UMEÅ UNIVERSITY


Master's Thesis in Engineering Physics, 30 ECTS
Master of Science Programme in Engineering Physics, 300 ECTS
Spring term 2021

# On the Topic of Unconstrained Black-Box Optimization with Application to Pre-Hospital Care in Sweden 

Author: Tim Anthony

Master's Thesis in Engineering Physics(30 ECTS)
Supervisor: Armin Eftekhari
Examiner: Jun Yu

# On the Topic of Unconstrained Black-Box Optimization with Application to Pre-Hospital Care in Sweden 

Master's Thesis in Engineering Physics, 30 ECTS
Tim Anthony, Timpam1995@gmail.com
Supervisor: Armin Eftekhari, Department of Mathematics and Mathematical Statistics, Umeå University.
Examiner: Jun Yu, Department of Mathematics and Mathematical Statistics, Umeå University.

Department of Physics
Linnaus Väg 20, 901 87, Umeå, Sweden
www.physics.umu.se
Copyright © 2021. All Rights Reserved.


#### Abstract

In this thesis, the theory and application of black-box optimization methods are explored. More specifically, we looked at two families of algorithms, descent methods and response surface methods (closely related to trust region methods). We also looked at possibilities in using a dimension reduction technique called active subspace which utilizes sampled gradients. This dimension reduction technique can make the descent methods more suitable to high-dimensional problems, which turned out to be most effective when the data have a ridge-like structure. Finally, the optimization methods were used on a real-world problem in the context of pre-hospital care where the objective is to minimize the ambulance response times in the municipality of Umeå by changing the positions of the ambulances.

Before applying the methods on the real-world ambulance problem, a simulation study was performed on synthetic data, aiming at finding the strengths and weaknesses of the different models when applied to different test functions, at different levels of noise.

The results showed that we could improve the ambulance response times across several different performance metrics compared to the response times of the current ambulance positions. This indicates that there exist adjustments that can benefit the pre-hospital care in the municipality of Umeå. However, since the models in this thesis work find local and not global optimums, there might still exist even better ambulance positions that can improve the response time further.


## Sammanfattning

I denna rapport undersöks teorin och tillämpningarna av diverse blackbox optimeringsmetoder. Mer specifikt så har vi tittat på två familjer av algoritmer, descentmetoder och responsytmetoder (nära besläktade med tillitsregionmetoder). Vi tittar också på möjligheterna att använda en dimensionreduktionsteknik som kallas active subspace som använder samplade gradienter för att göra descentmetoderna mer lämpade för högdimensionella problem, vilket visade sig vara mest effektivt när datat har en struktur där ändringar i endast en riktning har effekt på responsvärdet. Slutligen användes optimeringsmetoderna på ett verkligt problem från sjukhusvården, där målet var att minimera svarstiderna för ambulansutryckningar i Umeå kommun genom att ändra ambulanspositionerna.

Innan metoderna tillämpades på det verkliga ambulansproblemet genomfördes också en simuleringsstudie på syntetiskt data. Detta för att hitta styrkorna och svagheterna hos de olika modellerna genom att undersöka hur dem hanterar ett flertal testfunktioner under olika nivåer av brus.

Resultaten visade att vi kunde förbättra ambulansernas responstider över flera olika prestandamått jämfört med responstiderna för de nuvarande ambulanspositionerna. Detta indikerar att det finns förändringar av positioneringen av ambulanser som kan gynna den pre-hospitala vården inom Umeå kommun. Dock, eftersom modellerna i denna rapport hittar lokala och inte globala optimala punkter kan det fortfarande finnas ännu bättre ambulanspositioner som kan förbättra responstiden ytterligare.

## Contents

1 Introduction ..... 1
1.1 Motivation ..... 1
1.2 Methodology ..... 1
1.3 Research questions ..... 2
1.4 Main Contributions ..... 2
1.5 Outline of Report ..... 2
2 Theory ..... 3
2.1 General Optimization Theory ..... 3
2.1.1 Problem Formulation ..... 3
2.2 Convexity ..... 4
2.3 Continuity ..... 5
2.4 Linear Algebra ..... 6
2.4.1 Vector spaces ..... 6
2.4.2 Rank ..... 7
2.4.3 Symmetric Matrices ..... 7
2.4.4 Positive Semi-Definite Matrices ..... 8
2.4.5 Projection Onto a Subspace ..... 9
2.5 Active Subspace Method ..... 9
2.5.1 Probability Theory ..... 9
2.5.2 Decomposition ..... 10
2.5.3 Approximation ..... 12
2.5.4 Response Surface ..... 13
2.5.5 Monte Carlo Method ..... 14
2.5.6 Ridge Functions and Implications ..... 14
2.6 Stochastic Subspace Descent ..... 15
2.6.1 Iteration Process ..... 16
2.6.2 Finding the Projection Matrix ..... 16
2.6.3 Stochastic Coordinate Descent ..... 16
2.6.4 Convergence and Rate of Convergence for Stochastic Subspace De- scent Without Noise ..... 17
2.6.5 Convergence for Stochastic Subspace Descent Under Non-Convex Assumption ..... 18
2.6.6 Armijo Line Search ..... 19
2.7 Design of Experiments ..... 20
2.7.1 Response Surface Methodology ..... 21
2.7.2 Coded Variables ..... 21
2.7.3 Experimental Designs ..... 21
2.7.4 Models ..... 24
2.7.5 Canonical Analysis of Second Order Model ..... 26
2.7.6 Trust Region Subproblem ..... 28
3 Method and Implementation ..... 34
3.1 Active Subspace \& Stochastic Subspace Descent ..... 34
3.2 Active Subspace \& Stochastic Coordinate Descent ..... 36
3.3 Response Surface Method ..... 37
3.4 Simulation Study ..... 39
3.4.1 Test Functions ..... 40
3.5 Application to Pre-Hospital Care ..... 44
3.5.1 Performance Measures ..... 46
4 Result and Discussion ..... 48
4.1 Simulation Study ..... 48
4.2 Application to Pre-Hospital Care ..... 56
4.2.1 Median Time Performance Measure ..... 56
4.2.2 90:th Percentile Time Performance Measure ..... 57
4.2.3 Maximum DeSO Median Time Performance Measure ..... 58
5 Conclusion ..... 60
5.1 Simulation study ..... 60
5.2 Application to Pre-Hospital Care ..... 61
5.3 Future work ..... 61
Bibliography ..... 63
Appendices ..... 65
A Proof Theorem 1 ..... 66
B Proof Theorem 2 ..... 67
C Proof Theorem 4 ..... 68
D Gradient of ridge function ..... 69
E Proof Theorem 10 ..... 70
F Proof Theorem 11 ..... 72
G Proof Theorem 13 ..... 74
H Proof Theorem 14 ..... 78
I Tables - Application to Pre-Hospital Care ..... 79
I. 1 Median Time Performance Measure ..... 79
I. 2 90:th Percentile Time Performance Measure ..... 80
I. 3 Maximum DeSO Median Time Performance Measure ..... 80

## Chapter 1

## Introduction

### 1.1 Motivation

Computer models for complex systems often behave like black boxes due to their complexity. This has led to an interest in black-box optimization algorithms where the goal is to optimize said computer models, for example to maximize yield or to minimize production time.

One such black-box model exists in the context of pre-hospital care. It is a statistical simulation model, which has been developed by the northern regions in Sweden, SOS alarm and Umeå University [1]. This model uses historical call data to generate a simulation, which then can be used to decide how a change in ambulance positioning, ambulance scheduling or ambulance count would affect the response time to emergency calls. Due to the complexity of this model, it can be considered as a black-box model.

However, even if the computer models to simulate the outcome of a certain ambulance resource allocation exist, there are currently no way of finding the best resource allocation in order to minimize the response times besides guesswork and trial and error. Hence, there is a clear need of computerized data-driven optimization models which could help health officials in finding the optimal resource allocation of the ambulances, which in turn could save important seconds in a scenario where seconds is matter of life and death.

### 1.2 Methodology

The methods used in this report can be divided into two groups, descent methods and response surface methods (closely related to trust region methods). For the first group stochastic subspace descent, stochastic coordinate descent and gradient descent were used. We also looked at possibilities in using the dimension reduction technique called active subspace in order to lower the dimension of the problem, while keeping as much of
the original data as possible. For the second group of methods this report will cover an extensive explanation of the response surface methodology and trust region methodology.

### 1.3 Research questions

This thesis aims to answer the following two questions:

1. How can the methods described above be used in the context of black-box optimization?
2. How can we use these methods to position the ambulances to minimize the response times?

### 1.4 Main Contributions

The main contributions of this thesis work are:

1. Methodological - the development of descent-based optimization methods using active subspace methodology. These are the agreement method, the weighting method, and also the active subspace \& stochastic coordinate descent method.
2. Theoretical - convergence results which can be seen in theorems 10,11 and 13. These results were achieved by straightforward modifications of existing convergence proofs in order to show convergence when considering stochastic subspace descent.
3. Simulation study - where we performed a simulation study to better understand the strengths and weaknesses of the descent-based and trust region methods. Here we also look at the new proposed descent-based methods.
4. The application to a real problem - where we successfully show that optimization methods can be applied to the ambulance simulation problem to find an improved ambulance placement.

### 1.5 Outline of Report

For the outline of this report Chapter 2 contains the theory behind the different algorithms and methods. Chapter 3 describes the methods used and how they were applied to perform the simulation study and the ambulance optimization. In Chapter 4 the result is presented along with a discussion how the result should be interpreted. Finally, in Chapter 5 thoughts, insights and conclusions about the project are summarized and also suggestions about what future developments that could be possible are provided.

## Chapter 2

## Theory

### 2.1 General Optimization Theory

### 2.1.1 Problem Formulation

The general problem formulation for optimization [2] can be stated as

$$
\begin{equation*}
\min _{\mathbf{x} \in \mathbb{X}} f(\mathbf{x}) \tag{2.1}
\end{equation*}
$$

where $f(\mathbf{x})$ is the target function which we want to minimize, $\mathbf{x}=\left[x_{1}, \ldots, x_{m}\right]$ is our vector with variables and $\mathbb{X} \subseteq \mathbb{R}^{m}$ is the set defining the allowed solutions. An optimal solution to this problem, which minimizes the target function while staying inside $\mathbb{X}$ is called an optimal solution and is denoted as $\mathbf{x}^{*}$. The corresponding optimal value of the target function is denoted as $f^{*}=f\left(\mathbf{x}^{*}\right)$. Note that there are two types of optimal solutions which are local optima which means that $f(\mathbf{x})$ is minimized in a small neighbourhood around $\mathbf{x}$ and global optima which mean that $f(\mathbf{x})$ is minimized over all of $\mathbb{X}$.

The set of allowed solutions is usually expressed as a set of constraints as $\mathbb{X}=\left\{g_{i}(\mathbf{x}) \leq\right.$ $\left.b_{i}\right\}_{i=1}^{m}$ where $g_{i}(\mathbf{x})$ are functions and $b_{i}$ are constants. Depending on the character of these constraints and also the target function, the optimization problem can be classified further into different problem classes. Some of them are:

- Linear programming problems - if both the target function $f(\mathbf{x})$ and all the constraints $g_{i}(\mathbf{x})$ are linear functions
- Non-linear problems - if at the target function $f(\mathbf{x})$ or at least one of the constraints $g_{i}(\mathbf{x})$ are a non-linear function
- Integer programming problems - if the set $\mathbb{X}$ is limited to only consist of integers as $\mathbb{X} \subseteq \mathbb{Z}$
- Mixed integer programming (MIP) problems - if $\mathbb{X}$ is a mix of continuous and integer numbers
- Unconstrained optimization - if $\mathbb{X}=\mathbb{R}^{m}$ which is to say that we have no constraints.


### 2.2 Convexity

Here we define an important concept in optimization called convexity, and present some relevant results. First let us define a convex function and a convex set as following.

Definition 1 (Convex function) A function $f: \mathbb{X} \mapsto \mathbb{R}$ is convex if for every $\mathbf{x}, \mathbf{y} \in$ $\mathbb{X}$ and every $0 \leq \lambda \leq 1$ we have that

$$
f(\lambda \mathbf{x}+(1-\lambda) \mathbf{y}) \leq \lambda f(\mathbf{x})+(1-\lambda) f(\mathbf{y})
$$

From this definition we see that a function is convex if for every pair of two points $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{X}$, all intermediate points on the straight line $\lambda \mathbf{x}+(1-\lambda) \mathbf{y}, 0 \leq \lambda \leq 1$, between $\mathbf{x}$ and $\mathbf{y}$, have lower function value than the straight line between $f(\mathbf{x})$ and $f(\mathbf{y})$. If the inequality in the definition is strict, we can also call the function strictly convex. In this case we also require $0<\lambda<1$.

Definition 2 (Strictly convex function) A function $f: \mathbb{X} \mapsto \mathbb{R}$ is strictly convex if for every $\mathbf{x}, \mathbf{y} \in \mathbb{X}$ and every $0<\lambda<1$ we have that

$$
f(\lambda \mathbf{x}+(1-\lambda) \mathbf{y})<\lambda f(\mathbf{x})+(1-\lambda) f(\mathbf{y})
$$

Now, let $\nabla f(\mathbf{x})=\left[\frac{\partial f}{\partial x_{1}}(\mathbf{x}), \ldots, \frac{\partial f}{\partial x_{m}}(\mathbf{x})\right]^{T}$ be the gradient of $f(\mathbf{x})$, where $\frac{\partial f}{\partial x_{i}}(\mathbf{x})$ are the partial derivatives and the operator $(\cdot)^{T}$ is the transpose given by $\left[\mathbf{C}^{T}\right]_{i j}=[\mathbf{C}]_{j i}$ when applied to a matrix C. Note this definition of transpose also holds for us, since our gradient vector can be seen as a $1 \times m$ matrix. Furthermore, let the norm of a vector $\mathbf{v} \subseteq \mathbb{R}^{s}$ be $\|\mathbf{v}\|=\sqrt{v_{1}^{2}+\cdots+v_{s}^{2}}$. Now we can define an even stricter form of convexity which is strong convexity [3] as:
Definition 3 (Strongly convex function) A differentiable function $f: \mathbb{X} \mapsto \mathbb{R}$ is strongly convex with parameter $\mu>0$ (or $\mu$-strongly convex) if for every $\mathbf{x}, \mathbf{y} \in \mathbb{X}$ and every $0 \leq \lambda \leq 1$ we have that any of the following equivalent conditions are true:
1.

$$
f(\mathbf{y}) \geq f(\mathbf{x})+\nabla f(\mathbf{x})^{T}(\mathbf{y}-\mathbf{x})+\frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^{2}
$$

2. 

$$
(\nabla f(\mathbf{x})-\nabla f(\mathbf{y}))^{T}(\mathbf{x}-\mathbf{y}) \geq \mu\|\mathbf{x}-\mathbf{y}\|^{2}
$$

3. 

$$
f(\lambda \mathbf{x}+(1-\lambda) \mathbf{y}) \leq \lambda f(\mathbf{x})+(1-\lambda) f(\mathbf{y})-\frac{\lambda(1-\lambda) \mu}{2}\|\mathbf{x}-\mathbf{y}\|
$$

From condition 3 of the definition of strongly convex functions, we can see that strong convexity implies strict convexity. It is also evident that a strictly convex function also is convex. Now, we continue by defining the convex set.

Definition 4 (Convex set) A set $\mathbb{X}$ is convex if for every $\mathbf{x}, \mathbf{y} \in \mathbb{X}$ and every $0<$ $\lambda<1$ we have that

$$
\lambda \mathbf{x}+(1-\lambda) \mathbf{y} \in \mathbb{X} .
$$

From this definition we see that if for every two points in the set, all points on the straight line between them also belong to the same set, the set is said to be convex. A common way to think about convex set is to say that if you are in $\mathbb{X}$ and can "see" every other point in $\mathbb{X}$, then this set is convex. We now bring convexity of functions and convexity of sets together to define the convex problem.

Definition 5 (Convex problem) The optimization problem $\min _{\mathbf{x} \in \mathbb{X}} f(\mathbf{x})$ is convex if $f$ : $\mathbb{X} \mapsto \mathbb{R}$ is a convex function and $\mathbb{X}$ is a convex set.

If a problem is convex, we have that a point that locally minimizes $f$ also minimizes $f$ globally. Since it is easier to find local minimum points than global minimum points, the following result is of great interest.

Theorem 1 (Local minimum equals global minimum for convex problems) If $\min _{\mathbf{x} \in \mathbb{X}} f(\mathbf{x})$ is a convex optimization problem, then every local minimum in $\mathbb{X}$ is also a global minimum.
Proof: See Appendix A.

### 2.3 Continuity

Here we define the Lipschitz continuity property of a function [4].
Definition 6 (Lipschitz continuity) A function $f: \mathbb{X} \mapsto \mathbb{R}$ is Lipschitz continuous with parameter $L_{1}$ if for all $\mathbf{x}, \mathbf{y} \in \mathbb{X}$ we have that

$$
|f(\mathbf{x})-f(\mathbf{y})| \leq L_{1}\|\mathbf{x}-\mathbf{y}\| .
$$

Similarly, the definition of a Lipschitz continuous gradient of a function is as follows.
Definition 7 (Lipschitz continuous gradient) A function $f: \mathbb{X} \mapsto \mathbb{R}$ has a Lipschitz continuous gradient with parameter $L_{2}$ if for all $\mathbf{x}, \mathbf{y} \in \mathbb{X}$ we have that

$$
|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})| \leq L_{2} \mid\|\mathbf{x}-\mathbf{y}\| .
$$

Now, we will present an interesting implication of the Lipschitz continuous gradient property that will be useful later. Here we will use the inner product of two vectors of the same dimension, say $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{s}$, as $\langle\mathbf{a}, \mathbf{b}\rangle=\mathbf{a}^{T} \mathbf{b}=a_{1} b_{1}+\cdots+a_{s} b_{s}$.

Theorem 2 If the function $f: \mathbb{X} \mapsto \mathbb{R}$ has a Lipschitz continuous gradient with parameter $L_{2}$, then it follows for all $\mathbf{x}, \mathbf{y} \in \mathbb{X}$ that

$$
f(\mathbf{y}) \leq f(\mathbf{x})+\langle\mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x})\rangle+\frac{L_{2}}{2}\|\mathbf{y}-\mathbf{x}\|^{2}
$$

Proof: See Appendix B.

### 2.4 Linear Algebra

Here we present some useful definitions and results from linear algebra, which all are based on the texbook [5, p. 123-172, 288-312].

### 2.4.1 Vector spaces

If we let $\mathbf{A} \in \mathbb{R}^{n \times m}$ we can define the column space of $\mathbf{A}$ as following.
Definition 8 (Column space) The column space $\mathcal{C}(\mathbf{A}) \subseteq \mathbb{R}^{n}$ of $\mathbf{A}$ consists of all linear combinations of the columns in $\mathbf{A}$.

Hence for any given $\mathbf{A}$ and $\mathbf{b}=\left[b_{1}, \ldots, b_{n}\right]^{T} \in \mathbb{R}^{n}$, we are only able to solve $\mathbf{A x}=\mathbf{b}$ if $\mathbf{b} \in \mathcal{C}(\mathbf{A})$. Similarly, the null space of $\mathbf{A}$ can be defined as

Definition 9 (Null space) The null space $\mathcal{N}(\mathbf{A}) \subseteq \mathbb{R}^{m}$ of $\mathbf{A}$ consists of all vectors $\mathbf{x}$ which satisfy $\mathbf{A x}=\mathbf{0}$.

Now, we want to clarify what a basis is. To do this we first need to define the span of a vector space and what we mean by independence.

Definition 10 (Span) A set of vectors $\left\{\mathbf{v}_{i}\right\}_{i=1}^{k}, \mathbf{v}_{i} \in \mathbb{R}^{m}$, are said to span the vector space $\mathbb{S} \subseteq \mathbb{R}^{m}$ if every point $\mathbf{s} \in \mathbb{S}$ can be represented by a linear combination of the $\mathbf{v}_{i}:$ s, i.e. $\mathbf{s}=\sum_{i=1}^{k} c_{i} \mathbf{v}_{i}, \forall \mathbf{s}$.

Definition 11 (Linear independence) $A$ set of vectors $\left\{\mathbf{v}_{i}\right\}_{i=1}^{k}, \mathbf{v}_{i} \in \mathbb{R}^{m}$, are said to be linearly independent if the linear combination $\sum_{i=1}^{k} c_{i} \mathbf{v}_{i}$ only equals $\mathbf{0}$ when $c_{1}=$ $\cdots=c_{k}=0$.

Using these two definitions we can specify the basis as
Definition 12 (Basis) A basis for a vector space $\mathbb{S} \subseteq \mathbb{R}^{m}$ are a set of vectors $\left\{\mathbf{v}_{i}\right\}_{i=1}^{k}$, $\mathbf{v}_{i} \in \mathbb{R}^{m}$, satisfying:

1. the vectors $\mathbf{v}_{i}$ are linearly independent
2. the vectors $\left\{\mathbf{v}_{i}\right\}_{i=1}^{k}$ span the vector space $\mathbb{S}$.

Here we are noting that the condition regarding linear independence will prohibit the basis for having any redundant vectors while the condition that the basis need to span the vector space $\mathbb{S}$ will ensure that it is possible to represent every point in the vector space with our basis. One result of these conditions is that the points in the vector space are uniquely represented by a combination of the basis vectors. Another result is the fact that the number of basis vectors needed to represent a vector space always is the same, this lead us to the definition of the vector space dimension.

Definition 13 (Dimension) The basis of a vector space always contain the same number of vectors, this number is the dimension of that vector space.

### 2.4.2 Rank

Another important concept is the rank $r$ of the matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$.
Definition 14 (Rank) The rank $r$ is the number of independent and non-zero columns of A.

This concept correlates to the dimension of a vector space, since the rank of $\mathbf{A}$ equals the dimension of its column space. Note that the maximum rank for the matrix $\mathbf{A}$ is $r=\min \{m, n\}$, when $\mathbf{A}$ satisfy this condition it is said to have full column rank.

### 2.4.3 Symmetric Matrices

Now we consider the square matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, then we can define symmetry of a matrix:
Definition 15 (Symmetric matrix) If $\mathbf{S}=\mathbf{S}^{T}$ the matrix is symmetric.
Matrices that are symmetric and real hold the property that their eigenvectors are orthogonal and their eigenvalues are real, this is summarized by the spectral theorem below.

Theorem 3 (Spectral Theorem) If $\mathbf{S} \in \mathbb{R}^{n \times n}$ is symmetric, its eigenvalue decomposition $\mathbf{S}=\mathbf{Q} \Lambda \mathbf{Q}^{-1}$ always exists and has the following two properties:

1. the eigenvalues on the diagonal of $\boldsymbol{\Lambda}$ are real
2. the eigenvectors $\mathbf{q}_{i}$ in the columns of $\mathbf{Q}=\left[\mathbf{q}_{1}, \ldots, \mathbf{q}_{n}\right]$ are orthogonal, the matrix $\mathbf{Q}$ is thus orthogonal and it follows that $\mathbf{Q}^{-1}=\mathbf{Q}^{T}$. The eigendecomposition can hence be written as $\mathbf{S}=\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^{T}$.

