{m} \frac{1}{\Delta_l^2} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}^\dagger\right\| \frac{\sqrt{k}}{2} L_{\nabla^2 f} (2\Delta_u)^3 \\
&= 4m\sqrt{k} L_{\nabla^2 f} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}^\dagger\right\| \left(\frac{\Delta_u}{\Delta_l}\right)^2 \Delta_u.
\end{aligned}$$

Now, suppose that $T_1 = T_2 = \cdots = T_m = \overline{T}$. The equality follows from Proposition 5.12. We have

$$\begin{aligned}
\|\nabla_s^2 f(x^0; S, \overline{T}) - \mathrm{Proj}_{S,\overline{T}} H\| &= \left\|(S^\top)^\dagger \delta_s^2 f(x^0; S, T_{1:m}) - (S^\top)^\dagger S^\top H \overline{T}\overline{T}^\dagger\right\| \\
&\leq \frac{1}{\Delta_S} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\delta_s^2 f(x^0; S, T_{1:m}) - S^\top H \overline{T}\overline{T}^\dagger\right\| \\
&\leq \frac{1}{\Delta_S \Delta_T} \|(\widehat{S}^\top)^\dagger\| \|\widehat{T}^\dagger\| \|\|D_s^2 - S^\top H \overline{T}\|, \quad (5.19)
\end{aligned}$$

where D_s^2 is defined as in (5.4). Each entry $\left[D_s^2 - S^\top H \overline{T}\right]_{j,\ell}$ has the previous bound obtained in (5.18). Since

$$\|D_s^2 - S^\top H \overline{T}\| \leq \|D_s^2 - S^\top H \overline{T}\|_F,$$

using (5.18) in (5.19), we get the inequality

$$\begin{aligned}
\|\nabla_s^2 f(x^0; S, \overline{T}) - \mathrm{Proj}_{S,\overline{T}} H\| &\leq \frac{\sqrt{mk}}{2} L_{\nabla^2 f} \frac{(\Delta_S + \Delta_T)^3}{\Delta_S \Delta_T} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}^\dagger\right\| \\
&\leq 4\sqrt{mk} L_{\nabla^2 f} \frac{\Delta_u}{\Delta_l} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}^\dagger\right\| \Delta_u.
\end{aligned}$$

□

The previous error bounds show that the GSH is an order-1 accurate approximation of a partial Hessian, or an order-1 accurate approximation of the full Hessian whenever the GSH is S-determined/overdetermined and $T_{1:m}$-determined/overdetermined (equivalently, $(S^\top)^\dagger S^\top = \mathrm{Id}_n$ and $T_j T_j^\dagger = \mathrm{Id}_n$ for all j). The ratio $\frac{\Delta_u}{\Delta_l}$ suggests that the radii of the matrices should be taken to be of the same magnitude. If one of the radius is decreased, it

suggests that all radii should be decreased in the same ratio. The presence of m and k in the error bounds roughly suggests that there is no advantage of having S-overdetermined or $T_{1:m}$-overdetermined. This topic has been investigated in Section 4.3 for the generalized simplex gradient. Finally, the presence of the Lipschitz constant $L_{\nabla^2 f}$ indicates that the GSH is perfectly accurate if f is a polynomial of degree 2 or less.

When all matrices T_j are not necessarily equal, (5.14) provides an error bound between the GSH and $\operatorname{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)$ that covers all 16 possible cases. When all matrices T_j are equal, (5.16) covers all 16 possible cases.

Theorem 5.14 (Error bounds for the GCSH)**.** *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^4$ on $B_n(x^0; \overline{\Delta})$ where $x^0 \in \operatorname{dom} f$ is the point of interest and $\overline{\Delta} > 0$. Denote by $L_{\nabla^3 f}$ the Lipschitz constant of $\nabla^3 f$ on $\overline{B}_n(x^0; \overline{\Delta})$. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$, $T_j = \begin{bmatrix} t_j^1 & t_j^2 & \cdots & t_j^{k_j} \end{bmatrix} \in \mathbb{R}^{n \times k_j}$ with the ball $B_n(x^0 + s^j; \Delta_{T_j}) \subset B_n(x^0; \overline{\Delta})$ for all $j \in \{1, \ldots, m\}$. Then*

(i)

$$\begin{aligned} &\|\nabla_c^2 f(x^0; S, T_{1:m}) - \operatorname{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)\| \\ &\quad \leq 2m\sqrt{k} L_{\nabla^3 f} \left(\frac{\Delta_u}{\Delta_l}\right)^2 \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\left(\widehat{T}\right)^\dagger\right\| \Delta_u^2. \end{aligned} \tag{5.20}$$

(ii) If the conditions in Proposition 5.12 Item (i) *or* (ii) *hold, then*

$$\begin{aligned} &\|\operatorname{Proj}_{S,T_{1:m}} \nabla_c^2 f(x^0; S, T_{1:m}) - \operatorname{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)\| \\ &\quad = \|\nabla_c^2 f(x^0; S, T_{1:m}) - \operatorname{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)\| \\ &\quad \leq 2m\sqrt{k} L_{\nabla^3 f} \left(\frac{\Delta_u}{\Delta_l}\right)^2 \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\left(\widehat{T}\right)^\dagger\right\| \Delta_u^2. \end{aligned} \tag{5.21}$$

(iii) If $T_1 = T_2 = \cdots = T_m = \overline{T} \in \mathbb{R}^{n \times k}$, then

$$\begin{aligned} &\left\|\operatorname{Proj}_{S,\overline{T}} \nabla_c^2 f(x^0; S, \overline{T}) - \operatorname{Proj}_{S,\overline{T}} \nabla^2 f(x^0)\right\| \\ &\quad = \left\|\nabla_c^2 f(x^0; S, \overline{T}) - \operatorname{Proj}_{S,\overline{T}} \nabla^2 f(x^0)\right\| \\ &\quad \leq 2\sqrt{mk} L_{\nabla^3 f} \frac{\Delta_u}{\Delta_l} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\left(\widehat{T}\right)^\dagger\right\| \Delta_u^2. \end{aligned} \tag{5.22}$$

Proof. We have

$$
\begin{aligned}
&\left\|\nabla_c^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} H\right\| \\
&= \left\| (S^\top)^\dagger \delta_c^2 f(x^0; S, T_{1:m}) - \sum_{j=1}^m (S^\top)^\dagger e_m^j (e_m^j)^\top S^\top H T_j T_j^\dagger \right\| \\
&\le \frac{1}{\Delta_S} \left\| (\widehat{S}^\top)^\dagger \right\| \left\| \delta_c^2 f(x^0; S, T_{1:m}) - \sum_{j=1}^m (s^j)^\top H T_j T_j^\dagger \right\| \\
&= \frac{1}{\Delta_S} \left\| (\widehat{S}^\top)^\dagger \right\| \left\| \sum_{j=1}^m (e_m^j)^\top \delta_c^2 f(x^0; S, T_{1:m}) - (s^j)^\top H T_j T_j^\dagger \right\| \\
&\le \sum_{j=1}^m \frac{1}{\Delta_S \Delta_{T_j}} \left\| (\widehat{S}^\top)^\dagger \right\| \left\| \widehat{T}_j^{\,\dagger} \right\| \left\| \frac{1}{2} \left(\delta_s f(x^0 + s^i; T_i) + \delta_s f(x^0 - s^i; -T_i) - \delta_s f(x^0; T_i) - \delta_s f(x^0; -T_i) \right)^\top - (s^i)^\top H T_i \right\|.
\end{aligned}
\tag{5.23}
$$

Let

$$
w_j = \frac{1}{2} \left(\delta_s f(x^0 + s^j; T_j) + \delta_s f(x^0 - s^j; -T_j) - \delta_s f(x^0; T_j) - \delta_s f(x^0; -T_j) \right)^\top
$$

in $\mathbb{R}^{k_j}$. Next we find a bound for each entry $\left[w_j - (s^j)^\top H T_j \right]_{\ell_j}$, for $\ell_j \in \{1, \ldots, k_j\}$. Using (5.7), we know that

$$
\begin{aligned}
&\left[w_j - (s^j)^\top H T_j \right]_{\ell_j} \\
=&\frac{1}{2} \left(f(x^0 + s^j + t_j^{\ell_j}) + f(x^0 - s^j - t_j^{\ell_j}) - f(x^0 + s^j) - f(x^0 - s^j) - f(x^0 + t_j^{\ell_j}) - f(x^0 - t_j^{\ell_j}) + 2f(x^0) \right) \\
&- (s^j)^\top H t_j^{\ell_j}.
\end{aligned}
\tag{5.24}
$$

Each function value $f(x^0 + s^j + t_j^{\ell_j}), f(x^0 - s^j - t_j^{\ell_j}), f(x^0 + s^j), f(x^0 - s^j), f(x^0 + t_j^{\ell_j})$ and $f(x^0 - t_j^{\ell_j})$ can be written as a third-order Taylor expansion about x^0 plus a remainder term $R_3(x^0; \cdot)$. It follows that

$$
\begin{aligned}
&\left| \frac{1}{2} \left(f(x^0 + s^j + t_j^{\ell_j}) + f(x^0 - s^j - t_j^{\ell_j}) - f(x^0 + s^j) - f(x^0 - s^j) - f(x^0 + t_j^{\ell_j}) - f(x^0 - t_j^{\ell_j}) + 2f(x^0) \right) - (s^j)^\top H t_j^{\ell_j} \right| \\
&\le \frac{1}{2} \left(\|R_3(x^0 + s^j + t_j^{\ell_j})\| + \|R_3(x^0 - s^j - t_j^{\ell_j})\| + \|R_3(x^0 + s^j)\| + \|R_3(x^0 - s^j)\| + \|R_3(x^0 + t_j^{\ell_j}\| + \|R_3(x^0 - t_j^{\ell_j})\| \right) \\
&\le \frac{1}{48} \left(L_{\nabla^3 f} \|s^j + t_j^{\ell_j}\|^4 + \frac{1}{24} L_{\nabla^3 f} \| - s^j - t_j^{\ell_j}\|^4 + L_{\nabla^3 f} \| - s^j\|^4 + L_{\nabla^3 f} \|s^j\|^4 + L_{\nabla^3 f} \|t_j^{\ell_j}\| + L_{\nabla^3 f} \| - t_j^{\ell_j}\| \right) \\
&\le \frac{1}{8} L_{\nabla^3 f} (\Delta_S + \Delta_{T_j})^4.
\end{aligned}
\tag{5.25}
$$

Plugging the bound from (5.25) in (5.24), we get

$$\begin{aligned}
&\|\nabla_c^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} H\| \\
&\leq \sum_{j=1}^{m} \frac{1}{\Delta_S \Delta_{T_j}} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}_j^{\,\dagger}\right\| \sqrt{k_j}\frac{1}{8} L_{\nabla^3 f}(\Delta_S + \Delta_{T_j})^4 \\
&\leq \sum_{j=1}^{m} \frac{1}{\Delta_l^2} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}^\dagger\right\| \sqrt{k}\frac{1}{8} L_{\nabla^3 f}(2\Delta_u)^4 \\
&= 2m\sqrt{k} L_{\nabla^3 f} \left(\frac{\Delta_u}{\Delta_l}\right)^2 \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\left(\widehat{T}\right)^\dagger\right\| \Delta_u^2.
\end{aligned}$$

Now, suppose that $T_1 = T_2 = \cdots = T_m = \overline{T}$. The equality follows from (5.13). We have

$$\begin{aligned}
\|\nabla_c^2 f(x^0; S, \overline{T}) - \mathrm{Proj}_{S,\overline{T}} H\| &= \left\|(S^\top)^\dagger \delta_c^2 f(x^0; S, \overline{T}) - (S^\top)^\dagger S^\top H \overline{T}\overline{T}^\dagger\right\| \\
&\leq \frac{1}{\Delta_S} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\delta_c^2 f(x^0; S, \overline{T}) - S^\top H \overline{T}\overline{T}^\dagger\right\| \\
&\leq \frac{1}{\Delta_S \Delta_T} \|(\widehat{S}^\top)^\dagger\| \|\widehat{T}^\dagger\| \|\| D_c^2 - S^\top H \overline{T}\|, \quad (5.26)
\end{aligned}$$

where D_c^2 is defined as in (5.6). Each entry $\left[D_c^2 - S^\top H \overline{T}\right]_{j,\ell}$ is bounded by (5.25). Since

$$\|D_c^2 - S^\top H \overline{T}\| \leq \|D_c^2 - S^\top H \overline{T}\|_F,$$

using (5.25) in (5.26), we get the inequality

$$\begin{aligned}
\|\nabla_c^2 f(x^0; S, \overline{T}) - \mathrm{Proj}_{S,\overline{T}} H\| &\leq \frac{\sqrt{mk}}{24} L_{\nabla^3 f} \frac{(\Delta_S + \Delta_T)^4}{\Delta_S \Delta_T} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}^\dagger\right\| \\
&\leq \frac{2\sqrt{mk}}{3} L_{\nabla^3 f} \frac{\Delta_u}{\Delta_l} \left\|(\widehat{S}^\top)^\dagger\right\| \left\|\widehat{T}^\dagger\right\| \Delta_u^2.
\end{aligned}$$

□

The main differences between the error bounds presented for the GCSH in Theorem 5.14 and the GSH in Theorem 5.13 are

- the error bounds in Theorem 5.14 assume that $f \in \mathcal{C}^4$ and the error bounds in Theorem 5.13 assume $f \in \mathcal{C}^3$,
- the error bounds in Theorem 5.14 involve the Lipschitz constant $L_{\nabla^3 f}$ and the error bounds in Theorem 5.13 involve the Lipschitz constant $L_{\nabla^2 f}$,

– the error bounds in Theorem 5.14 are order-2 accurate and the error bounds in Theorem 5.13 are order-1 accurate.

Since the error bounds in Theorem 5.14 contained the Lipschitz constant $L_{\nabla^3 f}$, it follows that the GCSH is perfectly accurate whenever the function is a polynomial of degree 3 or less.

Item (*iii*) of Theorems 5.13 and 5.14 are more restrictive than Items (*i*) and (*ii*) as they requires all matrices T_j to be equal. In return, it provides symmetry. In the following proposition, we show that the transpose of the GSH (GCSH) of f at x^0 over S and $\overline{T}$ is the same as the GSH (GCSH) of f at x^0 over $\overline{T}$ and S.

Proposition 5.15. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n\times m}$ and $\overline{T} = \begin{bmatrix} t^1 & t^2 & \cdots & t^k \end{bmatrix} \in \mathbb{R}^{n\times k}$ with $x^0 \oplus (\pm S)$, $x^0 \oplus (\pm T_j)$ and $x^0 \oplus (\pm(S \oplus T_j))$ contained in $\operatorname{dom} f$ for all $j \in \{1, \ldots, m\}$. Then*

$$\left(\nabla_s^2 f(x^0; S, \overline{T})\right)^\top = \nabla_s^2 f(x^0; \overline{T}, S),$$

and

$$\left(\nabla_c^2 f(x^0; S, \overline{T})\right)^\top = \nabla_c^2 f(x^0; \overline{T}, S).$$

Proof. We have

$$\begin{aligned}
&\left(\nabla_s^2 f(x^0; S, \overline{T})\right)^\top \\
&= \left((S^\top)^\dagger \delta_s^2 f(x^0; S, \overline{T})\right)^\top \\
&= (\delta_s^2 f(x^0; S, \overline{T}))^\top S^\dagger \\
&= \begin{bmatrix} \nabla_s f(x^0 + s^1; \overline{T}) - \nabla_s f(x^0; \overline{T}) & \cdots & \nabla_s f(x^0 + s^m; \overline{T}) - \nabla_s f(x^0; \overline{T}) \end{bmatrix} S^\dagger \\
&= (\overline{T}^\top)^\dagger \begin{bmatrix} \delta_s f(x^0 + s^1; \overline{T}) - \delta_s f(x^0; \overline{T}) & \cdots & \delta_s f(x^0 + s^m; \overline{T}) - \delta_s f(x^0; \overline{T}) \end{bmatrix} S^\dagger \\
&= (\overline{T}^\top)^\dagger (D_s^2)^\top S^\dagger,
\end{aligned}$$

where D_s^2 is defined as in (5.4). Note that

$$[D_s^2]_{j,\ell} = f(x^0 + s^j + t^\ell) - f(x^0 + s^j) - f(x^0 + t^\ell) + f(x^0).$$

and

$$(D_s^2)^\top = \begin{bmatrix} \left(\delta_s f(x^0 + t^1; S) - \delta_s f(x^0; S)\right)^\top \\ \vdots \\ \left(\delta_s f(x^0 + t^k; S) - \delta_s f(x^0; S)\right)^\top \end{bmatrix}.$$

Hence,

$$\begin{aligned}\left(\nabla_s^2 f(x^0;S,\overline{T})\right)^\top &= (\overline{T}^\top)^\dagger \begin{bmatrix} \left(\delta_s f(x^0+t^1;S)-\delta_s f(x^0;S)\right)^\top \\ \vdots \\ \left(\delta_s f(x^0+t^k;S)-\delta_s f(x^0;S)\right)^\top \end{bmatrix} S^\dagger \\ &= (\overline{T}^\top)^\dagger \begin{bmatrix} \left((S^\dagger)^\top \delta_s f(x^0+t^1;S)-(S^\dagger)^\top \delta_s f(x^0;S)\right)^\top \\ \vdots \\ \left((S^\dagger)^\top \delta_s f(x^0+t^k;S)-(S^\dagger)^\top \delta_s f(x^0;S)\right)^\top \end{bmatrix} \\ &= (\overline{T}^\top)^\dagger \delta_s^2 f(x^0;\overline{T},S) = \nabla_s^2 f(x^0;\overline{T},S).\end{aligned}$$

The second equality follows from the definition of the GCSH (Definition 5.6). □

To conclude this section, let us mention that even when S and $\overline{T}$ are square matrices with full rank, the GSH (GCSH) is not necessarily a symmetric matrix. In the next section, we will see that if a *minimal poised set* is used, then the GSH (GCSH) is symmetric.

In the next section, we investigate how to obtain an order-1 or an order-2 accurate approximation of the full Hessian with a minimal number of distinct sample points.

5.4 Minimal poised sets

In this section, we investigate how to choose the matrices of directions S and T_j to obtain an approximation of the full Hessian with a minimal number of sample points. Reducing the number of distinct points is valuable, as it will decrease the number of distinct function evaluations necessary to compute the GSH or the GCSH. In blackbox optimization, it is often the case that the function hidden in the blackbox is computationally expensive to evaluate. For this reason, decreasing the number of function evaluations used by the GSH (GCSH) is a topic worth investigating.

For all of this section, we set $T_1 = \cdots = T_m = \overline{T} \in \mathbb{R}^{n\times k}$. We show that if S and $\overline{T}$ have a specific structure, then the number of distinct function evaluations necessary to compute the GSH is $(n+1)(n+2)/2$. We then explore some results that occur when such a structure is used. We begin by defining a *set for GSH computation*.

Definition 5.16. The set of all distinct points utilized in the computation of $\nabla_s^2 f(x^0; S, \overline{T})$ is said to be the set for *GSH computation* and is denoted by $\mathcal{S}_s(x^0; S, \overline{T})$.

Note that $\mathcal{S}_s(x^0; S, \overline{T})$ contains at most $(m+1)(k+1)$ distinct points, but can contain fewer points if some of them overlap. Next, we introduce the definition of *minimal poised set* for the GSH.

Definition 5.17 (Minimal poised set for the GSH)**.** Let $x^0 \in \mathbb{R}^n$ be the point of interest. Let $S \in \mathbb{R}^{n\times n}$ and $\overline{T} \in \mathbb{R}^{n\times n}$. We say that $\mathcal{S}_s(x^0; S, \overline{T})$ is a *minimal poised set for the GSH* at x^0 if and only if S and $\overline{T}$ are full rank and $\mathcal{S}_s(x^0; S, \overline{T})$ contains exactly $(n+1)(n+2)/2$ distinct points.

This definition requires S and $\overline{T}$ to have exactly n columns and to be full rank. This is the case where the GSH is S-determined and $\overline{T}$-determined. This implies that S and $\overline{T}$ are of minimal size to ensure that the GSH is an order-1 accurate approximation of the full Hessian.

We next show that it is possible to create a minimal poised set for the GSH.

Proposition 5.18. *Let $x^0 \in \mathbb{R}^n$ be the point of interest. Let the set $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^n \end{bmatrix} \in \mathbb{R}^{n\times n}$. Define the set U_ℓ for each index $\ell \in \{0, 1, \ldots, n\}$ as*

$$U_0 = S$$

and

$$U_\ell = \begin{bmatrix} s^1 - s^\ell & s^2 - s^\ell & \cdots & s^{\ell-1} - s^\ell & -s^\ell & s^{\ell+1} - s^\ell & \cdots & s^n - s^\ell \end{bmatrix}$$

in $\mathbb{R}^{n\times n}, \ell \neq 0$. For each ℓ, $|\mathcal{S}_s(x^0; S, \overline{T})| \leq (n+1)(n+2)/2$ where $\overline{T} = U_\ell$. Moreover, if S is full rank, then $|\mathcal{S}_s(x^0; S, \overline{T})| = (n+1)(n+2)/2$.

Proof. Without loss of generality, let $x^0 = \mathbf{0}$. First, suppose $\ell \in \{1, \ldots, n\}$. For arbitrary function f, consider the matrix $\delta_s^2 f(x^0; S, \overline{T})$ where $\overline{T} = U_\ell$. The computation of $\nabla_s f(x^0; U_\ell)$ evaluates f at the points

$$\begin{aligned} &\{\mathbf{0}, (s^1 - s^\ell), \ldots, (s^{\ell-1} - s^\ell), -s^\ell, (s^{\ell+1} - s^\ell), \ldots, (s^n - s^\ell)\} \\ &= \{\mathbf{0}\} \cup \{-s^\ell\} \cup \bigcup_{\substack{i=1 \\ i\neq\ell}}^{n} \left\{s^i - s^\ell\right\}. \end{aligned}$$

For $i \neq \ell$, the computation of $\nabla_s f(x^0 + s^i; U_\ell)$ evaluates f at the points

$$\begin{aligned} &\{s^i, s^i + (s^1 - s^\ell), \ldots, s^i + (s^{\ell-1} - s^\ell), s^i - s^\ell, s^i + (s^{\ell+1} - s^\ell), \ldots, s^i + (s^n - s^\ell)\} \\ &= \{s^i\} \cup \{s^i - s^\ell\} \cup \bigcup_{\substack{j=1 \\ j\neq\ell}}^{n} \{s^i + s^j - s^\ell\}. \end{aligned}$$

The computation of $\nabla_s f(x^0 + s^\ell; U_\ell)$ evaluates f at the points

$$\begin{aligned}&\{s^\ell, s^\ell + (s^1 - s^\ell), \ldots, s^\ell + (s^{\ell-1} - s^\ell), s^\ell - s^\ell, s^\ell + (s^{\ell+1} - s^\ell), \ldots, s^\ell + (s^n - s^\ell)\}\\ &= \{s^\ell\} \cup \{\mathbf{0}\} \cup \bigcup_{\substack{i=1\\ i\neq\ell}}^{n} \{s^i\} = \{\mathbf{0}\} \cup S.\end{aligned}$$

Thus, f is evaluated at the points

$$\begin{aligned}&\big(\{\mathbf{0}\} \cup \{-s^\ell\} \cup \bigcup_{\substack{i=1\\ i\neq\ell}}^{n} \{s^i - s^\ell\}\big) \cup \big(\bigcup_{\substack{i=1\\ i\neq\ell}}^{n} \{s^i\} \cup \bigcup_{\substack{i=1\\ i\neq\ell}}^{n} \{s^i - s^\ell\} \cup \bigcup_{\substack{i=1\\ i\neq\ell}}^{n} \bigcup_{\substack{j\geq i\\ j\neq\ell}} \{s^i + s^j - s^\ell\}\big) \cup \big(\{\mathbf{0}\} \cup S\big)\\ &= \{\mathbf{0}\} \cup S \cup \{-s^\ell\} \cup \bigcup_{\substack{i=1\\ i\neq\ell}}^{n} \{s^i - s^\ell\} \cup \bigcup_{\substack{i=1\\ i\neq\ell}}^{n} \bigcup_{\substack{j\geq i\\ j\neq\ell}} \{s^i + s^j - s^\ell\}.\end{aligned} \tag{5.27}$$

This is at most $1 + n + 1 + (n-1) + (n-1)(n-2)/2 = (n+1)(n+2)/2$ points.

Now, suppose $\ell = 0$. Using a similar process to the above, we find that f is evaluated at the points

$$\left(\{\mathbf{0}\} \cup \bigcup_{i=1}^{n} \{s^i\}\right) \cup \left(\bigcup_{i=1}^{n} \{2s^i\} \cup \bigcup_{i=1}^{n} \bigcup_{j>i} \{s^i + s^j\}\right). \tag{5.28}$$

This is at most $(n+1)(n+2)/2$ points.

Finally, if S is full rank, then the four sets in (5.27) and the four sets in (5.28) are disjoint, so we have exactly $(n+1)(n+2)/2$ function evaluations. □

Using $S = \mathrm{Id}_n$ in Proposition 5.18, we can create $n+1$ canonical minimal poised sets for the GSH.

Definition 5.19 (ℓ^{th}-canonical minimal poised set for the GSH)**.** Let $x^0 \in \mathbb{R}^n$ be the point of interest. Let $S = \mathrm{Id}_n$. Fix $\ell \in \{0, 1, \ldots, n\}$. Let

$$E_0 = \mathrm{Id}_n,$$

and

$$E_\ell = \begin{bmatrix} e^1 - e^\ell & e^2 - e^\ell & \cdots & e^{\ell-1} - e^\ell & -e^\ell & e^{\ell+1} - e^\ell & \cdots & e^n - e^\ell \end{bmatrix},$$

$\ell \neq 0$. Then $\mathcal{S}_s(x^0; \mathrm{Id}_n, \overline{T})$ where $\overline{T} = E_\ell$ is called the ℓ^{th}*-canonical minimal poised set for the GSH* at x^0.

From Proposition 5.18 and the fact that any matrix E_k is full rank in Definition 5.19, the ℓ^{th}-canonical minimal poised set for the GSH is indeed a minimal poised set for the GSH. Henceforth, we use the notation $\mathcal{M}_s(x^0; S, \overline{T})$ where $\overline{T} = U_\ell$ to denote a minimal poised set for the GSH at x^0 that takes the form constructed in Proposition 5.18.

Note that the order of the directions in S and $\overline{T}$ is arbitrary. Thus, it is immediately clear that if $\mathcal{S}_s(x^0; S, \overline{T})$ is a minimal poised set for the GSH and $P_1, P_2 \in \mathbb{R}^{n\times n}$ are permutation matrices, then $\mathcal{S}_s(x^0; SP_1, \overline{T}P_2)$ is also a minimal poised set for the GSH at x^0. The next proposition expands this idea and demonstrates how to construct minimal poised sets for the GSH.

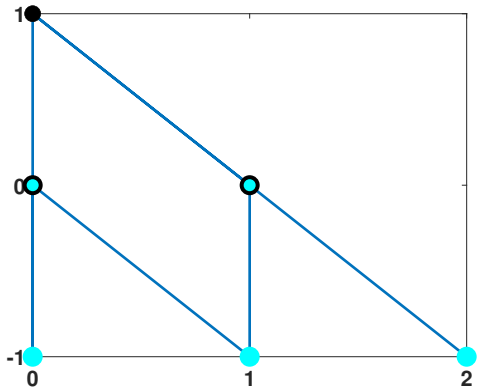
Proposition 5.20. *Let $x^0 \in \mathbb{R}^n$ be the point of interest. Let $S, \overline{T} \in \mathbb{R}^{n\times n}$. Let $N \in \mathbb{R}^{n\times n}$ be an invertible matrix and $P_1, P_2 \in \mathbb{R}^{n\times n}$ be permutation matrices. Then $\mathcal{S}_s(x^0; S, \overline{T})$ is a minimal poised set for the GSH at x^0 if and only if $\mathcal{S}_s(x^0; NSP_1, N\overline{T}P_2)$ is a minimal poised set for the GSH at x^0.*

Proof. The proof follows trivially from properties of matrices. □

It follows that $\mathcal{S}_s(x^0; S, \overline{T})$ is a minimal poised set for the GSH at x^0 if and only if the set $\mathcal{S}_s(x^0; \beta SP_1, \beta\overline{T}P_2)$ is a minimal poised set for the GSH at x^0, where β is a non-zero scalar and P_1, P_2 are permutaion matrices.

Example 5.21. Let $x^0 = (0, 0)$. The 2^{nd}-canonical minimal poised set for the GSH in $\mathbb{R}^2$ contains the points $(0, -1), (0, 0), (0, 1), (1, -1), (1, 0)$ and $(2, -1)$. In this case, $S = \{e^1, e^2\}$ and $\overline{T} = \{(e^1 - e^2), -e^2\}$. Figure 5.1 illustrates this set.

Figure 5.1: The 2^{nd}-canonical minimal poised set for the GSH at x^0 in $\mathbb{R}^2$.



The points in $\{(0,0)\} \cup \{(0,0) \oplus S\} = \{0, e^1, e^2\}$ are represented with solid black borders. These are the base points where simplex gradients will be computed. The lines represent the vectors corresponding to $\overline{T}$ emanate from $\{(0,0)\} \cup \{(0,0) \oplus S\}$. The points in $(\{(0,0)\} \cup \{(0,0) \oplus S\}) \oplus \overline{T}$ are represented with cyan cores. These are the points used to construct the simplex gradients. Notice the points $(0,0)$ and $(1,0)$ have both black borders and cyan cores. These are the common points that allow the number of function evaluations to be reduced to $(n+1)(n+2)/2 = 6$.

We next demonstrate that every minimal poised set for the GSH of the form $\mathcal{M}_s(x^0; S, \overline{T})$ where $\overline{T} = U_\ell$ is poised for quadratic interpolation. We then show that the converse is not true; it is possible to construct a set that is poised for quadratic interpolation, but does not take the form of $\mathcal{M}_s(x^0; S, \overline{T})$.

Proposition 5.22. *Let $S \in \mathbb{R}^{n \times n}$ be full rank. Select $\ell \in \{0, 1, \ldots, n\}$ and define $\overline{T} = U_\ell$ as in Proposition 5.18. Then $\mathcal{M}_s(x^0; S, \overline{T})$ is poised for quadratic interpolation.*

Proof. Noting that $\mathcal{M}_s(x^0; S, \overline{T})$ is poised for quadratic interpolation if and only if the set $\mathcal{M}_s(x^0; S, \overline{T}) \oplus -x^0$ is poised for quadratic interpolation, we assume without loss of generality that $x^0 = \mathbf{0}$.

Case I: $\overline{T} = U_\ell$ for $\ell \in \{1, 2, \ldots, n\}$. Without loss of generality, by Proposition 5.20, assume that $\ell = n$. The points contained in $\mathcal{M}_s(x^0; S, \overline{T})$ are

$$\{\mathbf{0}\} \cup S \cup \{-s^n\} \cup \bigcup_{i=1}^{n-1} \left\{s^i - s^n\right\} \cup \bigcup_{i=1}^{n-1} \bigcup_{\substack{j \geq i \\ j \neq n}} \{s^i + s^j - s^n\}$$

$$= \{\mathbf{0}\} \cup S \cup \{-s^n\} \cup \bigcup_{i=1}^{n-1} \left\{s^i - s^n\right\} \cup \{2s^i - s^n\}_{i=1}^{n-1} \cup \bigcup_{i=1}^{n-1} \bigcup_{\substack{j > i \\ j \neq n}} \left\{s^i + s^j - s^n\right\}.$$

We show that using this set, the only solution to (5.1) is the trivial solution. Considering the point $\mathbf{0}$, we obtain

$$\alpha_0 = 0.$$

Considering the points s^i for $i \in \{1, 2, \ldots, n\}$ and noting that for all $i \in \{1, 2, \ldots, n\}$, $s^i = Se^i$, we obtain

$$\overline{\alpha}^\top e^i + \frac{1}{2}(e^i)^\top \widehat{\mathbf{H}} e^i = 0, \tag{5.29}$$

where $\overline{\alpha}^\top = \alpha^\top S$ and $\widehat{\mathbf{H}} = S^\top \mathbf{H} S$. Note that $\widehat{\mathbf{H}}$ is symmetric. Considering the points $s^i - s^n$ for $i \in \{1, 2, \ldots, n-1\}$, we obtain

$$\overline{\alpha}^\top e^i - \overline{\alpha}^\top e^n - (e^i)^\top \widehat{\mathbf{H}} e^n + \frac{1}{2}(e^i)^\top \widehat{\mathbf{H}} e^i + \frac{1}{2}(e^n)^\top \widehat{\mathbf{H}} e^n = 0.$$

Using (5.29), this simplifies to

$$-\overline{\alpha}^\top e^n - (e^i)^\top \widehat{\mathbf{H}} e^n + \frac{1}{2}(e^n)^\top \widehat{\mathbf{H}} e^n = 0. \tag{5.30}$$

Considering the point $-s^n$, we find

$$-\overline{\alpha}^\top e^n + \frac{1}{2}(e^n)^\top \widehat{\mathbf{H}} e^n = 0, \tag{5.31}$$

which reduces (5.30) to

$$-(e^i)^\top \widehat{\mathbf{H}} e^n = 0 \quad \text{for} \quad i \in \{1, 2, \ldots, n-1\}.$$

Thus, $(e^i)^\top \widehat{\mathbf{H}} e^n = \widehat{\mathbf{H}}_{i,n} = \widehat{\mathbf{H}}_{n,i} = 0$ for all $i \in \{1, 2, \ldots, n-1\}$. Combining (5.29) at $i = n$ and (5.31) multiplied by -1, we get

$$\overline{\alpha}^\top e^n + \frac{1}{2}(e^n)^\top \widehat{\mathbf{H}} e^n = 0 = \overline{\alpha}^\top e^n - \frac{1}{2}(e^n)^\top \widehat{\mathbf{H}} e^n,$$

which implies that $(e^n)^\top \widehat{\mathbf{H}} e^n = \widehat{\mathbf{H}}_{n,n} = 0$. Considering the points $2s^i - s^n$ for $i \in \{1, 2, \ldots, n-1\}$, we get

$$2\overline{\alpha}^\top e^i - \overline{\alpha}^\top e^n + 2(e^i)^\top \widehat{\mathbf{H}} e^i + \frac{1}{2}(e^n)^\top \widehat{\mathbf{H}} e^n = 0.$$

Using (5.31), this simplifies to

$$2\overline{\alpha}^\top e^i + 2(e^i)^\top \widehat{\mathbf{H}} e^i = 0.$$

By multiplying (5.29) by 2 and substituting in the above equation, we get $(e^i)^\top \widehat{\mathbf{H}} e^i = \widehat{\mathbf{H}}_{i,i} = 0$ for all $i \in \{1, 2, \ldots, n-1\}$. This now implies

$$\overline{\alpha}_i = \overline{\alpha}^\top e^i = 0$$

for all $i \in \{1, 2, \ldots, n\}$, i.e., $\overline{\alpha} = \mathbf{0}$. Lastly, consider the points $s^i + s^j - s^n$ for $i \neq j, i, j \in \{1, 2, \ldots, n-1\}$. Since $\widehat{\mathbf{H}}_{i,i} = 0$ for $i \in \{1, \ldots, n\}$, $\widehat{\mathbf{H}}_{i,n} = \widehat{\mathbf{H}}_{n,i} = 0$, for $i \in \{1, 2, \ldots, n-1\}$ and $\overline{\alpha} = \mathbf{0}$, we obtain

$$(e^i)^\top \widehat{\mathbf{H}} e^j = 0.$$

Thus $\widehat{\mathbf{H}} = \mathbf{0}_{n \times n}$. Hence, the only solution to (5.1) is the trivial solution.

Case II: $\overline{T} = U_0$. The proof for this case is analogous to that of Case I. □

It follows from the previous proposition that when a minimal poised set for the GSH is used, the GSH is equal to the simplex Hessian as described in [CSV09b, Section 9.5] (see also [CV07, Section 3]). Therefore, the results obtained in [CV07, CSV09b] related to the simplex Hessian applies to the GSH. In this case, the GSH (and the simplex Hessian) is equal to the Hessian of the quadratic interpolation model passing through the sample points $\mathcal{M}_s(x^0; S, \overline{T})$ where $\overline{T} = U_\ell$. Hence, the results developed in [CSV09b, CSV08b, CV07] related to the Hessian of a quadratic interpolation model are valid for the GSH. One of the advantage of computing the GSH compared to the computation of the Hessian of the quadratic interpolation model has been discussed in Remark 5.7. Another advantage of the GSH compared to the simplex Hessian described in [CSV09b, CV07] is that it provides a simple explicit formula that is well-defined as long as the matrices of directions employed are non-empty. Hence, the formula for the GSH provides an approximation of the full Hessian, or a partial Hessian, for all possible cases defined in Definition 5.10.

Next, we provide an example that serves to show that a set of $(n+1)(n+2)/2$ distinct points in $\mathbb{R}^n$ that is poised for quadratic interpolation is not necessarily a minimal poised set for the GSH.

Example 5.23. Let $x^0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$ be the point of interest. Consider $\mathcal{X} = \{x^0 = \mathbf{0}, e^1, e^2, -e^1, -e^2, -e^1 - e^2\}$. Then $\mathcal{X}$ is poised for quadratic interpolation, but cannot be expressed as a minimal poised set for the GSH at x^0.

Proof. Using a similar approach as in Proposition 5.22, one can verify that $\mathcal{X}$ is poised for quadratic interpolation. Now we show that $\mathcal{X}$ is not a minimal poised set for the GSH at x^0 using brute force. We need to build $S = \{s^1, s^2\}$ such that the matrix corresponding to S is full rank and $x^0 \oplus S \subseteq \mathcal{X}$. Hence, the possible choices for S are

$$S \in \big\{ \{e^1, e^2\}, \{e^1, -e^2\}, \{e^1, -e^1 - e^2\}, \{-e^1, e^2\}, \\ \{-e^1 - e^2, e^2\}, \{-e^1, -e^2\}, \{-e^1, -e^1 - e^2\}, \{-e^1 - e^2, -e^2\} \big\}.$$

Case I: $S = \{e^1, e^2\}$. In this case, we need to build $\overline{T} = \{t^1, t^2\}$ such that the matrix corresponding to $\overline{T}$ is full rank and

$$\{t^1, e^1 + t^1, e^2 + t^1, t^2, e^1 + t^2, e^2 + t^2\} = \mathcal{X}.$$

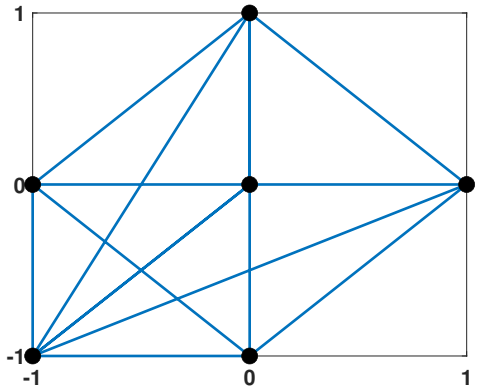
We see that the only possible choice of t^1 such that $\{t^1, e^1 + t^1, e^2 + t^1\} \subseteq \mathcal{X}$ is $t^1 = -e^1 - e^2$ (note $t^1 \neq \mathbf{0}$ as we require full rank). However, the only

possible choice of t^2 such that $\{t^2, e^1 + t^2, e^2 + t^2\} \subseteq \mathcal{X}$ is $t^2 = -e^1 - e^2$. As full rank implies t^1 cannot equal t^2, we see $S = \{e^1, e^2\}$ cannot provide the desired properties.

Cases II through VIII: The other options for S can be eliminated analogously.
Therefore, $\mathcal{X}$ cannot be expressed as a minimal poised set for the GSH at x^0. □

Figure 5.2 shows all possible directions connecting two points in the set $\mathcal{X}$ from Example 5.23. If $\mathcal{X}$ were a minimal poised set for the GSH at $x^0 = \mathbf{0}$, then it would be possible to choose two directions (lines in blue) emerging from x^0 that connect to other points in $\mathcal{Y}$ and these same two directions would be emerging from two other points.

Figure 5.2: A set that is poised for quadratic interpolation but not a minimal poised set for the GSH at x^0.



Next we provide formulae to obtain all the values of the coefficients involved in the quadratic interpolation function of f over a minimal poised set for the GSH of the form $\mathcal{M}_s(x^0; S, \overline{T})$ where $\overline{T} = U_\ell$, which we denote by $Q_f(x^0; S, \overline{T})(x)$.

Proposition 5.24. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$. Let $\mathcal{M}_s(x^0; S, \overline{T}) \subset \operatorname{dom} f$ where $\overline{T} = U_\ell$ be a minimal poised set for the GSH at x^0 as constructed in Proposition 5.18. If $\ell \in \{1, 2, \ldots, n\}$, then the Hessian matrix $\mathbf{H}$ of the quadratic interpolation function $Q_f(x^0; S, \overline{T})(x)$ is given by $\mathbf{H} = S^{-\top} \widehat{\mathbf{H}} S^{-1}$,*

where the entries of the symmetric matrix $\widehat{\mathbf{H}} \in \mathbb{R}^{n\times n}$ *are*

$$\begin{aligned}
\widehat{\mathbf{H}}_{i,\ell} &= -f(x^0+s^i-s^\ell)+f(x^0+s^i)+f(x^0-s^\ell)-f(x^0), \quad i\in\{1,2,\ldots,n\}\setminus\{\ell\},\\
\widehat{\mathbf{H}}_{i,i} &= f(x^0+2s^i-s^\ell)-2f(x^0+s^i-s^\ell)+f(x^0-s^\ell), \quad i\in\{1,2,\ldots,n\}\setminus\{\ell\},\\
\widehat{\mathbf{H}}_{\ell,\ell} &= f(x^0+s^\ell)+f(x^0-s^\ell)-2f(x^0),\\
\widehat{\mathbf{H}}_{i,j} &= f(x^0+s^i+s^j-s^\ell)-f(x^0+s^i-s^\ell)-f(x^0+s^j-s^\ell)+f(x^0-s^\ell),
\end{aligned}$$

for all $i,j\in\{1,2,\ldots,n\}\setminus\{\ell\}, i\neq j$. *If* $\ell=0$, *then*

$$\begin{aligned}
\widehat{\mathbf{H}}_{i,i} &= f(x^0+2s^i)-2f(x^0+s^i)+f(x^0), \quad i\in\{1,2,\ldots,n\},\\
\widehat{\mathbf{H}}_{i,j} &= f(x^0+s^i+s^j)-f(x^0+s^i)-f(x^0+s^j)+f(x^0), i,j\in\{1,2,\ldots,n\}, i\neq j.
\end{aligned}$$

For all $\ell\in\{0,1,\ldots,n\}$, *the vector* α *associated to* $Q_f(x^0;S,\overline{T})(x)$ *is given by* $\alpha = S^{-\top}\overline{\alpha}$, *where the entries of* $\overline{\alpha}\in\mathbb{R}^n$ *are*

$$\overline{\alpha}_i = f(x^0+s^i)-f(x^0)-\frac{1}{2}\widehat{\mathbf{H}}_{i,i}-(x^0)^\top\mathbf{H}s^i, \quad i\in\{1,2,\ldots,n\}.$$

The scalar α_0 *of* $Q_f(x^0;S,\overline{T})(x)$ *is*

$$\alpha_0 = f(x^0)-\alpha^\top x^0-\frac{1}{2}(x^0)^\top\mathbf{H}x^0.$$

Proof. The result is obtained by using Definition 5.2. Let

$$Q_f(x^0;S,\overline{T})(x) = \alpha_0+\alpha^\top x+\frac{1}{2}x^\top\mathbf{H}x,$$

where $a_0\in\mathbb{R}, \alpha\in\mathbb{R}^n$ and $\mathbf{H}=\mathbf{H}^\top\in\mathbb{R}^{n\times n}$. Suppose $\ell\in\{1,2,\ldots,n\}$. Evaluating $Q_f(x^0;S,\overline{T})(x)$ at x^0, we obtain

$$\alpha_0+\alpha^\top x^0+\frac{1}{2}(x^0)^\top\mathbf{H}x^0 = f(x^0). \tag{5.32}$$

Evaluating $Q_f(x^0;S,\overline{T})(x)$ at x^0+s^i and using (5.32), we obtain

$$f(x^0)+\overline{\alpha}^\top e^i+(x^0)^\top\overline{\mathbf{H}}e^i+\frac{1}{2}\widehat{\mathbf{H}}_{i,i} = f(x^0+s^i), \quad i\in\{1,\ldots,n\}, \tag{5.33}$$

where $\overline{\alpha}^\top=\alpha^\top S, \overline{\mathbf{H}}=\mathbf{H}S$ and $\widehat{\mathbf{H}}=S^\top\mathbf{H}S$. Evaluating $Q_f(x^0;S,\overline{T})(x)$ at $x^0+s^i-s^\ell$ and using (5.32) and (5.33), we obtain

$$f(x^0+s^i)-\overline{\alpha}^\top s^\ell-(x^0)^\top\overline{\mathbf{H}}e^\ell-\widehat{\mathbf{H}}_{i,\ell}+\frac{1}{2}\widehat{\mathbf{H}}_{\ell,\ell} = f(x^0+s^i-s^\ell), \tag{5.34}$$

$i \in \{1, \ldots, n\} \setminus \{\ell\}$. Evaluating $Q_f(x^0; S, \overline{T})(x)$ at $x^0 - s^\ell$ and using (5.32), we find

$$-\overline{\alpha}^\top e^\ell - (x^0)^\top \overline{\mathbf{H}} e^\ell + \frac{1}{2}\widehat{\mathbf{H}}_{\ell,\ell} = f(x^0 - s^\ell) - f(x^0). \tag{5.35}$$

Substituting (5.35) into (5.34), we obtain

$$\widehat{\mathbf{H}}_{i,\ell} = \widehat{\mathbf{H}}_{\ell,i} = -f(x^0 + s^i - s^\ell) + f(x^0 + s^i) + f(x^0 - s^\ell) - f(x^0), \tag{5.36}$$

$i \in \{1, \ldots, n\} \setminus \{\ell\}$. Evaluating $Q_f(x^0; S, \overline{T})(x)$ at $x^0 + 2s^i - s^\ell$ and using (5.32), (5.33) and (5.35), we find

$$2f(x^0 + s^i) - 2f(x^0) + f(x^0 - s^\ell) + \widehat{\mathbf{H}}_{i,i} - 2\widehat{\mathbf{H}}_{i,\ell} = f(x^0 + 2s^i - s^\ell). \tag{5.37}$$

Substituting (5.36) in (5.37), we find

$$\widehat{\mathbf{H}}_{i,i} = f(x^0 + 2s^i - s^\ell) + f(x^0 - s^\ell) - 2f(x^0 + s^i - s^\ell),$$

$i \in \{1, 2, \ldots, n\} \setminus \{\ell\}$. Evaluating $Q_f(x^0; S, \overline{T})(x)$ at $x^0 + s^i + s^j - s^\ell$ and using (5.32), (5.33) and (5.35), we find

$$\begin{aligned}\widehat{\mathbf{H}}_{i,j} &= \widehat{\mathbf{H}}_{j,i} \\ &= f(x^0 + s^i + s^j - s^\ell) - f(x^0 + s^i - s^\ell) - f(x^0 + s^j - s^\ell) + f(x^0 - s^\ell),\end{aligned}$$

for $i, j \in \{1, 2, \ldots, n\} \setminus \{\ell\}, i \neq j$. Rearranging (5.35), we get

$$\frac{1}{2}\widehat{\mathbf{H}}_{\ell,\ell} - f(x^0 - s^\ell) + f(x^0) = \overline{\alpha}^\top e^\ell + (x^0)^\top \overline{\mathbf{H}} e^\ell. \tag{5.38}$$

Substituting (5.38) into (5.33) for $i = \ell$, we obtain

$$\widehat{\mathbf{H}}_{\ell,\ell} = f(x^0 + s^\ell) + f(x^0 - s^\ell) - 2f(x^0).$$

The entries of the vector $\overline{\alpha}$ are found by isolating $\overline{\alpha}^\top e^i$ in (5.33). We obtain

$$\overline{\alpha}_i = f(x^0 + s^i) - f(x^0) - \frac{1}{2}\widehat{\mathbf{H}}_{i,i} - (x^0)^\top \mathbf{H} s^i, \quad i \in \{1, 2, \ldots, n\}$$

where $\alpha_i = S^{-\top}\overline{\alpha}_i$. Lastly, the scalar α_0 is obtained from (5.32). We find

$$\alpha_0 = f(x^0) - \alpha^\top x^0 - \frac{1}{2}(x^0)^\top \mathbf{H} x^0.$$

If $\ell = 0$, a similar process can be applied to obtain $\mathbf{H}, \alpha$ and α_0. □

Using $\overline{T} = U_\ell$, it is worth emphasizing that $Q_f(x^0; S, \overline{T})(x)$ can be obtained for free in terms of function evaluations whenever $\nabla_s^2 f(x^0; S, \overline{T})$ has already been computed. Indeed, all the coefficients of $Q_f(x^0; S, \overline{T})$ are computed using the same function evaluations used for finding $\nabla_s^2 f(x^0; S, \overline{T})$. Since $\mathcal{M}_s(x^0; S, \overline{T})$ is poised for quadratic interpolation, there exists one and only one Hessian matrix $\mathbf{H}$. It follows that the Hessian $\mathbf{H}$ of the quadratic interpolation function $Q_f(x^0; S, \overline{T})(x)$ must be equal to $\nabla_s^2 f(x^0; S, \overline{T})$. Note that the Hessian of the quadratic interpolation function $Q_f(x^0; S, \overline{T})$ is symmetric. Hence, the GSH is a symmetric matrix whenever a minimal poised set is used. An alternative proof of this statement is provided in the following proposition.