Proof: See [5, p. 339-343].
From the spectral theorem we are given another interesting and useful fact. This is due to the orthogonality of $\mathbf{Q}$, where we notice that the decomposition $\mathbf{S}=\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^{T}$ can be rewritten as

$$
\mathbf{S}=\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^{T}=\lambda_{1} \mathbf{q}_{1} \mathbf{q}_{1}^{T}+\cdots+\lambda_{n} \mathbf{q}_{n} \mathbf{q}_{n}^{T}
$$

This decomposition, which we call the spectral decomposition, follows from

$$
\begin{aligned}
\mathbf{S} & =\left[\mathbf{q}_{1}, \ldots, \mathbf{q}_{n}\right]\left[\begin{array}{lll}
\lambda_{1} & & \\
& \ddots & \\
& & \lambda_{n}
\end{array}\right]\left[\begin{array}{c}
\mathbf{q}_{1}^{T} \\
\vdots \\
\mathbf{q}_{n}^{T}
\end{array}\right] \\
& =\left[\mathbf{q}_{1}, \ldots, \mathbf{q}_{n}\right]\left[\begin{array}{c}
\lambda_{1} \mathbf{q}_{1}^{T} \\
\vdots \\
\lambda_{n} \mathbf{q}_{n}^{T}
\end{array}\right] \\
& =\lambda_{1} \mathbf{q}_{1} \mathbf{q}_{1}^{T}+\cdots+\lambda_{n} \mathbf{q}_{n} \mathbf{q}_{n}^{T} .
\end{aligned}
$$

Note here that each component $\lambda_{i} \mathbf{q}_{i} \mathbf{q}_{i}^{T}$ are themselves matrices of rank 1 . Since the $\mathbf{q}_{i}$ :s are orthogonal and hence also independent, the number of non-zero eigenvalues will determine the number of independent contributions, via $\lambda_{i} \mathbf{q}_{i} \mathbf{q}_{i}^{T}$, to the column space of $\mathbf{S}$. Hence, the rank of $\mathbf{S}$ is given by the number of non-zero eigenvalues. Or, by looking at it in the opposite way which will be of importance later, we can state that the number of non-zero eigenvalues of the symmetric matrix $\mathbf{S}$ is the same as its rank $r$.

### 2.4.4 Positive Semi-Definite Matrices

If we consider the a symmetric square matrix $\mathbf{S}$ as described above, we have the following definition of a positive semi-definite matrix.

Definition 16 (Positive semi-definite matrix) The symmetric matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ is positive semi-definite if $\mathbf{x}^{T} \mathbf{S} \mathbf{x} \geq 0$ for all vectors $\mathbf{x} \in \mathbb{R}^{n}$.

Note that if $\mathbf{S}$ would fulfill a strict inequality as $\mathbf{x}^{T} \mathbf{S} \mathbf{x}>0$ then it is called a positive definite matrix. The reason why positive semi-definite matrices are interesting is because all their eigenvalues are non-negative:

Theorem 4 (Non-negative eigenvalues of positive semi-definite matrices) If the symmetric matrix $\mathbf{S}$ is positive semi-definite, then $\lambda_{i} \geq 0$, for all eigenvalues of $\mathbf{S}$. Proof: See Appendix C.

Similarly, for positive definite matrices, we get strictly positive eigenvalues.

### 2.4.5 Projection Onto a Subspace

Consider a vector $\mathbf{b} \in \mathbb{R}^{n}$, now consider a subspace $\mathbb{V} \subset \mathbb{R}^{n}$ of $m<n$ dimensions which have a basis given by the matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$. If we want to project the vector $\mathbf{b}$ onto $\mathbb{V}$ this is done as

$$
\mathbf{p}=\mathbf{P b}
$$

where $\mathbf{p} \in \mathbb{V}$ is our projection point and $\mathbf{P}=\mathbf{A}\left(\mathbf{A}^{T} \mathbf{A}\right)^{-1} \mathbf{A}^{T} \in \mathbb{R}^{n \times n}$ is our projection matrix. Note that with this kind of projection the subspace $\mathbb{V}$ is equivalent to the column space of $\mathbf{A}$, and hence the dimension of $\mathbb{V}$ is the same as the rank of $\mathbf{A}$.

### 2.5 Active Subspace Method

Active subspace method is a dimensionality reduction technique that utilizes gradients to find the subspace on which the function is varying the most, this subspace is called the active subspace. The whole domain which is assumed to be high-dimensional is then projected onto this smaller active subspace, on which we can build a response surface (for example with linear regression) which now is feasible due to the relatively low dimension. If we now consider an new arbitrary point in the original domain, we can then project this point onto the active subspace where we can use our response surface to determine our response value.

### 2.5.1 Probability Theory

The probability space is the triplet $(\Omega, \mathcal{F}, P)[6$, p. 10$]$ where:

- $\Omega$ is the sample space which denotes the set of all possible outcomes.
- $\mathcal{F}$ is a $\sigma$-algebra of sets, which we call events.
- $P$ is a probability measure, which assigns a probability between 0 and 1 to all events in $\mathcal{F}$.
Let $\mathbf{X}$ be a $m$-dimensional random vector such that $\mathbf{X}$ is a measurable function from $\Omega$ to $\mathbb{R}^{m}$, i.e. $\mathbf{X}: \Omega \mapsto \mathbb{R}^{m}$. We can now define the joint distribution function of the random vector $\mathbf{X}$ as

$$
F_{X}(\mathbf{x})=P(\mathbf{X} \leq \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^{m}
$$

where the inequality is assumed to be element-wise, that is $\{\mathbf{X} \leq \mathbf{x}\}=\left\{X_{1} \leq\right.$ $\left.x_{1}, \ldots, X_{m} \leq x_{m}\right\}$. Assuming that the joint distribution function is absolutely continuous, we can define the (joint) density $\rho: \mathbb{R}^{m} \mapsto \mathbb{R}^{+}$as

$$
\rho(\mathbf{x})=\frac{\partial^{m} F_{x}(\mathbf{x})}{\partial x_{1} \cdots \partial x_{m}}, \quad \mathbf{x} \in \mathbb{R}^{m}
$$

where we note that $\rho(\mathbf{x})$ is non-zero only when being inside $\mathbb{X}$ since we are considering $\mathbf{x}$ to be constrained to the domain $\mathbb{X}$. Furthermore, the density has the property that $\int_{\mathbb{R}^{m}} \rho(\mathbf{x}) d \mathbf{x}=\int_{\mathbb{X}} \rho(\mathbf{x}) d \mathbf{x}=1$.

### 2.5.2 Decomposition

Formally, consider a continuous function $f: \mathbb{X} \mapsto \mathbb{R}$ where the domain $\mathbb{X}$ of $f$ lives in $m$-dimensional space, i.e. $\mathbb{X} \subseteq \mathbb{R}^{m}$. We assume that the domain $\mathbb{X}$ is centered at the origin with the same range in each of the $m$ dimensions and also that the function $f=f(\mathbf{x}), \mathbf{x}=\left[x_{1}, \ldots, x_{m}\right]^{T} \in \mathbb{X}$, is differentiable such that its gradient $\nabla_{x} f(\mathbf{x})=\left[\partial f / \partial x_{1}(\mathbf{x}), \ldots, \partial f / \partial x_{m}(\mathbf{x})\right]^{T}$ exists. Note that $\nabla_{x} f(\mathbf{x})=\nabla f(\mathbf{x})$, however this alternative notation will be beneficial later. Now, define $\mathbf{C}$ as the expectation of the outer product of the $f$-gradients

$$
\begin{equation*}
\mathbf{C} \equiv E\left[\nabla_{x} f(\mathbf{x}) \nabla_{x} f(\mathbf{x})^{T}\right]=\int_{\mathbb{X}} \nabla_{x} f(\mathbf{x}) \nabla_{x} f(\mathbf{x})^{T} \rho(\mathbf{x}) d \mathbf{x} \tag{2.2}
\end{equation*}
$$

Properties of the matrix $\mathbf{C}$ are stated in Lemma 1.

## Lemma 1 (Properties of matrix C)

1) (Symmetry) The matrix $\mathbf{C}$ is symmetric.
2) (Positive semi-definiteness) The matrix $\mathbf{C}$ is positive semi-definite.

Proof: See [7, p. 119-120].
We can now eigen-decompose $\mathbf{C}$ as

$$
\mathbf{C}=\mathbf{W} \boldsymbol{\Lambda} \mathbf{W}^{T}
$$

where $\mathbf{W}$ is an orthogonal matrix whose columns contain the eigenvectors of $\mathbf{C}$, and $\boldsymbol{\Lambda}$ is the corresponding diagonal eigenvalue matrix which contain the eigenvalues $\lambda_{i}$, $i \in 1, \ldots, m$, in decreasing order. We know these eigenvalues will be real due to the symmetry of $\mathbf{C}$ and non-negative due to the positive semi-definiteness of $\mathbf{C}$.

Lemma 2 Let $\mathbf{w}_{i}$ be the $i$ :th column of $\mathbf{W}$, then the expected squared value of gradient $\nabla_{x} f$ in the direction of $\mathbf{w}_{i}$ is given by

$$
E\left[\left(\left(\nabla_{x} f\right)^{T} \mathbf{w}_{i}\right)^{2}\right]=\lambda_{i}
$$

Proof: See [8]
From Lemma 2 we see that the average squared change of $f$ in the direction of eigenvector $\mathbf{w}_{i}$ is proportional to the corresponding eigenvalue $\lambda_{i}$. This motivates us to decompose $\mathbf{W}$ into the $n<m$ most important eigenvectors as

$$
\mathbf{W}=\left[\mathbf{W}_{1} \mathbf{W}_{2}\right]
$$

since we in this way will capture the directions where $\nabla_{x} f$ changes mostly in the $n$ active directions specified by the $m \times n$ matrix $\mathbf{W}_{1}$. An example of this in two dimensions is the decomposition is depicted in Figure 2.1.


Figure 2.1: Visualization of the decomposed eigenvectors $\mathbf{W}_{1}$ and $\mathbf{W}_{2}$ together with the contours of the corresponding two-dimensional function $f(x, y)=x^{2}+y^{2}+(x+2 y)^{2}$. Note that the function most rapidly along the eigenvector $\mathbf{W}_{1}$.

Along with the decomposition of the eigenvectors, we have the corresponding decomposition of the eigenvalue matrix as

$$
\boldsymbol{\Lambda}=\left(\begin{array}{cc}
\Lambda_{1} & 0 \\
0 & \Lambda_{2}
\end{array}\right)
$$

Defining the subspaces $\mathbb{Y}=\left\{\mathbf{y}: \mathbf{y}=\mathbf{W}_{1}^{T} \mathbf{x}, \mathbf{x} \in \mathbb{X}\right\} \subseteq \mathbb{R}^{n}$ and $\mathbb{Z}=\left\{\mathbf{z}: \mathbf{z}=\mathbf{W}_{2}^{T} \mathbf{x}, \mathbf{x} \in\right.$ $\mathbb{X}\} \subseteq \mathbb{R}^{(m-n)}$, we can now project $\mathbf{x}$ onto these subspaces as

$$
\mathbf{y}=\mathbf{W}_{1}^{T} \mathbf{x}
$$

and

$$
\mathbf{z}=\mathbf{W}_{2}^{T} \mathbf{x}
$$

where we have that $\mathbf{y} \in \mathbb{Y}$ and $\mathbf{z} \in \mathbb{Z}$. Also, the orthogonality of $\mathbf{W}$ implies that $\mathbb{X}=\mathbb{Y} \oplus \mathbb{Z}$ and we can thus rewrite $\mathbf{x}$ as $\mathbf{x}=\mathbf{W} \mathbf{W}^{T} \mathbf{x}=\mathbf{W}_{1} \mathbf{W}_{1}^{T} \mathbf{x}+\mathbf{W}_{2} \mathbf{W}_{2}^{T} \mathbf{x}=$ $\mathbf{W}_{1} \mathbf{y}+\mathbf{W}_{2} \mathbf{z}$. Furthermore, $\mathbb{Y}$ and $\mathbb{Z}$ are known as our active subspace and inactive subspace, respectively.

Lemma 3 Let $\mathbf{y}$ and $\mathbf{z}$ be defined as above, note that $f(\mathbf{x})=f\left(\mathbf{W}_{1} \mathbf{y}+\mathbf{W}_{2} \mathbf{z}\right)$ and let the gradients with respect to $\mathbf{y}$ and $\mathbf{z}$ be $\nabla_{y} f(\mathbf{x})=\left[\partial f / \partial y_{1}(\mathbf{x}), \ldots, \partial f / \partial y_{n}(\mathbf{x})\right]^{T}$ and $\nabla_{z} f(\mathbf{x})=\left[\partial f / \partial z_{1}(\mathbf{x}), \ldots, \partial f / \partial z_{m-n}(\mathbf{x})\right]^{T}$. Then these gradients satisfy

$$
E\left[\nabla_{y} f(\mathbf{x})^{T} \nabla_{y} f(\mathbf{x})\right]=\sum_{i=1}^{n} \lambda_{i}
$$

and

$$
E\left[\nabla_{z} f(\mathbf{x})^{T} \nabla_{z} f(\mathbf{x})\right]=\sum_{i=n+1}^{m} \lambda_{i}
$$

Proof: See [8]
From Lemma 3 we see that the variation along each subspace $\mathbb{Y}$ and $\mathbb{Z}$ is dependent on the corresponding eigenvalues. Furthermore, if $\sum_{i=n+1}^{m} \lambda_{i}=0$ then $\nabla_{z} f(\mathbf{x}), \forall \mathbf{x} \in \mathbb{X}$, and in this case we call the function $\mathbf{z}$-invariant.

### 2.5.3 Approximation

We have now decomposed our subspace $\mathbb{X}$ into $\mathbb{Y}$ where $f(\mathbf{x})$ changes relatively much and $\mathbb{Z}$ where $f(\mathbf{x})$ hopefully changes very little or ideally nothing at all. The next step is to approximate $f(\mathbf{x})$ by a function $G(\mathbf{y})$ that only considers points in $\mathbb{Y}$, the reason for this is that the lower dimension of $\mathbb{Y}$ compared to $\mathbb{X}$ allows us to avoid the curse of dimensionality. It is then feasible to build a response surface $\tilde{G}(\mathbf{y})$ on $\mathbb{Y}$ which can be used to approximate our original function as $f(\mathbf{x}) \approx \tilde{G}\left(\mathbf{W}_{1}^{T} \mathbf{x}\right)$.
Starting off we redefine our density function $\rho(\mathbf{x})$ as the density $\pi$ of the coordinates $\mathbf{y}$ and $\mathbf{z}$ as

$$
\pi(\mathbf{y}, \mathbf{z}) \equiv \rho(\mathbf{x}(\mathbf{y}, \mathbf{z}))=\rho\left(\mathbf{W}_{1} \mathbf{y}+\mathbf{W}_{2} \mathbf{z}\right)
$$

from which we obtain the marginal densities as $\pi(\mathbf{y})=\int_{\mathbb{Z}} \pi(\mathbf{y}, \mathbf{z}) d \mathbf{z}$ and $\pi(\mathbf{z})=\int_{\mathbb{Y}} \pi(\mathbf{y}, \mathbf{z}) d \mathbf{y}$, while the conditional densities are given by $\pi(\mathbf{y} \mid \mathbf{z})=\pi(\mathbf{y}, \mathbf{z}) / \pi(\mathbf{z})$ and $\pi(\mathbf{z} \mid \mathbf{y})=\pi(\mathbf{y}, \mathbf{z}) / \pi(\mathbf{y})$. We can now define $G: \mathbb{Y} \mapsto \mathbb{R}$ as

$$
\begin{equation*}
G(\mathbf{y}) \equiv E[f \mid \mathbf{y}]=\int_{\mathbb{Z}} f\left(\mathbf{W}_{1} \mathbf{y}+\mathbf{W}_{2} \mathbf{z}\right) \pi(\mathbf{z} \mid \mathbf{y}) d \mathbf{z} \tag{2.3}
\end{equation*}
$$

where we can use $G$ to approximate $f$ as

$$
f(\mathbf{x}) \approx G\left(\mathbf{W}_{1}^{T} \mathbf{x}\right)
$$

The approximation error is given by the following theorem.

Theorem 5 The mean square error of the approximation of $f(\mathbf{x})$ by $G\left(\mathbf{W}_{1}^{T} \mathbf{x}\right)$ is given by

$$
E\left[\left(f(\mathbf{x})-G\left(\mathbf{W}_{1}^{T} \mathbf{x}\right)\right)^{2}\right] \leq C_{1}\left(\sum_{i=n+1}^{m} \lambda_{i}\right)
$$

where $C_{1}$ is a constant that is dependent on the domain $\mathbb{X}$ and the density function $\rho$. Proof: See [8].

From Theorem 5 combined with Lemma 3 we see that a smaller value of $\nabla_{z} f(\mathbf{x})$ implies a better approximation, and if the function $f$ is $\mathbf{z}$-invariant we will perfectly approximate $f(\mathbf{x})$ by $G\left(\mathbf{W}_{1} \mathbf{x}\right)$ on all of $\mathbb{X}$.

### 2.5.4 Response Surface

We now have an approximation of the function $f$ on $\mathbb{X}$ by the lower dimensional function $G$ on $\mathbb{Y} \subset \mathbb{X}$. Our aim now is to build a polynomial response surface $\tilde{G}: \mathbb{Y} \mapsto \mathbb{R}$ by sampling $P$ design points from $G(\mathbf{y})$ according to some preset design. These design points $\mathbf{y}_{k} \in \mathbb{Y} \subseteq \mathbb{R}^{n}, k=1, \ldots, P$, along with their corresponding evaluations $G_{k} \equiv$ $G\left(\mathbf{y}_{k}\right)$ are then used by an appropriate response surface model $\mathrm{M}:\left\{\left(\mathbf{y}_{k}, G_{k}\right)\right\}_{k=1}^{P} \mapsto \tilde{G}$ to build our response surface $\tilde{G}$. This response surface can then be used to approximate $f$ as $f(\mathbf{x}) \approx \tilde{G}\left(\mathbf{W}_{1}^{T} \mathbf{x}\right)$.

There are many different types of available response surface models that can be used [8]. Here we choose to look at the polynomial regression model of degree $d$ due to its simplicity. It is described by

$$
\mathrm{G}=\mathrm{Z} \beta+\boldsymbol{\epsilon}
$$

where $\mathbf{G}=\left[G_{1}, \ldots, G_{P}\right]^{T}$ are our sample evaluations, $\boldsymbol{\beta}=\left[\beta_{0}, \ldots, \beta_{n d}\right]^{T}$ are our coefficients, $\boldsymbol{\epsilon}=\left[\epsilon_{1}, \ldots, \epsilon_{P}\right]^{T}$ are the error terms which we assume to be distributed as $\epsilon_{i} \stackrel{i i d}{\in} \mathbb{N}\left(0, \sigma^{2}\right)$, and the Vandermonde matrix $\mathbf{Z} \in \mathbb{R}^{P \times(1+n d)}$, is given by

$$
\mathbf{Z}=\left(\begin{array}{cccccc}
1 & y_{1,1} & y_{1,1}^{2} & \cdots & y_{1, n}^{(d-1)} & y_{1, n}^{d} \\
1 & y_{2,1} & y_{2,1}^{2} & \cdots & y_{2, n}^{(d-1)} & y_{2, n}^{d} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
1 & y_{P, 1} & y_{P, 1}^{2} & \cdots & y_{P, n}^{(d-1)} & y_{P, n}^{d}
\end{array}\right)
$$

as our design matrix where the indices are given by $\mathbf{y}_{k}=\left[y_{k, 1}, \ldots, y_{k, n}\right]$. We then estimate $\boldsymbol{\beta}$ as $\hat{\boldsymbol{\beta}}$ by ordinary least squares as

$$
\hat{\boldsymbol{\beta}}=\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\|\mathbf{Z} \boldsymbol{\beta}-\mathbf{G}\|_{2}^{2}=\left(\mathbf{Z}^{T} \mathbf{Z}\right)^{-1} \mathbf{Z}^{T} \mathbf{G} .
$$

We can now describe our response surface as $\tilde{G}(\mathbf{y})=\mathbf{y}_{(\text {aug })}^{T} \hat{\boldsymbol{\beta}}$ where $\mathbf{y}_{(\text {aug })}=\left[1, y_{1}, y_{2}^{2}, \ldots, y_{n}^{d}\right]^{T}$ is the augmented vector of a new point $\mathbf{y}=\left[y_{1}, \ldots, y_{n}\right]^{T}$. Now, by using the response surface we can approximate the original function as $f(\mathbf{x}) \approx \tilde{G}\left(\mathbf{W}_{1}^{T} \mathbf{x}\right)=\left(\mathbf{x}^{T} \mathbf{W}_{1}\right)_{(\text {aug })} \hat{\boldsymbol{\beta}}$.

### 2.5.5 Monte Carlo Method

Since the function $f$ might be very complex, or even unknown, we can not rely on calculating integrals containing $f$ analytically. Thus we need another way to compute $\mathbf{C}$ in equation (2.2) and $G(\mathbf{y})$ in equation (2.3). Here we will use a Plain Monte Carlo[9, p. 13-14] approach to approximate these two integrals. Starting off, we can find the Monte Carlo approximation of $\mathbf{C}$ as

$$
\begin{equation*}
\hat{\boldsymbol{C}}=\frac{1}{M} \sum_{i=1}^{M} \nabla_{x} f\left(\mathbf{x}_{i}\right) \nabla_{x} f\left(\mathbf{x}_{i}\right)^{T} \tag{2.4}
\end{equation*}
$$

where $\mathbf{x}_{i}, i=1, \ldots, M$, are samples drawn from $\mathbb{X}$ according to the probability density $\rho(\mathbf{x})$. Similarly as before, we can then continue by eigen-decomposing this $\hat{\boldsymbol{C}}$ as

$$
\hat{\boldsymbol{C}}=\hat{\boldsymbol{W}} \hat{\boldsymbol{\Lambda}} \hat{\boldsymbol{W}}^{T}
$$

where we further decompose the eigenvector matrix as

$$
\hat{\boldsymbol{W}}=\left[\hat{\boldsymbol{W}}_{1} \hat{\boldsymbol{W}}_{2}\right]
$$

and the eigenvalue matrix as

$$
\hat{\Lambda}=\left(\begin{array}{cc}
\hat{\Lambda}_{1} & \mathbf{0} \\
\mathbf{0} & \hat{\boldsymbol{\Lambda}}_{2}
\end{array}\right)
$$

We can now tackle the second integral regarding $G(\mathbf{y})$ in equation (2.3) with the Monte Carlo approach. This is done as

$$
G(\mathbf{y}) \approx \hat{G}(\mathbf{y})=\frac{1}{N} \sum_{i=1}^{N} f\left(\hat{\boldsymbol{W}}_{1} \mathbf{y}+\hat{\boldsymbol{W}}_{2} \mathbf{z}_{i}\right)
$$

where $\mathbf{z}_{i}$ are drawn from the conditional density $\pi(\mathbf{z} \mid \mathbf{y})$. We can then use $\hat{G}(\mathbf{y})$ instead of $G(\mathbf{y})$ to build a response surface as explained in section 2.5.4 above.

### 2.5.6 Ridge Functions and Implications

A special class of functions, called ridge functions, constitutes the case when the active subspace method works really well. The ridge functions are defined as functions $f$ : $\mathbb{R}^{m} \mapsto \mathbb{R}$ of the form [10]

$$
f(\mathbf{x})=g\left(\mathbf{a}^{T} \mathbf{x}\right)
$$

where $g: \mathbb{R} \mapsto \mathbb{R}$ and $\mathbf{a}=\left[a_{1}, \ldots, a_{m}\right]^{T} \in \mathbb{R}^{m} \backslash\{\mathbf{0}\}$ is a fixed direction where we assume that a has unit length as $\|\mathbf{a}\|=1$. Here we can realize that if we are starting from $\mathbf{x}$ and move a distance $c \in \mathbb{R}$ in the direction of any unit vector $\mathbf{b}$ that is orthogonal of $\mathbf{a}$, $(\mathbf{a} \cdot \mathbf{b}=0)$, the function value will stay the same. That is $f(\mathbf{x}+c \mathbf{b})=g(\mathbf{a} \cdot(\mathbf{x}+c \mathbf{b}))=$ $g(\mathbf{a} \cdot \mathbf{x}+c(\mathbf{a} \cdot \mathbf{b}))=g(\mathbf{a} \cdot \mathbf{x})=f(\mathbf{x})$.
When $f$ is a ridge function, its gradient can be expressed as

$$
\nabla f(\mathbf{x})=\left(\frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}\right) \mathbf{a}
$$

where we note that $\left(\frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}\right)$ is a constant. The derivation of this result can be seen in appendix $D$. If we now use this gradient in the computation of $\mathbf{C}$ in equation (2.2) we get

$$
\begin{aligned}
\mathbf{C} & =\int_{\mathbf{X}} \nabla_{x} f(\mathbf{x}) \nabla_{x} f(\mathbf{x})^{T} \rho(\mathbf{x}) d \mathbf{x} \\
& =\mathbf{a a}^{T} \int_{\mathbb{X}}\left(\frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}\right)^{2} \rho(\mathbf{x}) d \mathbf{x} \\
& =k \mathbf{a a}^{T}
\end{aligned}
$$

where $k$ is a constant that is dependent on the function $g$, the fixed direction a, the density $\rho$ and the domain $\mathbb{X}$. Since we know that $\mathbf{C}$ is a symmetric matrix, it will exhibit the spectral decomposition $\mathbf{C}=\lambda_{1} \mathbf{q}_{1} \mathbf{q}_{1}^{T}+\cdots+\lambda_{m} \mathbf{q}_{m} \mathbf{q}_{m}^{T}$. Since this matrix is of rank 1, it will only have one non-zero eigenvalue $\lambda_{1}=k$ with corresponding eigenvector $\mathbf{q}_{1}=\mathbf{a}$, while $\lambda_{2}=\cdots=\lambda_{m}=0$. From theorem 5 we can then see that our approximation $G\left(\mathbf{W}_{1}^{T} \mathbf{x}\right)$ of $f(\mathbf{x})$ is exact, even if we choose the number of active directions to be $n=1$.
Similar connections between the active subspace method and ridge functions have been done before [11], [12].