Proposition 5.25. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^n \end{bmatrix} \in \mathbb{R}^{n \times n}$ and $\overline{T} \in \mathbb{R}^{n \times n}$ with $x^0 \oplus S \oplus \overline{T}, x^0 \oplus S, x^0 \oplus \overline{T}$ contained in $\operatorname{dom} f$ for all $i \in \{1, \ldots, n\}$. If S is full rank and $\overline{T} = U_\ell$ for some $\ell \in \{0, 1, \ldots, n\}$, then*

$$\nabla_s^2 f(x^0; S, \overline{T}) = \left(\nabla_s^2 f(x^0; S, \overline{T})\right)^\top.$$

Proof. Suppose $\ell = 0$. Then $\overline{T} = U_0 = S$ and we obtain

$$\begin{aligned} \nabla_s^2 f(x^0; S, S) &= S^{-\top} \delta_s^2 f(x^0; S, S) \\ &= S^{-\top} D_s^2 S^{-1}, \end{aligned}$$

where D_s^2 is defined as in (5.4). Note that

$$\begin{aligned} [D_s^2]_{i,j} &= f(x^0 + s^i + s^j) - f(x^0 + s^i) - f(x^0 + s^i) - f(x^0 + s^j) + f(x^0) \\ &= [D_s^2]_{j,i} \end{aligned}$$

for all $i, j \in \{1, \ldots, n\}$. Hence $D_s^2 = (D_s^2)^\top$. It follows that

$$\nabla_s^2 f(x^0; S, \overline{T}) = S^{-\top} (D_s^2)^\top S^{-1} = \left(\nabla_s^2 f(x^0; S, \overline{T})\right)^\top.$$

Now, suppose $\overline{T} = U_\ell$ where $\ell \in \{1, \ldots, n\}$. Define $v = -\mathbf{1} - e^\ell \in \mathbb{R}^n$. First, notice that

$$U_\ell = S + s^\ell v^\top.$$

Using Proposition (3.3), we find

$$\begin{aligned} U_\ell^{-1} &= S^{-1} - \frac{S^{-1} s^\ell v^\top S^{-1}}{1 + v^\top S^{-1} s^\ell} \\ &= S^{-1} - \frac{e^\ell v^\top S^{-1}}{1 + v^\top e^\ell} \\ &= S^{-1} + e^\ell v^\top S^{-1}. \end{aligned}$$

So the GSH is

$$\begin{aligned}\nabla_s^2 f(x^0; S, \overline{T}) &= S^{-\top} D_s^2 U_\ell^{-1} \\ &= S^{-\top} D_s^2 \left(S^{-1} + e^\ell v^\top S^{-1} \right) \\ &= S^{-\top} \left(D_s^2 + D_s^2 e^\ell v^\top \right) S^{-1}.\end{aligned}$$

To finish the proof, we show that $D_s^2 + D_s^2 e^\ell v^\top = \left(D_s^2 + D_s^2 e^\ell v^\top \right)^\top$. For $i, j \in \{1, 2, \ldots, n\} \setminus \{\ell\}$, we have

$$\begin{aligned}&\left[D_s^2 + D_s^2 e^\ell v^\top \right]_{i,j} \\ &= f(x^0 + s^i + s^j - s^\ell) - f(x^0 + s^i) - f(x^0 + s^j - s^\ell) + f(x^0) \\ &\quad - f(x^0 + s^i - s^\ell) + f(x^0 + s^i) + f(x^0 - s^\ell) - f(x^0) \\ &= f(x^0 + s^i + s^j - s^\ell) - f(x^0 + s^j - s^\ell) - f(x^0 + s^i - s^\ell) \\ &\quad + f(x^0 - s^\ell) \\ &= \left[D_s^2 + D_s^2 e^\ell v^\top \right]_{j,i}.\end{aligned}$$

For $i \in \{1, 2 \ldots, n\}$, we have

$$\left[D_s^2 + D_s^2 e^\ell v^\top \right]_{i,\ell} = -f(x^0 + s^i - s^\ell) + f(x^0 + s^i) + f(x^0 - s^\ell) - f(x^0)$$

and

$$\begin{aligned}&\left[D_s^2 + D_s^2 e^\ell v^\top \right]_{\ell,i} \\ &= f(x^0 + s^\ell + s^i - s^\ell) - f(x^0 + s^\ell) - f(x^0 + s^i - s^\ell) + f(x^0) \\ &\quad - \left(f(x^0 + s^\ell - s^\ell) - f(x^0 + s^\ell) - f(x^0 - s^\ell) + f(x^0) \right) \\ &= f(x^0 + s^i) - f(x^0 + s^i - s^\ell) - f(x^0) + f(x^0 - s^\ell).\end{aligned}$$

Therefore, $D_s^2 + D_s^2 e^\ell v^\top = \left(D_s^2 + D_s^2 e^\ell v^\top \right)^\top$ and the result follows. □

In the case where S is not a square matrix, we can obtain a symmetric GSH by setting $\overline{T} = S$.

A minimal poised set for the GCSH may also be defined. We briefly discussed this topic to conclude this section.

Definition 5.26 (Minimal poised set for the GCSH)**.** Let $x^0 \in \mathbb{R}^n$ be the point of interest. Let $S \in \mathbb{R}^{n \times n}$ and $\overline{T} \in \mathbb{R}^{n \times n}$. We say that $\mathcal{S}_c(x^0; S, \overline{T})$ is a *minimal poised set for the GCSH* at x^0 if and only if S and $\overline{T}$ are full rank and $\mathcal{S}_c(x^0; S, \overline{T})$ contains exactly $n^2 + n + 1$ distinct points.

The next proposition provides a possible way to choose the matrices S and $\overline{T}$ so that the set of sample points is a minimal poised set for the GCSH.

Proposition 5.27. *Let $S \in \mathbb{R}^{n\times n}$ be full rank and let $\overline{T} = -S$. Then $\mathcal{S}_c(x^0; S, \overline{T})$ is a minimal poised set for the GCSH at x^0.*

Proof. The matrices S and $\overline{T}$ are clearly square matrices with full rank. The set $\mathcal{S}_c(x^0; S, \overline{T})$ contains the following sample points:

$$x^0 \oplus S \oplus -S, x^0 \oplus -S \oplus S, x^0 \oplus \pm S, x^0.$$

Since the set $x^0 \oplus S \oplus -S$ is equal to to the set $x^0 \oplus -S \oplus S$, we drop $x^0 \oplus S \oplus -S$. The set $x^0 \oplus \pm S$ contains $2n$ distinct sample points.
The set $x^0 \oplus -S \oplus S$ contains $n^2 - n + 1$ distinct sample points and it contains the point x^0. Hence, the number of distinct sample points is

$$2n + (n^2 - n + 1) = n^2 + n + 1.$$

Therefore, $\mathcal{S}_c(x^0; S, \overline{T})$ is a minimal poised set for the GCSH. □

Note that $n^2 + n + 1 > (n + 1)(n + 2)/2$ whenever $n \geq 2$. Hence, more function evaluations are required when a minimal poised set for the GCSH is used compared to a minimal set for the GSH, but in return it provides an order-2 accurate approximation of the full Hessian.

We conclude this section by presenting a result regarding the symmetry of the GCSH.

Proposition 5.28. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ and let $x^0 \in \operatorname{dom} f$ be the point of interest. Let $S \in \mathbb{R}^{n\times m}$ and $\overline{T} \in \mathbb{R}^{n\times k}$ with $x^0 \oplus S \oplus \overline{T}, x^0 \oplus (-S) \oplus (-\overline{T}), x^0 \oplus (\pm S), x^0 \oplus (\pm\overline{T})$ contained in $\operatorname{dom} f$. If $\overline{T} = \pm S$, then*

$$\nabla^2_c f(x^0; S, \overline{T}) = \left(\nabla^2_c f(x^0; S, \overline{T})\right)^\top.$$

Proof. The result when $\overline{T} = S$ follows immediately from the definition (Definition 5.6). In the case where $\overline{T} = -S$, using Lemma 5.8, we have

$$\begin{aligned}(\nabla^2_c f(x^0; S, \overline{T}))^\top &= \delta^2_c(x^0; S, \overline{T})^\top S^\dagger \\ &= (-S^\top)^\dagger (D^2_c)^\top S^\dagger\end{aligned}$$

where

$$\begin{aligned}&[D^2_c]_{i,j} \\ &= \tfrac{1}{2}\left(f(x^0 + s^i - s^j) - f(x^0 + s^i) + f(x^0 - s^i + s^j) - f(x^0 - s^i) - f(x^0 - s^j) - f(x^0 + s^j) + 2f(x^0)\right) \\ &= [D^2_c]_{j,i}\end{aligned}$$

for all $i, j \in \{1, \ldots, m\}$. Therefore,

$$(\nabla_c^2 f(x^0; S, \overline{T}))^\top = (S^\top)^\dagger D_c^2 (-S)^\dagger = \nabla_c^2 f(x^0; S, \overline{T}). \qquad \square$$

In the next section, we investigate how to choose the matrices of directions S and T_j when we are only interested by a proper subset of the entries of the Hessian.

5.5 Approximating partial Hessians

In this section, we provide details on how to choose the matrices S and T_j when we are interested in a proper subset of the entries of the Hessian. In particular, we investigate how to approximate the diagonal entries of the Hessian, the off-diagonal entries of the Hessian and a column of the Hessian. The number of function evaluations required is discussed in each case. The relation between the *centered simplex Hessian diagonal* (CSHD) introduced in [JB22] and the GCSH is clarified. The CSHD is an approximation technique that provides an order-2 accurate approximation of the diagonal entries of the Hessian. In each case, an error bound is provided and the number of function evaluations required is discussed. We begin by presenting results on how to approximate some, or all, diagonal entries of the Hessian.

Approximating the diagonal entries of the Hessian

An explicit formula to compute all the diagonal entries of the Hessian which is well-defined regardless of the number of sample points utilized is discussed in [JB22]. We begin by showing that this technique called centered simplex Hessian diagonal (CSHD) is a specific case of a GCSH when the appropriate matrices of directions S and T_j are employed. First, recall the definitions of the *Hadamard product* and the CSHD.

Definition 5.29. [HJ90] Let $M \in \mathbb{R}^{n \times m}$ and $N \in \mathbb{R}^{n \times m}$. The Hadamard product of M and N, denoted $M \odot N$ is the component-wise product. That is $[M \odot N]_{i,j} = M_{i,j} N_{i,j}$ for all $i \in \{1, \ldots n\}$ and $j \in \{1, \ldots, m\}$.

Definition 5.30 (Centered simplex Hessian diagonal)**.** [JB22] Let the function $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$, $x^0 \in \operatorname{dom} f$ be the point of interest, $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and $W = \begin{bmatrix} s^1 \odot s^1 & \cdots & s^m \odot s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$. Assume that $x^0 \oplus (\pm S) \subset \operatorname{dom} f$. The centered simplex Hessian diagonal of f at x^0 over S, denoted by $d\nabla^2 f(x^0; S)$ is a vector in $\mathbb{R}^n$ given by

$$d\nabla^2 f(x^0; S) = (W^\top)^\dagger \delta_d f(x^0; S),$$

where

$$\delta_d f(x^0; S) = \begin{bmatrix} f(x^0 + s^1) + f(x^0 - s^1) - 2f(x^0) \\ \vdots \\ f(x^0 + s^m) + f(x^0 - s^m) - 2f(x^0) \end{bmatrix} \in \mathbb{R}^m .$$

Definition 5.31 (partial diagonal matrix)**.** Let $M \in \mathbb{R}^{n\times m}, m \leq n$. We say that M is a partial diagonal matrix if there exists a diagonal matrix $D \in \mathbb{R}^{n\times n}$ such that for each column $Me^j, j \in \{1, \dots m\}$, there exists an index $i \in \{1, \dots, n\}$ such that $Me^j = De^i$.

In other words, a partial diagonal matrix is a subset of the columns of a single diagonal matrix. For example the matrix

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 2 \end{bmatrix}$$

is a partial diagonal matrix, but

$$\widetilde{M} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

is not a partial diagonal matrix. Note that a partial diagonal matrix is full column rank if and only if it does not contain a column equal to the zero vector in $\mathbb{R}^n$.

The following lemma provides details about the Moore–Penrose pseudo-inverse of a partial diagonal matrix with full column rank.

Lemma 5.32. *Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n\times m}$ where $m \leq n$ be a partial diagonal matrix with full column rank. Then*

$$S^\dagger = \left[(s^1)^\dagger (s^2)^\dagger \cdots (s^m)^\dagger \right]^\top .$$

Proof. Let u_j be the index in $\{1, \dots, m\}$ of the only non-zero entry in column s^j. Since S is full column rank, using (3.1), we have

$$\begin{aligned} S^\dagger &= (S^\top S)^{-1} S^\top \\ &= \left(\operatorname{Diag} \begin{bmatrix} (s^1_{u_1})^2 & \cdots & (s^m_{u_m})^2 \end{bmatrix} \right)^{-1} S^\top \\ &= \operatorname{Diag} \begin{bmatrix} \frac{1}{(s^1_{u_1})^2} & \cdots & \frac{1}{(s^m_{u_m})^2} \end{bmatrix} S^\top \\ &= \begin{bmatrix} \frac{1}{s^1_{u_1}} e^{u_1} & \cdots \frac{1}{s^m_{u_m}} e^{u_m} \end{bmatrix}^\top . \end{aligned}$$

Note that each column s^j in S is full column rank. Using (3.1), we find $(s^j)^\dagger = \frac{1}{s^j_{u_j}}(e^{u_j})^\top$ for all $j \in \{1,\dots,m\}$. Therefore,

$$S^\dagger = \begin{bmatrix}(s^1)^\dagger & (s^2)^\dagger & \cdots & (s^m)^\dagger\end{bmatrix}^\top. \qquad \square$$

The following theorem provides a sufficient condition for the GCSH to return the same approximation of the diagonal entries of the Hessian than the CSHD.

Theorem 5.33. *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}, x^0 \in \operatorname{dom} f$ be the point of interest, $S = \begin{bmatrix}s^1 & s^2 & \cdots & s^m\end{bmatrix} \in \mathbb{R}^{n\times m}$ and $T_j = -s^j \in \mathbb{R}^n$ for all $j \in \{1,\dots,m\}$. Let $z \in \mathbb{R}^n$ be a vector containing the n diagonal entries of $\nabla^2_c f(x^0; S, T_{1:m})$. That is $z_i = \left[\nabla^2_c f(x^0; S, T_{1:m})\right]_{i,i}$ for all $i \in \{1,\dots,n\}$. If S is a partial diagonal matrix with full column rank, then $z = d\nabla^2 f(x^0; S)$.*

Proof. Let $A = \begin{bmatrix}S & -S\end{bmatrix} \in \mathbb{R}^{n\times 2m}$ and $T_{m+j} = s^j$ for $j \in \{1,\dots,m\}$. We have

$$\begin{aligned}\nabla^2_c f(x^0; S, T_{1:m}) &= \nabla^2_s f(x^0; A, T_{1:2m}) \qquad \text{(by Proposition 5.9)}\\ &= (A^\top)^\dagger \begin{bmatrix} \left(\nabla_s f(x^0 + s^1; -s^1) - \nabla_s f(x^0; -s^1)\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0 + s^m; -s^m) - \nabla_s f(x^0; -s^m)\right)^\top \\ \left(\nabla_s f(x^0 - s^1; s^1) - \nabla_s f(x^0; s^1)\right)^\top \\ \vdots \\ \left(\nabla_s f(x^0 - s^m; s^m) - \nabla_s f(x^0; s^m)\right)^\top \end{bmatrix}.\end{aligned}$$

Since $(A^\top)^\dagger = \frac{1}{2}\begin{bmatrix}(S^\top)^\dagger & -(S^\top)^\dagger\end{bmatrix}$, and expanding each row of the form $\left(\nabla_s f(x^0 \pm s^j; \mp s^j) - \nabla_s f(x^0; \mp s^j)\right)^\top$, we obtain

$$\begin{aligned}&\nabla^2_c f(x^0; S, T_{1:m})\\ &= \frac{1}{2}\begin{bmatrix}(S^\top)^\dagger & -(S^\top)^\dagger\end{bmatrix} \begin{bmatrix} (-s^1)^\dagger\left(-f(x^0 + s^1) - f(x^0 - s^1) + 2f(x^0)\right) \\ \vdots \\ (-s^m)^\dagger\left(-f(x^0 + s^m) - f(x^0 - s^m) + 2f(x^0)\right) \\ (s^1)^\dagger\left(2f(x^0) - f(x^0 - s^1) - f(x^0 + s^1)\right) \\ \vdots \\ (s^m)^\dagger\left(2f(x^0) - f(x^0 - s^m) - f(x^0 + s^m)\right) \end{bmatrix}\end{aligned}$$

$$= (S^\dagger)^\top \begin{bmatrix} (s^1)^\dagger \left(f(x^0 - s^1) + f(x^0 + s^1) - 2f(x^0)\right) \\ \vdots \\ (s^m)^\dagger \left(f(x^0 - s^m) + f(x^0 + s^m) - 2f(x^0)\right) \end{bmatrix}$$

$$= \begin{bmatrix} ((s^1)^\dagger)^\top & \cdots & ((s^m)^\dagger)^\top \end{bmatrix} \begin{bmatrix} (s^1)^\dagger \left(f(x^0 - s^1) + f(x^0 + s^1) - 2f(x^0)\right) \\ \vdots \\ (s^m)^\dagger \left(f(x^0 - s^m) + f(x^0 + s^m) - 2f(x^0)\right) \end{bmatrix}$$

by Lemma 5.32. Let $z \in \mathbb{R}^n$ be the vector containing the n diagonal entries of the previous equation. Then

$$\begin{aligned} z &= \begin{bmatrix} ((s^1)^\top)^\dagger \odot ((s^1)^\top)^\dagger & \cdots & ((s^m)^\top)^\dagger \odot ((s^m)^\top)^\dagger \end{bmatrix} \delta_d f(x^0; S) \\ &= (W^\top)^\dagger \delta_d f(x^0; S) = d\nabla^2 f(x^0; S). \end{aligned}$$ □

By defining the sets T_j as in Theorem 5.33, the CSHD and the GCSH use the same set of sample points. However, if S is not a partial diagonal matrix with full column rank, then the vector z containing the diagonal entries of the GCSH is not necessarily equal to the CSHD. Moreover, the GCSH is not necessarily a diagonal matrix. The following two examples illustrate these claims.

Example 5.34. Let

$$S = \begin{bmatrix} s^1 & s^2 & s^3 \end{bmatrix} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0.2 \\ 0 & 0 & 0 \end{bmatrix}.$$

Let $T_j = -s^j$ for all $j \in \{1, 2, 3\}$. Let $f(x) = -2x_1^4 + x_2^4 + 10x_3^4$ and $x^0 = \begin{bmatrix} 2 & -2 & 5 \end{bmatrix}^\top$. Note that

$$\nabla^2 f(x^0) = \text{Diag}[-96\ 48\ 3000].$$

The GCSH is

$$\nabla_c^2 f(x^0; S, T_{1:3}) = \text{Diag}[-96.04\ 48.068\ 0],$$

and the CSHD is

$$d\nabla^2 f(x^0; S) = \begin{bmatrix} -96.04 & 48.0765 & 0 \end{bmatrix}^\top.$$

The next example shows that the GCSH is not necessarily a diagonal matrix even when we use the same set of sample points.

Example 5.35. Let

$$S = \begin{bmatrix} s^1 & s^2 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.1 \\ 0 & 0.1 \\ 0 & 0 \end{bmatrix}.$$

Let $T_j = -s^j$ for all $j \in \{1, 2\}$ Consider the same function and point of interest as in the previous example. That is $f(x) = -2x_1^4 + x_2^4 + 10x_3^4$ and $x^0 = \begin{bmatrix} 2 & -2 & 5 \end{bmatrix}^\top$. Then the GCSH is

$$\nabla_c^2 f(x^0; S, T_{1:2}) = \begin{bmatrix} -96.04 & 0 & 0 \\ 72.03 & -24.01 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and the CSHD is

$$d\nabla^2 f(x^0; S) = \begin{bmatrix} -96.04 & 48.02 & 0 \end{bmatrix}^\top.$$

The next theorem presents a general error bound when the matrices of directions T_j used in the computation of the GCSH have the form $T_j = -s^j$ for all $j \in \{1, \ldots, m\}$. We begin by introducing a result concerning the projection of a matrix over S and $T_{1:m}$.

Proposition 5.36. *Let $M \in \mathbb{R}^{n \times n}$. Let $S = \begin{bmatrix} s^1 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and let $T_j = -s^j$ for all $j \in \{1, \ldots, m\}$. If S is a partial diagonal matrix with full column rank, then*

$$\operatorname{Proj}_{S,T_{1:m}} M = \operatorname{Proj}_{S,T_{1:m}} \operatorname{Diag}\begin{bmatrix} M_{1,1} & \cdots & M_{n,n} \end{bmatrix}.$$

Moreover, if $(e^i)^\top S \neq \mathbf{0}_m^\top$ for some $i \in \{1, \ldots, n\}$, then

$$\left[\operatorname{Proj}_{S,T_{1:m}} M\right]_{i,i} = M_{i,i}.$$

If $(e^i)^\top S = \mathbf{0}_m^\top$ for some $i \in \{1, \ldots, n\}$, then

$$\left[\operatorname{Proj}_{S,T_{1:m}} M\right]_{i,i} = 0.$$

Proof. We have

$$\begin{aligned} \sum_{j=1}^{m} (S^\top)^\dagger e^j (e^j)^\top S^\top M T_j T_j^\dagger &= \sum_{j=1}^{m} ((s^j)^\top)^\dagger (s^j)^\top M (-s^j)(-s^j)^\dagger \\ &= \sum_{j=1}^{m} e^{u_j} (e^{u_j})^\top M e^{u_j} (e^{u_j})^\top \end{aligned}$$

where u_j represents the index of the only non-zero entry in s^j, $u_j \in \{1, \dots n\}$, and $j \in \{1, \dots, m\}$. From the definition of a partial diagonal matrix, we know that $u_j \neq u_{\bar{j}}$ whenever $j \neq \bar{j}$, j and $\bar{j}$ in $\{1, \dots, m\}$. Noticing that $e^{u_j}(e^{u_j})^\top = \text{Diag}(e^{u_j})$, we get

$$\begin{aligned}
&\sum_{j=1}^{m} (S^\top)^\dagger e^j (e^j)^\top S^\top M T_j T_j^\dagger \\
&= \sum_{j=1}^{m} \text{Diag}(e^{u_j}) M \, \text{Diag}(e^{u_j}) \\
&= \sum_{j=1}^{m} \text{Diag}(e^{u_j}) \cdot M_{u_j,u_j} = \text{Proj}_{S,T_{1:m}} \text{Diag}[M_{1,1} \ \cdots \ M_{n,n}].
\end{aligned}$$

The rest of the proof follows immediately from the fact that $m \leq n$, and $u_j \neq u_{\bar{j}}$ whenever $j \neq \bar{j}$, j and $\bar{j}$ in $\{1, \dots, m\}$. □

The notation D is now used to represent the diagonal matrix in $\mathbb{R}^{n \times n}$ containing the diagonal entries of the Hessian $\nabla^2 f(x^0)$. That is $D_{i,i} = [\nabla^2 f(x^0)]_{i,i}$ for all $i \in \{1, \dots, n\}$. If S is a diagonal matrix with full column rank and $T_j = -s^j$ for all $j \in \{1, \dots, n\}$, it follows from the previous proposition that

$$\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0) = \text{Proj}_{S,T_{1:m}} D.$$

In other words, the projection of the full true Hessian is a diagonal matrix that keeps intact all diagonal entries of the true Hessian. In the case where S is a non-square partial diagonal matrix, then it makes the (i, i) diagonal entry of the true Hessian equal to zero if S does not contain a multiple of the identity column e^i. Also, since S is full column rank, it follows from Propositions 5.36 and 5.12(*ii*) that $\nabla_s^2 f(x^0; S, T_{1:m})$ and $\nabla_c^2 f(x^0; S, T_{1:m})$ are diagonal matrices.

The next theorem presents an error bound when the GCSH is used to approximate some, or all, diagonal entries of the true Hessian.

Theorem 5.37 (Error bound for the diagonal entries of the Hessian)**.** *Let $f : \text{dom}\, f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^4$ on an open domain containing $B_n(x^0; \Delta_S)$ where $x^0 \in \text{dom}\, f$ is the point of interest and $\Delta_S > 0$ is the radius of $S = \begin{bmatrix} s^1 & \cdots s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$. Let $T_j = -s^j$ for all $j \in \{1, \dots, m\}$. Denote by $L_{\nabla^3 f} \geq 0$ the Lipschitz constant of $\nabla^3 f$ on $\overline{B}_n(x^0; \Delta_S)$. If S is a partial*

diagonal matrix with full column rank, then

$$\begin{aligned}&\left\|\mathrm{Proj}_{S,T_{1:m}} \nabla_c^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)\right\| \\ &= \left\|\nabla_c^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} D\right\| \leq \frac{1}{12} L_{\nabla^3 f} \Delta_S^2. \end{aligned} \tag{5.39}$$

Proof. By Proposition 5.12 (ii) and Proposition 5.36, we get the equality. To make notation more compact, let $\delta = \delta_d f(x^0; S) \in \mathbb{R}^m$. We have

$$\begin{aligned}&\left\|\nabla_c^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} D\right\| \\ &= \left\|\sum_{i=1}^m ((s^i)^\top)^\dagger (s^i)^\top (S^\top)^\dagger \,\mathrm{Diag}(\delta) S^\dagger (-s^i)(-s^i)^\dagger - \sum_{i=1}^m ((s^i)^\top)^\dagger (s^i)^\top D s^i (s^i)^\dagger\right\| \\ &\leq \max_{i=1,\ldots,m} \left(\|((s^i)^\top)^\dagger\| \|(s^i)^\dagger\| \left| (s^i)^\top (S^\top)^\dagger \,\mathrm{Diag}(\delta) S^\dagger s^i - (s^i)^\top D s^i \right| \right) \\ &= \max_{j=1,\ldots,m} \left(\frac{1}{\|s^j\|^2} \left| \delta_j - (s^j)^\top D s^j \right| \right). \end{aligned}$$

By Taylor's Theorem, using a similar process than the proof in [JB22, Theorem 3.3], we obtain

$$\begin{aligned}\left\|\nabla_c^2 f(x^0; S, T_{1:m}) - \mathrm{Proj}_{S,T_{1:m}} D\right\| &\leq \max_{j=1,\ldots,m} \left(\frac{1}{\|s^j\|^2} \frac{1}{12} L_{\nabla^3 f} \|s^j\|^4 \right) \\ &= \max_{j=1,\ldots,m} \left(\frac{1}{12} L_{\nabla^3 f} \|s^j\|^2 \right) \\ &\leq \frac{1}{12} L_{\nabla^3 f} \Delta_S^2. \end{aligned}$$

□

By defining S and T_j as in the previous theorem, $\nabla_c^2 f(x^0; S, T_{1:m})$ is S-underdetermined/determined and $T_{1:m}$-underdetermined. Hence, the general error bound proposed for the GCSG in Theorem 5.14(ii) is also valid. The previous proof utilized properties of partial diagonal matrices to obtain a tighter error bound than the one proposed in Theorem 5.14(ii).

The previous theorem shows how to obtain an order-2 accurate approximation of some, or all, diagonal entries of the Hessian. This requires $2n + 1$ function evaluations when S is square. If we are interested in approximating only one diagonal entry of a Hessian $\nabla^2 f(x^0)$, say $\left[\nabla^2 f(x^0)\right]_{i,i}$, then the computational cost is three function evaluations. In this case, we can choose $S = he^i$ and $T_1 = -he^i$. Each additional diagonal entry can be obtained for two more function evaluations.

Other matrices of directions S and T_j may be used to obtain an approximation of all diagonal entries of a Hessian. For instance, the following

matrices can be used:

$$S = \begin{bmatrix} s^1 & \cdots & s^n \end{bmatrix} = h\,\mathrm{Id}, \quad T_j = s^j, \quad \text{for all } j \in \{1, \ldots, n\}, h \neq 0.$$

In this case, S is a diagonal matrix with full column rank and it follows from Proposition that $\mathrm{Proj}_{S,T_{1:m}} \nabla^2 f(x^0) = \mathrm{Proj}_{S,T_{1:m}} D = D$ where $D \in \mathbb{R}^{n\times n}$ is the diagonal matrix such that $D_{i,i} = [\nabla^2 f(x^0)]_{i,i}$ for all $i \in \{1, \ldots, n\}$. By Theorem 5.13(ii), this choice of matrices provides an order-1 accurate approximation of all diagonal entries of the Hessian. The computation of $\nabla^2_s f(x^0; S, T_{1:n})$ requires $2n+1$ function evaluations. Hence, it is preferable to choose the matrices of directions S and T_j as in Theorem 5.37 since it provides a greater order of accuracy for the same number of function evaluations.

In the next section, approximating some, or all, off-diagonal entries of the Hessian is investigated.

Approximating the off-diagonal entries of the Hessian

In this section, how to approximate some, or all, off-diagonal entries of the Hessian is examined.

First, recall that the Hessian $\nabla^2 f(x^0)$ is symmetric whenever $f \in \mathcal{C}^2$. Therefore, it is sufficient to consider the off-diagonal entries $[\nabla^2 f(x^0)]_{i,j}$ such that $i < j$. It is possible to approximate some, or all, off-diagonal entries of the Hessian by setting the matrices of directions S and T_j in the following way. Define

$$\begin{aligned}
&\widetilde{S} \in \mathbb{R}^{n\times n-1} : \text{a partial diagonal matrix with full column rank} \\
&\qquad\qquad \text{such that the } n^{\text{th}} \text{ row is equal to } \mathbf{0}_{n-1}^\top, \\
&S = \begin{bmatrix} s^1 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n\times m} : \text{a non-empty subset of the columns of } \widetilde{S}, \\
&T = \begin{bmatrix} t^1 & \cdots & t^n \end{bmatrix} \in \mathbb{R}^{n\times n} : \text{a diagonal matrix with full column rank,} \\
&\widetilde{T}_j = \begin{bmatrix} t^{u_j+1} & \cdots & t^{n-1} & t^n \end{bmatrix} \in \mathbb{R}^{n\times n-u_j} \text{ where } u_j \text{ represents the index} \\
&\text{of the non-zero entry in } s^j, j \in \{1, \ldots, m\}, \\
&T_j \in \mathbb{R}^{n\times k_j} : \text{a subset of directions contained in } \widetilde{T}_j \text{ for all } j \in \{1, \ldots, m\}.
\end{aligned} \tag{5.40}$$

(5.41)

In the next theorem, the matrix $U \in \mathbb{R}^{n\times n}$ denotes a strictly upper triangular matrix such that

$$U_{i,j} = \begin{cases} \left[\nabla^2 f(x^0)\right]_{i,j}, & \text{if} \quad 1 \leq i < j \leq n, \\ 0, & \text{otherwise.} \end{cases}$$

Using a similar process as the one in Proposition 5.36, it can be shown that

$$\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0) = \text{Proj}_{S,T_{1:m}} U$$

and that the GSH (GCSH) is a strictly upper triangular matrix whenever the matrices of directions S and T_j are defined as in (5.40) and (5.41).

The following two error bounds follow from Theorem 5.13(*ii*) and Theorem 5.14(*ii*), respectively.

Corollary 5.38 (Error bound for the off-diagonal entries of the Hessian)**.** *Let $f : \text{dom}\, f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^4$ on $B_n(x^0; \overline{\Delta})$ where $x^0 \in \text{dom}\, f$ is the point of interest and $\overline{\Delta} > 0$. Denote by $L_{\nabla^2 f} \geq 0$ and $L_{\nabla^3 f} \geq 0$ the Lipschitz constant of $\nabla^2 f$ and $\nabla^3 f$ on $\overline{B}_n(x^0; \overline{\Delta})$, respectively. Let $S = \begin{bmatrix} s^1 & s^2 & \cdots & s^m \end{bmatrix} \in \mathbb{R}^{n \times m}$ and $T_j \in \mathbb{R}^{n \times k_j}$ be defined as in* (5.40) *and* (5.41)*, respectively. Assume that $B_n(x^0 + s^j; \Delta_{T_j}) \subset B_n(x^0; \overline{\Delta})$ for all $j \in \{1, \ldots, m\}$.*

(i) Then

$$\begin{aligned} &\left\|\text{Proj}_{S,T_{1:m}} \nabla_s^2 f(x^0; S, T_{1:m}) - \text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)\right\| \\ &= \|\nabla_s^2 f(x^0; S, T_{1:m}) - \text{Proj}_{S,T_{1:m}} U\| \\ &\leq 4m\sqrt{k} L_{\nabla^2 f} \left(\frac{\Delta_u}{\Delta_l}\right)^2 \|(\widehat{S}^\top)^\dagger\| \|\widehat{T}^\dagger\| \Delta_u, \end{aligned}$$

and

(ii)

$$\begin{aligned} &\left\|\text{Proj}_{S,T_{1:m}} \nabla_c^2 f(x^0; S, T_{1:m}) - \text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)\right\| \\ &= \left\|\nabla_c^2 f(x^0; S, T_{1:m}) - \text{Proj}_{S,T_{1:m}} U\right\| \\ &\leq 2m\sqrt{k} L_{\nabla^3 f} \left(\frac{\Delta_u}{\Delta_l}\right)^2 \|(\widehat{S}^\top)^\dagger\| \|\widehat{T}^\dagger\| \Delta_u^2. \end{aligned}$$

A possible simple choice for S and T_j if **all** off-diagonal entries are of interest is to set

$$S = h \begin{bmatrix} e^1 & \cdots & e^{n-1} \end{bmatrix}, \tag{5.42}$$

$$T_j = h \begin{bmatrix} e^{j+1} & \cdots & e^n \end{bmatrix}, \quad \text{for all} \quad j \in \{1, \ldots, n-1\} \tag{5.43}$$

where $h \neq 0$. In this case, the GSH is an order-1 accurate approximation of all off-diagonal entries of the Hessian. To compute this GSH, the function

must be evaluated at the points $x^0, x^0 \oplus S, x^0 \oplus T_j$ and $x^0 + s^j \oplus T_j$ for all $j \in \{1, \dots n-1\}$. Hence, the number of distinct function evaluations is

$$1 + (n-1) + (n-1) + \frac{(n-1)n}{2} - (n-2) = n + \frac{(n-1)n}{2} = \frac{n(n+1)+2}{2}.$$

In the previous equation, we subtracted $(n-2)$ since $x^0 \oplus h \begin{bmatrix} e^2 & \cdots e^{n-1} \end{bmatrix}$ appears in $x^0 \oplus T_j$ and $x^0 \oplus S$.

Note that this number of function evaluations is smaller than $(n+1)(n+2)/2$ whenever $n \geq 1$, which is the number of function evaluations require to compute a GSH with a minimal poised set (Definition 5.17). Therefore, if we are only interested in the off-diagonal entries of a Hessian, it is preferable to set the matrices S and T_j as described in this section rather than using a minimal poised set for GSH.

In the previous corollary, Item (ii) shows that the GCSH is an order-2 accurate approximation of all off-diagonal entries of the Hessian. In this case, the sample points used are $x^0, x^0 \oplus (\pm S), x^0 \oplus (\pm T_j), x^0 \oplus S \oplus T_j$, and $x^0 \oplus (-S) \oplus (-T_j)$ for all $j \in \{1, \dots n-1\}$. The number of distinct function evaluations is

$$1 + 2\left(\frac{n(n+1)}{2}\right) = n^2 + n + 1.$$

Notice that that this is the same amount of function evaluations utilized when using a minimal poised set for GCSH (Definition 5.26). Therefore, there is no advantage in terms of function evaluations to choose S and T_j as described in (5.40) and (5.41) over a minimal poised set for the GCSH.

It does not seem to be possible to obtain an order-1 accurate approximation of all off-diagonal entries of a Hessian with less than $\frac{n(n+1)}{2} + 1$ function evaluations, and an order-2 accurate approximation with less than n^2+n+1 function evaluations. An obvious future research direction is to investigate this conjecture and mathematically prove (disprove) it.

In the next section, how to approximate one row of the Hessian is discussed.

Approximating a row of the Hessian

In this section, we discuss how to approximate some, or all, entries of a row in the Hessian. Since the Hessian is symmetric, approximating a row also provides an approximation of the corresponding column.

Let $M \in \mathbb{R}^{n \times n}$. We denote by $R_i \in \mathbb{R}^{n \times n}$ the square matrix such that $R_i = \text{Diag}(e^i)M$ for all $i \in \{1, \dots, n\}$. We begin by introducing the following lemma.

Lemma 5.39. *Let $M \in \mathbb{R}^{n\times n}$, $S = he^i \in \mathbb{R}^n$ where $h \neq 0$, and $\overline{T} \in \mathbb{R}^{n\times k}$. Define $R_i = \operatorname{Diag}(e^i)M$ for all $i \in \{1,\dots,n\}$. Then for all $i \in \{1,\dots,n\}$,*

$$\operatorname{Proj}_{S,\overline{T}} M = \operatorname{Proj}_{S,\overline{T}} R_i.$$

Proof. We have

$$\begin{aligned}
\operatorname{Proj}_{S,\overline{T}} M &= ((he^i)^\top)^\dagger (he^i)^\top M\overline{T}\overline{T}^\dagger \\
&= e^i(e^i)^\top M\overline{T}\overline{T}^\dagger \\
&= R_i\overline{T}\overline{T}^\dagger \\
&= (e^i)(e^i)^\top R_i\overline{T}\overline{T}^\dagger \\
&= ((he^i)^\top)^\dagger (he^i)^\top R_i\overline{T}\overline{T}^\dagger = \operatorname{Proj}_{S,\overline{T}} R_i. \qquad \square
\end{aligned}$$

In words, the previous result says that the projection onto S and $\overline{T}$ of a matrix M is equal to the projection onto S and $\overline{T}$ of row j of this matrix whenever $S = he^i$.

When S and $\overline{T}$ are defined as in the previous proposition, S is full column rank and it follows from Proposition 5.12 (*ii*) that $\nabla_s^2 f(x^0; S, \overline{T}) = \operatorname{Diag}(e^i)\nabla_s^2 f(x^0; S, \overline{T})$ and $\nabla_c^2 f(x^0; S, \overline{T}) = \operatorname{Diag}(e^i)\nabla_c^2 f(x^0; S, \overline{T})$. Moreover, the projection of the Hessian is

$$\operatorname{Proj}_{S,\overline{T}} \nabla^2 f(x^0) = \operatorname{Proj}_{S,\overline{T}} \operatorname{Diag}(e^i)\nabla^2 f(x^0)$$

for all $i \in \{1,\dots,n\}$.

Next, we present two error bounds; one for the GSH and one for the GCSH. These error bounds follow immediately from Theorems 5.13 (*iii*) and 5.14 (*iii*), respectively.

Corollary 5.40 (General error bounds for one row of a Hessian)**.** *Let $f : \operatorname{dom} f \subseteq \mathbb{R}^n \to \mathbb{R}$ be $\mathcal{C}^4$ on $B_n(x^0; \overline{\Delta})$ where $x^0 \in \operatorname{dom} f$ is the point of interest and $\overline{\Delta} > 0$. Denote by $L_{\nabla^2 f} \geq 0$ and $L_{\nabla^3 f} \geq 0$ the Lipschitz constant of $\nabla^2 f$ and $\nabla^3 f$ on $\overline{B}_n(x^0; \overline{\Delta})$, respectively. Let $S = he^i \in \mathbb{R}^n$ where $h \neq 0$, and $\overline{T} \in \mathbb{R}^{n\times k}$. Assume that $B_n(x^0 + he^i; \Delta_T) \subset B_n(x^0; \overline{\Delta})$. Then*

(i)

$$\begin{aligned}
&\left\| \operatorname{Proj}_{S,\overline{T}} \nabla_s^2 f(x^0; S, \overline{T}) - \operatorname{Proj}_{S,\overline{T}} \nabla^2 f(x^0) \right\| \\
&= \left\| \nabla_s^2 f(x^0; S, \overline{T}) - \operatorname{Proj}_{S,\overline{T}} \operatorname{diag}(e^j)\nabla^2 f(x^0) \right\| \\
&\leq 4\sqrt{k} L_{\nabla^2 f} \left(\frac{\Delta_u}{\Delta_l} \right) \|\widehat{T}^\dagger\| \Delta_u,
\end{aligned}$$

and

(ii)

$$\begin{aligned}&\left\|\mathrm{Proj}_{S,\overline{T}}\,\nabla_c^2 f(x^0;S,\overline{T}) - \mathrm{Proj}_{S,\overline{T}}\,\nabla^2 f(x^0)\right\|\\ &= \left\|\nabla_c^2 f(x^0;S,\overline{T}) - \mathrm{Proj}_{S,\overline{T}}\,\mathrm{Diag}(e^i)\nabla^2 f(x^0)\right\|\\ &\leq 2\sqrt{k}L_{\nabla^3 f}\left(\frac{\Delta_u}{\Delta_l}\right)\|\widehat{T}^\dagger\|\Delta_u^2.\end{aligned}$$

Note that $\|(\widehat{S}^\top)^\dagger\|$ does not appear in the previous error bounds since $\|(\widehat{S}^\top)^\dagger\| = 1$.

One simple choice to approximate all entries of the i^{th} row is to choose

$$S = he^i,\ \overline{T} = h\mathrm{Id}_n$$

where $h \neq 0$. In this case, $\nabla_s^2 f(x^0;S,\overline{T})$ is an order-1 accurate approximation of the whole i^{th} row of the Hessian. This choice uses the set of sample points $x^0, x^0+he^i, x^0\oplus h\mathrm{Id}_n$ and $x^0+he^i\oplus h\mathrm{Id}_n$. In this case, the number of function evaluations is

$$1+1+n+n-1 = 2n+1.$$

We subtract one in the previous equation since one point is reused: x^0+he^i. Note that $2n+1 \leq (n+1)(n+2)/2$ for all $n \in \{1,2,\dots\}$. Therefore, if we are only interested by the entries of row i, setting S and $\overline{T}$ in this fashion saves function evaluations compared to using a minimal poised set for the GSH.

To obtain an order-2 accurate approximation of the whole i^{th} row of the Hessian, we may choose once again

$$S = he^i, \overline{T} = h\mathrm{Id}_n,$$

where $h \neq 0$. In this case, the set of sample points is $x^0, x^0 \pm he^i, x^0 \oplus (\pm h\mathrm{Id}_n), x^0+he^i\oplus h\mathrm{Id}$, and $x^0-he^i\oplus -\mathrm{Id}_n$. Two sample points are reused: $x^0 \pm he^i$. The number of function evaluations is

$$1+2(2n) = 4n+1.$$

Note that $4n+1 < n^2+n+1$ when $n \geq 4$. Therefore, if $n \in \{1,2,3\}$, then using a minimal poised set for GCSH is preferable since it uses less function evaluations.

It seems that the minimum number of function evaluations to obtain an order-1 accurate approximation of a full row (column) in a Hessian is $2n + 1$. To obtain an order-2 accurate approximation of a full row, the minimum amount seems to be n^2+n+1 when $n \in \{1, 2, 3\}$ and $4n+1$ when $n \geq 4$. Future research could focus on mathematically proving (disproving) this claim.

5.6 Summary and future research directions

In this chapter, we presented the generalized simplex Hessian (GSH), an explicit and compact formula for approximating the Hessian of a function f. The GSH is well-defined as long as the matrices S and T_j are non-empty. In particular, the matrices S and T_j do not need to be full row rank nor square to compute the GSH. We also presented a "centered" version of the GSH called the generalized centered simplex Hessian (GCSH) and presented its relation to the GSH.

In Theorems 5.13 and 5.14, we developed error bounds for the GSH and GCSH. Under some assumptions, the GSH is an order-1 accurate approximation of the partial Hessian $\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)$, and the GCSH is an order-2 accurate approximation of the partial Hessian $\text{Proj}_{S,T_{1:m}} \nabla^2 f(x^0)$. When all matrices of directions in the set $T_{1:m}$ are equal, then the projection of the GSH (GCSH) is equal to the GSH (GCSH). we developed error bounds between the projection of the GSH (GCSH) and the projection of the true Hessian that fully covers all 16 possible cases defined in Definition 5.10. When all matrices T_j are not equal, then we proposed error bounds between the GSH (GCSH) and the projection of the true Hessian. If S is full column rank or T_j is full row rank for all j, then the GSH (GCSH) is equal to its projection and we proposed an error between the projection of the GSH (GCSH) and the projection of the true Hessian.

In Section 5.4, we defined a minimal poised set for the GSH. When such a set is used to compute a GSH, the number of distinct function evaluations required is $(n + 1)(n + 2)/2$. We investigated the relationship between a minimal poised set for the GSH and poisedness for quadratic interpolation, proving that a minimal poised set for the GSH of the form $\mathcal{M}(x^0; S, \overline{T})$ where $\overline{T} = U_\ell$ is well-poised for quadratic interpolation, but that the converse does not necessarily hold. In this case, we know that $\nabla^2_s f(x^0; S, \overline{T})$ is equal to the Hessian of the quadratic interpolation function $Q_f(x^0; S, \overline{T})$. We also developed explicit formulae for obtaining the parameters of the quadratic interpolation function of f over a minimal poised set for the GSH of the

form $\mathcal{M}(x^0; S, \overline{T})$, where $T = U_\ell$.

A minimal poised set for the GCSH is defined and it demonstrates how to obtain an order-2 accurate approximation of the full Hessian with n^2+n+1 distinct sample points. A future research direction could be to investigate if there exist other possible choices other than $\overline{T} = -S$ that makes the set of sample points a minimal poised set for the GCSH. Furthermore, proving that $n^2 + n + 1$ is the minimal number of function evaluations to obtain an order-2 accurate approximation of the full Hessian via this approximation technique is a future research direction.

In Section 5.5, we provided details on how to choose the matrices S and T_j when we are only interested in a proper subset of the entries of the Hessian. In particular, we investigated how to approximate the diagonal entries of a Hessian, the off-diagonal entries of a Hessian and a row of a Hessian. The number of function evaluations to obtain an order-1 accurate approximation, or an order-2 accurate approximation of the entries of the Hessian of interest has been discussed. The relation between the CSHD introduced in [JB22] and the GCSH is clarified.

Future research should explore if the sets $U_0, U_1, \ldots, U_n$ as defined in Proposition 5.18 are the only possible choices for $\overline{T} \in \mathbb{R}^{n\times n}$ such that $\mathcal{S}_s(x^0; S, \overline{T})$, where $S \in \mathbb{R}^{n\times n}$ is full rank, is a minimal poised set for GSH at x^0. It can be proved that it is indeed the case by using brute force in $\mathbb{R}$ and $\mathbb{R}^2$, but it is still unclear how to generalized this claim in an arbitrary dimension n.

When a matrix of directions is nondetermined, it is possible to remove columns of the matrix to make it full rank. This may lead to a GSH (GCSH) with the property that its projection is equal to the GSH (GCSH). An efficient method to accomplish this task could be explored.

Another future research direction could be to develop an approximation technique for higher derivatives. For example, inspired from the formula for the GSH, an approximation technique of the three-dimensional matrix containing all third-order derivatives could be developed.

On a final note, Matlab implementations of the GSH and the GCSH are available upon request.

Chapter 6

Positive spanning sets

Positive spanning sets and positive bases have been studied since at least the 1950s. The theory was first developed by Davis in [Dav54] and McKinney in [McK62]. In the last few decades, their popularity has drastically increased due to their value in blackbox optimization. The value of positive spanning sets in blackbox optimization was revealed in 1996 [LT96] when it was shown that if the gradient of a function at a point exists and is nonzero, then there exists a vector d in any positive spanning set such that d is a descent direction of the function at that point. It follows that positive spanning sets make it possible to define a *stationarity condition* in terms of a finite number of directions rather than in terms of all directions [BH20].

Since then, several DFO algorithms relying on positive spanning sets have been developed. More specifically, positive spanning sets are employed in direct search methods such as pattern search [CDV08, Tor97, VV09], generalized pattern search [AH17], grid-based methods [CP01, CP02], generating set search [KLT03], mesh adaptive direct search [AADLD09, AD06, Aud14] and implicit filtering [Kel11].

A handful of papers have focused on the theory behind positive spanning sets and their characterization [Dav54, Gry22, McK62, Reg16c, Rom87]. Davis established that the maximal size s of a positive basis in dimension n is $s = 2n$ [Dav54]. A shorter proof of this result was published in 2011 by Audet [Aud11]. Also, it is straightforward to show that the minimal size s of a positive basis is $s = n+1$ [Dav54]. Minimal and maximal positive bases are now well-understood and their structure can be rigorously characterized [Reg16c]. A few results on intermediate positive bases ($n+1 < s < 2n$) can be found in [Rea65, Rea66, Rom87, She71].

With regards to blackbox optimization, the key instrument to measure the quality of a positive spanning set is called *cosine measure* [KLT03] (see also [Tor97]). The cosine measure is used to indicate the quality of a positive spanning set and it is this measure that determines whether the positive spanning set is *optimal*. Roughly speaking, a high value of the cosine measure indicates that the positive spanning set covers the space more uniformly. The convergence properties of certain DFO methods depend on the cosine

measure of the positive spanning sets employed. However, before 2020, no deterministic methods were proposed to calculate the cosine measure of a given positive spanning set.

In [Næv18], the maximal cosine measures for maximal and minimal positive bases are found and the structure of positive bases attaining these upper bounds for the cosine measure are characterized. However, the maximal cosine measures for intermediate positive bases was not developed.

The two main goals of this chapter are the following. In Section 6.2, we introduce the first deterministic algorithm to compute the cosine measure of any finite positive spanning set. The algorithm is proven to return the exact value of the cosine measure in finite time. Hence, this section provides a procedure to compare positive bases to each other. In Section 6.3, we focus on finding the structure of intermediate positive bases with maximal cosine measure. We define two subsets of positive bases with nice properties. We investigate the problem of finding an intermediate positive basis with maximal cosine measure on these two subsets. We develop properties of this type of intermediate positive bases and show that the algorithm to compute the cosine measure introduced in Section 6.2 can be simplified in the presence of such positive bases.

We begin by presenting background results and notation that is useful to understand this chapter.

6.1 Preliminaries

Similar to the previous chapters, it will be convenient to regard a set of vectors as a matrix whose columns are the vectors in the set. In this chapter, we always assume that the sets of vectors considered do not contain the zero vector and do not contain repeated vectors. The definitions of a *block diagonal matrix* and an *elementary block diagonal matrix* follow.

Definition 6.1 (Block diagonal matrix)**.** A *block diagonal matrix* $A \in \mathbb{R}^{n\times m}$ is a block matrix such that the diagonal blocks $M_i \in \mathbb{R}^{n_i\times m_i}$ are matrices of any size $1 \leq m_i \leq m$, $1 \leq n_i \leq n$, and all off-diagonal blocks are zero matrices.

Note that we allowed the matrix A and the block matrices to be non-square in the previous definition. In this section, a block diagonal matrix with q diagonal blocks M_i is written as $\mathrm{Diag}(M_1, \ldots, M_q)$. Note that it is possible to write a block diagonal matrix in different ways.

Example 6.2. The matrix $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ can be written as a block diagonal matrix with:

- One block: $A = \text{Diag}(M_1)$ where $M_1 = A$.
- Two blocks: $A = \text{Diag}(M_1, M_2)$ where $M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $M_2 = \begin{bmatrix} 1 \end{bmatrix}$.
- Three blocks: $A = \text{Diag}(M_1, M_2, M_3)$ where $M_1 = M_2 = M_3 = \begin{bmatrix} 1 \end{bmatrix}$.

In the next definition, we say that a matrix $A \in \mathbb{R}^{n \times n}$ is *permutation-similar* to a matrix $B \in \mathbb{R}^{n \times n}$ if there exists a permutation matrix $P \in \mathbb{R}^{n \times n}$ such that $B = PAP^\top$. Otherwise, we say that A is not permutation-similar to B.

Definition 6.3 (Elementary block diagonal matrix)**.** Let $A \in \mathbb{R}^{n \times n}$. We say that A is an *elementary block diagonal matrix* if and only A is not permutation-similar to a block diagonal matrix in $\mathbb{R}^{n \times n}$ with 2 diagonal blocks.

If a diagonal block of a block diagonal matrix A satisfies the previous definition, we say that this block is an *elementary diagonal block of A.*

Example 6.4.

- The matrix $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ is an elementary block diagonal matrix.
- The matrix $B = \begin{bmatrix} 5 & 0 & 4 \\ 0 & 1 & 0 \\ 3 & 0 & 2 \end{bmatrix}$ is **not** an elementary block diagonal matrix since B is permutation-similar to $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 3 \\ 0 & 4 & 5 \end{bmatrix}$ which has two diagonal blocks.
- In Example 6.2, the diagonal block $M_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ of A is **not** elementary. The diagonal block $M_2 = \begin{bmatrix} 1 \end{bmatrix}$ of A is elementary.

The definition of a positive span and a positive spanning set of $\mathbb{R}^n$ follow.

Definition 6.5 (Positive span and positive spanning set of $\mathbb{R}^n$)**.** The *positive span* of a finite set of vectors $\mathbb{S} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ in $\mathbb{R}^n$, denoted by pspan($\mathbb{S}$), is the set

$$\{v \in \mathbb{R}^n : v = \alpha_1 d^1 + \cdots + \alpha_s d^s, \alpha_j \geq 0, j = 1, 2, \ldots, s\}.$$

A *positive spanning set* of $\mathbb{R}^n$ of size s, denoted by $\mathbb{P}_{n,s}$, is a set of s non-zero vectors such that pspan($\mathbb{P}_{n,s}$) = $\mathbb{R}^n$.

Technically, the term non-negative span and non-negative spanning set would be more appropriate in the previous definition. To be consistent with the literature, we did not rename these objects.

The *relative interior of* pspan($\mathbb{S}$) is denoted by pspan$^+$($\mathbb{S}$), and is equal to

$$\text{pspan}^+(\mathbb{S}) = \{v \in \mathbb{R}^n : v = \alpha_1 d^1 + \cdots + \alpha_s d^s, \alpha_j > 0, j = 1, 2, \ldots, s\}. \tag{6.1}$$

To define a positive basis of $\mathbb{R}^n$, the concept of *positive independence* is introduced.

Definition 6.6 (Positive independence)**.** A set of vectors

$$\mathbb{S} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$$

contained in $\mathbb{R}^n$ is *positively independent* if and only if $d^j \notin \text{pspan}(\mathbb{S} \setminus d^j)$ for all $j \in \{1, 2, \ldots s\}$.

Definition 6.7 (Positive basis of $\mathbb{R}^n$)**.** A *positive basis* of $\mathbb{R}^n$ of size s, denoted by $\mathbb{D}_{n,s}$, is a positively independent set of s vectors whose positive span is $\mathbb{R}^n$.

Equivalently, a positive basis of $\mathbb{R}^n$ can be defined as a set of non-zero vectors of $\mathbb{R}^n$ whose positive span is $\mathbb{R}^n$, but for which no proper subset exhibits the same property.

It is known that the size s of a positive basis of $\mathbb{R}^n$ is between $n + 1$ and $2n$ inclusively. When $s = n + 1$, we say the positive basis is a *minimal positive basis*. A minimal positive basis of $\mathbb{R}^n$ will be written $\mathbb{D}_n$ (the second argument in the subscript referring to the size may be omitted when $s = n + 1$). When $n + 1 < s < 2n$, we say the positive basis is an *intermediate positive basis*. Note that there is no positive bases of intermediate size when $n \in \{1, 2\}$. When $s = 2n$, we say the positive basis is a *maximal positive basis*. If $s > n + 1$, we say the positive basis is a *non-minimal positive basis*.

The following theorem describes the structure of maximal positive bases.

Theorem 6.8. *[Reg16c, Theorem 6.3] Suppose* $\mathbb{B}_n = \begin{bmatrix} d^1 & d^2 & \cdots & d^n \end{bmatrix}$ *is a basis of* $\mathbb{R}^n$*. Then for any choice of* $\alpha_1, \ldots, \alpha_n > 0$*, the set*

$$\mathbb{D}_{n,2n} = \begin{bmatrix} d^1 & \cdots & d^n & -\alpha_1 d^1 & \cdots & -\alpha_n d^n \end{bmatrix}$$

is a positive basis of $\mathbb{R}^n$*. Conversely, every maximal positive basis of* $\mathbb{R}^n$ *has the form* $\mathbb{D}'_{n,2n} = \begin{bmatrix} d^1 & \cdots & d^n & -\alpha_1 d^1 & \cdots & -\alpha_n d^n \end{bmatrix}$ *up to reordering of the vectors, where* $\mathbb{B}'_n = \begin{bmatrix} d^1 & \cdots & d^n \end{bmatrix}$ *is a basis of* $\mathbb{R}^n$ *and* $\alpha_1, \ldots, \alpha_n > 0$*.*

The next proposition introduces a property of a positive spanning set of $\mathbb{R}^n$.

Proposition 6.9. *[Reg16c, Theorem 2.3] If* $\mathbb{S} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ *positively spans* $\mathbb{R}^n$*, then* $\mathbb{S} \setminus \{d^j\}$ *linearly spans* $\mathbb{R}^n$ *for any* $j \in \{1, \ldots, s\}$*.*

The principal tool for determining the quality of a positive spanning set and how well it covers the space $\mathbb{R}^n$ is the cosine measure.

Definition 6.10 (Cosine Measure)**.** The *cosine measure* of a finite set $\mathbb{S}$ of non-zero vectors is defined by

$$\operatorname{cm}(\mathbb{S}) = \min_{\substack{\|u\|=1 \\ u \in \mathbb{R}^n}} \max_{d \in \mathbb{S}} \frac{u^\top d}{\|d\|}.$$

Definition 6.11 (The cosine vector set)**.** Let $\mathbb{S}$ be a set of non-zero vectors in $\mathbb{R}^n$. The *cosine vector set of* $\mathbb{S}$, denoted by $c\mathbf{V}(\mathbb{S})$, is defined as

$$c\mathbf{V}(\mathbb{S}) = \operatorname*{argmin}_{\substack{\|u\|=1 \\ u \in \mathbb{R}^n}} \max_{d \in \mathbb{S}} \frac{u^\top d}{\|d\|}.$$

Note that the length of the vectors in $\mathbb{S}$ does not affect the value of the cosine measure. For this reason, unless mentioned otherwise, we assume that all vectors in the sets considered are unit vectors in the remainder of the this chapter. Note that the cosine vector set $c\mathbf{V}(\mathbb{S})$ is non-empty and may contain more than one vector.

Values of the cosine measure near zero suggest the positive spanning property is approaching a deterioration. A high value of cosine measure indicates that the vectors in the set more uniformly cover the space. In other words, the vectors are spaced farther away from one another.

When $n = 1$, there is only one positive basis of unit vectors:

$$\mathbb{D}_1 = \begin{bmatrix} 1 & -1 \end{bmatrix}.$$

Its cosine measure is equal to 1. The positive basis $\mathbb{D}_1$ is both minimal size and maximal size. Note that given any finite positive spanning set $\mathbb{P}_{n,s}$ where $n \geq 2$, the cosine measure is bounded by $0 < \text{cm}(\mathbb{P}_{n,s}) < 1$. To prove these bounds, we recall a proposition that helps to prove the lower bound.

Proposition 6.12. *[CSV09b, Theorem 2.3] Let $\mathbb{S} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ be a set of non-zero vectors in $\mathbb{R}^n$. Then $\mathbb{S}$ is a positive spanning set of $\mathbb{R}^n$ if and only if the following holds:*

(i) For every non-zero vector v in $\mathbb{R}^n$, there exists an index $j \in \{1, \ldots, s\}$ such that $v^\top d^j < 0$.

(ii) For every non-zero vector w in $\mathbb{R}^n$, there exists an index $\overline{j} \in \{1, \ldots, s\}$ such that $w^\top d^{\overline{j}} > 0$.

Proposition 6.13. *Let $\mathbb{P}_{n,s} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ be a finite positive spanning set of $\mathbb{R}^n$ where $n \geq 2$. Then the cosine measure of $\mathbb{P}_{n,s}$ is bounded by*

$$0 < \text{cm}(\mathbb{P}_{n,s}) < 1.$$

Proof. Without loss of generality, assume that d^j are unit vectors for all $j \in \{1, 2, \ldots, s\}$. Let $u_* \in c\mathbf{V}(\mathbb{P}_{n,s})$. By Proposition 6.12, there exists a vector $d^j \in \mathbb{P}_{n,s}$ such that $u_*^\top d^j > 0$. Therefore, $\text{cm}(\mathbb{P}_{n,s}) \geq (u_*^\top d^j > 0$.

Consider the upper bound. Since $\mathbb{P}_{n,s}$ is a finite set of vectors, there exists a non-zero unit vector u such that $u \neq d^j$ for all $j \in \{1, 2, \ldots, s\}$. Since the dot product of unit vectors is equal to 1 if and only if the two vectors are equal, it follows that $u^\top d^j < 1$ for all $j \in \{1, 2, \ldots, s\}$ and so $\max_{d \in \mathbb{P}_{n_s}} u^\top d < 1$. Therefore, the cosine measure $\text{cm}(\mathbb{P}_{n,s}) < 1$. □

Note the fact that $\mathbb{P}_{n,s}$ is a positive spanning set of $\mathbb{R}^n$ is not used for the upper bound, only the fact that $\mathbb{P}_{n,s}$ is a finite set is sufficient. Next, we define an *optimal positive basis*.

Definition 6.14 (Optimal positive basis of $\mathbb{R}^n$)**.** A positive basis $\mathbb{D}_{n,s}$ of $\mathbb{R}^n$ is said to be *optimal* over a non-empty set $\mathbb{S}$ if

$$\text{cm}(\mathbb{D}_{n,s}) \geq \text{cm}(\mathbb{D}'_{n,s})$$

for any positive basis $\mathbb{D}'_{n,s}$ of $\mathbb{R}^n$ in $\mathbb{S}$.

The set of positive bases containing all positive bases of size s in $\mathbb{R}^n$ will be denoted by $\mathcal{P}_{n,s}$. When we write that $\mathbb{D}_{n,s}$ is optimal without mentioning the set considered, it is implied that the set considered is $\mathcal{P}_{n,s}$. As mentioned

earlier, the structure and properties of optimal bases of minimal size and maximal size are well-known and have been rigorously proved in [Næv18]. We will denote by $\mathring{\mathbb{D}}_n$ an optimal positive basis of minimal size. Note that in $\mathbb{R}$, the only positive basis of unit vectors $\mathbb{D}_1 = \begin{bmatrix} 1 & -1 \end{bmatrix}$ is an optimal positive basis of minimal size and hence, it could be denoted by $\mathring{\mathbb{D}}_1$. The definition of the *all activity set* follow.

Definition 6.15 (The active set of vectors and the all activity set)**.** Let $\mathbb{S}$ be a set of non-zero vectors in $\mathbb{R}^n$ and let $u_* \in c\mathbf{V}(\mathbb{P}_{n,s})$. The *active set of vectors in* $\mathbb{S}$ *on* u_* is denoted by $\mathbf{A}(u_*, \mathbb{S})$ and defined by

$$\mathbf{A}(u_*, \mathbb{S}) = \left\{ \frac{d^\top}{\|d\|} \in \mathbb{S} : \frac{d^\top u_*}{\|d\|} = \mathrm{cm}(\mathbb{S}) \right\}.$$

The *all activity set of* $\mathbb{S}$ is denoted by $\overline{\mathbf{A}}(\mathbb{S})$, and defined by

$$\overline{\mathbf{A}}(\mathbb{S}) = \bigcup_{u \in c\mathbf{V}(\mathbb{S})} \mathbf{A}(u, \mathbb{S}).$$

The definition of *Gram matrices* and two lemmas that are helpful in Section 6.2 are presented next.

Definition 6.16 (Gram matrix)**.** Let $\mathbb{S} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ be vectors in $\mathbb{R}^n$ with dot product $(d^i)^\top d^j$. The Gram matrix of the vectors $d^1, d^2, \ldots, d^s$ with respect to the dot product, denoted by $\mathbf{G}(\mathbb{S})$, is given by $\mathbf{G}(\mathbb{S}) = \mathbb{S}^\top \mathbb{S} \in \mathbb{R}^{s \times s}$.

Lemma 6.17. *[Næv18, Lemma 1] Let* $\mathbb{B}_n = \begin{bmatrix} d^1 & d^2 & \cdots & d^n \end{bmatrix}$ *be a basis of unit vectors in* $\mathbb{R}^n$*. Then there exists a unit vector* $u_{\mathbb{B}_n} \in \mathbb{R}^n$ *such that* $u_{\mathbb{B}_n}^\top d^i = \gamma_{\mathbb{B}_n} > 0$ *for all* $i \in \{1, 2, \ldots, n\}$ *where*

$$\gamma_{\mathbb{B}_n} = \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1} \mathbf{1}}}.$$

Note that the unit vector $u_{\mathbb{B}_n}$ such that $u_{\mathbb{B}_n}^\top d^i = \gamma_{\mathbb{B}_n}$ for all i's is unique since $\{d^1, \ldots, d^n\}$ is a set of n linearly independent vectors. Also, note that $\gamma_{\mathbb{B}_n} < 1$ whenever $n \geq 2$.

In fact, the positive value of the n equal dot products is unique whenever u is a unit vector and $\{d^1, \ldots, d^n\}$ is a set of linearly independent vectors.

Lemma 6.18. *Let* $\mathbb{B}_n = \begin{bmatrix} d^1 & d^2 & \cdots & d^n \end{bmatrix}$ *be a basis of unit vectors in* $\mathbb{R}^n$*. Suppose* u *is a unit vector such that* $u^\top d^1 = \cdots = u^\top d^n = \alpha > 0$*. Then* $\alpha = \gamma_{\mathbb{B}_n}$*, where* $\gamma_{\mathbb{B}_n}$ *is defined as in Lemma 6.17.*

Proof. It suffices to show $\alpha > 0$ is unique, so that Lemma 6.17 implies it must be the value $\gamma_{\mathbb{B}_n}$.

Suppose there exists two distinct unit vectors, say u and u' such that

$$u^\top d^1 = \cdots = u^\top d^n = \alpha > 0 \text{ and } u'^\top d^1 = \cdots = u'^\top d^n = \alpha' > 0.$$

Since $\mathbb{B}_n$ is a basis, it follows that $\alpha \neq \alpha'$ and there exists $\rho_1,\ldots,\rho_n \in \mathbb{R}$ such that $\sum_{i=1}^n \rho_i d^i = u$. Multiplying both sides by by $u'^\top$, shows that $\sum_{i=1}^n \rho_i \alpha' = u'^\top u$. Alternately, multiplying both sides by $u^\top$ yields $\sum_{i=1}^n \rho_i \alpha = 1$. Letting $\bar{\rho} = \sum_{i=1}^n \rho_i$, provides

$$\bar{\rho}\alpha' = u^\top u' \text{ and } \bar{\rho}\alpha = 1.$$

Similarly, since $\mathbb{B}_n$ is a basis, there exist $\beta_1, \ldots, \beta_n \in \mathbb{R}$ such that

$$\sum_{i=1}^n \beta_i d^i = u'.$$

Letting $\bar{\beta} = \sum_{i=1}^n \beta_i$ and multiplying this by $u^\top$ and $u'^\top$, yields

$$\bar{\beta}\alpha = u^\top u' \text{ and } \bar{\beta}\alpha' = 1.$$

Applying $\bar{\rho} = 1/\alpha$ and $\bar{\beta} = 1/\alpha'$ into the second equality, we get $\frac{\alpha'}{\alpha} = \frac{\alpha}{\alpha'}$. Since $\alpha > 0$ and $\alpha' > 0$, this yields $\alpha = \alpha'$. □

Now we recall the definition of a *principal submatrix* and properties of *positive definite matrices*.

Definition 6.19 (Principal submatrix)**.** [HJ90, Section 0.7.1] Let $A \in \mathbb{R}^{n\times m}$. For index sets $\alpha \subseteq \{1,\ldots,n\}$ and $\beta \subseteq \{1,\ldots,m\}$, we denote by $A[\alpha,\beta]$ the submatrix of entries that lie in the rows of A indexed by α and the columns indexed by β. If $\alpha = \beta$, the submatrix $A[\alpha,\alpha]$ is a principal submatrix of A.

Lemma 6.20. [HJ90, Obs. 7.1.2] *Let $A \in \mathbb{R}^{n\times n}$ be a real symmetric matrix. If A is positive definite, then all of its principal submatrices are positive definite.*

Lemma 6.21. [HJ90, Theorem 7.2.7] *Let A be a symmetric matrix in $\mathbb{R}^{n\times n}$. The matrix A is positive definite if and only if there is a $B \in \mathbb{R}^{m\times n}$ with full column rank such that $A = B^\top B$.*

Lemma 6.22. [HJ90, Theorem 7.2.1] *A non-singular symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive definite if and only if A^{-1} is positive definite.*

Note that Lemma 6.21 and Lemma 6.22 explain why the real number $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1}\mathbf{1}$ involved in the definition of $\gamma_{\mathbb{B}_n}$ (Lemma 6.17) is positive for any basis $\mathbb{B}_n$ of $\mathbb{R}^n$.

The last lemma of this section will be helpful to find the cosine measure of a positive basis $\mathbb{D}_{n,s}$ when a basis $\mathbb{B}_n \in \mathbf{A}(u, \mathbb{D}_{n,s})$ is written as a block matrix.

Lemma 6.23 (Inverse of a block matrix)**.** [LS02a, Corollary 4.1] *Let $G \in \mathbb{R}^{n\times n}$ be a positive definite matrix of the form*

$$G = \begin{bmatrix} A & B^\top \\ B & D \end{bmatrix}$$

where A is a symmetric invertible matrix and D is a symmetric matrix. Then

$$G^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B^\top (D - BA^{-1}B^\top)^{-1}BA^{-1} & -A^{-1}B^\top (D - BA^{-1}B^\top)^{-1} \\ -(D - BA^{-1}B^\top)^{-1}BA^{-1} & (D - BA^{-1}B^\top)^{-1} \end{bmatrix}.$$

Next we introduce a deterministic algorithm to compute the cosine measure of any finite positive spanning set.

6.2 An algorithm to compute the cosine measure of a finite positive spanning set

In this section, a deterministic algorithm that returns the cosine measure for any finite positive spanning set $\mathbb{P}_{n,s}$ of $\mathbb{R}^n$ (or any positive basis $\mathbb{D}_{n,s}$ of $\mathbb{R}^n$) is provided. After introducing the algorithm, it is shown that the algorithm returns the exact value of the cosine measure.

Algorithm 1: The cosine measure of a finite positive spanning set of $\mathbb{R}^n$

Given $\mathbb{P}_{n,s}$, a finite positive spanning set of $\mathbb{R}^n$,

1. For all bases $\mathbb{B}_n \subset \mathbb{P}_{n,s}$, compute

(1.1) $\gamma_{\mathbb{B}_n} = \dfrac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1}\mathbf{1}}}$,

(1.2) $u_{\mathbb{B}_n} = \mathbb{B}_n^{-\top}\gamma_{\mathbb{B}_n}\mathbf{1}$ (unit vector associated to $\gamma_{\mathbb{B}_n}$),

(1.3) $p_{\mathbb{B}_n} = [p_{\mathbb{B}}^1 \ \cdots \ p_{\mathbb{B}}^s] = u_{\mathbb{B}_n}^\top \mathbb{P}_{n,s}$ (dot product vector),

(1.4) $p_{\mathbb{B}_n}^{\max} = \max\limits_{1\leq j\leq s} p_{\mathbb{B}_n}^j$ (maximum value in $p_{\mathbb{B}_n}$).

2. Return

(2.1) $\mathrm{cm}(\mathbb{P}_{n,s}) = \min\limits_{\mathbb{B}_n \subset \mathbb{P}_{n,s}} p_{\mathbb{B}_n}^{\max}$ (cosine measure of $\mathbb{P}_{n,s}$),

(2.2) $c\mathbf{V}(\mathbb{P}_{n,s}) = \{u_{\mathbb{B}_n} : p_{\mathbb{B}_n}^{\max} = \mathrm{cm}(\mathbb{P}_{n,s})\}$ (cosine vector set of $\mathbb{P}_{n,s}$).

The previous algorithm investigates all the bases contained in $\mathbb{P}_{n,s}$. Note that any finite positive spanning set of $\mathbb{R}^n$ contains at least one basis of $\mathbb{R}^n$ (by Proposition 6.9). An obvious upper bound for the maximal number of bases contained in a finite positive spanning set $\mathbb{P}_{n,s}$ is $\binom{s}{n}$. The exact number of bases contained in minimal positive bases ($s = n+1$) and maximal positive bases ($s = 2n$) are easily derived.

Proposition 6.24. *Let $\mathbb{D}_n$ be a minimal positive basis of $\mathbb{R}^n$. Then $\mathbb{D}_n$ contains $n+1$ bases of $\mathbb{R}^n$.*

Proof. From Proposition 6.9, any set of n vectors is a basis of $\mathbb{R}^n$. Therefore, $\mathbb{D}_n$ contains $\binom{n+1}{n} = n+1$ bases of $\mathbb{R}^n$. □

Proposition 6.25. *Let $\mathbb{D}_{n,2n}$ be a maximal positive basis. Then $\mathbb{D}_{n,2n}$ contains 2^n bases of $\mathbb{R}^n$.*

Proof. By Theorem 6.8 and since multiplying a column of a matrix by a non-zero real number is an elementary operation that does not affect the rank of a matrix, we may assume that $\mathbb{D}_{n,2n}$ has the form

$$\mathbb{D}_{n,2n} = \begin{bmatrix} d^1 & d^2 & \cdots & d^n & -d^1 & -d^2 & -d^n \end{bmatrix},$$

where $d^i \in \mathbb{R}^n$ is a non-zero vector for all $i \in \{1, 2, \ldots, n\}$. Note that any basis contained in $\mathbb{D}_{n,2n}$ has the form $\mathbb{B}_n = \begin{bmatrix} \pm d^1 & \pm d^2 & \cdots & \pm d^n \end{bmatrix}$. Therefore, $\mathbb{D}_{n,2n}$ contains 2^n bases. □

Note that 2^n is smaller than $\binom{2n}{n}$ whenever $n \in \{2, 3, \dots\}$. Finding the supremum for the number of bases contained in an intermediate positive basis ($n+1 < s < 2n$) is more challenging and will be explored in Section 6.3. Nevertheless, the number of bases in any positive basis $\mathbb{D}_{n,s}$ (or any finite positive spanning set $\mathbb{P}_{n,s}$) is a finite number greater than one and hence, the algorithm always find an exact solution in finite time.

To prove that Algorithm 1 returns the exact cosine measure of a finite positive spanning set for any size $s \in \{2, 3, \dots\}$, we introduce the following lemma.

Lemma 6.26. *Let $\epsilon \neq 0$ and let u and v be unit vectors in $\mathbb{R}^n$. Then*

(i) $\|u + \epsilon v\| = 1$ if and only if $\epsilon = -2u^\top v$, and

(ii) $\|u + \epsilon v\| < 1$ implies $\|u - \epsilon v\| > 1$.

Proof. Since

$$\|u + \epsilon v\|^2 = 1 + (2\epsilon u^\top v + \epsilon^2) \text{ and } \|u - \epsilon v\|^2 = 1 - (2\epsilon u^\top v - \epsilon^2),$$

it follows that $\|u + \epsilon v\| = 1$ if and only if $2\epsilon u^\top v + \epsilon^2 = 0$. Since $\epsilon \neq 0$, the first result follows.

Considering $\|u + \epsilon v\| < 1$, notice that

$$\begin{aligned}
\|u + \epsilon v\| < 1 &\implies \|u + \epsilon v\|^2 < 1 \\
&\implies 1 + (2\epsilon u^\top v + \epsilon^2) < 1 \\
&\implies 2\epsilon u^\top v + \epsilon^2 < 0 \\
&\implies 2\epsilon u^\top v - \epsilon^2 < -2\epsilon^2 < 0 \\
&\implies -(2\epsilon u^\top v - \epsilon^2) > 0 \\
&\implies 1 - (2\epsilon u^\top v - \epsilon^2) = \|u - \epsilon v\|^2 > 1 \\
&\implies \|u - \epsilon v\| > 1.
\end{aligned}$$

□

The previous lemma is used in the following proposition.

Proposition 6.27. *Let $\mathbb{P}_{n,s}$ be a positive spanning set of $\mathbb{R}^n$ and let $u_* \in c\mathbf{V}(\mathbb{P}_{n,s})$. Then*

$$\operatorname{span}(\mathbf{A}(u_*, \mathbb{P}_{n,s})) = \mathbb{R}^n .$$

Proof. Without loss of generality, assume that all vectors d in $\mathbb{P}_{n,s}$ are unit vectors. Suppose that $\operatorname{span}(\mathbf{A}(u_*, \mathbb{P}_{n,s})) \neq \mathbb{R}^n$, i.e., the rank of $\mathbf{A}(u_*, \mathbb{P}_{n,s})$ is strictly less than n. This implies that the kernel of $\mathbf{A}(u_*, \mathbb{P}_{n,s})$ is non-empty.

Let v be a unit vector in the kernel of $\mathbf{A}(u_*, \mathbb{P}_{n,s})$. This means that $d^\top v = 0$ for all d in $\mathbf{A}(u_*, \mathbb{P}_{n,s})$.

Notice that, if $d \in \mathbb{P}_{n,s} \setminus \mathbf{A}(u_*, \mathbb{P}_{n,s})$, then

$$d^\top u_* < \mathrm{cm}(\mathbb{P}_{n,s}).$$

Consider the vector $u_* + \epsilon v$. Then there exists an ϵ such that $0 < \epsilon < |-2u_*^\top v|$ and

$$\frac{d^\top (u_* \pm \epsilon v)}{\|u_* \pm \epsilon v\|} < \mathrm{cm}(\mathbb{P}_{n,s})$$

for all $d \in \mathbb{P}_{n,s} \setminus \mathbf{A}(u_*, \mathbb{P}_{n,s})$. Moreover, since $d^\top v = 0$, it follows that

$$\frac{d^\top (u_* \pm \epsilon v)}{\|u_* \pm \epsilon v\|} = \frac{d^\top u_*}{\|u_* \pm \epsilon v\|} \pm 0 = \frac{\mathrm{cm}(\mathbb{P}_{n,s})}{\|u_* \pm \epsilon v\|}$$

for all $d \in \mathbf{A}(u_*, \mathbb{P}_{n,s})$. By Lemma 6.26(i), $\epsilon \neq -2u_*^\top v$ implies that $\|u_* + \epsilon v\| \neq 1$. By Lemma 6.26(ii), if $\|u_* + \epsilon v\| < 1$, then $\|u_* - \epsilon v\| > 1$. Select w in $\{u_* + \epsilon v, u_* - \epsilon v\}$ such that $\|w\| > 1$. Then

$$\frac{d^\top w}{\|w\|} < \mathrm{cm}(\mathbb{P}_{n,s})$$

for all $d \in \mathbb{P}_s^n$. This contradicts the definition of cosine measure. Therefore, $\mathrm{span}(\mathbf{A}(u_*, \mathbb{P}_{n,s})) = \mathbb{R}^n$. □

Note that the positive spanning set property of $\mathbb{P}_{n,s}$ in the previous proposition is sufficient to prove the result. The positive independence property of positive bases is not necessary to obtain the result. This provide sufficient background to complete the proof that Algorithm 1 returns the exact cosine measure of any finite positive spanning set of $\mathbb{R}^n$.

Corollary 6.28. *Let $\mathbb{P}_{n,s}$ be a finite positive spanning set of $\mathbb{R}^n$ and let $u_* \in c\mathbf{V}(\mathbb{P}_{n,s})$. Then $\mathbf{A}(u_*, \mathbb{P}_{n,s})$ contains a basis of $\mathbb{R}^n$.*

This is a classical result in linear algebra. See [Bro88, Theorem 2.11] for example. It follows from the previous corollary that the all activity set $\overline{\mathbf{A}}(\mathbb{D}_{n,s})$, defined in Definition 6.15, must also contain a basis of $\mathbb{R}^n$.

Theorem 6.29. *Let $\mathbb{P}_{n,s}$ be a finite positive spanning set of $\mathbb{R}^n$. Then Algorithm 1 returns the exact value of the cosine measure* $\mathrm{cm}(\mathbb{P}_{n,s})$.

Proof. Without loss of generality, let $\mathbb{P}_{n,s} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ be a finite positive spanning set of unit vectors in $\mathbb{R}^n$ and let $u_* \in c\mathbf{V}(\mathbb{P}_{n,s})$. By Corollary 6.28, $\mathbf{A}(u_*, \mathbb{P}_{n,s})$ contains a basis of $\mathbb{R}^n$. Without loss of generality, let this basis be $\mathbb{B}_n^* = \begin{bmatrix} d^1 & d^2 & \cdots & d^n \end{bmatrix}$. Hence,

$$\operatorname{cm}(\mathbb{P}_{n,s}) = (d^1)^\top u_* = \cdots = (d^n)^\top u_* > 0,$$

where u_* is a unit vector. By Lemma 6.18, it follows that

$$\operatorname{cm}(\mathbb{P}_{n,s}) = \gamma_{\mathbb{B}_n^*} = \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n^*)^{-1}\mathbf{1}}}.$$

Note that $\gamma_{\mathbb{B}_n^*} = \max_{1\leq j\leq s}(d^j)^\top u_* = p^{\max}_{\mathbb{B}_n^*}$ since $\gamma_{\mathbb{B}_n^*} = \operatorname{cm}(\mathbb{P}_{n,s})$. Therefore, by definition of the cosine measure,

$$\min_{\mathbb{B}_n \subset \mathbb{P}_{n,s}} p^{\max}_{\mathbb{B}_n} = \operatorname{cm}(\mathbb{P}_{n,s}),$$

where $\mathbb{B}_n$ is a basis of $\mathbb{R}^n$ contained in $\mathbb{P}_{n,s}$. □

In the next proposition, we show that multiplying a positive spanning set by an orthonormal matrix does not affect the value of the cosine measure.

Proposition 6.30. *Let $\mathbb{P}_{n,s}$ be a positive spanning set of $\mathbb{R}^n$ with cosine measure equal to $\alpha > 0$. Let $M \in \mathbb{R}^{n\times n}$ be an orthonormal matrix. Then* $\operatorname{cm}(M\mathbb{P}_{n,s}) = \alpha$.

Proof. We follow Algorithm 6.2 to show that the cosine measure of $M\mathbb{P}_{n,s}$ is equal to α. By the properties of the rank of a matrix, a set $\mathbb{S}$ of n vectors contained in $\mathbb{P}_{n,s}$ is basis of $\mathbb{R}^n$ if and only if $M\mathbb{S}$ is a basis of $\mathbb{R}^n$ contained $M\mathbb{P}_{n,s}$. Let $\mathbb{B}_n$ be any basis of $\mathbb{R}^n$ contained in $\mathbb{P}_{n,s}$ and let $\widetilde{\mathbb{B}}_n = M\,\mathbb{B}_n$ be its corresponding basis of $\mathbb{R}^n$ contained $M\mathbb{P}_{n,s}$. In Step (1.1), we obtain

$$\begin{aligned}
\gamma_{\widetilde{\mathbb{B}}_n} &= \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\widetilde{\mathbb{B}}_n)^{-1}\mathbf{1}}} \\
&= \frac{1}{\sqrt{\mathbf{1}^\top (\mathbb{B}_n^\top M^\top M\,\mathbb{B}_n)^{-1}\mathbf{1}}} \\
&= \frac{1}{\sqrt{\mathbf{1}^\top (\mathbb{B}_n^\top\,\mathbb{B}_n)^{-1}\mathbf{1}}} = \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1}\mathbf{1}}} = \gamma_{\mathbb{B}_n}.
\end{aligned}$$

In Step (1.3), we obtain

$$\begin{aligned} p_{\widetilde{\mathbb{B}}_n} &= u_{\widetilde{\mathbb{B}}_n}^\top M\mathbb{P}_{n,s} \\ &= \left((M\,\mathbb{B}_n)^{-\top}\gamma_{\widetilde{\mathbb{B}}_n}\mathbf{1}\right)^\top M\mathbb{P}_{n,s} \\ &= \gamma_{\widetilde{\mathbb{B}}_n}\mathbf{1}^\top\,\mathbb{B}_n^{-1}\,M^\top M\mathbb{P}_{n,s} \\ &= \gamma_{\mathbb{B}_n}\mathbf{1}^\top\,\mathbb{B}_n^{-1}\,\mathbb{P}_{n,s} = u_{\mathbb{B}_n}^\top\mathbb{P}_{n,s} = p_{\mathbb{B}_n}. \end{aligned}$$

In Step (1.4), we obtain $p^{\max}_{\widetilde{\mathbb{B}}_n} = p^{\max}_{\mathbb{B}_n}$ since $p_{\widetilde{\mathbb{B}}_n} = p_{\mathbb{B}_n}$. Therefore,

$$\operatorname{cm}(M\mathbb{P}_{n,s}) = \min_{\widetilde{\mathbb{B}}_n \subset M\mathbb{P}_{n,s}} p^{\max}_{\widetilde{\mathbb{B}}_n} = \min_{\mathbb{B}_n \subset \mathbb{P}_{n,s}} p^{\max}_{\mathbb{B}_n} = \operatorname{cm}(\mathbb{P}_{n,s}).$$

□

Next, we briefly explore the complexity of Algorithm 1. The complexity of an algorithm is a count of the number of floating point operations (flops) required to complete the algorithm. As noted above, the maximum number of iterations required by the algorithm for a finite positive spanning set $\mathbb{P}_{n,s}$ is $\binom{s}{n}$; unless a maximal positive basis is inputted, in which case the required number of iterations is 2^n (Proposition 6.25). The complexity in big-oh notation per iteration is next.

Proposition 6.31. *Let $\mathbb{P}_{n,s}$ be a finite positive spanning set of $\mathbb{R}^n$. Then Algorithm 1 has a complexity of $O(n^3)$ flops per iteration (assuming basic matrix inversion techniques)*[3].

Proof. Computing the Gram matrix $\mathbf{G}(\mathbb{B}_n)$ requires $O(n^3)$ flops. Using basic methods, the matrix inversion of the Gram matrix uses $O(n^3)$ flops. The matrix multiplication in Step (1.1) is $O(2n^2)$ and the square roots and division are negligible. The matrix $\mathbb{B}_n$ is $n \times n$, so inversion is $O(n^3)$. Matrix multiplication in Step (1.3) is $O(n^2)$ and the maximum in Step (1.4) is negligible. All of the operations in Step 2. are negligible. So, the major effort is the construction of the Gram matrices and the matrix inversions, resulting in $O(n^3)$ flops per iteration. □

Note, the complexity above could be improved slightly if more advanced matrix inversion methods are used [GVL96]. However, the complexity of constructing the Gram matrix will remain $O(n^3)$, so little is gained by doing this.

Algorithm 1 can be shortened for minimal positive bases ($s = n+1$) and maximal positive bases ($s = 2n$).

[3]Proposition 6.31 corrects a typo in [HJB20, Proposition 20]

Theorem 6.32. *Let* $\mathbb{D}_n = \begin{bmatrix} d^1 & d^2 & \cdots & d^{n+1} \end{bmatrix}$ *be a minimal positive basis of* $\mathbb{R}^n$*. Then*

$$\gamma_{\mathbb{B}_n} = p^{\max}_{\mathbb{B}_n}$$

for all bases $\mathbb{B}_n \subset \mathbb{D}_n$ *where* $\gamma_{\mathbb{B}_n}$ *and* $p^{\max}_{\mathbb{B}_n}$ *are defined as in Algorithm 1. Moreover,*

$$\operatorname{cm}(\mathbb{D}_n) = \min_{\mathbb{B}_n \subset \mathbb{D}_n} \gamma_{\mathbb{B}_n}.$$

Proof. Let $\mathbb{B}_n$ be a basis of $\mathbb{R}^n$ contained in $\mathbb{D}_n$. Since $\gamma_{\mathbb{B}_n} > 0$, it follows that $d^\top u_{\mathbb{B}_n} < 0$ (by Proposition 6.12) where d is the only vector in $\mathbb{D}_n \setminus \mathbb{B}_n$. Therefore, $p^{\max}_{\mathbb{B}_n} = \gamma_{\mathbb{B}_n}$ for all bases $\mathbb{B}_n \subset \mathbb{D}_n$ and it follows that

$$\operatorname{cm}(\mathbb{D}_n) = \min_{\mathbb{B}_n \subset \mathbb{D}_n} \gamma_{\mathbb{B}_n}.$$

□

Theorem 6.33. *Let* $\mathbb{D}_{n,2n} = \begin{bmatrix} d^1 & d^2 & \cdots & d^{2n} \end{bmatrix}$ *be a maximal positive basis of* $\mathbb{R}^n$*. Then*

$$\gamma_{\mathbb{B}_n} = p^{\max}_{\mathbb{B}_n}$$

for all bases $\mathbb{B}_n \subset \mathbb{D}_{n,2n}$ *where* $\gamma_{\mathbb{B}_n}$ *and* $p^{\max}_{\mathbb{B}_n}$ *are defined as in Algorithm 1. Moreover,*

$$\operatorname{cm}(\mathbb{D}_{n,2n}) = \min_{\mathbb{B}_n \subset \mathbb{D}_{n,2n}} \gamma_{\mathbb{B}_n}.$$

Proof. Without loss of generality, by Theorem 6.8, let

$$\mathbb{D}_{n,2n} = \begin{bmatrix} d^1 & \cdots & d^n & -d^1 & \cdots & -d^n \end{bmatrix}$$

be a positive basis of unit vectors for $\mathbb{R}^n$. Note that every basis contained in $\mathbb{D}_{n,2n}$ has the form $\begin{bmatrix} \pm d^1 & \pm d^2 & \dots & \pm d^n \end{bmatrix}$. Hence, without loss of generality relabelling if necessary, let $\mathbb{B}_n = \begin{bmatrix} d^1 & \cdots & d^n \end{bmatrix}$. So

$$u_{\mathbb{B}_n}^\top d^1 = \cdots = u_{\mathbb{B}_n}^\top d^n = \gamma_{\mathbb{B}_n} > 0.$$

It follows that $u_{\mathbb{B}_n}^\top(-d^i) < 0$ for all $i \in \{1, \dots, n\}$. Therefore, $\gamma_{\mathbb{B}_n} = p^{\max}_{\mathbb{B}_n}$ for all bases $\mathbb{B}_n$ contained in $\mathbb{D}_{n,2n}$ and it follows that

$$\operatorname{cm}(\mathbb{D}_{n,2n}) = \min_{\mathbb{B}_n \subset \mathbb{D}_{n,2n}} \gamma_{\mathbb{B}_n}.$$

□

A consequence of the previous two theorems is that it is not necessary to compute $p_{\mathbb{B}_n}$, and $p^{\max}_{\mathbb{B}_n}$ in Algorithm 1. This means that Step (1.3) and Step

(1.4) can be deleted from Algorithm 1. The cosine measure (Step (2.1)) and the cosine vector set (Step (2.2)) can be found by simply setting

$$\mathrm{cm}(\mathbb{D}_{n,s}) = \min_{\mathbb{B}_n \subset \mathbb{D}_{n,s}} \gamma_{\mathbb{B}_n}$$

and

$$c\mathbf{V}(\mathbb{D}_{n,s}) = \{u_{\mathbb{B}_n} : \gamma_{\mathbb{B}_n} = \mathrm{cm}(\mathbb{D}_{n,s})\}$$

whenever $s = n+1$ or $s = 2n$. Unfortunately, this does not impact the complexity per iteration, as constructing the Gram matrices and the matrix inversions are still required.

The next example shows that the previous abridged algorithm does not guarantee to return the value of the cosine measure for intermediate positive bases ($n+1 < s < 2n$).