### 2.6 Stochastic Subspace Descent

Stochastic subspace descent (SSD) is a type of gradient descent method where we in each iteration consider our descent in a randomized $l$-dimensional subspace of the original $d$ dimensional space where $l<m$. The gradient is then projected upon this subspace, thereafter an fixed-sized step is taken in $l$-dimensional space in the direction where the target function decreasing the most. The reason why this method is advantageous in the context of blackbox optimization is that by projecting onto a lower-dimensional space we have to do fewer function evaluations for each iteration step to obtain our estimated
gradient via finite differentiation. A special case of this method arises when we choose $l=1$ and restrict our search directions to only be along the coordinate axes, we then get the so called stochastic coordinate descent (SCD) algorithm.

### 2.6.1 Iteration Process

The unconstrained optimization problem we are considering here is of the form described in equation (2.1), but with $\mathbb{X}=\mathbb{R}^{m}$. To perform the iteration process, we are starting out with an initial guess $\mathbf{x}_{0}$ and then updating our guess according to the following iteration scheme

$$
\begin{equation*}
\mathbf{x}_{k+1}=\mathbf{x}_{k}-\alpha \mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right) \tag{2.5}
\end{equation*}
$$

where $\alpha$ is our current step size, $\mathbf{P}_{k}=\mathbf{M}_{k} \mathbf{M}_{k}^{T}$ is the projection matrix at step $k$ that project onto the randomized $l$-dimensional subspace and $\mathbf{M}_{k} \in \mathbb{R}^{m \times l}$ is a random matrix. Note here that when calculating the gradient it can (and should) be computed along the directions of the random subspace as $\left(\mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right)$ since we then can avoid computing the full gradient.

### 2.6.2 Finding the Projection Matrix

For $\mathbf{P}_{k}$ to be a suitable projection matrix we need the random matrices $\mathbf{M}_{k}$ to have the properties $E \mathbf{M}_{k} \mathbf{M}_{k}^{T}=\mathbf{I}_{m}$ (such that $E\left[\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right]=\nabla f\left(\mathbf{x}_{k}\right)$ ) and $\mathbf{M}_{k}^{T} \mathbf{M}_{k}=\frac{m}{l} \mathbf{I}_{l}$ (columns of $\mathbf{M}_{k}$ orthogonal), and one such matrix $\mathbf{M}_{k}$ is the scaled Haar distributed matrix [13]. For instructions on how to generate the Haar matrix, the reader is referred to [14].

### 2.6.3 Stochastic Coordinate Descent

Stochastic coordinate descent is the special case of stochastic subspace descent where we in each iteration step limit the subspace to a one-dimensional space along the direction of one of the coordinate axes $i_{k}$. The random coordinate direction $i_{k}$ for the $k$ :th iteration step could be drawn from a random distribution, however this distribution doesn't necessarily have to be uniform. The gradient is then calculated along this random coordinate direction as $\left[\nabla f\left(\mathbf{x}_{k}\right)\right]_{i_{k}}$ whereby we arrive at the iteration scheme

$$
\mathbf{x}_{k+1}=\mathbf{x}_{k}-\alpha\left[\nabla f\left(\mathbf{x}_{k}\right)\right]_{i_{k}}
$$

Relating this to equation (2.5) we notice that this is equivalent to the Stochastic Subspace Descent scheme if we are considering the projection matrix $\mathbf{P}_{k}=\mathbf{M}_{k} \mathbf{M}_{k}^{T}$ with $\mathbf{M}_{k} \in$ $\mathbb{R}^{m \times 1}$ where

$$
\left[M_{k}\right]_{i}=\left\{\begin{array}{ll}
\sqrt{m}, & \text { if } i=i_{k} \\
0, & \text { if } i \neq i_{k}
\end{array}, i=\{1, \ldots, m\}\right.
$$

Notice here that the random coordinates are drawn from the uniform distribution $P\left(i_{k}=\right.$ $i)=1 / m, \forall i$. The convergence of the stochastic coordinate descent is thus expressed by theorems (6-8), with the random subspace now being of dimension $l=1$.

### 2.6.4 Convergence and Rate of Convergence for Stochastic Subspace Descent Without Noise

In [13] the following assumptions are made regarding convergence and rate of convergence:
(A0): the matrices $\mathbf{M}_{k}$ are independent random matrices such that $E \mathbf{M}_{k} \mathbf{M}_{k}^{T}=\mathbf{I}_{m}$ and $\mathbf{M}_{k}^{T} \mathbf{M}_{k}=\frac{m}{l} \mathbf{I}_{l}$ where $m>l$
(A1): the function $f(\mathbf{x})$ has a Lipschitz-continuous gradient with parameter $L$
(A2): the minimum value of $f$ is $f^{*}$, which is considered known and bounded as $f^{*}>-\infty$
(A3): for some $0<\gamma \leq L$ and for every $\mathbf{x}$ the function satisfy the Polyak-Lojasiewicz (PL) inequality

$$
f(\mathbf{x})-f^{*} \leq \frac{\|\nabla f(\mathbf{x})\|^{2}}{2 \gamma}
$$

(A4): $f$ is strongly convex with parameter $\gamma>0$ for all $\mathbf{x} \in \mathbb{R}^{d}$, where $L \geq \gamma$
(A5): $f$ is convex and the optimal value is achieved on the domain $\mathbb{D}$ such that $\mathbf{x}^{*} \in \mathbb{D}$. Also, there exists a constant $R>0$ which bound the distance to the optimal solution for solutions better than the initial guess $\mathbf{x}_{0}$ as

$$
\max _{\mathbf{x}, \mathbf{x}^{*}}\left\{\left\|\mathbf{x}-\mathbf{x}^{*}\right\|: f(\mathbf{x}) \leq f\left(\mathbf{x}_{0}\right)\right\} \leq R
$$

Remembering that a sequence of random vectors $\{\mathbf{s}\}_{i}$ is said to converge almost surely to the random vector $\mathbf{s}^{*}$ if and only if $P\left(\mathbf{s}_{i} \rightarrow \mathbf{s}^{*}\right)=1$, which we denote as $\mathbf{s}_{i} \xrightarrow{\text { a.s. }} \mathbf{s}^{*}$. Similarly, the sequence is said to converge in $L_{1}$ if and only if $E\left\|\mathbf{s}_{i}-\mathbf{s}^{*}\right\|_{1} \rightarrow 0$, which we denote as $\mathbf{s}_{i} \xrightarrow{L_{7}} \mathbf{s}^{*}$. The first convergence theorem which ensures convergence and states the rate of convergence is now presented, under the assumptions of the PL-inequality and continuous differentability of $f(\mathbf{x})$. For proofs of theorems 6-8, see [13].

Theorem 6 (Convergence and rate of convergence under the PL inequality) Assume A0, A1, A2, A3 hold and that $\mathbf{x}_{0}$ is our arbitrary and known initialization. Then the iteration scheme presented in equation (2.5) with step size $0<\alpha<2 l / L m$ leads to almost surely convergence as $f\left(\mathbf{x}_{k}\right) \xrightarrow{\text { a.s. }} f^{*}$ and convergence in $L_{1}$ as $f\left(\mathbf{x}_{k}\right) \xrightarrow{L_{7}} f^{*}$. By choosing $\alpha=l / L d$ the expected rate of convergence is given by $0 \leq E\left(f\left(\mathbf{x}_{k}\right)\right)-f^{*} \leq$ $\beta^{k}\left(f\left(\mathbf{x}_{0}\right)-f^{*}\right)$ where $\beta=(1-l \gamma / m L)$.

If we instead of the PL-inequality assume strong convexity, the following theorem ensure that the stochastic subspace descent method converges.

Theorem 7 (Convergence under strong convexity) Assume A0, A1, A2, A4 hold and that $\mathbf{x}_{0}$ is our arbitrary initialization. Then the iteration scheme presented in equation (2.5) with step size $0<\alpha<2 l /$ Lm leads to almost surely convergence as $\mathbf{x}_{k} \xrightarrow{\text { a.s. }} \mathbf{x}^{*}$ where $\mathbf{x}^{*}$ is the unique minimizer of $f^{*}$.

Finally, the theorem of rate of convergence is stated, where we now assume convexity and that all solutions distance between the minimizer and all better than the initial guess is bounded by the value $R$.

Theorem 8 (Rate of convergence under convexity) Assume A0, A1, A2, A5 hold and that $\mathbf{x}_{0}$ is our arbitrary initialization. Then the iteration scheme presented in equation (2.5) with step size $0<\alpha<2 l / L d$ converges as $E\left(f\left(\mathbf{x}_{k}\right)\right)-f^{*} \leq 2 m L R^{2} / k l$.

### 2.6.5 Convergence for Stochastic Subspace Descent Under NonConvex Assumption

We here aim to present a general convergence result with minimal assumptions regarding convexity. In other words, the results presented here avoid assuming any kind of convexity nor the PL inequality. Furthermore, two of the results also assume noise in the gradient. To start off, the convergence of the gradient descent with a noisy gradient is presented.

Theorem 9 (Convergence of gradient descent with noisy gradient) Let $f: \mathbb{R}^{m} \mapsto$ $\mathbb{R}$ be the function to be minimized, and let this function have a Lipschitz continuous gradient with parameter $L$. Then let $\epsilon_{k} \sim \rho$ be the random noise vector added to the gradients of $f$, where $\rho$ is an arbitrary distribution, such that our available gradient becomes $g\left(\mathbf{x}_{k}\right)=\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}$. Assume that this noise is unbiased $E\left[\boldsymbol{\epsilon}_{k}\right]=\mathbf{0}$ and that its variance is bounded as $E\left[\left\|\boldsymbol{\epsilon}_{k}\right\|^{2}\right] \leq \sigma^{2}$, for all $k$. Now, let our gradient descent scheme be

$$
\mathbf{x}_{k+1}=\mathbf{x}_{k}-\alpha_{k} g\left(\mathbf{x}_{k}\right) .
$$

If our step size $\alpha_{k}$ for all $k$ satisfy $\alpha_{k} \leq \frac{1}{2 L}, \lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}^{2}<\infty$ and $\lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}=\infty$ then one iterate $\mathbf{x}_{k}$ will be a stationary point such that $\nabla f\left(\mathbf{x}_{k}\right)=0$, for some $k, 0 \leq$ $k \leq K$, when $K \rightarrow \infty$.

Proof: See proof of theorem 6.1 in [15, p. 307-313].
Now we present a result where instead of a noisy gradient, we assume that the gradient is approximated by a stochastic subspace. This result is achieved by a straightforward modification of the proof in Theorem 9.

Theorem 10 (Convergence of stochastic subspace descent) Let $f: \mathbb{R}^{m} \mapsto \mathbb{R}$ be the function to be minimized, and let this function have a Lipschitz continuous gradient with parameter $L$. Then let our available gradient be $g\left(\mathbf{x}_{k}\right)=\mathbf{P}_{k} \nabla f\left(\mathbf{x}_{k}\right)=$
$\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)$, where $\mathbf{M}_{k} \in \mathbb{R}^{m \times l}$ are the random matrices and the $\mathbf{P}_{k}:$ s are the projection matrices onto our rank l subspaces. Assume that these random matrices for all iterations satisfy $E\left[\mathbf{M}_{k} \mathbf{M}_{k}^{T}\right]=\mathbf{I}_{m}$ and $\mathbf{M}_{k}^{T} \mathbf{M}_{k}$. Now, let our gradient descent scheme be

$$
\mathbf{x}_{k+1}=\mathbf{x}_{k}-\alpha_{k} g\left(\mathbf{x}_{k}\right)
$$

If our step size $\alpha_{k}$ for all $k$ satisfy $\alpha_{k} \leq \frac{l}{2 m L}$ and $\lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}=\infty$ then one iterate $\mathbf{x}_{k}$ will be a stationary point such that $\nabla f\left(\mathbf{x}_{k}\right)=0$, for some $k, 0 \leq k \leq K$, when $K \rightarrow \infty$.

Proof: See Appendix E.
We will now combine the assumptions from the previous two theorems by assuming that the gradients are both noisy and approximated by a stochastic subspace. This result is achieved by a straightforward modification of the proof in Theorem 9.

Theorem 11 (Convergence of stochastic subspace descent with noisy gradient)
Let $f: \mathbb{R}^{m} \mapsto \mathbb{R}$ be the function to be minimized, and let this function have a Lipschitz continuous gradient with parameter $L$. Then let $\boldsymbol{\epsilon}_{k} \sim \rho$ be the random noise vector added to the stochastic subspace approximated gradients of $f$, where $\rho$ is an arbitrary distribution, such that our available gradient becomes $g\left(\mathbf{x}_{k}\right)=\mathbf{P}_{k}\left(\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}\right)=$ $\mathbf{M}_{k} \mathbf{M}_{k}^{T}\left(\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}\right)$. Here $\mathbf{M}_{k} \in \mathbb{R}^{m \times l}$ are the random matrices and the $\mathbf{P}_{k}$ :s are the projection matrices onto our rank l subspaces. Assume that these random matrices for all iterations satisfy $E\left[\mathbf{M}_{k} \mathbf{M}_{k}^{T}\right]=\mathbf{I}_{m}$ and $\mathbf{M}_{k}^{T} \mathbf{M}_{k}$. Furthermore, assume that the noise is unbiased $E\left[\boldsymbol{\epsilon}_{k}\right]=\mathbf{0}$ and that its variance is bounded as $E\left[\left\|\boldsymbol{\epsilon}_{k}\right\|^{2}\right] \leq \sigma^{2}$, for all $k$. Also, assume $\mathbf{M}_{k}$ and $\boldsymbol{\epsilon}_{k}$ to be independent for all $k$. Now, let our gradient descent scheme be

$$
\mathbf{x}_{k+1}=\mathbf{x}_{k}-\alpha_{k} g\left(\mathbf{x}_{k}\right)
$$

If our step size $\alpha_{k}$ for all $k$ satisfy $\alpha_{k} \leq \frac{l}{2 m L}, \lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}^{2}<\infty$ and $\lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}=\infty$ then one iterate $\mathbf{x}_{k}$ will be a stationary point such that $\nabla f\left(\mathbf{x}_{k}\right)=0$, for some $k, 0 \leq$ $k \leq K$, when $K \rightarrow \infty$.

Proof: See Appendix F.

### 2.6.6 Armijo Line Search

Instead of using a fixed step length $\alpha$ or a vanishing step size like $\alpha_{k}=\frac{\alpha_{0}}{k}$ which satisfies the conditions in the convergence results above, it is usually more efficient using a line search method in the direction of the gradients to avoid taking either to small or to long steps. Here the ideal step size $\alpha_{k}$, given the computed gradient $g\left(\mathbf{x}_{k}\right)$, would be obtained as a solution to the exact line search problem

$$
\alpha_{k}=\underset{\alpha \geq 0}{\operatorname{argmin}} f\left(\mathbf{x}_{k}-\alpha g\left(\mathbf{x}_{k}\right)\right) .
$$

However, solving the exact line search problem is expensive if one doesn't have knowledge about the underlying function $f$. Hence, inexact line search methods are employed to
identify a suitable step length at a more moderate cost. One such inexact line search method is the backtracking Armijo line search [16] which gives $\alpha_{k}=\gamma_{k} \beta^{m_{k}}, \gamma_{k}>0$, $\beta \in(0,1)$ where $m_{k}$ is the smallest non-negative integer such that

$$
\begin{equation*}
f\left(\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}\right) \leq f\left(\mathbf{x}_{k}\right)+\sigma \alpha_{k} g\left(\mathbf{x}_{k}\right)^{T} \mathbf{d}_{k} \tag{2.6}
\end{equation*}
$$

is fulfilled, where $\sigma \in\left(1, \frac{1}{2}\right)$ and $\mathbf{d}_{k}$ is the descent direction satisfying $g\left(\mathbf{x}_{k}\right)^{T} \mathbf{d}_{k} \leq 0$. The descent direction can for example be chosen to be $\mathbf{d}_{k}=-g\left(\mathbf{x}_{k}\right)+\beta_{k-1} \mathbf{d}_{k-1}$ for the conjugate gradient method, $\mathbf{d}_{k}=-\nabla^{2} f\left(\mathbf{x}_{k}\right) g\left(\mathbf{x}_{k}\right)$ for Newton's method or $\mathbf{d}_{k}=-g\left(\mathbf{x}_{k}\right)$ for gradient descent.

We will now state convergence results for schemes fulfilling the backtracking Armijo line search condition in equation (2.6), first for an arbitrary descent direction and then for a descent direction projected onto a stochastic subspace $\mathbf{P}_{k}$. This second theorem is achieved by a straightforward modification of the proof of the first theorem in order to take the stochastic subspace descent direction into account.

Theorem 12 (Convergence of backtracking Armijo Line search) Assume assumptions $A 1$ and A2 hold. Also, let $\mathbf{d}_{k}$ be a descent direction satisfying $g\left(\mathbf{x}_{k}\right)^{T} \mathbf{d}_{k} \leq 0, \forall k$. If the iteration scheme $\mathbf{x}_{k+1}=\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}$ satisfies the backtracking Armijo line search condition in equation (2.6), and by choosing the initial step size $\gamma_{k}=-\frac{\mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|^{2}}$, then for all $k$

$$
\begin{equation*}
\lim _{k \rightarrow \infty}\left(\frac{g\left(\mathbf{x}_{k}\right)^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|}\right)^{2}=0 \tag{2.7}
\end{equation*}
$$

Proof: See proof of theorem 3.1 in [17].
Note that the convergence property in equation (2.7) is equivalent to the fact that either $\lim _{k \rightarrow \infty}\left\|g\left(\mathbf{x}_{k}\right)\right\|^{2}=0$ or that $\lim _{k \rightarrow \infty} g\left(\mathbf{x}_{k}\right)^{T} \mathbf{d}_{k}=0$ which in turn implies that $\lim _{k \rightarrow \infty} \gamma_{k}=0$.
Theorem 13 (Convergence of SSD scheme with backtracking Armijo line search) Assume assumptions A0, A1 and A2 hold. Also, let $\mathbf{d}_{k}=-\mathbf{M}_{\mathbf{k}} \mathbf{M}_{\mathbf{k}}{ }^{T} g\left(\mathbf{x}_{k}\right)$ be our search direction. If the iteration scheme $\mathbf{x}_{k+1}=\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}$ satisfies the backtracking Armijo line search condition in equation (2.6), and by choosing the initial step size $\gamma=-\frac{\mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|^{2}}$, then for all $k$

$$
\lim _{k \rightarrow \infty}\left\|g\left(\mathbf{x}_{k}\right)\right\|^{2}=0
$$

Proof: See Appendix G.

### 2.7 Design of Experiments

Design of Experiments (DOE) is a set of statistical and mathematical tools designed to help improve process yield, finding optimal conditions, reducing costs and minimizing
development time [18]. This is achieved by sequential experiments with carefully chosen operating conditions. Here we will focus on the part of DOE used for optimization.

### 2.7.1 Response Surface Methodology

Response surface methodology (RSM) is the part of DOE whose objective is to optimize a target function, given a set of adjustable input parameters. This is done by performing experiments in a region of interests called Significant Design Space, which is a space that we will denote by $\mathbb{V}_{S D S}$. Usually this space is a hyper-cube, and can therefore be described by the center point of the space $\mathbf{C} \in \mathbb{R}^{m \times 1}$ and the length in each dimension $\mathbf{L} \in \mathbb{R}^{m \times 1}$. The idea is then to sample points in $\mathbb{V}_{S D S}$ according to a chosen experimental design (also known as response surface design). These sampled design points and corresponding response values, which we denote $S_{S D S}$, will then be used to fit a statistical model that will be used to represent the behaviour inside $\mathbb{V}_{S D S}$. We will then optimize the response by changing the operating conditions, based on the output of this surrogate model, to move our $\mathbb{V}_{S D S}$ towards the location where we think the optimum reside. This methodology is then iterated until we found our optimal conditions. This type of optimization which is based on a surrogate model is known as Metamodel-Based Design Optimization (MBDO)[19].

### 2.7.2 Coded Variables

When working with DOE methodology it is common to code, or normalize, the variables. The reason for doing this is because when coding the variables in different dimensions the variables will then be on the same scale in the $\mathbb{V}_{S D S}$, i.e. between -1 and 1 . Also, coding of the variables will in many cases lead to easier interpretation of the estimated metamodel coefficients. If we consider a point with uncoded, or natural, coordinates $\mathbf{x} \in \mathbb{R}^{m \times 1}$ in the current $\mathbb{V}_{S D S}$ then the transformation to coded coordinates $\mathbf{x}_{\text {code }} \in \mathbb{R}^{m \times 1}$ is given by

$$
\mathbf{x}_{\text {code }}=2(\mathbf{x}-\mathbf{C}) \oslash \mathbf{L}
$$

where $\oslash$ is the Hadamarad matrix division [20] which indicates element-wise division between two matrices.

### 2.7.3 Experimental Designs

In choosing experimental designs, we firstly need to consider which model we want to fit to the sampled points $S_{S D S}$. Secondly, we have to decide which design features that are the most beneficial for our specific problem. Some of these features are:

1. Space filling property, which allows more information to be sampled throughout $\mathbb{V}_{S D S}$
2. Allowing model checking, such as lack of fit tests
3. Allowing extra points to be sampled, retroactively
4. Requiring few points to be sampled
5. Robustness against outliers
6. Allowing the internal error to be measured

## $2^{m}$ Full Factorial Design:

The simplest design is the $2^{m}$ Full Factorial Design, which consists of an equal number of samples from each corner of the $\mathbb{V}_{S D S}$. This design is suitable for fitting first order models where we don't think the underlying function have any curvature and we also want to be economical with our sample points. Examples of this design in two and three dimensions can be seen in Figure 2.2.

(a) $2^{2}$ Full Factorial Design

(b) $2^{3}$ Full Factorial Design

Figure 2.2: Visualization of the $2^{m}$ Full Factorial Designs for two and three dimensions.

## Central Composite Design:

The Central Composite Design (CCD) is one of the most used designs in RSM[18]. This design consists of a full factorial design, but also $n_{C}$ center points and $2 m$ axial points. This model is useful for fitting second order models and also for estimating an internal error due to the repeated measurements of the center points. Another advantage is that if we detect lack of fit of a first order model from the full factorial design, we can simply add the center and axial points to get a central composite design, which can then be used to fit the second order model. Examples of this design in two and three dimensions can be seen in Figure 2.3.

(a) Two dimensional CCD

(b) Three dimensional CCD

Figure 2.3: Visualization of the Central Composite Designs for two and three dimensions.