Example 6.34 (Algorithm 1 cannot be shortened for all intermediate positive bases)**.** Let

$$\mathbb{D}_{3,5} = \begin{bmatrix} 1 & 0 & 0 & -0.8 & 0 \\ 0 & 1 & 0 & 0 & -0.9 \\ 0 & 0 & 1 & -0.6 & -\sqrt{0.18} \end{bmatrix}.$$

Then $\mathbb{D}_{3,5}$ is an intermediate positive basis of $\mathbb{R}^3$. Computation shows that

$$\min_{\mathbb{B}_3 \subset \mathbb{D}_{3,5}} \gamma_{\mathbb{B}_3} \approx 0.2038.$$

and the unit vector associated to the minimal $\gamma_{\mathbb{B}_3}$ is

$$u_{\mathbb{B}_3} \approx \begin{bmatrix} 0.4115 & 0.2038 & -0.8883 \end{bmatrix}^\top$$

where

$$\mathbb{B}_3 = \begin{bmatrix} 0 & -0.8 & 0 \\ 1 & 0 & -0.9 \\ 0 & -0.6 & -\sqrt{0.19} \end{bmatrix}.$$

Computing $p_{\mathbb{B}_3}$ and $p_{\mathbb{B}_3}^{\max}$, yields

$$p_{\mathbb{B}_3} \approx \begin{bmatrix} 0.4115 & 0.2038 & -0.8883 & 0.2038 & 0.2038 \end{bmatrix}$$

and so

$$p_{\mathbb{B}_3}^{\max} \approx 0.4115 \neq \gamma_{\mathbb{B}_3}.$$

Note that the cosine measure of $\mathbb{D}_{3,5}$ is found when considering

$$\mathbb{B}_3^* = \begin{bmatrix} 1 & 0 & -0.8 \\ 0 & 1 & 0 \\ 0 & 0 & -0.6 \end{bmatrix}.$$

Thus, $\gamma_{\mathbb{B}_3^*} \approx 0.3015$ and $u_{\mathbb{B}_3^*} \approx \begin{bmatrix} 0.3015 & 0.3015 & -0.9045 \end{bmatrix}^\top$. The dot product vector is $p_{\mathbb{B}_3^*} \approx \begin{bmatrix} 0.3015 & 0.3015 & -0.9045 & 0.3015 & 0.1229 \end{bmatrix}$ and so

$$\mathrm{cm}(\mathbb{D}_{3,5}) = p_{\mathbb{B}_3^*}^{\max} \neq \min_{\mathbb{B}_3 \subset \mathbb{D}_{3,5}} \gamma_{\mathbb{B}_3}.$$

We conclude this section by presenting an interesting fact about optimal positive bases of minimal size. It is shown that the cosine vector set of an optimal minimal positive basis is an optimal minimal positive basis.

Proposition 6.35. *Let $\mathring{\mathbb{D}}_n$ be an optimal minimal positive basis of $\mathbb{R}^n$. Then $c\mathbf{V}(\mathring{\mathbb{D}}_n)$ is an optimal minimal positive basis of $\mathbb{R}^n$.*

Proof. We know that there are $n+1$ bases of $\mathbb{R}^n$ contained in a minimal positive basis $\mathring{\mathbb{D}}_n = \begin{bmatrix} d^1 & \cdots & d^{n+1} \end{bmatrix}$ of $\mathbb{R}^n$ (Proposition 6.24). From [Næv18, Theorem 1], since $\mathring{\mathbb{D}}_n$ is optimal, we know that $(d^i)^\top d^j = \frac{-1}{n}$ for all $i, j \in \{1, \ldots, n+1\}, i \neq j$. Let $\mathbb{B}_n^i$ be a basis of $\mathbb{R}^n$ contained in $\mathring{\mathbb{D}}_n, i \in \{1, \ldots, n+1\}$ and let $u^i \in \mathbb{R}^n$ be the vector associated to the basis $\mathbb{B}_n^i$ as defined in Step (1.2) of Algorithm 1. Then

$$\gamma_{\mathbb{B}_n^i} = \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n^i)^{-1}\mathbf{1}}} = \mathrm{cm}\,(\mathring{\mathbb{D}}_n) \quad \text{for all} \quad i \in \{1, \ldots, n+1\}.$$

Hence, u^i is in $c\mathbf{V}(\mathring{\mathbb{D}}_n)$ for all $i \in \{1, \ldots, n+1\}$ and so there are exactly n+1 vectors in $c\mathbf{V}(\mathring{\mathbb{D}}_n)$. To finish the proof, we show that

$$(u^i)^\top (u^i) = 1 \quad \text{and} \quad (u^i)^\top u^j = \frac{-1}{n} \quad \text{for all} \quad i, j \in \{1, \ldots, n+1\}, i \neq j.$$

First, we have

$$\begin{aligned} (u^i)^\top u^i &= \left(\frac{1}{n}\right) \mathbf{1}^\top (\mathbb{B}_n^i)^{-1} (\mathbb{B}_n^i)^{-\top} \mathbf{1} \left(\frac{1}{n}\right) \\ &= \left(\frac{1}{n^2}\right) \mathbf{1}^\top \mathbf{G}(\mathbb{B}_n^i)^{-1} \mathbf{1} \\ &= \left(\frac{1}{n^2}\right) n^2 = 1. \end{aligned}$$

Second, Let $\mathbb{B}_n^i$ and $\mathbb{B}_n^j$ be two distinct bases of $\mathbb{R}^n$ contained in $\mathring{\mathbb{D}}_n$ and let $u^i, u^j \in \mathbb{R}^n$ be the unit vectors associated to $\mathbb{B}_n^i$ and $\mathbb{B}_n^j$, respectively. We have

$$\begin{aligned}(u^i)^\top u^j &= \left(\frac{1}{n}\right)\mathbf{1}^\top(\mathbb{B}_n^i)^{-1}(\mathbb{B}_n^j)^{-\top}\mathbf{1}\left(\frac{1}{n}\right)\\ &= \left(\frac{1}{n^2}\right)\mathbf{1}^\top\left((\mathbb{B}_n^j)^\top\mathbb{B}_n^i\right)^{-1}\mathbf{1}\end{aligned}$$

Without loss of generality, we may write $\mathbb{B}_n^i$ and $\mathbb{B}_n^j$ as block matrices in the following way:

$$\mathbb{B}_n^i = \begin{bmatrix}D & d^n\end{bmatrix} \quad \text{and} \quad \mathbb{B}_n^j = \begin{bmatrix}D & d^{n+1}\end{bmatrix}$$

where $D = \begin{bmatrix}d^1 & \cdots & d^{n-1}\end{bmatrix} \in \mathbb{R}^{n\times n-1}$. Hence,

$$\begin{aligned}\left((\mathbb{B}_n^j)^\top\mathbb{B}_n^i\right)^{-1} &= \begin{bmatrix}\mathbf{G}(D) & D^\top d^n\\ (d^{n+1})^\top D & (d^n)^\top d^{n+1}\end{bmatrix}^{-1}\\ &= \begin{bmatrix}\mathbf{G}(D) & \left(-\frac{1}{n}\right)\mathbf{1}_{n-1}\\ \left(-\frac{1}{n}\right)\mathbf{1}_{n-1}^\top & -\frac{1}{n}\end{bmatrix}^{-1}.\end{aligned}$$

Letting $v = \left(-\frac{1}{n}\right)\mathbf{1}_{n-1} \in \mathbb{R}^{n-1}, G = \mathbf{G}(D) \in \mathbb{R}^{n-1\times n-1}, c = -\frac{1}{n}$ and using Lemma 6.23, we obtain

$$\begin{aligned}&\left((\mathbb{B}_n^j)^\top\mathbb{B}_n^i\right)^{-1}\\ &= \begin{bmatrix}G & v\\ v^\top & c\end{bmatrix}^{-1}\\ &= \begin{bmatrix}G^{-1}+G^{-1}v(c-v^\top G^{-1}v)^{-1}v^\top G^{-1} & -G^{-1}v(c-v^\top G^{-1}v)^{-1}\\ -(c-v^\top G^{-1}v)^{-1}v^\top G^{-1} & (c-v^\top G^{-1}v)^{-1}\end{bmatrix}.\end{aligned}$$

The sum of the entries in $\left((\mathbb{B}_n^j)^\top\mathbb{B}_n^i\right)^{-1}$ is given by

$$\begin{aligned}&\mathbf{1}^\top\left((\mathbb{B}_n^j)^\top\mathbb{B}_n^i\right)^{-1}\mathbf{1}\\ &= \mathbf{1}_{n-1}^\top G^{-1}\mathbf{1}_{n-1} + (c-v^\top G^{-1}v)^{-1}\mathbf{1}_{n-1}^\top G^{-1}vv^\top G^{-1}\mathbf{1}_{n-1}\\ &\quad - 2(c-v^\top G^{-1}v)^{-1}\mathbf{1}_{n-1}^\top G^{-1}v + (c-v^\top G^{-1}v)^{-1}\\ &= \mathbf{1}_{n-1}^\top G^{-1}\mathbf{1}_{n-1} + (c-v^\top G^{-1}v)^{-1}\left(\mathbf{1}_{n-1}^\top G^{-1}vv^\top G^{-1}\mathbf{1}_{n-1} - (2)\mathbf{1}_{n-1}^\top G^{-1}v + 1\right)\end{aligned} \tag{6.2}$$

Let us investigate some of the terms in (6.2) separately. Using Proposition 3.3, we find

$$\mathbf{1}_{n-1}^{\top} G^{-1} \mathbf{1}_{n-1} = \frac{n(n-1)}{2}.$$

It follows that

$$\begin{aligned}(c - v^{\top} G^{-1} v)^{-1} &= \left(-\frac{1}{n} - \frac{1}{n^2} \mathbf{1}_{n-1}^{\top} G^{-1} \mathbf{1}_{n-1} \right)^{-1} \\ &= \left(-\frac{1}{n} - \frac{1}{n^2} \frac{n(n-1)}{2} \right)^{-1} \\ &= -\frac{2n}{n+1}.\end{aligned}$$

Also,

$$\begin{aligned}\mathbf{1}_{n-1}^{\top} G^{-1} v v^{\top} G^{-1} \mathbf{1}_{n-1} &= \left(-\frac{1}{n} \right)^2 \mathbf{1}_{n-1}^{\top} G^{-1} \mathbf{1}_{n-1} \mathbf{1}_{n-1}^{\top} G^{-1} \mathbf{1}_{n-1} \\ &= \left(-\frac{1}{n} \right)^2 \left(\frac{n(n-1)}{2} \right)^2 \\ &= \frac{(n-1)^2}{4},\end{aligned}$$

and

$$(2) \mathbf{1}_{n-1}^{\top} G^{-1} v = -2 \left(\frac{1}{n} \right) \mathbf{1}_{n-1}^{\top} G^{-1} \mathbf{1}_{n-1} = -2 \left(\frac{1}{n} \right) \frac{n(n-1)}{2} = 1 - n$$

All together, Equation (6.2) is now

$$\begin{aligned}\mathbf{1}^{\top} \left((\mathbb{B}_n^j)^{\top} \mathbb{B}_n^i \right)^{-1} \mathbf{1} &= \frac{n(n-1)}{2} - \frac{2n}{n+1} \left(\frac{(n-1)^2}{4} + n - 1 + 1 \right) \\ &= -n.\end{aligned}$$

Finally, we obtain that the dot product

$$(u^i)^{\top} u^j = \left(\frac{1}{n^2} \right) \mathbf{1}^{\top} \left((\mathbb{B}_n^j)^{\top} \mathbb{B}_n^i \right)^{-1} \mathbf{1} = \left(\frac{1}{n^2} \right) (-n) = -\frac{1}{n}. \qquad \square$$

In the next section, the problem of finding intermediate positive bases with maximal cosine measure is explored. While exploring this topic, we discover that Algorithm 1 may be simplified when the positive basis has a specific structure.

6.3 Structures of positive bases

In [Næv18], the structure of optimal positive bases of minimal size and maximal size are described. However, optimal positive bases of intermediate sizes are not provided. In this section, we investigate optimality for intermediate positive bases in depth. We define two subsets of positive bases with nice properties. We investigate the problem of finding an optimal positive basis on these two subsets. We develop properties of this type of intermediate positive bases and show that the algorithm to compute the cosine measure introduced in the previous section can be simplified in the presence of such positive bases.

We begin by some additional background information. In 1987, Romanowicz introduced the concepts of a basis of a subspace L in $\mathbb{R}^n$, a positive basis of a subspace L in $\mathbb{R}^n$, and critical vectors [Rom87]. Romanowicz used these concepts to characterize the structure of positive bases. As such, they will be helpful to characterize the structure of any non-minimal positive basis of $\mathbb{R}^n$.

Definition 6.36 (Basis of a subspace and positive basis of a subspace)**.** A subset $P_1 \in \mathbb{R}^{n\times m}$, $1 \leq m \leq n$ of a basis $\mathbb{B}_n$ of $\mathbb{R}^n$ is called a *basis of a subspace* L_1 *in* $\mathbb{R}^n$ if $\text{span}(P_1) = L_1$.
A subset $P_2 \in \mathbb{R}^{n\times r}, 2 \leq r \leq s$, of a positive basis $\mathbb{D}_{n,s}$ is called a *positive basis of a subspace* L_2 *in* $\mathbb{R}^n$ if $\text{pspan}(P_2) = L_2$.

Denote by $\dim(L)$ the dimension of a subspace L in $\mathbb{R}^n$. A positive basis of a subspace L in $\mathbb{R}^n$ $(1 \leq \dim(L) \leq n)$ will be denoted by $\mathbb{D}^n_{m,r}$ where $m = \dim(L)$, and r is the size of the positive basis of L in $\mathbb{R}^n$. When $m = n$ (that is the case when the subspace is $\mathbb{R}^n$ itself), we may omit the superscript and simply write $\mathbb{D}_{n,s}$ to regain the notation from Sections 6.1 and 6.2. When the positive basis of a subspace in $\mathbb{R}^n$ is minimal size, we may omit the size r in the subscript, and simply write $\mathbb{D}^n_m$.

Example 6.37. Let $\mathbb{D}_{3,5} = \begin{bmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 1 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$. It can be proved that $\mathbb{D}_{3,5}$ is a positive basis of $\mathbb{R}^3$. Also, we have that

$$\mathbb{D}^3_2 = \begin{bmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \end{bmatrix}$$

is a positive basis of $L_1 = \{x \in \mathbb{R}^3 : x_3 = 0\}$ in $\mathbb{R}^3$ and

$$\mathbb{D}_1^3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{bmatrix}$$

is a positive basis of $L_2 = \{x \in \mathbb{R}^3 : x_1 = x_2 = 0\}$ in $\mathbb{R}^3$.

In Definition 6.14, the meaning of an optimal positive basis of $\mathbb{R}^n$ is provided. We will also use the word optimal to describe a minimal positive basis of a proper subspace L in $\mathbb{R}^n$. In the remainder of this chapter, we say that a minimal positive basis $\mathbb{D}_m^n$ of a proper subspace L in $\mathbb{R}^n$ is *optimal* if the cosine measure of $\mathbb{D}_m^n$ restricted to the subspace L

$$\mathrm{cm}_L(\mathbb{D}_m^n) := \min_{\substack{\|u\|=1 \\ u \in L}} \max_{d \in \mathbb{D}_m^n} \frac{u^\top d}{\|d\|} > \mathrm{cm}_L(\widetilde{\mathbb{D}}_m^n)$$

for any minimal positive basis $\widetilde{\mathbb{D}}_m^n$ of the subspace L in $\mathbb{R}^n$. For example, $\mathbb{D}_2^3$ in Example 6.37 is not an optimal minimal positive basis of L_1 in $\mathbb{R}^3$ since

$$\begin{bmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

is not an optimal minimal positive basis of $\mathbb{R}^2$, and $\mathbb{D}_1^3$ is an optimal minimal positive basis of L_2 in $\mathbb{R}^3$ since all possible positive bases $\widehat{\mathbb{D}}_1^3$ of L_2 in $\mathbb{R}^3$ have $\mathrm{cm}_L(\widehat{\mathbb{D}}_1^3) = 1$.

Definition 6.38 (Critical set, critical vectors and complete critical set)**.** Let $\mathbb{D}_{n,s}$ be a positive basis of $\mathbb{R}^n$. Let C be a subset of $\mathbb{R}^n$. We say C is a *critical set of* $\mathbb{D}_{n,s}$ if

$$\mathrm{pspan}\left((\mathbb{D}_{n,s} \setminus \{d\}) \cup C\right) \neq \mathbb{R}^n \tag{6.3}$$

for each $d \in \mathbb{D}_{n,s}$. Elements of C are called *critical vectors*. The *complete critical set* is denoted by $C(\mathbb{D}_{n,s})$ and contains all critical set C satisfying (6.3).

Note that $\mathbf{0} \in \mathbb{R}^n$ is a critical vector for all positive bases in $\mathbb{R}^n$. The following example provides the critical set of a minimal positive basis in $\mathbb{R}^2$. It is proved in [Rom87] that for a minimal positive basis $\mathbb{D}_n = \begin{bmatrix} d^1 & d^2 & \cdots & d^{n+1} \end{bmatrix}$ in $\mathbb{R}^n, n \geq 2$, we have

$$C(\mathbb{D}_n) = -\bigcup_{i \neq j} \mathrm{pspan}\left(\mathbb{D}_{n,s} \setminus \{d^i, d^j\}\right). \tag{6.4}$$

Also, it is proved in [Rom87] that a maximal positive basis $\mathbb{D}_{n,2n}, n \geq 1$, has the property

$$C(\mathbb{D}_{n,2n}) = \{\mathbf{0}\}. \tag{6.5}$$

Example 6.39. Consider the (optimal) minimal positive basis

$$\mathring{\mathbb{D}}_2 = \begin{bmatrix} d^1 & d^2 & d^3 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}.$$

It follows from Equation (6.4) that

$$C(\mathring{\mathbb{D}}_2) = -\operatorname{pspan}(d^1) \cup -\operatorname{pspan}(d^2) \cup -\operatorname{pspan}(d^3).$$

Theorem 6.40 (Structure of a non-minimal positive basis [Rom87, Theorem 1])**.** *Let $n \geq 2$ and $s \geq n+2$. The set of s vectors in $\mathbb{R}^n$, $D_{n,s}$, is a positive basis of $\mathbb{R}^n$ if and only if $D_{n,s}$ admits the partition*

$$D_{n,s} = \mathbb{D}^n_{m_1} \cup (\mathbb{D}^n_{m_2} \oplus c^1) \cup \cdots \cup (\mathbb{D}^n_{m_q} \oplus c^{q-1}) \tag{6.6}$$

where $\mathbb{D}^n_{m_1}, \ldots, \mathbb{D}^n_{m_q}$ are minimal positive bases of subspaces $L_1, \ldots, L_q$ in $\mathbb{R}^n$, $\mathbb{R}^n = L_1 \oplus L_2 \oplus \cdots \oplus L_q, L_i \cap L_j = \{\mathbf{0}\}$ for $i \neq j$, $1 \leq \dim L_i \leq n-1$, and $c^j \in \mathbb{R}^n$ ($j \in \{1, \ldots, q-1\}$) is a critical vector of the positive basis $\mathbb{D}^n_{M_j, M_j+j}$ of the subspace $L_1 \oplus \cdots \oplus L_j$ in $\mathbb{R}^n$, where

$$\mathbb{D}^n_{M_1, M_1+1} = \mathbb{D}^n_{m_1}$$

and

$$\mathbb{D}^n_{M_j} = \mathbb{D}^n_{m_1} \cup (\mathbb{D}^n_{m_2} \oplus c^1) \cup \cdots \cup (\mathbb{D}^n_{m_j} \oplus c^{j-1})$$

for all $j \in \{2, \ldots, q\}, q \geq 2$.

Let us provide an example to clarify the meaning of the previous theorem.

Example 6.41. The set

$$D_{3,5} = \begin{bmatrix} d^1 & d^2 & d^3 & d^4 & d^5 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

is a positive basis of $\mathbb{R}^3$, since $\mathbb{D}^3_2 = \begin{bmatrix} d^1 & d^2 & d^3 \end{bmatrix}$ is a minimal positive basis of the subspace $L_2 = \left\{x \in \mathbb{R}^3 : x_3 = 0\right\}$ in $\mathbb{R}^3$, the set $\mathbb{D}^3_1 = \begin{bmatrix} d^4 & d^5 \end{bmatrix}$ is a minimal positive basis of the subspace $L_1 = \{x \in \mathbb{R}^3 : x_1 = x_2 = 0\}$ in $\mathbb{R}^3$,

$\mathbb{R}^3 = L_2 \oplus L_1, L_2 \cap L_1 = \{\mathbf{0}\}$ and $\mathbf{0}$ is a critical vector for all positive bases. Hence, $D_{3,5}$ admits the partition

$$D_{3,5} = \mathbb{D}_2^3 \cup \mathbb{D}_1^3.$$

The set

$$\begin{aligned}\widetilde{D}_{3,5} &= \begin{bmatrix} d^1 & d^2 & d^3 & \tilde{d}^4 & \tilde{d}^5 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & -1 & -1 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}\end{aligned}$$

is a positive basis of $\mathbb{R}^3$. Notice $c = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^\top$ is a critical vector of the positive basis $\mathring{\mathbb{D}}_2^3$ of $\mathbb{R}^2$ in $\mathbb{R}^3$. Hence, $\widetilde{D}_{3,5}$ can be written as the partition

$$\widetilde{D}_{3,5} = \mathbb{D}_2^3 \cup \left(\mathbb{D}_1^3 \oplus c\right).$$

The following lemma shows that the number of positive bases of subspaces in a partition of a positive basis of $\mathbb{R}^n$ given in (6.6) is $q = s - n$.

Lemma 6.42. *Let $\mathbb{D}_{n,s}$ be a non-minimal positive basis in $\mathbb{R}^n$. Then the number of positive bases $\mathbb{D}_{m_j}$ of a subspace L_j in $\mathbb{R}^n$ in the partition* (6.6) *is*

$$q = s - n.$$

Proof. We have

$$\sum_{j=1}^{q}(m_j + 1) = s,$$

$$\sum_{j=1}^{q} m_j = n,$$

where $1 \leq m_j \leq n-1, n+1 < s \leq 2n$. Hence,

$$s - n = \left(\sum_{j=1}^{q}(m_j + 1)\right) - n = q + \left(\sum_{j=1}^{q} m_j\right) - n = q. \qquad \square$$

In general, given a set of vectors, it is difficult to verify if it can be written in the form 6.6. This is due to the presence of critical vectors, but also because the subspaces appearing in the partition are not necessarily orthogonal to one another. For these reasons, we define two subsets of positive bases with nice properties: *critical-free positive bases (CFPB)* and *critical-free orthogonal positive bases (CFOPB)*.

Definition 6.43 (Critical-free positive basis)**.** The positive basis $\mathbb{D}_{n,s}$ of $\mathbb{R}^n$ is in the set $\Omega_{n,s}$ and is said to be a *critical-free positive basis* if $\mathbb{D}_{n,s}$ admits the partition

$$\mathbb{D}_{n,s} = \mathbb{D}^n_{m_1} \cup \mathbb{D}^n_{m_2} \cup \cdots \cup \mathbb{D}^n_{m_{s-n}} \tag{6.7}$$

where $\mathbb{D}^n_{m_1}, \ldots, \mathbb{D}^n_{m_{s-n}}$ are minimal positive positive bases of the subspaces $L_1, \ldots, L_{s-n}$ in $\mathbb{R}^n$ (respectively), $\mathbb{R}^n = L_1 \oplus L_2 \oplus \cdots \oplus L_{s-n}$, $1 \leq \dim L_i = m_i \leq n$, for all $i \in \{1, \ldots, s-n\}$ and such that $L_i \cap L_j = \{\mathbf{0}\}$ for $i \neq j$ whenever $s - n \geq 2$.

Definition 6.44 (Critical-free orthogonal positive basis)**.** Let $\mathbb{D}_{n,s}$ be a positive basis in $\mathbb{R}^n$. We say that $\mathbb{D}_{n,s}$ is in $\Omega^+_{n,s}$ and is said to be a *critical-free orthogonal positive basis (CFOPB)* of $\mathbb{R}^n$ if it is in $\Omega_{n,s}$ and all positive bases $\mathbb{D}^n_{m_i}$ of the subspaces L_i in $\mathbb{R}^n$ in a partition of $\mathbb{D}_{n,s}$ are pairwise orthogonal whenever $i \geq 2$. That is $(\mathbb{D}^n_{m_i})^\top \mathbb{D}^n_{m_j} = \mathbf{0} \in \mathbb{R}^{m_i+1 \times m_j+1}$ for all $i \neq j, i, j \in \{1, \ldots, , s-n\}$ whenever $s - n \geq 2$.

Notice that, for positive bases of intermediate sizes, we have

$$\Omega^+_{n,s} \subset \Omega_{n,s} \subset \mathcal{P}_{n,s}$$

For minimal positive bases, we have

$$\Omega^+_{n,n+1} = \Omega_{n,n+1} = \mathcal{P}_{n,n+1}.$$

For maximal positive bases, we have

$$\Omega^+_{n,2n} \subset \Omega_{n,2n} = \mathcal{P}_{n,2n}.$$

We begin by investigating CFPB. The next corollary follows directly from Theorem 6.40. It describes the structure of a basis $\mathbb{B}_n$ in $\mathbb{R}^n$ contained in a non-minimal CFPB.

Corollary 6.45. *Let $\mathbb{D}_{n,s}$ be a CFPB of $\mathbb{R}^n$. Let $\mathbb{B}_n$ be any basis of $\mathbb{R}^n$ contained in $\mathbb{D}_{n,s}$. Then $\mathbb{B}_n$ admits the partition*

$$\mathbb{B}_n = \mathbb{B}^n_{m_1} \cup \mathbb{B}^n_{m_2} \cup \cdots \cup \mathbb{B}^n_{m_{s-n}}$$

where $\mathbb{B}^n_{m_i} \in \mathbb{R}^{n \times m_i}$ is a basis of the subspace L_i in $\mathbb{R}^n$ for all $i \in \{1, \ldots, s-n\}$, $\mathbb{R}^n = L_1 \oplus \cdots \oplus L_{s-n}$, $1 \leq \dim L_i = m_i \leq n$, $L_i \cap L_j = \{\mathbf{0}\}$ for $i \neq j$ (whenever $s - n \geq 2$) and such that

$$\mathbb{B}^n_{m_i} \subset \mathbb{D}^n_{m_i},$$

for all $i \in \{1, \ldots, s-n\}$.

In the previous corollary, note that a positive basis $\mathbb{D}^n_{m_i}$ of a subspace L_i in $\mathbb{R}^n$ contains m_i+1 bases of L_i (this follows from Proposition 6.9). Forming a set by picking one basis from each subspace L_i of $\mathbb{R}^n$ always forms a basis of $\mathbb{R}^n$. Hence, the number of bases of $\mathbb{R}^n$ contained in a positive basis in $\Omega_{n,s}$ is

$$\prod_{i=1}^{s-n}(m_i+1). \tag{6.8}$$

For example, we obtain that there are $n+1$ bases of $\mathbb{R}^n$ in a minimal positive basis and 2^n bases of $\mathbb{R}^n$ in a maximal positive basis. This agrees with Propositions 6.24 and 6.25 in the previous section.

Next it is shown that Algorithm 1 can be simplified when $\mathbb{D}_{n,s}$ is a CFPB of $\mathbb{R}^n$.

Proposition 6.46 (Property of a CFPB)**.** *Let $\mathbb{D}_{n,s}$ be a positive basis of $\mathbb{R}^n$ in $\Omega_{n,s}$. Let $\mathbb{B}_n$ be a basis of $\mathbb{R}^n$ contained in $\mathbb{D}_{n,s}$. Let $u_{\mathbb{B}_n}$ be the unit vector such that $u_{\mathbb{B}_n}^\top \mathbb{B}_n = \gamma_{\mathbb{B}_n}\mathbf{1}^\top$ (where $\gamma_{\mathbb{B}_n}$ is defined as in Algorithm 1 Step (1.1)). Then*

$$u_{\mathbb{B}_n}^\top d \le 0$$

for all vectors $d \in \mathbb{D}_{n,s}\setminus\{\mathbb{B}_n\}$. Consequently, Steps (1.3),(1.4) in Algorithm 1 can be omitted and Steps (2.1),(2.2) become (respectively)

$$\mathrm{cm}(\mathbb{D}_{n,s}) = \min_{\mathbb{B}_n\subset\mathbb{D}_{n,s}} \gamma_{\mathbb{B}_n}, \tag{6.9}$$

$$c\mathbf{V}(\mathbb{D}_{n,s}) = \{u_{\mathbb{B}_n} : \gamma_{\mathbb{B}_n} = \mathrm{cm}(\mathbb{D}_{n,s})\}. \tag{6.10}$$

Proof. Let $\mathbb{B}_n$ be any basis of $\mathbb{R}^n$ contained in $\mathbb{D}_{n,s}$. By Corollary 6.45, $\mathbb{B}_n$ can be written as $\mathbb{B}_n = \mathbb{B}^n_{m_1}\cup\cdots\cup\mathbb{B}^n_{m_{s-n}}$ where $\mathbb{B}^n_{m_i}$ is contained in the positive basis $\mathbb{D}^n_{m_i}$ of the subspace L_i in $\mathbb{R}^n$. Let $u_{\mathbb{B}_n}$ be the unit vector defined in Step (1.2) of Algorithm 1. Consider the minimal positive basis $\mathbb{D}^n_{m_i}$ of a subspace L_i for some $i\in\{1,\ldots,s-n\}$. We know that there is only one vector in the set $\mathbb{D}^n_{m_i}\setminus\{\mathbb{B}^n_{m_i}\}$. Denote this vector by d. We know that the projection of $u_{\mathbb{B}_n}$ onto L_i, denoted by $\mathrm{proj}_{L_i}u_{\mathbb{B}_n}$ has dot product

$$\left(\mathrm{proj}_{L_i}u_{\mathbb{B}_n}\right)^\top d < 0$$

whenever $\mathrm{proj}_{L_i}u_{\mathbb{B}_n}\neq\mathbf{0}$ by Proposition 6.12, and it is equal to zero when $\mathrm{proj}_{L_i}u_{\mathbb{B}_n}=\mathbf{0}$. We obtain

$$u_{\mathbb{B}_n}^\top d = \left(\mathrm{proj}_{L_i}u_{\mathbb{B}_n}\right)^\top d \le 0.$$

Therefore, $u_{\mathbb{B}_n}^\top d \leq 0$ for all $d \in \mathbb{D}_{n,s} \setminus \{\mathbb{B}_n\}$.

As mentioned in Lemma 6.18, we know that $\gamma_{\mathbb{B}_n} > 0$ for any basis $\mathbb{B}_n$ contained in $\mathbb{D}_{n,s}$. It follows that $\gamma_{\mathbb{B}_n}$ is the maximum value of the dot product between $u_{\mathbb{B}_n}$ and the vectors in $\mathbb{D}_{n,s}$. Therefore, we may delete Steps (1.3), (1.4) and simply set $\mathrm{cm}(\mathbb{D}_{n,s})$ and $c\mathbf{V}(\mathbb{D}_{n,s})$ as in (6.9) and (6.10), respectively. □

For all CFPB of $\mathbb{R}^n$, we may use the following simplified algorithm.

Algorithm 2: The cosine measure of a critical-free positive basis

Given $\mathbb{D}_{n,s}$, a CFPB of $\mathbb{R}^n$,

1. For all bases $\mathbb{B}_n \subset \mathbb{D}_{n,s}$, compute

(1.1) $\gamma_{\mathbb{B}_n} = \dfrac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1}\mathbf{1}}}$,

(1.2) $u_{\mathbb{B}_n} = \mathbb{B}_n^{-\top} \gamma_{\mathbb{B}_n} \mathbf{1}$ (unit vector associated to $\gamma_{\mathbb{B}_n}$).

2. Return

(2.1) $\mathrm{cm}(\mathbb{D}_{n,s}) = \min\limits_{\mathbb{B}_n \subset \mathbb{D}_{n,s}} \gamma_{\mathbb{B}_n}$ (cosine measure of $\mathbb{D}_{n,s}$),

(2.2) $c\mathbf{V}(\mathbb{D}_{n,s}) = \{u_{\mathbb{B}_n} : \gamma_{\mathbb{B}_n} = \mathrm{cm}(\mathbb{D}_{n,s})\}$ (cosine vector set of $\mathbb{D}_{n,s}$).

Before exploring CFOPB, we examine the structure of intermediate positive bases in $\mathbb{R}^3$.

6.4 An optimal CFPB in $\mathbb{R}^3$

In this section, we find the structure of an optimal positive basis over $\Omega_{3,5}$. We show that an optimal CFPB is an CFOPB.

In $\mathbb{R}^3$, there is only one possible intermediate size:5. Let $\mathbb{D}_{3,5}$ be a CFPB. Then $\mathbb{D}_{3,5}$ admits the partition

$$\mathbb{D}_{3,5} = \mathbb{D}_1^3 \cup \mathbb{D}_2^3$$

where $\mathbb{D}_1^3$ is a minimal positive basis of a subspace L_1 in $\mathbb{R}^3$ and $\mathbb{D}_2^3$ is a positive basis of a subspace L_2 in $\mathbb{R}^3$ such that $L_1 \oplus L_2 = \mathbb{R}^3$, $L_1 \cap L_2 = \{\mathbf{0}\}$. Note that this is the only possible partition that includes at least two positive bases of a subspace in $\mathbb{R}^n$. Also, $\mathbb{D}_{3,5}$ cannot contain more than two subpositive bases. Therefore, this is the only possible partition for a CFPB in

$\mathbb{R}^3$ of size 5. We can realign the positive basis $\mathbb{D}_{3,5}$ so that $\mathbb{D}_1^3$ is equal to

$$\mathbb{D}_1^3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{bmatrix}.$$

This realignnment can be obtained using rotation matrices. Since rotation matrices are orthonormal matrices, it does not affect the value of the cosine measure of $\mathbb{D}_{3,5}$ (Proposition 6.30).

It follows that any basis $\mathbb{B}_3$ of $\mathbb{R}^3$ contained in $\mathbb{D}_{3,5}$ admits the partition

$$\mathbb{B}_3 = \mathbb{B}_1^3 \cup \mathbb{B}_2^3$$

where $\mathbb{B}_1^3 = \pm \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ and $\mathbb{B}_2^3$ is a basis of L_2 in $\mathbb{R}^3$ contained in $\mathbb{D}_2^3$. We next show that the two subspaces L_1 and L_2 of $\mathbb{R}^3$ must be orthogonal for $\mathbb{D}_{3,5}$ to be optimal over $\Omega_{3,5}$. First, a lemma is introduced. It will be used in the main proposition of this sub-section.

Lemma 6.47. *Let $\mathbb{D}_{3,5}$ be a CFPB of $\mathbb{R}^3$. Let $\mathbb{B}_3 = \begin{bmatrix} \mathbb{B}_2^3 & \mathbb{B}_1^3 \end{bmatrix}$ be a basis of $\mathbb{R}^3$ contained in the all activity set $\overline{\mathbf{A}}(\mathbb{D}_{3,5})$. Let $v = (\mathbb{B}_2^3)^\top \mathbb{B}_1^3 \in \mathbb{R}^2$. Then*

$$\mathbf{1}^\top \mathbf{G}(\mathbb{B}_3)^{-1}\mathbf{1} = \mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1} + c\left(\mathbf{1}^\top \begin{bmatrix} -\mathbf{G}(\mathbb{B}_2^3)^{-1}v \\ 1 \end{bmatrix}\right)^2$$

where $c = \left(1 - v^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}v\right)^{-1}$. Moreover,

$$\mathbf{1}^\top \mathbf{G}(\mathbb{B}_3)\mathbf{1} \geq \mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1} + 1 \tag{6.11}$$

with equality if and only $v = \mathbf{0}$.

Proof. To make notation tighter, let $\mathbf{G}(\mathbb{B}_3) = \mathbf{G}_3$ and $\mathbf{G}(\mathbb{B}_2^3) = \mathbf{G}_2$. By Lemma 6.23, the inverse of $\mathbf{G}_3$ is

$$\begin{aligned}
\mathbf{G}_3^{-1} &= \begin{bmatrix} \mathbf{G}_2^{-1} + \mathbf{G}_2^{-1}v\left(1 - v^\top \mathbf{G}_2^{-1}v\right)^{-1} v^\top \mathbf{G}_2^{-1} & -\mathbf{G}_2^{-1}v\left(1 - v^\top \mathbf{G}_2^{-1}v\right)^{-1} \\ -\left(1 - v^\top \mathbf{G}_2^{-1}v\right)^{-1} v^\top \mathbf{G}_2^{-1} & \left(1 - v^\top \mathbf{G}_2^{-1}v\right)^{-1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{G}_2^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + c\begin{bmatrix} \mathbf{G}_2^{-1}vv^\top \mathbf{G}_2^{-1} & -\mathbf{G}_2^{-1}v \\ (-\mathbf{G}_2^{-1}v)^\top & 1 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{G}_2^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + c\begin{bmatrix} -\mathbf{G}_2^{-1}v \\ 1 \end{bmatrix}\begin{bmatrix} (-\mathbf{G}_2^{-1}v)^\top & 1 \end{bmatrix}.
\end{aligned}$$

It follows that the sum of all entries in $\mathbf{G}_3^{-1}$ is

$$\mathbf{1}^\top \mathbf{G}_3^{-1}\mathbf{1} = \mathbf{1}^\top \mathbf{G}_2^{-1}\mathbf{1} + c\left(\mathbf{1}^\top \begin{bmatrix} -\mathbf{G}_2^{-1}v \\ 1 \end{bmatrix}\right)^2. \tag{6.12}$$

The second term in the previous equation is now investigated. We know that $\mathbf{G}_3^{-1}$ is a positive definite matrix by Lemma 6.22. Since c is a principal submatrix of a positive definite matrix, c is positive definite by Lemma 6.20. It follows that $c \geq 1$ with equality if and only if $v = \mathbf{0}$. Now, we consider three cases. If $\mathbf{1}^\top(-\mathbf{G}_2)^{-1}v = 0$, this means that $v_1 + v_2 = 0$. If $v_1 = v_2 = 0$, then

$$c\left(\mathbf{1}^\top \begin{bmatrix} -\mathbf{G}_2^{-1}v \\ 1 \end{bmatrix}\right)^2 = 1.$$

If $v_1 + v_2 = 0$ and $v_1 = -v_2 \neq 0$, then we get $c > 1$ and so

$$c\left(\mathbf{1}^\top \begin{bmatrix} -\mathbf{G}_2^{-1}v \\ 1 \end{bmatrix}\right)^2 > 1.$$

Lastly, we show that $\mathbf{1}^\top(-\mathbf{G}_2)^{-1}v < 0$ is not possible. Suppose that $\mathbf{1}^\top(-\mathbf{G}_2)^{-1}v < 0$. Then $\mathbf{1}^\top(-\mathbf{G}_2)^{-1}(-v) > 0$. Therefore, the basis form with $\mathbb{B}_2^3$ and $-\mathbb{B}_1^3$, say $\widetilde{\mathbb{B}}_3$, has a grand sum $\mathbf{1}^\top\mathbf{G}(\widetilde{\mathbb{B}}_3)^{-1}\mathbf{1}$ strictly greater than $\mathbf{1}^\top\mathbf{G}(\mathbb{B}_3)^{-1}\mathbf{1}$. This is a contradiction to the assumption that $\mathbf{1}^\top\mathbf{G}(\mathbb{B}_3)^{-1}\mathbf{1}$ is the maximum for all possible bases of $\mathbb{R}^3$ contained in $\mathbb{D}_{3,5}$ since it is in $\overline{\mathbf{A}}(\mathbb{D}_{3,5})$. Therefore, we must have $\mathbf{1}^\top(-\mathbf{G}_2)^{-1}v \geq 0$ and it follows that

$$c\left(\mathbf{1}^\top \begin{bmatrix} -\mathbf{G}_2^{-1}v \\ 1 \end{bmatrix}\right)^2 \geq 1,$$

with equality if and only if $v = \mathbf{0}$. □

Proposition 6.48 (Orthogonality of the subspaces)**.** *Let $\mathbb{D}_{3,5} = \mathbb{D}_1^3 \cup \mathbb{D}_2^3$ be a positive basis of size 5 in $\mathbb{R}^3$ where $\mathbb{D}_1^3$ is a positive basis of a subspace L_1 in $\mathbb{R}^3$ and $\mathbb{D}_2^3$ is a positive basis of the subspace L_2 in $\mathbb{R}^3$, $L_1 \oplus L_2 = \mathbb{R}^3$ such that $L_1 \cap L_2 = \{\mathbf{0}\}$. If $\mathbb{D}_{3,5}$ is optimal over $\Omega_{3,5}$, then*

$$L_1 \perp L_2.$$

Proof. By contradiction. Suppose $\mathbb{D}_{3,5}$ is optimal and that the two subspaces are not orthogonal to each other. Let $\mathbb{B}_3$ be a basis of $\mathbb{R}^3$ in $\overline{\mathbf{A}}(\mathbb{D}_{3,5})$. We know that $\mathbb{B}_3$ admits a partition

$$\mathbb{B}_3 = \mathbb{B}_2^3 \cup \mathbb{B}_1^3,$$

where $\mathbb{B}_2^3$ is contained in $\mathbb{D}_2^3$ and $\mathbb{B}_1^3$ is contained in $\mathbb{D}_1^3$. By Lemma 6.47, we conclude that the only possible way that $\mathbb{D}_{3,5}$ is optimal is to have $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1}$ strictly less than the best value for a positive basis where both subspaces are orthogonal since orthogonality of the subspaces decreases the grand sum $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_3)^{-1}\mathbf{1}$ for a fix value of $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1}$. We now show that it is not possible to obtain a value of $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1}$ strictly less than the value obtained when $\mathbb{B}_2^3$ is picked from an optimal minimal positive basis $\mathring{\mathbb{D}}_2$ of $\mathbb{R}^2$ in $\mathbb{R}^3$. In this case, $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1} = 2^2 = 4$. By contradiction. Suppose that $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1} < 4$. This means that there exists a basis of L_2 in $\mathbb{R}^3$ contained in $\mathbb{D}_2^3$, say $\widetilde{\mathbb{B}}_2^3$, such that

$$\mathbf{1}^\top \mathbf{G}(\widetilde{\mathbb{B}}_2^3)^{-1}\mathbf{1} > 4 > \mathbf{1}^\top \mathbf{G}(\mathbb{B}_2^3)^{-1}\mathbf{1}.$$

Form $\widetilde{\mathbb{B}}_3$ by choosing $\widetilde{\mathbb{B}}_2^3$ and a vector d contained in $\mathbb{D}_1^3$ such that

$$c\left(\mathbf{1}^\top \begin{bmatrix} -\mathbf{G}(\widetilde{\mathbb{B}}_2^3)^{-1}v \\ 1 \end{bmatrix}\right)^2 > 1,$$

where $v = (\widetilde{\mathbb{B}}_2^3)^\top d \in \mathbb{R}^2$. Since $\mathbb{B}_3$ is in $\overline{\mathbf{A}}(\mathbb{D}_{3,5})$, it maximizes the grand sum $\mathbf{1}^\top \mathbf{G}(\cdot)^{-1}\mathbf{1}$ for all positive bases of $\mathbb{R}^3$ contained in $\mathbb{D}_{3,5}$. Hence,

$$\mathbf{1}^\top \mathbf{G}(\widetilde{\mathbb{B}}_3)\mathbf{1} \leq \mathbf{1}^\top \mathbf{G}(\mathbb{B}_3)^{-1}\mathbf{1}.$$

Let $\mathbb{D}'_{3,5} = \mathrm{Diag}(\mathring{\mathbb{D}}_2, \mathring{\mathbb{D}}_1)$ and $\mathbb{B}'_3 \in \overline{\mathbf{A}}(\mathbb{D}'_{3,5})$. Note that $\mathbb{D}'_{3,5}$ is in $\Omega_{3,5}$. Since all terms in (6.12) for $\mathbf{1}^\top \mathbf{G}(\mathbb{B}'_3)^{-1}\mathbf{1}$ are strictly less than all corresponding terms in (6.12) for $\mathbf{1}^\top \mathbf{G}(\widetilde{\mathbb{B}}_3)^{-1}\mathbf{1}$, we obtain

$$\mathbf{1}^\top \mathbf{G}(\mathbb{B}'_3)^{-1}\mathbf{1} < \mathbf{1}^\top \mathbf{G}(\widetilde{\mathbb{B}}_3)^{-1}\mathbf{1} \leq \mathbf{1}^\top \mathbf{G}(\mathbb{B}_3)^{-1}\mathbf{1}.$$

By Proposition 6.46, this means that $\mathrm{cm}(\mathbb{D}'_{3,5}) > \mathrm{cm}(\mathbb{D}_{3,5})$. This is a contradiction to the assumption that $\mathbb{D}_{3,5}$ is optimal over $\Omega_{3,5}$.

Therefore, if $\mathbb{D}_{3,5}$ is optimal, then the two subspaces L_1 and L_2 must be orthogonal. □

Corollary 6.49. *Let* $\mathbb{D}_{3,5} = \mathrm{Diag}\left(\mathring{\mathbb{D}}_2, \mathring{\mathbb{D}}_1\right)$. *Then* $\mathbb{D}_{3,5}$ *is optimal over* $\Omega_{3,5}$.

Proof. Let $\mathbb{D}'_{3,5} = \mathbb{D}_2^3 \cup \mathbb{D}_1^3$ be an optimal positive basis over $\Omega_{3,5}$ where $\mathbb{D}_2^3$ is a positive basis of the subspace L_2 in $\mathbb{R}^3$ and $\mathbb{D}_1^3$ is a positive basis of the subspace L_1 in $\mathbb{R}^3$. By Proposition 6.48, $(\mathbb{D}_2^3)^\top \mathbb{D}_1^3 = \mathbf{0}_{3\times 2} \in \mathbb{R}^{3\times 2}$. Let

$\mathbb{B}_3 = \mathbb{B}_2^3 \cup \mathbb{B}_1^3$ be a basis of $\mathbb{R}^3$ in $\overline{\mathbf{A}}(\mathbb{D}'_{3,5})$. Realigning the positive basis $\mathbb{D}'_{3,5}$ if necessary, the Gram matrix of $\mathbb{B}_3$ is

$$\mathbf{G}(\mathbb{B}_3) = \mathrm{Diag}(\mathbf{G}(\mathbb{B}_2), 1),$$

where $\mathbb{B}_2$ is a basis of $\mathbb{R}^2$. The inverse of $\mathbf{G}(\mathbb{B}_3)$ is

$$\mathbf{G}(\mathbb{B}_3)^{-1} = \mathrm{Diag}(\mathbf{G}(\mathbb{B}_2)^{-1}, 1).$$

The cosine measure is given by

$$\mathrm{cm}(\mathbb{D}'_{3,5}) = \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_2)^{-1}\mathbf{1} + 1}}$$

The minimal value of $\mathbf{1}^\top \mathbf{G}(\mathbb{B}_2)^{-1}\mathbf{1}$ is obtained if and only if $\mathbb{B}_2$ is picked from an optimal positive basis $\mathring{\mathbb{D}}_2$ in $\mathbb{R}^2$. Hence, $\mathbb{D}_2^3$ contained an optimal minimal positive basis of $\mathbb{R}^2$ in $\mathbb{R}^3$, and we get that $\mathrm{cm}(\mathbb{D}'_{3,5}) = \mathrm{cm}(\mathbb{D}_{3,5})$. Therefore, $\mathbb{D}_{3,5}$ is optimal over $\Omega_{3,5}$. □

The following figure illustrates an optimal positive basis of $\mathbb{R}^3$ over $\Omega_{3,s}$ for each possible size ($s = 4, 5, 6$). Note that $\mathring{\mathbb{D}}_3$ is optimal over $\mathcal{P}_{3,4}$ and $\mathring{\mathbb{D}}_{3,6}$ is optimal over $\mathcal{P}_{3,6}$ since $\Omega_{n,s} = \mathcal{P}_{n,s}$ whenever $s \in \{n+1, 2n\}$.

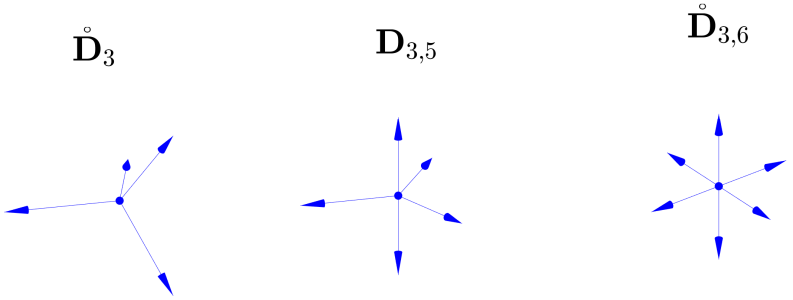


Figure 6.1: An optimal positive basis of $\mathbb{R}^3$ over $\Omega_{3,s}$ for each possible size s.

It is still unclear if $\mathbb{D}_{3,5} = \mathrm{Diag}(\mathring{\mathbb{D}}_2, \mathring{\mathbb{D}}_1)$ is optimal over $\mathcal{P}_{3,5}$. To prove that, it must be shown there exists no positive basis of $\mathbb{R}^3$ with 5 vectors that admits a partition where not all the critical vectors are equal to zero that provides a greater cosine measure than $\mathrm{cm}(\mathbb{D}_{3,5})$. A numerical experiment has been conducted and the results suggest that $\mathbb{D}_{3,5}$ is optimal over $\mathcal{P}_{3,5}$, but no rigorous proof has been done yet. This topic is an obvious future research question to explore.

The next section investigates optimality of CFOPBs in a general space $\mathbb{R}^n$.

6.5 CFOPB in $\mathbb{R}^n$

In the previous section, we have shown that the two subspaces of $\mathbb{R}^3$ must be orthogonal for a positive basis of size 5 to be optimal over $\Omega_{3,5}$. In other words, a CFPB must be a CFOPB to be optimal over $\Omega_{3,5}$. It seems reasonable to believe that a CFPB must be a CFOPB to be optimal over $\Omega_{n,s}$ where $n > 3$ and $n + 1 < s < 2n$. However, rigorously proving that it is the case (or proving that it is false) has still not been completed.

In this section, we investigate the properties of an optimal positive basis over $\Omega^+_{n,s}$. We will also see that an optimal positive basis over $\Omega^+_{n,s}$ has a nice structure that makes it easy to generate on a computer. We discuss how to verify if a given positive basis is in $\Omega^+_{n,s}$ and then present an efficient algorithm to compute the cosine measure of any CFOPB.

The next theorem presents two sufficient conditions for a positive basis in $\Omega^+_{n,s}$ to be optimal over $\Omega^+_{n,s}$. In the next theorem, the notation $\text{rem}(\frac{a}{b})$ is used to denote the remainder of the division a/b, where b is non-zero. Also, the notation $|I|$, where I is a finite index set, is used to represent the number of elements in I.

Theorem 6.50. *Let $\mathbb{D}_{n,s}$ be a positive basis of $\mathbb{R}^n$ in $\Omega^+_{n,s}$. If the following two properties are satisfied, then $\mathbb{D}_{n,s}$ is optimal over $\Omega^+_{n,s}$.*

(i) All the minimal positive bases $\mathbb{D}_{m_i}$ of L_i in $\mathbb{R}^n$ involved in a partition of $\mathbb{D}_{n,s}$ are optimal, and

(ii) the dimensions m_i of the positive bases $\mathbb{D}^n_{m_i}$ of a subspace L_i in $\mathbb{R}^n$ satisfy

$$m_j = \left\lfloor \frac{n}{s-n} \right\rfloor, \quad j \in J, \qquad m_k = \left\lceil \frac{n}{s-n} \right\rceil, \quad k \in K,$$

where J and K are disjoint index set such that

$$J \cup K = \{1, 2, \ldots, s-n\}, \qquad |J| = s - n - \text{rem}\left(\frac{n}{s-n}\right),$$
$$|K| = \text{rem}\left(\frac{n}{s-n}\right).$$

Proof. Suppose that $\mathbb{D}_{n,s}$ is a positive basis of $\mathbb{R}^n$ in $\Omega^+_{n,s}$ such that properties *(i)*, *(ii)* are satisfied and that $\mathbb{D}_{n,s}$ is not optimal. This means that there exists an optimal positive basis of size s in $\mathbb{R}^n$, say $\mathbb{D}'_{n,s}$, such that $\mathrm{cm}(\mathbb{D}'_{n,s}) > \mathrm{cm}(\mathbb{D}_{n,s})$. Let $\mathbb{B}_n \in \overline{\mathbf{A}}(\mathbb{D}'_{n,s})$. By Corollary 6.45, it follows that $\mathbb{B}_n$ admits the partition

$$\mathbb{B}_n = \mathbb{B}^n_{m_1} \cup \mathbb{B}^n_{m_2} \cup \cdots \cup \mathbb{B}^n_{m_{s-n}}$$

where $\mathbb{B}^n_{m_i} \in \mathbb{R}^{n \times m_i}$ is a basis of the subspace L_i in $\mathbb{R}^n$ for all $i \in \{1, \ldots, s-n\}$. Also, we know that $\mathbb{B}^n_{m_i} \subset \mathbb{D}^n_{m_i}$ where $\mathbb{D}^n_{m_i}$ is a minimal positive basis of L_i in $\mathbb{R}^n$ for all $i \in \{1, \ldots, s-n\}$. Since $\mathbb{D}'_{n,s}$ is in $\Omega^+_{n,s}$, all the subspaces L_i are orthogonal to each other. Hence, the Gram matrix of $\mathbb{B}_n$ is

$$\mathbf{G}(\mathbb{B}_n) = \mathrm{Diag}(\mathbf{G}(\mathbb{B}^n_{m_1}), \ldots, \mathbf{G}(\mathbb{B}^n_{m_{s-n}})).$$

The inverse of $\mathbf{G}(\mathbb{B}_n)$ is

$$\mathbf{G}(\mathbb{B}_n)^{-1} = \mathrm{Diag}(\mathbf{G}(\mathbb{B}^n_{m_1})^{-1}, \ldots, \mathbf{G}(\mathbb{B}^n_{m_{s-n}})^{-1}).$$

Hence, the cosine measure of $\mathbb{D}'_{n,s}$ is given by

$$\mathrm{cm}(\mathbb{D}'_{n,s}) = \frac{1}{\sqrt{\sum_{i=1}^{s-n} \mathbf{1}^\top \mathbf{G}(\mathbb{B}^n_{m_i})^{-1}\mathbf{1}}}.$$

Since $\mathbb{D}'_{n,s}$ is optimal, we have that

$$\sum_{i=1}^{s-n} \mathbf{1}^\top \mathbf{G}(\mathbb{B}^n_{m_i})^{-1}\mathbf{1}$$

is minimal. Hence, we must have that each $\mathbb{B}^n_{m_i}$ is contained in an optimal minimal positive basis $\mathring{\mathbb{D}}^n_{m_i}$ of L_i in $\mathbb{R}^n$. So the sum in the previous equation is equal to

$$\sum_{i=1}^{s-n} \mathbf{1}^\top \mathbf{G}(\mathbb{B}^n_{m_i})^{-1}\mathbf{1} = \sum_{i=1}^{s-n} m_i^2.$$

The minimal possible value of $\sum_{i=1}^{s-n} m_i^2$ is obtained by solving the following optimization problem:

$$\text{Minimize} \sum_{i=1}^{s-n} m_i^2 \quad \text{subject to} \sum_{i=1}^{s-n} m_i = n, \quad m_i \in \mathbb{N}\,. \tag{6.13}$$

The integer solution is given by

$$m_j = \left\lfloor \frac{n}{s-n} \right\rfloor, \quad j \in J, \qquad\qquad m_k = \left\lceil \frac{n}{s-n} \right\rceil, \quad k \in K,$$

where J and K are disjoint index set such that

$$J \cup K = \{1, 2, \ldots, s-n\}, \qquad |J| = s - n - \operatorname{rem}\left(\frac{n}{s-n}\right),$$
$$|K| = \operatorname{rem}\left(\frac{n}{s-n}\right).$$

But then we obtain that

$$\operatorname{cm}(\mathbb{D}_{n,s}) = \operatorname{cm}(\mathbb{D}'_{n,s}).$$

A contradiction. Therefore, a positive basis of $\mathbb{R}^n$ satisfying Properties *(i)*, *(ii)* must be optimal over $\Omega^+_{n,s}$. □

Corollary 6.51 (The cosine measure of an optimal positive basis over $\Omega^+_{n,s}$)**.** *Let $\mathbb{D}_{n,s}$ be an optimal positive basis over $\Omega^+_{n,s}$. Let $\mathtt{r} = \operatorname{rem}\left(\frac{n}{s-n}\right)$. Then*

$$\operatorname{cm}(\mathbb{D}_{n,s}) = \frac{1}{\sqrt{(s-n-\mathtt{r})\left\lfloor \frac{n}{s-n} \right\rfloor^2 + \mathtt{r}\left\lceil \frac{n}{s-n} \right\rceil^2}}.$$

In particular, when $\mathring{\mathbb{D}}_n$ is an optimal minimal positive basis over $\Omega^+_{n,n+1}$ $(= \mathcal{P}_{n,n+1})$, we obtain

$$\operatorname{cm}(\mathring{\mathbb{D}}_n) = \frac{1}{n}.$$

This agrees with the value of the cosine measure provided in [Næv18, Theorem 1] for a minimal positive basis to be optimal over $\mathcal{P}_{n,n+1}$. When $\mathbb{D}_{n,2n}$ is an optimal positive basis over $\Omega^+_{n,2n}$, we obtain

$$\operatorname{cm}(\mathbb{D}_{n,2n}) = \frac{1}{\sqrt{n}}. \tag{6.14}$$

Once again, this value agrees with the value provided in [Næv18, Theorem 2] for a maximal positive basis to be optimal over $\mathcal{P}_{n,2n}$. Hence, for both minimal and maximal positive bases to be optimal over $\mathcal{P}_{n,n+1}$ and $\mathcal{P}_{n,2n}$, respectively, it is necessary that they are contained in $\Omega^+_{n,n+1}$ and $\Omega^+_{n,2n}$, respectively. This provides an argument to believe that this is also the case

for intermediate positive bases. However, two facts remain to be proved (or disproved) in $\mathbb{R}^n$ before concluding that it is the case (or not).

The following table provides the diagonal blocks contained in an optimal positive basis of $\mathbb{R}^n$ (over $\Omega^+_{n,s}$) of the form $\mathbb{D}_{n,s} = \text{Diag}(\mathring{\mathbb{D}}_{m_1}, \ldots, \mathring{\mathbb{D}}_{m_{s-n}})$. The notation $(\mathring{\mathbb{D}}_{m_i})^k$ where k is a positive integer means that the diagonal block $\mathring{\mathbb{D}}_{m_i}$ appears k times as a diagonal block in $\mathbb{D}_{n,s}$.

Table 6.1: The diagonal blocks in an optimal positive basis $\mathbb{D}_{n,s} = \text{Diag}(\mathring{\mathbb{D}}_{m_1}, \ldots, \mathring{\mathbb{D}}_{m_{s-n}})$ over $\Omega^+_{n,s}$

$s \setminus n$	2	3	4	5	6	7	8
3	$\mathring{\mathbb{D}}_2$	-	-	-	-	-	-
4	$(\mathring{\mathbb{D}}_1)^2$	$\mathring{\mathbb{D}}_3$	-	-	-	-	-
5	-	$\mathring{\mathbb{D}}_2, \mathring{\mathbb{D}}_1$	$\mathring{\mathbb{D}}_4$	-	-	-	-
6	-	$(\mathring{\mathbb{D}}_1)^3$	$(\mathring{\mathbb{D}}_2)^2$	$\mathring{\mathbb{D}}_5$	-	-	-
7	-	-	$\mathring{\mathbb{D}}_2, (\mathring{\mathbb{D}}_1)^2$	$\mathring{\mathbb{D}}_2, \mathring{\mathbb{D}}_3$	$\mathring{\mathbb{D}}_6$	-	-
8	-	-	$(\mathring{\mathbb{D}}_1)^4$	$(\mathring{\mathbb{D}}_2)^2, \mathring{\mathbb{D}}_1$	$(\mathring{\mathbb{D}}_3)^2$	$\mathring{\mathbb{D}}_7$	-
9	-	-	-	$\mathring{\mathbb{D}}_2, (\mathring{\mathbb{D}}_1)^3$	$(\mathring{\mathbb{D}}_2)^3$	$\mathring{\mathbb{D}}_4, \mathring{\mathbb{D}}_3$	$\mathring{\mathbb{D}}_8$
10	-	-	-	$(\mathring{\mathbb{D}}_1)^5$	$(\mathring{\mathbb{D}}_2)^2, (\mathring{\mathbb{D}}_1)^2$	$\mathring{\mathbb{D}}_3, (\mathring{\mathbb{D}}_2)^2$	$(\mathring{\mathbb{D}}_4)^2$
11	-	-	-	-	$\mathring{\mathbb{D}}_2, (\mathring{\mathbb{D}}_1)^4$	$(\mathring{\mathbb{D}}_2)^3, \mathring{\mathbb{D}}_1$	$(\mathring{\mathbb{D}}_3)^2, \mathring{\mathbb{D}}_2$
12	-	-	-	-	$(\mathring{\mathbb{D}}_1)^6$	$(\mathring{\mathbb{D}}_2)^2, (\mathring{\mathbb{D}}_1)^3$	$(\mathring{\mathbb{D}}_2)^4$
13	-	-	-	-	-	$\mathring{\mathbb{D}}_2, (\mathring{\mathbb{D}}_1)^5$	$(\mathring{\mathbb{D}}_2)^3, (\mathring{\mathbb{D}}_1)^2$
14	-	-	-	-	-	$(\mathring{\mathbb{D}}_1)^7$	$(\mathring{\mathbb{D}}_2)^2, (\mathring{\mathbb{D}}_1)^4$
15	-	-	-	-	-	-	$\mathring{\mathbb{D}}_2, (\mathring{\mathbb{D}}_1)^6$
16	-	-	-	-	-	-	$(\mathring{\mathbb{D}}_1)^8$

A Matlab code is available on request to generate an optimal positive basis over $\Omega^+_{n,s}$ of any dimension n and size s. Note that each minimal positive basis of a subspace in $\mathbb{R}^n$ may be realigned in their respective subspace to include a specific vector of the subspace. The whole positive basis may also be realigned to include a specific vector of $\mathbb{R}^n$. These realignments do not affect the value of the cosine measure as it can be done by multiplying $\mathbb{D}_{n,s}$ with an orthonormal matrix (Proposition 6.30). A method to accomplish these realignments is provided in [JBNS19].

Next we briefly investigate the relation between the problem of finding the structure of an optimal positive basis and the more popular problem of maximizing the minimum distance between points on the unit sphere.

Relation with the problem of maximizing the minimum distance between s points on a unit sphere

We begin by defining the *Cosine measure problem* and the *Minimum distance problem*.

Let $\mathbb{D}_{n,s} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ be a positive basis of $\mathbb{R}^n$ where each vector d^j is a unit vector, $j \in \{1, \ldots, s\}$. The Cosine measure problem (CMP) is defined as

$$\text{Maximize} \quad \min_{\|u\|=1} \max_{d^j \in \mathbb{D}_{n,s}} \cos(\theta_{u,d^j})$$

where $0 < \theta_{u,d^j} \leq 2\pi$ is the angle in radians between the unit vector u and $d^j \in \mathbb{D}_{n,s}$. Let P_{n,s_p} be a set of s_p points written in vector form on the unit sphere $S_n(\mathbf{0}; 1)$ where $s_p \geq 2$. The Minimum distance problem (MDP) is defined as

$$\text{Maximize} \quad \min_{p^i, p^j \in P_{n,s_p}, i \neq j} \|p^i - p^j\|.$$

The MDP can be rewritten as

$$\begin{aligned} & \text{Maximize} && \min_{p^i, p^j \in P_{n,s_p}, i \neq j} \|p^i - p^j\| \\ \iff & \text{Maximize} && \min_{p^i, p^j \in P_{n,s_p}, i \neq j} \theta_{p^i, p^j} \\ \iff & \text{Minimize} && \max_{p^i, p^j \in P_{n,s_p}, i \neq j} \cos(\theta_{p^i, p^j}). \end{aligned}$$

Note that the MDP has been investigated for a number of points greater than $2n$. Results on the MDP can be found in [CS98] for example. If we think of the unit vectors in a positive basis $\mathbb{D}_{n,s}$ of $\mathbb{R}^n$ as points on the unit sphere in $\mathbb{R}^n$, we may ask ourselves if the optimal positive bases over $\Omega^+_{n,s}$ found in Section 6.5 are solutions to the MDP. Conversely, when the number of points $n + 1 \leq s_p \leq 2$, is a solution to the MDP necessarily an optimal positive basis of $\mathbb{R}^n$ of size s_p? These two questions are now answered.

An important result regarding the MDP is the following.

Theorem 6.52. *[Kup01, Theorem 2] If $n+2$ points lie in the n-dimensional Euclidean unit ball, then at least one of the distances between the points is smaller than or equal to $\sqrt{2}$.*

Equivalently, one of the angle between two points is less than or equal to $\pi/2$ radians. Note that this upper bound is also an upper bound for a

number of points greater than $n+2$. Hence, if $\mathbb{D}_{n,s}$ is a positive basis of $\mathbb{R}^n$, where $s \geq n+2$, such that

$$\min_{d^i, d^j \in \mathbb{D}_{n,s}, i \neq j} \|d^i - d^j\| = \sqrt{2},$$

then it must solve the MDP.

In the next lemma, it is shown that the minumum distance between two distinct points created by the unit directions in an optimal CFOPB is $\sqrt{2}$.

Lemma 6.53. *Let* $\mathbb{D}_{n,s} = \begin{bmatrix} d^1 & d^2 & \cdots & d^s \end{bmatrix}$ *be an optimal CFOPB where* $n+2 \leq s \leq 2n$. *Then*

$$\min_{d^i, d^j \in \mathbb{D}_{n,s}, i \neq j} \|d^i - d^j\| = \sqrt{2}.$$

Proof. Suppose d^k and d^ℓ, $k \neq \ell$, are not in the same minimal positive basis of subspace in $\mathbb{R}^n$ in a partition of $\mathbb{D}_{n,s}$ as described in (6.6). Then

$$(d^k)^\top (d^\ell) = cos(\theta_{d^k, d^\ell}) = 0$$

since all positive bases of the subspaces are pairwise orthogonal to each other. Hence, $\|d^k - d^\ell\| = \sqrt{2}$.

Suppose d^k and d^ℓ, $k \neq \ell$, are in the same minimal positive basis of a subspace in $\mathbb{R}^n$, say $\mathbb{D}^n_m$ where $1 \leq m \leq n-1$. Then

$$(d^k)^\top d^\ell = \cos(\theta_{d_k, d_\ell}) = -1/m < 0$$

for any $1 \leq m \leq n-1$. Hence, $\|d^k - d^\ell\| > \sqrt{2}$.

Therefore, $\min_{d^i, d^j \in \mathbb{D}_{n,s}, i \neq j} \|d^i - d^j\| = \sqrt{2}$. □

When the number of points is $n+1$, it is known (see [CS98]) that a solution to the MDP has the property

$$\max_{p^i, p^j \in P_{n,sp}, i \neq j} \cos(\theta_{p^i, p^j}) = -\frac{1}{n}.$$

From [Næv18, Theorem 1], we know it is also the case for an optimal positive basis $\mathring{\mathbb{D}}_n$ of $\mathbb{R}^n$.

Theorem 6.54. *An optimal positive basis over* $\Omega^+_{n,s}$ *where* $n+1 \leq s \leq 2n$ *is a solution to the MDP.*

Proof. This follows immediately from Theorem 6.52, Lemma 6.53 and the previous paragraph. □

The next example shows that a solution to the MDP is not necessarily an optimal positive basis of $\mathbb{R}^n$.

Example 6.55. Let

$$P_{3,5} = \begin{bmatrix} p^1 & \cdots & p^5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1/\sqrt{2} & 0 & 0 \\ 0 & 1 & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

be a set of 5 points on the unit sphere in $\mathbb{R}^3$. Then

$$\max_{p_i, p_j \in P_{3,5}, i \neq j} \|p^i - p^j\| = \sqrt{2}.$$

By Theorem 6.52, the set of points $P_{3,5}$ solves the MDP. However, the positive basis $\mathbb{D}_{3,5} = P_{3,5}$ has cosine measure

$$\operatorname{cm}(\mathbb{D}_{3,5}) \approx 0.3574 < 0.4472 \approx \frac{1}{\sqrt{5}},$$

where $\frac{1}{\sqrt{5}}$ is the value of the cosine measure for an optimal positive basis over $\Omega^+_{3,5}$ (Corollary 6.49). Hence, $P_{3,5}$ cannot be an optimal positive basis of $\mathbb{R}^3$.

To conclude this section, it is shown that the efficiency of Algorithm 1 may be significantly improved when the input is a CFOPB.

Computing the cosine measure of a CFOPB

We begin by briefly describing how CFOPBs can be identified using the partition provided in (6.6). Second, an efficient algorithm to compute the cosine measure of any CFOPB is proposed.

Next we show that the Gram matrix of a minimal positive basis $\mathbb{D}_n$ of $\mathbb{R}^n$ is an elementary block diagonal matrix (defined in Definition 6.3).

Proposition 6.56. *Let $\mathbb{D}_n$ be a minimal positive basis of $\mathbb{R}^n$. Then $\mathbf{G}(\mathbb{D}_n)$ is an elementary block diagonal matrix.*

Proof. We proceed by contradiction and suppose that $\mathbf{G}(\mathbb{D}_n)$ is not an elementary block diagonal matrix. Therefore there exists some matrix G' with more than one elementary diagonal block and some permutation matrix P

such that $G' = P^{-1}\mathbf{G}(\mathbb{D}_n)P$. We reorder the columns of $\mathbb{D}_n$ by considering the new matrix $\mathbb{D}_n P$. Note that $(\mathbb{D}_n P)^\top \mathbb{D}_n P = G'$ so G' is the Gram matrix of this new matrix. Therefore we found a way to reorder the elements of $\mathbb{D}_n$ in such a way that we can partition the set as

$$\mathbb{D}_n = D_1 \cup D_2 \quad \text{with} \quad \mathbf{G}(\mathbb{D}_n) = \operatorname{Diag}(\mathbf{G}(D_1), \mathbf{G}(D_2)).$$

Note that these blocks of $\mathbf{G}(\mathbb{D}_n)$ do not have to be elementary. Let $|D_1| = m_1, |D_2| = m_2$. We know that $m_1 + m_2 = n + 1$.
Since $\mathbb{D}_n$ is a minimal positive basis, it is of the form $\mathbb{B}_n \cup \{v\}$ where $\mathbb{B}_n$ is a basis of $\mathbb{R}^n$ and $v \in -\operatorname{pspan}^+(\mathbb{B}_n)$. Therefore any strict subset of $\mathbb{D}_n$ is linearly independent and in particular, both of the sets D_i are bases of their span. We now prove that they are also positive bases of their span. First, it should be noted that because these two sets are associated to two different diagonal blocks of $\mathbf{G}(\mathbb{D}_n)$, the elements of D_1 are orthogonal to those of D_2. Moreover, since $\mathbb{D}_n$ is a positive spanning set of $\mathbb{R}^n$, there exists a positive vector $x \in \mathbb{R}^n_+$ such that

$$\mathbb{D}_n x = \mathbf{0}.$$

Let us split $x \in \mathbb{R}^n$ into two positive vectors $x^1 \in \mathbb{R}^{m_1}_+$ and $x^2 \in \mathbb{R}^{m_2}_+$. We get the following formula:

$$D_1 x^1 + D_2 x^2 = \mathbf{0}.$$

Let $y = D_1 x^1$. This vector is a linear combination of elements of D_1 but also of elements of D_2 since $y = -D_2 x^2$. Based on the fact that D_1 and D_2 are orthogonal to each other, we conclude that $y^\top y = 0$ and so $y = \mathbf{0}$. This last result means that there exists a positive vector x^1 such that $D_1 x^1 = \mathbf{0}$, implying that D_1 is a positive spanning set of its span. This is a contradiction since a set of vectors cannot be both a basis and a positive spanning set of its span. □

Proposition 6.57. *Let $\mathbb{D}_{n,s}$ be a CFOPB of $\mathbb{R}^n$. Then the partitioning*

$$\mathbb{D}_{n,s} = \bigcup_{i=1}^{s-n} \mathbb{D}^n_{m_i} \tag{6.15}$$

is unique up to reordering.

Proof. We begin by reordering $\mathbb{D}_{n,s}$ so that

$$\mathbb{D}_{n,s} = \begin{bmatrix} \mathbb{D}^n_{m_1} & \cdots & \mathbb{D}^n_{m_{s-n}} \end{bmatrix}$$

and

$$\mathbf{G}(\mathbb{D}_{n,s}) = \mathrm{Diag}(\mathbf{G}(\mathbb{D}^n_{m_1}), \ldots, \mathbf{G}(\mathbb{D}^n_{m_{s-n}})),$$

where the blocks of $\mathbf{G}(\mathbb{D}_{n,s})$ are elementary by Proposition 6.56. Consider any ordering and associated partition of $\mathbb{D}_{n,s}$ of the form

$$\widehat{\mathbb{D}_{n,s}} = \begin{bmatrix} \widehat{\mathbb{D}}^n_{m'_1} & \cdots & \widehat{\mathbb{D}}^n_{m'_{s-n}} \end{bmatrix}$$

with

$$\mathbf{G}(\widehat{\mathbb{D}_{n,s}}) = \mathrm{Diag}(\mathbf{G}(\widehat{\mathbb{D}}^n_{m'_1}), \ldots, \mathbf{G}(\widehat{\mathbb{D}}^n_{m'_{s-n}})).$$

Without loss of generality, it can be assumed that $\mathbb{D}^n_{m_1} \cap \widehat{\mathbb{D}}^n_{m'_1} \neq \emptyset$. Suppose $\mathbb{D}^n_{m_1} \nsubseteq \widehat{\mathbb{D}}^n_{m'_1}$. Note that $\mathbb{D}^n_{m_1} = \begin{bmatrix} d^1 & \cdots & d^k \end{bmatrix}$, $\widehat{\mathbb{D}}^n_{m'_1} = \begin{bmatrix} \widehat{d}^1 & \cdots & \widehat{d}^{k'} \end{bmatrix}$. Let X be the largest subset of $\mathbb{D}^n_{m_1}$ such that $X \subseteq \widehat{\mathbb{D}}^n_{m'_1}$ and $Y = \mathbb{D}^n_{m_1} \backslash X$. By hypothesis, both of these sets are non-empty. Moreover, $Y \not\subset \widehat{\mathbb{D}}^n_{m'_1}$ so for any element of Y there exists $j \neq 1$ such that this element is in $\widehat{\mathbb{D}}^n_{m'_j}$. Since $X \subset \widehat{\mathbb{D}}_{m'_1}$ we conclude that any element of Y is orthogonal to X, so Y is orthogonal to X. This is a contradiction since this implies that $\mathbf{G}(\mathbb{D}^n_{m_1})$ is not an elementary diagonal block of $\mathbf{G}(\mathbb{D}_{n,s})$ as it can be split in two. Therefore, $\mathbb{D}^n_{m_1} \subseteq \widehat{\mathbb{D}}^n_{m'_1}$. Assuming that $\widehat{\mathbb{D}}^n_{m'_1} \nsubseteq \mathbb{D}^n_{m_1}$ leads to a similar contradiction. Therefore, $\mathbb{D}^n_{m_1} = \widehat{\mathbb{D}}_{m'_1}$. Repeating the same process, we obtain that for all $i \in \{1, \ldots, s\}$, there exists a $j \in \{1, \ldots, s\}$ such that $\mathbb{D}^n_{m_i} = \widehat{\mathbb{D}}_{m'_j}$. Therefore up to reordering, the partition (6.15) is unique. □

Notice that the previous proposition is also true for minimal positive bases of a subspace L in $\mathbb{R}^n$.

Theorem 6.58. *Let $\mathbb{D}_{n,s}$ be a positive basis of $\mathbb{R}^n$. Then $\mathbb{D}_{n,s}$ is a CFOPB of $\mathbb{R}^n$ if and only if $\mathbf{G}(\mathbb{D}_{n,s})$ is a block diagonal matrix with exactly $s - n$ elementary diagonal blocks.*

Proof. Suppose that $\mathbb{D}_{n,s}$ is a CFOPB of $\mathbb{R}^n$. If $s = n + 1$, then the result follows immediately from Proposition 6.56. Therefore, we can assume that $n + 1 < s \leq 2n$. Since $\mathbb{D}_{n,s}$ is a CFOPB, it can be written as the following partition of pairwise orthogonal minimal positive bases of the subspaces:

$$\mathbb{D}_{n,s} = \mathbb{D}^n_{m_1} \cup \mathbb{D}^n_{m_2} \cup \cdots \cup \mathbb{D}^n_{m_{s-n}}.$$

Reordering the columns of $\mathbb{D}_{n,s}$ such that $\mathbb{D}_{n,s} = \begin{bmatrix} \mathbb{D}^n_{m_1} & \cdots & \mathbb{D}^n_{m_{s-n}} \end{bmatrix}$, the associated Gram matrix is given by

$$\mathbf{G}(\mathbb{D}_{n,s}) = \mathrm{Diag}\left(\mathbf{G}(\mathbb{D}^n_{m_1}), \ldots, \mathbf{G}(\mathbb{D}^n_{m_{s-n}})\right).$$

Moreover, by Proposition 6.56, each diagonal block in $\mathbf{G}(\mathbb{D}_{n,s})$ is elementary.

Conversely, suppose that $\mathbf{G}(\mathbb{D}_{n,s})$ is a block diagonal matrix with exactly $s-n$ elementary diagonal blocks $D^n_{m_i}$ of size m_i, for $i \in \{1,\dots,s-n\}$, $\sum_{i=1}^{s-n} m_i = n$. Reordering the columns of $\mathbf{G}(\mathbb{D}_{n,s})$, we may write

$$\mathbf{G}(\mathbb{D}_{n,s}) = \text{Diag}\left(\mathbf{G}(D_{m_1}),\dots,\mathbf{G}(D_{m_{s-n}})\right).$$

If $s = n+1$, then $\mathbb{D}_n$ is obviously a critical-free positive basis. We now consider that $n+1 < s \leq 2n$. By definition of $\mathbf{G}(\mathbb{D}_{n,s})$, we have $(D^n_{m_i})^\top D_{m_j} = \mathbf{0}_{m_i+1\times m_j+1}$ for all $i,j \in \{1,\dots,s-n\}, i \neq j$. To finish the proof, we show that for all i, $D^n_{m_i}$ is a positive basis of its span in $\mathbb{R}^n$. Indeed, For any $x \in \mathbb{R}^n$ we can split x in $s-n$ positive vectors $x^1,\dots,x^s$ such that $\mathbb{D}_{n,s}x = \sum_{i=1}^{s-n} D_{m_i}x^i$. In the particular case where $x > \mathbf{0}$ and $\mathbb{D}_{n,s}x = \mathbf{0}$, based on the fact that the blocks $D^n_{m_i}$ are pairwise orthogonal, it follows that for all i, $(D^n_{m_i}x^i)^\top(D^n_{m_i}x^i) = 0$ and so $D^n_{m_i}x^i = \mathbf{0}$. Therefore, $\mathbb{D}^n_{m_i}$ is a positive spanning set of a subspace of dimension $m_i \geq 1$ in $\mathbb{R}^n$. Finally, each $D^n_{m_i}$ must be of size m_i+1, as

$$\sum_{i=1}^{s-n}(m_i+1) = s = |\mathbb{D}_{n,s}|.$$

Thus, the $s-n$ blocks $D^n_{m_i}$ are all minimal positive bases of their linear span in $\mathbb{R}^n$. Since they are also pairwise orthogonal, we proved that $\mathbb{D}_{n,s}$ is a CFOPB. □

The previous theorem provides a way to verify if a given positive basis of $\mathbb{R}^n$ is a CFOPB. Algorithm 3 provides a pseudo-code to verify if a given positive basis $\mathbb{D}_{n,s}$ is a CFOPB.

Algorithm 3: Determining if a positive basis is a CFOPB

Given a positive basis $\mathbb{D}_{n,s}$ of $\mathbb{R}^n$,

1. If $s = n + 1$, then $\mathbb{D}_n$ is in $\Omega^+_{n,n+1}$. Otherwise, go to **2.**

2. Compute the Gram matrix $\mathbf{G}(\mathbb{D}_{n,s}) = \mathbb{D}^\top_{n,s}\mathbb{D}_{n,s}$.

3. Determine if there exists a permutation matrix $P \in \mathbb{R}^{s\times s}$ such that

$$P\mathbf{G}(\mathbb{D}_{n,s})P^\top$$

is a block diagonal matrix with **exactly** $s - n$ **elementary diagonal blocks**.

4. Return
If such a matrix P exists, then $\mathbb{D}_{n,s}$ is in $\Omega^+_{n,s}$.
Otherwise, $\mathbb{D}_{n,s}$ is not in $\Omega^+_{n,s}$.

The difficulty in the previous algorithm is to decide if there exists such a permutation matrix P in Step **3**. A future research direction is to investigate a method to decide if such a permutation matrix exists or not. It seems like a graph theory approach and the *breadth-first search algorithm* can be utilized [MB08, Chapter 6] to answer this question.

Now we introduce an efficient deterministic algorithm to compute the cosine measure of a CFOPB. Algorithm 4 represents a major improvement over Algorithm 1 in terms of efficiency. First, a pseudo-code of the algorithm is presented. The algorithm is analyzed afterward.

Algorithm 4: Cosine measure of a CFOPB

Given a positive basis $\mathbb{D}_{n,s}$ of $\mathbb{R}^n$ in $\Omega^+_{n,s}$,

1. Identify the $s-n$ minimal sub-positive bases $\mathbb{D}^n_{m_i}$ in the partition of $\mathbb{D}_{n,s}$.

2. Denote by $\mathbb{B}^n_{m_i}$ a basis of a subspace of dimension m_i contained in $\mathbb{D}^n_{m_i}$. For each minimal sub-positive basis $\mathbb{D}^n_{m_i}, i \in \{1,\dots,s-n\}$, compute

(2.1) $\beta_i = \max\limits_{\mathbb{B}^n_{m_i} \subset \mathbb{D}^n_{m_i}} \mathbf{1}^\top \mathbf{G}(\mathbb{B}^n_{m_i})^{-1}\mathbf{1}$,

(2.2) $\mathcal{B}_i = \left\{ \mathbb{B}^n_{m_i} : \mathbf{1}^\top \mathbf{G}(\mathbb{B}^n_{m_i})^{-1}\mathbf{1} = \beta_i, \mathbb{B}^n_{m_i} \subset \mathbb{D}^n_{m_i} \right\}$.

3. Return

(3.1) $\operatorname{cm}(\mathbb{D}_{n,s}) = \dfrac{1}{\sqrt{\sum\limits_{i=1}^{s-n} \beta_i}}$,

(3.2) $c\mathbf{V}(\mathbb{D}_{n,s}) = \left\{ u \in \mathbb{R}^n : u = [\mathbb{B}^n_{m_1} \ \cdots \ \mathbb{B}^n_{m_{s-n}}]^{-\top}\mathbf{1} \operatorname{cm}(\mathbb{D}_{n,s}), \mathbb{B}^n_{m_i} \in \mathcal{B}_i \right\}$.

The next step is to show that the previous algorithm returns the exact cosine measure and the complete cosine vector set.

Theorem 6.59. *Let $\mathbb{D}_{n,s}$ be a CFOPB of $\mathbb{R}^n$, i.e., $\mathbb{D}_{n,s} \in \Omega^+_{n,s}$. Then Algorithm 4 returns the exact cosine measure of $\mathbb{D}_{n,s}$ and the cosine vector set of $\mathbb{D}_{n,s}$.*

Proof. To prove this statement, we work from Algorithm 2 and show that it can be simplified into Algorithm 4 when the input is a CFOPB.

First, let us show that the algorithm returns the exact cosine measure. Let $\mathbb{B}_n = \begin{bmatrix} \mathbb{B}^n_{m_1} & \cdots & \mathbb{B}^n_{m_{s-n}} \end{bmatrix}$ be a basis of $\mathbb{R}^n$ contained in $\mathbb{D}_{n,s}$. Then

$$\begin{aligned}
\gamma_{\mathbb{B}_n} &= \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1}\mathbf{1}}} \\
&= \frac{1}{\sqrt{\mathbf{1}^\top \operatorname{Diag}\left(\mathbf{G}(\mathbb{B}^n_{m_1}), \cdots, \mathbf{G}(\mathbb{B}^n_{m_{s-n}})\right)^{-1}\mathbf{1}}} \\
&= \frac{1}{\sqrt{\mathbf{1}^\top \operatorname{Diag}\left(\mathbf{G}(\mathbb{B}^n_{m_1})^{-1}, \cdots, \mathbf{G}(\mathbb{B}^n_{m_{s-n}})^{-1}\right)\mathbf{1}}}
\end{aligned}$$

$$= \frac{1}{\sqrt{\sum_{i=1}^{s-n} \mathbf{1}^\top \mathbf{G}(\mathbb{B}_{m_i}^n)^{-1}\mathbf{1}}}.$$

Hence, the cosine measure of $\mathbb{D}_{n,s}$ is given by

$$\begin{aligned}
\mathrm{cm}\,(\mathbb{D}_{n,s}) &= \min_{\mathbb{B}_n \subset \mathbb{D}_{n,s}} \gamma_{\mathbb{B}_n} \\
&= \min_{\substack{\mathbb{B}_{m_i}^n \subset \mathbb{D}_{m_i}^n \\ i \in \{1,\ldots,s-n\}}} \frac{1}{\sqrt{\sum_{i=1}^{s-n} \mathbf{1}^\top \mathbf{G}(\mathbb{B}_{m_i}^n)^{-1}\mathbf{1}}} \\
&= \frac{1}{\sqrt{\sum_{i=1}^{s-n} \max_{\mathbb{B}_{m_i}^n \subset \mathbb{D}_{m_i}^n} \mathbf{1}^\top \mathbf{G}(\mathbb{B}_{m_i}^n)^{-1}\mathbf{1}}}.
\end{aligned}$$

Letting

$$\beta_i = \max_{\mathbb{B}_{m_i}^n \subset \mathbb{D}_{m_i}^n} \mathbf{1}^\top \mathbf{G}(\mathbb{B}_{m_i}^n)^{-1}\mathbf{1}, \quad i \in \{1,\ldots,s-n\},$$

we obtain

$$\mathrm{cm}\,(\mathbb{D}_{n,s}) = \frac{1}{\sqrt{\sum_{i=1}^{s-n} \beta_i}}$$

which is exactly (3.1) in Algorithm 4. Last, let us show that the algorithm returns the cosine vector set. To make notation tighter, let $\mathrm{cm}\,(\mathbb{D}_{n,s}) = c$. Beginning from the definition of the vector $u_{\mathbb{B}_n}$ and the cosine vector set given in Step (1.2) and Step (2.2) of Algorithm 2, we find

$$\begin{aligned}
&c\mathbf{V}(\mathbb{D}_{n,s}) \\
&= \left\{u_{\mathbb{B}_n} \in \mathbb{R}^n : u_{\mathbb{B}_n} = \mathbb{B}_n^{-\top}\mathbf{1}\gamma_{\mathbb{B}_n}, p_{\mathbb{B}_n}^{\max} = \mathrm{cm}\,(\mathbb{D}_{n,s})\right\} \\
&= \left\{u \in \mathbb{R}^n : u = \mathbb{B}_n^{-\top}\mathbf{1}\gamma_{\mathbb{B}_n}, \gamma_{\mathbb{B}_n} = \mathrm{cm}\,(\mathbb{D}_{n,s})\right\} \\
&= \left\{u_{\mathbb{B}_n} \in \mathbb{R}^n : u_{\mathbb{B}_n} = [\mathbb{B}_{m_1} \ \cdots \ \mathbb{B}_{m_{s-n}}]^{-\top}\mathbf{1}c, \mathbf{1}^\top \mathbf{G}(\mathbb{B}_{m_i}^n)^{-1}\mathbf{1} = \beta_i, i = 1,\ldots,s-n\right\} \\
&= \left\{u_{\mathbb{B}_n} \in \mathbb{R}^n : u_{\mathbb{B}_n} = [\mathbb{B}_{m_1}^n \ \cdots \ \mathbb{B}_{m_{s-n}}^n]^{-\top}\mathbf{1}c, \mathbb{B}_{m_i}^n \in \mathcal{B}_i, i = 1,\ldots,s-n\right\}
\end{aligned}$$

which is (3.2) in Algorithm 4. □

Since all minimal positive bases of $\mathbb{R}^n$ are in $\Omega_{n,n+1}^+$, Algorithm 4 can be used to obtain the cosine measure of the *canonical positive basis* of $\mathbb{R}^n$,

denoted by $\widehat{\mathbb{D}}_n$, and defined by

$$\widehat{\mathbb{D}}_n = \begin{bmatrix} \mathrm{Id}_n & -\frac{1}{\sqrt{n}}\mathbf{1} \end{bmatrix}. \tag{6.16}$$

The next proposition provides an equation to obtain the cosine measure of the canonical minimal positive basis of $\mathbb{R}^n$. As far as we know, this result is not available in the literature.

Proposition 6.60. *Let* $\widehat{\mathbb{D}}_n = \begin{bmatrix} \mathrm{Id}_n & -\frac{1}{\sqrt{n}}\mathbf{1} \end{bmatrix}$ *where* $n \in \mathbb{N}$. *Then the cosine measure is*

$$\mathrm{cm}\,(\widehat{\mathbb{D}}_n) = \frac{1}{\sqrt{n^2 + 2(n-1)\sqrt{n}}}.$$

Proof. Since $\widehat{\mathbb{D}}_n$ is a CFOPB of $\mathbb{R}^n$, we know that

$$\mathrm{cm}\,(\widehat{\mathbb{D}}_n) = \min_{\mathbb{B}_n \subset \widehat{\mathbb{D}}_n} \gamma_{\mathbb{B}_n} = \min_{\mathbb{B}_n \subset \widehat{\mathbb{D}}_n} \frac{1}{\sqrt{\mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1}\mathbf{1}}},$$

where $\mathbb{B}_n$ is a basis of $\mathbb{R}^n$ contained in $\widehat{\mathbb{D}}_n$. Suppose $\mathbb{B}_n = \mathrm{Id}_n$. Then

$$\gamma_{\mathbb{B}_n} = \frac{1}{\sqrt{n}} \geq \frac{1}{\sqrt{n^2 + 2(n-1)\sqrt{n}}}$$

for all $n \in \{1, 2, \dots\}$ and with equality only if $n = 1$. Now, suppose $\mathbb{B}_n = \widehat{\mathbb{D}}_n \backslash \{e^i\}$ for some $i \in \{1, \dots, n\}$. We obtain

$$\mathbf{G}(\mathbb{B}_n) = \begin{bmatrix} \mathrm{Id}_{n-1} & -\frac{1}{\sqrt{n}}\mathbf{1} \\ -\frac{1}{\sqrt{n}}\mathbf{1}^\top & 1 \end{bmatrix}.$$

The inverse of $\mathbf{G}(\mathbb{B}_n)$ is

$$\mathbf{G}(\mathbb{B}_n)^{-1} = \begin{bmatrix} \mathrm{Id}_{n-1} + \mathbf{1}_{n-1\times n-1} & \sqrt{n}\mathbf{1} \\ \sqrt{n}\mathbf{1}^\top & n \end{bmatrix}.$$

Hence,

$$\begin{aligned} \mathbf{1}^\top \mathbf{G}(\mathbb{B}_n)^{-1}\mathbf{1} &= \sum_{i=1}^{n}\sum_{j=1}^{n} \left[\mathbf{G}(\mathbb{B}_n)^{-1}\right]_{i,j} \\ &= (n-1)^2 + (n-1) + 2(n-1)\sqrt{n} + n \\ &= n^2 + 2(n-1)\sqrt{n}. \end{aligned}$$

Therefore, $\mathrm{cm}\,(\widehat{\mathbb{D}}_n) = \gamma_{\mathbb{B}_n} = \frac{1}{\sqrt{n^2+2(n-1)\sqrt{n}}}$. □

The canonical positive basis can be generalized to consider non-minimal sizes.

Definition 6.61 (Canonical positive bases)**.** The canonical positive basis of $\mathbb{R}^n$ of size $n+1 < s \leq 2n$ is denoted by $\widehat{\mathbb{D}}_{n,s}$ and defined by

$$\widehat{\mathbb{D}}_{n,s} = \begin{bmatrix} \mathrm{Id}_n & B \end{bmatrix},$$

where $B = \begin{bmatrix} b^1 & \cdots & b^{s-n} \end{bmatrix} \in \mathbb{R}^{n \times s-n}$, and

$$b^k = -e^k \quad \text{for all } k \in \{1, \ldots, s-n-1\}, \quad b^{s-n} = -\sum_{i=s-n}^{n} \frac{e^i}{\sqrt{2n-s+1}}.$$

For example, the canonical positive bases in $\mathbb{R}^2$ are

$$\widehat{\mathbb{D}}_3 = \begin{bmatrix} \mathrm{Id}_3 & -\frac{1}{\sqrt{3}}\mathbf{1} \end{bmatrix}, \widehat{\mathbb{D}}_{3,5} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 1 & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}, \widehat{\mathbb{D}}_{3,6} = \begin{bmatrix} \mathrm{Id}_3 & -\mathrm{Id}_3 \end{bmatrix}.$$

When $n+1 < s \leq 2n$, note that the canonical positive basis $\widehat{\mathbb{D}}_{n,s}$ can be written as

$$\widehat{\mathbb{D}}_{n,s} = \mathrm{Diag}((\mathring{\mathbb{D}}_1)^{s-n-1}, \widehat{\mathbb{D}}_{2n-s+1})$$

where the notation $(\mathring{\mathbb{D}}_1)^{s-n-1}$ means that the positive basis $\mathring{\mathbb{D}}_1$ of $\mathbb{R}$ appears $s-n-1$ times as an elementary diagonal block. It follows that the canonical positive basis is a CFOPB for size $n+1 \leq s \leq 2n$. However, the canonical positive basis is optimal over $\Omega^+_{n,s}$ only if $s = 2n$. Using a similar process than the proof in Proposition 6.60, it can be shown that the cosine measure of the canonical positive basis $\widehat{\mathbb{D}}_{n,s}$ for any size $n+1 \leq s \leq 2n$ is

$$\mathrm{cm}(\widehat{\mathbb{D}}_{n,s}) = \frac{1}{\sqrt{n-1+\left(2n-s+\sqrt{2n-s+1}\right)^2}}. \tag{6.17}$$

When $s = n+1$, we recover the formula presented in Proposition 6.60, and when $s = 2n$, we recover the formula presented in (6.14).

The next proposition shows that Algorithm 4 only computes s matrix inverses: one at each iteration. Before introducing the next proposition, recall that any minimal positive basis of $\mathbb{R}^n$ contains exactly $n+1$ bases of $\mathbb{R}^n$ (Proposition 6.24).

Proposition 6.62. *Let $\mathbb{D}_{n,s}$ be a positive basis of $\mathbb{R}^n$ in $\Omega^+_{n,s}$. Let $\mathbb{D}_{n,s} = \mathbb{D}^n_{m_1} \cup \cdots \cup \mathbb{D}^n_{m_{s-n}}$ be its partition. Algorithm 4 computes the inverse of*

$$\sum_{i=1}^{s-n}(m_i + 1) = s$$

matrices.