## Latin Hypercube Design:

The Latin Hypercube Design (LHD) is a design with good space filling properties. In explaining the generation of this design, consider $p$ points to be generated in $m$ dimensions in coded coordinates. Firstly, each dimension is partitioned into $p$ equally spaced intervals, each of which can be described as $I_{k}^{v}=[(k-1) / p, k / p], k=\{1, \ldots, p\}$,
$v=\{1, \ldots, m\}$. The $p$ points are then placed such that there is one point in each $I_{k}^{v}$, which then form the matrix

$$
\mathbf{X}=\left(\begin{array}{ccc}
x_{1,1} & \ldots & x_{1, m} \\
\vdots & \ddots & \vdots \\
x_{p, 1} & \ldots & x_{p, m} .
\end{array}\right)
$$

The columns are then randomly permuted to form the permuted matrix

$$
\mathbf{X}^{\prime}=\left(\begin{array}{ccc}
x_{1,1}^{\prime} & \ldots & x_{1, m}^{\prime} \\
\vdots & \ddots & \vdots \\
x_{p, 1}^{\prime} & \ldots & x_{p, m}^{\prime}
\end{array}\right)
$$

To obtain a permuted matrix $\mathbf{X}^{\prime}$ with points well-spread in space, we can try to minimize the performance measure [21] of pairwise correlations

$$
\rho^{2}=\frac{\sum_{i=2}^{m} \sum_{j=1}^{i-1} \rho_{i j}^{2}}{m(m-1) / 2}
$$

where $\rho_{i j}$ is the pairwise correlation between each coordinate which we estimate with the sample correlation $r_{i j}$ as

$$
r_{i j}=\frac{\sum_{k=1}^{p}\left(x_{k, i}^{\prime}-\bar{x}^{\prime}{ }_{\cdot, i}\right)\left(x_{k, j}^{\prime}-\bar{x}^{\prime}{ }_{\cdot, j}\right)}{\sqrt{\sum_{k=1}^{p}\left(x_{k, i}^{\prime}-\bar{x}^{\prime},{ }^{\prime},\right)^{2}} \sqrt{\sum_{k=1}^{p}\left(x_{k, j}^{\prime}-\bar{x}^{\prime},, j\right)^{2}}} .
$$

However, trying to minimize the performance measure based on pairwise correlations can sometimes produce LHD:s where points lie close to each other. Therefore, another performance measure based on maxmin distance [22] is commonly used. Here the idea is to maximize minimum distance as

$$
\max _{i, j} \min _{i, j}\left(\operatorname{dist}\left(\mathbf{x}_{i,}^{\prime},, \mathbf{x}_{j,}^{\prime},\right)\right), \quad i, j \in\{1, \ldots, p\}, i \neq j
$$

where dist is a distance measure between the two rows.

### 2.7.4 Models

The most commonly used models for fitting RSM surfaces of $m$ dimensions are the first order model

$$
y=\beta_{0}+\sum_{i=1}^{m} \beta_{i} x_{i}+\epsilon
$$

and the second order model

$$
y=\beta_{0}+\sum_{i=1}^{m} \beta_{i} x_{i}+\sum_{i=1}^{m} \beta_{i i} x_{i}^{2}+\sum_{i=j+1}^{m} \sum_{j=1}^{m} \beta_{i j} x_{i} x_{j}+\epsilon
$$

Here $y$ is the response, the $x_{i}$ :s are the predictors, the $\beta$ :s are the coefficients and $\epsilon$ is the internal error of each observation which is assumed to be identically normally distributed as $N\left(0, \sigma^{2}\right)$. The $\beta$ :s can then be estimated as $\hat{\beta}$, for example by using least squares, to give the fitted models

$$
\hat{y}=\hat{\beta}_{0}+\sum_{i=1}^{m} \hat{\beta}_{i} x_{i}
$$

and

$$
\begin{equation*}
\hat{y}=\hat{\beta}_{0}+\sum_{i=1}^{m} \hat{\beta}_{i} x_{i}+\sum_{i=1}^{m} \hat{\beta}_{i i} x_{i}^{2}+\sum_{i=j+1}^{m} \sum_{j=1}^{m} \hat{\beta}_{i j} x_{i} x_{j} \tag{2.8}
\end{equation*}
$$

for the first and second order models, respectively. These fitted models describe a surface in $m$-dimensional space and are considered as the response surface, since it explains the response we expect to get on the surface defined by $\mathbb{V}_{S D S}$. Furthermore, to fit these regression models the number of observations needs to be at least as many as the number of unknown coefficients $\beta$. Given $m$ dimensions we thus have 1 constant term, $m$ linear terms and $m(m-1) / 2$ interaction terms and $m$ purely squared terms. So in order to fit the first order model we need at least

$$
n_{1}=1+m
$$

observations while for the second order model we need at least

$$
\begin{equation*}
n_{2}=1+\frac{3}{2} m+\frac{1}{2} m^{2} \tag{2.9}
\end{equation*}
$$

observations.

### 2.7.5 Canonical Analysis of Second Order Model

Here we are going to describe how to find and classify a stationary point of the response surface of the second order model described in (2.8). That is to find a point where the gradient of $\hat{y}$ is zeros and classify it as a minimum point, a maximum point or a saddle point. This is done so that we find the $\mathbf{x}$ that minimize the response $\hat{y}$, given the parameters $\beta$. Starting off we can by letting

$$
\mathbf{x}=\left[\begin{array}{c}
x_{1} \\
x_{2} \\
\vdots \\
x_{m}
\end{array}\right], \quad \mathbf{b}=\left[\begin{array}{c}
\hat{\beta}_{1} \\
\hat{\beta}_{2} \\
\vdots \\
\hat{\beta}_{m}
\end{array}\right], \quad[\mathbf{B}]_{i j}= \begin{cases}\hat{\beta}_{i i} & \text { if } i=j \\
\hat{\beta}_{i j} / 2 & \text { if } i \neq j\end{cases}
$$

where $\beta_{i j}=\beta_{j i}$ rewrite the fitted second order model as

$$
\hat{y}=\hat{\beta}_{0}+\mathbf{x}^{T} \mathbf{b}+\mathbf{x}^{T} \mathbf{B} \mathbf{x} .
$$

To find the stationary points $\mathbf{x}_{s}$ we differentiate with respect to $\mathbf{x}$ and set the result equal to zero as

$$
\frac{\partial \hat{y}}{\partial \mathbf{x}}=\mathbf{b}+2 \mathbf{B} \mathbf{x}=\mathbf{0} .
$$

Solving for $\mathbf{x}$ then gives the stationary point as

$$
\mathbf{x}_{s}=-\frac{1}{2} \mathbf{B}^{-1} \mathbf{b} .
$$

which plugged into the fitted model gives the estimate of the response at the stationary point as

$$
\begin{aligned}
\hat{y}_{s} & =\hat{\beta}_{0}+\left(-\frac{1}{2} \mathbf{B}^{-1} \mathbf{b}\right)^{T} \mathbf{b}+\left(-\frac{1}{2} \mathbf{B}^{-1} \mathbf{b}\right)^{T} \mathbf{B}\left(-\frac{1}{2} \mathbf{B}^{-1} \mathbf{b}\right) \\
& =\hat{\beta}_{0}-\frac{1}{2} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{b}+\frac{1}{2} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{B} \mathbf{B}^{-1} \mathbf{b} \\
& =\hat{\beta}_{0}-\frac{1}{4} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{b}
\end{aligned}
$$

where we have used that $\mathbf{B}^{-1}$ is symmetric due to the symmetry of $\mathbf{B}$. To easier being able to analyze the response surface we will define a new coordinate system where we have moved the stationary point to the origin. The translated coordinates is then given by

$$
\mathbf{z}=\mathbf{x}-\mathbf{x}_{s} .
$$

Representing the response surface in these new coordinates we get

$$
\begin{aligned}
\hat{y} & =\hat{\beta}_{0}+\left(\mathbf{z}+\mathbf{x}_{s}\right)^{T} \mathbf{b}+\left(\mathbf{z}+\mathbf{x}_{s}\right)^{T} \mathbf{B}\left(\mathbf{z}+\mathbf{x}_{s}\right) \\
& =\hat{\beta}_{0}+\left(\mathbf{z}-\frac{1}{2} \mathbf{B}^{-1} \mathbf{b}\right)^{T} \mathbf{b}+\left(\mathbf{z}-\frac{1}{2} \mathbf{B}^{-1} \mathbf{b}\right)^{T} \mathbf{B}\left(\mathbf{z}-\frac{1}{2} \mathbf{B}^{-1} \mathbf{b}\right) \\
& =\hat{\beta}_{0}+\mathbf{z}^{T} \mathbf{b}-\frac{1}{2} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{b}+\mathbf{z}^{T} \mathbf{B} \mathbf{z}-\frac{1}{2} \mathbf{z}^{T} \mathbf{B} \mathbf{B}^{-1} \mathbf{b}-\frac{1}{2} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{B} \mathbf{z}+\frac{1}{4} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{B} \mathbf{B}^{-1} \mathbf{b} \\
& =\hat{\beta}_{0}+\mathbf{z}^{T} \mathbf{b}-\frac{1}{2} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{b}+\mathbf{z}^{T} \mathbf{B} \mathbf{z}-\frac{1}{2} \mathbf{z}^{T} \mathbf{b}-\frac{1}{2} \mathbf{b}^{T} \mathbf{z}+\frac{1}{4} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{b} \\
& =\hat{\beta}_{0}-\frac{1}{4} \mathbf{b}^{T} \mathbf{B}^{-1} \mathbf{b}+\mathbf{z}^{T} \mathbf{B} \mathbf{z} \\
& =\hat{y}_{s}+\mathbf{z}^{T} \mathbf{B} \mathbf{z}
\end{aligned}
$$

where we again used the fact that the inverse of a symmetric matrix is symmetric and also that $\mathbf{z}^{T} \mathbf{b}=\mathbf{b}^{T} \mathbf{z}$. We can now see that the response surface expressed in $\mathbf{z}$-coordinates don't have any linear terms, but only the constant term and the second order terms. We now want to rotate our coordinate system such that the principal axes align with the coordinate axes. To do this we first eigen-decompose $\mathbf{B}$ as $\mathbf{B}=\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^{T}$, where $\mathbf{Q}$ is the orthogonal matrix with the eigenvectors of $\mathbf{B}$ in its columns and $\boldsymbol{\Lambda}$ is the diagonal matrix with the corresponding eigenvalues on its diagonal. We then rotate our translated coordinate system such that the coordinate axes align with the principal axes of $\mathbf{B}$ to get the rotated coordinates

$$
\mathbf{w}=\mathbf{Q}^{T} \mathbf{z}
$$

Since $\mathbf{Q}$ is an orthogonal matrix, $\mathbf{Q}^{T}=\mathbf{Q}^{-1}$, which means that the rotated coordinates can be re-written as $\mathbf{w}=\mathbf{Q}^{-1} \mathbf{z}$ which implies that the translated coordinates can be written as $\mathbf{z}=\mathbf{Q w}$. Plugging in this expression of $\mathbf{z}$ in the response surface equation and using the eigen-decomposition of $\mathbf{B}$ we get

$$
\begin{aligned}
\hat{y} & =\hat{y}_{s}+(\mathbf{Q w})^{T} \mathbf{B}(\mathbf{Q w}) \\
& =\hat{y}_{s}+\mathbf{w}^{T} \mathbf{Q}^{T} \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^{T} \mathbf{Q} \mathbf{w} \\
& =\hat{y}_{s}+\mathbf{w}^{T} \boldsymbol{\Lambda} \mathbf{w} \\
& =\hat{y}_{s}+\sum_{i=1}^{m} \lambda_{i} w_{i}^{2}
\end{aligned}
$$

which is the canonical form of the response model. Since the terms $w_{i}^{2}$ are all positive and the $\lambda_{i}$ are real due to the symmetry of $\mathbf{B}$ we can now easily charaterize our stationary point. If all eigenvalues are positive we can see that $\hat{y}$ will increase no matter which direction from $\mathbf{x}_{s}$ we move in, hence $\mathbf{x}_{s}$ is a minimum point. If on the other hand all eigenvalues are negative $\hat{y}$ will instead decrease if we move away from $\mathbf{x}_{s}$ and our stationary point is in this case a maximum point. Finally, if the eigenvalues are both positive and negative some directions will decrease our response value while others will increase it, hence $\mathbf{x}_{s}$ is a saddle point.

Furthermore, from this analysis we can also see how much the estimated response change depending on which principal direction we move in based on the eigenvalues since a larger absolute value of the eigenvalue imply that the response change more in the corresponding direction. Also, if only one of the $m$ eigenvalues are non-zero, say $\lambda_{j}$, we can see that the fitted response surface is a ridge function in the direction of the $j$ :th column of $\mathbf{Q}$.

For further reading, the interested reader is referred to [23].

### 2.7.6 Trust Region Subproblem

However, when trying to to optimize the second order surrogate model in our $\mathbb{V}_{S D S}$ by using the canonical analysis the found stationary point could be a maximum value, a saddle point or be located outside of our $\mathbb{V}_{S D S}$ where the result of the model lacks validity. Hence we need another way of optimizing the surrogate model. We are here borrowing concepts from trust region methodology for solving the so called trust region subproblem which is to find

$$
\begin{aligned}
& \min g(\mathbf{x})=\mathbf{x}^{T} \mathbf{b}+\mathbf{x}^{T} \mathbf{B} \mathbf{x}, \\
& \text { s.t. }\|\mathbf{x}\|_{2} \leq \Delta
\end{aligned}
$$

where $\Delta$ is a scalar denoting the maximum radius and $\mathbf{x}$ is the trust region step. Note here that we are using a circular region of interest centered around origo which is the so called 'trust region' where we are searching for our optimal conditions instead of a square one, the reason for which will be apparent shortly. Also, note that the model $g(\mathbf{x})$ is equivalent to the second order model except for the constant term which mean that the locations of the corresponding minimizers will coincide.

## Cauchy point

A straight-forward way of tackling the trust region subproblem is to calculate Cauchy point. If we let the gradient of the centre be $\mathbf{g}=\nabla g(\mathbf{0})=\mathbf{b}+2 \mathbf{B 0}=\mathbf{b}$ and the direction of the steepest descent be $\mathbf{p}^{S}=-\mathbf{g} /\|\mathbf{g}\|$ then the definition of the Cauchy point is
Definition 17 (Cauchy point) The Cauchy point $\mathbf{p}^{C}$ is defined as the point along the steepest descent from the origin which minimizes $g(\mathbf{d})$ inside the trust region, i.e.
$\mathbf{p}^{C} \equiv \tau \mathbf{p}^{S}$ where

$$
\tau=\operatorname{argmin}\left\{g\left(\tau \mathbf{p}^{S}\right):\left\|\tau \mathbf{p}^{S}\right\| \leq \Delta\right\} .
$$

The following theorem states how to calculate the Cauchy point.
Theorem 14 (Cauchy point) The Cauchy point is given by

$$
\mathbf{p}^{C}=\tau \mathbf{p}^{S}
$$

where

$$
\tau= \begin{cases}\min \left\{\Delta, \frac{\|\mathbf{g}\|^{3}}{2 \mathbf{2 g}^{T} \mathbf{B g}}\right\} & \text { if } \mathbf{g}^{T} \mathbf{B g}>0 \\ \Delta & \text { if } \mathbf{g}^{T} \mathbf{B g} \leq 0\end{cases}
$$

Proof: See Appendix H.

## Conjugate gradient method

The Cauchy point is a cheap and straightforward way of minimizing $g(\mathbf{x})$ the trust region subproblem, but we can do better. The Cauchy point algorithm can be improved by using conjugate gradient methodology which will be explained here. But since the conjugate gradient algorithm doesn't take into account the maximum allowed step length and it assumes positive-definiteness on the second order terms which might not hold, it has to be adjusted to be appropriate to the trust region subproblem. This adjustments will then be showcased, which the will result in the Steinhaug's conjugate gradient method which then will be described.

The conjugate gradient method is an iterative algorithm for solving

$$
\begin{equation*}
A x=b \tag{2.10}
\end{equation*}
$$

where $\mathbf{x}$ is unknown and $\mathbf{A}$ is symmetric and positive definite. Relating this formulation to the trust region subproblem, where we can set $\mathbf{B}=-\frac{1}{2} \mathbf{A}$ to obtain $g(\mathbf{x})=\mathbf{x}^{T} \mathbf{b}-$ $\frac{1}{2} \mathbf{x}^{T} \mathbf{A x}$, we see that in finding the stationary point by setting the gradient to zero $\nabla g(\mathbf{x})=\mathbf{b}-\mathbf{A} \mathbf{x}=\mathbf{0}$ we arrive at the same problem as in equation (2.10).

The main idea is to generate a sequence of steps $D=\left\{\mathbf{d}_{1}, \ldots, \mathbf{d}_{m}\right\}$ and sequentially update the guess of $\mathbf{x}$ as

$$
\mathbf{x}_{i+1}=\mathbf{x}_{i}+\alpha_{i} \mathbf{d}_{i}
$$

where $\alpha_{i}$ is the corresponding step sizes. Furthermore, we also require that the search directions $\mathbf{d}_{i}$ :s are mutually conjugate with respect to $\mathbf{A}$.

Definition 18 (Conjugate vectors) Vectors $\mathbf{d}_{i}$ and $\mathbf{d}_{j}, i \neq j$, are said to be conjugate with respect to A, or A-orthogonal, if

$$
\left\langle\mathbf{d}_{i}, \mathbf{d}_{j}\right\rangle_{\mathbf{A}}=\mathbf{d}_{i}^{T} \mathbf{A} \mathbf{d}_{j}=\mathbf{0}
$$

To impose A-orthogonality on the search directions we use the conjugate Gram-Schmidt process

$$
\begin{equation*}
\mathbf{d}_{i}=\mathbf{u}_{i}-\sum_{j<i} \frac{\left\langle\mathbf{d}_{j}, \mathbf{u}_{i}\right\rangle_{\mathbf{A}}}{\left\langle\mathbf{d}_{j}, \mathbf{d}_{j}\right\rangle_{\mathbf{A}}} \mathbf{d}_{j}=\mathbf{u}_{i}+\sum_{j<i} \beta_{i j} \mathbf{d}_{j} \tag{2.11}
\end{equation*}
$$

where $\left\{\mathbf{u}_{i}\right\}_{i=0}^{m-1}$ is a set of linearly independent vectors. However, since the conjugate Gram-Schmidt process requires knowledge about all the earlier values of $\mathbf{d}_{j}$, those have to be saved in memory. Luckily this can be avoided by choosing $\mathbf{u}_{i}=\mathbf{r}_{i}$ where $\mathbf{r}_{i}$ are the residuals given by

$$
\mathbf{r}_{i}=\mathbf{b}-\mathbf{A} \mathbf{x}_{i}
$$

With this choice it is then possible to show that $\beta_{i j}$ simplifies to

$$
\beta_{i j}= \begin{cases}\frac{1}{\alpha_{i-1}} \cdot \frac{\mathbf{r}_{i}^{T} \mathbf{r}_{i}}{\left\langle\mathbf{d}_{i-1}, \mathbf{d}_{i-1}\right\rangle_{\mathbf{A}}}, & i=j+1 \\ 0, & \text { otherwise }\end{cases}
$$

Hence we have that

$$
\begin{align*}
\mathbf{d}_{i+1} & =\mathbf{r}_{i}+\beta_{i+1, i} \mathbf{d}_{i} \\
& =\mathbf{r}_{i}+\frac{1}{\alpha_{i}} \cdot \frac{\mathbf{r}_{i+1}^{T} \mathbf{r}_{i+1}}{\left\langle\mathbf{d}_{i}, \mathbf{d}_{i}\right\rangle_{\mathbf{A}}} \mathbf{d}_{i} \tag{2.12}
\end{align*}
$$

To find the step sizes, we use that the errors $\mathbf{e}_{i}=\mathbf{x}_{i}-\mathbf{x}$ are $A$-orthogonal to the previous search direction

$$
\left\langle\mathbf{d}_{i}, \mathbf{e}_{i+1}\right\rangle_{\mathbf{A}}=0
$$

the decomposition of the error as $\mathbf{e}_{i+1}=\mathbf{e}_{i}+\alpha_{i} \mathbf{d}_{i}$ and $\mathbf{r}_{i}=\mathbf{b}-\mathbf{A} \mathbf{x}_{i}=-\mathbf{A}\left(\mathbf{x}_{i}-\mathbf{x}\right)=-\mathbf{A} \mathbf{e}_{i}$ to find

$$
\alpha_{i}=\frac{\mathbf{d}_{i}^{T} \mathbf{r}_{i}}{\left\langle\mathbf{d}_{i}, \mathbf{d}_{i}\right\rangle_{\mathbf{A}}} .
$$

We now want to simplify $\alpha_{i}$ further by showing that $\mathbf{d}_{i}^{T} \mathbf{r}_{i}=\mathbf{r}_{i}^{T} \mathbf{r}_{i}$. Since the conjugate gradient method converges, which is true when the assumption of a symmetric and positive definite $\mathbf{A}$ holds, we have that $\mathbf{e}_{m}=0$ and thus by the decomposition of the error terms

$$
\begin{aligned}
0 & =\mathbf{e}_{m} \\
& =\mathbf{e}_{m-1}-\alpha_{m-1} \mathbf{d}_{m-1} \\
& =\ldots \\
& =\mathbf{e}_{j}-\sum_{k=j}^{m-1} \alpha_{k} \mathbf{d}_{k}
\end{aligned}
$$

or equivalently, $\mathbf{e}_{j}=\sum_{k=j} \alpha_{k} \mathbf{d}_{k}$. If we now premultiply this expressing by $\mathbf{d}_{i}^{T} \mathbf{A}, i<j$, we have by conjugacy that

$$
\mathbf{d}_{i}^{T} \mathbf{A} \mathbf{e}_{j}=\sum_{k=j}^{m-1} \alpha_{k} \mathbf{d}_{i}^{T} \mathbf{A} \mathbf{d}_{k}=0
$$

By again using that $\mathbf{r}_{i}=-\mathrm{Ae}_{i}$ we get the intermediate result that

$$
\mathbf{d}_{i}^{T} \mathbf{r}_{j}=0, \quad i<j
$$

which states that the residuals are orthogonal to all previous search directions. Now, by using the conjugate Gram-Schmidt process expression in equation (2.11) with the choice $\mathbf{u}_{i}=\mathbf{r}_{j}$, pre-multipling by $\mathbf{r}_{i}$ and applying the intermediate result gives us

$$
\begin{aligned}
\mathbf{d}_{i}^{T} \mathbf{r}_{i} & =\mathbf{r}_{i}^{T} \mathbf{r}_{i}+\sum_{j<i} \beta_{i j} \mathbf{d}_{j} \mathbf{r}_{i} \\
& =\mathbf{r}_{i}^{T} \mathbf{r}_{i}
\end{aligned}
$$

and we can now write the step length as

$$
\alpha_{i}=\frac{\mathbf{r}_{i}^{T} \mathbf{r}_{i}}{\left\langle\mathbf{d}_{i}, \mathbf{d}_{i}\right\rangle_{\mathbf{A}}} .
$$

We can now use this step length in the expression of the search direction in equation (2.12) to get

$$
\mathbf{d}_{i+1}=\mathbf{r}_{i+1}+\frac{\mathbf{r}_{i+1}^{T} \mathbf{r}_{i+1}}{\mathbf{r}_{i}^{T} \mathbf{r}_{i}} \mathbf{d}_{i} .
$$

The only unknown variable now is the consecutive residual $\mathbf{r}_{i+1}$, which we can calculate as

$$
\begin{aligned}
\mathbf{r}_{i+1} & =-\mathbf{A} \mathbf{e}_{i+1} \\
& =-\mathbf{A}\left(\mathbf{e}_{i}+\alpha \mathbf{d}_{i}\right) \\
& =\mathbf{r}_{i}-\alpha_{i} \mathbf{A d}_{i} .
\end{aligned}
$$