Proof. There are $s - n$ positive bases $\mathbb{D}^n_{m_i}$ of a subspace L_i of dimension m_i in a partition of $\mathbb{D}_{n,s}$ and each positive basis $\mathbb{D}^n_{m_i}$ of a subspace L_i contains $m_i + 1$ bases of the subspace L_i in $\mathbb{R}^n$. Therefore, Step (2.1) of Algorithm 4 computes the following number of matrix inverses:

$$\begin{aligned}\sum_{i=1}^{s-n}(m_i + 1) &= \sum_{i=1}^{s-n} m_i + \sum_{i=1}^{s-n} 1 \\ &= n + (s - n) = s.\end{aligned}$$

□

Note that this is a significant decrease compared to the potentially $\binom{s}{n}$ matrix inverses computed by Algorithms 1 and 2. For example, if we consider an optimal maximal positive basis $\mathbb{D}_{20,40}$ of $\mathbb{R}^{20}$ the new algorithm investigates $s = 40$ set of vectors and computes only 40 matrix inverses. However, Algorithm 1 investigates $\binom{40}{20} \approx 1.3785 \times 10^{11}$ sets of n vectors, then must determine if each set is a basis of $\mathbb{R}^n$ and finally, computes $2^{20} = 1\ 048\ 576$ matrix inverses.

6.6 Summary and future research directions

In Section 6.2, we presented a deterministic algorithm to compute the cosine measure of any finite positive spanning set of $\mathbb{R}^n$. One weakness of Algorithm 1 is that the algorithm needs to investigate $\binom{s}{n}$ sets of vectors and decide if the set is a basis of $\mathbb{R}^n$. Indeed, as s increases, this number becomes extremely large. In 2021, Regis presented an alternate algorithm to compute the cosine measure of any finite set of vectors [Reg21]. Unfortunately, the algorithm presented in [Reg21] does not represent an improvement in terms of efficiency.

In Section 6.3, we investigated the structure of intermediate positive bases in $\mathbb{R}^n$. Using results from Romanowicz, we demonstrated that any positive basis of $\mathbb{R}^n$, $\mathbb{D}_{n,s}$, can be partitioned into $s - n$ positive bases of a subspace of $\mathbb{R}^n$ plus a critical vector. We have defined critical-free positive bases of $\mathbb{R}^n$ (CFPB), i.e., the positive bases that can be written as a partition

of minimal positive bases of subspaces of $\mathbb{R}^n$ in which all critical vectors are zero. A critical-free orthogonal positive basis of $\mathbb{R}^n$ has been defined to be a CFPB of $\mathbb{R}^n$ and where all subspaces involved in the partition (6.15) are pairwise orthogonal to each other. Algorithm 1 can be simplified for CFPBs. However, the resulting algorithm, Algorithm 2, is not a significant improvement in terms of efficiency. When $\mathbb{D}_{n,s}$ is a CFPB of $\mathbb{R}^n$, the number of bases of $\mathbb{R}^n$ contained in $\mathbb{D}_{n,s}$ was identified in (6.8).

In Section 6.4, we provided the structure of an optimal positive basis of intermediate size over $\Omega_{3,5}$, and proved that the two subspaces of $\mathbb{R}^n$ must be orthogonal to each other. We conjecture that this result also holds in $\mathbb{R}^n$. In reviewing this result, note that the key step would be to extend Lemma 6.23 to include a broader class of positive bases.

In Section 6.5, we focused on CFOPB of $\mathbb{R}^n$, the set of positive bases that can be written as a partition in which all critical vectors are zero and all minimal positive bases of subspaces of $\mathbb{R}^n$ are orthogonal to each other whenever $s > n + 1$. We determined a characterization of the structure of CFOPB (Theorem 6.50), and therefore determined the optimal cosine measure for positive bases in $\Omega^+_{n,s}$. It turns out that it is simple and efficient to generate such a positive basis with software such as Matlab. A Matlab code is available upon request. When a positive basis is a CFOPB, Algorithm 4 can be used to compute the cosine measure. Algorithm 4 represents a major advancement in terms of efficiency compared to Algorithm 1. It requires only s iterations and computes s matrix inverses.

Algorithm 4 does not necessarily work for a positive basis which is in $\Omega_{n,s}$ but not in $\Omega^+_{n,s}$. In this case, the inverse of the Gram matrix $\mathbf{G}(\mathbb{B}_n)$, is not necessarily a block diagonal matrix with $s - n$ elementary diagonal blocks. It follows that the proof of Theorem 6.59 does not hold in this case. Developing an efficient algorithm to compute the cosine measure of a positive basis for which the only assumption is that there exists a partition as described in (6.6) where all critical vectors equal to $\mathbf{0}$ is a future research direction to be investigated.

In order to characterize optimality for positive bases of $\mathbb{R}^n$ of intermediate size completely, two questions need to be answered. First, must all critical vectors in a partition of an optimal positive basis be zero? Second, must all subspaces involved in a partition of an optimal positive basis be orthogonal? We conjecture that the answer to both questions is yes and will further examine this in future research.

A future research direction is to to explore the relation between the CMP and the *Sum distance problem* [HS11]. The sum distance problem may be

defined as

$$\text{Maximize} \sum_{p^i, p^j \in P_{n,s_p}} \|p^i - p^j\|.$$

When the number of points is 5 in $\mathbb{R}^3$, the solution found in [HS11] can be viewed as an optimal positive basis over $\Omega^+_{3,5}$. Hence, it is possible that the CMP and the Sum distance problem are equivalent problems when $n+1 \leq s \leq 2n$.

Another research direction is to explore the properties of *positive k-spanning sets* and their value in DFO algorithms. The positive k-span of a set of vectors $\mathcal{S}$ is defined as

$$\text{pspan}_k(\mathcal{S}) = \bigcap_{\substack{\mathcal{R}\subset\mathcal{S} \\ |\mathcal{R}|=|\mathcal{S}|-k+1}} \text{pspan}(\mathcal{R}).$$

A positive k-spanning set of $\mathbb{R}^n$ is a set of vectors $\mathcal{S}$ such that $\text{pspan}_k(\mathcal{S}) = \mathbb{R}^n$. In other words, a positive k-spanning set of $\mathbb{R}^n$ is a set of vectors that remains a positive spanning set of $\mathbb{R}^n$ even if we remove any combinations of $k-1$ vectors from the set. A positive k-basis of $\mathbb{R}^n$ can be defined as a positive k-spanning set for which no proper subset exhibits the same property. The notion of the k-cosine measure of a set of non-zero vectors $\mathcal{S}$, denoted by $\text{cm}_k(\mathcal{S})$, can be defined as follows:

$$cm_k(\mathcal{S}) = \min_{\substack{\|u\|=1 \\ u\in\mathbb{R}^n}} \max_{\substack{|\mathcal{R}|=k \\ \mathcal{R}\subseteq\mathcal{S}}} \min_{d\in\mathcal{S}} \frac{u^\top d}{\|d\|}.$$

With these definitions in mind, we may now try to find the structure of positive k-bases of $\mathbb{R}^n$ with maximal k-cosine measure. The value of positive k-spanning sets in parallel DFO algorithm should be explored. Positive k-spanning sets offer the security that even if $k-1$ of the sample points created from the directions in the positive k-spanning set are unreliable, or the function values at these points are not obtained (for instance, one of the computer could fail to evaluate a sample point), the remaining directions preserve the valuable properties of a positive spanning set of $\mathbb{R}^n$.

Chapter 7

DFO algorithms

In this chapter, some of the main theoretical concepts discussed in Chapters 4, 5, and 6 are implemented in DFO algorithms. In Section 7.1, the generalized simplex Hessian (GSH) (Definition 5.5) is employed in a *derivative-free trust-region algorithm*. In Section 7.2, optimal critical-free orthogonal positive bases (CFOPB) are employed in a *generalized pattern search algorithm*. The main goal of this section is to verify if the theoretical concepts analyzed in the previous chapters may help to design more efficient DFO algorithms. Note that it is not our intention to show that the two algorithms implemented are more efficient than state-of-the-art DFO algorithms. The goal is to compare two versions of the algorithm implemented where all parameters are identical in both versions except the parameter that we want to compare.

7.1 A Derivative-free trust-region algorithm using a calculus-based approach and the GSH to build the model function

Trust-region methods are a popular class of algorithms for finding the solutions of non-linear minimization optimization problems [CGT00, NW06]. Trust-region algorithms build a model of the objective function in a neighborhood of the incumbent solution. The region in which the model function behaves similarly to the objective function is called the *trust region* and is defined through a *trust-region radius*. The optimization algorithm then finds a point in the trust region at which the model sufficiently decreases. This step is known as the *trust-region sub-problem*, and the point that provides a sufficient decrease is called the *trial point*. The value of the objective function is then computed at the trial point. If the ratio of the *achieved reduction* versus the *reduction in the model* is sufficient, then the incumbent solution is updated and set to be equal to the trial point. If the ratio is not sufficient, then the *trust-region radius* is decreased. This method iterates until a stopping condition implies that a local minimizer has been located.

Extensive research has been done on this topic since 1944, when Levenberg published what is known to be the first paper related to trust-region methods [Lev44]. Early work on trust-region methods includes the work of Dennis and Mei [DM79], Dennis and Schnabel [DS96], Fletcher [Fle80], Goldfeldt, Quandt, and Trotter [GQT66], Hebden [Heb73], Madsen [Mad75], Moré [Mor83], Moré and Sorensen [MS83], Osborne [Osb76], Powell [Pow70a, Pow70b, Pow70c, Pow75, Pow84], Sorensen [Sor81, Sor82], Steihaug [Ste83], Toint [Toi78, Toi79, Toi81a, Toi81b], and Winfield [Win70, Win73], to name a few. The name *trust region* seems to have been used for the first time in 1978 by Dennis in [Den78].

In the works mentioned above, trust-region methods were designed and analyzed under the assumption that first-order information about the objective function is available and that second-order information (i.e., Hessians) may, or may not, be available. In the case in which both first-order and second-order information are not available, or it is hard to directly obtain, *derivative-free trust-region (DFTR) methods* can be used. This type of method has become more popular in the last two decades due to the rise of blackbox optimization problems. Early works on DFTR methods include those of Conn, Scheinberg, and Toint [CST97a], Marazzi and Nocedal [MN02], Powell [Pow02, Pow03], and Colson and Toint [CT05], to name a few.

In 2009, the convergence properties of general DFTR algorithms for unconstrained optimization problems were rigorously investigated [CSV09a]. The pseudo-code of an algorithm that converges to a first-order critical point and the pseudo-code of an algorithm that converges to a second-order critical point were provided. A complete review of the DFTR methods for unconstrained optimization problems is available in [CSV09b, Chapters 10 and 11], and [AH17, Chapter 11]. There now exist several DFTR algorithms for solving unconstrained optimization problems, such as Advanced DFO-TRNS [LLRV19], BOOSTERS [OB07], CSV2 [BLG13], DFO [CST97a, CST98], UOBYQA [Pow02], and WEDGE [MN02]. In recent years, DFTR algorithms have also been developed for constrained optimization problems. When an optimization problem is bound-constrained, some of the algorithms available in the literature are BC-DFO [GTT11], BOBYQA [Pow09], ORBIT [WRS08, WS11], SNOBFIT [HN08], and TRB-POWELL [AEP11]. Other DFTR algorithms dealing with more general constrained optimization problems include CONDOR [BB05], CONORBIT [RW17], DEFT-FUNNEL [ST15], LCOBYQA [GHA14], LINCOA [Pow15], and S [CKP15].

Recall that a *blackbox* is defined as any process that returns an output (possibly infinity) whenever an input is provided, but where the inner mech-

anism of the process is not analytically available to the optimizer [AH17]. In this section, we consider a situation in which the objective function, F, is obtained by manipulating several blackboxes. For instance, F could be the product of two functions, say, f_1 and f_2, where the function values for f_1 are obtained through one blackbox and the function values for f_2 are obtained through a different blackbox. An objective function that is defined by manipulating more than one function will be called a *composite objective function* in this chapter.

Composite objective functions have inspired a particular direction of research under the assumption that the functions involved do not have the same computational costs. For instance, Khan et al. [KLW18] developed an algorithm for minimizing $F = \phi + h \circ f$, where ϕ is smooth with known derivatives, h is a known non-smooth piecewise linear function, and f is smooth but expensive to evaluate. In [LMZ21], Larson et al. investigated the minimization problem $F = h \circ f$, where h is non-smooth and inexpensive to compute and f is smooth, but its Jacobian is not available. Recently, Larson and Menickelley developed algorithms for bound-constrained non-smooth composite minimization problems in which the objective function F has the form $F = h(f(x))$, where h is cheap to evaluate, and f requires considerable time to evaluate.

These ideas have led to research on more general calculus-based approaches to approximate gradients or Hessians of composite functions.
In [HJB18, Har20, Reg15], the authors provided calculus rules (integer power, product, quotient, and chain) for generalized simplex gradients. In 2020, these results were advanced to the generalized centered simplex gradient in [HJP20] and to the generalized simplex Hessian in [HJBP20]. In [CHJB22], a unified framework that provides general error bounds for gradient and Hessian approximation techniques by using calculus rules was presented.

Previous research has shown that a calculus-based approach to approximating gradients or Hessians can be substantially more accurate than a non-calculus approach in several situations [CHJB22, HJB18, HJP20, HJBP20]. However, it is still unclear if these theoretical results translate to an improvement of efficiency in a DFO algorithm. The main goal of this section is to compare two versions of a DFTR algorithm designed to solve a box-constrained blackbox optimization problem in which the objective function is a composite function: one that employs a calculus-based approach and one that does not employ a calculus-based approach. It is worth emphasizing that it is not our intention to show that the DFTR algorithm developed in this section is better than state-of-the-art DFO algorithms. The main goal

is to analyze any benefits resulting from using a calculus-based approach in a DFTR algorithm designed to minimize a composite objective function.

In the remaining of this section, we will refer to a calculus-based approach and a non-calculus approach to approximate gradients and Hessians. Let us clarify the meaning of these two approaches.

We begin with the non-calculus approach. Let $F : \mathbb{R}^n \to \mathbb{R}$ and let $Q_F(x^k)$ be a quadratic interpolation of F at x^k using $(n+1)(n+2)/2$ distinct sample points *poised for quadratic interpolation* (Definition 5.1). An approximation of the gradient of F at x^k, denoted by g^k, is obtained by computing $\nabla Q_F(x^k)$, and an approximation of the Hessian of F at x^k, denoted by H^k, is obtained by computing $\nabla^2 Q_F(x^k)$.

We now explain the calculus-based approach. Let $F : \mathbb{R}^n \to \mathbb{R}$ be constructed using $f_1 : \mathbb{R}^n \to \mathbb{R}$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$. Let Q_{f_1} and Q_{f_2} be quadratic interpolation functions of f_1 and f_2, respectively. When a calculus-based approach is employed and the composite objective function F has the form $F = f_1 \cdot f_2$, then

$$g^k = \nabla(Q_{f_1} \cdot Q_{f_2})(x^k) = f_1(x^k)\nabla Q_{f_2}(x^k) + f_2(x^k)\nabla Q_{f_1}(x^k), \tag{7.1}$$

and

$$\begin{aligned} H^k &= \nabla^2(Q_{f_1} \cdot Q_{f_2})(x^k) \\ &= f_2(x^k)\nabla^2 Q_{f_1}(x^k) + \nabla Q_{f_1}(x^k)\left(\nabla Q_{f_2}(x^k)\right)^\top \\ &\quad + \nabla Q_{f_2}(x^k)\left(\nabla Q_{f_1}(x^k)\right)^\top + f_1(x^k)\nabla^2 Q_{f_2}(x^k). \end{aligned} \tag{7.2}$$

Similarly, when the composite objective function F has the form $F = \frac{f_1}{f_2}$, then (assuming $f_2(x^k) \neq 0$)

$$g^k = \nabla\left(\frac{Q_{f_1}}{Q_{f_2}}\right)(x^k) = \frac{f_2(x^k)\nabla Q_{f_1}(x^k) - f_1(x^k)\nabla Q_{f_2}(x^k)}{[f_2(x^k)]^2}, \tag{7.3}$$

and

$$\begin{aligned} H^k &= \nabla^2\left(\frac{Q_{f_1}}{Q_{f_2}}\right)(x^k) \\ &= \frac{1}{[f_2(x^k)]^3}\Big[[f_2(x^k)]^2\nabla^2 Q_{f_1}(x^k) - f_1(x^k)f_2(x^k)\nabla^2 Q_{f_2}(x^k) \\ &\quad + 2f_1(x^k)\nabla Q_{f_2}(x^k)\nabla Q_{f_2}(x^k)^\top \\ &\quad - f_2(x^k)\left(\nabla Q_{f_1}(x^k)\nabla Q_{f_2}(x^k)^\top + \nabla Q_{f_2}(x^k)\nabla Q_{f_1}(x^k)^\top\right)\Big]. \end{aligned} \tag{7.4}$$

To compute H^k, the technique called generalized simplex Hessian (GSH) introduced in Chapter 5 is used (see Definition 5.5). Recall that when the two matrices S and $\overline{T}$ involved in the computation of the GSH are square matrices with full rank and the set of points for GSH computation $\mathcal{S}_s(x^k; S, \overline{T})$ contains $(n+1)(n+2)/2$ distinct sample points, the GSH is equal to the Hessian of the quadratic interpolation function (details in Section 5.4). In particular, this is the case when a *minimal poised set for the GSH* is used (Definition 5.17).

Next we introduce the definition of a class of fully linear models.

Definition 7.1 (Class of fully linear models)**.** [AH17, Definition 9.1] Given $f \in \mathcal{C}^1, x \in \mathbb{R}^n$ and $\bar{\Delta} > 0$, we say that $\{\widetilde{f}_\Delta : \Delta \in (0, \bar{\Delta}]\}$ is a class of fully linear models of f at x parametrized by Δ if there exists a pair of scalars $\kappa_f \geq 0$ and $\kappa_g \geq 0$ such that, given any $\Delta \in (0, \bar{\Delta}]$, the model $\widetilde{f}_\Delta$ satisfies

1. $|f(x+s) - \widetilde{f}_\Delta(x+s)| \leq \kappa_f(x)\Delta^2 \quad$ for all $s \in \overline{B}_n(0; \Delta)$,

2. $\|\nabla f(x+s) - \nabla \widetilde{f}_\Delta(x+s)\| \leq \kappa_g(x)\Delta \quad$ for all $s \in \overline{B}_n(0; \Delta)$.

We are now ready to introduce the pseudo-code of the DFTR algorithm that will be used to compare the calculus-based approach to the non-calculus approach.

7.1.1 The algorithm

In this section, the algorithm designed to minimize a box constrained blackbox optimization problem involving a composite objective function is described. The algorithm will be used to do our comparison between the calculus-based approach and the non-calculus appproach in Section 7.1.2.

Let $\Delta_{max} > 0$ be the maximum trust-region radius allowed in the DFTR algorithm. The minimization problem considered is

$$\min_{\ell \leq x \leq u} F(x) \tag{7.5}$$

where $F : \operatorname{dom} F \subseteq \mathbb{R}^n \to \mathbb{R}$ is twice continuously-differentiable on an open domain containing

$$\bigcup_{\ell \leq x \leq u} \overline{B}_n(x; \Delta_{\max}) \tag{7.6}$$

where $\ell, u \in \mathbb{R}^n$ and the inequalities $\ell \leq x \leq u$ are taken component-wise. It is assumed that F is a composite function obtained from two blackboxes with a similar computational cost.

Before introducing the pseudo-code of the algorithm, let us clarify some details about the model function and the *trust-region sub-problem.*

The model function built at an iteration k will be denoted by m^k. The model is a quadratic function that can be written as

$$m^k(x^k + s^k) = F(x^k) + (g^k)^\top s^k + \frac{1}{2}(s^k)^\top H^k s^k, \tag{7.7}$$

where g^k denotes an approximation of the gradient $\nabla F(x^k)$, H^k is a symmetric approximation of the Hessian $\nabla^2 F(x^k)$ and $s^k \in \mathbb{R}^n$ is the independent variable.

When a calculus-based approach is used, both g^k and H^k are built using the calculus-based approach presented earlier. Similarly, if the non-calculus approach is used, then both g^k and H^k are built with the non-calculus approach. We define the sampling radius, Δ_s^k as the maximum distance between the incumbent solution x^k and a sampling point used to build the approximations g^k and H^k.

The trust-region sub-problem solved in the DFTR algorithm (Step 2 of Algorithm 5) at an iteration k is

$$\begin{aligned} \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad & m^k(x^k + s) = F(x^k) + (g^k)^\top s + \frac{1}{2}(s)^\top H^k s \\ \text{subject to} \quad & \|s\| \leq \Delta^k, \quad \text{and} \quad \ell \leq x^k + s \leq u, \end{aligned} \tag{7.8}$$

where $\Delta^k > 0$ is the trust-region radius at an iteration k.

Recall that the first-order necessary condition to solve (7.5) is that the *projected gradient* is equal to zero. The projected gradient of the model function m^k at $s = 0$ onto the feasible box, delimited by ℓ and u where $\ell < u$, is defined by

$$\pi_m^k = x^k - \text{Proj}_{[\ell,u]}(x^k - \nabla m^k(x^k)) = x^k - \text{Proj}_{[\ell,u]}(x^k - g^k)$$

where the projection operator, $\text{Proj}_{[\ell,u]}(y)$, is defined component-wise by

$$[\text{Proj}_{[\ell,u]}(y)]_i = \begin{cases} \ell_i & \text{if} \quad y_i < \ell_i, \\ u_i & \text{if} \quad y_i > u_i, \\ y_i & \text{if} \quad \text{otherwise.} \end{cases} \tag{7.9}$$

Similarly, we define the projected gradient of the objective function at $s = 0$ onto the box by

$$\pi_F^k = x^k - \mathrm{Proj}_{[\ell,u]}(x^k - \nabla F(x^k)).$$

Next a pseudo-code of our DFTR algorithm is presented. This is essentially the pseudo-code in [CSV09a, Algorithm 4.1] adapted for a box constrained problem (see also Algorithm 10.1 in [CSV09b] and Algorithm 2.1 in [HR22]). The algorithm is a first-order method, in the sense that it converges to a first-order critical point.

Algorithm 5: DFTR algorithm pseudo-code

0. Initialization.

Choose a feasible initial point x^0, an initial trust-region radius $\Delta^0 > 0$, an initial sampling radius $0 < \Delta_s^0 \leq \Delta^0$, and a maximal trust-region radius $\Delta_{\max} > 0$.
Build an initial fully linear model $m^0(x^0 + s)$ on $\overline{B}_n(x^0; \Delta^0)$.
Denote by g^0 and H^0 the gradient and Hessian of the initial model at $s = 0$.
Choose the constants $0 \leq \eta_1 \leq \eta_2 < 1$ (with $\eta_2 \neq 0$), $0 < \gamma < 1 < \gamma_{inc}, \epsilon_{stop} > 0, \mu > 0$.
Set $k = 0$.

For $k = 0, 1, 2, \ldots$

1. Criticality step

If $\|\pi_m^k\| \leq \epsilon_{stop}$ and $\Delta^k \leq \mu\|\pi_m^k\|$
stop. Return x^k.
If $\|\pi_m^k\| \leq \epsilon_{stop}$ and $\Delta^k > \mu\|\pi_m^k\|$
set $\Delta^k \leftarrow \min\{\mu\|\pi_m^k\|, \Delta^k\}$.
If $\Delta_s^k > \Delta^k$
set $\Delta_s^k \leftarrow \Delta^k$
update the model m^k to make it fully linear on $\overline{B}_n(x^k; \Delta^k)$.

2. Trust-region sub-problem

Find an approximate solution s^k to the trust-region sub-problem (7.8).

3. Acceptance of the trial point

Compute

$$\rho^k = \frac{F(x^k) - F(x^k + s^k)}{m^k(x^k) - m^k(x^k + s^k)}.$$

If $\rho^k \geq \eta_1$
set $x^{k+1} \leftarrow x^k + s^k$ and build a fully linear model m^{k+1}.
Otherwise ($\rho_k < \eta_1$),
set $m^{k+1} \leftarrow m^k$ and $x^{k+1} \leftarrow x^k$.

4. Trust-region radius update

$$\Delta^{k+1} \in \begin{cases} [\Delta^k, \min\{\gamma_{inc}\Delta^k, \Delta_{\max}\}], & \text{if} \quad \rho^k \geq \eta_2, \\ \{\gamma\Delta^k\}, & \text{if} \quad \rho^k < \eta_2. \end{cases}$$

If $\Delta_s^k > \Delta^k$
set $\Delta_s^{k+1} \leftarrow \Delta^k$.
Increment k by one and go to **Step 1**.

End For

Several details about the pseudo-code require some clarifications. The procedure employed in our algorithm to build the model function at an arbitrary iteration k guarantees that the model m^k is fully linear on $\overline{B}_n(x^k;\Delta^k)$. To see this, first note that when a model m^k is built in Algorithm 5, we always have that the sampling radius $\Delta_s^k \leq \Delta^k$. It is known that the gradient of a quadratic interpolation function built with $(n+1)(n+2)/2$ sample points poised for quadratic interpolation is $O(\Delta_s^2)$. The Hessian of this quadratic interpolation is $O(\Delta_s)$. It is shown in [CHJB22] that the calculus-based approach to approximate gradients described in (7.1) and (7.3) are $O(\Delta_s^2)$. It is also shown in [CHJB22, HJBP20] that the calculus-based approach to approximate Hessians described in (7.2) and (7.4) are $O(\Delta_s)$. Hence, g^k and H^k are $O(\Delta^k)$ accurate. The next proposition shows that if g^k and H^k are both $O(\Delta^k)$ accurate approximations of the gradient and Hessian at x^k, respectively, then the model function m^k is fully linear on $\overline{B}_n(x^k;\Delta^k)$.

Proposition 7.2. *Let $F \in \mathcal{C}^2$, $x^k \in \operatorname{dom} F, s^k \in \mathbb{R}^n$ and $\Delta^k > 0$. Assume that $x^k + s^k \in \overline{B}_n(x^k;\Delta^k)$ where k is any iteration of Algorithm 5. Let the model function m^k be defined as in Equation (7.7). If g^k is an $O(\Delta^k)$ accurate approximation of the gradient of F at x^k and H^k is an $O(\Delta^k)$ accurate approximation of the Hessian of F at x^k, then the model m^k is fully linear on $\overline{B}_n(x^k;\Delta^k)$.*

Proof. For any $x^k + s^k \in \overline{B}_n(x^k;\Delta^k)$, we have

$$\begin{aligned}
&\|\nabla F(x^k + s^k) - \nabla m^k(x^k + s^k)\| \\
&= \|\nabla F(x^k + s^k) - g^k - (s^k)^\top H^k\| \\
&\leq \|\nabla F(x^k + s^k) - \nabla F(x^k) + \nabla F(x^k) - g^k\| + \Delta^k\|H^k\| \\
&\leq L_g\Delta^k + C_1\Delta^k + \|H^k\|\Delta^k
\end{aligned}$$

where $L_g \geq 0$ is the Lipschitz constant of ∇F on $\overline{B}_n(x^k;\Delta^k)$, and $C_1 \geq 0$. Note that $\|H^k\|$ is bounded above since

$$\begin{aligned}
\|H^k\| &= \|H^k - \nabla^2 F(x^k) + \nabla^2 F(x^k)\| \\
&\leq C_2\Delta^k + \|\nabla^2 F(x^k)\|,
\end{aligned}$$

where $C_2 \geq 0$. Since F is twice continuously differentiable on the box constraint, we have $\|\nabla^2 F(x^k)\| \leq M$ for some non-negative scalar M independent of k. Therefore, $\|H^k\| \leq C_2\Delta_{max} + M$. Letting $\kappa_g = (L_g + C_1 + C_2\Delta_{max} + M)$, we obtain

$$\|\nabla F(x^k + s^k) - \nabla m^k(x^k + s^k)\| \leq \kappa_g\Delta^k.$$

Hence, the first property in Definition 7.1 is verified. The second property can be obtained by using a similar process than the one used in [AH20, Proposition 19.1]. Therefore, the model m^k is fully linear on $\overline{B}_n(x^k; \Delta^k)$. □

It follows from Proposition 7.2 that Algorithm 5 always build fully linear model on $\overline{B}_n(x^k; \Delta^k)$.

In [HR22], Hough and Roberts analyze the convergence properties of a DFTR algorithm for an optimization problem of the form

$$\min_{x \in C} F(x)$$

where $C \subseteq \mathbb{R}^n$ is closed and convex with non-empty interior and $F : \mathbb{R}^n \to \mathbb{R}$. The main algorithm developed in their paper, Algorithm 2.1, follows the same steps than our algorithm. Since a box $[\ell, u]$ with $\ell < u$ is a convex closed set with nonempty interior, the convergence results proved in [HR22] may be applied to Algorithm 5. For completeness, we recall the three assumptions and convergence results of [HR22].

Assumption 7.3. *The objective function F is bounded below and continuously differentiable. Furthermore, the gradient ∇F is Lipschitz continuous with constant $L_{\nabla F} \geq 0$ in $\cup_k \overline{B}_n(x^k; \Delta_{\max})$.*

Assumption 7.4. *There exists $\kappa_H \geq 0$ such that $\|H^k\| \leq \kappa_H$ for all k.*

Assumption 7.5. *There exists a constant $c_1 \in (0, 1)$ such that the computed step s^k satisfies $x^k + s^k \in C$, $\|s^k\| \leq \Delta^k$ and the generalized Cauchy decrease condition:*

$$m^k(x^k) - m^k(x^k + s^k) \geq c_1 \pi_m^k \min\left(\frac{\pi_m^k}{1 + \|H^k\|}, \Delta^k, 1\right).$$

Theorem 7.6. *Suppose Algorithm 5 is applied to the minimization problem (7.5). Suppose Assumptions 7.3, 7.4, and 7.5 hold. Then*

$$\lim_{k \to \infty} \pi_F^k = \lim_{k \to \infty} x^k - \text{Proj}_{[\ell,u]}(x^k - \nabla F(x^k)) = 0.$$

In the next section, we provide details on the different choices made while implementing Algorithm 5.

Implementing the algorithm

Let us begin by specifying the value used for all the parameters involved in the algorithm. Our choices have been influenced by some preliminary numerical results and the values proposed in the literature such as [CGT00, Chapter 6].

In our implementation, the parameters are set to

$$\begin{aligned}
\Delta^0 &= 1 && \text{(initial trust-region radius)},\\
\Delta_s^0 &= 0.5 && \text{(Initial sampling radius)},\\
\Delta_{\max} &= 1 \times 10^3 && \text{(maximal trust-region radius)},\\
\eta_1 &= 0.1 && \text{(parameter for accepting the trial point)},\\
\eta_2 &= 0.9 && \text{(parameter for the trust-region radius update)},\\
\gamma &= 0.5 && \text{(parameter to decrease trust-region radius)},\\
\gamma_{inc} &= 2 && \text{(parameter to increase the trust-region radius)},\\
\epsilon_{stop} &= 1 \times 10^{-5} && \text{(parameter to verify optimality)},\\
\mu &= 1 && \text{(parameter to verify the size of the trust-region radius)}.
\end{aligned}$$

To build an approximation of the Hessian H^k, we compute a generalized simplex Hessian as defined in Definition 5.5. The two matrices of directions S and $\overline{T}$ must be chosen: at every iteration k, the matrices S^k and $\overline{T}^k$ are set to

$$\begin{aligned}
S^k &= \frac{\Delta_s^k}{2}\,\mathrm{Id}_n,\\
\overline{T}^k &= \frac{\Delta_s^k}{2}\,\mathrm{Id}_n\,.
\end{aligned}$$

Multiplying the identity matrix by $\frac{\Delta_s}{2}$ to form S^k and $\overline{T}^k$ guarantees that the sampling radius to build an approximation of the Hessian is equal to Δ_s. Setting S^k and $\overline{T}^k$ in this fashion creates $(n+1)(n+2)/2$ distinct sample points poised for quadratic interpolation. This implies that the generalized simplex Hessian is equal to the Hessian of the quadratic interpolation function. Moreover, this choice guarantees that the generalized simplex Hessian is a symmetric matrix (Proposition 5.25). Clearly, other matrices S^k and $\overline{T}^k$ could be used and S^k does not necessarily need to be equal to $\overline{T}^k$ nor to be always a multiple of the identity matrix for every iteration k. More details on how to choose S and $\overline{T}$ so that the the resulting set of sample points is poised for quadratic interpolation have been provided in Section 5.4.

To build an approximation of the gradient at iteration k, the gradient of the quadratic interpolation function built using the same $(n+1)(n+2)/2$ sample points used to obtain H^k is simply computed. Therefore, building a model m^k at any iteration $k \in \{0, 1, \dots\}$ requires $(n+1)(n+2)/2$ function evaluations.

Note that the sample points utilized to build g^k and H^k may be outside of the box constraint. This is allowed in our implementation.

To solve the trust-region sub-problem, the Matlab command quadprog with the algorithm trust-region-reflective is used. In theory, this method satisfies Assumption 7.5.

Reducing numerical errors

We next discuss two strategies that decrease the risk of numerical errors. When the sampling radius Δ_s is sufficiently small, numerical errors occur while computing g^k and H^k and this can cause g^k and H^k to be very bad approximations of the gradient and Hessian at x^k. To avoid this situation, a minimal sampling radius $\Delta_{s\,\min}$ is defined. Every time the sampling radius Δ_s^k is updated in Algorithm 5 (this may happen in Step 1 or Step 5), the rule implemented is the following:

$$\Delta_s^k \leftarrow \max\{\Delta_{s\,\min}, \Delta_s^k\}. \tag{7.10}$$

In our implementation, $\Delta_{s\,\min} = 1 \times 10^{-4}$. Numerical errors can occur at relatively large values of sampling radius Δ_s. The following example illustrates this situation. It motivates our choice to set $\Delta_{s\,\min} = 1 \times 10^{-4}$.

Example 7.7. Let

$$F(x) = f_1(x) \cdot f_1(x) = \left(0.5x^\top \begin{bmatrix} 10 & 9 \\ 9 & 10 \end{bmatrix} x + \begin{bmatrix} 10 & 9 \end{bmatrix} x\right)^2$$

Let $x^0 = \begin{bmatrix} 5 & 5 \end{bmatrix}^\top$. Set $S = \overline{T} = \frac{h}{2}\,\mathrm{Id}_2$, where $h > 0$. Note that the sampling radius is $\Delta_s = h$. Table 7.1 presents the relative error of $\nabla_s^2 F(x^0; S, \overline{T})$, denoted by $\mathrm{RE}(\nabla_s^2 F(x^0; S, \overline{T}))$.

Table 7.1: Relative error of $\nabla_s^2 F(x^0; S, \overline{T})$ for different values of Δ_s

Δ_s	$\mathrm{RE}(\nabla_s^2 F(x^0; S, \overline{T}))$
5e-01	4.7e-02
1e-01	9.3e-03
1e-02	9.2e-04
1e-03	9.2e-05
1e-04	8.8e-06
1e-05	4.5e-05

We see that numerical errors occur at a value of Δ_s between 1×10^{-4} and 1×10^{-5}.

A maximal sampling radius $\Delta_{s\,\max}$ is also defined to ensure that the sampling radius Δ_s does not get excessively large when the trust-region radius is large. The parameter is set to $\Delta_{s\,\max} = 0.5$. Therefore, after checking Equation (7.10), the following update on Δ_s^k is done:

$$\Delta_s^k \leftarrow \min\{\Delta_{s\,\max}, \Delta_s^k\}.$$

In Step 3 in the computation of ρ^k, the denominator satisfies

$$\begin{aligned} m^k(x^k) - m^k(x^k + s^k) &= F(x^k) - \left(F(x^k) + (g^k)^\top s^k + \frac{1}{2}(s^k)^\top H^k s^k \right) \\ &= (g^k)^\top s^k + \frac{1}{2}(s^k)^\top H^k s^k. \end{aligned} \tag{7.11}$$

As mentioned in [CGT00, Section 17.4], to reduce numerical errors, we use (7.11) to compute this value.

To compute the ratio ρ^k, we again follow the advice given in [CGT00, Section 17.4.2] and proceed in the following way. Let $\epsilon = 10^4 \cdot \epsilon_M$ where ϵ_M is the relative machine precision. Let $\delta^k = \epsilon \max(1, |F(x^k)|)$. Define

$$\delta F^k = F(x^k + s^k) - F(x^k) - \delta^k,$$

$$\delta m^k = m^k(x^k + s^k) - m^k(x^k) - \delta^k = F(x^k) + (g^k)^\top s^k + \frac{1}{2}(s^k)^\top H^k s^k - \delta^k.$$

Then

$$\rho^k \in \begin{cases} 1, & \text{if} \quad |\delta F^k| < \epsilon \quad \text{and} \quad |F(x^k)| > \epsilon, \\ \dfrac{\delta F^k}{\delta m^k}, & \text{otherwise.} \end{cases}$$

7.1.2 Numerical experiments

Algorithm 5 is implemented on Matlab 2021a using both the calculus-based and non-calculus approach to approximating gradients and Hessians. These implementations are tested on a suite of test problems detailed below. The Matlab function fmincon is also tested on each problem for all experiments mentioned below. This is done as a validation step, to demonstrate that Algorithm 5 is implemented correctly (and reasonably competitive against currently used methods). To conclude this section, we present details on the different situations tested for the product rule and the quotient rule.

We begin by providing details about *data profiles*.

Data profiles

To do the comparisons in this section, data profiles are built [MW09]. The convergence test for the data profiles is

$$f(x) \leq f_L + \tau \left(f(x^0) - f_L \right) \tag{7.12}$$

where $\tau > 0$ is a tolerance parameter and f_L is the best known minimum value for each problem p in $\mathcal{P}$. Let $t_{p,s}$ be the number of function evaluations require to satisfy (7.12) on a problem $p \in \mathcal{P}$ using a solver $s \in \mathcal{S}$ given a maximum number of function evaluations μ_f. The parameter μ_f is set to $\mu_f = 1000(n_p)$ where n_p is the dimension of problem $p \in \mathcal{P}$. Note that μ_f is set to ∞ if Equation (7.12) is not satisfied after μ_f function evaluations. The data profile of a solver $s \in \mathcal{S}$ is defined by

$$d_s(\alpha) = \frac{1}{|\mathcal{P}|} \text{size} \left\{ p \in \mathcal{P} : \frac{t_{p,s}}{n_p + 1} \leq \alpha \right\}.$$

Three different values of τ will be used to build data profiles: $10^{-1}, 10^{-3}$, and 10^{-5}.

Numerical experiment with the product rule

In these numerical experiments, the composite objective function F has the form $F = f_1 \cdot f_2$, where $f_i : \mathbb{R}^n \to \mathbb{R}, i \in \{1, 2\}$. Three different situations are considered:

- f_1 and f_2 are linear functions,
- f_1 is a linear function and f_2 is a quadratic function,

– f_1 and f_2 are quadratic functions.

Recall that a linear function $L : \mathbb{R}^n \to \mathbb{R}$ has the form

$$L(x) = b^\top x + c$$

where b $\in \mathbb{R}^n$, and $c \in \mathbb{R}$. A quadratic function $Q : \mathbb{R}^n \to \mathbb{R}$ has the form

$$Q(x) = \frac{1}{2} x^\top A x + b^\top x + c,$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix. At the beginning of each experiment, the seed of the random number generator is fixed to 54321 using the command rng. The dimension of the space is between 1 and 30 and it is generated with the command randi. All integers will be generated with randi. In each experiment, the coefficients involved in f_1 and f_2 are integers between -10 and 10 inclusively. Each component of the starting point x^0 is an integer between -5 and 5 inclusively. The box constraint is built in the following way:

$$\ell_i = x_i^0 - 1 \quad \text{for all } i \in \{1, \ldots, n\}, \tag{7.13}$$

$$u_i = x_i^0 + 1 \quad \text{for all } i \in \{1, \ldots, n\}. \tag{7.14}$$

This creates a randomly generated test problem of the form (7.5). This problem is solved via Algorithm 5 using the starting point x^0 and using both versions: the calculus-based approach and the non-calculus approach. Each of the three situations listed above is repeated 100 times. Each time, all the parameters are generated again: the dimension n, the starting point x^0, the box constraint $[\ell, u]$, and the functions f_1, f_2.

Numerical experiment with the the quotient rule: easy case

In these experiments, the composite objective function takes the form $F = \frac{f_1}{f_2}$, where f_i is either a linear function or a quadratic function, for $i \in \{1, 2\}$. We begin by building the function f_2 such that his real roots, if any, are relatively far from the box the constraint. To do so, each component of the starting point x^0 is taken to be an integer between 1 and 100 inclusively. The coefficients in f_2 are taken to be integers between 1 and 10 inclusively. When f_2 is a quadratic function, note that having all entries in the matrix A to be positive does not necessary imply that A is positive semi-definite. The box constraint is built in the same fashion as the previous experiment. Thus, the bounds are

$$\ell_i = x_i^0 - 1 \geq 0 \quad \text{for all } i \in \{1, \ldots, n\},$$
$$u_i = x_i^0 + 1 \quad \text{for all } i \in \{1, \ldots, n\}.$$

Note that if $r \in \mathbb{R}^n$ is a root of f_2, then r must have at least one negative component. Since $f_2(\ell) > 0$ and f_2 is an increasing function, there is no root of f_2 in the box constraint. Hence F is twice continuously-differentiable on the box constraint. The four following situations are tested:

- f_1 and f_2 are linear functions,
- f_1 is a linear function and f_2 is quadratic function,
- f_1 is a quadratic function and f_2 is a linear function,
- f_1 and f_2 are quadratic functions.

Each situation is repeated 100 times.

Numerical experiment with the the quotient rule: hard case

Last, the quotient rule is tested again, but this time, the function f_2 is built so that there is a root of f_2 near (but not within) the box constraint. The differences with the previous quotient experiments are the following. Each component of the starting point are taken to be integers between -5 and 5 inclusively. Let $f_2 = \widetilde{f_2} + c$ where $c \in \mathbb{R}$. The coefficients in $\widetilde{f_2}$ are taken to be between -10 and 10 inclusively. Before generating the constant c in f_2, the minimum value on the box constraint of $\widetilde{f_2}$, say $\widetilde{f_2}^*$, is found. Then c is set to $c = 0.001 - \widetilde{f_2}^*$. Hence f_2 is built such that the minimum value of f_2 on the box constraint is 0.001 and $f_2(x) \geq 0.001$ for all x in the box constraint.

7.1.3 Results

The data profiles for each of the three experiments are now presented. In the data profiles, the vertical axis represents the portion of problem solved and the horizontal axis represents the ratio of function evaluations α. We begin by presenting the data profiles for the product rule. The following 9 data profiles (3 data profiles per situation) are obtained (Figures 7.1, 7.2, 7.3).

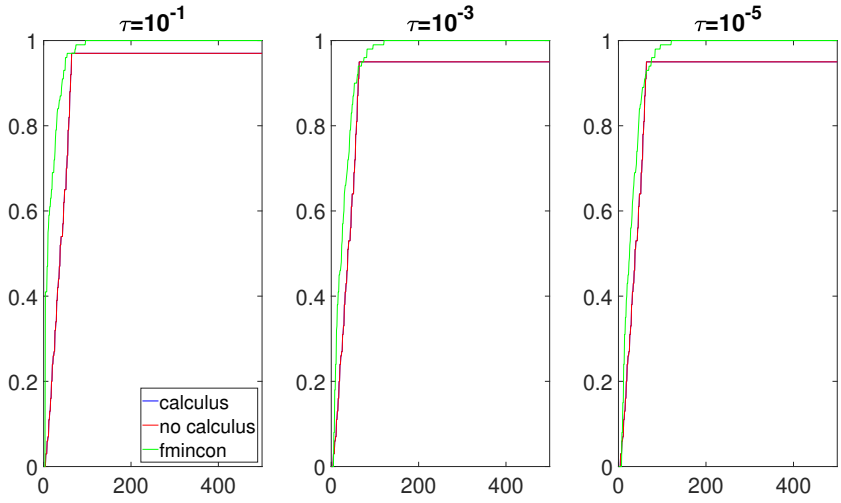


Figure 7.1: Data profiles when $F = f_1 \cdot f_2$, and f_1, f_2 are linear functions

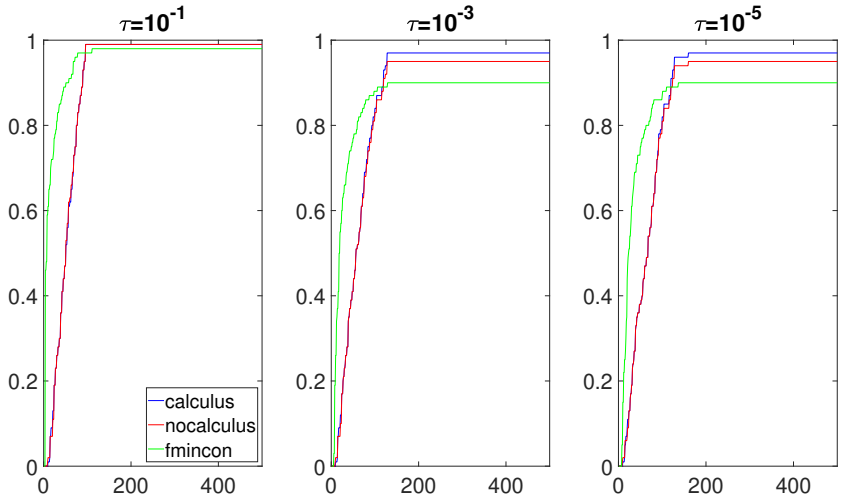


Figure 7.2: Data profiles when $F = f_1 \cdot f_2$, f_1 is a quadratic function, and f_2 is a linear function

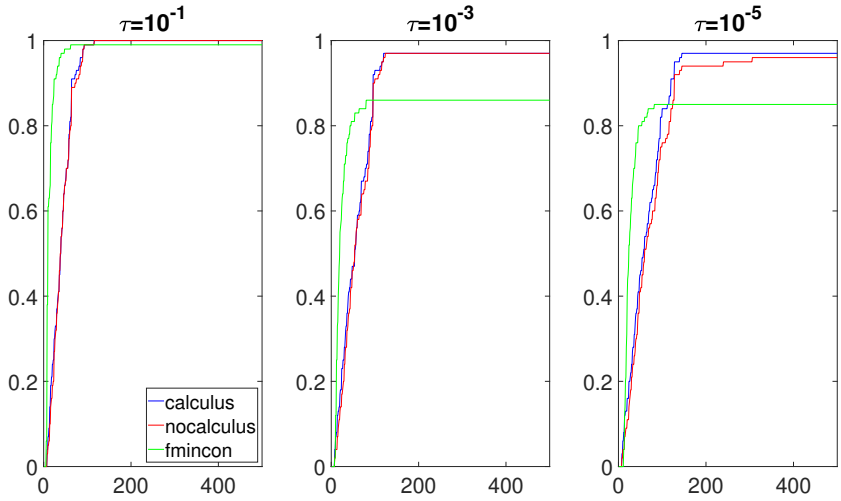


Figure 7.3: Data profiles when $F = f_1 \cdot f_2$, and f_1, f_2 are quadratic functions

Next, we present the data profiles for the quotient rule as described in Section 7.1.2. The following 12 data profiles are obtained (Figure 7.4, Figure 7.5, Figure 7.6, and Figure 7.7).