Summarizing, the conjugate gradient algorithm can be described as

```
Algorithm 1 Conjugate Gradient Algorithm
    function ConjugateGradient( \(\mathbf{b}, \mathbf{A}, \mathbf{x}_{0}\) )
        \(\mathbf{d}_{0}, \mathbf{r}_{0} \leftarrow \mathbf{b}-\mathbf{A x}_{0}\)
        \(i=0\)
        while \(\mathbf{r}_{i} \neq 0\) do
            \(\alpha_{i} \leftarrow \frac{\mathbf{r}_{i}^{T} \mathbf{r}_{i}}{\left\langle\mathbf{d}_{i}, \mathbf{d}_{i}\right\rangle_{\mathbf{A}}}\)
            \(\mathbf{x}_{i+1} \leftarrow \mathbf{x}_{i}+\alpha_{i} \mathbf{d}_{i}\)
            \(\mathbf{r}_{i+1} \leftarrow \mathbf{r}_{i}-\alpha_{i} \mathbf{A d}_{i}\)
            \(\mathbf{d}_{i+1} \leftarrow \mathbf{r}_{i+1}+\frac{\mathbf{r}_{i+1}^{T} \mathbf{r}_{i+1}}{\mathbf{r}_{i}^{T} \mathbf{r}_{i}} \mathbf{d}_{i}\)
            \(i \leftarrow i+1\)
        return ( \(\mathrm{x}_{i}\) )
```


## Steinhaug's conjugate gradient method

As earlier mentioned, the conjugate gradient algorithm doesn't take into account that we are limited to our circle given by $\|\mathbf{x}\|_{2} \leq \Delta$ in our search. It also assumes that the matrix corresponding to the second order terms is positive-definite which also might not always hold for the second order models considered. Therefore a slightly modified algorithm called Steinhaug's conjugate gradient method [24] is presented, which solves these two problems. The algorithm is described by

```
Algorithm 2 Steinhaug's Conjugate gradient Algorithm
    function COnJugateGradient \(\left(\mathbf{b}, \mathbf{A}, \mathbf{x}_{0}, \Delta, \epsilon\right)\)
        \(\mathbf{d}_{0}, \mathbf{r}_{0} \leftarrow \mathbf{b}-\mathbf{A x}_{0}\)
        for \(i=0: d\) do
            if \(\left|\mid \mathbf{r}_{i} \| \leq \epsilon\right.\) then
                return ( \(\mathrm{x}_{i}\) )
            if \(\mathbf{d}_{i}^{T} \mathbf{A d}_{i} \leq 0\) then
            Find \(\alpha_{i}\) s.t. \(\left\|\mathbf{x}_{i}+\alpha_{i} \mathbf{d}_{i}\right\|=\Delta\)
            return \(\left(\mathbf{x}_{\mathbf{i}}+\alpha_{i} \mathbf{d}_{i}\right)\)
            \(\alpha_{i} \leftarrow \frac{\mathbf{r}_{i}^{T} \mathbf{r}_{i}}{\left\langle\mathbf{d}_{i}, \mathbf{d}_{i}\right\rangle_{\mathbf{A}}}\)
            if \(\left\|\mathbf{x}_{i}+\alpha_{i} \mathbf{d}_{i}\right\| \geq \Delta\) then
                Find \(\alpha_{i}\) s.t. \(\left\|\mathbf{x}_{i}+\alpha_{i} \mathbf{d}_{i}\right\|=\Delta\)
                    return \(\left(\mathbf{x}_{\mathbf{i}}+\alpha_{i} \mathbf{d}_{i}\right)\)
            else
                    \(\mathbf{x}_{i+1} \leftarrow \mathbf{x}_{i}+\alpha_{i} \mathbf{d}_{i}\)
                    \(\mathbf{r}_{i+1} \leftarrow \mathbf{r}_{i}-\alpha_{i} \mathbf{A d}_{i}\)
            \(\mathbf{d}_{i+1} \leftarrow \mathbf{r}_{i+1}+\frac{\mathbf{r}_{i+1}^{T} \mathbf{r}_{i+1}}{\mathbf{r}_{i}^{T} \mathbf{r}_{i}} \mathbf{d}_{i}\)
        return \(\left(\mathrm{x}_{i}\right)\)
```


## Chapter 3

## Method and Implementation

### 3.1 Active Subspace \& Stochastic Subspace Descent

Using the active subspace along with the stochastic subspace descent can be done in two ways. The first one is inspired by Metropolis-Hastings algorithm in statistical physics, where we use a rejection-acceptance sampling method to find the stochastic subspace $\mathbf{P}_{k}$. This is done by calculating a quantity which we call agreement which will serve as the probability of accepting a proposed random subspace $\mathbf{Q}$. This agreement-factor will be higher when the suggested subspace aligns more with the eigenvectors in $\hat{\mathbf{W}}$ corresponding to larger eigenvalues. This method will be referred to as the agreement method. The second way is by using $\hat{\mathbf{C}}$ to directly weigh the calculation of the projection matrix $\mathbf{P}$. This method will be referred to as the weighting method.

## Agreement method:

Let the agreement be defined by

$$
\begin{equation*}
\text { agreement }=\left\|\hat{\Lambda}_{\text {norm }}^{\frac{1}{2}} \hat{\mathbf{W}}^{T} \mathbf{Q}\right\|_{F r}^{2} \tag{3.1}
\end{equation*}
$$

where $\mathbf{Q}$ is a randomly generated unscaled Haar distributed matrix and $\hat{\boldsymbol{\Lambda}}_{\text {norm }}$ is the Monte-Carlo estimated eigenvalue matrix which has been normalized such that the eigenvalues sum to 1, i.e. $\hat{\Lambda}_{\text {norm }}=\hat{\Lambda} /\|\hat{\Lambda}\|_{1}$. Also, the norm $\|\cdot\|_{F r}$ is the so-called Frobenius norm. Due to the power, the agreement is bounded below by 0 , however we also want to show that it is bounded above by 1 in order to use it as a probability. To show this, we start by noting that the expression of the agreement can be re-written as agreement $=\sum_{i, j}\left[\boldsymbol{\Lambda}\left(\hat{\mathbf{W}}^{T} \mathbf{Q}\right)^{\mathrm{O}^{2}}\right]_{i, j}$, where the " $\bigcirc 2$ " denotes the Hadamarad power [25] which indicates the element-wise power of a matrix. Furthermore, we have that $\hat{\mathbf{W}}$ is an orthogonal matrix in $\mathbb{R}^{m \times m}$ and $\mathbf{Q}$ is a matrix in $\mathbb{R}^{m \times l}$ where the columns are orthonormal. We can thus define the matrix $\mathbf{A}=\hat{\mathbf{W}}^{T} \mathbf{Q} \in \mathbb{R}^{m \times l}$, which then have
orthonormal columns. Substitution of $\mathbf{A}$ in the rewritten expression of the agreement then gives

$$
\begin{equation*}
\text { agreement }=\sum_{i, j}\left[\hat{\boldsymbol{\Lambda}}(\mathbf{A})^{\circ 2}\right]_{i, j}=\sum_{i} \lambda_{i} \sum_{j}[\mathbf{A}]_{i, j}^{2} . \tag{3.2}
\end{equation*}
$$

To bound the agreement, we extend the matrix $\mathbf{A}$ by adding $m-l$ orthonormal columns such that the matrix $\mathbf{A}_{\text {ext }} \in \mathbb{R}^{m \times m}$ becomes an orthognal matrix. We then have that

$$
\begin{equation*}
\text { agreement }=\sum_{i} \lambda_{i} \sum_{j}[\mathbf{A}]_{i, j}^{2} \leq \sum_{i} \lambda_{i} \sum_{j}\left[\mathbf{A}_{\text {ext }}\right]_{i, j}^{2} \tag{3.3}
\end{equation*}
$$

Now, since the matrix $\mathbf{A}_{\text {ext }}$ is orthonogal the euclidian norm of each row (and also each column) is one. Hence, we have that $\sum_{j}\left[\mathbf{A}_{e x t}\right]_{i, j}^{2}=1, \forall i$. Since the the eigenvalues sum to one due to the normalization, we can hence express the upper bound of the agreement as

$$
\begin{equation*}
\text { agreement } \leq \sum_{i} \lambda_{i} \sum_{j}\left[\mathbf{A}_{e x t}\right]_{i, j}^{2}=\sum_{i} \lambda_{i}=1 . \tag{3.4}
\end{equation*}
$$

## Weighting method:

To generate a weighted unscaled Haar distributed matrix $\mathbf{Q}^{(W)}$, we simply premultiply Q by $\hat{C}$ as

$$
\begin{equation*}
\mathbf{Q}^{(W)}=\hat{\mathbf{C}} \mathbf{Q} \tag{3.5}
\end{equation*}
$$

where the estimation of $\mathbf{C}$ once again is calculated and updated every $q$ steps. The motivation behind this idea is to consider the decomposition $\hat{\mathbf{C}}=\hat{\mathbf{W}} \hat{\mathbf{\Lambda}} \hat{\mathbf{W}}^{T}$ applied to the unscaled Haar distributed matrix $\mathbf{Q}$. We then see that $\mathbf{D}=\hat{\mathbf{W}}^{T} \mathbf{Q}$ rotates $\mathbf{Q}$, where the resulting matrix is scaled by $\hat{\boldsymbol{\Lambda}}$ such that the directions with larger eigenvalues are magnified. Finally, we rotate back to the original coordinate system via the matrix $\hat{\mathbf{W}}$ which is the opposite rotation of $\hat{\mathbf{W}}^{T}$ due to orthogonality.

## Algorithm

The pseudocode for the algorithms based on stochastic subspace descent, where the input variable SSDmethod decides whether we are using the agreement or weighting method, can now be described by

```
Algorithm 3 SSD Active Subspace Algorithm
    function SCD_ActiveSubspace \((\gamma, \beta, \sigma\), max Evals,maxIncreases \(, \epsilon, q, m, l, \ldots\)
                                    \(\mathbf{x}_{0}, f, S S D\) method)
        evals \(=0\), increaseIterations \(=0, k=1, \mathbf{x}=\mathbf{x}_{0}\)
        \(\hat{\mathbf{C}}=\mathbf{0}, \hat{\boldsymbol{\Lambda}}=\mathbf{1} / m, \hat{\mathbf{W}}=\operatorname{diag}(\mathbf{1})\)
        while evals < maxEvals AND increaseIterations < maxIncreases do
            if \(\operatorname{modulus}(k, q)==0\) then
                evals, \(\nabla f \leftarrow\) calculateGradient \((\mathbf{x}, f)\)
                \(\hat{\mathbf{C}}, \hat{\boldsymbol{\Lambda}}, \hat{\mathbf{W}} \leftarrow\) updateActiveSubspaceDecomposition \((\hat{\mathbf{C}}, \nabla f)\)
                \(g(\mathbf{x}) \leftarrow \nabla f\)
            else
                if \(S S\) Dmethod \(==^{\prime}\) agreement then
                    \(\hat{\boldsymbol{\Lambda}}_{\text {norm }} \leftarrow \operatorname{normalize}(\hat{\boldsymbol{\Lambda}})\)
                    agreement \(\leftarrow 0\)
                    while agreement \(<X\), where \(X \sim U(0,1)\) do
                        \(\mathbf{Q} \leftarrow\) generateUnscaledHaarDistributedMatrix \((m, l)\)
                    agreement \(\leftarrow\) calculateAgreement \(\left(\mathbf{Q}, \hat{\boldsymbol{\Lambda}}_{\text {norm }}, \hat{\mathbf{W}}\right)\)
                    \(\mathbf{M} \leftarrow \operatorname{scaleMatrix}(\mathbf{Q})\)
                if SSDmethod \(==^{\prime}\) weighted \({ }^{\prime}\) then
                        \(\mathbf{Q} \leftarrow\) generateUnscaledHaarDistributedMatrix \((d, l)\)
                \(\mathbf{Q}^{(W)} \leftarrow \hat{\mathbf{C}} \mathbf{Q}\)
                \(\mathbf{M} \leftarrow \operatorname{scaleMatrix}\left(\mathbf{Q}^{(W)}\right)\)
                evals, \(g(\mathbf{x}) \leftarrow \mathbf{M M}^{T} \nabla f(\mathbf{x})\)
            \(\alpha\), evals \(\leftarrow \operatorname{armijoLineSearch}(\gamma, \beta, \sigma, g(\mathbf{x}), \mathbf{x}, f)\)
            \(\mathbf{x}_{\text {new }} \leftarrow \mathbf{x}-\alpha g(\mathbf{x})\)
            if \(f\left(\mathbf{x}_{\text {new }}\right)<f(\mathbf{x})-\epsilon\) then
                \(\mathbf{x} \leftarrow \mathbf{x}_{\text {new }}\)
                increaseIterations \(\leftarrow 0\)
            else
                        increaseIterations \(\leftarrow\) increaseIterations +1
            \(k \leftarrow k+1\)
        return ( x )
```


### 3.2 Active Subspace \& Stochastic Coordinate Descent

Here we are combining the active subspace methodology with stochastic coordinate descent. This is done by every $m$ :th iteration calculate the full gradient in the SCD scheme, and then use this information to approximate the matrix $\mathbf{C}$ with the Monte Carlo approach described in equation (2.4) to obtain $\hat{\mathbf{C}}$. The matrix $\hat{\mathbf{C}}$ is then eigendecomposed to produce the eigen-directions $\hat{\mathbf{W}}$ and the corresponding eigenvalues in the
diagonal of $\hat{\boldsymbol{\Lambda}}$. The eigenvalues are then normalized such that they sum to one. These normalized eigenvalues are then used in consecutive iterations as weights to decide which of the eigendirections in $\hat{\mathbf{W}}$ we will use as descent directions. This is appropriate since we want to focus more on the directions of which the underlying function seems to fluctuate when choosing descent direction. This algorithm can thus be described by

```
Algorithm 4 SCD Active Subspace Algorithm
    function SCD_ActiveSubspace \(\left(\gamma, \beta, \sigma\right.\), maxEvals,maxIncreases, \(\left.\epsilon, q, m, \mathbf{x}_{0}, f\right)\)
        evals \(=0\), increaseIterations \(=0, k=1, \mathbf{x}=\mathbf{x}_{0}\)
        \(\hat{\mathbf{C}}=\mathbf{0}, \hat{\mathbf{\Lambda}}=\mathbf{1} / m, \hat{\mathbf{W}}=\operatorname{diag}(\mathbf{1})\)
        while evals < maxEvals AND increaseIterations < maxIncreases do
            if \(\operatorname{modulus}(k, q)==0\) then
                evals, \(\nabla f \leftarrow\) calculateGradient \((\mathbf{x}, f)\)
                \(\hat{\mathbf{C}}, \hat{\boldsymbol{\Lambda}}, \hat{\mathbf{W}} \leftarrow\) updateActiveSubspaceDecomposition \((\hat{\mathbf{C}}, \nabla f)\)
                \(g(\mathbf{x}) \leftarrow \nabla f\)
            else
                \(\hat{\boldsymbol{\Lambda}}_{\text {norm }} \leftarrow \operatorname{normalize}(\hat{\boldsymbol{\Lambda}})\)
                \(i_{k} \leftarrow\) selectRandomActiveSubspaceCoordinate \((\hat{\boldsymbol{\Lambda}}, \hat{\mathbf{W}})\)
                \([\nabla f]_{i_{k}}\), evals \(\leftarrow\) calculateGradientComponent \(\left(\mathbf{x}, i_{k}, f\right)\)
                \(g(\mathbf{x}) \leftarrow[\nabla f]_{i_{k}}\)
            \(\alpha\), evals \(\leftarrow \operatorname{armijoLineSearch}(\gamma, \beta, \sigma, g(\mathbf{x}), \mathbf{x}, f)\)
            \(\mathbf{x}_{\text {new }} \leftarrow \mathbf{x}-\alpha g(\mathbf{x})\)
            if \(f\left(\mathbf{x}_{\text {new }}\right)<f(\mathbf{x})-\epsilon\) then
                \(\mathbf{x} \leftarrow \mathbf{x}_{\text {new }}\)
                increaseIterations \(\leftarrow 0\)
            else
                increaseIterations \(\leftarrow\) increaseIterations +1
            \(k \leftarrow k+1\)
        return ( x )
```


### 3.3 Response Surface Method

Here the approach for implementation of the Response Surface Method will be described. Due to its similarities with the trust region method from optimization, this method will be refereed to as the trusty $R S M$.

## Experimental design

Both the central composite design (CCD) and the latin hypercube design (LHD) will be used, where the LHD is used with the maxmin performance measure.

## Model

The model used is the second order regression model.

## Trust region subproblem

To solve the trust region subproblem, we will first try with the canonical analysis to find an exact solution. However, this only works when the matrix $\mathbf{B}$ containing the second order terms is positive-definite (implying that the stationary point is a minimum point) and the calculated stationary point is inside our region of trust. If this direct approach fails, we will instead solve the subproblem approximately by using Steinhaug's method.

## Updating region of trust

The choice of the size of the region where we sample our points and fit the model is an essential component of the response surface method. For example, if the region is too small convergence will be slow since the next iteration point is bound to be inside the current region. However, if the region is too large the fitted model might be a poor approximation of the true behaviour and will therefore be unable to find a good minimum point to the trust region subproblem. To answer this dilemma, an adaptive approach is used where the region of trust is enlarged or shrunken based on the fitting performance of the second order model. Let $\mathbf{x}_{k}$ be the current point and let $\mathbf{x}^{(s)}$ an trial point given as the solution of the trust regions subproblem. Then define the performance measure $\rho$ as

$$
\begin{equation*}
\rho=\frac{f\left(\mathbf{x}_{k}\right)-f\left(\mathbf{x}^{(s)}\right)}{\hat{y}\left(\mathbf{x}_{k}\right)-\hat{y}\left(\mathbf{x}^{(s)}\right)} \tag{3.6}
\end{equation*}
$$

where $f$ is the true function calls and $\hat{y}$ is the response from the fitted second order model. Now choose parameters $0<c_{1}<c_{2}<1,0<t_{1}<1<t_{2}$ and $\eta \in(0,1 / 4]$, then updation of the trust region size will be

$$
\Delta_{k+1}= \begin{cases}t_{1} \Delta_{k}, & \text { if } \rho<c_{1}  \tag{3.7}\\ t_{2} \Delta_{k}, & \text { if } \rho>c_{2} \\ \Delta_{k}, & \text { if } c_{1} \leq \rho \leq c_{2}\end{cases}
$$

Further, the centre of the region of trust will be updated as

$$
\mathbf{x}_{k+1}= \begin{cases}\mathbf{x}^{(s)}, & \text { if } \rho>\eta  \tag{3.8}\\ \mathbf{x}_{k}, & \text { otherwise }\end{cases}
$$

## Algorithm

The complete algorithm can now be described as

```
Algorithm 5 Trusty RSM
    function Trusty_RSM \(\left(f, d, \mathbf{x}_{0}, \Delta_{0}, t_{1}, t_{2}, c_{1}, c_{2}, \eta\right.\), max Evals \(\left., b_{t o l}, \Delta_{\text {min }}\right)\)
        \(\mathrm{x} \leftarrow \mathrm{x}_{0}\)
        \(\Delta \leftarrow \Delta_{0}\)
        evals \(\leftarrow 0, k \leftarrow 1\)
        while evals \(<\) maxEvals AND \(\|\mathbf{b}\|>b_{\text {tol }}\) AND \(\Delta>\Delta_{\min }\) do
            coded \(\leftarrow\) generateDesignPoints \((d)\)
            uncoded \(\leftarrow\) transformToUncoded \((\operatorname{coded}, \Delta, \mathbf{x})\)
            evals, \(\mathbf{y} \leftarrow\) evaluateCoordinates \((f\), uncoded \()\)
            \(\hat{\beta}_{0}, \mathbf{b}, \mathbf{B} \leftarrow\) fitRegressionModel (coded, \(\mathbf{y}\) )
            \(\mathbf{x}^{(s)} \leftarrow\) trustRegionSubproblemOptimization \(\left(\hat{\beta}_{0}, \mathbf{b}, \mathbf{B}, \Delta\right)\)
            evals, \(\rho \leftarrow\) calculatePerformanceMeasure \(\left(f, \mathbf{x}, \mathbf{x}^{(s)}, \hat{\beta}_{0}, \mathbf{b}, \mathbf{B}\right)\)
            \(\mathbf{x} \leftarrow \operatorname{updatePosition}\left(\mathbf{x}, \mathbf{x}^{(s)}, \rho, \eta, \Delta\right)\)
            \(\Delta \leftarrow\) updateTrustRegionSize \(\left(t_{1}, t_{2}, c_{1}, c_{2}, \rho, \Delta\right)\)
        return ( x )
```


### 3.4 Simulation Study

To test the different algorithms and investigate the strengths and weaknesses of respective algorithm a simulation study is performed. To do this we are considering four test functions $f$, where we to each function add a Gaussian white noise $\epsilon \sim N\left(0, \sigma^{2}\right)$, where $\sigma^{2}$ is the variance. Here we are considering four levels of noise:

1. No noise: $\sigma=0$,
2. Low noise: $\sigma=0.1$,
3. Medium noise: $\sigma=1$, and
4. High noise: $\sigma=10$.

To measure the efficiency, the mean and medium number of function evaluation will be recorded. To measure the performance we will consider the mean, the median and the mean square error (MSE). Due to the noise we can obtain function calls that have a lower value than the lowest possible value of the function $f$ alone, which might distort the performance measures. To avoid these unrealistic low values, we are first optimizing on the noisy function to get the optimal location $\mathbf{x}^{*}$ as

$$
\begin{equation*}
\hat{\mathbf{x}}=\underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x})+\epsilon \tag{3.9}
\end{equation*}
$$

which then is evaluated without adding any noise to get our estimated optimum as

$$
\begin{equation*}
\hat{y}=f\left(\mathbf{x}^{*}\right) \tag{3.10}
\end{equation*}
$$

which is then used in the performance measure calculations.
To obtain the simulation data we are running the optimization on each method, function and noise combination 1000 times. To initialize we are considering the initial points to be uniformly distributed inside the domain specified by the lower domain vector $\mathbf{x}_{l b}$ and the upper domain vector $\mathbf{x}_{u b}$.

The methods considered for this simulation study are:

- Trusty RSM with latin hypercube design,
- Trusty RSM with central composite design,
- SSD agreement method,
- SSD weighted method,
- SCD active subspace method,
- $\operatorname{SSD}$ (as a reference method),
- SCD (as a reference method), and
- Gradient descent (as a reference method).


### 3.4.1 Test Functions

The four test functions are the Wood test function, the uncoupled Rosenbrock test function (in 10 dim ), the Powell test function, and a ridge function (in 6 dim). All these test functions are unconstrained optimization functions with a unique global optimum, and therefore suitable test functions for our methods. The first three test functions are non-convex, and therefore harder problems, which will help us evaluate how well our algorithms can tackle these kind of functions. The fourth test function is convex, but chosen to be a ridge function since we know that the active subspace methodology should be beneficial in this case (see section 2.5.6), which we want to show empirically.

## Wood test function

The first test function is the Wood test function given by

$$
\begin{align*}
f\left(x_{1}, x_{2}, x_{3}, x_{4}\right)= & 100\left(x_{1}^{2}-x_{2}\right)^{2}+\left(x_{1}-1\right)^{2}+\left(x_{3}-1\right)^{2}+90\left(x_{3}^{2}-x_{4}\right)^{2}+  \tag{3.11}\\
& 10.1\left(\left(x_{2}-1\right)^{2}+\left(x_{4}-1\right)^{2}\right)+19.8\left(x_{2}-1\right)\left(x_{4}-1\right) .
\end{align*}
$$

Its global minimum is given by $f(\mathbf{1})=0$, and here we have chosen the initial domain vectors to be $\mathbf{x}_{l b}=-\mathbf{1}$ and $\mathbf{x}_{u b}=\mathbf{3}$. By fixing either the first or last two coordinates, this function can be visualized, which can be seen in Figure 3.1 and 3.2.


Figure 3.1: Response surface and contour plot of Wood test function by fixing $x_{3}=x_{4}=$ 1.


Figure 3.2: Response surface and contour plot of Wood test function by fixing $x_{1}=x_{2}=$ 1.