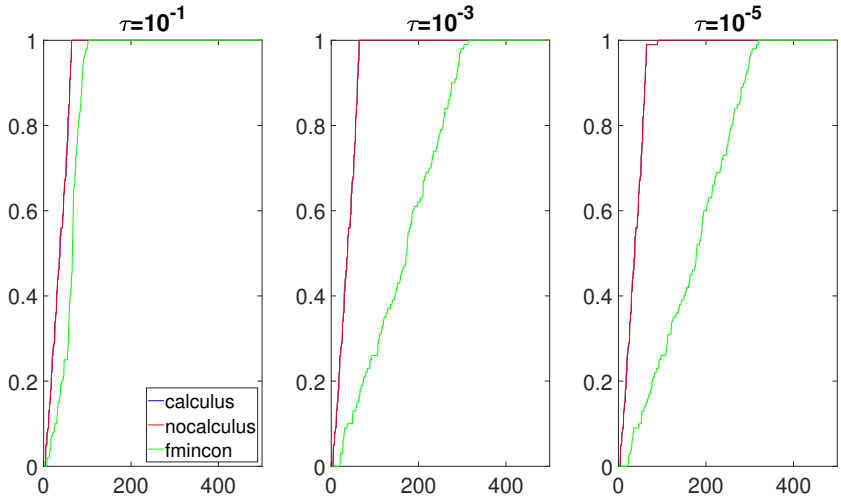


Figure 7.4: Data profiles when $F = \frac{f_1}{f_2}$ and f_1, f_2 are linear functions

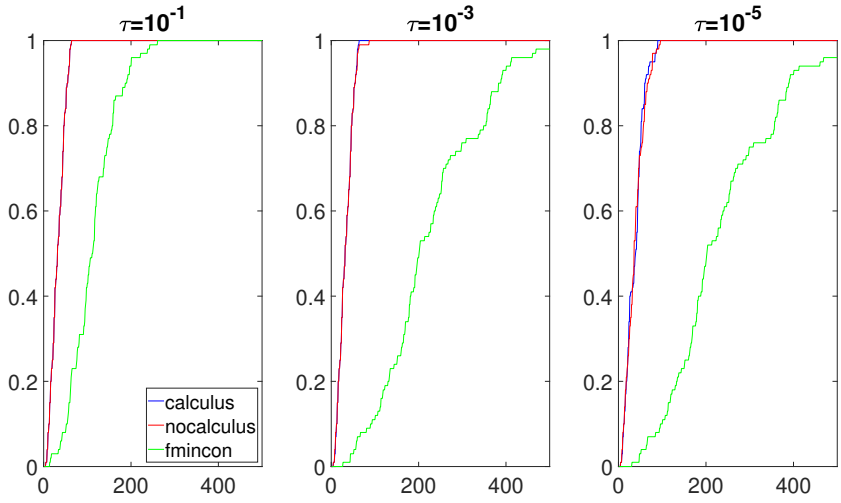


Figure 7.5: Data profiles when $F = \frac{f_1}{f_2}$, f_1 is a linear function and f_2 is a quadratic function

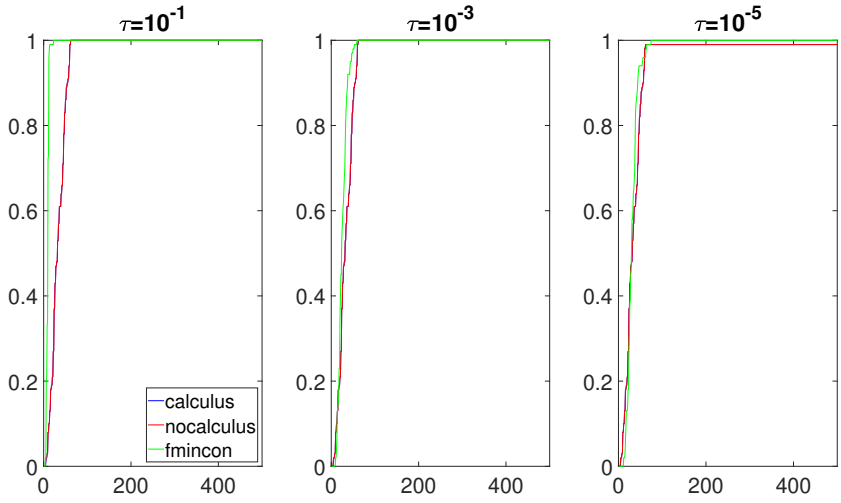


Figure 7.6: Data profiles when $F = \frac{f_1}{f_2}$, f_1 is a quadratic function, and f_2 is a linear function

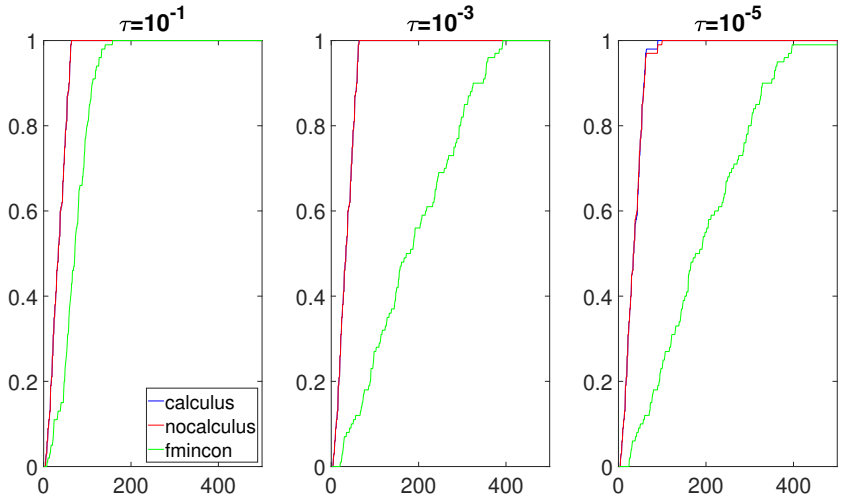


Figure 7.7: Data profiles when $F = \frac{f_1}{f_2}$ and f_1, f_2 are quadratic functions

Last, we present the data profiles for the quotient rule as described in 7.1.2. The following 12 data profiles are obtained (Figure 7.8, Figure 7.9, Figure 7.10, and Figure 7.11).

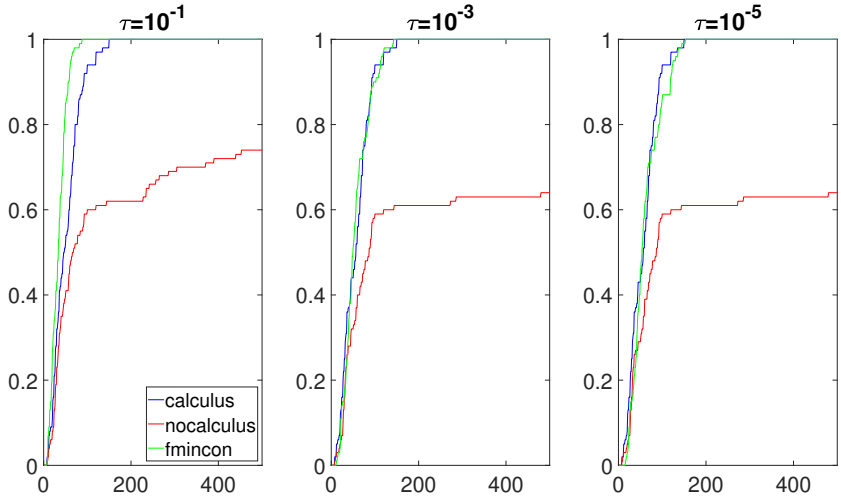


Figure 7.8: Data profiles when $F = \frac{f_1}{f_2}$ and f_1, f_2 are linear functions

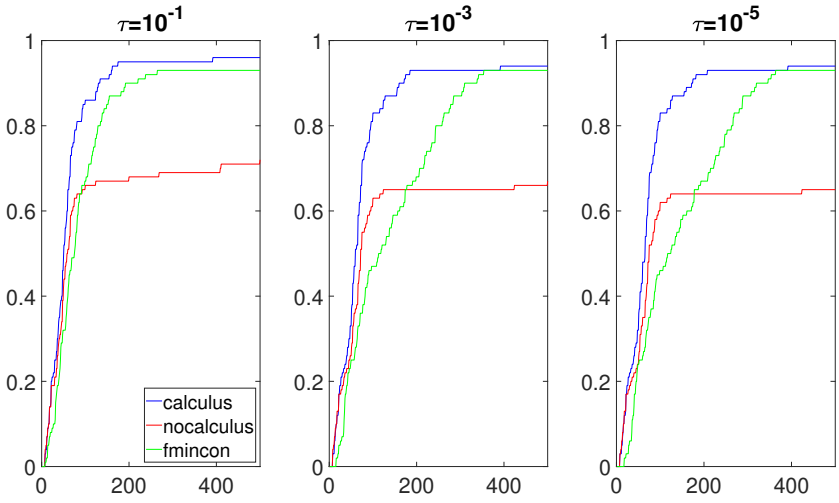


Figure 7.9: Data profiles when $F = \frac{f_1}{f_2}$, f_1 is a linear function, and f_2 is a quadratic function

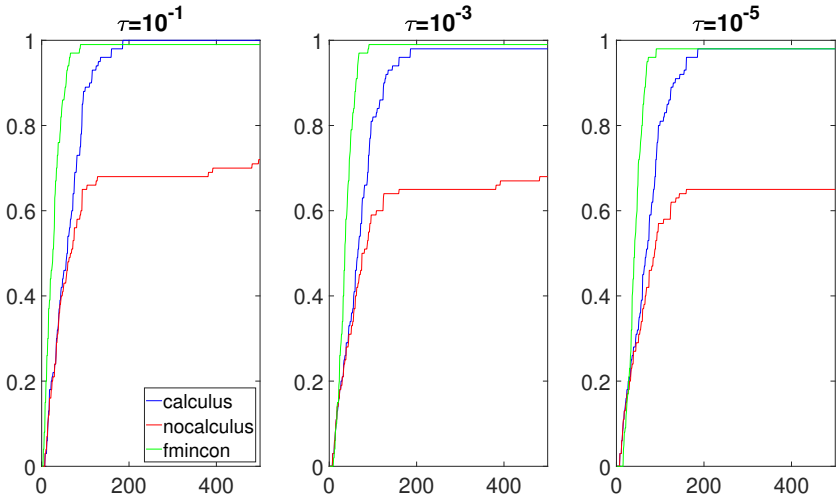


Figure 7.10: Data profiles when $F = \frac{f_1}{f_2}$, f_1 is a quadratic function, and f_2 is a linear function

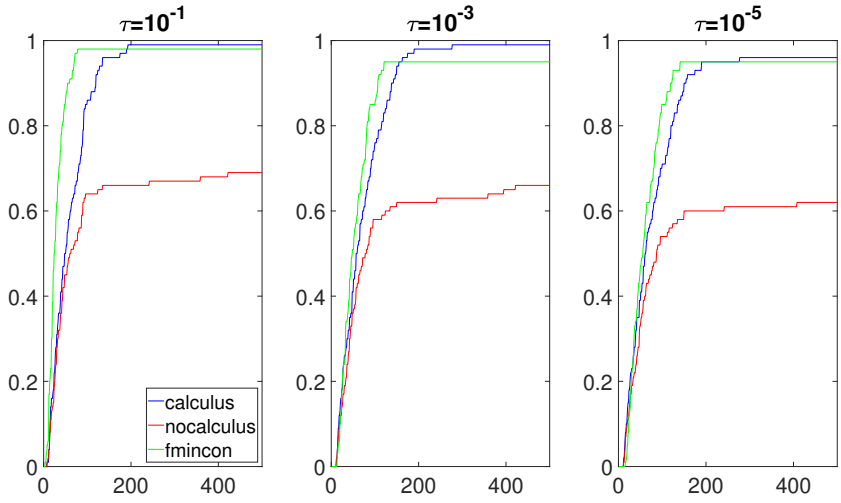


Figure 7.11: Data profiles when $F = \frac{f_1}{f_2}$, and f_1, f_2 are quadratic functions

7.1.4 Discussion

The data profiles presented in Section 7.1.3 are now analyzed. First, the data profiles related to the product rule are scrutinized.

Numerical experiment with the product rule

Figure 7.1 agrees with the theory: the calculus-based approach provides the exact same results as the non-calculus approach. Indeed, when both f_1 and f_2 are linear, theory states that both approaches build models with an exact gradient and an exact Hessian. As such, both method should behave identically.

Figures 7.2 and 7.3 show that the calculus-based approach is slightly more efficient and robust. Note that a calculus-based approach is building model functions m^k with an exact gradient g^k and an exact Hessian H^k. This is not the case with a non-calculus approach when F is a cubic function or quartic function. However, the accuracy of the approximate gradients and Hessians in the models m^k is sufficiently good not to make a significant difference with the exact models built with a calculus-based approach. We also observe that Algorithm 5 is more robust than fmincon for these two situations which supports that our algorithm is competitive with currently used solvers.

Although there are no drastic differences between the calculus-based approach and the non-calculus approach, we conclude that the calculus-based approach is better or at least as good as the non-calculus approach on these 3 situations.

Numerical experiment with the quotient rule: the easy case

We now analyze Figures 7.4, 7.5, 7.6, and 7.7. The performances of both approaches are almost identical. The calculus-based approach provides slightly better results. Compared to fmincon, Algorithm 5 is competitive and even outperforms fmincon when the numerator of F is a linear function (Figure 7.4). We note that having no roots of f_2 near the box constraint implies that the value of the composite objective function F does not change drastically on the box constraint. In other words, the Lipschitz constant of F on the box constraint is not a huge number. This helps to obtain accurate gradient g^k when using the non-calculus approach. As such, in these four situations, the accuracy of the approximate gradients and Hessians computed with the non-calculus approach is similar to the calculus-based approach. Hence, the performance of the non-calculus approach is almost as good as the calculus-based approach.

Numerical experiment with the quotient rule: the hard case

The most interesting results are obtained when the composite objective function F has the form $F = \frac{f_1}{f_2}$, and f_2 has a real root near the box constraint $[\ell, u]$. Figures 7.8, 7.9, 7.10, and 7.11 clearly show that the calculus-based approach is significantly more efficient and robust than the non-calculus approach for these four situations. Note that the composite objective function F is twice continuously-differentiable on the box constraint, but F is not twice continuously-differentiable on the expanded box constraint (7.6). Using a non-calculus approach, it could be the case that g^k, or H^k, is undefined if one of the sample point lands exactly on a real root of f_2. This is very unlikely and it did not happen during the experiment. Using a calculus-based approach, this issue cannot occur as f_1 and f_2 are twice continuously-differentiable everywhere. This is clearly an advantage of the calculus-based approach. If a sample point is near a root of f_2, the value of F at this point could be a very large number. This could make the accuracy of g^k and H^k very poor when using the non-calculus approach.

Note that the Lipschitz constants of ∇F and $\nabla^2 F$ on the box constraint are very large numbers. Hence, the error bounds associated to the approxi-

mate gradient g^k (Theorem 3 in [CSV08a]) and the approximate Hessian H^k (Theorem 5.13) are very large numbers. This tells us that it is possible to obtain very bad approximations for g^k and H^k when using the non-calculus approach.

To see how the non-calculus approach can fail to provide accurate approximations of the gradient and Hessian, we present Example 7.8.

Example 7.8. Let

$$F(x) = \left(\frac{f_1}{f_2}\right)(x) = \frac{10x + 10}{-10x^2 + 10x + 20.0001}.$$

Suppose $x^k = -1$. The derivative of F at x^k is $F'(x^k) = 1e + 05$ and the second-order derivative of F at $x^k = -1$ is $F''(x) = -6e + 10$. Table 7.2 provides the value, and the relative error associated to the approximations g^k and H^k obtained using the non-calculus approach and $S = \overline{T} = h$ for different values of h. The relative error of an approximation technique is denoted by RE$(\cdot)$ in the following table.

Table 7.2: An example where the non-calculus approach is inaccurate

h	g^k	**RE**(g^k)	H^k	**RE**(H^k)
5e-01	1.1e+00	9.9e-01	-1.2e+00	1.0e+00
1e-01	5.1e+00	9.9e-01	-3.3e+01	1.0e+00
1e-02	5.0e+01	9.9e-01	-3.3e+03	1.0e+00
1e-03	4.9e+02	9.9e01	-3.3e+05	1.0e+00
1e-04	4.8e+03	9.5e-01	-3.1e+07	9.9e-01
1e-05	3.5e+04	6.4e-01	-2.1e+09	9.6e-01
1e-06	9.1e+04	8.6e-02	-2.8e+10	5.1e-01
1e-07	9.9e+04	1.6e-03	-5.4e+10	8.4e-02
1e-08	9.9e+04	1.7e-05	-5.9e+10	8.9e-03
1e-09	1.0e+05	3.3e-07	-5.9e+10	1.1e-03
1e-10	1.0e+05	1.8e-09	-5.9e+10	8.9e-05
1e-11	1.0e+05	0.0e+00	-6.0e+10	2.9e-06
1e-12	-1.0e+05	2.0e+00	1.7e+13	2.9e+02

We observe that numerical errors occur at $h = 1 \times 10^{-11}$ for the approximate gradient g^k and at $h = 1 \times 10^{-12}$ for the approximate Hessian H^k. Note that in this experiment, the sampling radius is $\Delta_s = 2h$. Assuming numerical errors do not occur, the theory guarantees that the relative error

will be of order Δ_s^2 for g^k and of order Δ_s for H^k when Δ_s is sufficiently small. However, in this experiment, numerical errors occur before attaining the expected accuracy for g^k and H^k. Therefore, on this experiment, g^k and H^k are inaccurate.

To recapitulate, the numerical results obtained in Section 7.1.3 clearly suggest that a calculus-based approach should be used. In particular, in all cases tested, the calculus-based approach was better or as good as the non-calculus approach. Since a calculus-based approach is not more difficult to implement than a non-calculus approach, the former approach seems to be the best approach to implement and use whenever a composite objective function is optimized.

7.2 A generalized pattern search algorithm using optimal CFOPB

A generalized pattern search (GPS) method can be classified as a directional direct-search method as discussed in Section 2.3. A pattern search algorithm was introduced by Torczon in [Tor97]. A few years later, a generalized pattern-search framework was proposed in [AD04]. We begin by summarizing this framework and provide details about the convergence results available in the literature.

A GPS method can be divided in two phases: the search step and the poll step. The search step consists of evaluating the objective function at a finite number of points. Many strategies can be used to select these points. For instance, a heuristic algorithm could be used [CSV09b]. Another strategy is to reuse the existent surrogate models (if any) to explore the objective function in a new area. If a point is found such that the function value at this point is strictly less than the function value at the incumbent solution, then the search step is declared successful and the incumbent solution is updated. The search step is not necessary in the convergence analysis developed for the method [CSV09b].

When the search step is unsuccessful, a poll step is executed. The poll step is a local search around the incumbent solution x^k. It explores a set of points defined as

$$P^k = \{x^k + \alpha^k d : d \in \mathbb{P}^k_{n,s}\},$$

where α^k is the step size parameter at iteration k, $d \in \mathbb{R}^n$ is a direction in the positive spanning set $\mathbb{P}^k_{n,s}$ of $\mathbb{R}^n$. In this Chapter, note that the superscript k in $\mathbb{P}^k_{n,s}$ is used to refer to the iteration counter, and not the dimension of the space as in Chapter 6. If $f(x^k + \alpha^k d) < f(x^k)$, for some point $x^k + \alpha^k d$ in P^k, then the poll step is declared successful. The previous inequality is usually referred as a *simple decrease condition*. In the case where the poll step fails to find a point in P^k such that the value of the objective function is lower than $f(x^k)$, then the poll step is declared unsuccessful. The step size parameter is then decreased and the incumbent solution is unchanged. When the objective function is continuously differentiable and x^k is not a stationary point, we know that the poll step will be successful after finitely many decreases of the step size parameter α^k. This follows from the property of positive spanning sets that guarantees one of the directions in $\mathbb{P}^k_{n,s}$ is a descent direction (Proposition 6.12). This is the main reason for using positive spanning sets in the poll step.

Different type of polling strategies may be used. A *complete poll strategy*

evaluates all poll points at each iteration. Then the poll points with the lowest function value is compared to the incumbent solution x^k. In contrast, an *opportunistic poll strategy* terminates the poll step as soon as a point in the polling set P^k that improves the incumbent solution is evaluated. An *ordered opportunistic poll strategy* orders the poll points before evaluating the objective function.

In the case where the search step or the poll step is successful, the step size can be unchanged or increased.

At each iteration k, a different positive spanning set of $\mathbb{R}^n$ can be used. If a simple decrease condition is used, then this number must be finite for the convergence results to hold [CSV09b]. When a *sufficient decrease condition* is used, this requirement can be relaxed. When a sufficient decrease solution is employed in a GPS method, a point t is accepted and set to be the incumbent solution only if

$$f(t) < f(x^k) - \beta(\alpha^k)$$

where β is a *forcing function* that can be set to $\beta(x) = x^2$ for example [CSV09b, Section 7.7]. When using a forcing function, the convergence results hold with an infinite number of positive spanning sets can as long as the cosine measure (defined in Definition 6.10) of the set of positive spanning sets is uniformly bounded away from 0.

Convergence results for a GPS method have been developed for different categories of optimization problems over the years. In [Tor97], Torczon analyzed GPS methods for unconstrained smooth optimization problems. In [LT99], the convergence theory is extended to bound constrained optimization problem. In [LT00], convergence results are developed for optimization problems with a finite number of linear constraints. The convergence theory for this type of problems is extended and clarified in [AD04]. The poll points are required to lie on the *mesh* in the convergence analysis developed. The mesh can be thought as an enumerable discretization of the space of variables. Another major results of the previous paper is to show that the step size adjustment parameter can take irrational values if a sufficient decrease condition is used rather than a simple decrease condition (in this case, the step size adjustment parameter needs to be a rational number). A second-order convergence analysis for smooth problems is proposed in [Abr05]. When the optimization problem is non-smooth, GPS methods may converge to a non-stationary point [CSV09a, Section 7.4]. Extra assumptions are needed to develop convergence results similar to the smooth case. This weakness of the GPS method has motivated a further extension of this type of method: the *mesh adaptive direct search* method [AA06, AD06].

The main goal of this section is to implement a GPS algorithm and use optimal CFOPB in the poll step. Three numerical experiments are conducted. These experiments verify if the following three topics have an impact on the performance of the GPS algorithm implemented: the cardinality of the set $\mathcal{D}$ of positive spanning sets $\mathbb{P}_{n,s}^k$, the value of the cosine measure of the positive spanning sets employed, and the size s of the positive spanning sets.

7.2.1 The algorithm

The GPS algorithm implemented follows the pseudo-code in [CSV09b, Algorithm 7.2] imposing sufficient decrease as discussed in [CSV09b, Section 7.7]. The pseudo-code is designed for an unconstrained smooth optimization problem. In accordance with the results in [CSV09b, Section 7.7], sufficient decrease is used in both the search step and poll step of the pseudo-code.

Algorithm 6: GPS algorithm pseudo-code

0. Initialization

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, choose

$x^0 \in \mathbb{R}^n$:	an initial starting point.
$\mathcal{D}$:	a finite set of positive spanning sets.
α^0 :	the initial step size parameter. Positive real number.
ξ :	step size adjustment. Rational number in $(0, 1)$.

For $k = 0, 1, 2, \ldots$

1. Search step

Try to find a point x with $f(x) < f(x^k) - (\alpha^k)^2$ by evaluating f at a finite number of points. If such a point is found, set $x^{k+1} = x$. Declare the iteration successful and skip the poll step.

2. Poll step

Choose a positive spanning set $\mathbb{P}^k_{n,s}$ from the set $\mathcal{D}$. If a poll point $x^k + \alpha^k d$ is found such that

$$f(x^k + \alpha^k d) < f(x^k) - (\alpha^k)^2, \tag{7.15}$$

then set $x^{k+1} = x^k + a^k d$. Declare the iteration successful. Otherwise, declare the iteration unsuccessful, and set $x^{k+1} = x^k$.

3. Step parameter update

If the iteration is successful, set $\alpha^{k+1} = \xi^{-1}\alpha^k$. Otherwise (iteration is unsuccessful), decrease the step size parameter: $\alpha^{k+1} = \xi\alpha^k$.

Increment k by one and go to **Step 1**.

End For

The previous pseudo-code does not specify a stopping criterion. In practice, a common stopping criterion is to stop when the step size parameter a^k is less than a certain chosen tolerance α_{stop}.

The main first-order convergence result in the smooth case when a sufficient decrease condition is used is now recalled. To obtain this result, the following assumptions are used in [CSV09b].

(i) The level set $L(x^0) = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$ is compact.

(ii) There exists a value $\alpha > 0$ such that $\alpha^k > \alpha$ for all k.

(iii) Let $c_1, c_2 > 0$ be some fixed positive constants. The positive spanning

set $\mathbb{P}_{n,s}^k$ of $\mathbb{R}^n$ used at iteration k is chosen from the set

$$\left\{\widehat{\mathbb{P}}_{n,s} : \mathrm{cm}(\widehat{\mathbb{P}}_{n,s}) > c_1, \|\widehat{d}\| \leq c_2, \widehat{d} \in \widehat{\mathbb{P}}_{n,s}\right\}.$$

(iv) The function f is smooth in an open domain containing $L(x^0)$.

Theorem 7.9. *[CSV09b, Theorem 7.4] Let Assumptions (i), (ii), (iii) and (iv) hold. Then the sequence $\{x^k\}$ has a limit point $x^* \in \mathbb{R}^n$ for which*

$$\nabla f(x^*) = \mathbf{0}.$$

7.2.2 Numerical experiments

Algorithm 6 is implemented in Matlab 2021a. The values chosen for the parameters involved in the algorithm have been influenced by the values used in [AH17, Chapter 7].

In our implementation, the parameters are set to

$$\begin{aligned} \alpha^0 &= 1 && \text{(initial step size)}, \\ \xi &= \frac{1}{2} && \text{(step size adjustment)}. \end{aligned}$$

All experiments are tested on the Moré Garbow Hillstrom test set provided in [MGH81]. This set of problems contains 35 unconstrained optimization problems. The functions are generally smooth and the dimensions vary between 2 and 30 inclusively. Since the dimension n_p and the number of functions q_p involved in some of the problems p need to be decided by the user, Table 7.3 provides the values n_p and q_p chosen. The minimum value f_{L_p} for each problem p is also provided (with an accuracy of two decimal places). In each problem p, the objective function f takes the form $f(x) = \sum_{i=1}^{q_p}(f_i(x))^2$, where each $f_i(x)$ is provided in [MGH81].

Table 7.3: The Moré Garbow Hillstrom test set

Problem p	n_p	q_p	f_{L_p}
1. Rosenbrock	2	2	0.00e+00
2. Freudenstein	2	2	4.89e+01
3. PowellBS	2	2	0.00e+00
4. BrownBS	2	3	0.00e+00
5. Beale	2	3	0.00e+00
6. Jenrich	2	4	1.24e+02
7. Helical	3	3	0.00e+00
8. Bard	3	15	8.21e-03
9. Gaussian	3	15	1.12e-08
10. Meyer	3	16	8.79e+01
11. Gulf	3	20	0.00e+00
12. Box3D	3	3	0.00e+00
13. PowellS	4	4	0.00e+00
14. Wood	4	6	0.00e+00
15. Kowalik	4	11	3.07e-04
16. Brown	4	4	8.58e+04
17. Osborne1	5	33	5.46e-05
18. Biggs	6	6	0.00e+00
19. Osborne2	11	65	4.01e-02
20. Watson	12	31	4.72e-10
21. RosenbrockExt	4	4	0.00e+00
22. PowellExt	8	8	0.00e+00
23. Penalty1	10	11	7.08e-05
24. Penalty2	10	20	2.93e-04
25. VariablyDim	7	9	0.00e+00
26. Trigonometric	7	7	0.00e+00
27. BrownAlm	9	9	0.00e+00
28. DiscreteBnd	5	5	0.00e+00
29. DiscreteInt	3	3	0.00e+00
30. BroydenTri	5	5	0.00e+00
31. BroydenBan	20	20	0.00e+00
32. LinearFR	15	18	3.00e+00
33. LinearR1	30	30	7.13e+00
34. LinearR1W0	25	25	7.38e+00
35. Chebyquad	2	2	0.00e+00

To create the set of positive spanning sets $\mathcal{D}$, an initial positive spanning set is generated. This initial positive spanning set will be referred as the *primitive*. The size s of the primitive is chosen randomly using the command randi. All directions in the primitive are created as unit vectors. A cardinality for the set $\mathcal{D}$ is chosen. Then all positive spanning sets in the set $\mathcal{D}$ are created by applying a random rotation matrix to the primitive. At iteration 0, one positive spanning set in the set $\mathcal{D}$ is chosen randomly. The same positive spanning set is kept whenever the iteration is successful. If the iteration k is unsuccessful, then the positive spanning set used at iteration $k+1$ is updated, and chosen randomly from the set $\mathcal{D}$. The poll strategy utilized in all experiments is complete.

To judge the performance of the algorithms, data profiles are built (see Section 7.1.2 for details on data profiles). The initial starting points and minimum values provided f_{L_p} in [MGH81] are used. The maximum number of function evaluations μ_f on a problem p is set to

$$\mu_f = 10000 n_p.$$

Three different values of tolerance parameter τ are used: $10^{-1}, 10^{-3}$, and 10^{-5}.

We now provide details on the numerical experiments conducted. These experiments are designed to test three concepts. At the beginning of each problem p in all experiments, the seed is set to $1234567 + p$. In all experiments, no search step is utilized.

Numerical experiment about the cardinality of the set $\mathcal{D}$

In this experiment, we verify if the cardinality of the set $\mathcal{D}$ containing the positive spanning sets has an impact on the performance of Algorithm 6. The primitive is taken to be an optimal CFOPB. Four versions are tested.

- CARD1: the cardinality of $\mathcal{D}$ is 1.
- CARD10: the cardinality of $\mathcal{D}$ is 10.
- CARD100: the cardinality of $\mathcal{D}$ is 100.
- CARD1000: the the cardinality of $\mathcal{D}$ is 1000.

Numerical experiment about the cosine measure

In this experiment, we test three versions of the algorithm to verify if using optimal CFOPBs increases the performance of Algorithm 6. The set $\mathcal{D}$ has cardinality equal to 100. The three versions tested are now described.

– OCFO: the primitive is an optimal CFOPB.

– CAN: the primitive is a canonical positive basis (Definition 6.61).

– RAND: this version creates the primitive by generating a canonical positive basis and multiplying it by a random invertible matrix. Then the vectors are normalized. A proof that this process creates a positive basis can be found in [AH17, Lemma 6.3]. The invertible matrix is created using the command randi and a range of $[-10000, 10000]$ for each entry in the matrix.

Since a canonical positive basis in $\mathbb{R}^n$ of size $2n$ is an optimal CFOPB, the size s of the primitive for each problem in this experiment is chosen randomly to be an integer in $[n+1, 2n-1]$.

Numerical experiment about the size s of the positive bases in $\mathcal{D}$

In this experiment, we verify if the size s of the primitive has an impact on the performance of Algorithm 6. The primitive is chosen to be an optimal CFOPB. The cardinality of the set $\mathcal{D}$ is set to 100. The three versions tested are described next.

– MIN: the primitive has size $s = n + 1$.

– INT: the primitive has size $s = n + 1 + \left\lceil \frac{n-1}{2} \right\rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function.

– MAX: the primitive has size $s = 2n$.

Since there are no intermediate sizes when $n = 2$, the problems $p \in \{1, 2, 3, 4, 5, 6, 35\}$ are removed from the test set in this experiment.

The data profiles obtained for each experiment are presented next.

7.2.3 Results

Figure 7.12 presents the three data profiles obtained while verifying if one of the four values tested for the cardinality of the set $\mathcal{D}$ provides better results than the others.

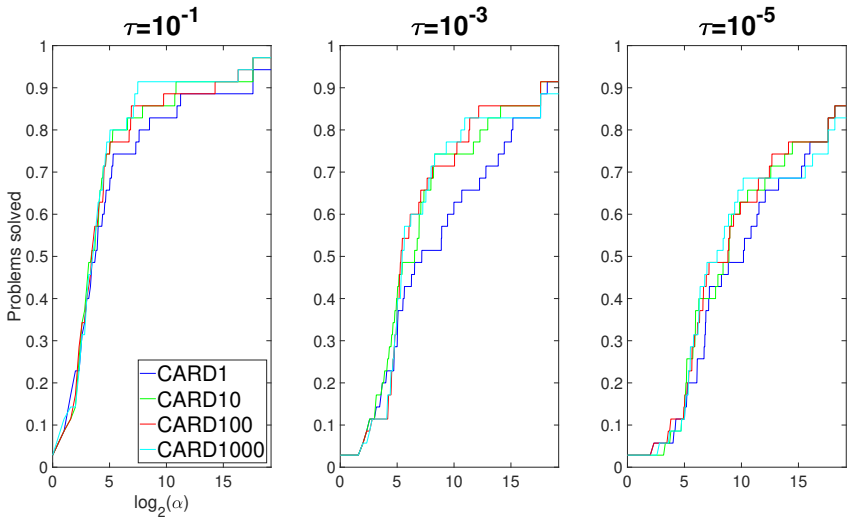


Figure 7.12: Data profiles for the cardinality of the set $\mathcal{D}$

Figure 7.13 presents the three data profiles obtained while verifying if the value of the cosine measure of the primitive may have an impact on the performance of Algorithm 6.

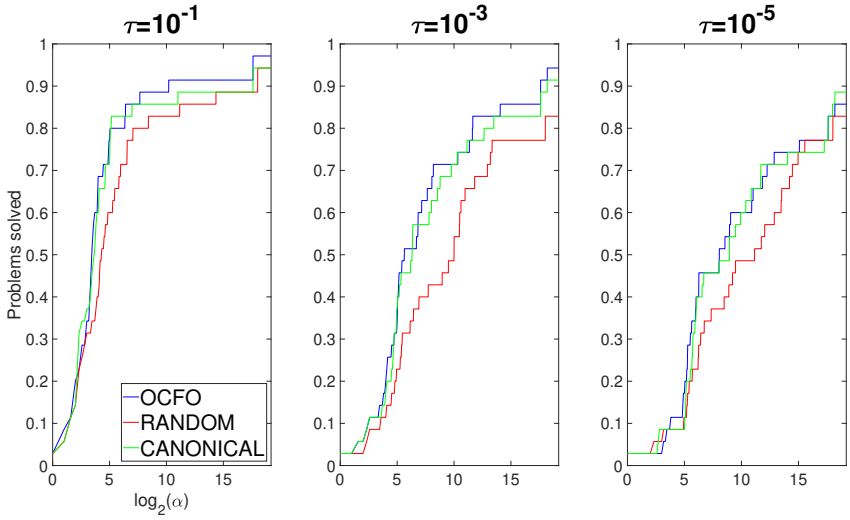


Figure 7.13: Data profiles for the cosine measure of the primitive positive basis

Figure 7.14 presents the three data profiles obtained while verifying if one of the three different sizes tested for the primitive positive basis provides better results than the others.

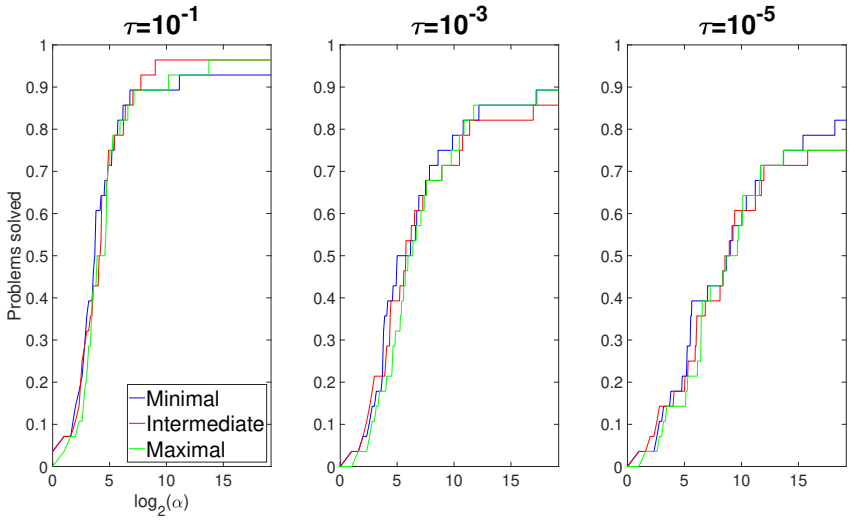


Figure 7.14: Data profiles for the size s of the primitive positive basis

7.2.4 Discussion

In the first experiment, the version CARD1 is the version that provides the worst results in general. This implies that using the same optimal CFOPB at each iteration is not the best approach to use. Based on the three data profiles obtained, using a set of positive bases $\mathcal{D}$ with cardinality more than one increases the efficiency of Algorithm 6. However, it does not seem true that the bigger the set $\mathcal{D}$ is, the better Algorithm 6 performs. For instance, CARD1000 does not always provide better results than the 3 other versions for all 3 values of τ tested. The set $\mathcal{D}$ in the approach CARD100 seems to contain a sufficient number of optimal CFOPBs.

In the second experiment, the version called OCFO that uses an optimal CFOPB as the primitive is the most robust version of the three versions tested when $\tau = 10^{-1}$ and $\tau = 1 \times 10^{-3}$. When $\tau = 1 \times 10^{-5}$, OCFO solves 1 problem less than CAN, and one problem more than RAND. The curve for OCFO dominates almost all the time the curve of RAND, and most of the time the curve of CAN for all three values of τ tested.

Table 7.4 presents the values of the cosine measure computed for each problem for all 3 versions of the algorithm. To compute the cosine measure,

Algorithm 4 is used for the versions OCFO and CAN since the primitive positive basis in both versions is a CFOPB. To compute the cosine measure of the primitive in RAND, the slower algorithm 1 is used since it is very unlikely that this version builds a CFOPB as the primitive.

Table 7.4: The cosine measure of the primitive positive basis

Problem p	n_p	s_p	**OCFO**	**CAN**	**RAND**
1. Rosenbrock	2	3	5.00e-01	3.82e-01	1.07e-02
2. Freudenstein	2	3	5.00e-01	3.82e-01	1.89e-01
3. PowellBS	2	3	5.00e-01	3.82e-01	2.89e-02
4. BrownBS	2	3	5.00e-01	3.82e-01	1.43e-01
5. Beale	2	3	5.00e-01	3.82e-01	2.77e-02
6. Jenrich	2	3	5.00e-01	3.82e-01	1.91e-01
7. Helical	3	4	3.33e-01	2.50e-01	6.94e-03
8. Bard	3	5	4.47e-01	3.57e-01	8.24e-02
9. Gaussian	3	4	3.33e-01	2.50e-01	5.67e-06
10. Meyer	3	4	3.33e-01	2.50e-01	3.23e-02
11. Gulf	3	4	3.33e-01	2.50e-01	4.53e-02
12. Box3D	3	4	3.33e-01	2.50e-01	1.49e-02
13. PowellS	4	5	2.50e-01	1.88e-01	1.32e-02
14. Wood	4	6	3.53e-01	2.43e-01	2.61e-06
15. Kowalik	4	5	2.50e-01	1.88e-01	1.03e-02
16. Brown	4	5	2.50e-01	1.88e-01	3.32e-02
17. Osborne1	5	6	2.00e-01	1.52e-01	3.63e-03
18. Biggs	6	8	2.35e-01	1.50e-01	9.36e-07
19. Osborne2	11	17	2.18e-01	1.23e-01	2.08e-02
20. Watson	12	17	1.82e-01	9.64e-02	2.86e-07
21. RosenbrockE	4	5	2.50e-01	1.88e-01	1.43e-02
22. PowellExt	8	9	1.25e-01	9.82e-02	1.68e-03
23. Penalty1	10	14	1.96e-01	1.09e-01	7.01e-06
24. Penalty2	10	15	2.23e-01	1.24e-01	3.42e-06
25. VariablyDim	7	11	2.77e-01	1.79e-01	3.71e-03
26. Trigonometric	7	12	3.01e-01	2.24e-01	1.46e-03
27. BrownAlm	9	16	2.77e-01	2.13e-01	4.89e-06
28. DiscreteBnd	5	6	2.00e-01	1.52e-01	1.59e-02
29. DiscreteInt	3	4	3.33e-01	2.50e-01	3.56e-02
30. BroydenTri	5	9	3.77e-01	3.18e-01	8.33e-04
31. BroydenBan	20	23	8.63e-02	4.61e-02	1.37e-06
32. LinearFR	15	20	1.49e-01	7.22e-02	5.82e-07
33. LinearR1	30	39	9.90e-02	3.80e-02	Unknown
34. LinearR1W0	25	28	6.91e-02	3.67e-02	1.56e-07
35. Chebyquad	2	3	5.00e-01	3.82e-01	1.76e-01

In Problem $p = 33$, note that the computer utilized with 32GB of RAM is not able to compute the cosine measure of the primitive built in RAND. This shows the limitation of Algorithm 6.2. On the other hand, it takes roughly 0.005 seconds to compute the cosine measure of the primitive positive basis in the versions OCFO and CAN via Algorithm 4.

Table 7.4 shows that RAND builds a primitive positive basis with a lower value of cosine measure for all 34 problems where a comparison is possible. From Table 7.4, we observe that no replications of the same primitive basis occurred in RAND since all values of cosine measures are different.

Recall that RAND builds a positive basis by multiplying the canonical positive basis with a random invertible matrix. Even though it did not occur in this experiment, let us provide an example showing that it is possible that this process creates a positive basis with a higher cosine measure than the canonical positive basis.

Example 7.10. Let

$$M = \begin{bmatrix} -1 & 10 \\ 10 & -1 \end{bmatrix},$$

and consider the canonical positive basis

$$\widehat{\mathbb{D}}_{2,3} = \begin{bmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Then

$$\mathrm{cm}(\widehat{\mathbb{D}}_{2,3}) = \frac{1}{\sqrt{4+2\sqrt{2}}} \approx 0.3827,$$

and

$$\mathrm{cm}(M\widehat{\mathbb{D}}_{2,3}) \approx 0.4282.$$

Now that it is clear that RAND is the version that uses positive bases with the lowest cosine measure, let us analyze the data profiles. When $\tau \in \{10^{-1}, 10^{-3}\}$, OCFO outperforms the two other approaches. When $\tau = 10^{-5}$, OCFO and CAN provide similar results. CAN is able to solve one more problem than OCFO. The results suggest that using optimal CFOPBs positive bases is a better approach than the approach generating positive bases randomly which has lower values of cosine measure. However, there are several limitations in this experiment and a deeper investigation is necessary before claiming that optimal CFOPBs increase the performance of a GPS algorithm. A next step could be to use optimal CFOPBs of all sizes in state-of-the-art algorithms such as the ones in the software Nomad [ALDRMT21]. Then many numerical experiments could be conducted.

In the fourth experiment, there is no clear winner between the three sizes s tested. It seems that the minimal size $s = n + 1$ slightly increases the efficiency of Algorithm 6. When $\tau = 10^{-5}$, MIN is able to solve 2 more problems than INT and MAX. The results suggest that using intermediate positive bases do not negatively impact the performance of Algorithm 6. When $\tau = 10^{-1}$, INT provides better results than MAX and it is more robust than MIN. These results suggest that using intermediate size positive bases might be beneficial when the degree of accuracy of the solution required is low.

7.3 Summary and future research directions

In Section 7.1, we have implemented in DFTR algorithm that uses the GSH to build model functions. A calculus-based approach is compared to a non-calculus approach. When the composite objective function is a quotient ($F = \frac{f_1}{f_2}$) and there is a real root of f_2 near the box constraint, the calculus-based approach greatly outperforms the other methods tested. In this case, the sampling radius needs to be very small to obtain a relatively accurate gradient and Hessian when using the non-calculus approach. In some situations, numerical errors may occur before obtaining the accuracy required for convergence. In general, based on the results obtained, it seems reasonable to think that a calculus-based approach will outperform a non-calculus approach when the Lipschitz constant of the gradient and\or Hessian of F on the box constraint are large numbers. By increasing the range of numbers allowed to generate the functions f_1 and f_2 when using the product rule in the experiment described in Section 7.1.2, it is relatively easy to create a test set of problems in which the differences in performance between the calculus-based approach and the non-calculus approach are more pronounced than the results presented in this section (Figures 7.1–7.3), which are in favor of the calculus-based approach. However, the gain in performance is not as drastic as the results obtained with the quotient rule in our third experiment (Figures 7.8–7.11). Further investigation is needed to determine situations in which the calculus-based approach significantly outperforms the non-calculus approach when using the product rule.

Several modifications to Algorithm 5 were tested to see if it is possible to improve the performance of the non-calculus approach. None of those modifications resulted in a significant increase of performances for all four experiments.

We remark that our implementation of Algorithm 5 is simple. This

is intentional, as the goal is to compare the calculus-based approach with the non-calculus approach. Nonetheless, we remark that, recently, Nocedal et al. found that a DFTR algorithm that used quadratic models built from a forward-finite-difference gradient and a forward-finite-difference Hessian can be surprisingly competitive with state-of-the-art DFO algorithms [MXON22]. From our perspective, now that the calculus-based approach has been established as the superior method within a model-based DFO algorithm, the next logical step is to use this knowledge to improve current state-of-the-art software (e.g., [ALDRMT21]). Applying a calculus-based approach to real-world optimization problems is, then, the obvious direction of future research.

Another direction to explore is inspired by model-based methods for high-dimensional blackbox optimization [CR22]. In [CR22], the authors used subspace decomposition to reduce a high-dimensional blackbox optimization problem into a sequence of low-dimensional blackbox optimization problems. At each iteration, the subspace is rotated, requiring new models to be frequently constructed. An examination of whether calculus rules could be adapted to help in this situation would be an interesting direction for future research.

This further links to a direction of future research based on underdetermined gradients and Hessians. An underdetermined approximation does not contain accurate information about all entries of a gradient/Hessian, but uses fewer function evaluations than the determined case as we have seen in Chapters 4 and 5 . It would be valuable to explore if calculus-based approaches could be merged with underdetermined approximations to create a more efficient version of Algorithm 5.

In Section 7.2, a simple version of a GPS algorithm is implemented (Algorithm 6). Three numerical experiments are conducted:

- Four different cardinalities for the set $\mathcal{D}$ are tested. The results suggest that the set should contain more than one optimal CFOPB.

- Three different types of positive bases that can be categorized in terms of the values of the cosine measure are tested. The results suggest that using optimal CFOPBs improves the performance of Algorithm 7 compared to a version using positive bases with cosine measure near 0 (the version called RAND).

- Three different sizes for the primitive positive basis are tested. The results do not show any major differences between the three versions. In general, it seems that using minimal size optimal CFOPBs slightly

improves the performance of Algorithm 6. Using intermediate size optimal CFOPBs might improve the performance of Algorithm 6 when the degree of accuracy required is low.

The results obtained contain several limitations. The test set used could be described as homogeneous since all objective function have the same form. Another weakness of the test set is that it contains 14 problems of dimension 2 or 3. Furthermore, one can argue that the algorithm used is rudimentary. However, this can also be seen as a strength as it might help to test the three characteristics of interest since there are no procedures in the algorithm to compensate for the potentially bad results obtained with one of the approaches tested. Rather than trying to improve our implementation of Algorithm 6, a more valuable next step should be to incorporate optimal CFOPB of intermediate sizes in state-of-the-art DFO algorithms such as the ones in the Nomad software. Then more numerical experiments should be conducted.

Another future research direction to explore is the use of sets with less than $n+1$ vectors in the poll step. In this case, the set cannot be a positive spanning set of $\mathbb{R}^n$, but it can contain a positive basis of a subspace of $\mathbb{R}^n$. Hence, it might be possible to develop convergence results on a subspace of $\mathbb{R}^n$.

The process to create a random positive basis in the version called RAND shows that only limited results are known about the behavior of the cosine measure when a positive basis is multiplied by an invertible matrix which is not an orthonormal matrix. We have seen that the cosine measure can increase (Example 7.10), decrease (all positive bases created in the RAND version) or stay the same (orthonormal matrix; Proposition 6.30). A future research direction could to explore if it is possible to predict the value of the cosine measure of a positive basis built by multiplying an initial positive basis with an invertible matrix if we only use information from the initial positive basis and the invertible matrix.

Chapter 8

Conclusion

In Chapters 1, 2, the context is clarified, the value of DFO methods is discussed, and a brief overview of DFO methods developed over the years is presented. In Chapter 3, background results and the notation used in this thesis is presented.

In Chapters 4, 5 and 6, this thesis has presented theoretical advancements related to concepts that can be used in in DFO algorithms. In the last part of this thesis, i.e., Chapter 7, the main tools developed have been tested in DFO algorithms and it has been shown that they can be valuable in practice. A summary of the main achievements of each chapter follows.

In Chapter 4, general error bounds covering all possible cases (nondetermined, underdetermined, determined and overdetermined) are developed for the gradient approximation techniques called generalized simplex gradient (GSG) and generalized centered simplex gradient (GCSG). The main achievement of this chapter is to provide one general error bound that covers all possible cases. When the matrix of directions S used to compute the GSG is full row rank, then the GSG is an order-1 accurate approximation of the true gradient. If S is not full row rank, then the GSG is an order-1 accurate approximation of a partial gradient defined as the projection of the true gradient onto the span generated by the column of S. Another secondary achievement in this chapter is the development of error bounds ad infinitum and a formula to obtain the GSG as the number of sample points tends to infinity in a fixed region. These results have helped to understand the behavior of the GSG and its associate error bound in the overdetermined case.

In Chapter 5, we have introduced a compact and explicit formula based on matrix algebra to approximate the Hessian of a function at a point of interest. This technique, called the generalized simplex Hessian (GSH), is well-defined as long as the matrices of directions used are non-empty. In other words, it is well-defined for any non-zero number of sample points. For this reason, it can be viewed as a generalization of a simplex Hessian that is only defined when the set of sample points contained $(n+1)(n+2)/2$ sample points poised for quadratic interpolation. One of the main advan-

tage of the GSH compared to other Hessian approximation techniques is the ease to implement in a software such as Matlab. When the second matrix of directions called $\overline{T}$ is identical for all rows in the difference matrix $\delta_s f(x^0; S, \overline{T})$, a general error bound covering all possible 16 cases is developed and it shows that the GSH is an order-1 accurate approximation of the true Hessian whenever S and $\overline{T}$ are full row rank. If S or $\overline{T}$ is not full row rank, then the GSH is an order-1 accurate approximation of a partial Hessian. A "centered" version of the GSH is defined and called the generalized centered simplex Hessian (GCSH). Error bounds for the GCSH show that the GCSH is an order-2 accurate approximation of the full Hessian whenever the matrices of directions S and $\overline{T}$ are full row rank. If S or $\overline{T}$ is not full row rank, then the GCSH is an order-2 accurate approximation of a partial Hessian. The notion of minimal poised set for the GSH is defined to provide information about the possible structure of the matrices of directions S and $\overline{T}$ to obtain an order-1 accurate approximation of the true Hessian with $(n+1)(n+2)/2$ function evaluations. A minimal poised set for the GCSH is also defined and it provides information on the possible structure of S and $\overline{T}$ to obtain an order-2 accurate approximation of the true Hessian with n^2+n+1 function evaluations. In the last section of Chapter 5, possible choices for the matrices S and T_j are provided and analyzed when we are only interested by a proper subset of the entries of the Hessian such as the diagonal entries, the off-diagonal entries, or one specific row.

In Chapter 6, the first deterministic algorithm to compute the cosine measure of a finite positive spanning set is presented. Then the problem of finding the structure of intermediate positive bases with maximal cosine measure is investigated. To conquer this problem, two types of positive bases are defined: critical-free positive bases (CFPB) and critical-free orthogonal positive bases (CFOPB). The structure of a CFPB in $\mathbb{R}^3$ that maximizes the cosine measure is found: it is a CFOPB where all minimal positive bases of the subspaces are optimal. The structure of a CFOPB in $\mathbb{R}^n$ that maximizes the cosine measure is also found: all minimal positive bases of the subspaces must be optimal and the dimensions of the subspaces must be of a certain dimension (Theorem 6.50). It turns out that the general algorithm to compute the cosine measure (Algorithm 1) can be simplified when computing the cosine measure of any CFPB (Algorithm 2). Furthermore, it is proved that the algorithm can be drastically simplified for CFOPBs (Algorithm 4).

In Section 7.1, the main tools developed in the previous chapters are tested in DFO algorithms. A derivative-free trust region algorithm designed to solve optimization problems where the objective function involves more than one blackbox is implemented. The algorithm build model functions

using the GSH. Two versions of the algorithm are implemented: one version uses a calculus-based approach to build model functions and the second version do not use a calculus-based approach to build model functions. The results suggest that a calculus-based approach should be prioritized when the objective function involves more than one blackbox.

In Section 7.2, a basic GPS algorithm is implemented. Three experiments are conducted. The main experiment suggests that using positive bases with high cosine measure such as optimal CFOPBs is preferable compared to positive bases with cosine measure near zero. The experiment on the cardinality of the set $\mathcal{D}$ containing the positive bases suggest that using more than one CFOPB increases the performance of the algorithm tested. The experiment on the size s of the CFOPBs in the set $\mathcal{D}$ suggests that intermediate size CFOPBs may be valuable when the accuracy of the minimizer required is low.

The work accomplished in this thesis has answered many questions. On the other hand, it has raised many unanswered questions and potentially valuable research directions. To quote Earl C. Kelley [Kel51], "the answers we have found have only served to raise a whole set of new questions. In some ways we feel that we are as confused as ever, but we think we are confused on a higher level and about more important things."

Bibliography

[AA06] M. Abramson and C. Audet. Convergence of mesh adaptive direct search to second-order stationary points. *SIAM Journal on Optimization*, 17(2):606–619, 2006. → pages 9, 13, 192

[AAA+18] S. Alarie, N. Amaioua, C. Audet, S. Le Digabel, and L. Leclaire. Selection of variables in parallel space decomposition for the mesh adaptive direct search algorithm. *Les Cahiers du GERAD. G-2018-38*, 2018. → pages 13

[AACW09] M. Abramson, C. Audet, J. Chrissis, and J. Walston. Mesh adaptive direct search algorithms for mixed variable optimization. *Optimization Letters*, 3:35–47, 2009. → pages 6

[AAD04] M. Abramson, C. Audet, and J. Dennis. Generalized pattern searches with derivative information. *Mathematical Programming*, 100:3–25, 2004. → pages 13

[AAD07] M. Abramson, C. Audet, and J. Dennis. Filter pattern search algorithms for mixed variable constrained optimization problems. *Pacific Journal of Optimization*, pages 477–500, 2007. → pages 6

[AADLD09] M. Abramson, C. Audet, J. Dennis, and S. Le Digabel. OrthoMADS: A deterministic MADS instance with orthogonal directions. *SIAM Journal on Optimization*, 20(2):948–966, 2009. → pages 13, 118

[ABLD08] C. Audet, V. Béchard, and S. Le Digabel. Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. *Journal of Global Optimization*, 41(2):299–318, 2008. → pages 7

[Abr05] M. Abramson. Second-order behavior of pattern search. *SIAM Journal on Optimization*, 16(2):515–530, 2005. → pages 9, 13, 192

[AD01] C. Audet and J. Dennis. Pattern search algorithms for mixed variable programming. *SIAM Journal on Optimization*, 11(3):573–594, 2001. → pages 6

[AD04] C. Audet and J. Dennis. A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization*, 14(4):980–1010, 2004. → pages 13, 191, 192

[AD06] C. Audet and J. Dennis. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006. → pages 13, 118, 192

[AEP11] M. Arouxét, N. Echebest, and E. Pilotta. Active-set strategy in Powell's method for optimization without derivatives. *Computational & Applied Mathematics*, 30(1):171–196, 2011. → pages 167

[AFS14] M. Abramson, L. Frimannslund, and T. Steihaug. A subclass of generating set search with convergence to second-order stationary points. *Optimization Methods and Software*, 29(5):900–918, 2014. → pages 9, 13

[AH17] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Switzerland, 2017. → pages 4, 6, 8, 9, 10, 11, 13, 14, 19, 20, 26, 70, 72, 118, 167, 168, 170, 195, 198

[AH20] C. Audet and W. Hare. Model-based methods in derivative-free nonsmooth optimization. In *Numerical Nonsmooth Optimization*, pages 655–691. Springer, 2020. → pages 5, 175

[AILDT14] C. Audet, A. Ianni, S. Le Digabel, and C. Tribes. Reducing the number of function evaluations in mesh adaptive direct search algorithms. *SIAM Journal on Optimization*, 24(2):621–642, 2014. → pages 13

[ALDRMT21] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. NOMAD version 4: Nonlinear optimization with

the MADS algorithm. *arXiv preprint arXiv:2104.11627*, 2021. → pages 202, 204

[And22] N. Andrei. *Modern Numerical Nonlinear Optimization*, volume 195. Springer Nature, 2022. → pages 19, 68

[ASZ08] C. Audet, G. Savard, and W. Zghal. Multiobjective optimization through a series of single-objective formulations. *SIAM Journal on Optimization*, 19(1):188–210, 2008. → pages 8

[ASZ10] C. Audet, G. Savard, and W. Zghal. A mesh adaptive direct search algorithm for multiobjective optimization. *European Journal of Operational Research*, 204(3):545–556, 2010. → pages 8

[Aud11] C. Audet. A short proof on the cardinality of maximal positive bases. *Optimization Letters*, 5(1):191–194, 2011. → pages 118

[Aud14] C. Audet. A survey on direct search methods for blackbox optimization and their applications. In *Mathematics without boundaries*, pages 31–56. Springer, 2014. → pages 5, 118

[Bar51] M.S. Bartlett. An inverse matrix adjustment arising in discriminant analysis. *The Annals of Mathematical Statistics*, 22(1):107–111, 1951. → pages 17

[BB05] F. Berghen and H. Bersini. CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181(1):157–175, 2005. → pages 167

[BBN19] A. Berahas, R. Byrd, and J. Nocedal. Derivative-free optimization of noisy functions via quasi-Newton methods. *SIAM Journal on Optimization*, 2019. To appear. → pages 11

[BCL+20] J. Burke, F. Curtis, A. Lewis, M. Overton, and L. Simões. Gradient sampling methods for nonsmooth optimization. *Numerical Nonsmooth Optimization: State of the Art Algorithms*, pages 201–225, 2020. → pages 7

[Bec17] A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, 2017. → pages 8

[BFB16] R. Burden, J. Faires, and A. Burden. *Numerical Analysis 10/e IE*. Brooks/Cole Cengage Learning, 2016. → pages 11, 17, 75

[BGP09] C. Bogani, M. Gasparo, and A. Papini. Generalized pattern search methods for a class of nonsmooth optimization problems with structure. *Journal of Computational and Applied Mathematics*, 229(1):283–293, 2009. → pages 7

[BH20] A. Beck and N. Hallak. On the convergence to stationary points of deterministic and randomized feasible descent directions methods. *SIAM Journal on Optimization*, 30(1):56–79, 2020. → pages 118

[BHJB21] P. Braun, W. Hare, and G. Jarry-Bolduc. Limiting behavior of derivative approximation techniques as the number of points tends to infinity on a fixed interval in $\mathbb{R}$. *Journal of Computational and Applied Mathematics*, 386:113218, 22, 2021. → pages 30, 31, 46, 67

[BK98] D. Bortz and C. Kelley. The simplex gradient and noisy optimization problems. In *Computational Methods for Optimal Design and Control*, pages 77–90. Springer, 1998. → pages 19

[BKS08] A. Bagirov, B. Karasözen, and M. Sezer. Discrete gradient method: derivative-free method for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 137(2):317–334, 2008. → pages 7

[BLG13] S. Billups, J. Larson, and P. Graf. Derivative-free optimization of expensive functions with computational error using weighted regression. *SIAM Journal on Optimization*, 23(1):27–53, 2013. → pages 11, 20, 167

[Box66] M. Box. A comparison of several current optimization methods, and the use of transformations in constrained problems. *The Computer Journal*, 9(1):67–77, 1966. → pages 9

[BPC66] N. Banichuk, V. Petrov, and F. Chernous'ko. The solution of variational and boundary value problems by the method of local variations. *USSR Computational Mathematics and Mathematical Physics*, 6(6):1–21, 1966. → pages 12

[Bro88] W. Brown. *A Second Course in Linear Algebra*. Wiley, 1988. → pages 129

[BU06] A. Bagirov and J. Ugon. Piecewise partially separable functions and a derivative-free algorithm for large scale nonsmooth optimization. *Journal of Global Optimization*, 35(2):163, 2006. → pages 7

[BW23] R. Bollapragada and S. Wild. Adaptive sampling quasi-Newton methods for zeroth-order stochastic optimization. *Mathematical Programming Computation*, 15:327–364, 2023. → pages 7

[CDV08] A. Custódio, J. Dennis, and L. Vicente. Using simplex gradients of nonsmooth functions in direct search methods. *IMA Journal of Numerical Analysis*, 28(4):770–784, 2008. → pages 7, 13, 19, 118

[CGT00] A. Conn, N. Gould, and P. Toint. *Trust Region Methods*. SIAM, 2000. → pages 5, 8, 9, 11, 166, 176, 178

[CGT12] C. Cartis, N. Gould, and P. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM Journal on Optimization*, 22(1):66–86, 2012. → pages 11

[CHJB22] Y. Chen, W. Hare, and G. Jarry-Bolduc. Error analysis of surrogate models constructed through operations on submodels. *Mathematics of Operations Research*, 2022. → pages 168, 174

[CK16] X. Chen and C. Kelley. Optimization with hidden constraints and embedded monte carlo computations. *Optimization and Engineering*, 17:157–175, 2016. → pages 7

[CKP15] P. Conejo, E. Karas, and L. Pedroso. A trust-region derivative-free algorithm for constrained optimization. *Optimization Methods and Software*, 30(6):1126–1145, 2015. → pages 167

[CLD13] A. Conn and S. Le Digabel. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. *Optimization Methods & Software*, 28(1):139–158, 2013. → pages 9

[CLPS18] G. Cocchi, G. Liuzzi, A. Papini, and M. Sciandrone. An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints. *Computational Optimization and Applications*, 69(2):267–296, 2018. → pages 8, 11

[CM15] A. Custódio and J. Madeira. GLODS: Global and local optimization using direct search. *Journal of Global Optimization*, 62(1):1–28, 2015. → pages 8

[CM18] A. Custódio and J. Madeira. MultiGLODS: global and local multiobjective optimization using direct search. *Journal of Global Optimization*, 72:323–345, 2018. → pages 8

[CMS18] R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming*, 169:447–487, 2018. → pages 7

[CMVV11] A. Custódio, J. Madeira, A. Vaz, and L. Vicente. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140, 2011. → pages 8

[CP01] I. Coope and C. Price. On the convergence of grid-based methods for unconstrained optimization. *SIAM Journal on Optimization*, 11(4):859–869, 2001. → pages 118

[CP02] I. Coope and C. Price. Positive bases in numerical optimization. *Computational Optimization and Applications*, 21(2):169–175, 2002. → pages 118

[CR22] C. Cartis and L. Roberts. Scalable subspace methods for derivative-free nonlinear least-squares optimization. *Mathematical Programming*, pages 1–64, 2022. → pages 204

[CRV10] A. Custódio, H. Rocha, and L. Vicente. Incorporating minimum Frobenius norm models in direct search. *Computational Optimization and Applications*, 46(2):265–278, 2010. → pages 9, 11, 68

[CS98] J. Conway and N. Sloane. *Sphere Packings, Lattices and Groups*, volume 290. Springer New York, third edition, 1998. → pages 152, 153

[CST97a] A. Conn, K. Scheinberg, and P. Toint. On the convergence of derivative-free methods for unconstrained optimization. *Approximation Theory and Optimization: Tributes to MJD Powell*, pages 83–108, 1997. → pages 10, 167

[CST97b] A. Conn, K. Scheinberg, and P. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79(1):397–414, 1997. → pages 10

[CST98] A. Conn, K. Scheinberg, and P. Toint. A derivative free optimization algorithm in practice. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 4718, 1998. → pages 167

[CSV08a] A. Conn, K. Scheinberg, and L. Vicente. Geometry of interpolation sets in derivative free optimization. *Mathematical Programming*, 111(1-2):141–172, 2008. → pages 10, 20, 26, 69, 80, 189

[CSV08b] A. Conn, K. Scheinberg, and L. Vicente. Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation. *IMA Journal of Numerical Analysis*, 28(4):721–748, 2008. → pages 11, 20, 26, 68, 69, 71, 72, 80, 97

[CSV09a] A. Conn, K. Scheinberg, and L. Vicente. Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points. *SIAM Journal on Optimization*, 20(1):387–415, 2009. → pages 8, 9, 167, 172, 192

[CSV09b] A. Conn, K. Scheinberg, and L. Vicente. *Introduction to Derivative-Free Optimization*. SIAM, 2009. → pages 5, 6, 10, 11, 13, 26, 69, 71, 72, 75, 97, 123, 167, 172, 191, 192, 193, 194, 195

[CT05] B. Colson and P. Toint. Optimizing partially separable functions without derivatives. *Optimization Methods and Software*, 20(4-5):493–508, 2005. → pages 167

[CT19] I. Coope and R. Tappenden. Efficient calculation of regular simplex gradients. *Computational Optimization and Applications*, 72(3):561–588, 2019. → pages 19

[CT21] I. Coope and R. Tappenden. Gradient and diagonal Hessian approximations using quadratic interpolation models and aligned regular bases. *Numerical Algorithms*, 88:767–791, 2021. → pages 19, 20, 68

[CV07] A. Custódio and L. Vicente. Using sampling and simplex derivatives in pattern search methods. *SIAM Journal on Optimization*, 18(2):537–555, 2007. → pages 12, 19, 68, 69, 97

[Dav54] C. Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76(6):733–746, 1954. → pages 118

[DEMR08] M. Diniz-Ehrhardt, J. Martinez, and M. Raydan. A derivative-free nonmonotone line-search technique for unconstrained optimization. *Journal of Computational and Applied Mathematics*, 219(2):383–397, 2008. → pages 11

[Den78] J. Dennis. A brief introduction to quasi-Newton methods. *Numerical analysis*, 22:19–52, 1978. → pages 167

[DF06] G. Deng and M. Ferris. Adaptation of the UOBYQA algorithm for noisy functions. In *Proceedings of the 38th Conference on Winter Simulation*, WSC '06, page 312–319. Winter Simulation Conference, 2006. → pages 7

[DF09] G. Deng and M. Ferris. Variable-number sample-path optimization. *Mathematical Programming*, 117(1-2):81–109, 2009. → pages 7

[DH13] C. Davis and W. Hare. Exploiting known structures to approximate normal cones. *Mathematics of Operations Research*, 38(4):665–681, 2013. → pages 19

[DLGG84] R. De Leone, M. Gaudioso, and L. Grippo. Stopping criteria for linesearch methods without derivatives. *Mathematical programming*, 30(3):285–300, 1984. → pages 11

[DLT03] E. Dolan, R. Lewis, and V. Torczon. On the local convergence of pattern search. *SIAM Journal on Optimization*, 14(2):567–583, 2003. → pages 13

[DM79] J. Dennis and H. Mei. Two new unconstrained optimization algorithms which use function and gradient values. *Journal of Optimization Theory and Applications*, 28(4):453–482, 1979. → pages 167

[DS96] J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996. → pages 167

[DW22] K. Dzahini and S. Wild. Stochastic trust-region algorithm in random subspaces with convergence and expected complexity analyses. *arXiv preprint arXiv:2207.06452*, 2022. → pages 7

[FK06] D. Finkel and C. Kelley. Additive scaling and the direct algorithm. *Journal of Global Optimization*, 36:597–608, 2006. → pages 8

[Fle65] R. Fletcher. Function minimization without evaluating derivatives—a review. *The Computer Journal*, 8(1):33–41, 1965. → pages 9

[Fle80] R. Fletcher. *Practical Methods of Optimization: Volume 1 Unconstrained Optimization*. Wiley, 1980. → pages 167

[FLLR14] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi. A linesearch-based derivative-free approach for nonsmooth constrained optimization. *SIAM Journal on Optimization*, 24(3):959–992, 2014. → pages 7

[FM52] E. Fermi and N. Metropolis. Numerical solution of a minimum problem. Technical report, Los Alamos Scientific Laboratory of the University of California, 1952. → pages 12

[Fol01] G. Folland. How to integrate a polynomial over a sphere. *The American Mathematical Monthly*, 108(5):446–448, 2001. → pages 54, 55

[FS07] L. Frimannslund and T. Steihaug. A generating set search method using curvature information. *Computational Optimization and Applications*, 38(1):105–121, 2007. → pages 13

[FS11] L. Frimannslund and T. Steihaug. On a new method for derivative free optimization. *International Journal of Advanced Software Engineering*, 2011. → pages 13

[GC91] A. Griewank and G. Corliss. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. Society for Industrial and Applied Mathematics, Philadelphia, 1991. → pages 4

[GC00] M. Gen and R. Cheng. *Genetic Algorithms and Engineering Optimization*. Wiley, New York, 2000. → pages 9

[Gee09] Z. Geem. *Harmony Search Algorithms for Structural Design Optimization*, volume 239. Springer, Berlin, 2009. → pages 9

[GH12] F. Gao and L. Han. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1):259–277, 2012. → pages 12

[GHA14] E. Gumma, M. Hashim, and M. Ali. A derivative-free algorithm for linearly constrained optimization problems. *Computational Optimization and Applications*, 57(3):599–621, 2014. → pages 167

[GJV16] R. Garmanjani, D. Júdice, and L. Vicente. Trust-region methods without using derivatives: worst case complexity and the nonsmooth case. *SIAM Journal on Optimization*, 26(4):1987–2011, 2016. → pages 7

[GK06] G. Gray and T. Kolda. Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. *ACM Transactions on Mathematical Software (TOMS)*, 32(3):485–507, 2006. → pages 13

[GLD15] R. Gramacy and S. Le Digabel. The mesh adaptive direct search algorithm with treed Gaussian process surrogates. *Pa-*

cific Journal of Optimization, 11(3):419–447, 2015. → pages 9

[GLL88] L. Grippo, F. Lampariello, and S. Lucidi. Global convergence and stabilization of unconstrained minimization methods without derivatives. *Journal of Optimization Theory and Applications*, 56(3):385–406, 1988. → pages 11

[Gol12] J. Golan. *The Linear Algebra a Beginning Graduate Student Ought to Know*. Springer, Dordrecht, 3rd edition, 2012. → pages 16

[GQT66] S. Goldfeld, R. Quandt, and H. Trotter. Maximization by quadratic hill-climbing. *Econometrica: Journal of the Econometric Society*, pages 541–551, 1966. → pages 167

[GR15] L. Grippo and F. Rinaldi. A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations. *Computational Optimization and Applications*, 60(1):1–33, 2015. → pages 11, 19

[GRV16] S. Gratton, C. Royer, and L. Vicente. A second-order globally convergent direct-search method and its worst-case complexity. *Optimization*, 65(6):1105–1128, 2016. → pages 9

[Gry22] D. Grynkiewicz. *The Characterization of Finite Elasticities: Factorization Theory in Krull Monoids via Convex Geometry*, volume 2316. Springer International Publishing, Cham, 1st edition, 2022. → pages 118

[GS07] L. Grippo and M. Sciandrone. Nonmonotone derivative-free methods for nonlinear equations. *Computational Optimization and Applications*, 37(3):297–328, 2007. → pages 11

[GTT11] S. Gratton, P. Toint, and A. Tröltzsch. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optimization Methods and Software*, 26(4-5):873–894, 2011. → pages 167

[GVL96] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996. → pages 131

[Har20] W. Hare. A discussion on variational analysis in derivative-free optimization. *Set-Valued and Variational Analysis*, pages 1–17, 2020. → pages 168

[Has08] O. Hasancebi. Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures. *Computers & structures*, 86(1):119–132, 2008. → pages 9

[Heb73] M. Hebden. *An Algorithm for Minimization Using Exact Second Derivatives*. Theoretical Physics Division, Atomic Energy Research Establishment, 1973. → pages 167

[HJ61] R. Hooke and T. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961. → pages 12

[HJ90] R. Horn and C. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original. → pages 15, 104, 125

[HJB18] W. Hare and G. Jarry-Bolduc. Calculus identities for generalized simplex gradients: Rules and applications. *Submitted to SIAM Journal on Optimization*, 2018. → pages 19, 168

[HJB20] W. Hare and G. Jarry-Bolduc. A deterministic algorithm to compute the cosine measure of a finite positive spanning set. *Optimization Letters*, 14(6):1305–1316, 2020. → pages v, 131

[HJB23] W. Hare and G. Jarry-Bolduc. About the performance of a calculus-based approach to building model functions in a derivative-free trust-region algorithm. *Algorithms*, 16(2), 2023. → pages v, 10

[HJBP20] W. Hare, G. Jarry-Bolduc, and C. Planiden. Hessian approximations. *arXiv preprint arXiv:2011.02584*, 2020. → pages 168, 174

[HJBP22] W. Hare, G. Jarry-Bolduc, and C. Planiden. A matrix algebra approach to approximate Hessians. *Preprint*, 588, 2022. → pages v, 11

[HJBP23a] W. Hare, G. Jarry-Bolduc, and C. Planiden. Limiting behaviour of the generalized simplex gradient as the number of points tends to infinity on a fixed shape in $\mathbb{R}^n$. *Set-Valued and Variational Analysis*, 31(1):1–33, 2023. → pages v

[HJBP23b] W. Hare, G. Jarry-Bolduc, and C. Planiden. Nicely structured positive bases with maximal cosine measure. *Optimization Letters*, pages 1–21, 2023. → pages v

[HJP20] W. Hare, G. Jarry–Bolduc, and C. Planiden. Error bounds for overdetermined and underdetermined generalized centred simplex gradients. *IMA Journal of Numerical Analysis*, 42(1):744–770, 12 2020. → pages v, 168

[HKT01] P. Hough, T. Kolda, and V. Torczon. Asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Scientific Computing*, 23(1):134–156, 2001. → pages 13

[HL14] W. Hare and Y. Lucet. Derivative-free optimization via proximal point methods. *Journal of Optimization Theory and Applications*, 160:204–220, 2014. → pages 7, 11

[HN08] W. Huyer and A. Neumaier. SNOBFIT–stable noisy optimization by branch and fit. *ACM Transactions on Mathematical Software (TOMS)*, 35(2):1–25, 2008. → pages 8, 167

[HN13] W. Hare and J. Nutini. A derivative-free approximate gradient sampling algorithm for finite minimax problems. *Computational Optimization and Applications*, 56(1):1–38, 2013. → pages 7

[HNT13] W. Hare, J. Nutini, and S. Tesfamariam. A survey of non-gradient optimization methods in structural engineering. *Advances in Engineering Software*, 59:19–28, 2013. → pages 5

[HR22] M. Hough and L. Roberts. Model-based derivative-free methods for convex-constrained optimization. *SIAM Journal on Optimization*, 32(4):2552–2579, 2022. → pages 172, 175

[HS11] X. Hou and J. Shao. Spherical distribution of 5 points with maximal distance sum. *Discrete & Computational Geometry*, 46(1):156–174, 2011. → pages 164, 165

[HWS09] J. He, L. Watson, and M. Sosonkina. Algorithm 897: VTDIRECT95: serial and parallel codes for the global optimization algorithm DIRECT. *ACM Transactions on Mathematical Software (TOMS)*, 36(3):1–24, 2009. → pages 8

[Jam21] S. Jamali. A new second order derivative free method for numerical solution of non-linear algebraic and transcendental equations using interpolation technique. *Journal of Mechanics of Continua and Mathematical Sciences*, 16(4), 2021. → pages 9

[JB19] G. Jarry-Bolduc. Calculus identities for generalized simplex gradients : rules and applications. Master's thesis, University of British Columbia, 2019. → pages 19

[JB22] G. Jarry-Bolduc. Approximating the diagonal of a Hessian: which sample set of points should be used. *Numerical Algorithms*, 91(3):1349–1361, 2022. → pages v, 68, 70, 104, 110, 117

[JBNS19] G. Jarry-Bolduc, P. Nadeau, and S. Singh. Uniform simplex of an arbitrary orientation. *Optimization Letters*, pages 1–11, 2019. → pages 151

[Kel51] E. Kelley. *The Workshop Way of Learning*. Harper, New York, 1951. → pages 208

[Kel99a] C. Kelley. Detection and remediation of stagnation in the nelder–mead algorithm using a sufficient decrease condition. *SIAM Journal on Optimization*, 10(1):43–55, 1999. → pages 12, 19

[Kel99b] C. Kelley. *Iterative Methods for Optimization*, volume 18. SIAM, 1999. → pages 10, 11, 19, 20

[Kel11] C. Kelley. *Implicit Filtering*, volume 23. SIAM, 2011. → pages 11, 68, 118

[KGV83] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science (American Association for the Advancement of Science)*, 220(4598):671–680, 1983. → pages 9

[KK12] A. Kaveh and M. Khayatazad. A new meta-heuristic method: Ray optimization. *Computers & structures*, 112-113:283–294, 2012. → pages 9

[KLT03] T. Kolda, R. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003. → pages 13, 118

[KLW18] K. A Khan, J. Larson, and S. Wild. Manifold sampling for optimization of nonconvex functions that are piecewise linear compositions of smooth components. *SIAM Journal on Optimization*, 28(4):3001–3024, 2018. → pages 168

[Kup01] W. Kuperberg. An extremum property characterizing the n-dimensional regular cross-polytope. *arXiv preprint math/0112290*, 2001. → pages 152

[KZ10] S. Kim and D. Zhang. Convergence properties of direct search methods for stochastic optimization. In *Proceedings of the 2010 Winter Simulation Conference*, pages 1003–1011, 2010. → pages 7

[LB16] J. Larson and S. Billups. Stochastic derivative-free optimization using a trust region framework. *Computational Optimization and applications*, 64:619–645, 2016. → pages 7

[Lev44] K. Levenberg. A method for the solution of certain problems in least squares. *Applied Mathematics*, 2:164–168, 1944. → pages 167

[LL11] G. Luh and C. Lin. Optimal design of truss-structures using particle swarm optimization. *Computers & structures*, 89(23):2221–2232, 2011. → pages 9

[LLR12] G. Liuzzi, S. Lucidi, and F. Rinaldi. Derivative-free methods for bound constrained mixed-integer optimization. *Computational Optimization and Applications*, 53:505–526, 2012. → pages 6

[LLR15] G. Liuzzi, S. Lucidi, and F. Rinaldi. Derivative-free methods for mixed-integer constrained optimization problems. *Journal of Optimization Theory and Applications*, 164:933–965, 2015. → pages 6

[LLR16] G. Liuzzi, S. Lucidi, and F. Rinaldi. A derivative-free approach to constrained multiobjective nonsmooth optimization. *SIAM Journal on Optimization*, 26(4):2744–2774, 2016. → pages 7, 8

[LLR20] G. Liuzzi, S. Lucidi, and F. Rinaldi. An algorithmic framework based on primitive directions and nonmonotone line searches for black-box optimization problems with integer variables. *Mathematical Programming Computation*, 12:673–702, 2020. → pages 6

[LLRV19] G. Liuzzi, S. Lucidi, F. Rinaldi, and L. Vicente. Trust-region methods for the derivative-free optimization of nonsmooth black-box functions. *SIAM Journal on Optimization*, 29(4):3012–3035, 2019. → pages 4, 7, 167

[LMW16] J. Larson, M. Menickelly, and S. Wild. Manifold sampling for l_1 nonconvex optimization. *SIAM Journal on Optimization*, 26(4):2540–2563, 2016. → pages 7

[LMW19] J. Larson, M. Menickelly, and S. Wild. Derivative-free optimization methods. *Acta Numerica*, 28(2010):287–404, 2019. → pages 5, 6, 10, 11, 12, 13

[LMZ21] J. Larson, M. Menickelly, and B. Zhou. Manifold sampling for optimizing nonsmooth nonconvex compositions. *SIAM Journal on Optimization*, 31(4):2638–2664, 2021. → pages 7, 168

[LPK02] N. Lagaros, M. Papadrakakis, and G. Kokossalakis. Structural optimization using evolutionary algorithms. *Computers & Structures*, 80(7):571–589, 2002. → pages 9

[LPW12] J. Lagarias, B. Poonen, and M. Wright. Convergence of the restricted Nelder–Mead algorithm in two dimensions. *SIAM Journal on Optimization*, 22(2):501–532, 2012. → pages 12

[LRWW98] J. Lagarias, J. Reeds, M. Wright, and P. Wright. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998. → pages 12

[LS02a] T. Lu and S. Shiou. Inverses of 2× 2 block matrices. *Computers & Mathematics with Applications*, 43(1-2):119–129, 2002. → pages 126

[LS02b] S. Lucidi and M. Sciandrone. On the global convergence of derivative-free methods for unconstrained optimization. *SIAM Journal on Optimization*, 13(1):97–116, 2002. → pages 11

[LT96] R. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. Technical report, Institute for Computer Applications in Science and Engineering, Hampton VA, 1996. → pages 118

[LT99] R. Lewis and V. Torczon. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization*, 9(4):1082–1099, 1999. → pages 192

[LT00] R. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000. → pages 192

[LT17] P. Lax and M. Terrell. *Multivariable Calculus with Applications*. Springer, 2017. → pages 26

[LW15] S. Le Digabel and S. Wild. A taxonomy of constraints in black-box simulation-based optimization. Technical Report 1505.07881, ArXiv, 2015. → pages 6

[LZY15] Q. Liu, J. Zeng, and G. Yang. MrDIRECT: a multilevel robust DIRECT algorithm for global optimization problems. *Journal of Global Optimization*, 62(2):205–227, 2015. → pages 8

[M+65] J Matyas et al. Random optimization. *Automation and Remote control*, 26(2):246–253, 1965. → pages 9

[Mad75] K. Madsen. An algorithm for minimax solution of overdetermined systems of non-linear equations. *IMA Journal of Applied Mathematics*, 16(3):321–328, 1975. → pages 167

[MB08] U. Murty and J. Bondy. *Graph Theory*. Springer, 2008. → pages 158

[McK62] R. McKinney. Positive bases for linear spaces. *Transactions of the American Mathematical Society*, 103(1):131–148, 1962. → pages 118

[MGH81] J. Moré, B. Garbow, and K. Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software (TOMS)*, 7(1):17–41, 1981. → pages 195, 197

[MMW12] A. Muc and M. Muc-Wierzgoń. An evolution strategy in structural optimization problems for plates and shells. *Composite Structures*, 94(4):1461–1470, 2012. → pages 9

[MN02] M. Marazzi and J. Nocedal. Wedge trust region methods for derivative free optimization. *Mathematical Programming*, 91(2):289–305, 2002. → pages 167

[Mor83] J. Moré. Recent developments in algorithms and software for trust region methods. In *Mathematical programming: The state of the art*, pages 258–287. Springer-Verlag, 1983. → pages 167

[MS83] J. Moré and D. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983. → pages 167

[MW09] J. Moré and S. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009. → pages 179

[MXON22] Hao-Jun M., M. Qiming Xuan, F. Oztoprak, and J. Nocedal. On the numerical performance of finite-difference-based methods for derivative-free optimization. *Optimization Methods and Software*, 0(0):1–23, 2022. → pages 204

[NA15] E. Newby and M. Ali. A trust-region-based derivative free algorithm for mixed integer programming. *Computational Optimization and Applications*, 60:199–229, 2015. → pages 6

[Næv18] G. Nævdal. Positive bases with maximal cosine measure. *Optimization Letters*, pages 1–8, 2018. → pages 119, 124, 134, 137, 150, 153

[NFSL11] A. Neumaier, H. Fendl, H. Schilly, and T. Leitner. VXQR: derivative-free unconstrained optimization based on QR factorizations. *Soft Computing*, 15(11):2287–2298, 2011. → pages 11

[NM65] J. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. → pages 9, 12

[NT02] L. Nazareth and P. Tseng. Gilding the lily: A variant of the Nelder-Mead algorithm based on golden-section search. *Computational Optimization and Applications*, 22(1):133–144, 2002. → pages 12

[NW06] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 2006. → pages 75, 166

[OB07] R. Oeuvray and M. Bierlaire. A new derivative-free algorithm for the medical image registration problem. *International Journal of Modelling and Simulation*, 27(2):115–124, 2007. → pages 167

[OB09] R. Oeuvray and M. Bierlaire. Boosters: A derivative-free algorithm based on radial basis functions. *International Journal of Modelling Simulation*, 29(1):26–36, 2009. → pages 20

[Osb76] M. Osborne. Nonlinear least squares—the Levenberg algorithm revisited. *Journal of the Australian Mathematical Society*, 19(3):343–357, 1976. → pages 167

[PCB02] C. Price, I. Coope, and D. Byatt. A convergent variant of the Nelder-Mead algorithm. *Journal of Optimization Theory and Applications*, 113(1):5–19, 2002. → pages 12

[Pow64] M. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964. → pages 9

[Pow70a] M. Powell. A Fortran subroutine for solving systems of nonlinear algebraic equations. Technical report, Atomic Energy Research Establishment, Harwell, England (United Kingdom), 1970. → pages 167

[Pow70b] M. Powell. A hybrid method for nonlinear equations. *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114, 1970. → pages 167

[Pow70c] M. Powell. A new algorithm for unconstrained optimization. In *Nonlinear programming*, pages 31–65. Elsevier, 1970. → pages 167

[Pow75] M. Powell. Convergence properties of a class of minimization algorithms. In *Nonlinear programming 2*, pages 1–27. Elsevier, o.l. mangasarian, r.r. meyer and s.m. robinson edition, 1975. → pages 167

[Pow84] M. Powell. On the global convergence of trust region algorithms for unconstrained minimization. *Mathematical Programming*, 29(3):297–303, 1984. → pages 167

[Pow94] M. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994. → pages 10

[Pow98] M. Powell. The use of band matrices for second derivative approximations in trust region algorithms. In *Advances in Nonlinear Programming: Proceedings of the 96 International Conference on Nonlinear Programming*, pages 3–28. Springer, 1998. → pages 68

[Pow01] M. Powell. On the lagrange functions of quadratic models that are defined by interpolation. *Optimization Methods and Software*, 16(1-4):289–309, 2001. → pages 10

[Pow02] M. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–582, 2002. → pages 10, 167

[Pow03] M. Powell. On trust region methods for unconstrained minimization without derivatives. *Mathematical Programming*, 97(3):605–623, 2003. → pages 10, 167

[Pow04a] M. Powell. Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. *Mathematical Programming*, 100(1):183–215, 2004. → pages 10, 20, 68

[Pow04b] M. Powell. On the use of quadratic models in unconstrained minimization without derivatives. *Optimization Methods and Software*, 19(3-4):399–411, 2004. → pages 10, 68

[Pow04c] M. Powell. On updating the inverse of a KKT matrix. *Numerical Linear Algebra and Optimization*, pages 56–78, 2004. → pages 11, 68

[Pow06] M. Powell. The NEWUOA software for unconstrained optimization without derivatives. In *Large-scale nonlinear optimization*, pages 255–297. Springer, 2006. → pages 11, 68

[Pow07] M. Powell. A view of algorithms for optimization without derivatives. *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, 43(5):170–174, 2007. → pages 11, 68

[Pow08] M. Powell. Developments of NEWUOA for minimization without derivatives. *IMA Journal of Numerical Analysis*, 28(4):649–664, 2008. → pages 11, 68

[Pow09] M. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, pages 26–46, 2009. → pages 167

[Pow13] M. Powell. Beyond symmetric Broyden for updating quadratic models in minimization without derivatives. *Mathematical Programming*, 138(1):475–500, 2013. → pages 11

[Pow15] M. Powell. On fast trust region methods for quadratic models with linear constraints. *Mathematical Programming Computation*, 7(3):237–267, 2015. → pages 167

[PS20] C. Paquette and K. Scheinberg. A stochastic line search method with expected complexity analysis. *SIAM Journal on Optimization*, 30(1):349–376, 2020. → pages 7

[PT17] M. Porcelli and P. Toint. BFO, a trainable derivative-free brute force optimizer for nonlinear bound-constrained optimization and equilibrium computations with continuous and discrete variables. *ACM Transactions on Mathematical Software (TOMS)*, 44(1):1–25, 2017. → pages 6

[PTVF07] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge university press, 2007. → pages 12

[Ras63] L. Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automation & Remote Control*, 24:1337–1342, 1963. → pages 9

[Rea65] J. Reay. A new proof of the Bonnice-Klee theorem. *Proceedings of the American Mathematical Society*, 16(4):585–587, 1965. → pages 118

[Rea66] J. Reay. Unique minimal representations with positive bases. *The American Mathematical Monthly*, 73(3):253–261, 1966. → pages 118

[Reg15] R. Regis. The calculus of simplex gradients. *Optimization Letters*, 9(5):845–865, 2015. → pages 19, 20, 22, 26, 168

[Reg16a] R. Regis. Multi-objective constrained black-box optimization using radial basis function surrogates. *Journal of Computational Science*, 16:140–155, 2016. → pages 8

[Reg16b] R. Regis. On the convergence of adaptive stochastic search methods for constrained and multi-objective black-box optimization. *Journal of Optimization Theory and Applications*, 170:932–959, 2016. → pages 8

[Reg16c] R. Regis. On the properties of positive spanning sets and positive bases. *Optimization and Engineering*, 17(1):229–262, 2016. → pages 118, 122

[Reg21] R. Regis. On the properties of the cosine measure and the uniform angle subspace. *Computational Optimization and Applications*, 78(3):915–952, 2021. → pages 163

[Rom87] Z. Romanowicz. Geometric structure of positive bases in linear spaces. *Applicationes Mathematicae*, 19(3-4):557–567, 1987. → pages 118, 137, 138, 139

[Rom07] S. Roman. *Advanced Linear Algebra*, volume 135. Springer, New York, 3rd edition, 2007. → pages 16

[Ros60] H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960. → pages 9

[RS05] R. Regis and C. Shoemaker. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31:153–171, 2005. → pages 20

[RS07] R. Regis and C. Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19(4):497–509, 2007. → pages 11

[RS13] L. Rios and N. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56:1247–1293, 2013. → pages 13

[RW17] R. Regis and S. Wild. CONORBIT: constrained optimization by radial basis function interpolation in trust regions. *Optimization Methods and Software*, 32(3):552–580, 2017. → pages 167

[Ryk80] A. Rykov. Simplex direct search algorithms. *Automation & Remote Control*, 41(6):784–793, 1980. → pages 12

[SAHT20] K. Skandalos, H. Afshari, W. Hare, and S. Tesfamariam. Multi-objective optimization of inter-story isolated buildings using metaheuristic and derivative-free algorithms. *Soil Dynamics and Earthquake Engineering*, 132:106058, 2020. → pages 8, 9

[SAM10] S. Sankaran, C. Audet, and A. Marsden. A method for stochastic constrained optimization using derivative-free surrogate pattern search and collocation. *Journal of Computational Physics*, 229(12):4664–4682, 2010. → pages 7

[SCA09] T. Sriver, J. Chrissis, and M. Abramson. Pattern search ranking and selection algorithms for mixed variable simulation-based optimization. *European Journal of Operational Research*, 198(3):878–890, 2009. → pages 6, 7

[SG13] M. Saka and Z. Geem. Mathematical and metaheuristic applications in design optimization of steel frame structures: An extensive review. *Mathematical Problems in Engineering*, 2013:1–33, 2013. → pages 9

[She71] G. Shephard. Diagrams for positive bases. *Journal of the London Mathematical Society*, 2(1):165–175, 1971. → pages 118

[SHH62] W. Spendley, R. Hext, and F. Himsworth. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4(4):441–461, 1962. → pages 12

[Sor81] D. Sorensen. Trust-region methods for unconstrained minimization. Technical report, Argonne National Lab., IL (USA), 1981. → pages 167

[Sor82] D. Sorensen. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982. → pages 167

[ST10] K. Scheinberg and P. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM Journal on Optimization*, 20(6):3512–3532, 2010. → pages 10

[ST15] P. Sampaio and P. Toint. A derivative-free trust-funnel method for equality-constrained nonlinear optimization. *Computational Optimization and Applications*, 61(1):25–49, 2015. → pages 167

[Ste83] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983. → pages 167

[SWJ98] M. Schonlau, W. Welch, and D. Jones. Global versus local search in constrained optimization of computer models. *Lecture Notes–Monograph Series*, pages 11–25, 1998. → pages 20

[Toi78] P. Toint. Some numerical results using a sparse matrix updating formula in unconstrained optimization. *Mathematics of Computation*, 32(143):839–851, 1978. → pages 167

[Toi79] P. Toint. On the superlinear convergence of an algorithm for solving a sparse minimization problem. *SIAM Journal on Numerical Analysis*, 16(6):1036–1045, 1979. → pages 167

[Toi81a] P. Toint. Convergence properties of a class of minimization algorithms that use a possibly unbounded sequence of quadratic approximations, 1981. → pages 167

[Toi81b] P. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In *Sparse Matrices and their Uses*, pages 57–88. Academic press, London, 1981. → pages 167

[Tor91] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, 1(1):123–145, 1991. → pages 12

[Tor97] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997. → pages 2, 118, 191, 192

[Tse99] P. Tseng. Fortified-descent simplicial search method: A general approach. *SIAM Journal on Optimization*, 10(1):269–288, 1999. → pages 12

[VC12] L. Vicente and A. Custódio. Analysis of direct searches for discontinuous functions. *Mathematical Programming*, 133:299–325, 2012. → pages 13

[VDHL17] K. Vu, C. D'Ambrosio, Y. Hamadi, and L. Liberti. Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, 24(3):393–424, 2017. → pages 8

[Vic13] L. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, 1(1-2):143–153, 2013. → pages 13

[VKPS17] A. Verdério, E. Karas, L. Pedroso, and K. Scheinberg. On the construction of quadratic models for derivative-free trust-region algorithms. *EURO Journal on Computational Optimization*, 5(4):501–527, 2017. → pages 11

[VV09] A. Vaz and L. Vicente. Pswarm: a hybrid solver for linearly constrained global derivative-free optimization. *Optimization Methods & Software*, 24(4-5):669–685, 2009. → pages 118

[Wil08] S. Wild. MNH: A derivative-free optimization algorithm using minimal norm Hessians. *Tenth Copper Mountain Conference on Iterative Methods*, 2008. → pages 11

[Wil09] S. Wild. *Derivative-Free Optimization Algorithms for Computationally Expensive Functions*. PhD thesis, Cornell University, 2009. → pages 11

[Win70] D. Winfield. *Function and Functional Optimization by Interpolation in Data Tables*. PhD thesis, Harvard University, 1970. → pages 10, 68, 167

[Win73] D. Winfield. Function minimization by interpolation in a data table. *IMA Journal of Applied Mathematics*, 12(3):339–347, 1973. → pages 167

[WRS08] S. Wild, R. Regis, and C. Shoemaker. ORBIT: optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008. → pages 68, 167

[WS11] S. Wild and C. Shoemaker. Global convergence of radial basis function trust region derivative-free algorithms. *SIAM Journal on Optimization*, 21(3):761–781, 2011. → pages 20, 167

[WS13] S. Wild and C. Shoemaker. Global convergence of radial basis function trust-region algorithms for derivative-free optimization. *SIAM Review*, 55(2):349–371, 2013. → pages 11

[Zha14] Z. Zhang. Sobolev seminorm of quadratic functions with applications to derivative-free optimization. *Mathematical Programming*, 146(1-2):77–96, 2014. → pages 11