## Uncoupled Rosenbrock test function (10 dim)

The second test function is the uncoupled Rosenbrock test function in $m$ dimensions ( $m$ even)

$$
\begin{equation*}
f(\mathbf{x})=\sum_{i=1}^{m / 2}\left[100\left(x_{2 i-1}^{2}-x_{2 i}\right)^{2}+\left(x_{2 i-1}-1\right)^{2}\right] \tag{3.12}
\end{equation*}
$$

where we have chosen $m=10$. Its global minimum is given by $f(\mathbf{1})=0$, and here we have chosen the initial domain vectors to be $\mathbf{x}_{l b}=\mathbf{0}$ and $\mathbf{x}_{u b}=\mathbf{2}$. Letting $m=2$, the uncoupled Rosenbrock test function is plotted, which can be seen in Figure 3.3.


Figure 3.3: Response surface and contour plot of the uncoupled Rosenbrock test function in two dimensions.

## Powell test function

The third test function is the Powell test function given by

$$
\begin{equation*}
f\left(x_{1}, x_{2}, x_{3}, x_{4}\right)=\left(x_{1}+10 x_{2}\right)^{2}+5\left(x_{3}-x_{4}\right)^{2}+\left(x_{2}-2 x_{3}\right)^{4}+10\left(x_{1}-x_{4}\right)^{4} . \tag{3.13}
\end{equation*}
$$

Its global minimum is given by $f(\mathbf{0})=0$, and here we have chosen the initial domain vectors to be $\mathbf{x}_{l b}=-\mathbf{5}$ and $\mathbf{x}_{u b}=\mathbf{5}$. By fixing either the first or last two coordinates, this function can be visualized, which can be seen in Figure 3.4 and 3.5.


Figure 3.4: Response surface and contour plot of Powell test function by fixing $x_{3}=$ $x_{4}=0$.


Figure 3.5: Response surface and contour plot of Powell test function by fixing $x_{1}=$ $x_{2}=0$.

## Ridge function (6 dim)

The fourth test function is the $m$-dimensional ridge function given by

$$
\begin{equation*}
f(\mathbf{x})=\left(\sum_{i=1}^{m} i x_{i}\right)^{2} \tag{3.14}
\end{equation*}
$$

where we have chosen $m=6$. This function has an infinite number of optimas $\mathbf{x}^{*}$ fulfilling $\left(\sum_{i=1}^{m} i x_{i}\right)=0$, where all the optimas fulfill $f\left(\mathbf{x}^{*}\right)=0$. Here the initial domain vectors are chosen to be $\mathbf{x}_{l b}=-\mathbf{1 0}$ and $\mathbf{x}_{u b}=\mathbf{1 0}$. Letting $m=2$, the ridge test function is plotted, which can be seen in Figure 3.6.


Figure 3.6: Response surface and contour plot of the ridge test function in two dimensions.

### 3.5 Application to Pre-Hospital Care

For applying the optimization models to a real problem we are looking at the problem of minimizing the ambulance driving time by moving the 4 ambulances located in the municipality of Umeå. To obtain the ambulance alarm data a computer simulation is used. This simulation uses historical call data and statistical models which are described in [26], and is developed by Jonas Westin. Due to randomness in the models of this simulation, the result is stochastic in the sense that identical input parameters might yield different results. The ambulance alarm model works by simulating approximately

100 days of call data for ambulances within the municipality of Umea. The calls during this period then contain the following information:

- position of the alarm,
- DeSO zone of the alarm,
- starting time (how long until the ambulance leaves the station after receiving the call),
- driving time,
- priority class.

The DeSO zones are demografic statistics areas (sv: demografiska statistikområden) which are areas that by construction have between 700 and 2700 inhabitants while also trying to align its borders with for example roads, rivers and railways [27]. The different DeSO zones of the municipality of Umeå can be seen in Figure 3.7. Note that smaller DeSO zones indicates a more dense population.


Figure 3.7: Visualization of the different DeSO zones in the municipality of Umeå.

The different priorities are priority 1 , priority 2 and priority 3 . Here priority 1 alarm means that the patient is having life-threatening symptoms, for example a stroke, and the arrival time of the ambulance might affect the chance of survival of the patient. Priority 2 alarms is when the patient is not having life-threatening symptoms, but might be seriously ill or severely hurt, such that the situation require urgent measures. A typical priority 2 alarm could for example be a fracture. Priority 3 alarm is when the response time of the ambulance doesn't matter since the patient is in a stable condition. This
third priority class also includes scheduled assignments, where the ambulance acts in a taxi-like fashion.

Furthermore, we also have that both the ambulance positions and location of the alarms are limited by a zone system called SCB's statistic zones. Here each zone is 250 x 250 meters in densely populated areas, or $1000 \times 1000$ meters otherwise. Due to this zone partitioning, moving the position of an ambulance within one of these statistic zones will not affect its response time. Similarly, different alarm calls at different positions within the zones will experience the same expected waiting time. Because of this 'resolution' of the data, the gradient based methods are not appropriate and will hence not be used. Also, since we don't know how the response surface will behave we want a design with good space filling properties, which is why the trusty RSM method with LHD design will be used for the optimization of this problem.

The current positioning of all the 4 ambulances is at the ambulance station in Alidhem. This positioning is shown in Figure 3.8. Even though all ambulances are located at the same place, this positioning is fairly good due to its proximity of the central areas and closeness to larger roads such as the E4 road. This current ambulance placement will be used both as initial positioning for our optimization and also as a reference positioning which we will compare our results against.


Figure 3.8: Current position of all 4 ambulances in the municipality of Umeå.

### 3.5.1 Performance Measures

To measure the performance of the driving time of the ambulances to the alarm position, we will use the median time as performance measure since it is a commonly used metric in pre-hospital care. However, since more than half of the calls usually came from
the central parts of the municipality, the median time performance measure will be independent of calls originating far away from the centre which might cause the driving times to the rural areas to be very long. To combat this neglect of the rural areas we will also look at two more performance measures which are the 90 :th percentile response time and the time of the longest DeSO zone median time. To summarize, the following performance measures will be used:

1. Median times
2. $90:$ th percentile time
3. Maximum of the DeSO median response time of each DeSO zone

## Chapter 4

## Result and Discussion

### 4.1 Simulation Study

## Wood test function

The results for the Wood test function can be seen in Tables 4.1 to 4.4. From the first table with no noise we can see that the trusty RSM methods perform better, both in terms of the performance metrics and the number of function evaluations needed until convergence. Looking at all the descent based methods, we see that they all have the same median value. This is due to the fact that they share the same termination test and that more than half of the runs converge to the same point with response value 0.00693 .

Looking at the noisy tables we can now see that the descent methods require a lower number of function evaluations compared to the trusty RSM methods. This difference increase with the noise. From these tables we can also see that the trusty RSM methods handle noise better since their performance measures increase relatively less compared to the descent methods when we are increasing the levels of the noise.

Moreover, from the tables (especially when $\sigma=0$ ) we can see that there is a big difference between the mean and the median values. This discrepancy is because we have that most values are quite close to the optimum, which makes the median low, but there are a few outliers with high values which are driving up the mean. Due to these high-valued outliers we also get the high MSE.

Table 4.1: No noise $(\sigma=0)$ result for the Wood test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 0.127 | 0.000362 | 0.991 | 1336.08 | 1219 |
| TrustyRSM (CCD) | 0.0559 | 0.00043 | 0.434 | 1793.15 | 1561 |
| SSD agreement $(l=2)$ | 0.313 | 0.00693 | 2.41 | 10861.7 | 11318.5 |
| SSD weighted $(l=2)$ | 0.416 | 0.00693 | 3.22 | 22499.4 | 22351 |
| SSD $(l=2)$ | 0.369 | 0.00693 | 2.85 | 10603.3 | 10907 |
| SCD active subspace | 0.848 | 0.00693 | 6.63 | 53439.9 | 55824.5 |
| SCD | 0.132 | 0.00693 | 0.991 | 17557.2 | 19448.5 |
| GD | 0.235 | 0.00693 | 1.8 | 18748.7 | 17722 |

Table 4.2: Low noise ( $\sigma=0.1$ ) result for the Wood test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 1.83 | 0.775 | 8.17 | 668.275 | 652 |
| TrustyRSM (CCD) | 1 | 0.273 | 3.51 | 1044.02 | 1041 |
| SSD agreement $(l=2)$ | 30.5 | 4.43 | $8.06 \mathrm{e}+04$ | 766.208 | 694 |
| SSD weighted $(l=2)$ | 27.2 | 5.94 | $1.76 \mathrm{e}+04$ | 814.557 | 735.5 |
| SSD $(l=2)$ | 17.7 | 4.73 | $1.12 \mathrm{e}+04$ | 758.237 | 678 |
| SCD active subspace | 360 | 67.8 | $9.68 \mathrm{e}+05$ | 398.981 | 320 |
| SCD | 63.7 | 7.8 | $6.9 \mathrm{e}+04$ | 703.683 | 507 |
| GD | 276 | 25.3 | $7.79 \mathrm{e}+05$ | 403.464 | 350 |

Table 4.3: Medium noise $(\sigma=1)$ result for the Wood test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 3.22 | 2.48 | 18.3 | 594.859 | 589 |
| TrustyRSM (CCD) | 2.81 | 1.78 | 16.7 | 904.994 | 885 |
| SSD agreement $(l=2)$ | 60.5 | 11.7 | $1.22 \mathrm{e}+05$ | 389.388 | 360 |
| SSD weighted $(l=2)$ | 63.8 | 16.9 | $2.71 \mathrm{e}+04$ | 415.315 | 368 |
| SSD $(l=2)$ | 56.8 | 12.2 | $9.24 \mathrm{e}+04$ | 389.254 | 364.5 |
| SCD active subspace | 423 | 110 | $8.62 \mathrm{e}+05$ | 281.069 | 235 |
| SCD | 106 | 20 | $1.53 \mathrm{e}+05$ | 396.239 | 333 |
| GD | 412 | 40.9 | $1.37 \mathrm{e}+06$ | 304.141 | 270.5 |

Table 4.4: High noise ( $\sigma=10$ ) result for the Wood test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 12.6 | 9.59 | 279 | 541.96 | 547 |
| TrustyRSM (CCD) | 18.2 | 15.4 | 498 | 723.618 | 729 |
| SSD agreement $(l=2)$ | 287 | 123 | $3.27 \mathrm{e}+05$ | 200.76 | 187 |
| SSD weighted $(l=2)$ | 343 | 133 | $5.49 \mathrm{e}+05$ | 208.402 | 200 |
| SSD $(l=2)$ | 291 | 126 | $3.76 \mathrm{e}+05$ | 202.462 | 187 |
| SCD active subspace | 597 | 224 | $1.53 \mathrm{e}+06$ | 175.25 | 150 |
| SCD | 199 | 57.7 | $3.11 \mathrm{e}+05$ | 229.143 | 210 |
| GD | 570 | 176 | $1.75 \mathrm{e}+06$ | 192.856 | 166 |

## Uncoupled Rosenbrock test function (10 dim)

The results for the Uncoupled Rosenbrock test function in 10 dimensions can be seen in Tables 4.5 to 4.8 . From the first table with no noise we can see that the trusty RSM with the LHD design performs best. However, the trusty RSM with the CCD design performs the worst since it have both the highest MSE and require the most function evaluations.

Looking at the noisy tables we can now see that the descent methods again require a lower number of function evaluations compared to the trusty RSM methods, especially compared to the trusty RSM with the CCD design. Again, this difference increase with the noise.

From all these tables we also see that the trusty RSM methods require many function evaluations for this test function. The reason for this is partly because of the minimum number of observations required to fit the second order model, as described in equation (2.9), which scaled with the dimension as $\mathcal{O}\left(m^{2}\right)$.

Table 4.5: No noise ( $\sigma=0$ ) result for the 10 -dimensional uncoupled Rosenbrock test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 0.002 | 0.0016 | $6.12 \mathrm{e}-06$ | 13454.3 | 13401 |
| TrustyRSM (CCD) | 0.0669 | 0.0122 | 0.0402 | 162183 | 159516 |
| SSD agreement $(l=2)$ | 0.0127 | 0.0127 | 0.000163 | 62136.9 | 58024 |
| SSD weighted $(l=2)$ | 0.0127 | 0.0127 | 0.000162 | 99544.8 | 89688 |
| SSD $(l=2)$ | 0.0127 | 0.0128 | 0.000163 | 59921.8 | 56449 |
| SCD active subspace | 0.012 | 0.0123 | 0.000147 | 63353.9 | 65249 |
| SCD | 0.0129 | 0.0128 | 0.000169 | 62716 | 58476 |
| GD | 0.0121 | 0.0118 | 0.000147 | 17987.7 | 17739.5 |

Table 4.6: Low noise ( $\sigma=0.1$ ) result for the 10 -dimensional uncoupled Rosenbrock test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 1.02 | 0.672 | 2.05 | 1890.6 | 1877 |
| TrustyRSM (CCD) | 1.08 | 0.816 | 1.92 | 42474.9 | 38703 |
| SSD agreement $(l=2)$ | 7.44 | 1.48 | 650 | 870.319 | 865 |
| SSD weighted $(l=2)$ | 15.6 | 4.82 | $1.35 \mathrm{e}+03$ | 1071.65 | 1041.5 |
| SSD $(l=2)$ | 7.01 | 1.38 | 605 | 856.328 | 833 |
| SCD active subspace | 48.8 | 30.7 | $6.22 \mathrm{e}+03$ | 419.645 | 342.5 |
| SCD | 9.49 | 3.73 | 615 | 722.508 | 721.5 |
| GD | 49.8 | 9.59 | $1.01 \mathrm{e}+04$ | 820.058 | 767 |

Table 4.7: Medium noise ( $\sigma=1$ ) result for the 10-dimensional uncoupled Rosenbrock test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 3.04 | 2.17 | 19 | 1784.27 | 1743 |
| TrustyRSM (CCD) | 1.33 | 1.09 | 2.65 | 37169.6 | 36611 |
| SSD agreement $(l=2)$ | 40.6 | 24.2 | $3.98 \mathrm{e}+03$ | 367.811 | 344 |
| SSD weighted $(l=2)$ | 53.4 | 35.7 | $5.76 \mathrm{e}+03$ | 410.435 | 366 |
| SSD $(l=2)$ | 44.1 | 26.8 | $4.56 \mathrm{e}+03$ | 363.67 | 344 |
| SCD active subspace | 55.4 | 37.4 | $7.28 \mathrm{e}+03$ | 246.434 | 219 |
| SCD | 23.8 | 9.04 | $2.35 \mathrm{e}+03$ | 476.162 | 475 |
| GD | 77.9 | 42.9 | $1.41 \mathrm{e}+04$ | 422.298 | 378 |

Table 4.8: High noise $(\sigma=10)$ result for the 10 -dimensional uncoupled Rosenbrock test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 36 | 21.5 | $3.34 \mathrm{e}+03$ | 1608.46 | 1609 |
| TrustyRSM (CCD) | 6.2 | 3.37 | 94.6 | 33880.9 | 33473 |
| SSD agreement $(l=2)$ | 197 | 181 | $5.03 \mathrm{e}+04$ | 128.205 | 113 |
| SSD weighted $(l=2)$ | 178 | 159 | $4.24 \mathrm{e}+04$ | 154.666 | 113 |
| SSD $(l=2)$ | 195 | 177 | $4.94 \mathrm{e}+04$ | 129.074 | 113 |
| SCD active subspace | 88.3 | 61.5 | $1.44 \mathrm{e}+04$ | 171.12 | 151 |
| SCD | 115 | 94.3 | $2 \mathrm{e}+04$ | 224.625 | 206 |
| GD | 207 | 189 | $5.46 \mathrm{e}+04$ | 176.581 | 169 |

## Powell test function

The results for the Powell test function can be seen in Tables 4.9 to 4.12 . From the first table with no noise we can see that the trusty RSM methods performs best since they have better performance measures and lower number of function evaluations.

Looking at the noisy tables we can now see that the descent methods require a lower number of function evaluations compared to the trusty RSM method for the two highest levels of noise. By looking mainly at the MSE we can also see that the descent methods have similar performance for all levels of noise, while the trusty RSM methods gets increasingly worse. The reason for this could possibly be because the Powell test function is bowl-shaped, as can be seen in Figure 3.4 and Figure 3.5, where the function gets steeper further away from the minimum. Hence, when you are sufficiently far from the
minimum, the signal to noise ratio will be roughly the same for all levels of noise since the signal is so dominant.

Table 4.9: No noise ( $\sigma=0$ ) result for the Powell test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 0.000112 | $3.15 \mathrm{e}-05$ | $7.13 \mathrm{e}-08$ | 379.924 | 358 |
| TrustyRSM $(\mathrm{CCD})$ | $9.29 \mathrm{e}-05$ | $1.13 \mathrm{e}-05$ | $6.05 \mathrm{e}-08$ | 419.21 | 339 |
| SSD agreement $(l=2)$ | 0.0071 | 0.00811 | $5.43 \mathrm{e}-05$ | 1279.4 | 1313.5 |
| SSD weighted $(l=2)$ | 0.00728 | 0.00821 | $5.77 \mathrm{e}-05$ | 2600.77 | 2622 |
| SSD $(l=2)$ | 0.00685 | 0.00802 | $5.12 \mathrm{e}-05$ | 1129.63 | 1157 |
| SCD active subspace | 0.00695 | 0.00749 | $5.07 \mathrm{e}-05$ | 5495.15 | 6073 |
| SCD | 0.00738 | 0.00812 | $5.74 \mathrm{e}-05$ | 2156.97 | 2280.5 |
| GD | 0.00743 | 0.00822 | $5.87 \mathrm{e}-05$ | 1219.5 | 1362.5 |

Table 4.10: Low noise ( $\sigma=0.1$ ) result for the Powell test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 0.083 | 0.0482 | 0.0216 | 707.209 | 694 |
| TrustyRSM (CCD) | 0.0548 | 0.0102 | 0.0201 | 855.958 | 807 |
| SSD agreement $(l=2)$ | 98.9 | 1.52 | $8.59 \mathrm{e}+05$ | 1086.01 | 945.5 |
| SSD weighted $(l=2)$ | 99.3 | 5.89 | $8.35 \mathrm{e}+05$ | 1188.41 | 1034 |
| SSD $(l=2)$ | 231 | 1.68 | $3.63 \mathrm{e}+06$ | 1031.05 | 896.5 |
| SCD active subspace | $2.83 \mathrm{e}+03$ | 270 | $6.75 \mathrm{e}+07$ | 516.181 | 425.5 |
| SCD | 303 | 9.82 | $2.21 \mathrm{e}+06$ | 635.678 | 556 |
| GD | 747 | 81 | $4.99 \mathrm{e}+06$ | 488.596 | 397 |

Table 4.11: Medium noise ( $\sigma=1$ ) result for the Wood test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 1.45 | 0.751 | 5.96 | 652.105 | 652 |
| TrustyRSM (CCD) | 0.835 | 0.195 | 4.19 | 806.662 | 781 |
| SSD agreement $(l=2)$ | 234 | 10.6 | $8.08 \mathrm{e}+06$ | 647.655 | 591.5 |
| SSD weighted $(l=2)$ | 302 | 23.9 | $4.19 \mathrm{e}+06$ | 678.861 | 622.5 |
| SSD $(l=2)$ | 303 | 11.3 | $5.42 \mathrm{e}+06$ | 630.339 | 572 |
| SCD active subspace | $3.44 \mathrm{e}+03$ | 506 | $6.94 \mathrm{e}+07$ | 401.654 | 330.5 |
| SCD | 358 | 22.3 | $2.22 \mathrm{e}+06$ | 504.607 | 458 |
| GD | $1 \mathrm{e}+03$ | 107 | $1.29 \mathrm{e}+07$ | 421.617 | 359 |

Table 4.12: High noise $(\sigma=10)$ result for the Wood test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 12 | 7.84 | 330 | 593.263 | 589 |
| TrustyRSM (CCD) | 6.53 | 2.63 | 134 | 755.468 | 755 |
| SSD agreement $(l=2)$ | 623 | 139 | $5.14 \mathrm{e}+06$ | 302.347 | 281 |
| SSD weighted $(l=2)$ | $1 \mathrm{e}+03$ | 184 | $1.28 \mathrm{e}+07$ | 303.783 | 269 |
| SSD $(l=2)$ | 635 | 123 | $6.31 \mathrm{e}+06$ | 290.758 | 279.5 |
| SCD active subspace | $4.13 \mathrm{e}+03$ | 919 | $8.73 \mathrm{e}+07$ | 251.8 | 212 |
| SCD | 682 | 118 | $8.28 \mathrm{e}+06$ | 314.056 | 275 |
| GD | $1.52 \mathrm{e}+03$ | 314 | $1.62 \mathrm{e}+07$ | 266.898 | 241 |

## Ridge test function

The results for the Ridge test function can be seen in tables 4.13 to 4.16 . From the first table with no noise we can see that the trusty RSM methods performs best. Here the trusty RSM method with LHD design method have the lower number of function evaluations while the trusty RSM method with CCD design have more evaluations but lower performance measures. Another thing to note here is that the methods using active subspace methodology outperforms their counterparts since they have slightly better performance measures and lower number of function evaluations, this is especially true for the SSD agreement method and the SCD active subspace method. This difference is expected since the active subspace method is known to work well when the function have a ridge-structure as explained in Section 2.5.6.

Looking at the noisy tables we can now see that the difference between the active subspace methods and their counterparts becomes less noticeable. By looking at the mean
we can here also notice that the SSD based descent algorithms seems more robust against runs yielding high $f^{*}$ compared to the other descent methods, since they have much lower mean values at all levels of noise.

Table 4.13: No noise ( $\sigma=0$ ) result for the Ridge test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | $4.45 \mathrm{e}-17$ | $3.89 \mathrm{e}-24$ | $4.12 \mathrm{e}-31$ | 120.335 | 117 |
| TrustyRSM (CCD) | $9.14 \mathrm{e}-22$ | $1.14 \mathrm{e}-28$ | $7.77 \mathrm{e}-40$ | 320.488 | 313 |
| SSD agreement $(l=2)$ | $2.11 \mathrm{e}-05$ | $2.33 \mathrm{e}-05$ | $5.03 \mathrm{e}-10$ | 208.661 | 195 |
| SSD weighted $(l=2)$ | $2.33 \mathrm{e}-05$ | $2.39 \mathrm{e}-05$ | $5.63 \mathrm{e}-10$ | 217.699 | 223 |
| SSD $(l=2)$ | $2.1 \mathrm{e}-05$ | $2.35 \mathrm{e}-05$ | $5.05 \mathrm{e}-10$ | 223.313 | 223 |
| SCD active subspace | $2.48 \mathrm{e}-05$ | $2.6 \mathrm{e}-05$ | $6.25 \mathrm{e}-10$ | 210.446 | 217 |
| SCD | $2.56 \mathrm{e}-05$ | $2.64 \mathrm{e}-05$ | $6.64 \mathrm{e}-10$ | 273.405 | 269 |
| GD | $2.45 \mathrm{e}-05$ | $2.49 \mathrm{e}-05$ | $6.12 \mathrm{e}-10$ | 263.851 | 269 |

Table 4.14: Low noise ( $\sigma=0.1$ ) result for the Ridge test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 0.00545 | 0.000344 | 0.000267 | 669.74 | 726 |
| TrustyRSM (CCD) | $4.02 \mathrm{e}-05$ | $1.36 \mathrm{e}-07$ | $3.41 \mathrm{e}-07$ | 339.364 | 313 |
| SSD agreement $(l=2)$ | 3.3 | 0.0142 | $6.91 \mathrm{e}+03$ | 299.66 | 290 |
| SSD weighted $(l=2)$ | 16.2 | 0.0133 | $1.08 \mathrm{e}+05$ | 296.15 | 285.5 |
| SSD $(l=2)$ | 8.35 | 0.0129 | $4.53 \mathrm{e}+04$ | 298.346 | 289.5 |
| SCD active subspace | 412 | 0.0459 | $3.34 \mathrm{e}+06$ | 230.719 | 215 |
| SCD | 384 | 0.0492 | $2.87 \mathrm{e}+06$ | 231.401 | 223 |
| GD | $1 \mathrm{e}+03$ | 19.4 | $7.52 \mathrm{e}+06$ | 211.387 | 178 |

Table 4.15: Medium noise ( $\sigma=1$ ) result for the Ridge test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 0.113 | 0.0223 | 0.067 | 756.392 | 755 |
| TrustyRSM (CCD) | 0.01 | $1.66 \mathrm{e}-05$ | 0.0039 | 1068.2 | 469 |
| SSD agreement $(l=2)$ | 15.8 | 0.175 | $1.83 \mathrm{e}+05$ | 264.934 | 255 |
| SSD weighted $(l=2)$ | 8.98 | 0.197 | $5.72 \mathrm{e}+04$ | 263.635 | 252.5 |
| SSD $(l=2)$ | 14.3 | 0.161 | $1.27 \mathrm{e}+05$ | 270.415 | 258 |
| SCD active subspace | 329 | 0.409 | $1.86 \mathrm{e}+06$ | 219.818 | 205 |
| SCD | 381 | 0.488 | $2.35 \mathrm{e}+06$ | 216.734 | 203 |
| GD | 978 | 13.6 | $6.82 \mathrm{e}+06$ | 205.682 | 180 |

Table 4.16: High noise $(\sigma=10)$ result for the Ridge test function.

| Method | Mean | Median | MSE | Mean fVals | Median fVals |
| :--- | :--- | :--- | :--- | :--- | :--- |
| TrustyRSM (LHD) | 2.55 | 0.816 | 28.7 | 746.619 | 755 |
| TrustyRSM (CCD) | 0.242 | 0.0217 | 0.662 | 2288.51 | 2263 |
| SSD agreement $(l=2)$ | 12.5 | 2.54 | $2.86 \mathrm{e}+04$ | 228.917 | 215.5 |
| SSD weighted $(l=2)$ | 15.8 | 2.89 | $1.64 \mathrm{e}+04$ | 216.013 | 206 |
| SSD $(l=2)$ | 29.4 | 2.28 | $1.46 \mathrm{e}+05$ | 231.735 | 219 |
| SCD active subspace | 355 | 3.85 | $2.48 \mathrm{e}+06$ | 202.735 | 192 |
| SCD | 455 | 4.44 | $3.06 \mathrm{e}+06$ | 199.344 | 187 |
| GD | $1.04 \mathrm{e}+03$ | 31.4 | $8.77 \mathrm{e}+06$ | 197.667 | 169 |

### 4.2 Application to Pre-Hospital Care

### 4.2.1 Median Time Performance Measure

The positioning obtained from the optimization with the median time performance measure can be seen in Figure 4.1. Here we can see that the positions of the ambulances all strive to be located the central parts of the municipal of Umeå. This behaviour is expected, since as we mentioned before, more than half of the calls are from the urban parts.

Comparing the initial location with the optimized location by running the simulation for each positioning 10 times we have that the mean of the performance measure for the initial location is 426.4 seconds, while the mean of the optimized location is 387.6 seconds. This is a 38.8 seconds, or $9.1 \%$, decrease. The table with all these 10 runs can be seen in Appendix I.1.


Figure 4.1: Optimization result when using the median time performance measure. The initial location where the ambulances are currently positioned is also included as a reference point.

### 4.2.2 90:th Percentile Time Performance Measure

The positioning obtained from the optimization with the 90:th percentile time performance measure can be seen in Figure 4.2. Here we can see that the positions of the ambulances still are fairly central, but have moved a bit further from the city centre compared to the median performance measure.

Comparing the initial location with the optimized location by running the simulation for each positioning 10 times we have that the mean of the performance measure for the initial location is 1086.2 seconds, while the mean of the optimized location is 938.1 seconds. This is a 148.1 seconds, or $13.6 \%$, decrease. The table with all these 10 runs can be seen in Appendix I.2.


Figure 4.2: Optimization result when using the 90 :th percentile time performance measure. The initial location where the ambulances are currently positioned is also included as a reference point.

### 4.2.3 Maximum DeSO Median Time Performance Measure

The positioning obtained from the optimization with the maximum DeSO median time performance measure can be seen in Figure 4.3. Here we can see that the positions of the ambulances now have moved to cover a larger part of the municipal. The biggest difference, is that the optimization now suggest that two ambulances should be moved to the northern part of the municipal. We still have one ambulance in the center of the city and the last ambulance should be located slightly south of the city. Further, we see that the two northern ambulances are quite close to each other, which might be counter intuitive. But by noting that in the case that one of those ambulances are on a mission in their respective DeSO zone and another call comes from the same zone, this closeness means that the second ambulance have the possibility to cover this new call in the zone of the first ambulance.

Comparing the initial location with the optimized location by running the simulation for each positioning 10 times we have that the mean of the performance measure for the initial location is 1567.7 seconds, while the mean of the optimized location is 1203.4 seconds. This is a 356.5 seconds, or $23.2 \%$, decrease. The table with all these 10 runs can be seen in Appendix I.3.


Figure 4.3: Optimization result when using the maximum DeSO median time performance measure. The initial location where the ambulances are currently positioned is also included as a reference point.

## Chapter 5

## Conclusion

### 5.1 Simulation study

From the simulation study we see that the trusy RSM methods performs the best. However, they scale badly when increasing the dimensions as the number of observations needed to fit the second order model scales as $\mathcal{O}\left(m^{2}\right)$, which we could see from equation (2.9).

In the simulation study we also see that there is only a small improvement of the descent based methods when using them in conjunction with the active subspace methodology, which only is apparent for the ridge function with no or little noise. The reason for this could be the way we are building our matrix $\hat{\mathbf{C}}$, which is by sampling at our current position every $q$ iterations when we compute the full gradient. If we for example let $q=5$, this means that after 50 iterations we have still only computed the full gradient, for updating $\hat{\mathbf{C}}, 10$ times. Furthermore, if we are dealing with a noisy function this few calculated full gradients might even show opposite behaviour compared to the underlying function. This will then rather prohibit instead of helping us to achieve a fast rate of convergence, since we then might have an active subspace that is not aligned with the true function, and therefore will prioritize a subspace selection that might even be orthogonal to steepest descent of our current position. A remedy for this problem could be to pre-calculate the active subspace before beginning the optimization. However, this will only be useful if we know that we will do many optimization runs since this pre-calculation itself might become costly.

In general, we could see that the trusty RSM methods performed the best over all functions and levels of noises since those handled the different types of functions and levels of noise well. The descent-based methods were worse, especially when adding more noise, which could be due to the fact that they were heavily affected by single function evaluations that contained much noise. The reason why the trusty RSM methods weren't as affected by single high-noise evaluations was probably since these methods use many function evaluations in each iteration. This lead to the noise averaging out in the fitting
of the second order model (as one would expect from the law of large numbers) since the function evaluations are unbiased.

### 5.2 Application to Pre-Hospital Care

For the application to the pre-hospital care problem of the ambulance optimization, we could find possible improvement for all three performance measures. Furthermore, could see that the current positioning is good if we only look at the median performance measure, which is the one most commonly used when looking at pre-hospital care, since we only were able to improve upon this measurement by around 30 seconds (or $6.2 \%$ ). But if we instead look at the 90 :th percentile performance measure, or the even more extreme maximum DeSO median time performance measure, there was room for more improvement. This is then speaking for the fact that the pre-hospital care is favoring people living close to the more urban parts of the municipality while it disfavours people living in the more rural areas.

### 5.3 Future work

One interesting improvement of the trusty RSM methods would be to try to project the points on a lower dimensional subspace, similar to as done with the SSD methods, as a workaround to the quadratic scaling of the number of observations needed to fit the regression model. This could possibly even be done with the active subspace approaches.

Another interesting way to improve the models could be to focus a bit more on exploration compared to exploitation for the trusty RSM models. This since we only are using the current ambulance positioning as a starting point and initialize our search from there. Even if we choose a large trust radius, the algorithm will favor point close to this initial position due to the fact that it is a relatively good positioning and thus can be seen a local minimum. This is most problematic in case of the median response time performance measure. We therefore think that the algorithm could benefit from more exploration where we in an initial phase explore the permitted space $\mathbb{X}$ to identify all the areas where a potential global minimum might reside. After identifying these promising areas, they are then exploited to achieve a more accurate estimation of the best points.

A third idea is to use the cheaper descent methods in the beginning of the optimization to come close to the optimum point. After this first phase, the algorithm then switches over to the more accurate, but also more expensive, trusty RSM method to improve upon the accuracy of the first phase.

Finally, for the ambulance optimization, the results in this report serve as a proof of concept that we can get improvements with some custom performance measures. However, the models covered here are relatively small scale, and a future project could be to extend those models to optimize the ambulance positioning for all ambulances in the
northern parts of Sweden. This is possible since the ambulance call simulation is built to handle this larger region. Also, this problem is interesting since it is more realistic due to the fact that ambulances sometimes co-operate across the borders of different municipals. A final step for this extended model would then be to validate its result by actually changing the ambulance positions in the real world temporarily and measure the improvements.

## Bibliography

[1] Pre-hospital resource optimization. [Online]. Available: www.prehospitalresursoptimering. se (visited on $06 / 11 / 2021$ ).
[2] J. Lundgren, M. Rönnqvist, and P. Värnbrand, Optimeringslära. Studentlitteratur AB, Lund, 2008, ISBN: 9789144053141.
[3] X. Zhou, "On the fenchel duality between strong convexity and lipschitz continuous gradient", 2018. arXiv: 1803.06573 [math. OC].
[4] J. Heinonen, Lectures on Lipschitz Analysis. University of Jyväskylä, 2005, ISBN: 9789513923181.
[5] G. Strang, Introduction to Linear Algebra. Cambridge Press, 2016, ISBN: 978-0-9802327-7-6.
[6] A. Gut, Probability: A Graduate Course. Springer, 2005, ISBN: 0-387-22833-0.
[7] ——, An Intermediate Course in Probability, 2nd ed. Springer, 2009, ISBN: 9781489984463.
[8] P. G. Constantine, E. Dow, and Q. Wang, "Active subspace methods in theory and practice: applications to kriging surfaces", SIAM Journal on Scientific Computing, vol. 36, no. 4, A1500-A1524, Jan. 2014. Doi: 10.1137/130916138.
[9] I. T. Dimov, Monte Carlo Methods for Applied Scientist. World Scientific, 2008, ISBN: 981-02-2329-3.
[10] R. Aliev and V. Ismailov, On the representation by bivariate ridge functions, 2016. arXiv: 1606.07940 [math. CA].
[11] P. G. Constantine, A. Eftekhari, J. Hokanson, and R. A. Ward, "A near-stationary subspace for ridge approximation", Computer Methods in Applied Mechanics and Engineering, vol. 326, pp. 402-421, Nov. 2017. DoI: 10.1016/j.cma.2017.07.038.
[12] P. G. Constantine, Z. del Rosario, and G. Iaccarino, Data-driven dimensional analysis: algorithms for unique and relevant dimensionless groups, 2017. arXiv: 1708.04303 [math.NA].
[13] D. Kozak, S. Becker, A. Doostan, and L. Tenorio, "Stochastic Subspace Descent", 2019. arXiv: 1904.01145 [math. OC].
[14] F. Mezzadri, "How to generate random matrices from the classical compact groups", 2006. arXiv: math-ph/0609050 [math-ph].
[15] G. Lan, First-order and Stochastic Optimization Methods for Machine Learning. Jan. 2020, ISBN: 978-3-030-39567-4. DOI: 10.1007/978-3-030-39568-1.
[16] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives.", Pacific Journal of Mathematics, vol. 16, no. 1, pp. 1-3, 1966. Doi: pjm/1102995080.
[17] S. Zhen-Jun, "Convergence of line search methods for unconstrained optimization", Applied Mathematics and Computation, vol. 157, no. 2, pp. 393-405, 2004. Doi: 10.1016/j.amc.2003.08.058.
[18] D. Montgomery, Design and Analysis of Experiments, International Student Version 8:th ed. John Wiley \& Sons, 2008, ISBN: 9781118097939.
[19] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization", Journal of Mechanical Design, vol. 129, no. 4, pp. 370-380, May 2006. DOI: 10.1115/1.2429697.
[20] G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar, "Supplementary material: Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting", ACM Transactions on Graphics, vol. 31, pp. 1-11, Jul. 2012. DOI: $10.1145 / 2185520.2335431$.
[21] V. R. Joseph and Y. Hung, "Orthogonal-maxmin latin hypercube designs", Statistica Sinica, vol. 18, no. 1, pp. 171-186, 2008.
[22] E. R. Van Dam, B. Husslage, D. Den Hertog, and H. Melissen, "Maximin latin hypercube designs in two dimensions", Operations Research, vol. 55, no. 1, pp. 158169, 2007. Doi: 10.1287/opre.1060.0317.
[23] R. Carlson and J. E. Carlson, "Canonical analysis of response surfaces: A valuable tool for process development", Organic Process Research Development, vol. 9, no. 3, pp. 321-330, 2005. DOi: 10.1021/op040023o.
[24] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization", SIAM Journal on Numerical Analysis, vol. 20, pp. 626-637, 1983. DOI: 10.1137/0720042.
[25] R. Reams, "Hadamard inverses, square roots and products of almost semidefinite matrices", Linear Algebra and its Applications, vol. 288, pp. 35-43, 1999. Doi: 10.1016/S0024-3795(98)10162-3.
[26] F. L. Bayisa, M. Ådahl, P. Rydén, and O. Cronie, "Large-scale modelling and forecasting of ambulance calls in northern Sweden using spatio-temporal log-Gaussian Cox processes", Spatial Statistics, vol. 39, 2020. DOI: 10.1016/S0024-3795(98) 10162-3.
[27] DeSO - Demografiska statistikområden. [Online]. Available: https://www.scb. se / hitta- statistik / regional- statistik-och- kartor / regionala- indelningar / deso---demografiska-statistikomraden (visited on $05 / 21 / 2021$ ).
[28] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, 1st ed. Springer, 2014, ISBN: 1461346916.

## Appendices

## Appendix A

## Proof Theorem 1

By contradiction. Assume the point $\mathbf{x}_{0} \in \mathbb{X}$ is a local minimum but not a global minimum, then there exists a point $\mathbf{x}^{*} \in \mathbb{X}$ such that $f\left(\mathbf{x}^{*}\right)<f\left(\mathbf{x}_{0}\right)$. Now, let $\mathbf{y}=$ $\lambda \mathbf{x}^{*}+(1-\lambda) \mathbf{x}_{0}$ where $0 \leq \lambda \leq 1$ and we know that the point $\mathbf{y} \in \mathbb{X}$ due to the convexity of the set $\mathbb{X}$. By convexity of the function $f$ and since $f\left(\mathbf{x}^{*}\right)<f\left(\mathbf{x}_{0}\right)$ we have that $f(\mathbf{y}) \leq \lambda f\left(\mathbf{x}^{*}\right)+(1-\lambda) f\left(\mathbf{x}_{0}\right)<f\left(\mathbf{x}_{0}\right)$. If we then let $\lambda \rightarrow 0$ we have that $\mathbf{y} \rightarrow \mathbf{x}_{0}$. Due to $f(\mathbf{y})<f\left(\mathbf{x}_{0}\right)$ we have a contradiction when $\mathbf{y}$ enters the neighbourhood of $\mathbf{x}_{0}$ since $\mathbf{x}_{0}$ then cannot be a local minimum point and the conclusion follows.

## Appendix B

## Proof Theorem 2

This proof is inspired by the proof of Lemma 1.2.3 in [28]. Here we are going to start off by letting $\boldsymbol{\delta}=\mathbf{y}-\mathbf{x}$, we can now express $f(\mathbf{y})$ as

$$
\begin{aligned}
f(\mathbf{y}) & =f(\mathbf{x})+\int_{0}^{1}\langle\nabla f(\mathbf{x}+t \boldsymbol{\delta}), \boldsymbol{\delta}\rangle d t \\
& =f(\mathbf{x})+\langle\nabla f(\mathbf{x}), \boldsymbol{\delta}\rangle+\int_{0}^{1}\langle\nabla f(\mathbf{x}+t \boldsymbol{\delta})-\nabla f(\mathbf{x}), \boldsymbol{\delta}\rangle d t
\end{aligned}
$$

where we in the last equality just added and subtracted $\langle\nabla f(\mathbf{x}), \boldsymbol{\delta}\rangle$. By using the triangle inequality for integrals, the Cauchy-Schwarz inequality and the Lipschitz continuous gradient property we can write

$$
\begin{aligned}
|f(\mathbf{y})-f(\mathbf{x})-\langle\nabla f(\mathbf{x}), \boldsymbol{\delta}\rangle| & =\left|\int_{0}^{1}\langle\nabla f(\mathbf{x}+t \boldsymbol{\delta})-f(\mathbf{x}), \boldsymbol{\delta}\rangle d t\right| \\
& \leq \int_{0}^{1}|\langle\nabla f(\mathbf{x}+t \boldsymbol{\delta})-f(\mathbf{x}), \boldsymbol{\delta}\rangle| d t \\
& \leq \int_{0}^{1}\|\nabla f(\mathbf{x}+t \boldsymbol{\delta})-f(\mathbf{x})\| \cdot\|\boldsymbol{\delta}\| d t \\
& \leq \int_{0}^{1} L_{2}\|\mathbf{x}+t \boldsymbol{\delta}-\mathbf{x}\| \cdot\|\boldsymbol{\delta}\| d t \\
& =\int_{0}^{1} L_{2} t\|\boldsymbol{\delta}\|^{2} d t \\
& =\frac{L_{2}\|\boldsymbol{\delta}\|^{2}}{2} .
\end{aligned}
$$

We can now see that this bound is equivalent to

$$
-\frac{L_{2}\|\boldsymbol{\delta}\|^{2}}{2} \leq f(\mathbf{y})-f(\mathbf{x})-\langle\nabla f(\mathbf{x}), \boldsymbol{\delta}\rangle \leq \frac{L_{2}\|\boldsymbol{\delta}\|^{2}}{2}
$$

and the conclusion follows by using the upper bound and the substitution of $\boldsymbol{\delta}$.

## Appendix C

## Proof Theorem 4

Since the matrix is positive semi-definite it fulfills $\mathbf{x}^{T} \mathbf{S} \mathbf{x} \geq 0$, for any vector $\mathbf{x}$. By letting $\mathbf{x}$ be the i:th eigenvector $\mathbf{q}_{i}$ of $\mathbf{S}$ we thus gets $\mathbf{q}_{i}^{T} \mathbf{S q}_{i} \geq 0$. Now, we use the spectral decomposition of $\mathbf{S}$ and our expression becomes $\mathbf{q}_{i}^{T}\left(\lambda_{1} \mathbf{q}_{1} \mathbf{q}_{1}^{T}+\cdots+\lambda_{n} \mathbf{q}_{n} \mathbf{q}_{n}^{T}\right) \mathbf{q}_{i} \geq 0$. Due to the orthogonality of the eigenvectors, which follows from the symmetry of $\mathbf{S}$, this simplifies to $\mathbf{q}_{i}^{T}\left(\lambda_{1} \mathbf{q}_{1} \mathbf{q}_{1}^{T}+\cdots+\lambda_{n} \mathbf{q}_{n} \mathbf{q}_{n}^{T}\right) \mathbf{q}_{i}=\lambda_{i} \mathbf{q}_{i}^{T} \mathbf{q}_{i} \mathbf{q}_{i}^{T} \mathbf{q}_{i}=\lambda_{i} \geq 0$ where we also used that the norm of every eigenvector is one.

## Appendix D

## Gradient of ridge function

To find the gradient of $f(\mathbf{x})=g\left(\mathbf{a}^{T} \mathbf{x}\right)$ we start by rewriting the function as $f(\mathbf{x})=$ $g(h(\mathbf{x}))$ where $h(\mathbf{x})=\mathbf{a}^{T} \mathbf{x}$. Noting that $\partial h(\mathbf{x}) / \partial x_{k}=a_{k}$ we can then apply the chain rule to find the derivative of $f(\mathbf{x})$ with respect to coordinate $x_{k}$ as

$$
\begin{aligned}
\frac{d f(\mathbf{x})}{d x_{k}} & =\frac{d g\left(\mathbf{a}^{T} \mathbf{x}\right)}{d x_{k}} \\
& =\frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial h(\mathbf{x})} \frac{\partial h(\mathbf{x})}{\partial x_{k}} \quad \text { (chain rule) } \\
& =a_{k} \frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial h(\mathbf{x})} \\
& \left.=a_{k} \frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)} \quad \text { (Using definition of } h\right)
\end{aligned}
$$

Since the term $\frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}$ is independent of which coordinate $x_{k}$ we are differentiating along, the gradient expression becomes

$$
\begin{aligned}
\nabla f(\mathbf{x}) & =\left[\frac{d f(\mathbf{x})}{d x_{1}}, \ldots, \frac{d f(\mathbf{x})}{d x_{m}}\right]^{T} \\
& =\left[a_{1} \frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}, \ldots, a_{m} \frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}\right]^{T} \\
& =\left(\frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}\right)\left[a_{1}, \ldots, a_{m}\right]^{T} \\
& =\left(\frac{\partial g\left(\mathbf{a}^{T} \mathbf{x}\right)}{\partial\left(\mathbf{a}^{T} \mathbf{x}\right)}\right) \mathbf{a} .
\end{aligned}
$$

## Appendix E

## Proof Theorem 10

Reusing the ideas from the proof of theorem 9, using standard tricks from optimization theory and with a straightforward modification to handle the stochastic subspace instead of the noise this proof is constructed as following:

By first taking the function evaluation of the scheme and using the expression of $g\left(\mathbf{x}_{k}\right)$ we get

$$
\begin{aligned}
f\left(\mathbf{x}_{k+1}\right) & =f\left(\mathbf{x}_{k}-\alpha_{k} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right. \\
& \left.\leq f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\langle\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right), \nabla f\left(\mathbf{x}_{k}\right)\right\rangle+\alpha_{k}^{2} L\left\|\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right\|^{2} \\
& \left.=f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\langle\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right), \nabla f\left(\mathbf{x}_{k}\right)\right\rangle+\alpha_{k}^{2} L \nabla f\left(\mathbf{x}_{k}\right)^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right) \\
& \left.\left.=f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\langle\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right), \nabla f\left(\mathbf{x}_{k}\right)\right\rangle+\frac{\alpha_{k}^{2} L m}{l} \nabla f\left(\mathbf{x}_{k}\right)^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right)
\end{aligned}
$$

where we have used that $f$ has Lipschitz continuous gradient in conjunction with Theorem 2 and also $\mathbf{M}_{k}^{T} \mathbf{M}_{k}=\frac{m}{l} \mathbf{I}_{m}$. Define the filtration $\mathcal{F}_{k}=\sigma\left(\mathbf{M}_{1}, \ldots, \mathbf{M}_{k-1}\right), k>1$, $\mathcal{F}_{1}=\{\emptyset, \Omega\}$. Then by conditional expectation and $E\left[\mathbf{M}_{k} \mathbf{M}_{k}^{T}\right]=\mathbf{I}_{m}$ we have

$$
\begin{aligned}
E\left[f\left(\mathbf{x}_{k+1}\right) \mid \mathcal{F}_{k}\right] & \left.\left.\leq f\left(\mathbf{x}_{k}\right)-\alpha_{k} \| \nabla f\left(\mathbf{x}_{k}\right)\right)\left\|^{2}+\frac{\alpha_{k}^{2} L m}{l}\right\| \nabla f\left(\mathbf{x}_{k}\right)\right) \|^{2} \\
& \left.=f\left(\mathbf{x}_{k}\right)-\alpha_{k} \| \nabla f\left(\mathbf{x}_{k}\right)\right) \|^{2}\left(1-\frac{\alpha_{k} L m}{l}\right) \\
& \left.\leq f\left(\mathbf{x}_{k}\right)-\frac{\alpha_{k}}{2} \| \nabla f\left(\mathbf{x}_{k}\right)\right) \|^{2}
\end{aligned}
$$

where we in the last equality assumed that $\alpha_{k} \leq \frac{l}{2 L m}$. Taking the full expectation of $f\left(\mathbf{x}_{k+1}\right)$, using that the initial value $\mathbf{x}_{0}$ is known and telescoping we get

$$
\begin{align*}
E\left[f\left(\mathbf{x}_{K+1}\right)\right] & \leq E\left[f\left(\mathbf{x}_{K}\right)\right]-\frac{\alpha_{K}}{2}\left\|\nabla f\left(\mathbf{x}_{K}\right)\right\|^{2}  \tag{E.1}\\
& \leq \ldots  \tag{E.2}\\
& \leq E\left[f\left(\mathbf{x}_{0}\right)\right]-\frac{1}{2} \sum_{k=0}^{K} \alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}  \tag{E.3}\\
& \leq f\left(\mathbf{x}_{0}\right)-\frac{1}{2} \sum_{k=0}^{K} \alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2} . \tag{E.4}
\end{align*}
$$

Normalizing both sides by the total step length $\sum_{k=0}^{K} \alpha_{k}$ and multiplying by 2 we get

$$
\begin{align*}
\frac{\sum_{k=0}^{K} \alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}}{\sum_{k=0}^{K} \alpha_{k}} & \leq \frac{2\left(f\left(\mathbf{x}_{0}\right)-E\left[f\left(\mathbf{x}_{K+1}\right)\right]\right)}{\sum_{k=0}^{K} \alpha_{k}}  \tag{E.5}\\
& \leq \frac{2\left(f\left(\mathbf{x}_{0}\right)-f^{*}\right)}{\sum_{k=0}^{K} \alpha_{k}} \tag{E.6}
\end{align*}
$$

where the last equality follow from $f^{*}=f\left(\mathbf{x}^{*}\right) \leq E\left[f\left(\mathbf{x}_{k}\right)\right], \forall k$, where $\mathbf{x}^{*}$ is the global minimizer. Using the convex combination trick as in the previous proof we now have

$$
\begin{equation*}
\min _{0 \leq k \leq K}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2} \leq \frac{2\left(f\left(\mathbf{x}_{0}\right)-f^{*}\right)}{\sum_{k=0}^{K} \alpha_{k}} . \tag{E.7}
\end{equation*}
$$

If we now let $K \rightarrow \infty$ we can use the assumption $\lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}=\infty$ to get

$$
\begin{equation*}
\lim _{K \rightarrow \infty} \min _{0 \leq k \leq K}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2} \leq 0 \tag{E.8}
\end{equation*}
$$

and the proof is complete.

## Appendix F

## Proof Theorem 11

Reusing the ideas from the proof of theorem 9, using standard tricks from optimization theory and with a straightforward modification to handle the stochastic subspace with the noise this proof is constructed as following:

By first taking the function evaluation of the scheme and using the expression of $g\left(\mathrm{x}_{k}\right)$ we get

$$
\begin{aligned}
f\left(\mathbf{x}_{k+1}\right) & =f\left(\mathbf{x}_{k}-\alpha_{k} \mathbf{M}_{k} \mathbf{M}_{k}^{T}\left(\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}\right)\right) \\
& \leq f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\langle\mathbf{M}_{k} \mathbf{M}_{k}^{T}\left(\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}\right), \nabla f\left(\mathbf{x}_{k}\right)\right\rangle+\frac{\alpha_{k}^{2} L}{2}\left\|\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)+\mathbf{M}_{k} \mathbf{M}_{k}^{T} \boldsymbol{\epsilon}_{k}\right\|^{2}
\end{aligned}
$$

where we have used that $f$ has Lipschitz continuous gradient in conjunction with Theorem 2. Now, by the triangle inequality $\|a+b\|^{2} \leq(\|a\|+\|b\|)^{2}$ and Youngs inequality $a b \leq \frac{1}{2}\left(a^{2}+b^{2}\right)$ we get that $\|a+b\|^{2} \leq 2\left(\|a\|^{2}+\|b\|^{2}\right)$ and thus

$$
\begin{aligned}
f\left(\mathbf{x}_{k+1}\right) \leq & f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\langle\mathbf{M}_{k} \mathbf{M}_{k}^{T}\left(\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}\right), \nabla f\left(\mathbf{x}_{k}\right)\right\rangle+\alpha_{k}^{2} L\left(\left\|\mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}+\left\|\mathbf{M}_{k} \mathbf{M}_{k}^{T} \boldsymbol{\epsilon}_{k}\right\|^{2}\right) \\
= & f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\langle\mathbf{M}_{k} \mathbf{M}_{k}^{T}\left(\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}\right), \nabla f\left(\mathbf{x}_{k}\right)\right\rangle+\ldots \\
& \alpha_{k}^{2} L\left(\nabla f\left(\mathbf{x}_{k}\right)^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \boldsymbol{\epsilon}_{k}\right) \\
= & f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\langle\mathbf{M}_{k} \mathbf{M}_{k}^{T}\left(\nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}\right), \nabla f\left(\mathbf{x}_{k}\right)\right\rangle+\ldots \\
& \quad \frac{\alpha_{k}^{2} L d}{l}\left(\nabla f\left(\mathbf{x}_{k}\right)^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \nabla f\left(\mathbf{x}_{k}\right)+\boldsymbol{\epsilon}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \boldsymbol{\epsilon}_{k}\right)
\end{aligned}
$$

where the last equality comes from $\mathbf{M}_{k}^{T} \mathbf{M}_{k}=\frac{{ }^{m}}{l} \mathbf{I}_{m}$. Define the filtrations $\mathcal{F}_{k}^{(1)}=$ $\sigma\left(\boldsymbol{\epsilon}_{1}, \ldots, \boldsymbol{\epsilon}_{k-1}\right), \mathcal{F}_{k}^{(2)}=\sigma\left(\mathbf{M}_{1}, \ldots, \mathbf{M}_{k-1}\right), k>1, \mathcal{F}_{1}^{(1)}=\mathcal{F}_{1}^{(2)}=\{\emptyset, \Omega\}$. Then by conditional expectation, $E\left[\mathbf{M}_{k} \mathbf{M}_{k}^{T}\right]=\mathbf{I}_{m}$, properties of $\boldsymbol{\epsilon}_{k}$ and the independence of $\mathbf{M}_{\mathbf{k}}$ and $\boldsymbol{\epsilon}_{k}$ we have

$$
\begin{aligned}
E\left[f\left(\mathbf{x}_{k+1}\right) \mid \mathcal{F}_{k}^{(1)}, \mathcal{F}_{k}^{(2)}\right] & \leq f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}+\frac{L \alpha_{k}^{2} m}{l}\left(\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}+\sigma^{2}\right) \\
& =f\left(\mathbf{x}_{k}\right)-\alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}\left(1-\frac{L \alpha_{k} m}{l}\right)+\frac{L \alpha_{k}^{2} m \sigma^{2}}{l} \\
& \leq f\left(\mathbf{x}_{k}\right)-\frac{\alpha_{k}}{2}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}+\frac{L \alpha_{k}^{2} m \sigma^{2}}{l}
\end{aligned}
$$

where we in the last equality assumed that $\alpha_{k} \leq \frac{l}{2 d L}$. Taking the full expectation of $f\left(\mathbf{x}_{k+1}\right)$, using that the initial value $\mathbf{x}_{0}$ is known and telescoping we get

$$
\begin{align*}
E\left[f\left(\mathbf{x}_{K+1}\right)\right] & \leq E\left[f\left(\mathbf{x}_{K}\right)\right]-\frac{\alpha_{K}}{2}\left\|\nabla f\left(\mathbf{x}_{K}\right)\right\|^{2}+\frac{L \alpha_{k}^{2} m \sigma^{2}}{l}  \tag{F.1}\\
& \leq \ldots  \tag{F.2}\\
& \leq E\left[f\left(\mathbf{x}_{0}\right)\right]-\frac{1}{2} \sum_{k=0}^{K} \alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}+\frac{L m \sigma^{2}}{l} \sum_{k=0}^{K} \alpha_{k}^{2}  \tag{F.3}\\
& \leq f\left(\mathbf{x}_{0}\right)-\frac{1}{2} \sum_{k=0}^{K} \alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}+\frac{L m \sigma^{2}}{l} \sum_{k=0}^{K} \alpha_{k}^{2} . \tag{F.4}
\end{align*}
$$

Normalizing both sides by the total step length $\sum_{k=0}^{K} \alpha_{k}$ and multiplying by 2 we get

$$
\begin{align*}
\frac{\sum_{k=0}^{K} \alpha_{k}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2}}{\sum_{k=0}^{K} \alpha_{k}} & \leq \frac{2\left(f\left(\mathbf{x}_{0}\right)-E\left[f\left(\mathbf{x}_{K+1}\right)\right]\right)}{\sum_{k=0}^{K} \alpha_{k}}+\frac{\frac{L m \sigma^{2}}{l} \sum_{k=0}^{K} \alpha_{k}^{2}}{\sum_{k=0}^{K} \alpha_{k}}  \tag{F.5}\\
& \leq \frac{2\left(f\left(\mathbf{x}_{0}\right)-f^{*}\right)}{\sum_{k=0}^{K} \alpha_{k}}+\frac{\frac{L m \sigma^{2}}{l} \sum_{k=0}^{K} \alpha_{k}^{2}}{\sum_{k=0}^{K} \alpha_{k}} \tag{F.6}
\end{align*}
$$

where the last equality follow from $f^{*}=f\left(\mathbf{x}^{*}\right) \leq E\left[f\left(\mathbf{x}_{k}\right)\right], \forall k$, where $\mathbf{x}^{*}$ is the global minimizer. Realizing that the left hand side is an convex combination of the form $\sum_{i} \beta_{i} c_{i}$, $\sum_{i} \beta_{i}=1, \beta_{i} \geq 0$ with the property $\min c_{i} \leq \sum_{i} \beta_{i} c_{i} \leq \max c_{i}$ we now have

$$
\begin{equation*}
\min _{0 \leq k \leq K}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2} \leq \frac{2\left(f\left(\mathbf{x}_{0}\right)-f^{*}\right)}{\sum_{k=0}^{K} \alpha_{k}}+\frac{2 \frac{L m \sigma^{2}}{l} \sum_{k=0}^{K} \alpha_{k}^{2}}{\sum_{k=0}^{K} \alpha_{k}} . \tag{F.7}
\end{equation*}
$$

If we now let $K \rightarrow \infty$ we can use the assumptions $\lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}^{2}<\infty$ and $\lim _{K \rightarrow \infty} \sum_{k=0}^{K} \alpha_{k}=$ $\infty$ we get

$$
\begin{equation*}
\lim _{K \rightarrow \infty} \min _{0 \leq k \leq K}\left\|\nabla f\left(\mathbf{x}_{k}\right)\right\|^{2} \leq 0 \tag{F.8}
\end{equation*}
$$

and the proof is complete.

## Appendix G

## Proof Theorem 13

This proof is similar to the proof showed in theorem 3.1 in [17], but we are here showing how this proof can be modified to extend the convergence result when considering the backtracking Armijo serch when considering stochastic subspaces.

Start off by dividing the step length $\alpha_{k}=\gamma \beta^{m_{k}}$ into the two cases

$$
\left\{\begin{array}{l}
K_{1}=\left\{k: \alpha_{k}=\gamma\right\} \\
K_{2}=\left\{k: \alpha_{k}<\gamma\right\}
\end{array}\right.
$$

Denote $\mathbf{f}_{k}=f\left(\mathbf{x}_{k}\right)$ and $\mathbf{g}_{k}=g\left(\mathbf{x}_{k}\right)$. We can then see that the backtracking Armijo line search condition in equation (2.6) can be rewritten as

$$
\begin{equation*}
\mathbf{f}_{k}-f\left(\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}\right) \geq-\sigma \alpha_{k} \mathbf{g}_{k}^{T} \mathbf{d}_{k} . \tag{G.1}
\end{equation*}
$$

Thus we have that

$$
\begin{cases}\mathbf{f}_{k}-f\left(\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}\right) \geq-\sigma \gamma \mathbf{g}_{k} \mathbf{d}_{k}, & \text { if } k \in K_{1} \\ \mathbf{f}_{k k}-f\left(\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}\right) \geq-\sigma \alpha_{k} \mathbf{g}_{k} \mathbf{d}_{k}, & \text { if } k \in K_{2}\end{cases}
$$

Since $m_{k}$ in $\alpha_{k}=\gamma \beta^{m_{k}}$ is chosen to be the smallest non-negative integer to fulfilling equation (G.1), we know that using the step length $\alpha_{k} / \beta$ would violate the inequality, hence

$$
\begin{equation*}
\mathbf{f}_{k}-f\left(\mathbf{x}_{k}+\frac{\alpha_{k} \mathbf{d}_{k}}{\beta}\right)<-\frac{\sigma \alpha_{k} \mathbf{g}_{k} \mathbf{d}_{k}}{\beta}, \quad k \in K_{2} . \tag{G.2}
\end{equation*}
$$

The mean value theorem states that there exists a point $\mathbf{c} \in[\mathbf{a}, \mathbf{b}]$ such that $f(\mathbf{b})-f(\mathbf{a})=$ $g(\mathbf{c})^{T}(\mathbf{b}-\mathbf{a})$. By using the mean value theorem on the left hand side of equation (G.2) where we recognize $\mathbf{a}=\mathbf{x}+\frac{\alpha_{k} \mathbf{d}_{k}}{\beta}, \mathbf{b}=\mathbf{x}_{k}$ and $\mathbf{c}=\theta_{k} \mathbf{a}+\left(1-\theta_{k}\right) \mathbf{b}=\mathbf{x}_{k}+\frac{\theta_{k} \alpha_{k} \mathbf{d}_{k}}{\beta}$, $\theta_{k}=[0,1]$, we get

$$
-g\left(\mathbf{x}_{k}+\frac{\theta_{k} \alpha_{k} \mathbf{d}_{k}}{\beta}\right)^{T} \frac{\alpha_{k} \mathbf{d}_{k}}{\beta}<-\frac{\sigma \alpha_{k} \mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\beta}
$$

Dividing both sides by $-\frac{\alpha_{k}}{\beta}$ then yields

$$
\begin{equation*}
g\left(\mathbf{x}_{k}+\frac{\theta_{k} \alpha_{k} \mathbf{d}_{k}}{\beta}\right)^{T} \mathbf{d}_{k}>\sigma \mathbf{g}_{k}^{T} \mathbf{d}_{k} \tag{G.3}
\end{equation*}
$$

Now, we want to find a lower bound of $\alpha_{k}$ for $k \in K_{2}$. Starting off, by using $\theta_{k} \in[0,1]$, we have

$$
\begin{aligned}
L \frac{\alpha_{k}\left\|\mathbf{d}_{k}\right\|^{2}}{\beta} & \geq L \frac{\theta_{k} \alpha_{k}\left\|\mathbf{d}_{k}\right\|^{2}}{\beta} \\
& =L \frac{\theta_{k} \alpha_{k} \mathbf{d}_{k}}{\beta} \cdot\left\|\mathbf{d}_{k}\right\| \\
& \geq\left\|g\left(\mathbf{x}_{k}+\frac{\theta_{k} \alpha_{k} \mathbf{d}_{k}}{\beta}\right)-\mathbf{g}_{k}\right\| \cdot\left\|\mathbf{d}_{k}\right\|
\end{aligned}
$$

where we used the lipschitz continuous property of the gradient in the last inequality. By first using the Cauchy-Schwarz inequality $(\|\mathbf{x}\| \cdot\|\mathbf{y}\| \geq\langle\mathbf{x}, \mathbf{y}\rangle)$ and then using equation (G.3) we get

$$
\begin{aligned}
L \frac{\alpha_{k}\left\|\mathbf{d}_{k}\right\|^{2}}{\beta} & \geq\left(g\left(\mathbf{x}_{k}+\frac{\theta_{k} \alpha_{k} \mathbf{d}_{k}}{\beta}\right)-\mathbf{g}_{k}\right)^{T} \mathbf{d}_{k} \\
& >-(1-\sigma) \mathbf{g}_{k}^{T} \mathbf{d}_{k}
\end{aligned}
$$

Rewriting this expression then yield the lower bound of $\alpha_{k}$ as

$$
\begin{equation*}
\alpha_{k} \geq-\frac{\beta(1-\sigma) \mathbf{g}_{k}^{T} \mathbf{d}_{k}}{L\left\|\mathbf{d}_{k}\right\|^{2}}, \quad k \in K_{2} \tag{G.4}
\end{equation*}
$$

Using this lower bound and by using $\gamma=-\frac{\mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|^{2}}$ we have

$$
\begin{cases}\mathbf{f}_{k}-f\left(\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}\right) \geq \sigma\left(\frac{\mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|}\right)^{2}, & \text { if } k \in K_{1} \\ \mathbf{f}_{k}-f\left(\mathbf{x}_{k}+\alpha_{k} \mathbf{d}_{k}\right) \geq \frac{\beta \sigma(1-\sigma)}{L}\left(\frac{\mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|}\right)^{2}, & \text { if } k \in K_{2}\end{cases}
$$

By defining $\eta=\min \left(\sigma, \frac{\beta \sigma(1-\sigma)}{L}\right)$ we have

$$
\begin{equation*}
\mathbf{f}_{k}-\mathbf{f}_{k+1} \geq \eta\left(\frac{\mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|}\right)^{2}, \quad \forall k \tag{G.5}
\end{equation*}
$$

If we now use the stochastic subspace descent as $\mathbf{d}_{k}=-\mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}$ we get

$$
\begin{aligned}
\mathbf{f}_{k+1} & \leq \mathbf{f}_{k}-\eta\left(\frac{\mathbf{g}_{k}^{T} \mathbf{d}_{k}}{\left\|\mathbf{d}_{k}\right\|}\right)^{2} \\
& =\mathbf{f}_{k}-\eta \frac{\left(\mathbf{g}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}\right)^{2}}{\left\|\mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}\right\|^{2}} \\
& =\mathbf{f}_{k}-\eta \frac{\left(\mathbf{g}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}\right)^{2}}{\mathbf{g}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}} \\
& =\mathbf{f}_{k}-\frac{\eta l}{d} \cdot \frac{\left(\mathbf{g}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}\right)^{2}}{\mathbf{g}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}} \\
& =\mathbf{f}_{k}-\frac{\eta l}{d} \cdot\left(\mathbf{g}_{k}^{T} \mathbf{M}_{k} \mathbf{M}_{k}^{T} \mathbf{g}_{k}\right)
\end{aligned}
$$

where we in the second to last equality have used that $\mathbf{M}_{k}^{T} \mathbf{M}_{k}=\frac{d}{l} \mathbf{I}_{l}$. Now, define the filtration $\mathcal{F}_{k}=\sigma\left(\mathbf{M}_{1}, \ldots, \mathbf{M}_{k-1}\right), k>1, \mathcal{F}_{1}=\{\emptyset, \Omega\}$. Then by conditional expectation and $E\left[\mathbf{M}_{k} \mathbf{M}_{k}^{T}\right]=\mathbf{I}_{m}$ we have

$$
E\left[\mathbf{f}_{K+1} \mid \mathcal{F}_{K}\right] \leq \mathbf{f}_{K}-\frac{\eta l}{m}\left\|\mathbf{g}_{K}\right\|^{2} .
$$

Taking the full expectation, using that that $\mathbf{f}^{*} \leq E\left[\mathbf{f}_{K+1}\right]$ and that $\mathbf{x}_{0}$ is known we have by telescoping that

$$
\begin{aligned}
\mathbf{f}^{*} & \leq E\left[\mathbf{f}_{K+1}\right] \\
& \leq E\left[\mathbf{f}_{k}\right]-\frac{\eta l}{m}\left\|\mathbf{g}_{k}\right\|^{2} \\
& \leq \ldots \\
& \leq \mathbf{f}_{0}-\frac{\eta l}{m} \sum_{k=0}^{K}\left\|\mathbf{g}_{k}\right\|^{2} .
\end{aligned}
$$

Rearranging and letting $K \rightarrow \infty$ we it follows from $\mathbf{f}_{0}-\mathbf{f}^{*}<\infty$ that

$$
\lim _{K \rightarrow \infty} \sum_{k=0}^{K}\left\|\mathbf{g}_{k}\right\|^{2} \leq \frac{m}{\eta l}\left(\mathbf{f}_{0}-\mathbf{f}^{*}\right)<\infty
$$

from which we see that $\lim _{K \rightarrow \infty}\left\|\mathbf{g}_{K}\right\|^{2}=0$, and the proof is complete.

## Appendix H

## Proof Theorem 14

We have

$$
g\left(\tau \mathbf{p}^{S}\right)=\left(\tau \mathbf{p}^{S}\right)^{T} \mathbf{b}+\left(\tau \mathbf{p}^{S}\right)^{T} \mathbf{B}\left(\tau \mathbf{p}^{S}\right)
$$

which we can differentiate with respect to $\tau$ to obtain

$$
g^{\prime}\left(\tau \mathbf{p}^{S}\right)=\left(\mathbf{p}^{S}\right)^{T} \mathbf{b}+\tau\left(\mathbf{p}^{S}\right)^{T} \mathbf{B}\left(\mathbf{p}^{S}\right)=-\frac{\mathbf{b}^{T} \mathbf{g}}{\|\mathbf{g}\|}+\tau \frac{\mathbf{g}^{T} \mathbf{B} \mathbf{g}}{\|\mathbf{g}\|^{2}}=-\frac{\|\mathbf{g}\|^{2}}{\|\mathbf{g}\|}+\tau \frac{\mathbf{g}^{T} \mathbf{B g}}{\|\mathbf{g}\|^{2}} .
$$

Case $1\left(\mathbf{g}^{T} \mathbf{B g} \leq 0\right)$ : Here we see that $m^{\prime}\left(\tau \mathbf{p}^{S}\right)$ is non-increasing, and to minimize $\mathbf{p}^{C}$ we thus want to pick $\tau$ as large as possible. Hence $\tau=\Delta$.
Case $2\left(\mathbf{g}^{T} \mathbf{B g}>0\right)$ : Solving $m^{\prime}\left(\tau \mathbf{p}^{S}\right)=0$ for $\tau$ gives

$$
\tau=\frac{\|\mathbf{g}\|^{3}}{\mathbf{g}^{T} \mathbf{B g}}
$$

But since we are limited to the trust region, we let we can safeguard ourselves by setting

$$
\tau=\min \left\{\Delta, \frac{\|\mathbf{g}\|^{3}}{\mathbf{g}^{T} \mathbf{B g}}\right\} .
$$

## Appendix I

## Tables - Application to Pre-Hospital Care

## I. 1 Median Time Performance Measure

Table I.1: Median times for original and optimized positioning for 10 runs, using the median time performance measure.

| Original positioning times $(\mathbf{s})$ | Optimized positioning times (s) |
| :--- | :--- |
| 426.7 | 387.6 |
| 426.7 | 388.1 |
| 426.7 | 387.3 |
| 426.7 | 388.5 |
| 427.6 | 387.5 |
| 426.7 | 387.5 |
| 426.7 | 387.5 |
| 426.7 | 387.4 |
| 426.7 | 387.5 |
| 427.4 | 387.5 |

## I. 2 90:th Percentile Time Performance Measure

Table I.2: Median times for original and optimized positioning for 10 runs, using the 90 :th percentile time performance measure.

| Original positioning times $\mathbf{( s )}$ | Optimized positioning times $\mathbf{( s )}$ |
| :--- | :--- |
| 1076.2 | 939.2 |
| 1087.5 | 939.2 |
| 1090.6 | 936.9 |
| 1081.1 | 940.7 |
| 1090.6 | 939.2 |
| 1079.8 | 939.0 |
| 1087.6 | 940.7 |
| 1085.8 | 927.9 |
| 1079.8 | 939.2 |
| 1088.8 | 938.9 |

## I. 3 Maximum DeSO Median Time Performance Measure

Table I.3: Median times for original and optimized positioning for 10 runs, using the maximum DeSO median time performance measure.

| Original positioning times (s) | Optimized positioning times (s) |
| :--- | :--- |
| 1695.7 | 1211.2 |
| 1559.1 | 1202.5 |
| 1559.1 | 1202.5 |
| 1695.7 | 1202.5 |
| 1695.7 | 1202.5 |
| 1559.1 | 1202.5 |
| 1550.7 | 1202.5 |
| 1695.7 | 1202.5 |
| 1559.1 | 1202.5 |
| 1550.7 | 1202.5 |

